Aufbau von Mehrmandantenarchitekturen für agentische KI auf AWS

AWS Präskriptive Leitlinien

AWS Präskriptive Leitlinien: Aufbau von Mehrmandantenarchitekturen für agentische KI auf AWS

Table of Contents

Aufbau von Mehrmandantenarchitekturen für agentische KI auf AWS

Aaron Sempf und Tod Golding, Amazon Web Services

Juli 2025 (Geschichte der Dokumente)

Agentic AI stellt einen disruptiven Paradigmenwechsel dar, der Unternehmen dazu zwingt, zu überdenken, wie sie ihre Systeme aufbauen, bereitstellen und betreiben sollen. Das Agentenmodell sieht vor, dass Teams nach neuen Wegen suchen, Systeme in einen oder mehrere Agenten zu zerlegen, die neue Wege, Möglichkeiten und Werte schaffen.

Ein Großteil der Diskussionen auf Agentenebene dreht sich um die Tools, Frameworks und Muster, die zur Erstellung und Implementierung von Agenten verwendet werden. Wir müssen nicht nur gute Tools zur Erstellung von Agenten einsetzen, sondern auch neue Integrationsprotokolle, Authentifizierungsstrategien und Erkennungsmechanismen, die als Grundlage für Agentenarchitekturen dienen können.

Während die Zahl der Agententools wächst, müssen Teams auch berücksichtigen, wie ihre Agenten traditionellere Architekturprobleme angehen. Skalierbarkeit, Noisy Neighbor, Resilienz, Kosten und betriebliche Effizienz sind grundlegende Themen, die bei der Entwicklung, Erstellung und Bereitstellung von Agenten berücksichtigt werden müssen. Unabhängig davon, wie autonom und intelligent Agenten sein mögen, müssen wir auch sicherstellen, dass sie Skaleneffekte, Effizienz und Agilität erzielen, die den Geschäftsanforderungen entsprechen.

Ziel dieses Leitfadens ist es, verschiedene Dimensionen der Präsenz von Agenten zu untersuchen. Dazu gehören die Untersuchung verschiedener Muster bei der Bereitstellung und Nutzung von Agenten und die Erläuterung verschiedener Strategien für die Erstellung von Agenten, die architektonischen Zielen entsprechen. Es bedeutet auch, zu untersuchen, wie Agenten in einer Umgebung mit mehreren Mandanten genutzt werden könnten, indem interne Konstrukte eingeführt werden, die normalerweise in einer Umgebung mit mehreren Mandanten erforderlich sind.

Zielgruppe

Dieser Leitfaden richtet sich an Architekten, Entwickler und Technologieführer, die KI-gestützte Multi-Tenant-Systeme aufbauen möchten.

Ziele

Dieser Leitfaden hilft Ihnen bei folgenden Aufgaben:

- Machen Sie sich mit der Bereitstellung von Agenten für mehrere Mandanten vertraut, untersuchen Sie sowohl isolierte als auch gepoolte Modelle und erfahren Sie, wie sich der Mandantenkontext auf die Implementierung von Agenten auswirkt
- Erfahren Sie mehr über das Agentenmanagement, einschließlich Onboarding, Mandantenisolierung und Ressourcenmanagement in Umgebungen mit einem oder mehreren Anbietern
- Evaluieren Sie Aspekte von Multi-Tenant-Agenten, einschließlich Datenbesitz, Überwachung und Tests

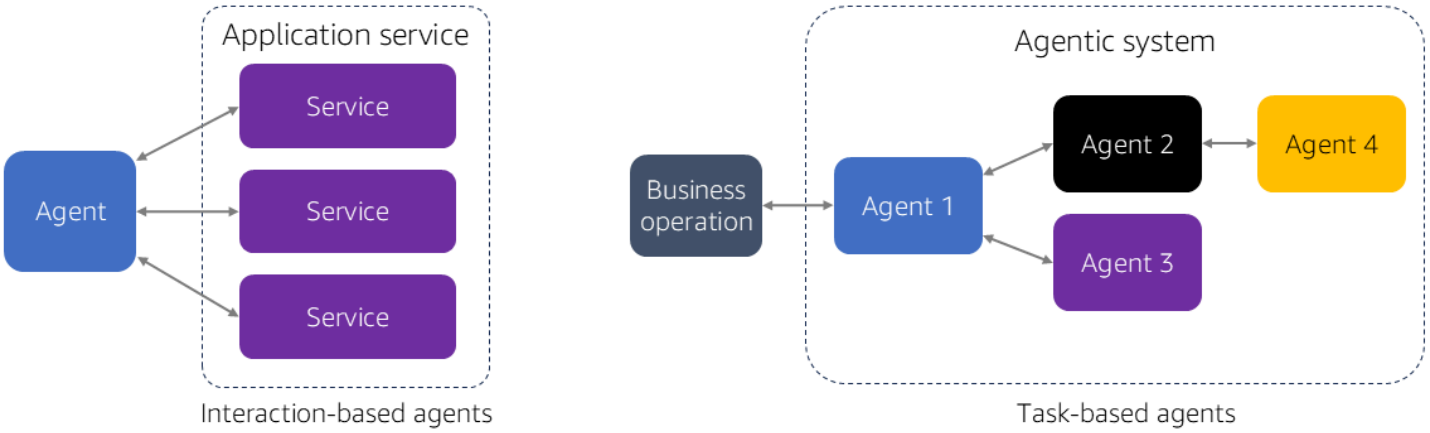
Über diese Inhaltsserie

Dieser Leitfaden ist Teil einer Reihe über agentic AI on. AWS Weitere Informationen und die anderen Leitfäden dieser Reihe finden Sie unter Agentic AI auf der Prescriptive Guidance-Website. AWS

Grundlagen für Agenten

Bevor wir uns mit architektonischen Details befassen, sollten wir die verschiedenen Rollen skizzieren, die Agenten spielen, da „Agent“ ein überladener Begriff ist, der auf viele Anwendungsfälle angewendet werden kann. Lassen Sie uns mit einigen allgemeinen Begriffen beginnen, die helfen können, sie zu kategorisieren.

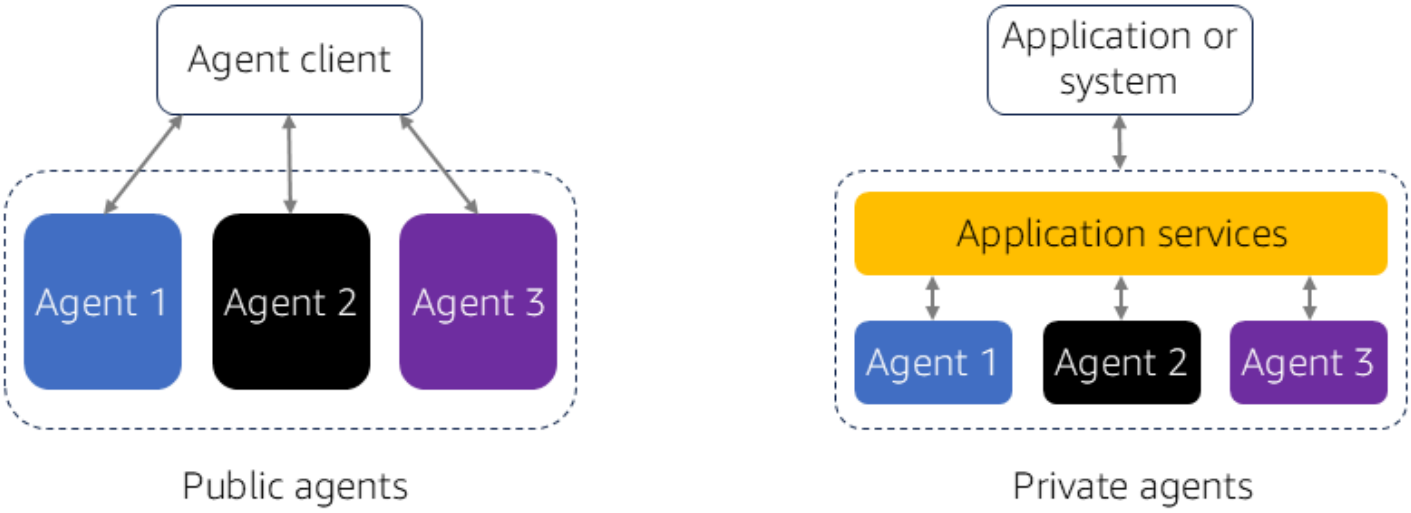
Auf der äußersten Ebene müssen wir damit beginnen, die Rolle und Art der Agenten zu klassifizieren. Dies ist eine Herausforderung, da es eine Vielzahl von Szenarien gibt, in denen Agenten für eine beliebige Anzahl von Problemen eingesetzt werden können. In dieser Diskussion konzentrieren wir uns jedoch darauf, was es bedeutet, einen Agenten in eine Anwendung oder ein System einzuführen. In diesem Modell legen wir den Schwerpunkt darauf, wie und wo Agenten die Erfahrung Ihres Systems am besten bereichern können. Die von Ihnen ausgewählten Optionen beeinflussen, wie Ihre Agenten aufgebaut, integriert und auf verschiedene Domänen und Anwendungsfälle angewendet werden. Das folgende Diagramm zeigt zwei Agentenmuster, die Builder verwenden.



Auf der linken Seite des Diagramms befindet sich ein interaktionsbasierter Agent. In diesem Modus erstellt ein Agent einen Einblick in ein vorhandenes System, um Interaktionen mit den zugrunde liegenden Diensten zu orchestrieren, um ein Ziel oder Ergebnis zu erreichen. Entscheidend ist, dass der Agent einem System als alternative Methode zur Steuerung der Funktionen und Fähigkeiten des Systems hinzugefügt wird. Stellen Sie sich zum Beispiel vor, dass ein unabhängiger Softwareanbieter (ISV) über ein Buchhaltungssystem mit Benutzererfahrung verfügt, das zur Ausführung von Vorgängen verwendet wird. Der interaktionsbasierte Agent vereinfacht die Interaktion mit diesen vorhandenen Funktionen. Es geht weniger darum zu lernen, wie man ein grob definiertes Ziel erreicht, sondern vielmehr darum, eine Möglichkeit zur Orchestrierung bekannter Pfade bereitzustellen.

Im Gegensatz dazu stellt das aufgabenbasierte System auf der rechten Seite des Diagramms einen anderen Ansatz dar. Die Agenten in diesem System nutzen ihr Wissen und ihre Fähigkeiten, um zu lernen, Aufgaben zu erledigen und Geschäftsergebnisse zu erzielen. Man könnte argumentieren, dass beide Modelle zu Geschäftsergebnissen führen, aber ein aufgabenbasiertes Modell stützt sich darauf, dass die Agenten selbst bestimmen, wie ein Ergebnis erzielt werden kann. Solche Agenten sind weniger deterministisch und verlassen sich stattdessen auf ihre Fähigkeit, zu lernen und sich weiterzuentwickeln. Im Gegensatz dazu sind interaktionsbasierte Agenten hauptsächlich darauf ausgelegt, eine Reihe bekannter Funktionen zu orchestrieren. Diese Unterschiede wirken sich darauf aus, wie Sie Agenten zur Unterstützung Ihres Unternehmens erstellen, einsetzen und integrieren.

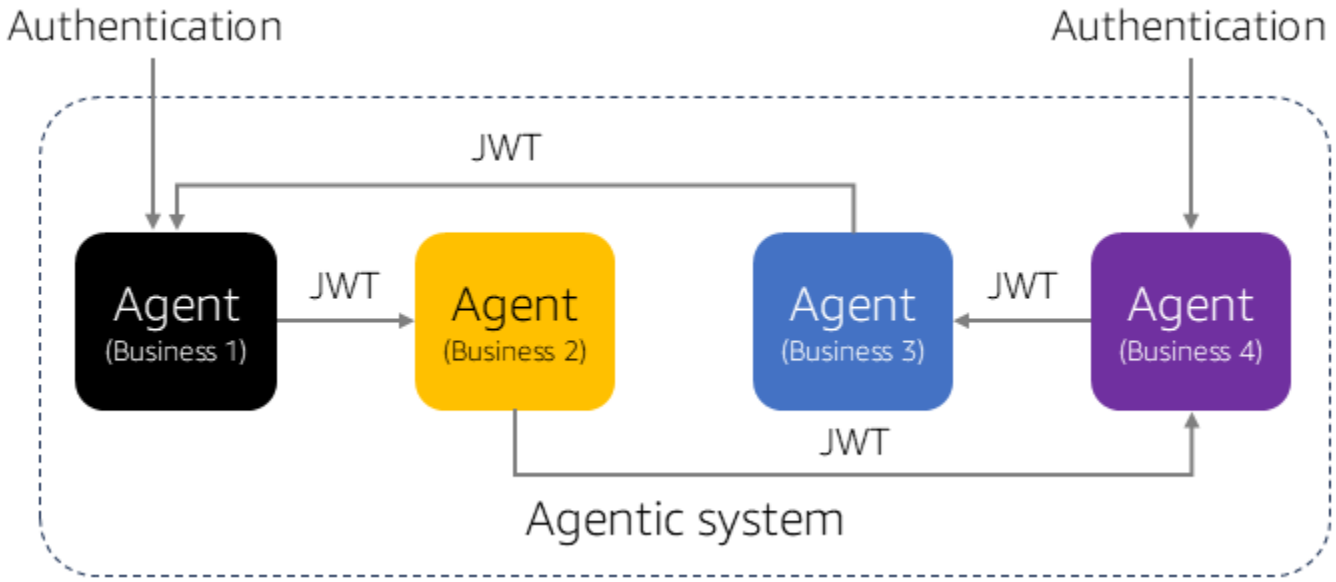
Außerdem benötigen wir Begriffe, die beschreiben, wie und wo wir Agenten einsetzen. Wo sich ein Agent innerhalb Ihres Systems befindet, kann sich darauf auswirken, wie das System aufgebaut, in welchem Umfang es sich befindet und wie es gesichert wird. Das folgende Diagramm skizziert zwei unterschiedliche Modelle, die auf Agenten angewendet werden könnten.



Auf der linken Seite des Diagramms befindet sich ein Bereitstellungssystem mit drei verschiedenen Agenten. Die Agenten sind externen Clients ausgesetzt, bei denen es sich um andere Agenten oder Anwendungen handeln kann. Bei diesem Modell werden Agenten als öffentliche Agenten bezeichnet.

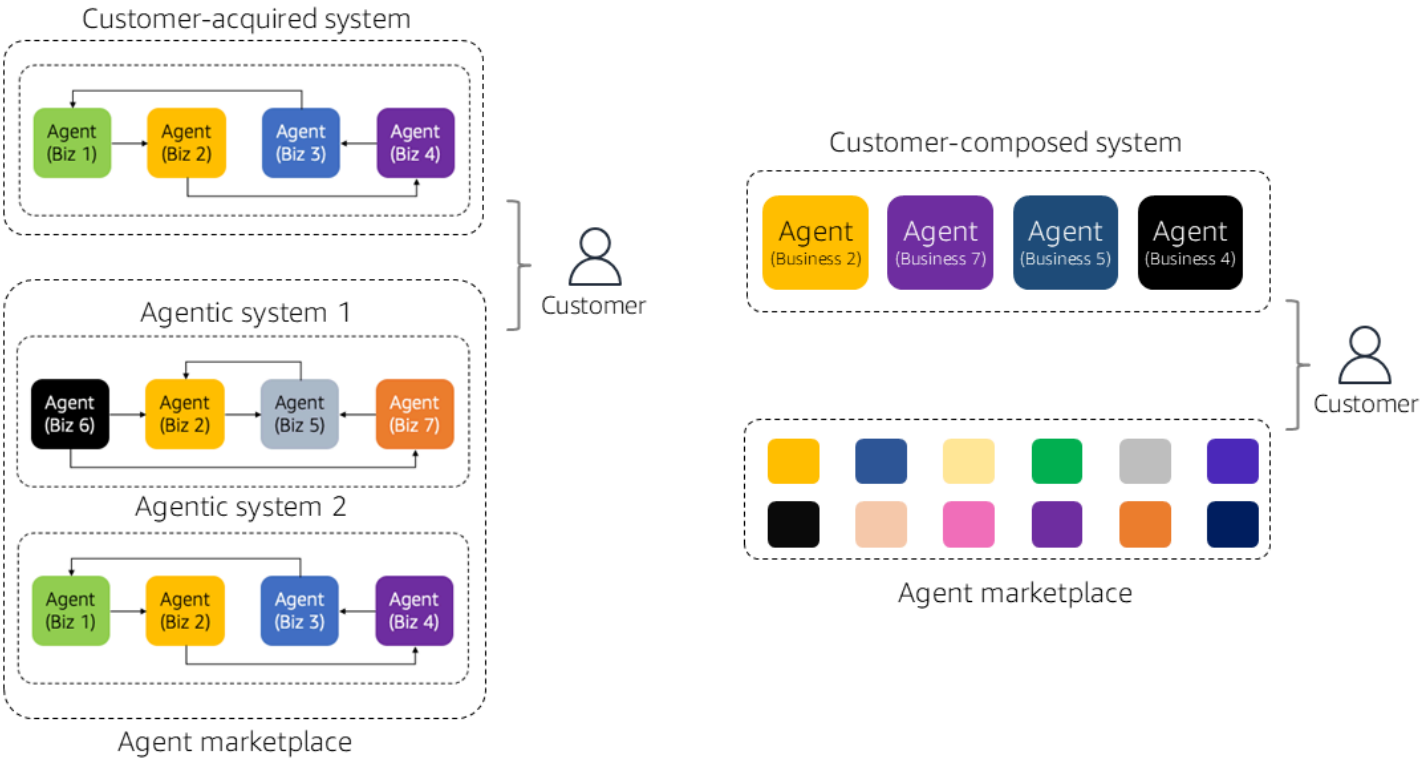
Im Gegensatz dazu zeigt das Diagramm auf der rechten Seite Agenten innerhalb der Implementierung der Lösung. In diesem Fall gibt es eine Reihe von Anwendungsdiensten, die von Benutzern oder Systemen genutzt werden. Diese Benutzer interagieren mit der Anwendung, ohne sich bewusst zu sein, dass Agenten Teil des Erlebnisses sind. Die Agenten werden dann von den Diensten des zugrunde liegenden Systems aufgerufen und orchestriert. Auf diese Weise eingesetzte Agenten werden als private Agenten bezeichnet.

Ein Großteil des Nutzens eines Agenten konzentriert sich auf das öffentliche Modell, bei dem Anbieter ihre Agenten möglicherweise veröffentlichen, um sie mit anderen Agenten von Drittanbietern zu integrieren. Die Agenten wären dann Teil eines Geflechts oder Netzes miteinander verbundener Dienste, die zusammen in der Lage sind, viele Anwendungsfälle abzudecken. Diese Agenten könnten zwar in vielen Bereichen eingesetzt werden, der business-to-business Anwendungsfall ist jedoch naheliegend. Das folgende Diagramm bietet einen konzeptuellen Überblick darüber, wie es aussehen könnte, ein Inkasso-Agent zusammenzustellen, das ein bestimmtes Problem löst.



Das Diagramm zeigt vier Geschäftsagenten, die zusammenarbeiten, um eine Reihe von Zielen zu erreichen. Wenn Agenten auf diese Weise zusammengesetzt sind, stellen sie ein Agentensystem dar, und es gibt viele Varianten solcher Systeme. Sie könnten ein vorkonfiguriertes Set von zusammenarbeitenden Agenten sein, die üblicherweise als eine Einheit verwendet werden. Oder das System könnte dynamisch von Kunden zusammengestellt werden, die eine Kombination von Agenten auswählen möchten, die ihren Bedürfnissen am besten entspricht.

Beide Ansätze bieten praktikable Möglichkeiten für die Integration von Agenten. Manche Agenten werden mit der Erwartung entwickelt, dass sie in spezifische Systeme integriert werden, in denen sie ihren Wert, ihre Reichweite und ihre Wirkung maximieren können. Diese Vorstellung von Agentensystemen wirft auch Fragen darüber auf, wie Agenten erworben werden, und es könnte viele Möglichkeiten geben, dieses Problem zu lösen. Das folgende Diagramm zeigt anhand von Beispielen, wie diese Agenten und Systeme durch Transaktionserfahrungen geschaffen werden können.

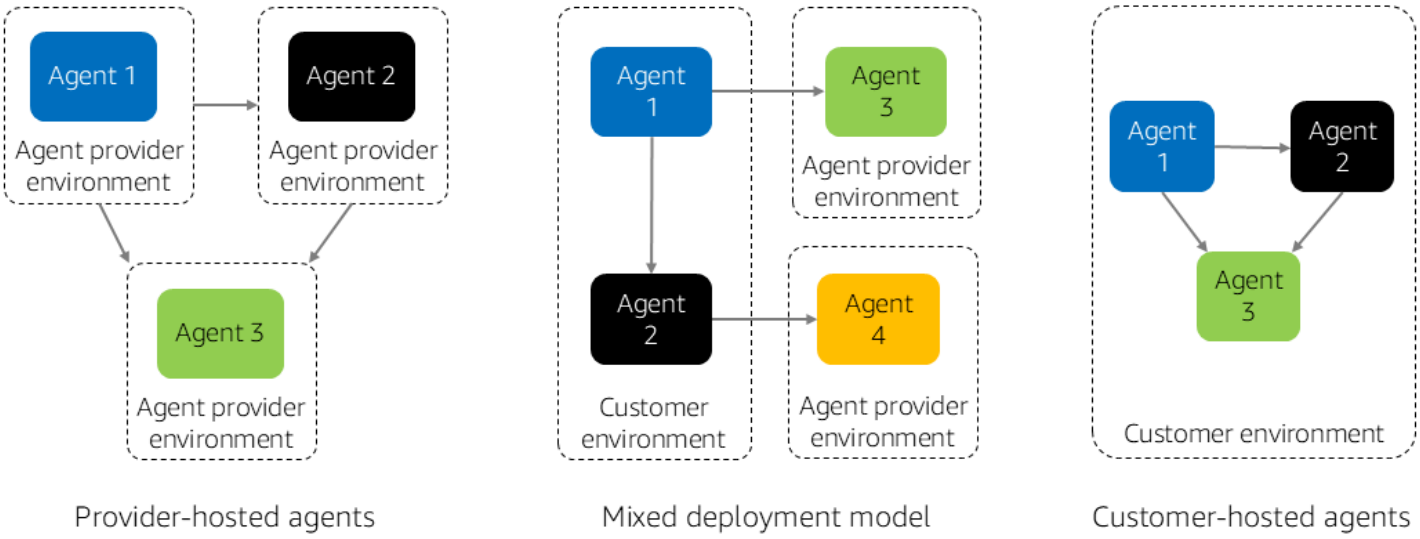


Es werden zwei Beispiele für Markterfahrungen gezeigt. Auf der linken Seite wird ein Marktplatz genutzt, um vorgefertigte Systeme zu erwerben. In diesem Szenario entdeckt und integriert der Marktplatz Systeme, die umfassendere Ziele verfolgen, die die Integration und Orchestrierung mehrerer Agenten erfordern.

Das Beispiel auf der rechten Seite zeigt einen Marktplatz, auf dem Agenten entdeckt und zu Agentensystemen zusammengefasst werden. In diesem Szenario können Kunden ein beliebiges System kompatibler, integrierter Agenten zusammenstellen, das ihren Anforderungen entspricht. Ob Agenten auf diese Weise zusammengestellt werden können, hängt vom Kompatibilitätsmodell und den Integrationsanforderungen der einzelnen Agenten ab.

Überlegungen zum Agenten-Hosting

Nachdem Sie sich mit den umfassenderen Agentenkonzepten vertraut gemacht haben, wollen wir nun besprechen, was es bedeutet, diese Agenten zu hosten und auszuführen. Wir müssen darüber nachdenken, wie und wo Berechnungen ausgeführt werden, wie sie skaliert werden, wie sie funktionieren und wie sie verwaltet werden. Gleichzeitig werden einige Muster, von denen wir erwarten, dass sie als Akteure auftreten werden, in größerem Umfang angewendet und übernommen. Das folgende Diagramm zeigt ein Beispiel für wahrscheinliche Permutationen.



Drei verschiedene Strategien werden hier vorgestellt. Auf der linken Seite des Diagramms sehen Sie ein Modell, bei dem unsere Agenten in den Umgebungen der einzelnen Agentenanbieter gehostet, skaliert und verwaltet werden. Diese Agenten werden als Dienste veröffentlicht und genutzt. Sie arbeiten nach einem sogenannten Agent-as-a-Service (AaaS) -Modell. Auf der rechten Seite befindet sich ein Modell, bei dem die Agenten eines Anbieters alle in einer speziellen Kundenumgebung gehostet werden.

In der Mitte des Diagramms befindet sich ein gemischtes Bereitstellungsmodell, das diese beiden Strategien kombiniert. Dabei werden einige Agenten lokal in der Kundenumgebung gehostet und mit einigen Agenten interagiert, die remote in der Umgebung eines Anbieters gehostet werden.

Eine vierte Option (nicht dargestellt) könnte darin bestehen, dass Agenten als Dienste mit geringem oder ohne Code erstellt werden, die durch Agenteninfrastrukturdienste skaliert und verwaltet werden. Wir werden diese nicht im Detail behandeln, da die Architektur und das Hosting der verwalteten Agenten in erster Linie von der Organisation bestimmt werden, der die Dienste gehören.

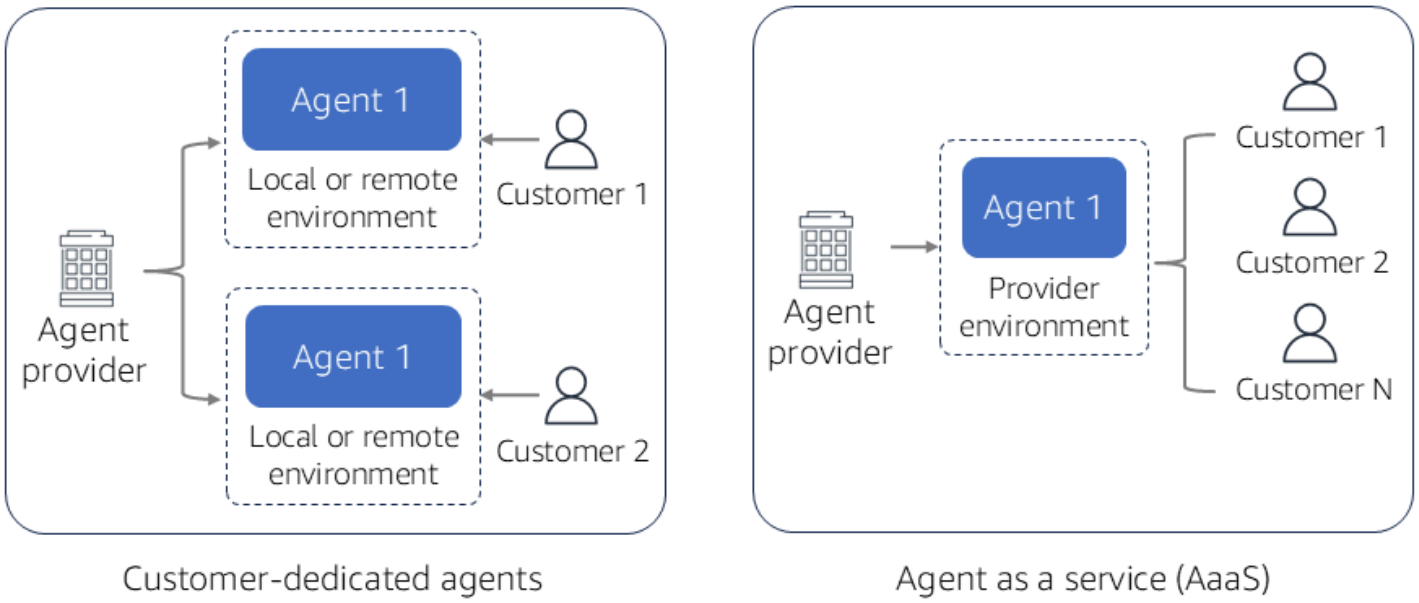
Sie können sich die Vielzahl von Faktoren vorstellen, die die Einführung eines dieser Modelle beeinflussen könnten. Compliance-, regulatorische und Sicherheitsbeschränkungen könnten beispielsweise dazu führen, dass sich Mitarbeiter auf vom Kunden gehostete Agenten konzentrieren. Skalierbarkeit, Agilität und Effizienz könnten Unternehmen dazu bringen, sich dem AaaS-Modell stärker zu nähern.

Das Schlüsselkonzept dabei ist, dass Agenten auf viele Arten eingesetzt und gehostet werden können und werden. Es ist Ihre Aufgabe, zu bestimmen, wie Agenten am besten eingesetzt werden können. Der Platzbedarf, die Sicherheit und die Bereitstellung wirken sich neben anderen Faktoren erheblich auf die Art und Weise aus, wie Sie Agenten aufbauen und betreiben. Private und öffentliche Agenten können beispielsweise unterschiedliche Designs und Release-Lebenszyklen haben.

Agenten treffen auf Mehrmandantenfähigkeit

Man kann sich Agenten leicht als Bausteine vorstellen, bei denen Agenten als eine Reihe von autonomen Komponenten betrachtet werden, die so zusammengestellt werden, dass sie die Anforderungen einer bestimmten Domäne oder eines bestimmten Geschäftsproblems erfüllen. Interessanter wird es, wenn wir darüber nachdenken, wie diese Agenten von Anbietern verpackt und genutzt werden. In vielerlei Hinsicht wird ein Agent zu einer Kosten- und Einnahmequelle für ein Unternehmen. Anbieter von Maklern müssen die verschiedenen Personas berücksichtigen, die ihre Dienste in Anspruch nehmen, das Nutzungsprofil der Personas und die Monetarisierungsstrategien, die es den Anbietern von Vermittlern ermöglichen, Preis- und Staffelungsmodelle zu entwickeln, die sich an den Verbrauchern orientieren.

Agentenanbieter könnten mehrere Modelle für den Einsatz ihrer Agenten unterstützen, um den Kundenanforderungen gerecht zu werden. Das folgende Diagramm zeigt eine konzeptionelle Ansicht der beiden wichtigsten Bereitstellungsmodelle für Agenten.

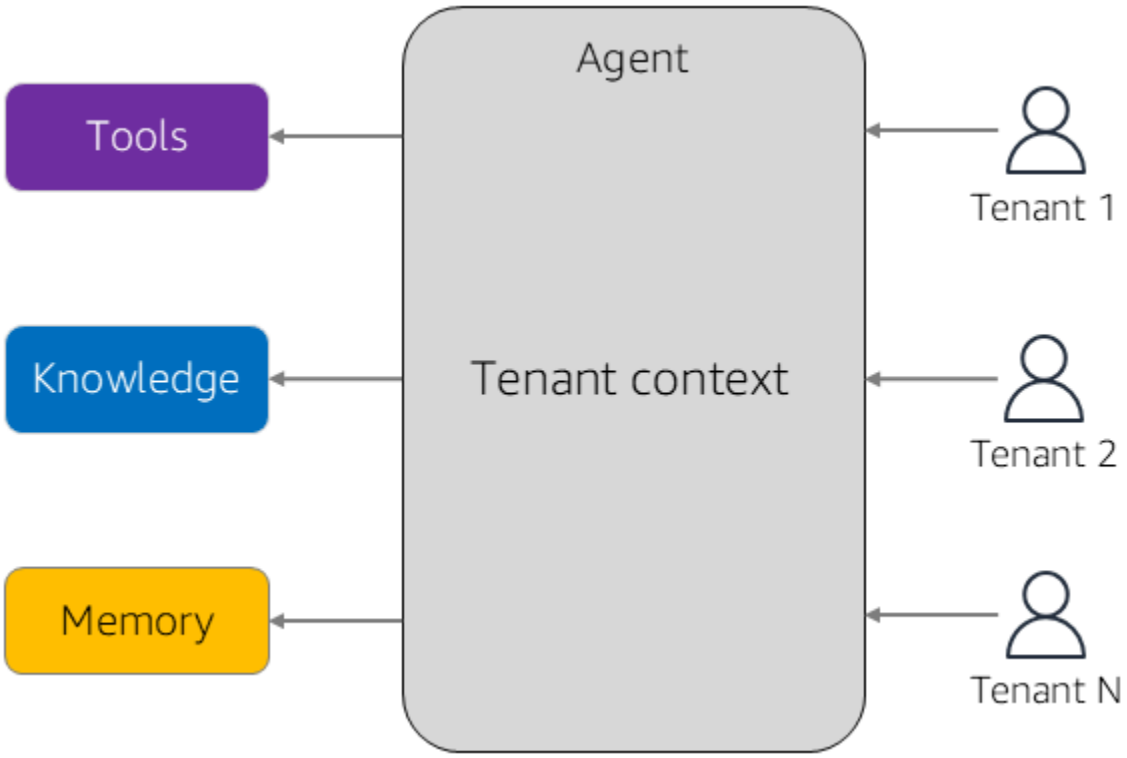


Auf der linken Seite des Diagramms ist das Modell für kundenorientierte Agenten dargestellt. Ein Agentenanbieter erstellt einen Agenten, indem er für jeden registrierten Kunden eine separate Agenteninstanz bereitstellt. Bei diesem Ansatz wären die Fähigkeiten des Agenten und seine Fähigkeit, sich Wissen anzueignen, auf die Umgebung eines bestimmten Kunden beschränkt. Letztlich handelt es sich um ein Kundenerlebnis, das einige der Komplexitäten und Vorteile der Unterstützung spezieller Kundenumgebungen mit sich bringt.

Im Gegensatz dazu zeigt das Diagramm auf der rechten Seite des Diagramms einen einzelnen Agenten, der in der Umgebung des Anbieters eingesetzt wird. Der Mitarbeiter bearbeitet Anfragen von mehreren Kunden, entwickelt sich weiter und lernt auf der Grundlage der kollektiven Erfahrungen aller Kunden. Jeder neue Kunde, der hinzugefügt wird, würde einfach einen weiteren gültigen Kunden des Agenten repräsentieren. Der Agent läuft wie ein Agent-as-a-Service-Modell (AaaS) und verwendet gemeinsame Konstrukte, um die Bedürfnisse eines Kunden zu erfüllen. In beiden Fällen können Agent-Nutzer Anwendungen, Systeme oder sogar andere Agenten sein.

Es gibt zwei Möglichkeiten, das AaaS-Modell zu betrachten. Das obige Modell bietet allen Kunden das gleiche Erlebnis. Das bedeutet, dass die internen Fähigkeiten des Agenten keine Spezialisierung beinhalten, die den Kontext des anfragenden Kunden berücksichtigt. Im Allgemeinen wird bei diesem Modus davon ausgegangen, dass der Aufgabenbereich, die Ziele und der Wert eines Agenten auf einem gemeinsamen Satz von Ressourcen, Kenntnissen und Ergebnissen beruhen, die für alle Kunden gelten.

Beim alternativen Ansatz zu AaaS beeinflusst der Kundenkontext die Erfahrung und Implementierung des Agenten. Das folgende Diagramm bietet einen konzeptionellen Überblick über den Footprint eines AaaS-Agenten in diesem Zusammenhang.



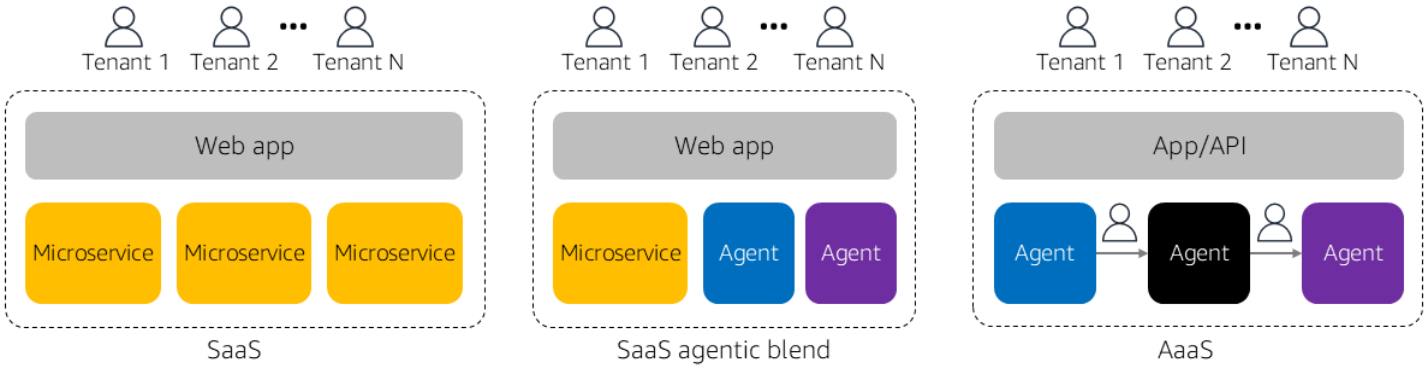
In dieser AaaS-Ansicht wirken sich Herkunft und Kontext der eingehenden Anfragen erheblich auf die Präsenz des Agenten aus. Die Ressourcen, Aktionen und Tools, die Teil der zugrunde liegenden

Implementierung des Agenten sind, können für jede eingehende Mandantenanfrage unterschiedlich sein. Der Wert eines Agenten hängt von seiner Fähigkeit ab, den Mandantenkontext zu nutzen, um Aktionen und Ergebnisse zu erzielen, die vom Status des Mandanten, seinem Wissen und anderen Faktoren beeinflusst werden. Einige Anfragen können zu einem eindeutigen Ergebnis für den Mandanten führen, während andere zu maßgeschneiderten Ergebnissen für jeden Mandanten führen können. Dies verleiht der Lernfähigkeit des Agenten eine neue Dimension. Dazu könnte auch gehören, stärker kontextbezogen zu sein und Wissen zu erwerben und anzuwenden, das die angestrebten Ergebnisse verbessert.

Für Anbieter bietet das AaaS-Modell viele Vorteile. Da mehrere Kunden einen einzigen Agenten nutzen, hat der Anbieter bessere Möglichkeiten, Skaleneffekte zu erzielen, die betriebliche Effizienz zu steigern, die Kosten zu kontrollieren und ein einheitliches Managementerlebnis zu schaffen. Dies hat das Potenzial für mehr Agilität, Innovation und Wachstum für das Agentengeschäft.

Diese Eigenschaften überschneiden sich mit den gleichen Prinzipien, die die Einführung des SaaS-Modells (Software as a Service) vorantreiben. Im Wesentlichen ist das AaaS-Modell als Multi-Tenant-Service konzipiert, der viele der gleichen Skalierungs-, Resilienz-, Isolations-, Onboarding- und Betriebsattribute wie in einer SaaS-Umgebung bietet. In vielerlei Hinsicht lehnt sich die AaaS-Erfahrung stark an die Strategien und Praktiken der SaaS-Anbieter an, aber es ist vernünftig, diese Begriffe zu trennen. Für unsere Zwecke liegt der Schwerpunkt hauptsächlich auf den Auswirkungen, die mit Bau- und Betriebsagenturen einhergehen, die Unterstützung mehrerer Mandanten benötigen.

Bei einem System, das alle Benutzer gleich behandeln kann und nicht die Verwaltung persistenter, sensibler oder kundenspezifischer Daten erfordert, würde das Konzept des Mietverhältnisses nur minimale Auswirkungen auf die Agenten haben. Bei Systemen, von denen erwartet wird, dass sie mehrere Kunden bedienen und gleichzeitig Datenisolierung, Anpassung und Kontextsensitivität wahren, könnte die Unterstützung mehrerer Mandanten ein wesentlicher Bestandteil des Designs, der Strategie und des Ziels eines Agenten sein. Das folgende Diagramm zeigt, wie Mehrmandantenfähigkeit in behördlichen Umgebungen eingesetzt werden kann.



Auf der linken Seite dieses Diagramms befindet sich eine klassische Multi-Tenant-Architektur. Sie umfasst eine Webanwendung und eine Reihe von Microservices, die Geschäftslogik implementieren. Mehrere Mandanten nutzen die gemeinsam genutzte Infrastruktur dieser Umgebung und skalieren, um den wechselnden Arbeitsbelastungen einer sich ständig weiterentwickelnden Mandantenpopulation gerecht zu werden. Die Umgebung wird für alle Mieter über eine einzige Glasscheibe betrieben und verwaltet.

Stellen Sie sich vor, wie dieses mentale Modell dem Agenten auf der rechten Seite dieses Diagramms zugeordnet wird. Ein Agent führt ein AaaS-Modell aus, das von einem oder mehreren Mandanten genutzt wird. Die Agenten könnten von mehreren Anbietern stammen, wobei der Mandantenkontext zwischen ihnen fließt, da eine einzelne Instanz eines Agenten Anfragen von mehreren Mandanten verarbeiten muss.

Das Beispiel in der Mitte dieses Diagramms ist ein Hybridmodell, bei dem Agenten Teil des gesamten SaaS-Erlebnisses sind. Einige Teile des Systems werden in einem traditionelleren Modell implementiert, während andere Teile des Systems auf Agenten angewiesen sind. Dieses Muster ist wahrscheinlich bei vielen SaaS-Angeboten üblich, insbesondere bei Unternehmen, die auf eine Agentenerfahrung umsteigen. Es ist üblich, dass dieses Modell fortbesteht, da nicht alle Systeme als reines AaaS bereitgestellt werden. Beachten Sie auch, dass die Mehrmandantenfähigkeit für die Agenten des Modells weiterhin gilt. Die Agenten können zwar in ein System eingebettet sein, sie können aber dennoch Anfragen von mehreren Mandanten bearbeiten.

Es liegt auf der Hand, sich zu fragen, ob Mehrmandantenfähigkeit wirklich wichtig ist. Sie könnten argumentieren, dass ein Agent Anfragen bearbeitet, sodass die Unterstützung des Mietverhältnisses möglicherweise nur geringe Auswirkungen hat. Wenn wir uns jedoch eingehender mit den Auswirkungen von Agenten mit mehreren Mandanten befassen, kann sich das Mietverhältnis direkt darauf auswirken, wie Agenten beeinflussen, wie auf Tools, Speicher, Daten und andere Agententeile zugegriffen, bereitgestellt und konfiguriert werden, um einzelne Mandanten zu unterstützen. Das Mietverhältnis beeinflusst auch, wie Skalierung, Drosselung, Preisgestaltung, Staffelung und andere geschäftliche Aspekte sich auf die Architektur Ihres Agenten auswirken.

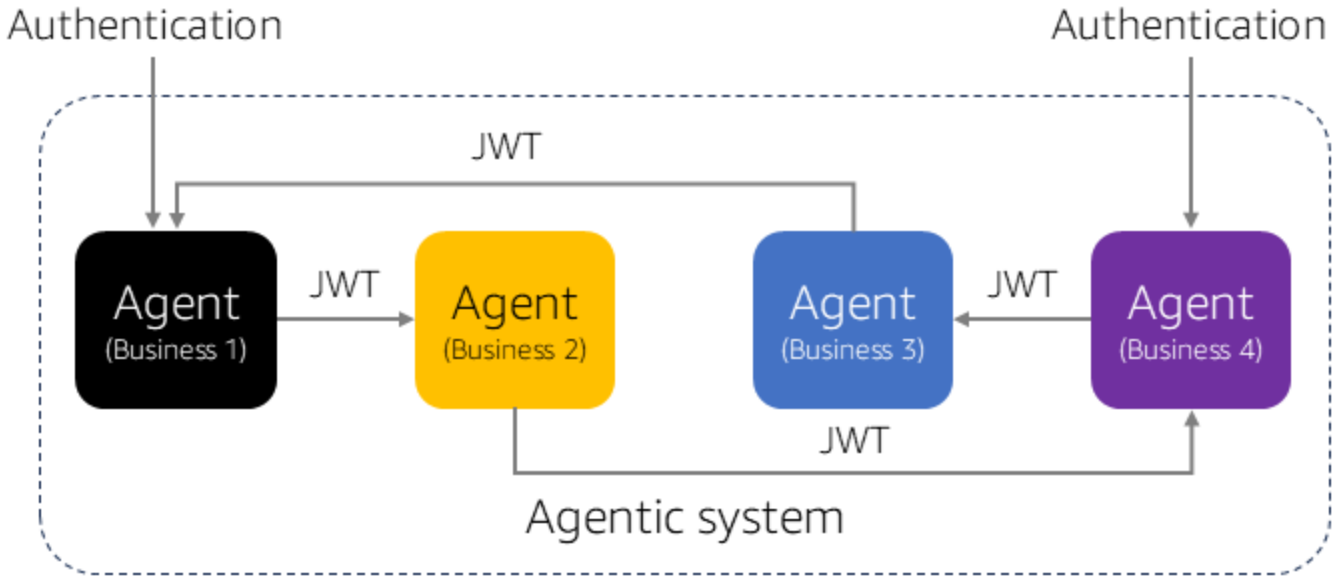
Eine Erkenntnis daraus ist, dass es Anwendungsfälle für Agenturen gibt, die Unterstützung mehrerer Mandanten erfordern. Die Herausforderung besteht darin, herauszufinden, wie Mehrmandantenfähigkeit das Gesamtdesign und die Architektur Ihrer Agentenerfahrung beeinflusst. Für einige Agenten stellt der Mehrmandanten-Support ein Alleinstellungsmerkmal dar, das es den Agenten ermöglicht, ihren Agenten einen mandantenspezifischen Kontext zuzuweisen, um zielgerichtete Ergebnisse zu erzielen.

In den folgenden Abschnitten erfahren Sie, wie nützlich die Terminologie und die Entwurfsmuster sind, die wir zur Beschreibung von mehrinstanzenfähigen SaaS-Architekturen erstellen. Diese Konzepte können in das AaaS-Modell übernommen werden, indem nützliche Aspekte übernommen werden, wodurch neue agentenspezifische Konzepte eingeführt werden, wo sie benötigt werden.

Identität, Mandantenkontext und Agentensysteme

Das Hinzufügen von Mandantenkontext zu einzelnen Agenten ist keine besondere Herausforderung. In vielen Fällen können sich Teams auf typische Mechanismen verlassen, die Benutzer und Systeme an Mandanten binden und Token, die den Mandanten kennen, an Agenten weitergeben. Dies ist relevant, wenn wir bedenken, wie der Kontext und die Identität von Mandanten mehrere Agenten unterstützen. In diesem Modell müssen Mandanten an eine Identität gebunden sein, die sich über alle zusammenarbeitenden Agenten erstreckt.

Im Allgemeinen erfordert die Agentendomäne ein stärker bereichsübergreifendes Identitätsmodell, das den aktuellen und zukünftigen Bedürfnissen der Agentensysteme entspricht. Agentenanbieter benötigen Identitätsmechanismen, die einzigartige Sicherheits-, Compliance- und Autorisierungsmodelle unterstützen, die mit den operativen Agentensystemen einhergehen. Dies ist besonders schwierig in Umgebungen, in denen Systeme von Kunden oder anderen Agenten zusammengestellt werden. Jeder Onboarding-Agent muss seine Identität und seinen Mandantenkontext mit den Interaktionen der Agenten verknüpfen. Das folgende Diagramm verdeutlicht die potenziellen Herausforderungen im Zusammenhang mit Identität und Mandantenkontext, die Teil von Interaktionen agent-to-agent (a2a) sind.



Dieses Diagramm zeigt eine Reihe von von Anbietern entwickelten Agenten, die als Teil des von uns behandelten Agentensystems interagieren. Es wurde jetzt mit Identitäts- und Mandantenkontext nachgerüstet. Dieses Szenario ist ein Beispiel für ein Agentensystem, das mehrere Einstiegspunkte unterstützt. Wir gehen davon aus, dass jeder Agent in diesem System seinen eigenen Authentifizierungsmechanismus benötigt, um das System oder den Benutzer einem bestimmten Mandanten zuzuordnen. Während diese Agenten interagieren, wird der Mandantenkontext an ein JSON-Web-Token (JWT) übergeben, das verwendet wird, um den Zugriff zu autorisieren und den Mandantenkontext in den Agenten einzufügen.

Konzeptionell besteht der Hauptunterschied zu diesem Szenario darin, dass Agenten unabhängig voneinander bereitgestellt und betrieben werden, was bedeutet, dass jeder Agent in der Lage sein muss, seine Identität zu ermitteln und den Zugriff zu autorisieren. Der Schlüssel liegt darin, dass seine Identität über eine gewisse verteilte Fähigkeit verfügen muss, um den Anforderungen des umfassenderen Agentensystems gerecht zu werden. Außerdem muss die Art und Weise, wie Agenten den Mandantenkontext teilen, aufeinander abgestimmt werden.

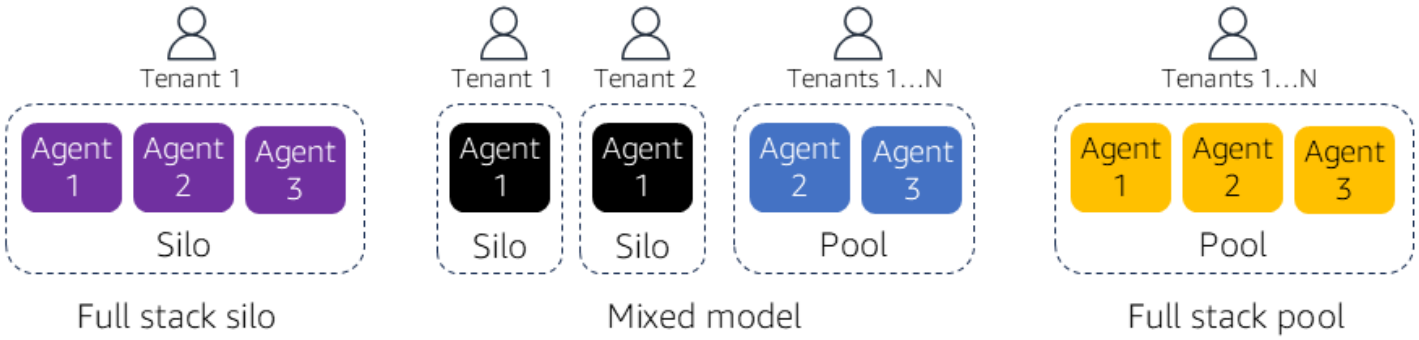
Anwendung des SaaS-Geschäftswerts auf AaaS

Im Allgemeinen berücksichtigen wir bei der Ausführung eines Systems in einem as-a-service Modell die Art der Erfahrung und die Art und Weise, wie sich der technische und betriebliche Fußabdruck auf die Geschäftsergebnisse auswirkt. Bei der Einführung von SaaS nutzen Unternehmen beispielsweise Skaleneffekte, betriebliche Effizienz, Kostenprofile und Flexibilität, um Wachstum, Margen und Innovation voranzutreiben.

Agenten, die als AaaS bereitgestellt werden, werden wahrscheinlich ähnliche Geschäftsergebnisse erzielen. Durch die Unterstützung mehrerer Mandanten kann ein Agent den Ressourcenverbrauch an den Aktivitäten der Mandanten ausrichten. Dies führt zu Skaleneffekten, die mit herkömmlichen SaaS-Umgebungen einhergehen. AaaS ermöglicht es Unternehmen auch, Agenten so zu verwalten, zu betreiben und einzusetzen, dass häufige Releases möglich sind und die Agentenanbieter flexibler arbeiten können. Der Schlüssel ist, dass das AaaS-Modell nicht von Technologie abhängt. Es entwickelt und fördert Geschäftsstrategien, die das Wachstum fördern, die Akzeptanz optimieren und den Betrieb vereinfachen.

Bereitstellungsmodelle für Agenten

Bei einer grundlegenden AaaS-Erfahrung kann ein Anbieter Agenten nach verschiedenen Mustern einsetzen. Es gibt eine Vielzahl von Faktoren, die beeinflussen, wie Agenten eingesetzt werden, um Kunden-, Leistungs-, Compliance-, Geografie- und Sicherheitsanforderungen zu erfüllen. Verschiedene Bereitstellungsstrategien wirken sich darauf aus, wie ein Agent konzipiert, implementiert und genutzt wird. An dieser Stelle können wir klassische Begriffe für mehrere Mandanten einführen, um verschiedene Bereitstellungsstrategien zu kennzeichnen. Das folgende Diagramm zeigt verschiedene Varianten für den Einsatz von Agenten in einer AaaS-Umgebung.



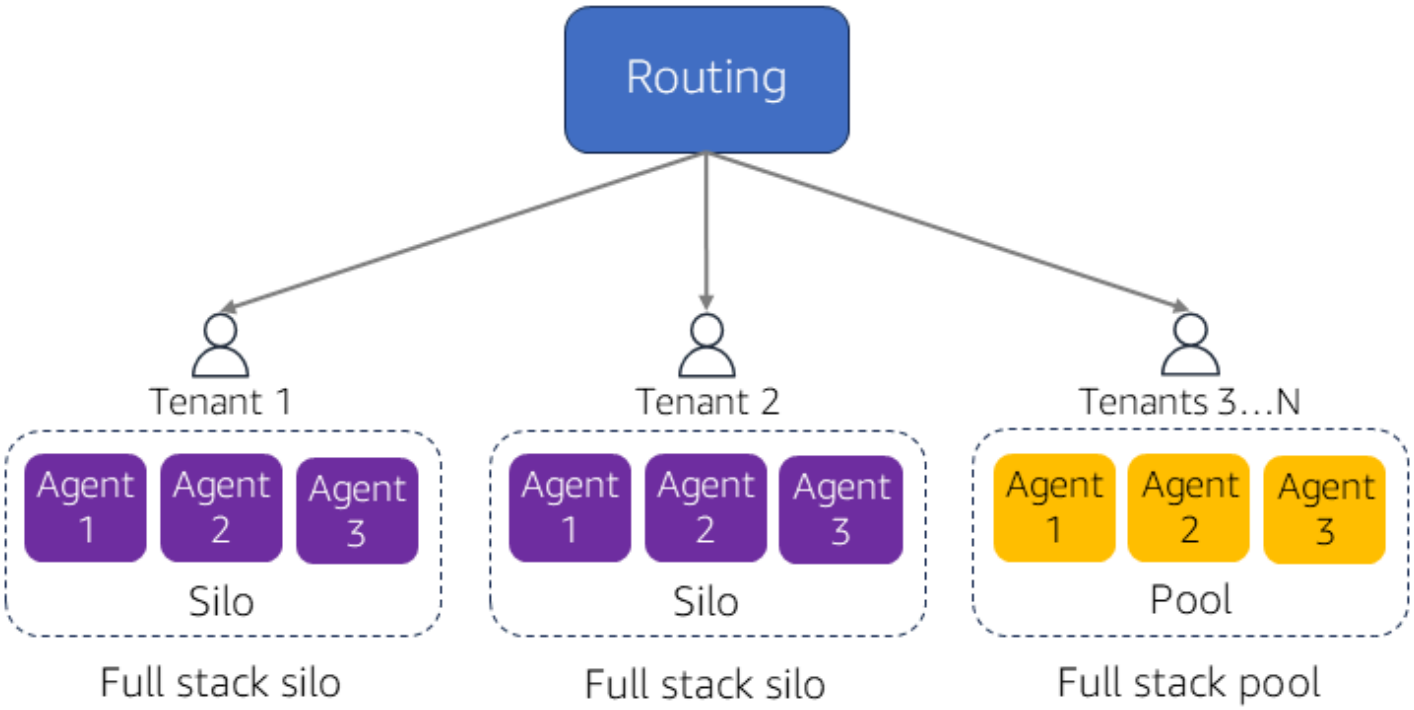
Dieses Diagramm stellt drei Modi der Agentenbereitstellung dar. Auf der linken Seite befindet sich ein isoliertes Modell, bei dem jedem Mandanten ein vollständig isoliertes Erlebnis und eine spezielle Gruppe von Agenten zur Verfügung steht. In diesem Szenario teilen sich Agenten Rechenleistung, Ressourcen oder Ausführungsumgebungen nicht mandantenübergreifend.

Das mittlere Beispiel veranschaulicht ein Hybridmodell, bei dem Mandanten eine Kombination aus isolierten und gepoolten Agenten verwenden. Agent 1 wird beispielsweise im isolierten Modus eingesetzt — jeder Mandant erhält eine eigene Instanz —, während die Agenten 2 und 3 in einem Poolmodell arbeiten und Ressourcen von allen Mandanten gemeinsam nutzen.

Auf der rechten Seite befindet sich ein vollständig gepooltes Modell, bei dem alle Agenten von mehreren Mandanten gemeinsam genutzt werden, was eine klassische Mehrmandantenbereitstellung ermöglicht. In diesem Szenario nutzen Mandanten eine gemeinsame Rechen-, Speicher- und Serviceinfrastruktur für die Ausführung der Agenten.

Die Idee ist, dass Agenten in verschiedenen Bereitstellungsmodellen arbeiten können, wobei Rechenleistung und abhängige Ressourcen entweder dediziert (isoliert) oder von allen Mandanten gemeinsam genutzt (gepoolt) werden. Diese Bereitstellungsstrategien schließen sich nicht gegenseitig aus. Agentendienste unterstützen häufig ein breites Spektrum von Kundenanforderungen und kombinieren beide Modelle, um ein ausgewogenes Verhältnis zwischen Leistung, Isolierung,

Kosten und Skalierbarkeit zu erreichen. Das folgende Diagramm zeigt ein Agentensystem, das mehrere Bereitstellungskonfigurationen innerhalb derselben Betriebsumgebung unterstützt.



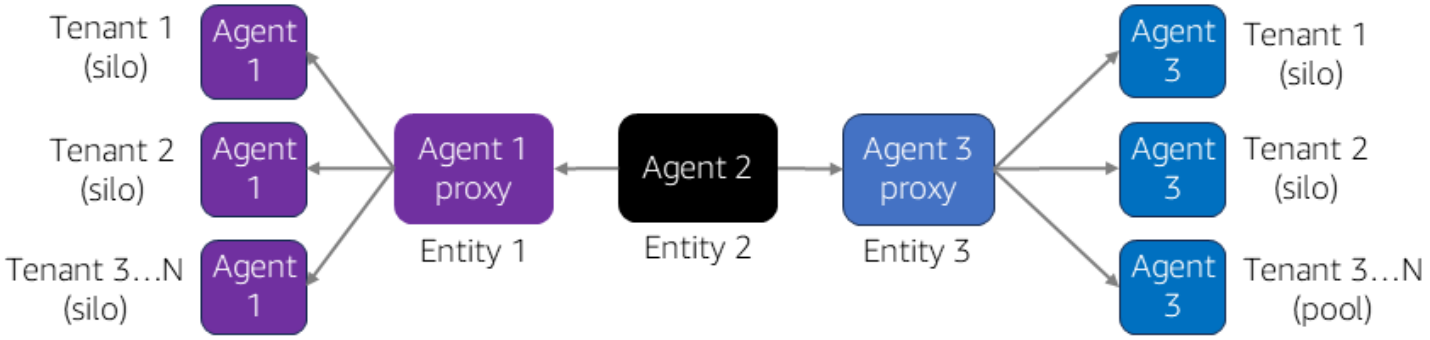
In diesem Diagramm verfügt ein Agentenanbieter über drei Agenten, die über Agent as a Service (AaaS) bereitgestellt werden. Sie unterstützen zwei Arten von Mandanten. Auf der linken Seite haben zwei Mieter Konformitäts- und Leistungsanforderungen, die sie durch ein Full-Stack-Silomodell erfüllen. Der verbleibende Mandant auf der rechten Seite wird in einem Poolmodell ausgeführt, bei dem sich die Mandanten Ressourcen teilen.

Wenn Agilität und betriebliche Effizienz das Ziel sind, versuchen Sie, die Auswirkungen zu begrenzen, die mit der Unterstützung von Bereitstellungsmodellen pro Mandant verbunden sind. Das bedeutet, Routing- und andere Erlebnismechanismen einzurichten, mit denen Agenten über eine zentrale Oberfläche verwaltet, bedient und bereitgestellt werden können.

Wenn Sie einen Agenten in einer Umgebung mit wenig oder gar keinem Code entwickeln, wird es keine isolierten oder gepoolten Agenten geben. Stattdessen können Agenten vollständig von einem anderen Agenten verwaltet werden. Isolierte und gepoolte Modelle eignen sich eher für Umgebungen, in denen eine Organisation den Aufbau und die Präsenz des Agenten kontrolliert. In diesem Fall sollten sich die Teams überlegen, welches Bereitstellungsmodell unterstützt werden soll.

Oberflächlich betrachtet haben diese Bereitstellungsmodelle keinen direkten Einfluss darauf, wie ein Agent in einem umfassenderen System funktioniert. Ein Agent hat möglicherweise keine direkte Kenntnis von anderen Agenten, die in einem Silo- oder Poolmodell eingesetzt werden. Stattdessen

können diese Bereitstellungsstrategien als Teil eines Routing-Konstrukts innerhalb einer Umgebung implementiert werden. Das folgende Diagramm zeigt ein Beispiel dafür, wie isolierte und gepoolte Modelle mithilfe einer Routing-Strategie implementiert werden können.



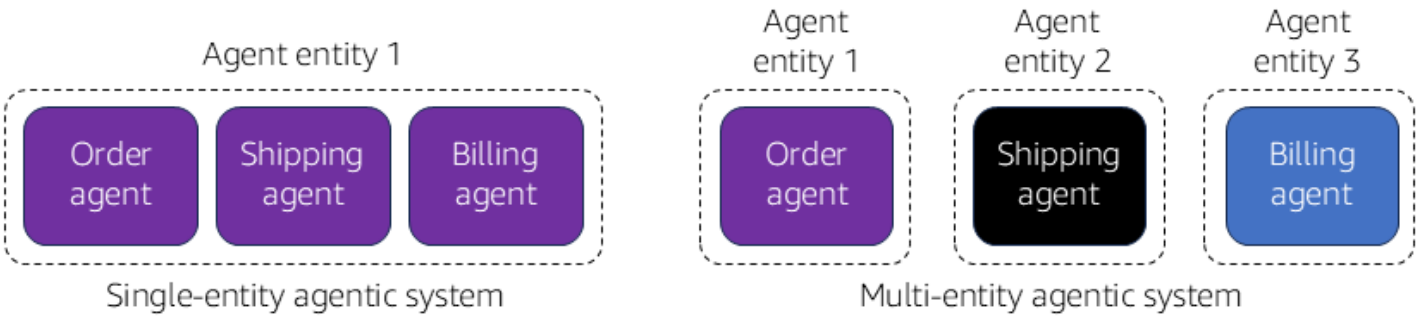
Dieses Beispiel umfasst drei Agenten von drei verschiedenen Anbietern. Jeder Agentenanbieter hat die Möglichkeit, seine eigene Bereitstellungsstrategie zu implementieren. Agent 1 verwendet beispielsweise einen Proxy, um eingehende Anfragen an eine Gruppe isolierter Mandantenagenten zu verteilen. Agent 2 benötigt kein Routing und unterstützt alle Mandantenanfragen über einen gemeinsamen Agenten. Bei Agent 3 handelt es sich um ein Hybridmodell, bei dem einige Mandanten isoliert und andere in einem Pool zusammengefasst sind.

Ob und wie Sie diese Bereitstellungsmodelle unterstützen, hängt von der Art Ihrer Lösung ab. Möglicherweise müssen Sie keines der beiden Modelle unterstützen. Möglicherweise gibt es jedoch Fälle, in denen Sie die Unterstützung dieser Strategie in Betracht ziehen müssen, z. B. in Bezug auf die Einhaltung von Vorschriften, Noisy Neighbor, Performance oder Tiering.

Einführung und Anwendung des Mandantenkontextes

Wenn wir Agenten entwickeln, die Mehrmandantenfähigkeit unterstützen, müssen wir zunächst überlegen, wie der Mandantenkontext eingerichtet werden kann, der dann verwendet wird, um mandantenspezifische Richtlinien, Strategien und Mechanismen innerhalb der Implementierung des Agenten anzuwenden.

Auf der grundlegendsten Ebene können Sie mithilfe der gängigen Tools und Mechanismen, die wir in klassischen Mehrmandantenarchitekturen verwenden, den Mandantenkontext in Agenten einführen. Dies kann durch einen API-Schlüssel oder verschiedene andere Validierungsmechanismen geschehen. OAuth Viele Beispiele hierfür konzentrieren sich auf die Auflösung eines authentifizierten Systems oder Benutzers in einen JSON-Web-Token-Schlüssel (JWT), der den Mandantenkontext enthält. Das JWT wird dann durch das System verbreitet. Das wird noch interessanter, wenn wir darüber nachdenken, wie man Agentensysteme zusammensetzt. Das folgende Diagramm zeigt ein Beispiel für zwei Arten von Agentenumgebungen.



In diesem Diagramm stellt das Modell auf der linken Seite ein Agentensystem dar, in dem alle Agenten einer einzigen Entität gehören, von ihr verwaltet und gehostet werden. Wenn Sie die volle Kontrolle über das gesamte Erlebnis haben, können Sie typische Strategien anwenden, um Mandanten an die einzelnen Agenten weiterzuleiten.

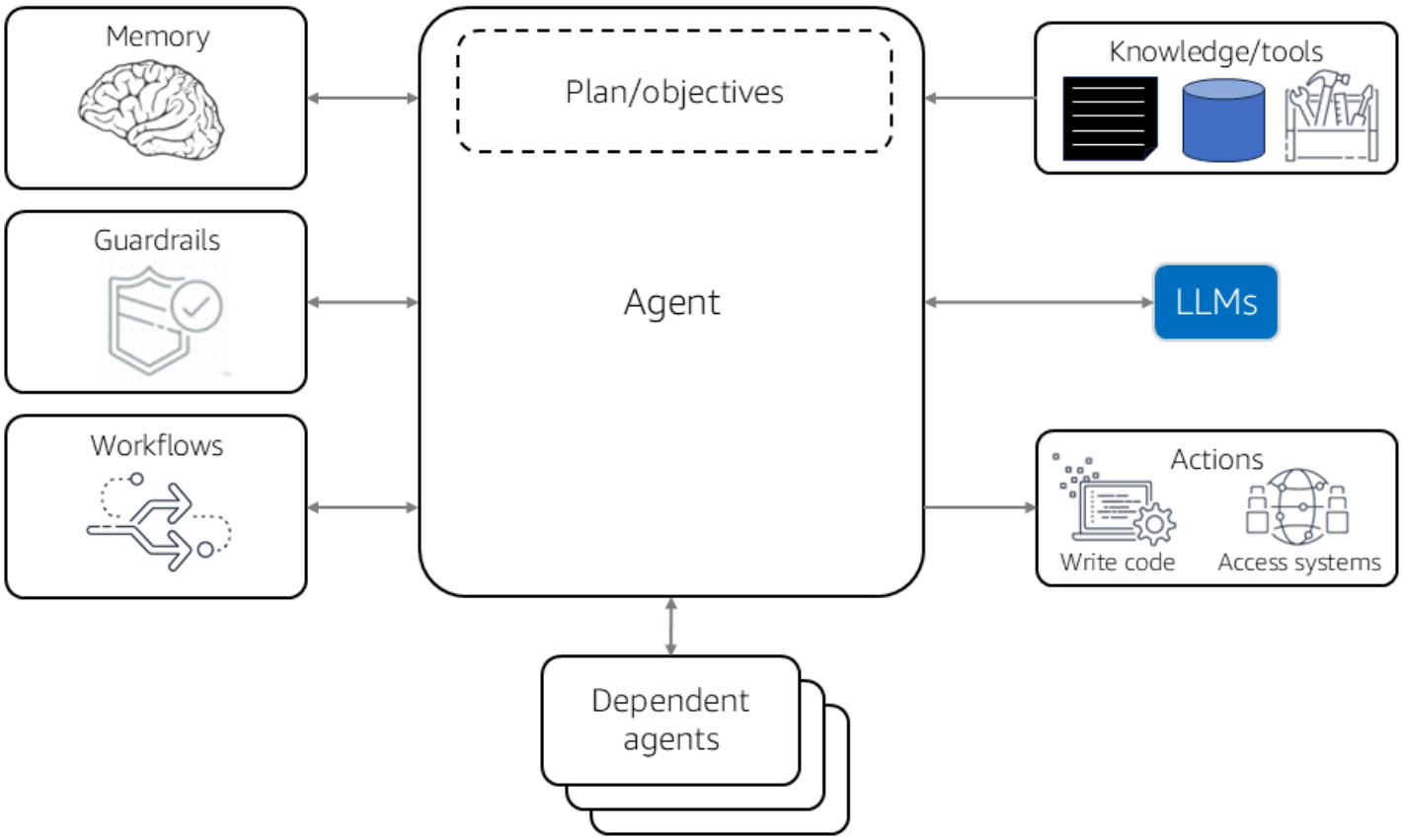
Das Modell auf der rechten Seite, das möglicherweise häufiger verwendet wird, stellt ein System von Agenten dar, die sich über mehrere Entitäten erstrecken. Die Agenten werden unabhängig voneinander erstellt, verwaltet und betrieben, sodass sie jeweils über ihre eigenen Authentifizierungs- und Autorisierungsschemata verfügen. Die Herausforderung dabei besteht darin, dass wir eine universelle Methode benötigen, um den Mandantenkontext zwischen diesen Agenten zu lösen und gemeinsam zu nutzen. Dies basiert auf einem stärker verteilten Modell, bei dem jeder Agent in der Lage sein muss, Systeme oder Benutzer zu authentifizieren und sie gemäß den angewandten Mechanismen einem Mandanten zuzuordnen.

Aufbau von Agenten, die mandantenorientiert sind

Die Mehrmandantenfähigkeit beeinflusst die Art und Weise, wie wir einzelne Agenten einsetzen. Denken Sie bei der Bearbeitung von Anfragen durch einen Agenten darüber nach, wie sich der Mandantenkontext darauf auswirkt, wie ein Agent auf Daten zugreift, Entscheidungen trifft und Aktionen auslöst. Um besser zu verstehen, wie und wo sich die Mehrmandantenfähigkeit auf das Profil Ihres Agenten auswirkt, sollten Sie zunächst herausfinden, wie Konstrukte Teil eines Agenten sein können.

Die Herausforderung besteht darin, dass Umfang, Art und Gestaltung der Agenten alles andere als konkret sind, da die Anbieter ihre eigenen Entscheidungen über die Gestaltung eines Agentenerlebnisses treffen. Letztlich besteht der Sinn eines Agenten darin, dass es sich um einen autonomen Lerndienst handelt, der auf eine Reihe von Tools, Datenquellen und Speicher zugreifen kann, um zu bestimmen, wie eine Aufgabe am besten gelöst werden kann.

Es ist weniger wichtig, genau zu wissen, welche Strategien und Muster ein Agent verwendet. In einem Mehrmandantenmodell ist es wichtiger, zu ermitteln, wie die verschiedenen Teile eines Agenten konfiguriert, wie darauf zugegriffen wird und wie sie angewendet werden. Stellen Sie sich eine potenzielle Agentenumgebung vor, die auf eine Reihe von Ressourcen und Mechanismen angewiesen ist, um ihre Ziele zu erreichen. Das folgende Diagramm zeigt ein Beispiel für einen solchen Agenten.



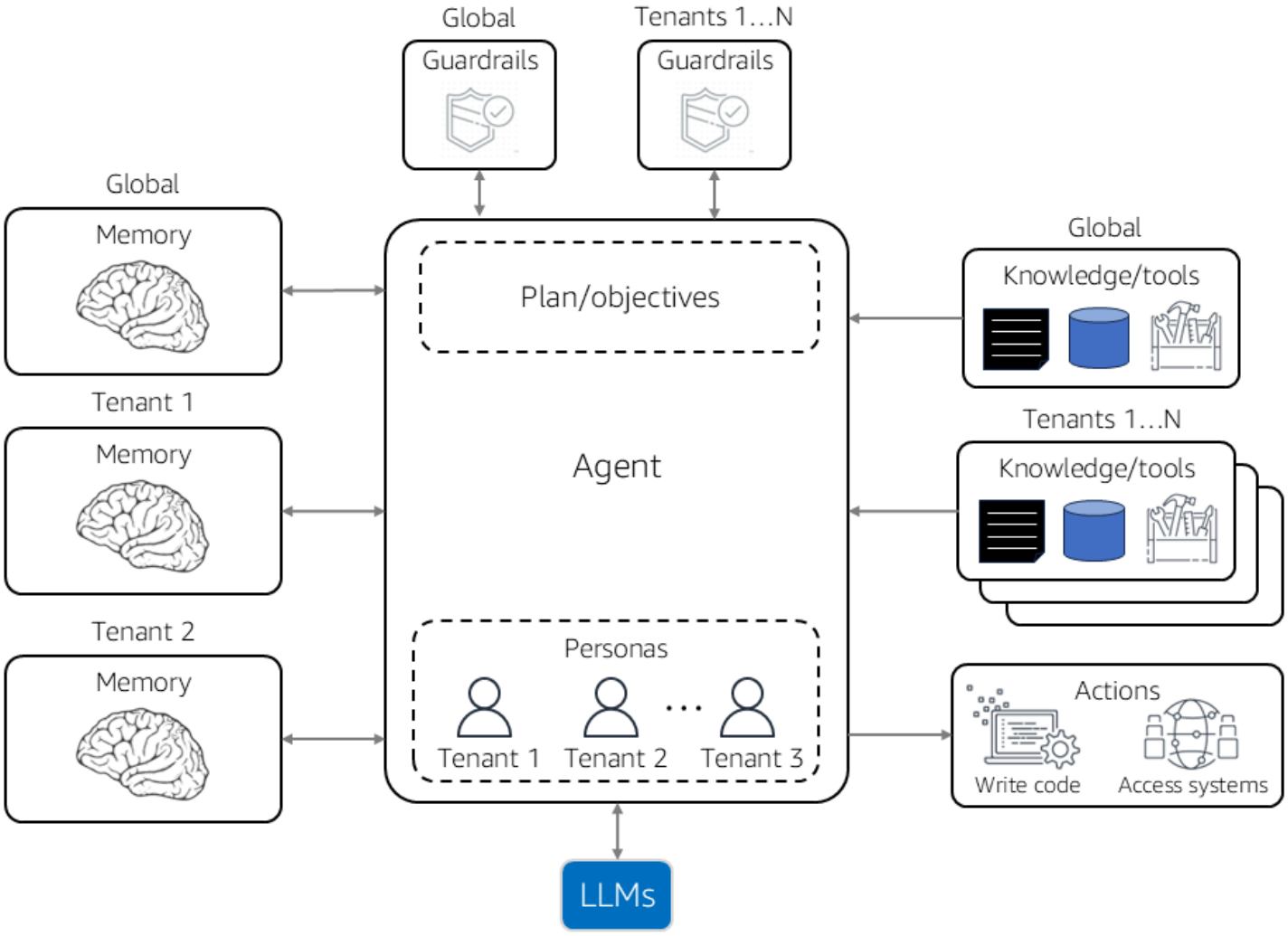
Dieses Diagramm stellt eine umfassende Palette agentischer Möglichkeiten dar und zeigt verschiedene Tools und Mechanismen, die kombiniert werden könnten, um ein Ziel zu erreichen. Auf der linken Seite des Diagramms sehen Sie, wie ein Agent im Rahmen seines Kontextes auf Speicher angewiesen ist, welche Richtlinien für die Festlegung der Richtlinien für seine Aktivitäten gelten, und Workflows, die auf bestimmte Aufgaben ausgerichtet sind. Manche mögen argumentieren, dass Workflows nicht in diesen Kontext aufgenommen werden sollten, aber es kann Szenarien geben, in denen Workflows integraler Bestandteil einer Agentenerfahrung sind.

Die rechte Seite des Diagramms zeigt, wie Inputs wie Wissen und Tools zusätzliche Einblicke und Kontext liefern können, die die Fähigkeiten des Agenten verbessern. Der Agent gibt dann Aktionen aus, z. B. das Schreiben von Code oder den Zugriff auf Systeme. Der untere Teil des Diagramms zeigt, wie Agenten von einem oder mehreren internen Agenten oder Agenten von Drittanbietern abhängig sind, die als Teil eines umfassenderen Systems orchestriert werden können.

Wir können uns jetzt überlegen, was es bedeutet, Mehrmandantenfähigkeit einzuführen. Das Mietverhältnis zwingt uns, darüber nachzudenken, wie und wo ein Agent Strategien und Mechanismen einführt, die Verhalten und Aktionen diktieren. Dies verleiht der Art und Weise, wie wir

über Agenten in Bezug auf ihr Wissen, ihr Lernen, ihre Werkzeuge und ihr Gedächtnis denken, eine weitere Dimension.

Lassen Sie uns nun überlegen, wie dieses Modell modifiziert werden kann, um Mehrmandantenfähigkeit zu unterstützen. Das folgende Diagramm zeigt ein Beispiel für ein Multi-Agent-Modell.



In diesem Diagramm stellen wir Mandanten-Personas vor, die bestimmen sollen, wie ein Agent den Mandantenkontext integriert. Auf der linken Seite des Diagramms wurde beispielsweise der Agentenspeicher so geändert, dass er mandantenspezifischen Speicher unterstützt. Das Gleiche gilt für die rechte Seite des Diagramms, wo der Agent mandantenspezifisches Wissen und Tools unterstützt. Dieselbe Unterstützung gilt auch für Leitplanken.

Dies kann ein extremes Beispiel sein, da nicht für alle Aspekte eines Multi-Tenant-Agents Ressourcen pro Mandant erforderlich sind. Der Punkt ist, dass Sie sich überlegen sollten, wie Sie die Effektivität Ihres Agenten erhöhen können, wenn Sie ihn auf bestimmte Mandanten zuschneiden.

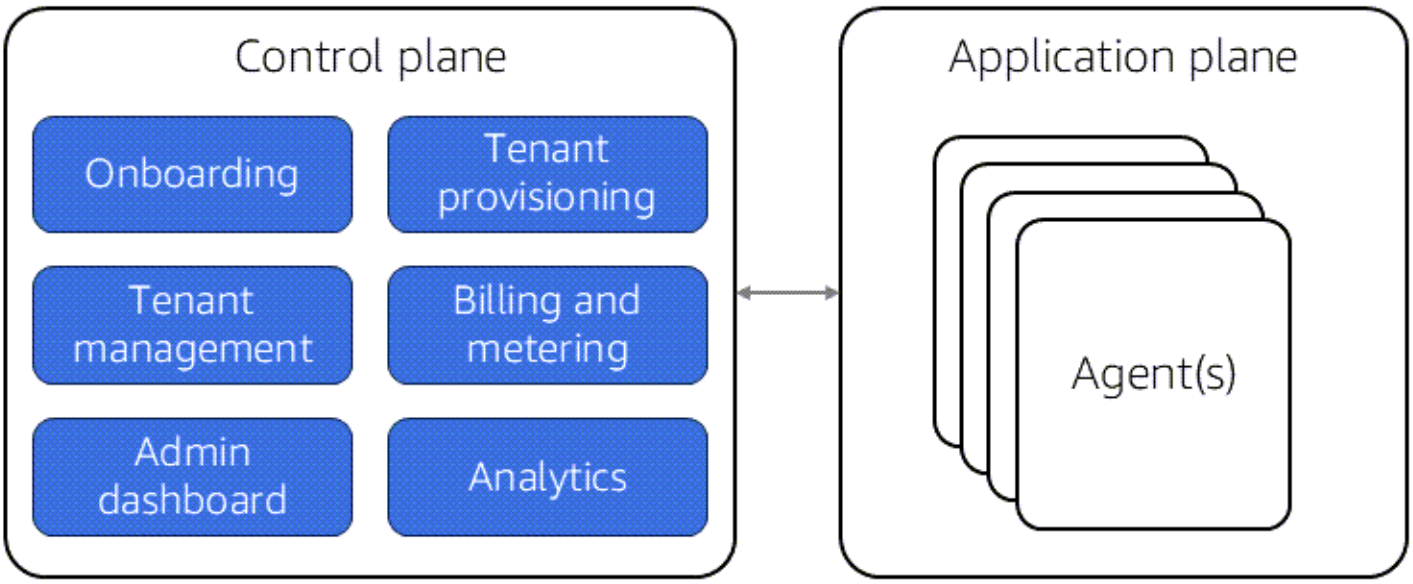
Dieser Ansatz ermöglicht es Ihrem Makler, seine Wirkung und seinen Wert zu steigern, in seinen Antworten einen relevanteren Kontext zu bieten und spezielle Fähigkeiten zu entwickeln. Der Agent wird dann in der Lage sein, Aufgaben zu erlernen, sich anzupassen und Aufgaben auszuführen, die für verschiedene Personas einzigartig geeignet sind.

Die Grundidee ist, dass sich der Mandantenkontext direkt darauf auswirkt, wie Sie Agenten erstellen. Es kann auch die Interaktionen von Mandanten mit externen Entitäten, einschließlich anderen Agenten, beeinflussen. Der Aufbau eines Multi-Tenant-Agents bringt traditionelle Herausforderungen wie laute Nachbarn, Isolierung von Mietern, Staffelung, Drosselung und Kostenmanagement mit sich. Das Design und die Architektur Ihres Agenten müssen diese grundlegenden Konzepte für mehrere Mandanten berücksichtigen, auf die wir im nächsten Abschnitt näher eingehen werden.

Einsatz von Steuerungsebenen in behördlichen Umgebungen

Best Practices für mehrere Mandanten unterteilen Implementierungen häufig in zwei verschiedene Teile: eine Steuerungsebene und eine Anwendungsebene. Die Kontrollebene bietet eine zentrale Oberfläche für den Zugriff auf Betriebs-, Management- und Orchestrierungsmechanismen, die alle Mandanten der Umgebung umfassen. Auf der Anwendungsebene befinden sich die Geschäftslogik, die Funktionen und die Funktionsfähigkeit.

Diese Aufgabenteilung gilt auch für Agentenmodelle. Ein Multi-Tenant-Agent erfordert ein gewisses Maß an zentralisierter Verwaltung, Bedienung und Erkenntnissen, und es ist sinnvoll, diese Anforderungen kontinuierlich über eine Kontrollebene zu erfüllen. Das folgende Diagramm zeigt in einer konzeptionellen Ansicht, wie diese Ebenen innerhalb einer Agent-as-a-Service (AaaS) - Umgebung aufgeteilt sind.

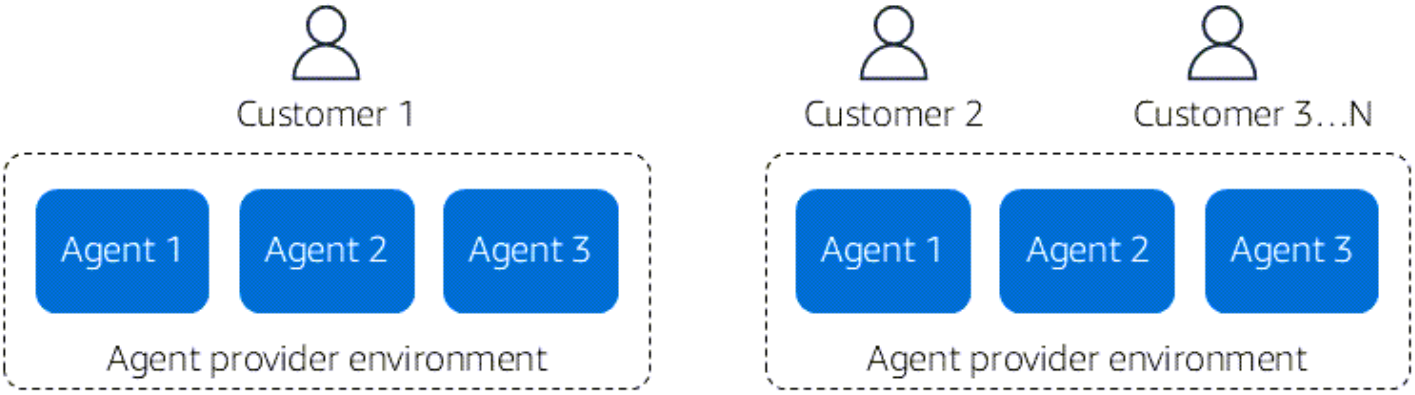


Dieses Diagramm zeigt die herkömmliche Trennung von Kontroll- und Anwendungsebene. Neu ist, dass die Steuerungsebene jetzt die Agenten verwaltet, aus denen sich eine AaaS-Umgebung zusammensetzt. Die Kontrollebene interagiert mit allen Agenten, da wir davon ausgehen, dass die Agenten von einem Anbieter erstellt, verwaltet und bereitgestellt werden.

Dieses Modell führt zu zusätzlichen Komplexitätsebenen, insbesondere in Bezug auf den Lebenszyklus der Agenten und die Koordination durch Dritte, behält jedoch die grundlegende Trennung der einzelnen Aspekte bei. Die Steuerungsebene bietet weiterhin dieselben

Kernfunktionen, indem sie die Konfiguration der Agenten orchestriert, die Überwachung von Mandanten und Agenten gewährleistet, Verbrauchs- und Messdaten für die Abrechnung sammelt und Mandantenrichtlinien verwaltet.

Dieses Szenario wird komplexer, wenn Sie ein System mit mehreren Agenten in Betracht ziehen, das Agenten verschiedener Anbieter umfasst. Das folgende Diagramm zeigt ein Beispiel für ein solches Modell.



Dieses Diagramm zeigt vier Agenten verschiedener Anbieter, die Teil eines multiagentischen Systems sind. Drittanbieter betreiben und implementieren weiterhin jeden Agenten, der so konfiguriert ist, dass er autorisierten Zugriff von einem oder mehreren Anbietern ermöglicht. Die Agenten unterliegen jedoch weiterhin der Kontrolle des Anbieters, sodass jeder Agent seine eigene Kontrollebene beibehält.

Im Wesentlichen verhalten sich diese Multi-Tenant-Agenten wie Dienste von Drittanbietern, die sich in andere Agenten integrieren lassen. Daher müssen sie über eine eigene Steuerungsebene verfügen, um den zentralen Betrieb, die Konfiguration und die Verwaltung der Funktionen eines Agenten zu gewährleisten.

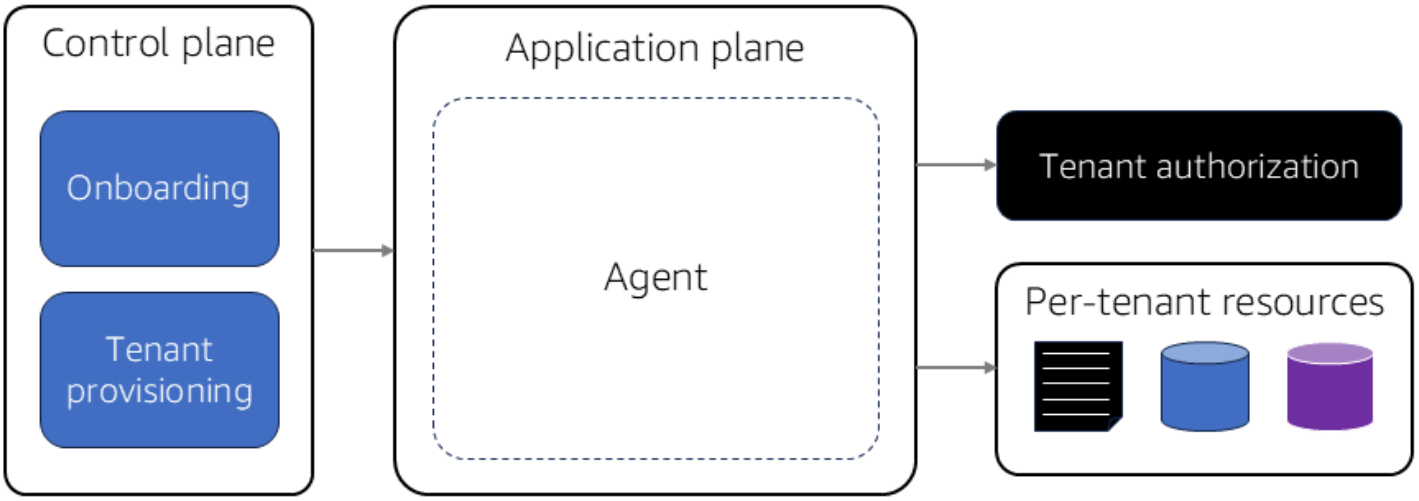
Wir gehen davon aus, dass es sich bei Agenten um unabhängige Dienste handelt, die in einer vom Anbieter gehosteten Umgebung ausgeführt werden. Dies kann jedoch in einem Szenario unklar sein, in dem ein Kunde von Agenten mehr Einschränkungen hinsichtlich der Art und Weise auferlegt, wie und wo ein Agent untergebracht werden soll.

Aus Mietern werden Agenten

Das Onboarding ist in der Regel ein wichtiger Bestandteil jeder AaaS-Umgebung. Die Art und Weise, wie Sie Mandanten erstellen, konfigurieren und bereitstellen, beinhaltet oft viele bewegliche Teile, Integrationen und Tools. Für das Onboarding von Agenten sind möglicherweise dieselben Dienste

erforderlich wie in einer AaaS-Steuerebene. Dazu gehören Mandantenidentität, Tiering, Bereitstellung von Ressourcen pro Mandant und Konfiguration von Mandantenrichtlinien.

Ihr Ansatz beim Onboarding von Agenten wird durch den Platzbedarf und das Mietermodell Ihrer Agentenumgebung beeinflusst. Isolierte und gepoolte Agenten haben jeweils ihre eigenen Nuancen, und die Wahl, entweder einen einzelnen Agenten oder mehrere Agenten zu verwenden, wirkt sich auch auf den Onboarding-Prozess aus. Das folgende Diagramm zeigt in einer konzeptionellen Ansicht, wie sich das Onboarding auf die Konfiguration eines Agenten auswirkt.



Jedes Mal, wenn Sie einen Agenten an Bord nehmen, muss die Kontrollebene die notwendigen Schritte unternehmen, damit der Mandant auf den Agenten zugreifen kann. Die Art und Weise, wie Mandanten eingeführt werden, hängt vom Autorisierungsmodell für Agenten ab. Gehen Sie jedoch davon aus, dass Sie eine Mandantenidentität erstellen, die Agentenanfragen einzelnen Mandanten zuordnet. Dieser Mandantenkontext bestimmt die Erfahrung der Agenten, indem er auf Routen, Bereiche und Zugriffskontrolle angewendet wird.

Beim Onboarding müssen Sie möglicherweise auch alle Ressourcen pro Mandant konfigurieren, die ein Agent verwendet. Hier verbindet der Mandantenbereitstellungsdienst der Steuerungsebene Ihren Agenten mit mandantenspezifischen Daten und Ressourcen, die der Agent abfragt.

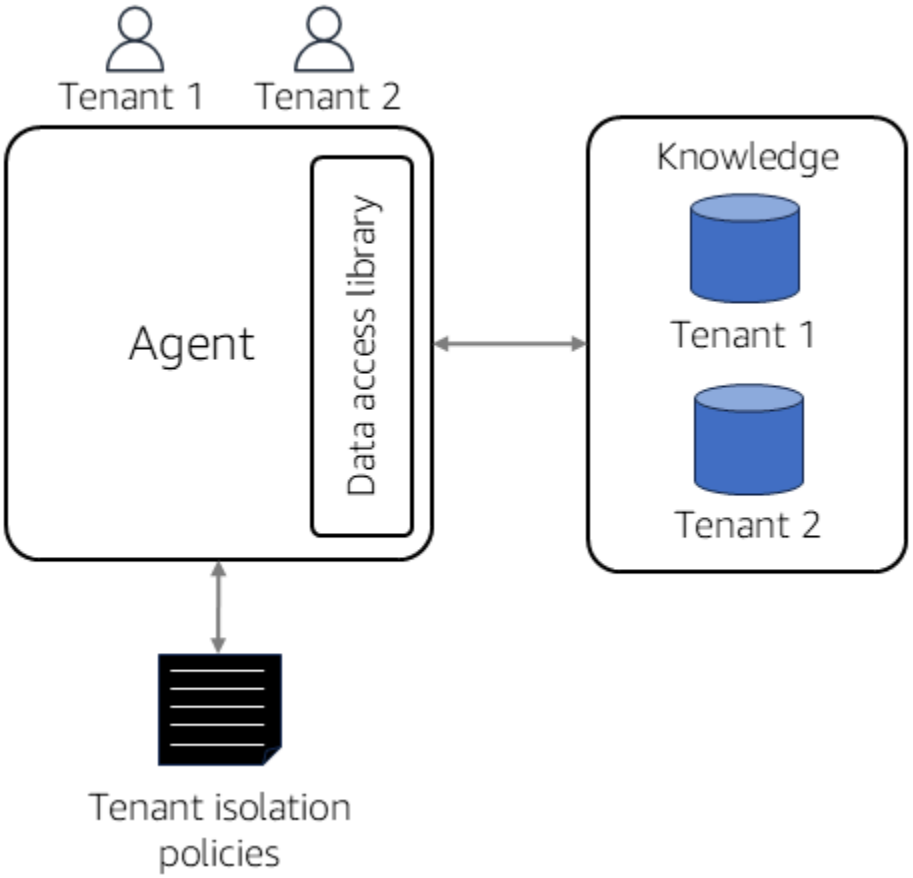
Wenn Ihr System auf der Integration von Agenten von Drittanbietern angewiesen ist, müssen Sie während des Onboarding-Prozesses auch auf die Bedürfnisse dieser Agenten eingehen. Wie das funktioniert, hängt von den Sicherheits- und Integrationsmechanismen für die Autorisierung des Zugriffs zwischen Agenten ab. Im Idealfall werden die erforderlichen Schritte zur Orchestrierung und Konfiguration von agent-to-agent Authentifizierung und Autorisierung durch automatisiertes Onboarding erledigt.

Durchsetzung der Mandantenisolierung

Die Mandantenisolierung ist ein Konzept, das für alle Einstellungen mit mehreren Mandanten gilt. Das bedeutet, dass Ihre Richtlinien und Strategien sicherstellen, dass ein Mandant nicht auf Ressourcen anderer Mandanten zugreifen kann. Für Agenten mit mehreren Mandanten müssen Sie möglicherweise Konstrukte und Mechanismen einführen, die dazu beitragen, die Anforderungen an die Mandantenisolierung und die Isolierung der Agenten durchzusetzen.

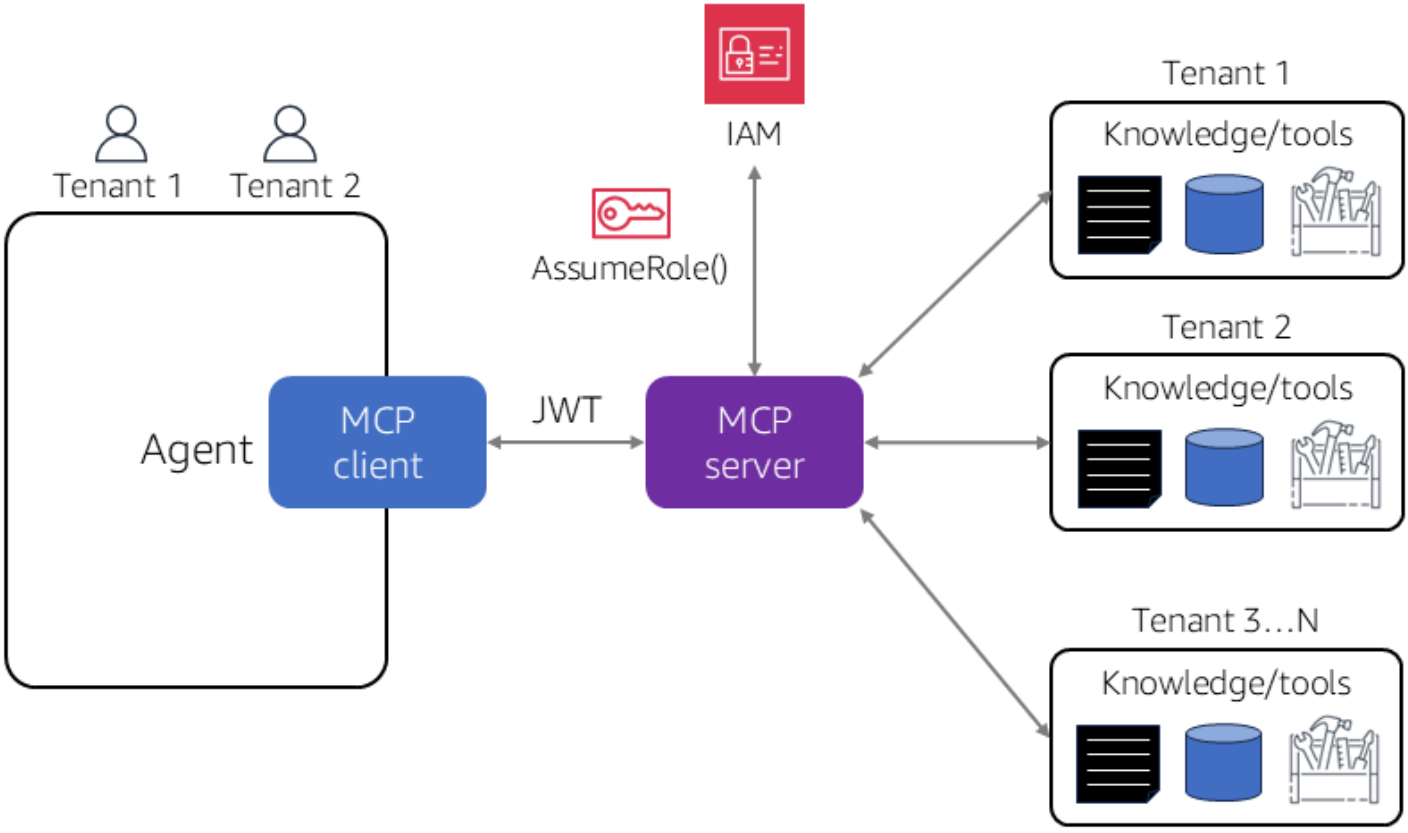
Die Anwendung der Mandantenisolierung ähnelt anderen Strategien, bei denen herkömmliche Systeme für mehrere Mandanten verwendet werden. Im Allgemeinen sollten Sie beim Aufbau einer AaaS-Architektur jeden Bereich in Ihrem System identifizieren, in dem eine Anfrage oder Aktion auf Ressourcen zugreifen kann, um festzustellen, ob die Anfrage Mandantengrenzen überschreitet. Microservices können beispielsweise Abhängigkeiten von dedizierten Amazon DynamoDB-Tabellen pro Mandant haben. Dazu müssen Sie Richtlinien einführen, die sicherstellen, dass kein anderer Mandant auf die Tabelle eines Mandanten zugreifen kann.

In diesem Fall sollten Sie die Mandantenisolierung aus Sicht eines Agenten und dessen Interaktionen mit allen Ressourcen pro Mandant in Betracht ziehen. Das folgende Diagramm zeigt ein konzeptionelles Beispiel dafür, wie Agenten Richtlinien zur Mandantenisolierung anwenden, um den Zugriff auf Mandantenressourcen zu kontrollieren.



Auf der rechten Seite dieses Diagramms verfügt der Agent über Informationen pro Mandant, die in separaten Vektordatenbanken gespeichert sind. Während der Agent eine Anfrage bearbeitet, untersucht er den Kontext des Mandanten, der die Anfrage stellt. Auf dieser Grundlage wendet der Agent eine geeignete Isolationsrichtlinie an, um sicherzustellen, dass Mandanten nicht auf Daten oder Ressourcen außerhalb ihrer festgelegten Grenzen zugreifen können.

Wenn Ihr Agent ein Model Context Protocol (MCP) verwendet, kann er auch Ihr Modell zur Mandantenisolierung implementieren. Das folgende Diagramm zeigt ein Beispiel für die Einführung von MCP und die Anwendung von Isolationsrichtlinien.



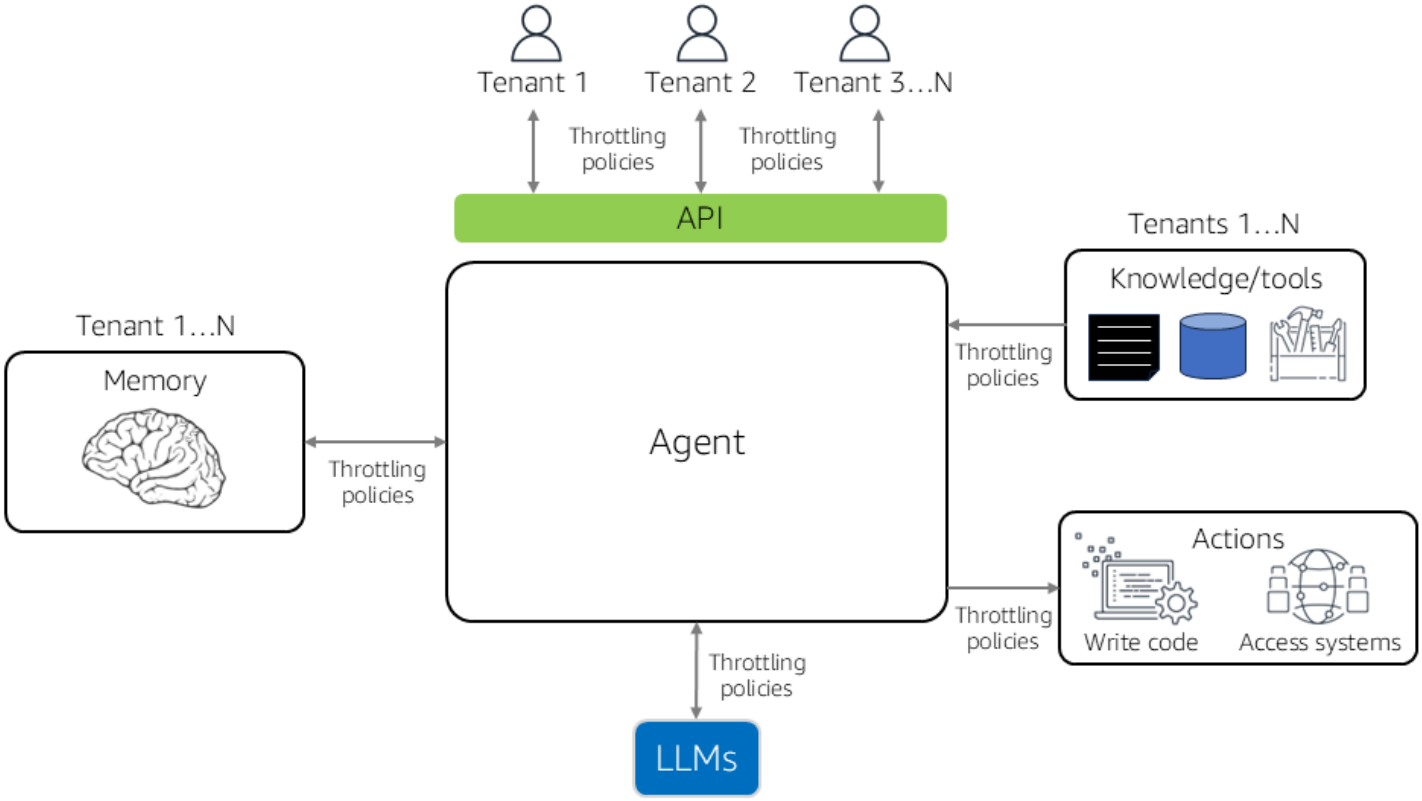
MCP ist ein standardisiertes Protokoll, das ein Agent zur Integration mit beliebigen Tools, Daten und Ressourcen verwendet. In diesem Beispiel interagieren ein MCP-Client und ein MCP-Server mit den mandantenspezifischen Kenntnissen und Tools, die auf der rechten Seite des Diagramms dargestellt sind. Der Mandantenkontext fließt vom Client zum Server, und der Server verwendet diesen Kontext, um mandantenbezogene Anmeldeinformationen vom (IAM-) Dienst abzurufen. AWS Identity and Access Management Die Anmeldeinformationen steuern den Zugriff auf die Ressourcen der einzelnen Mandanten und stellen so sicher, dass ein Mandant auf die Ressourcen eines anderen Mandanten zugreifen kann.

Da Agenten Mehrmandantenfähigkeit einsetzen, müssen sie Mechanismen einführen, die bei der Bearbeitung von Anfragen Richtlinien zur Mandantenisolierung anwenden. In einigen Fällen kann IAM dazu beitragen, den Zugriff auf Mandantenressourcen einzuschränken. In anderen Fällen müssen Sie möglicherweise andere Tools oder Frameworks einführen, um Richtlinien zur Mandantenisolierung anzuwenden.

Lauter Nachbar und Agenten

In einer AaaS-Umgebung mit mehreren Mandanten, in der sich mehrere Mandanten einen Agenten teilen, sollten Sie darüber nachdenken, wo und wie Richtlinien eingeführt werden können, um störende Umgebungsgeräusche zu verhindern. Mit Richtlinien kann eine allgemeine Drosselung eingeführt werden, die für den gesamten Verbrauch gilt, oder Sie können mandanten- oder stufenbasierte Richtlinien einrichten, die eine Drosselung auf der Grundlage einer bestimmten Persona anwenden. Sie könnten Mietern der Basisklasse stärkere Nutzungsbeschränkungen auferlegen als Mietern der Premiumklasse.

Dieses Konzept der Drosselung kann an mehreren Architekturpunkten angewendet werden. Das folgende Diagramm zeigt ein Beispiel für einige mögliche Bereiche, in denen Noisy-Neighbor-Richtlinien eingeführt werden könnten.



In unserer vorherigen Überprüfung der Implementierung mehrerer Agenten haben wir verschiedene Ressourcen untersucht, die Ihr Agent nutzen kann, und dabei das Potenzial für Ressourcen pro Mandant innerhalb eines Agenten hervorgehoben. Jeder Kontaktpunkt ist ein potenzieller Bereich, in dem Drosselungsrichtlinien eingeführt werden können, um sicherzustellen, dass Mieter die Nutzungslimits Ihres Systems oder die Staffelungsrichtlinien eines Mandanten nicht überschreiten.

Am besten lassen sich Schutzmaßnahmen gegen laute Nachbarn an Stellen in der Architektur einführen, an denen Mieter Ressourcen gemeinsam nutzen. Diese gemeinsam genutzten oder gepoolten Komponenten, wie Rechenleistung, Arbeitsspeicher und umfangreiche Sprachmodelle APIs, sind am anfälligsten für Leistungseinbußen, wenn ein einzelner Mandant unverhältnismäßig viel verbraucht.

Ein natürlicher Ort, an dem Drosselung angewendet wird, ist der Eintrittspunkt des Agenten, der manchmal auch als „äußerer Rand“ bezeichnet wird. Hier können Sie globale Limits oder tenant-tier-based Ratenlimits festlegen, bevor der Agent mit der Bearbeitung der Anfrage beginnt. Die Drosselung kann auch tiefer im Ausführungspfad erfolgen, z. B. wenn der Agent ein LLM aufruft, auf Speicher zugreift oder gemeinsam genutzte Tools aufruft.

Diese Richtlinien helfen Ihnen dabei, eine faire Nutzung durchzusetzen, die Widerstandsfähigkeit der Agenten bei hoher Auslastung aufrechtzuerhalten und für ein einheitliches Nutzererlebnis bei allen Mandanten zu sorgen. Je nach Ihren Zielen können Sie sich auf den allgemeinen Systemschutz (Resilienz) oder auf die detaillierte Verwaltung des Nutzererlebnisses konzentrieren (z. B. mit stufenbasierten Berechtigungen).

Daten, Betrieb und Tests

Agenten und Dateneigentum

Ein Überblick über die Implementierung von Agenten zeigt Szenarien auf, in denen sich ein Agent auf die Daten eines bestimmten Mandanten stützt. Betrachten Sie in diesem Fall den Datenlebenszyklus und, was noch wichtiger ist, wo sie gespeichert werden. Dies ist besonders wichtig für Branchen und Anwendungsfälle, in denen die Art der Daten beeinflusst, wie ein Agent auf sie zugreift.

AaaS-Anbieter müssen prüfen, wie Datenprobleme in einer Umgebung mit mehreren Mandanten gelöst werden können, was sich auf das Onboarding, die Isolierung und den Betrieb eines Agenten auswirken kann. Die anwendbaren Nuancen und Strategien variieren je nach den Tools, Technologien und Daten, die Sie nutzen. Sie können dies auf viele Arten angehen, was Sie bei der Erstellung eines AaaS-Angebots beachten sollten.

Agentenbetrieb für mehrere Mandanten

Denken Sie beim Aufbau von Agentenumgebungen darüber nach, wie Sie Ihre Agenten bedienen und verwalten können. Als Anbieter benötigen Sie Metriken, Daten, Einblicke und Protokolle, mit denen Sie den Zustand, die Größe und die Aktivität eines Agenten überwachen können. Dies ist in einer Agentenumgebung mit mehreren Mandanten ausgeprägter, in der Sie verstehen möchten, wie einzelne Mandanten Agentenressourcen verbrauchen.

Dies ist in Umgebungen mit mehreren Agenten noch wichtiger, wenn Sie Einblicke in die Interaktionen der Agenten benötigen. Die Möglichkeit, Aktivitäten zwischen Agenten zu profilieren und nachzuverfolgen, kann für die Behebung von Problemen, die sich auf den Umfang, die Genauigkeit und die Effizienz Ihres Systems auswirken, von entscheidender Bedeutung sein.

Betriebsteams können auch LLM-Interaktionen profilieren, um sich ein besseres Bild von den Belastungen zu machen, denen Agenten ausgesetzt sind. LLMs Diese Daten sind für die Optimierung der Agentenimplementierung unerlässlich. Sie können den Betriebsteams auch einen Überblick darüber geben, wie sich Agenten und Mietverhältnisse auf das Gesamtkostenprofil eines Systems auswirken.

Schulung und Test von Multi-Tenant-Agenten

Eine Herausforderung im Zusammenhang mit Gebäudeagenten besteht darin, dass von ihnen erwartet wird, dass sie lernen und sich weiterentwickeln. Das bedeutet auch, dass wir unseren Baustoff testen, ihn verfeinern und seine Genauigkeit verbessern müssen, bevor wir ihn in die Produktion überführen. Es gibt viele Bereiche, in denen Sie überprüfen und beurteilen können, ob Ihr Mitarbeiter die Absicht richtig einschätzt und kategorisiert oder geeignete Tools und Maßnahmen auswählt und einleitet. Die Liste der Variablen ist umfangreich, aber letztendlich geht es darum, sicherzustellen, dass Ihr Agent Ergebnisse erzielt, mit denen Sie Ihre Ziele erreichen.

Es würde den Rahmen dieses Dokuments sprengen, alle wichtigen Aspekte und Prinzipien im Zusammenhang mit dem Testen von Agents zu untersuchen. Beachten Sie jedoch, dass Teststrategien die Komplexität von AaaS-Umgebungen mit mehreren Mandanten erhöhen. Wenn ein Agent beispielsweise über Daten, Speicher und andere Konstrukte verfügt, die kontextuell auf jeden Mandanten angewendet werden, können die Ergebnisse eines Agenten durch die Ressourcen pro Mandant beeinflusst werden.

Wenn Sie einen Agenten verwenden, um ein Szenario zu simulieren, müssen Sie Ihre Simulation möglicherweise für mandantenspezifische Anwendungsfälle erweitern. Dementsprechend müssen Sie die Validierungsverfahren verfeinern, um Fälle zu berücksichtigen, in denen die Validierungskriterien für jeden Mandanten unterschiedlich sind.

Überlegungen und Diskussion

Wo passt SaaS?

Branchenexperten diskutieren aktiv darüber, wie Agenten die Software-as-a-Service (SaaS) - Landschaft beeinflussen. Es stimmt zwar, dass Agenten die Software für viele Systeme ändern, aber es ist weit hergeholt zu behaupten, dass Agenten Bereitstellungsmodelle überflüssig machen. Einige SaaS-Anbieter werden wahrscheinlich durch die Einführung von Agenten auf den Kopf gestellt werden, und andere werden ihr Leistungsversprechen möglicherweise völlig überdenken, indem sie sich auf ein Agent-as-a-Service-Modell (AaaS) stützen. Andere könnten ein Gleichgewicht finden, indem sie selektiv Agenten einsetzen, die auf spezifische Bedürfnisse zugeschnitten sind.

Dieses Thema ist interessant, da die Übernahme der besten SaaS-Prinzipien die nächste Entwicklung von SaaS darstellen könnte. Dies könnte bedeuten, dass SaaS auf dem Vormarsch ist, oder es könnte bedeuten, dass die grundlegenden Prinzipien von SaaS in einem agentenbasierten Modell zusammengefasst und umgesetzt werden. Es ist wahrscheinlich weniger wichtig zu entscheiden, wo die Terminologie letztendlich landet, aber es ist unwahrscheinlich, dass SaaS als Konzept verschwinden wird. Es ist wahrscheinlicher, dass Agenten den SaaS-Fußabdruck prägen werden.

Letztlich müssen wir entscheiden, welche Strategien auf AaaS angewendet werden können, was bedeutet, dass Unternehmen in die Lage versetzt werden müssen, Agentenarchitekturen und Geschäftsstrategien einzuführen, damit Anbieter die Effizienz, den Wert und die Wirkung ihrer Agentensysteme maximieren können. Agenten sind keine Blackboxen. Agenten verbrauchen Ressourcen, skalieren Abläufe, sind von Daten abhängig und verursachen Kosten — alles Faktoren, mit denen sich Anbieter auseinandersetzen müssen. Agentenanbieter müssen prüfen, wie Mehrmandanten-Prinzipien ihre Serviceangebote prägen und Betriebsmodelle optimieren können.

Erklärung

Die Agentenlandschaft entwickelt sich ständig weiter, wobei die Designs je nach Domäne, beabsichtigten Anwendungsfällen und Zielbranchen variieren. Ein Teil dieser Entwicklung beinhaltet die weitere Verfeinerung unseres Verständnisses von Strategien, Mustern und Kompromissen, die Architekten bei der Entwicklung und Herstellung von Agenten berücksichtigen.

Eine umfassende Agentenstrategie muss sowohl auf die geschäftlichen als auch auf die technischen Ziele abgestimmt sein. Dazu gehören die Definition von Zielmärkten und Personas, die Festlegung

von Preis- und Ressourcenmanagementstrategien und die Festlegung, wie Agenten in größere Systeme passen. Diese Überlegungen sind besonders wichtig bei der Bereitstellung von AaaS, bei der Skalierbarkeit, Kosteneffizienz und Innovation die Hauptziele sind.

Operative Fähigkeiten sind ebenso wichtig. Die Umgebung muss die Überwachung von Agentenaktivitäten, Integritätskennzahlen und Nutzungsmustern unterstützen. Dies wird in Systemen mit mehreren Agenten komplexer, bei denen der Betrieb zwischen unabhängigen Agenten koordiniert werden muss.

Insgesamt kratzt diese Diskussion über Agenten nur an der Oberfläche der verschiedenen architektonischen Überlegungen, die Teil agentischer Systeme sein könnten. Neben der Auswahl geeigneter Tools und Frameworks hängt der Erfolg auch von der Schaffung einer Architektur ab LLMs, die die Geschäftsanforderungen an Skalierbarkeit, Effizienz, Bereitstellung und Mehrmandantenfähigkeit erfüllt.

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Erste Veröffentlichung	—	14. Juli 2025

AWS Glossar zu präskriptiven Leitlinien

Die folgenden Begriffe werden häufig in Strategien, Leitfäden und Mustern von AWS Prescriptive Guidance verwendet. Um Einträge vorzuschlagen, verwenden Sie bitte den Link Feedback geben am Ende des Glossars.

Zahlen

7 Rs

Sieben gängige Migrationsstrategien für die Verlagerung von Anwendungen in die Cloud. Diese Strategien bauen auf den 5 Rs auf, die Gartner 2011 identifiziert hat, und bestehen aus folgenden Elementen:

- Faktorwechsel/Architekturwechsel – Verschieben Sie eine Anwendung und ändern Sie ihre Architektur, indem Sie alle Vorteile cloudnativer Feature nutzen, um Agilität, Leistung und Skalierbarkeit zu verbessern. Dies beinhaltet in der Regel die Portierung des Betriebssystems und der Datenbank. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank auf die Amazon Aurora PostgreSQL-kompatible Edition.
- Plattformwechsel (Lift and Reshape) – Verschieben Sie eine Anwendung in die Cloud und führen Sie ein gewisses Maß an Optimierung ein, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Amazon Relational Database Service (Amazon RDS) für Oracle in der. AWS Cloud
- Neukauf (Drop and Shop) – Wechseln Sie zu einem anderen Produkt, indem Sie typischerweise von einer herkömmlichen Lizenz zu einem SaaS-Modell wechseln. Beispiel: Migrieren Sie Ihr CRM-System (Customer Relationship Management) zu Salesforce.com.
- Hostwechsel (Lift and Shift) – Verschieben Sie eine Anwendung in die Cloud, ohne Änderungen vorzunehmen, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Oracle auf einer EC2-Instanz in der. AWS Cloud
- Verschieben (Lift and Shift auf Hypervisor-Ebene) – Verlagern Sie die Infrastruktur in die Cloud, ohne neue Hardware kaufen, Anwendungen umschreiben oder Ihre bestehenden Abläufe ändern zu müssen. Sie migrieren Server von einer lokalen Plattform zu einem Cloud-Dienst für dieselbe Plattform. Beispiel: Migrieren Sie eine Microsoft Hyper-V Anwendung zu AWS.
- Beibehaltung (Wiederaufgreifen) – Bewahren Sie Anwendungen in Ihrer Quellumgebung auf. Dazu können Anwendungen gehören, die einen umfangreichen Faktorwechsel erfordern und

die Sie auf einen späteren Zeitpunkt verschieben möchten, sowie ältere Anwendungen, die Sie beibehalten möchten, da es keine geschäftliche Rechtfertigung für ihre Migration gibt.

- Außerbetriebnahme – Dekommissionierung oder Entfernung von Anwendungen, die in Ihrer Quellumgebung nicht mehr benötigt werden.

A

ABAC

Siehe attributbasierte Zugriffskontrolle.

abstrahierte Dienste

Siehe Managed Services.

ACID

Siehe Atomarität, Konsistenz, Isolierung und Haltbarkeit.

Aktiv-Aktiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden (mithilfe eines bidirektionalen Replikationstools oder dualer Schreibvorgänge) und beide Datenbanken Transaktionen von miteinander verbundenen Anwendungen während der Migration verarbeiten. Diese Methode unterstützt die Migration in kleinen, kontrollierten Batches, anstatt einen einmaligen Cutover zu erfordern. Es ist flexibler, erfordert aber mehr Arbeit als eine aktiv-passive Migration.

Aktiv-Passiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden, aber nur die Quelldatenbank verarbeitet Transaktionen von verbindenden Anwendungen, während Daten in die Zieldatenbank repliziert werden. Die Zieldatenbank akzeptiert während der Migration keine Transaktionen.

Aggregatfunktion

Eine SQL-Funktion, die mit einer Gruppe von Zeilen arbeitet und einen einzelnen Rückgabewert für die Gruppe berechnet. Beispiele für Aggregatfunktionen sind `SUM` und`MAX`.

AI

Siehe künstliche Intelligenz.

AIOps

Siehe Operationen im Bereich künstliche Intelligenz.

Anonymisierung

Der Prozess des dauerhaften Löschens personenbezogener Daten in einem Datensatz. Anonymisierung kann zum Schutz der Privatsphäre beitragen. Anonymisierte Daten gelten nicht mehr als personenbezogene Daten.

Anti-Muster

Eine häufig verwendete Lösung für ein wiederkehrendes Problem, bei dem die Lösung kontraproduktiv, ineffektiv oder weniger wirksam als eine Alternative ist.

Anwendungssteuerung

Ein Sicherheitsansatz, bei dem nur zugelassene Anwendungen verwendet werden können, um ein System vor Schadsoftware zu schützen.

Anwendungsportfolio

Eine Sammlung detaillierter Informationen zu jeder Anwendung, die von einer Organisation verwendet wird, einschließlich der Kosten für die Erstellung und Wartung der Anwendung und ihres Geschäftswerts. Diese Informationen sind entscheidend für den Prozess der Portfoliofindung und -analyse und hilft bei der Identifizierung und Priorisierung der Anwendungen, die migriert, modernisiert und optimiert werden sollen.

künstliche Intelligenz (KI)

Das Gebiet der Datenverarbeitungswissenschaft, das sich der Nutzung von Computertechnologien zur Ausführung kognitiver Funktionen widmet, die typischerweise mit Menschen in Verbindung gebracht werden, wie Lernen, Problemlösen und Erkennen von Mustern. Weitere Informationen finden Sie unter Was ist künstliche Intelligenz?

Operationen mit künstlicher Intelligenz (AIOps)

Der Prozess des Einsatzes von Techniken des Machine Learning zur Lösung betrieblicher Probleme, zur Reduzierung betrieblicher Zwischenfälle und menschlicher Eingriffe sowie zur Steigerung der Servicequalität. Weitere Informationen zur Verwendung in der AWS Migrationsstrategie finden Sie im Operations Integration Guide. AIOps

Asymmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der ein Schlüsselpaar, einen öffentlichen Schlüssel für die Verschlüsselung und einen privaten Schlüssel für die Entschlüsselung verwendet. Sie können den

öffentlichen Schlüssel teilen, da er nicht für die Entschlüsselung verwendet wird. Der Zugriff auf den privaten Schlüssel sollte jedoch stark eingeschränkt sein.

Atomizität, Konsistenz, Isolierung, Haltbarkeit (ACID)

Eine Reihe von Softwareeigenschaften, die die Datenvalidität und betriebliche Zuverlässigkeit einer Datenbank auch bei Fehlern, Stromausfällen oder anderen Problemen gewährleisten.

Attributbasierte Zugriffskontrolle (ABAC)

Die Praxis, detaillierte Berechtigungen auf der Grundlage von Benutzerattributen wie Abteilung, Aufgabenrolle und Teamname zu erstellen. Weitere Informationen finden Sie unter ABAC AWS in der AWS Identity and Access Management (IAM-) Dokumentation.

autoritative Datenquelle

Ein Ort, an dem Sie die primäre Version der Daten speichern, die als die zuverlässigste Informationsquelle angesehen wird. Sie können Daten aus der maßgeblichen Datenquelle an andere Speicherorte kopieren, um die Daten zu verarbeiten oder zu ändern, z. B. zu anonymisieren, zu redigieren oder zu pseudonymisieren.

Availability Zone

Ein bestimmter Standort innerhalb einer AWS-Region , der vor Ausfällen in anderen Availability Zones geschützt ist und kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben Region bietet.

AWS Framework für die Einführung der Cloud (AWS CAF)

Ein Framework mit Richtlinien und bewährten Verfahren, das Unternehmen bei der Entwicklung eines effizienten und effektiven Plans für die erfolgreiche Umstellung auf die Cloud unterstützt. AWS AWS CAF unterteilt die Leitlinien in sechs Schwerpunktbereiche, die als Perspektiven bezeichnet werden: Unternehmen, Mitarbeiter, Unternehmensführung, Plattform, Sicherheit und Betrieb. Die Perspektiven Geschäft, Mitarbeiter und Unternehmensführung konzentrieren sich auf Geschäftskompetenzen und -prozesse, während sich die Perspektiven Plattform, Sicherheit und Betriebsabläufe auf technische Fähigkeiten und Prozesse konzentrieren. Die Personalperspektive zielt beispielsweise auf Stakeholder ab, die sich mit Personalwesen (HR), Personalfunktionen und Personalmanagement befassen. Aus dieser Perspektive bietet AWS CAF Leitlinien für Personalentwicklung, Schulung und Kommunikation, um das Unternehmen auf eine erfolgreiche Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie auf der AWS -CAF-Webseite und dem AWS -CAF-Whitepaper.

AWS Workload-Qualifizierungsrahmen (AWS WQF)

Ein Tool, das Workloads bei der Datenbankmigration bewertet, Migrationsstrategien empfiehlt und Arbeitsschätzungen bereitstellt. AWS WQF ist in () enthalten. AWS Schema Conversion Tool AWS SCT Es analysiert Datenbankschemas und Codeobjekte, Anwendungscode, Abhängigkeiten und Leistungsmerkmale und stellt Bewertungsberichte bereit.

B

schlechter Bot

Ein Bot, der Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen soll.

BCP

Siehe Planung der Geschäftskontinuität.

Verhaltensdiagramm

Eine einheitliche, interaktive Ansicht des Ressourcenverhaltens und der Interaktionen im Laufe der Zeit. Sie können ein Verhaltensdiagramm mit Amazon Detective verwenden, um fehlgeschlagene Anmeldeversuche, verdächtige API-Aufrufe und ähnliche Vorgänge zu untersuchen. Weitere Informationen finden Sie unter Daten in einem Verhaltensdiagramm in der Detective-Dokumentation.

Big-Endian-System

Ein System, welches das höchstwertige Byte zuerst speichert. Siehe auch Endianness.

Binäre Klassifikation

Ein Prozess, der ein binäres Ergebnis vorhersagt (eine von zwei möglichen Klassen). Beispielsweise könnte Ihr ML-Modell möglicherweise Probleme wie „Handelt es sich bei dieser E-Mail um Spam oder nicht?“ vorhersagen müssen oder „Ist dieses Produkt ein Buch oder ein Auto?“

Bloom-Filter

Eine probabilistische, speichereffiziente Datenstruktur, mit der getestet wird, ob ein Element Teil einer Menge ist.

Blau/Grün-Bereitstellung

Eine Bereitstellungsstrategie, bei der Sie zwei separate, aber identische Umgebungen erstellen. Sie führen die aktuelle Anwendungsversion in einer Umgebung (blau) und die neue

Anwendungsversion in der anderen Umgebung (grün) aus. Mit dieser Strategie können Sie schnell und mit minimalen Auswirkungen ein Rollback durchführen.

Bot

Eine Softwareanwendung, die automatisierte Aufgaben über das Internet ausführt und menschliche Aktivitäten oder Interaktionen simuliert. Manche Bots sind nützlich oder nützlich, wie z. B. Webcrawler, die Informationen im Internet indexieren. Einige andere Bots, sogenannte bösartige Bots, sollen Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen.

Botnetz

Netzwerke von Bots, die mit Malware infiziert sind und unter der Kontrolle einer einzigen Partei stehen, die als Bot-Herder oder Bot-Operator bezeichnet wird. Botnetze sind der bekannteste Mechanismus zur Skalierung von Bots und ihrer Wirkung.

branch

Ein containerisierter Bereich eines Code-Repositorys. Der erste Zweig, der in einem Repository erstellt wurde, ist der Hauptzweig. Sie können einen neuen Zweig aus einem vorhandenen Zweig erstellen und dann Feature entwickeln oder Fehler in dem neuen Zweig beheben. Ein Zweig, den Sie erstellen, um ein Feature zu erstellen, wird allgemein als Feature-Zweig bezeichnet. Wenn das Feature zur Veröffentlichung bereit ist, führen Sie den Feature-Zweig wieder mit dem Hauptzweig zusammen. Weitere Informationen finden Sie unter Über Branches (GitHub Dokumentation).

Zugang durch Glasbruch

Unter außergewöhnlichen Umständen und im Rahmen eines genehmigten Verfahrens ist dies eine schnelle Methode für einen Benutzer, auf einen Bereich zuzugreifen AWS-Konto , für den er normalerweise keine Zugriffsrechte besitzt. Weitere Informationen finden Sie unter dem Indikator Implementation break-glass procedures in den AWS Well-Architected-Leitlinien.

Brownfield-Strategie

Die bestehende Infrastruktur in Ihrer Umgebung. Wenn Sie eine Brownfield-Strategie für eine Systemarchitektur anwenden, richten Sie sich bei der Gestaltung der Architektur nach den Einschränkungen der aktuellen Systeme und Infrastruktur. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und Greenfield-Strategien mischen.

Puffer-Cache

Der Speicherbereich, in dem die am häufigsten abgerufenen Daten gespeichert werden.

Geschäftsfähigkeit

Was ein Unternehmen tut, um Wert zu generieren (z. B. Vertrieb, Kundenservice oder Marketing). Microservices-Architekturen und Entwicklungsentscheidungen können von den Geschäftskapazitäten beeinflusst werden. Weitere Informationen finden Sie im Abschnitt Organisiert nach Geschäftskapazitäten des Whitepapers Ausführen von containerisierten Microservices in AWS.

Planung der Geschäftskontinuität (BCP)

Ein Plan, der die potenziellen Auswirkungen eines störenden Ereignisses, wie z. B. einer groß angelegten Migration, auf den Betrieb berücksichtigt und es einem Unternehmen ermöglicht, den Betrieb schnell wieder aufzunehmen.

C

CAF

Weitere Informationen finden Sie unter Framework AWS für die Cloud-Einführung.

Bereitstellung auf Kanaren

Die langsame und schrittweise Veröffentlichung einer Version für Endbenutzer. Wenn Sie sich sicher sind, stellen Sie die neue Version bereit und ersetzen die aktuelle Version vollständig.

CCoE

Weitere Informationen finden Sie im Cloud Center of Excellence.

CDC

Siehe Erfassung von Änderungsdaten.

Erfassung von Datenänderungen (CDC)

Der Prozess der Nachverfolgung von Änderungen an einer Datenquelle, z. B. einer Datenbanktabelle, und der Aufzeichnung von Metadaten zu der Änderung. Sie können CDC für verschiedene Zwecke verwenden, z. B. für die Prüfung oder Replikation von Änderungen in einem Zielsystem, um die Synchronisation aufrechtzuerhalten.

Chaos-Technik

Absichtliches Einführen von Ausfällen oder Störungsereignissen, um die Widerstandsfähigkeit eines Systems zu testen. Sie können AWS Fault Injection Service (AWS FIS) verwenden, um Experimente durchzuführen, die Ihre AWS Workloads stress, und deren Reaktion zu bewerten.

CI/CD

Siehe Continuous Integration und Continuous Delivery.

Klassifizierung

Ein Kategorisierungsprozess, der bei der Erstellung von Vorhersagen hilft. ML-Modelle für Klassifikationsprobleme sagen einen diskreten Wert voraus. Diskrete Werte unterscheiden sich immer voneinander. Beispielsweise muss ein Modell möglicherweise auswerten, ob auf einem Bild ein Auto zu sehen ist oder nicht.

clientseitige Verschlüsselung

Lokale Verschlüsselung von Daten, bevor das Ziel sie AWS-Service empfängt.

Cloud-Exzellenzzentrum (CCoE)

Ein multidisziplinäres Team, das die Cloud-Einführung in der gesamten Organisation vorantreibt, einschließlich der Entwicklung bewährter Cloud-Methoden, der Mobilisierung von Ressourcen, der Festlegung von Migrationszeitplänen und der Begleitung der Organisation durch groß angelegte Transformationen. Weitere Informationen finden Sie in den CCoE-Beiträgen im AWS Cloud Enterprise Strategy Blog.

Cloud Computing

Die Cloud-Technologie, die typischerweise für die Ferndatenspeicherung und das IoT-Gerätemanagement verwendet wird. Cloud Computing ist häufig mit Edge-Computing-Technologie verbunden.

Cloud-Betriebsmodell

In einer IT-Organisation das Betriebsmodell, das zum Aufbau, zur Weiterentwicklung und Optimierung einer oder mehrerer Cloud-Umgebungen verwendet wird. Weitere Informationen finden Sie unter Aufbau Ihres Cloud-Betriebsmodells.

Phasen der Einführung der Cloud

Die vier Phasen, die Unternehmen bei der Migration in der Regel durchlaufen AWS Cloud:

- Projekt – Durchführung einiger Cloud-bezogener Projekte zu Machbarkeitsnachweisen und zu Lernzwecken
- Fundament — Tätigen Sie grundlegende Investitionen, um Ihre Cloud-Einführung zu skalieren (z. B. Einrichtung einer landing zone, Definition eines CCo E, Einrichtung eines Betriebsmodells)

- Migration – Migrieren einzelner Anwendungen
- Neuentwicklung – Optimierung von Produkten und Services und Innovation in der Cloud

Diese Phasen wurden von Stephen Orban im Blogbeitrag The Journey Toward Cloud-First & the Stages of Adoption im AWS Cloud Enterprise Strategy-Blog definiert. Informationen darüber, wie sie mit der AWS Migrationsstrategie zusammenhängen, finden Sie im Leitfaden zur Vorbereitung der Migration.

CMDB

Siehe Datenbank für das Konfigurationsmanagement.

Code-Repository

Ein Ort, an dem Quellcode und andere Komponenten wie Dokumentation, Beispiele und Skripts gespeichert und im Rahmen von Versionskontrollprozessen aktualisiert werden. Zu den gängigen Cloud-Repositorys gehören GitHub oderBitbucket Cloud. Jede Version des Codes wird Zweig genannt. In einer Microservice-Struktur ist jedes Repository einer einzelnen Funktionalität gewidmet. Eine einzelne CI/CD-Pipeline kann mehrere Repositorien verwenden.

Kalter Cache

Ein Puffer-Cache, der leer oder nicht gut gefüllt ist oder veraltete oder irrelevante Daten enthält. Dies beeinträchtigt die Leistung, da die Datenbank-Instance aus dem Hauptspeicher oder der Festplatte lesen muss, was langsamer ist als das Lesen aus dem Puffercache.

Kalte Daten

Daten, auf die selten zugegriffen wird und die in der Regel historisch sind. Bei der Abfrage dieser Art von Daten sind langsame Abfragen in der Regel akzeptabel. Durch die Verlagerung dieser Daten auf leistungsschwächere und kostengünstigere Speicherstufen oder -klassen können Kosten gesenkt werden.

Computer Vision (CV)

Ein Bereich der KI, der maschinelles Lernen nutzt, um Informationen aus visuellen Formaten wie digitalen Bildern und Videos zu analysieren und zu extrahieren. Amazon SageMaker AI bietet beispielsweise Bildverarbeitungsalgorithmen für CV.

Drift in der Konfiguration

Bei einer Arbeitslast eine Änderung der Konfiguration gegenüber dem erwarteten Zustand. Dies kann dazu führen, dass der Workload nicht mehr richtlinienkonform wird, und zwar in der Regel schrittweise und unbeabsichtigt.

Verwaltung der Datenbankkonfiguration (CMDB)

Ein Repository, das Informationen über eine Datenbank und ihre IT-Umgebung speichert und verwaltet, inklusive Hardware- und Softwarekomponenten und deren Konfigurationen. In der Regel verwenden Sie Daten aus einer CMDB in der Phase der Portfolioerkennung und -analyse der Migration.

Konformitätspaket

Eine Sammlung von AWS Config Regeln und Abhilfemaßnahmen, die Sie zusammenstellen können, um Ihre Konformitäts- und Sicherheitsprüfungen individuell anzupassen. Mithilfe einer YAML-Vorlage können Sie ein Conformance Pack als einzelne Entität in einer AWS-Konto AND-Region oder unternehmensweit bereitstellen. Weitere Informationen finden Sie in der Dokumentation unter Conformance Packs. AWS Config

Kontinuierliche Bereitstellung und kontinuierliche Integration (CI/CD)

Der Prozess der Automatisierung der Quell-, Build-, Test-, Staging- und Produktionsphasen des Softwareveröffentlichungsprozesses. CI/CD wird allgemein als Pipeline beschrieben. CI/CD kann Ihnen helfen, Prozesse zu automatisieren, die Produktivität zu steigern, die Codequalität zu verbessern und schneller zu liefern. Weitere Informationen finden Sie unter Vorteile der kontinuierlichen Auslieferung. CD kann auch für kontinuierliche Bereitstellung stehen. Weitere Informationen finden Sie unterKontinuierliche Auslieferung im Vergleich zu kontinuierlicher Bereitstellung.

CV

Siehe Computer Vision.

D

Daten im Ruhezustand

Daten, die in Ihrem Netzwerk stationär sind, z. B. Daten, die sich im Speicher befinden.

Datenklassifizierung

Ein Prozess zur Identifizierung und Kategorisierung der Daten in Ihrem Netzwerk auf der Grundlage ihrer Kritikalität und Sensitivität. Sie ist eine wichtige Komponente jeder Strategie für das Management von Cybersecurity-Risiken, da sie Ihnen hilft, die geeigneten Schutz- und Aufbewahrungskontrollen für die Daten zu bestimmen. Die Datenklassifizierung ist ein Bestandteil

der Sicherheitssäule im AWS Well-Architected Framework. Weitere Informationen finden Sie unter Datenklassifizierung.

Datendrift

Eine signifikante Abweichung zwischen den Produktionsdaten und den Daten, die zum Trainieren eines ML-Modells verwendet wurden, oder eine signifikante Änderung der Eingabedaten im Laufe der Zeit. Datendrift kann die Gesamtqualität, Genauigkeit und Fairness von ML-Modellvorhersagen beeinträchtigen.

Daten während der Übertragung

Daten, die sich aktiv durch Ihr Netzwerk bewegen, z. B. zwischen Netzwerkressourcen.

Datennetz

Ein architektonisches Framework, das verteilte, dezentrale Dateneigentum mit zentraler Verwaltung und Steuerung ermöglicht.

Datenminimierung

Das Prinzip, nur die Daten zu sammeln und zu verarbeiten, die unbedingt erforderlich sind. Durch Datenminimierung im AWS Cloud können Datenschutzrisiken, Kosten und der CO2-Fußabdruck Ihrer Analysen reduziert werden.

Datenperimeter

Eine Reihe präventiver Schutzmaßnahmen in Ihrer AWS Umgebung, die sicherstellen, dass nur vertrauenswürdige Identitäten auf vertrauenswürdige Ressourcen von erwarteten Netzwerken zugreifen. Weitere Informationen finden Sie unter Aufbau eines Datenperimeters auf. AWS

Vorverarbeitung der Daten

Rohdaten in ein Format umzuwandeln, das von Ihrem ML-Modell problemlos verarbeitet werden kann. Die Vorverarbeitung von Daten kann bedeuten, dass bestimmte Spalten oder Zeilen entfernt und fehlende, inkonsistente oder doppelte Werte behoben werden.

Herkunft der Daten

Der Prozess der Nachverfolgung des Ursprungs und der Geschichte von Daten während ihres gesamten Lebenszyklus, z. B. wie die Daten generiert, übertragen und gespeichert wurden.

betroffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

Data Warehouse

Ein Datenverwaltungssystem, das Business Intelligence wie Analysen unterstützt. Data Warehouses enthalten in der Regel große Mengen historischer Daten und werden in der Regel für Abfragen und Analysen verwendet.

Datenbankdefinitionssprache (DDL)

Anweisungen oder Befehle zum Erstellen oder Ändern der Struktur von Tabellen und Objekten in einer Datenbank.

Datenbankmanipulationssprache (DML)

Anweisungen oder Befehle zum Ändern (Einfügen, Aktualisieren und Löschen) von Informationen in einer Datenbank.

DDL

Siehe Datenbankdefinitionssprache.

Deep-Ensemble

Mehrere Deep-Learning-Modelle zur Vorhersage kombinieren. Sie können Deep-Ensembles verwenden, um eine genauere Vorhersage zu erhalten oder um die Unsicherheit von Vorhersagen abzuschätzen.

Deep Learning

Ein ML-Teilbereich, der mehrere Schichten künstlicher neuronaler Netzwerke verwendet, um die Zuordnung zwischen Eingabedaten und Zielvariablen von Interesse zu ermitteln.

defense-in-depth

Ein Ansatz zur Informationssicherheit, bei dem eine Reihe von Sicherheitsmechanismen und -kontrollen sorgfältig in einem Computernetzwerk verteilt werden, um die Vertraulichkeit, Integrität und Verfügbarkeit des Netzwerks und der darin enthaltenen Daten zu schützen. Wenn Sie diese Strategie anwenden AWS, fügen Sie mehrere Steuerelemente auf verschiedenen Ebenen der AWS Organizations Struktur hinzu, um die Ressourcen zu schützen. Ein defense-in-depth Ansatz könnte beispielsweise Multi-Faktor-Authentifizierung, Netzwerksegmentierung und Verschlüsselung kombinieren.

delegierter Administrator

In AWS Organizations kann ein kompatibler Dienst ein AWS Mitgliedskonto registrieren, um die Konten der Organisation und die Berechtigungen für diesen Dienst zu verwalten. Dieses Konto

wird als delegierter Administrator für diesen Service bezeichnet. Weitere Informationen und eine Liste kompatibler Services finden Sie unter Services, die mit AWS Organizations funktionieren in der AWS Organizations -Dokumentation.

Einsatz

Der Prozess, bei dem eine Anwendung, neue Feature oder Codekorrekturen in der Zielumgebung verfügbar gemacht werden. Die Bereitstellung umfasst das Implementieren von Änderungen an einer Codebasis und das anschließende Erstellen und Ausführen dieser Codebasis in den Anwendungsumgebungen.

Entwicklungsumgebung

Siehe Umgebung.

Detektivische Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, ein Ereignis zu erkennen, zu protokollieren und zu warnen, nachdem ein Ereignis eingetreten ist. Diese Kontrollen stellen eine zweite Verteidigungslinie dar und warnen Sie vor Sicherheitsereignissen, bei denen die vorhandenen präventiven Kontrollen umgangen wurden. Weitere Informationen finden Sie unter Detektivische Kontrolle in Implementierung von Sicherheitskontrollen in AWS.

Abbildung des Wertstroms in der Entwicklung (DVSM)

Ein Prozess zur Identifizierung und Priorisierung von Einschränkungen, die sich negativ auf Geschwindigkeit und Qualität im Lebenszyklus der Softwareentwicklung auswirken. DVSM erweitert den Prozess der Wertstromanalyse, der ursprünglich für Lean-Manufacturing-Praktiken konzipiert wurde. Es konzentriert sich auf die Schritte und Teams, die erforderlich sind, um durch den Softwareentwicklungsprozess Mehrwert zu schaffen und zu steigern.

digitaler Zwilling

Eine virtuelle Darstellung eines realen Systems, z. B. eines Gebäudes, einer Fabrik, einer Industrieanlage oder einer Produktionslinie. Digitale Zwillinge unterstützen vorausschauende Wartung, Fernüberwachung und Produktionsoptimierung.

Maßtabelle

In einem Sternschema eine kleinere Tabelle, die Datenattribute zu quantitativen Daten in einer Faktentabelle enthält. Bei Attributen von Dimensionstabellen handelt es sich in der Regel um Textfelder oder diskrete Zahlen, die sich wie Text verhalten. Diese Attribute werden häufig zum Einschränken von Abfragen, zum Filtern und zur Kennzeichnung von Ergebnismengen verwendet.

Katastrophe

Ein Ereignis, das verhindert, dass ein Workload oder ein System seine Geschäftsziele an seinem primären Einsatzort erfüllt. Diese Ereignisse können Naturkatastrophen, technische Ausfälle oder das Ergebnis menschlichen Handelns sein, wie z. B. unbeabsichtigte Fehlkonfigurationen oder ein Malware-Angriff.

Notfallwiederherstellung (DR)

Die Strategie und der Prozess, mit denen Sie Ausfallzeiten und Datenverluste aufgrund einer Katastrophe minimieren. Weitere Informationen finden Sie unter Disaster Recovery von Workloads unter AWS: Wiederherstellung in der Cloud im AWS Well-Architected Framework.

DML

Siehe Sprache zur Datenbankmanipulation.

Domainorientiertes Design

Ein Ansatz zur Entwicklung eines komplexen Softwaresystems, bei dem seine Komponenten mit sich entwickelnden Domains oder Kerngeschäftszielen verknüpft werden, denen jede Komponente dient. Dieses Konzept wurde von Eric Evans in seinem Buch Domaingesteuertes Design: Bewältigen der Komplexität im Herzen der Software (Boston: Addison-Wesley Professional, 2003) vorgestellt. Informationen darüber, wie Sie domaingesteuertes Design mit dem Strangler-Fig-Muster verwenden können, finden Sie unter Schrittweises Modernisieren älterer Microsoft ASP.NET (ASMX)-Webservices mithilfe von Containern und Amazon API Gateway.

DR

Siehe Disaster Recovery.

Erkennung von Driften

Verfolgung von Abweichungen von einer Basiskonfiguration. Sie können es beispielsweise verwenden, AWS CloudFormation um Abweichungen bei den Systemressourcen zu erkennen, oder Sie können AWS Control Tower damit Änderungen in Ihrer landing zone erkennen, die sich auf die Einhaltung von Governance-Anforderungen auswirken könnten.

DVSM

Siehe Abbildung des Wertstroms in der Entwicklung.

E

EDA

Siehe explorative Datenanalyse.

EDI

Siehe elektronischer Datenaustausch.

Edge-Computing

Die Technologie, die die Rechenleistung für intelligente Geräte an den Rändern eines IoT-Netzwerks erhöht. Im Vergleich zu Cloud Computing kann Edge Computing die Kommunikationslatenz reduzieren und die Reaktionszeit verbessern.

elektronischer Datenaustausch (EDI)

Der automatisierte Austausch von Geschäftsdokumenten zwischen Organisationen. Weitere Informationen finden Sie unter Was ist elektronischer Datenaustausch.

Verschlüsselung

Ein Rechenprozess, der Klartextdaten, die für Menschen lesbar sind, in Chiffretext umwandelt.

Verschlüsselungsschlüssel

Eine kryptografische Zeichenfolge aus zufälligen Bits, die von einem Verschlüsselungsalgorithmus generiert wird. Schlüssel können unterschiedlich lang sein, und jeder Schlüssel ist so konzipiert, dass er unvorhersehbar und einzigartig ist.

Endianismus

Die Reihenfolge, in der Bytes im Computerspeicher gespeichert werden. Big-Endian-Systeme speichern das höchstwertige Byte zuerst. Little-Endian-Systeme speichern das niedrigwertigste Byte zuerst.

Endpunkt

Siehe Service-Endpunkt.

Endpunkt-Services

Ein Service, den Sie in einer Virtual Private Cloud (VPC) hosten können, um ihn mit anderen Benutzern zu teilen. Sie können einen Endpunktdienst mit anderen AWS-Konten oder AWS Identity and Access Management (IAM AWS PrivateLink -) Prinzipalen erstellen und diesen

Berechtigungen gewähren. Diese Konten oder Prinzipale können sich privat mit Ihrem Endpunktservice verbinden, indem sie Schnittstellen-VPC-Endpunkte erstellen. Weitere Informationen finden Sie unter Einen Endpunkt-Service erstellen in der Amazon Virtual Private Cloud (Amazon VPC)-Dokumentation.

Unternehmensressourcenplanung (ERP)

Ein System, das wichtige Geschäftsprozesse (wie Buchhaltung, MES und Projektmanagement) für ein Unternehmen automatisiert und verwaltet.

Envelope-Verschlüsselung

Der Prozess der Verschlüsselung eines Verschlüsselungsschlüssels mit einem anderen Verschlüsselungsschlüssel. Weitere Informationen finden Sie unter Envelope-Verschlüsselung in der AWS Key Management Service (AWS KMS) -Dokumentation.

Umgebung

Eine Instance einer laufenden Anwendung. Die folgenden Arten von Umgebungen sind beim Cloud-Computing üblich:

- Entwicklungsumgebung – Eine Instance einer laufenden Anwendung, die nur dem Kernteam zur Verfügung steht, das für die Wartung der Anwendung verantwortlich ist. Entwicklungsumgebungen werden verwendet, um Änderungen zu testen, bevor sie in höhere Umgebungen übertragen werden. Diese Art von Umgebung wird manchmal als Testumgebung bezeichnet.
- Niedrigere Umgebungen – Alle Entwicklungsumgebungen für eine Anwendung, z. B. solche, die für erste Builds und Tests verwendet wurden.
- Produktionsumgebung – Eine Instance einer laufenden Anwendung, auf die Endbenutzer zugreifen können. In einer CI/CD Pipeline ist die Produktionsumgebung die letzte Bereitstellungsumgebung.
- Höhere Umgebungen – Alle Umgebungen, auf die auch andere Benutzer als das Kernentwicklungsteam zugreifen können. Dies kann eine Produktionsumgebung, Vorproduktionsumgebungen und Umgebungen für Benutzerakzeptanztests umfassen.

Epics

In der agilen Methodik sind dies funktionale Kategorien, die Ihnen helfen, Ihre Arbeit zu organisieren und zu priorisieren. Epics bieten eine allgemeine Beschreibung der Anforderungen und Implementierungsaufgaben. Zu den Sicherheitsepen AWS von CAF gehören beispielsweise Identitäts- und Zugriffsmanagement, Detektivkontrollen, Infrastruktursicherheit, Datenschutz und

Reaktion auf Vorfälle. Weitere Informationen zu Epics in der AWS -Migrationsstrategie finden Sie im Leitfaden zur Programm-Implementierung.

ERP

Siehe Enterprise Resource Planning.

Explorative Datenanalyse (EDA)

Der Prozess der Analyse eines Datensatzes, um seine Hauptmerkmale zu verstehen. Sie sammeln oder aggregieren Daten und führen dann erste Untersuchungen durch, um Muster zu finden, Anomalien zu erkennen und Annahmen zu überprüfen. EDA wird durchgeführt, indem zusammenfassende Statistiken berechnet und Datenvisualisierungen erstellt werden.

F

Faktentabelle

Die zentrale Tabelle in einem Sternschema. Sie speichert quantitative Daten über den Geschäftsbetrieb. In der Regel enthält eine Faktentabelle zwei Arten von Spalten: Spalten, die Kennzahlen enthalten, und Spalten, die einen Fremdschlüssel für eine Dimensionstabelle enthalten.

schnell scheitern

Eine Philosophie, die häufige und inkrementelle Tests verwendet, um den Entwicklungslebenszyklus zu verkürzen. Dies ist ein wichtiger Bestandteil eines agilen Ansatzes.

Grenze zur Fehlerisolierung

Dabei handelt es sich um eine Grenze AWS Cloud, z. B. eine Availability Zone AWS-Region, eine Steuerungsebene oder eine Datenebene, die die Auswirkungen eines Fehlers begrenzt und die Widerstandsfähigkeit von Workloads verbessert. Weitere Informationen finden Sie unter Grenzen zur AWS Fehlerisolierung.

Feature-Zweig

Siehe Zweig.

Features

Die Eingabedaten, die Sie verwenden, um eine Vorhersage zu treffen. In einem Fertigungskontext könnten Feature beispielsweise Bilder sein, die regelmäßig von der Fertigungslinie aus aufgenommen werden.

Bedeutung der Feature

Wie wichtig ein Feature für die Vorhersagen eines Modells ist. Dies wird in der Regel als numerischer Wert ausgedrückt, der mit verschiedenen Techniken wie Shapley Additive Explanations (SHAP) und integrierten Gradienten berechnet werden kann. Weitere Informationen finden Sie unter Interpretierbarkeit von Modellen für maschinelles Lernen mit AWS.

Featuretransformation

Daten für den ML-Prozess optimieren, einschließlich der Anreicherung von Daten mit zusätzlichen Quellen, der Skalierung von Werten oder der Extraktion mehrerer Informationssätze aus einem einzigen Datenfeld. Das ermöglicht dem ML-Modell, von den Daten profitieren. Wenn Sie beispielsweise das Datum „27.05.2021 00:15:37“ in „2021“, „Mai“, „Donnerstag“ und „15“ aufschlüsseln, können Sie dem Lernalgorithmus helfen, nuancierte Muster zu erlernen, die mit verschiedenen Datenkomponenten verknüpft sind.

Eingabeaufforderung mit wenigen Klicks

Bereitstellung einer kleinen Anzahl von Beispielen, die die Aufgabe und das gewünschte Ergebnis veranschaulichen, bevor das LLM aufgefordert wird, eine ähnliche Aufgabe auszuführen. Bei dieser Technik handelt es sich um eine Anwendung des kontextbezogenen Lernens, bei der Modelle anhand von Beispielen (Aufnahmen) lernen, die in Eingabeaufforderungen eingebettet sind. Bei Aufgaben, die spezifische Formatierungs-, Argumentations- oder Fachkenntnisse erfordern, kann die Eingabeaufforderung mit wenigen Handgriffen effektiv sein. Siehe auch Zero-Shot Prompting.

FGAC

Siehe detaillierte Zugriffskontrolle.

Feinkörnige Zugriffskontrolle (FGAC)

Die Verwendung mehrerer Bedingungen, um eine Zugriffsanfrage zuzulassen oder abzulehnen.

Flash-Cut-Migration

Eine Datenbankmigrationsmethode, bei der eine kontinuierliche Datenreplikation durch Erfassung von Änderungsdaten verwendet wird, um Daten in kürzester Zeit zu migrieren, anstatt einen schrittweisen Ansatz zu verwenden. Ziel ist es, Ausfallzeiten auf ein Minimum zu beschränken.

FM

Siehe Fundamentmodell.

Fundamentmodell (FM)

Ein großes neuronales Deep-Learning-Netzwerk, das mit riesigen Datensätzen generalisierter und unbeschrifteter Daten trainiert wurde. FMs sind in der Lage, eine Vielzahl allgemeiner Aufgaben zu erfüllen, z. B. Sprache zu verstehen, Text und Bilder zu generieren und Konversationen in natürlicher Sprache zu führen. Weitere Informationen finden Sie unter Was sind Foundation-Modelle.

G

Generative KI

Eine Untergruppe von KI-Modellen, die mit großen Datenmengen trainiert wurden und mit einer einfachen Textaufforderung neue Inhalte und Artefakte wie Bilder, Videos, Text und Audio erstellen können. Weitere Informationen finden Sie unter Was ist Generative KI.

Geoblocking

Siehe geografische Einschränkungen.

Geografische Einschränkungen (Geoblocking)

Bei Amazon eine Option CloudFront, um zu verhindern, dass Benutzer in bestimmten Ländern auf Inhaltsverteilungen zugreifen. Sie können eine Zulassungsliste oder eine Sperrliste verwenden, um zugelassene und gesperrte Länder anzugeben. Weitere Informationen finden Sie in der Dokumentation unter Beschränkung der geografischen Verteilung Ihrer Inhalte. CloudFront

Gitflow-Workflow

Ein Ansatz, bei dem niedrigere und höhere Umgebungen unterschiedliche Zweige in einem Quellcode-Repository verwenden. Der Gitflow-Workflow gilt als veraltet, und der Trunk-basierte Workflow ist der moderne, bevorzugte Ansatz.

goldenes Bild

Ein Snapshot eines Systems oder einer Software, der als Vorlage für die Bereitstellung neuer Instanzen dieses Systems oder dieser Software verwendet wird. In der Fertigung kann ein Golden Image beispielsweise zur Bereitstellung von Software auf mehreren Geräten verwendet werden und trägt zur Verbesserung der Geschwindigkeit, Skalierbarkeit und Produktivität bei der Geräteherstellung bei.

Greenfield-Strategie

Das Fehlen vorhandener Infrastruktur in einer neuen Umgebung. Bei der Einführung einer Neuausrichtung einer Systemarchitektur können Sie alle neuen Technologien ohne Einschränkung der Kompatibilität mit der vorhandenen Infrastruktur auswählen, auch bekannt als Brownfield. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und Greenfield-Strategien mischen.

Integritätsschutz

Eine allgemeine Regel, die dazu beiträgt, Ressourcen, Richtlinien und die Einhaltung von Vorschriften in allen Unternehmenseinheiten zu regeln (OUs). Präventiver Integritätsschutz setzt Richtlinien durch, um die Einhaltung von Standards zu gewährleisten. Sie werden mithilfe von Service-Kontrollrichtlinien und IAM-Berechtigungsgrenzen implementiert. Detektivischer Integritätsschutz erkennt Richtlinienverstöße und Compliance-Probleme und generiert Warnmeldungen zur Abhilfe. Sie werden mithilfe von AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector und benutzerdefinierten AWS Lambda Prüfungen implementiert.

H

HEKTAR

Siehe Hochverfügbarkeit.

Heterogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank in eine Zieldatenbank, die eine andere Datenbank-Engine verwendet (z. B. Oracle zu Amazon Aurora). Eine heterogene Migration ist in der Regel Teil einer Neuarchitektur, und die Konvertierung des Schemas kann eine komplexe Aufgabe sein. AWS bietet AWS SCT, welches bei Schemakonvertierungen hilft.

hohe Verfügbarkeit (HA)

Die Fähigkeit eines Workloads, im Falle von Herausforderungen oder Katastrophen kontinuierlich und ohne Eingreifen zu arbeiten. HA-Systeme sind so konzipiert, dass sie automatisch ein Failover durchführen, gleichbleibend hohe Leistung bieten und unterschiedliche Lasten und Ausfälle mit minimalen Leistungseinbußen bewältigen.

historische Modernisierung

Ein Ansatz zur Modernisierung und Aufrüstung von Betriebstechnologiesystemen (OT), um den Bedürfnissen der Fertigungsindustrie besser gerecht zu werden. Ein Historian ist eine Art von Datenbank, die verwendet wird, um Daten aus verschiedenen Quellen in einer Fabrik zu sammeln und zu speichern.

Daten zurückhalten

Ein Teil historischer, beschrifteter Daten, der aus einem Datensatz zurückgehalten wird, der zum Trainieren eines Modells für maschinelles Lernen verwendet wird. Sie können Holdout-Daten verwenden, um die Modellleistung zu bewerten, indem Sie die Modellvorhersagen mit den Holdout-Daten vergleichen.

Homogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank zu einer Zieldatenbank, die dieselbe Datenbank-Engine verwendet (z. B. Microsoft SQL Server zu Amazon RDS für SQL Server). Eine homogene Migration ist in der Regel Teil eines Hostwechsels oder eines Plattformwechsels. Sie können native Datenbankserviceprogramme verwenden, um das Schema zu migrieren.

heiße Daten

Daten, auf die häufig zugegriffen wird, z. B. Echtzeitdaten oder aktuelle Translationsdaten. Für diese Daten ist in der Regel eine leistungsstarke Speicherebene oder -klasse erforderlich, um schnelle Abfrageantworten zu ermöglichen.

Hotfix

Eine dringende Lösung für ein kritisches Problem in einer Produktionsumgebung. Aufgrund seiner Dringlichkeit wird ein Hotfix normalerweise außerhalb des typischen DevOps Release-Workflows erstellt.

Hypercare-Phase

Unmittelbar nach dem Cutover, der Zeitraum, in dem ein Migrationsteam die migrierten Anwendungen in der Cloud verwaltet und überwacht, um etwaige Probleme zu beheben. In der Regel dauert dieser Zeitraum 1–4 Tage. Am Ende der Hypercare-Phase überträgt das Migrationsteam in der Regel die Verantwortung für die Anwendungen an das Cloud-Betriebsteam.

I

IaC

Sehen Sie Infrastruktur als Code.

Identitätsbasierte Richtlinie

Eine Richtlinie, die einem oder mehreren IAM-Prinzipalen zugeordnet ist und deren Berechtigungen innerhalb der AWS Cloud Umgebung definiert.

Leerlaufanwendung

Eine Anwendung mit einer durchschnittlichen CPU- und Arbeitsspeicherauslastung zwischen 5 und 20 Prozent über einen Zeitraum von 90 Tagen. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen oder sie On-Premises beizubehalten.

IIoT

Siehe Industrielles Internet der Dinge.

unveränderliche Infrastruktur

Ein Modell, das eine neue Infrastruktur für Produktionsworkloads bereitstellt, anstatt die bestehende Infrastruktur zu aktualisieren, zu patchen oder zu modifizieren. Unveränderliche Infrastrukturen sind von Natur aus konsistenter, zuverlässiger und vorhersehbarer als veränderliche Infrastrukturen. Weitere Informationen finden Sie in der Best Practice Deploy using immutable infrastructure im AWS Well-Architected Framework.

Eingehende (ingress) VPC

In einer Architektur AWS mit mehreren Konten ist dies eine VPC, die Netzwerkverbindungen von außerhalb einer Anwendung akzeptiert, überprüft und weiterleitet. Die AWS Security Reference Architecture empfiehlt, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr und Inspektion einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

Inkrementelle Migration

Eine Cutover-Strategie, bei der Sie Ihre Anwendung in kleinen Teilen migrieren, anstatt eine einziges vollständiges Cutover durchzuführen. Beispielsweise könnten Sie zunächst nur einige Microservices oder Benutzer auf das neue System umstellen. Nachdem Sie sich vergewissert haben, dass alles ordnungsgemäß funktioniert, können Sie weitere Microservices oder Benutzer

schrittweise verschieben, bis Sie Ihr Legacy-System außer Betrieb nehmen können. Diese Strategie reduziert die mit großen Migrationen verbundenen Risiken.

Industrie 4.0

Ein Begriff, der 2016 von Klaus Schwab eingeführt wurde und sich auf die Modernisierung von Fertigungsprozessen durch Fortschritte in den Bereichen Konnektivität, Echtzeitdaten, Automatisierung, Analytik und KI/ML bezieht.

Infrastruktur

Alle Ressourcen und Komponenten, die in der Umgebung einer Anwendung enthalten sind.

Infrastructure as Code (IaC)

Der Prozess der Bereitstellung und Verwaltung der Infrastruktur einer Anwendung mithilfe einer Reihe von Konfigurationsdateien. IaC soll Ihnen helfen, das Infrastrukturmanagement zu zentralisieren, Ressourcen zu standardisieren und schnell zu skalieren, sodass neue Umgebungen wiederholbar, zuverlässig und konsistent sind.

industrielles Internet der Dinge (T) IIo

Einsatz von mit dem Internet verbundenen Sensoren und Geräten in Industriesektoren wie Fertigung, Energie, Automobilindustrie, Gesundheitswesen, Biowissenschaften und Landwirtschaft. Weitere Informationen finden Sie unter Aufbau einer digitalen Transformationsstrategie für das industrielle Internet der Dinge (IIoT).

Inspektions-VPC

In einer Architektur AWS mit mehreren Konten eine zentralisierte VPC, die Inspektionen des Netzwerkverkehrs zwischen VPCs (in demselben oder unterschiedlichen AWS-Regionen), dem Internet und lokalen Netzwerken verwaltet. In der AWS Security Reference Architecture wird empfohlen, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr sowie Inspektionen einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

Internet of Things (IoT)

Das Netzwerk verbundener physischer Objekte mit eingebetteten Sensoren oder Prozessoren, das über das Internet oder über ein lokales Kommunikationsnetzwerk mit anderen Geräten und Systemen kommuniziert. Weitere Informationen finden Sie unter Was ist IoT?

Interpretierbarkeit

Ein Merkmal eines Modells für Machine Learning, das beschreibt, inwieweit ein Mensch verstehen kann, wie die Vorhersagen des Modells von seinen Eingaben abhängen. Weitere Informationen finden Sie unter Interpretierbarkeit des Modells für maschinelles Lernen mit. AWS

IoT

Siehe Internet der Dinge.

IT information library (ITIL, IT-Informationsbibliothek)

Eine Reihe von bewährten Methoden für die Bereitstellung von IT-Services und die Abstimmung dieser Services auf die Geschäftsanforderungen. ITIL bietet die Grundlage für ITSM.

T service management (ITSM, IT-Servicemanagement)

Aktivitäten im Zusammenhang mit der Gestaltung, Implementierung, Verwaltung und Unterstützung von IT-Services für eine Organisation. Informationen zur Integration von Cloud-Vorgängen mit ITSM-Tools finden Sie im Leitfaden zur Betriebsintegration.

BIS

Siehe IT-Informationsbibliothek.

ITSM

Siehe IT-Servicemanagement.

L

Labelbasierte Zugangskontrolle (LBAC)

Eine Implementierung der Mandatory Access Control (MAC), bei der den Benutzern und den Daten selbst jeweils explizit ein Sicherheitslabelwert zugewiesen wird. Die Schnittmenge zwischen der Benutzersicherheitsbeschriftung und der Datensicherheitsbeschriftung bestimmt, welche Zeilen und Spalten für den Benutzer sichtbar sind.

Landing Zone

Eine landing zone ist eine gut strukturierte AWS Umgebung mit mehreren Konten, die skalierbar und sicher ist. Dies ist ein Ausgangspunkt, von dem aus Ihre Organisationen Workloads und Anwendungen schnell und mit Vertrauen in ihre Sicherheits- und Infrastrukturumgebung starten

und bereitstellen können. Weitere Informationen zu Landing Zones finden Sie unter Einrichtung einer sicheren und skalierbaren AWS -Umgebung mit mehreren Konten..

großes Sprachmodell (LLM)

Ein Deep-Learning-KI-Modell, das anhand einer riesigen Datenmenge vorab trainiert wurde. Ein LLM kann mehrere Aufgaben ausführen, z. B. Fragen beantworten, Dokumente zusammenfassen, Text in andere Sprachen übersetzen und Sätze vervollständigen. Weitere Informationen finden Sie unter Was sind. LLMs

Große Migration

Eine Migration von 300 oder mehr Servern.

SCHWARZ

Siehe Labelbasierte Zugriffskontrolle.

Geringste Berechtigung

Die bewährte Sicherheitsmethode, bei der nur die für die Durchführung einer Aufgabe erforderlichen Mindestberechtigungen erteilt werden. Weitere Informationen finden Sie unter Geringste Berechtigungen anwenden in der IAM-Dokumentation.

Lift and Shift

Siehe 7 Rs.

Little-Endian-System

Ein System, welches das niedrigwertigste Byte zuerst speichert. Siehe auch Endianness.

LLM

Siehe großes Sprachmodell.

Niedrigere Umgebungen

Siehe Umgebung.

M

Machine Learning (ML)

Eine Art künstlicher Intelligenz, die Algorithmen und Techniken zur Mustererkennung und zum Lernen verwendet. ML analysiert aufgezeichnete Daten, wie z. B. Daten aus dem Internet der

Dinge (IoT), und lernt daraus, um ein statistisches Modell auf der Grundlage von Mustern zu erstellen. Weitere Informationen finden Sie unter Machine Learning.

Hauptzweig

Siehe Filiale.

Malware

Software, die entwickelt wurde, um die Computersicherheit oder den Datenschutz zu gefährden. Malware kann Computersysteme stören, vertrauliche Informationen durchsickern lassen oder sich unbefugten Zugriff verschaffen. Beispiele für Malware sind Viren, Würmer, Ransomware, Trojaner, Spyware und Keylogger.

verwaltete Dienste

AWS-Services für die die Infrastrukturebene, das Betriebssystem und die Plattformen AWS betrieben werden, und Sie greifen auf die Endgeräte zu, um Daten zu speichern und abzurufen. Amazon Simple Storage Service (Amazon S3) und Amazon DynamoDB sind Beispiele für Managed Services. Diese werden auch als abstrakte Dienste bezeichnet.

Manufacturing Execution System (MES)

Ein Softwaresystem zur Verfolgung, Überwachung, Dokumentation und Steuerung von Produktionsprozessen, bei denen Rohstoffe in der Fertigung zu fertigen Produkten umgewandelt werden.

MAP

Siehe Migration Acceleration Program.

Mechanismus

Ein vollständiger Prozess, bei dem Sie ein Tool erstellen, die Akzeptanz des Tools vorantreiben und anschließend die Ergebnisse überprüfen, um Anpassungen vorzunehmen. Ein Mechanismus ist ein Zyklus, der sich im Laufe seiner Tätigkeit selbst verstärkt und verbessert. Weitere Informationen finden Sie unter Aufbau von Mechanismen im AWS Well-Architected Framework.

Mitgliedskonto

Alle AWS-Konten außer dem Verwaltungskonto, die Teil einer Organisation in sind. AWS Organizations Ein Konto kann jeweils nur Mitglied einer Organisation sein.

MES

Siehe Manufacturing Execution System.

Message Queuing-Telemetrietransport (MQTT)

Ein leichtes machine-to-machine (M2M) -Kommunikationsprotokoll, das auf dem Publish/Subscribe-Muster für IoT-Geräte mit beschränkten Ressourcen basiert.

Microservice

Ein kleiner, unabhängiger Dienst, der über genau definierte Kanäle kommuniziert APIs und in der Regel kleinen, eigenständigen Teams gehört. Ein Versicherungssystem kann beispielsweise Microservices beinhalten, die Geschäftsfunktionen wie Vertrieb oder Marketing oder Subdomains wie Einkauf, Schadenersatz oder Analytik zugeordnet sind. Zu den Vorteilen von Microservices gehören Agilität, flexible Skalierung, einfache Bereitstellung, wiederverwendbarer Code und Ausfallsicherheit. Weitere Informationen finden Sie unter Integration von Microservices mithilfe serverloser Dienste. AWS

Microservices-Architekturen

Ein Ansatz zur Erstellung einer Anwendung mit unabhängigen Komponenten, die jeden Anwendungsprozess als Microservice ausführen. Diese Microservices kommunizieren mithilfe von Lightweight über eine klar definierte Schnittstelle. APIs Jeder Microservice in dieser Architektur kann aktualisiert, bereitgestellt und skaliert werden, um den Bedarf an bestimmten Funktionen einer Anwendung zu decken. Weitere Informationen finden Sie unter Implementierung von Microservices auf. AWS

Migration Acceleration Program (MAP)

Ein AWS Programm, das Beratung, Unterstützung, Schulungen und Services bietet, um Unternehmen dabei zu unterstützen, eine solide betriebliche Grundlage für die Umstellung auf die Cloud zu schaffen und die anfänglichen Kosten von Migrationen auszugleichen. MAP umfasst eine Migrationsmethode für die methodische Durchführung von Legacy-Migrationen sowie eine Reihe von Tools zur Automatisierung und Beschleunigung gängiger Migrationsszenarien.

Migration in großem Maßstab

Der Prozess, bei dem der Großteil des Anwendungsportfolios in Wellen in die Cloud verlagert wird, wobei in jeder Welle mehr Anwendungen schneller migriert werden. In dieser Phase werden die bewährten Verfahren und Erkenntnisse aus den früheren Phasen zur Implementierung einer Migrationsfabrik von Teams, Tools und Prozessen zur Optimierung der Migration von Workloads durch Automatisierung und agile Bereitstellung verwendet. Dies ist die dritte Phase der AWS - Migrationsstrategie.

Migrationsfabrik

Funktionsübergreifende Teams, die die Migration von Workloads durch automatisierte, agile Ansätze optimieren. Zu den Teams in der Migrationsabteilung gehören in der Regel Betriebsabläufe, Geschäftsanalysten und Eigentümer, Migrationsingenieure, Entwickler und DevOps Experten, die in Sprints arbeiten. Zwischen 20 und 50 Prozent eines Unternehmensanwendungsportfolios bestehen aus sich wiederholenden Mustern, die durch einen Fabrik-Ansatz optimiert werden können. Weitere Informationen finden Sie in Diskussion über Migrationsfabriken und den Leitfaden zur Cloud-Migration-Fabrik in diesem Inhaltssatz.

Migrationsmetadaten

Die Informationen über die Anwendung und den Server, die für den Abschluss der Migration benötigt werden. Für jedes Migrationsmuster ist ein anderer Satz von Migrationsmetadaten erforderlich. Beispiele für Migrationsmetadaten sind das Zielsubnetz, die Sicherheitsgruppe und AWS das Konto.

Migrationsmuster

Eine wiederholbare Migrationsaufgabe, in der die Migrationsstrategie, das Migrationsziel und die verwendete Migrationsanwendung oder der verwendete Migrationsservice detailliert beschrieben werden. Beispiel: Rehost-Migration zu Amazon EC2 mit AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

Ein Online-Tool, das Informationen zur Validierung des Geschäftsszenarios für die Migration auf das bereitstellt. AWS Cloud MPA bietet eine detaillierte Portfoliobewertung (richtige Servergröße, Preisgestaltung, Gesamtbetriebskostenanalyse, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung, Migrationspriorisierung und Wellenplanung). Das MPA-Tool (Anmeldung erforderlich) steht allen AWS Beratern und APN-Partnerberatern kostenlos zur Verfügung.

Migration Readiness Assessment (MRA)

Der Prozess, bei dem mithilfe des AWS CAF Erkenntnisse über den Cloud-Bereitschaftsstatus eines Unternehmens gewonnen, Stärken und Schwächen identifiziert und ein Aktionsplan zur Schließung festgestellter Lücken erstellt wird. Weitere Informationen finden Sie im Benutzerhandbuch für Migration Readiness. MRA ist die erste Phase der AWS - Migrationsstrategie.

Migrationsstrategie

Der Ansatz, der verwendet wurde, um einen Workload auf den AWS Cloud zu migrieren. Weitere Informationen finden Sie im Eintrag 7 Rs in diesem Glossar und unter Mobilisieren Sie Ihr Unternehmen, um umfangreiche Migrationen zu beschleunigen.

ML

Siehe maschinelles Lernen.

Modernisierung

Umwandlung einer veralteten (veralteten oder monolithischen) Anwendung und ihrer Infrastruktur in ein agiles, elastisches und hochverfügbares System in der Cloud, um Kosten zu senken, die Effizienz zu steigern und Innovationen zu nutzen. Weitere Informationen finden Sie unter Strategie zur Modernisierung von Anwendungen in der AWS Cloud.

Bewertung der Modernisierungsfähigkeit

Eine Bewertung, anhand derer festgestellt werden kann, ob die Anwendungen einer Organisation für die Modernisierung bereit sind, Vorteile, Risiken und Abhängigkeiten identifiziert und ermittelt wird, wie gut die Organisation den zukünftigen Status dieser Anwendungen unterstützen kann. Das Ergebnis der Bewertung ist eine Vorlage der Zielarchitektur, eine Roadmap, in der die Entwicklungsphasen und Meilensteine des Modernisierungsprozesses detailliert beschrieben werden, sowie ein Aktionsplan zur Behebung festgestellter Lücken. Weitere Informationen finden Sie unter Evaluierung der Modernisierungsbereitschaft von Anwendungen in der AWS Cloud.

Monolithische Anwendungen (Monolithen)

Anwendungen, die als ein einziger Service mit eng gekoppelten Prozessen ausgeführt werden. Monolithische Anwendungen haben verschiedene Nachteile. Wenn ein Anwendungs-Feature stark nachgefragt wird, muss die gesamte Architektur skaliert werden. Das Hinzufügen oder Verbessern der Feature einer monolithischen Anwendung wird ebenfalls komplexer, wenn die Codebasis wächst. Um diese Probleme zu beheben, können Sie eine Microservices-Architektur verwenden. Weitere Informationen finden Sie unter Zerlegen von Monolithen in Microservices.

MPA

Siehe Bewertung des Migrationsportfolios.

MQTT

Siehe Message Queuing-Telemetrietransport.

Mehrklassen-Klassifizierung

Ein Prozess, der dabei hilft, Vorhersagen für mehrere Klassen zu generieren (wobei eines von mehr als zwei Ergebnissen vorhergesagt wird). Ein ML-Modell könnte beispielsweise fragen: „Ist dieses Produkt ein Buch, ein Auto oder ein Telefon?“ oder „Welche Kategorie von Produkten ist für diesen Kunden am interessantesten?“

veränderbare Infrastruktur

Ein Modell, das die bestehende Infrastruktur für Produktionsworkloads aktualisiert und modifiziert. Für eine verbesserte Konsistenz, Zuverlässigkeit und Vorhersagbarkeit empfiehlt das AWS Well-Architected Framework die Verwendung einer unveränderlichen Infrastruktur als bewährte Methode.

O

OAC

Siehe Origin Access Control.

EICHE

Siehe Zugriffsidentität von Origin.

COM

Siehe organisatorisches Change-Management.

Offline-Migration

Eine Migrationsmethode, bei der der Quell-Workload während des Migrationsprozesses heruntergefahren wird. Diese Methode ist mit längeren Ausfallzeiten verbunden und wird in der Regel für kleine, unkritische Workloads verwendet.

OI

Siehe Betriebsintegration.

OLA

Siehe Vereinbarung auf operativer Ebene.

Online-Migration

Eine Migrationsmethode, bei der der Quell-Workload auf das Zielsystem kopiert wird, ohne offline genommen zu werden. Anwendungen, die mit dem Workload verbunden sind, können während

der Migration weiterhin funktionieren. Diese Methode beinhaltet keine bis minimale Ausfallzeit und wird in der Regel für kritische Produktionsworkloads verwendet.

OPC-UA

Siehe Open Process Communications — Unified Architecture.

Offene Prozesskommunikation — Einheitliche Architektur (OPC-UA)

Ein machine-to-machine (M2M) -Kommunikationsprotokoll für die industrielle Automatisierung. OPC-UA bietet einen Interoperabilitätsstandard mit Datenverschlüsselungs-, Authentifizierungs- und Autorisierungsschemata.

Vereinbarung auf Betriebsebene (OLA)

Eine Vereinbarung, in der klargestellt wird, welche funktionalen IT-Gruppen sich gegenseitig versprechen zu liefern, um ein Service Level Agreement (SLA) zu unterstützen.

Überprüfung der Betriebsbereitschaft (ORR)

Eine Checkliste mit Fragen und zugehörigen bewährten Methoden, die Ihnen helfen, Vorfälle und mögliche Ausfälle zu verstehen, zu bewerten, zu verhindern oder deren Umfang zu reduzieren. Weitere Informationen finden Sie unter Operational Readiness Reviews (ORR) im AWS Well-Architected Framework.

Betriebstechnologie (OT)

Hardware- und Softwaresysteme, die mit der physischen Umgebung zusammenarbeiten, um industrielle Abläufe, Ausrüstung und Infrastruktur zu steuern. In der Fertigung ist die Integration von OT- und Informationstechnologie (IT) -Systemen ein zentraler Schwerpunkt der Industrie 4.0-Transformationen.

Betriebsintegration (OI)

Der Prozess der Modernisierung von Abläufen in der Cloud, der Bereitschaftsplanung, Automatisierung und Integration umfasst. Weitere Informationen finden Sie im Leitfaden zur Betriebsintegration.

Organisationspfad

Ein Pfad, der von erstellt wird und in AWS CloudTrail dem alle Ereignisse für alle AWS-Konten in einer Organisation protokolliert werden. AWS Organizations Diese Spur wird in jedem AWS-Konto , der Teil der Organisation ist, erstellt und verfolgt die Aktivität in jedem Konto. Weitere Informationen finden Sie in der CloudTrail Dokumentation unter Einen Trail für eine Organisation erstellen.

Organisatorisches Veränderungsmanagement (OCM)

Ein Framework für das Management wichtiger, disruptiver Geschäftstransformationen aus Sicht der Mitarbeiter, der Kultur und der Führung. OCM hilft Organisationen dabei, sich auf neue Systeme und Strategien vorzubereiten und auf diese umzustellen, indem es die Akzeptanz von Veränderungen beschleunigt, Übergangsprobleme angeht und kulturelle und organisatorische Veränderungen vorantreibt. In der AWS Migrationsstrategie wird dieses Framework aufgrund der Geschwindigkeit des Wandels, der bei Projekten zur Cloud-Einführung erforderlich ist, als Mitarbeiterbeschleunigung bezeichnet. Weitere Informationen finden Sie im OCM-Handbuch.

Ursprungszugriffskontrolle (OAC)

In CloudFront, eine erweiterte Option zur Zugriffsbeschränkung, um Ihre Amazon Simple Storage Service (Amazon S3) -Inhalte zu sichern. OAC unterstützt alle S3-Buckets insgesamt AWS-Regionen, serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) sowie dynamische PUT und `DELETE` Anfragen an den S3-Bucket.

Ursprungszugriffsidentität (OAI)

In CloudFront, eine Option zur Zugriffsbeschränkung, um Ihre Amazon S3 S3-Inhalte zu sichern. Wenn Sie OAI verwenden, CloudFront erstellt es einen Principal, mit dem sich Amazon S3 authentifizieren kann. Authentifizierte Principals können nur über eine bestimmte Distribution auf Inhalte in einem S3-Bucket zugreifen. CloudFront Siehe auch OAC, das eine detailliertere und verbesserte Zugriffskontrolle bietet.

ORR

Weitere Informationen finden Sie unter Überprüfung der Betriebsbereitschaft.

NICHT

Siehe Betriebstechnologie.

Ausgehende (egress) VPC

In einer Architektur AWS mit mehreren Konten eine VPC, die Netzwerkverbindungen verarbeitet, die von einer Anwendung aus initiiert werden. Die AWS Security Reference Architecture empfiehlt die Einrichtung Ihres Netzwerkkontos mit eingehendem und ausgehendem Datenverkehr sowie Inspektion, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

P

Berechtigungsgrenze

Eine IAM-Verwaltungsrichtlinie, die den IAM-Prinzipalen zugeordnet ist, um die maximalen Berechtigungen festzulegen, die der Benutzer oder die Rolle haben kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitys in der IAM-Dokumentation.

persönlich identifizierbare Informationen (PII)

Informationen, die, wenn sie direkt betrachtet oder mit anderen verwandten Daten kombiniert werden, verwendet werden können, um vernünftige Rückschlüsse auf die Identität einer Person zu ziehen. Beispiele für personenbezogene Daten sind Namen, Adressen und Kontaktinformationen.

Personenbezogene Daten

Siehe persönlich identifizierbare Informationen.

Playbook

Eine Reihe vordefinierter Schritte, die die mit Migrationen verbundenen Aufgaben erfassen, z. B. die Bereitstellung zentraler Betriebsfunktionen in der Cloud. Ein Playbook kann die Form von Skripten, automatisierten Runbooks oder einer Zusammenfassung der Prozesse oder Schritte annehmen, die für den Betrieb Ihrer modernisierten Umgebung erforderlich sind.

PLC

Siehe programmierbare Logiksteuerung.

PLM

Siehe Produktlebenszyklusmanagement.

policy

Ein Objekt, das Berechtigungen definieren (siehe identitätsbasierte Richtlinie), Zugriffsbedingungen spezifizieren (siehe ressourcenbasierte Richtlinie) oder die maximalen Berechtigungen für alle Konten in einer Organisation definieren kann AWS Organizations (siehe Dienststeuerungsrichtlinie).

Polyglotte Beharrlichkeit

Unabhängige Auswahl der Datenspeichertechnologie eines Microservices auf der Grundlage von Datenzugriffsmustern und anderen Anforderungen. Wenn Ihre Microservices über dieselbe Datenspeichertechnologie verfügen, kann dies zu Implementierungsproblemen oder zu

Leistungseinbußen führen. Microservices lassen sich leichter implementieren und erzielen eine bessere Leistung und Skalierbarkeit, wenn sie den Datenspeicher verwenden, der ihren Anforderungen am besten entspricht.

Portfoliobewertung

Ein Prozess, bei dem das Anwendungsportfolio ermittelt, analysiert und priorisiert wird, um die Migration zu planen. Weitere Informationen finden Sie in Bewerten der Migrationsbereitschaft.

predicate

Eine Abfragebedingung, die `true` oder zurückgibt`false`, was üblicherweise in einer Klausel vorkommt. `WHERE`

Prädikat Pushdown

Eine Technik zur Optimierung von Datenbankabfragen, bei der die Daten in der Abfrage vor der Übertragung gefiltert werden. Dadurch wird die Datenmenge reduziert, die aus der relationalen Datenbank abgerufen und verarbeitet werden muss, und die Abfrageleistung wird verbessert.

Präventive Kontrolle

Eine Sicherheitskontrolle, die verhindern soll, dass ein Ereignis eintritt. Diese Kontrollen stellen eine erste Verteidigungslinie dar, um unbefugten Zugriff oder unerwünschte Änderungen an Ihrem Netzwerk zu verhindern. Weitere Informationen finden Sie unter Präventive Kontrolle in Implementierung von Sicherheitskontrollen in AWS.

Prinzipal

Eine Entität AWS , die Aktionen ausführen und auf Ressourcen zugreifen kann. Diese Entität ist in der Regel ein Root-Benutzer für eine AWS-Konto, eine IAM-Rolle oder einen Benutzer. Weitere Informationen finden Sie unter Prinzipal in Rollenbegriffe und -konzepte in der IAM-Dokumentation.

Datenschutz von Natur aus

Ein systemtechnischer Ansatz, der den Datenschutz während des gesamten Entwicklungsprozesses berücksichtigt.

Privat gehostete Zonen

Ein Container, der Informationen darüber enthält, wie Amazon Route 53 auf DNS-Abfragen für eine Domain und deren Subdomains innerhalb einer oder mehrerer VPCs Domains antworten soll. Weitere Informationen finden Sie unter Arbeiten mit privat gehosteten Zonen in der Route-53-Dokumentation.

proaktive Steuerung

Eine Sicherheitskontrolle, die den Einsatz nicht richtlinienkonformer Ressourcen verhindern soll. Diese Steuerelemente scannen Ressourcen, bevor sie bereitgestellt werden. Wenn die Ressource nicht der Kontrolle entspricht, wird sie nicht bereitgestellt. Weitere Informationen finden Sie im Referenzhandbuch zu Kontrollen in der AWS Control Tower Dokumentation und unter Proaktive Kontrollen unter Implementierung von Sicherheitskontrollen am AWS.

Produktlebenszyklusmanagement (PLM)

Das Management von Daten und Prozessen für ein Produkt während seines gesamten Lebenszyklus, vom Design, der Entwicklung und Markteinführung über Wachstum und Reife bis hin zur Markteinführung und Markteinführung.

Produktionsumgebung

Siehe Umgebung.

Speicherprogrammierbare Steuerung (SPS)

In der Fertigung ein äußerst zuverlässiger, anpassungsfähiger Computer, der Maschinen überwacht und Fertigungsprozesse automatisiert.

schnelle Verkettung

Verwendung der Ausgabe einer LLM-Eingabeaufforderung als Eingabe für die nächste Aufforderung, um bessere Antworten zu generieren. Diese Technik wird verwendet, um eine komplexe Aufgabe in Unteraufgaben zu unterteilen oder um eine vorläufige Antwort iterativ zu verfeinern oder zu erweitern. Sie trägt dazu bei, die Genauigkeit und Relevanz der Antworten eines Modells zu verbessern und ermöglicht detailliertere, personalisierte Ergebnisse.

Pseudonymisierung

Der Prozess, bei dem persönliche Identifikatoren in einem Datensatz durch Platzhalterwerte ersetzt werden. Pseudonymisierung kann zum Schutz der Privatsphäre beitragen. Pseudonymisierte Daten gelten weiterhin als personenbezogene Daten.

publish/subscribe (pub/sub)

Ein Muster, das asynchrone Kommunikation zwischen Microservices ermöglicht, um die Skalierbarkeit und Reaktionsfähigkeit zu verbessern. In einem auf Microservices basierenden MES kann ein Microservice beispielsweise Ereignismeldungen in einem Kanal veröffentlichen, den andere Microservices abonnieren können. Das System kann neue Microservices hinzufügen, ohne den Veröffentlichungsservice zu ändern.

Q

Abfrageplan

Eine Reihe von Schritten, wie Anweisungen, die für den Zugriff auf die Daten in einem relationalen SQL-Datenbanksystem verwendet werden.

Abfrageplanregression

Wenn ein Datenbankserviceoptimierer einen weniger optimalen Plan wählt als vor einer bestimmten Änderung der Datenbankumgebung. Dies kann durch Änderungen an Statistiken, Beschränkungen, Umgebungseinstellungen, Abfrageparameter-Bindungen und Aktualisierungen der Datenbank-Engine verursacht werden.

R

RACI-Matrix

Siehe verantwortlich, rechenschaftspflichtig, konsultiert, informiert (RACI).

RAG

Siehe Erweiterte Generierung beim Abrufen.

Ransomware

Eine bösartige Software, die entwickelt wurde, um den Zugriff auf ein Computersystem oder Daten zu blockieren, bis eine Zahlung erfolgt ist.

RASCI-Matrix

Siehe verantwortlich, rechenschaftspflichtig, konsultiert, informiert (RACI).

RCAC

Siehe Zugriffskontrolle für Zeilen und Spalten.

Read Replica

Eine Kopie einer Datenbank, die nur für Lesezwecke verwendet wird. Sie können Abfragen an das Lesereplikat weiterleiten, um die Belastung auf Ihrer Primärdatenbank zu reduzieren.

neu strukturieren

Siehe 7 Rs.

Recovery Point Objective (RPO)

Die maximal zulässige Zeitspanne seit dem letzten Datenwiederherstellungspunkt. Damit wird festgelegt, was als akzeptabler Datenverlust zwischen dem letzten Wiederherstellungspunkt und der Serviceunterbrechung gilt.

Wiederherstellungszeitziel (RTO)

Die maximal zulässige Verzögerung zwischen der Betriebsunterbrechung und der Wiederherstellung des Dienstes.

Refaktorierung

Siehe 7 Rs.

Region

Eine Sammlung von AWS Ressourcen in einem geografischen Gebiet. Jeder AWS-Region ist isoliert und unabhängig von den anderen, um Fehlertoleranz, Stabilität und Belastbarkeit zu gewährleisten. Weitere Informationen finden Sie unter Geben Sie an, was AWS-Regionen Ihr Konto verwenden kann.

Regression

Eine ML-Technik, die einen numerischen Wert vorhersagt. Zum Beispiel, um das Problem „Zu welchem Preis wird dieses Haus verkauft werden?“ zu lösen Ein ML-Modell könnte ein lineares Regressionsmodell verwenden, um den Verkaufspreis eines Hauses auf der Grundlage bekannter Fakten über das Haus (z. B. die Quadratmeterzahl) vorherzusagen.

rehosten

Siehe 7 Rs.

Veröffentlichung

In einem Bereitstellungsprozess der Akt der Förderung von Änderungen an einer Produktionsumgebung.

umziehen

Siehe 7 Rs.

neue Plattform

Siehe 7 Rs.

Rückkauf

Siehe 7 Rs.

Ausfallsicherheit

Die Fähigkeit einer Anwendung, Störungen zu widerstehen oder sich von ihnen zu erholen. Hochverfügbarkeit und Notfallwiederherstellung sind häufig Überlegungen bei der Planung der Ausfallsicherheit in der. AWS Cloud Weitere Informationen finden Sie unter AWS Cloud Resilienz.

Ressourcenbasierte Richtlinie

Eine mit einer Ressource verknüpfte Richtlinie, z. B. ein Amazon-S3-Bucket, ein Endpunkt oder ein Verschlüsselungsschlüssel. Diese Art von Richtlinie legt fest, welchen Prinzipalen der Zugriff gewährt wird, welche Aktionen unterstützt werden und welche anderen Bedingungen erfüllt sein müssen.

RACI-Matrix (verantwortlich, rechenschaftspflichtig, konsultiert, informiert)

Eine Matrix, die die Rollen und Verantwortlichkeiten aller an Migrationsaktivitäten und Cloud-Operationen beteiligten Parteien definiert. Der Matrixname leitet sich von den in der Matrix definierten Zuständigkeitstypen ab: verantwortlich (R), rechenschaftspflichtig (A), konsultiert (C) und informiert (I). Der Unterstützungstyp (S) ist optional. Wenn Sie Unterstützung einbeziehen, wird die Matrix als RASCI-Matrix bezeichnet, und wenn Sie sie ausschließen, wird sie als RACI-Matrix bezeichnet.

Reaktive Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, die Behebung unerwünschter Ereignisse oder Abweichungen von Ihren Sicherheitsstandards voranzutreiben. Weitere Informationen finden Sie unter Reaktive Kontrolle in Implementieren von Sicherheitskontrollen in AWS.

Beibehaltung

Siehe 7 Rs.

zurückziehen

Siehe 7 Rs.

Retrieval Augmented Generation (RAG)

Eine generative KI-Technologie, bei der ein LLM auf eine maßgebliche Datenquelle verweist, die sich außerhalb seiner Trainingsdatenquellen befindet, bevor eine Antwort generiert wird. Ein RAG-Modell könnte beispielsweise eine semantische Suche in der Wissensdatenbank oder in

benutzerdefinierten Daten einer Organisation durchführen. Weitere Informationen finden Sie unter Was ist RAG.

Drehung

Der Vorgang, bei dem ein Geheimnis regelmäßig aktualisiert wird, um es einem Angreifer zu erschweren, auf die Anmeldeinformationen zuzugreifen.

Zugriffskontrolle für Zeilen und Spalten (RCAC)

Die Verwendung einfacher, flexibler SQL-Ausdrücke mit definierten Zugriffsregeln. RCAC besteht aus Zeilenberechtigungen und Spaltenmasken.

RPO

Siehe Recovery Point Objective.

RTO

Siehe Ziel für die Erholungszeit.

Runbook

Eine Reihe manueller oder automatisierter Verfahren, die zur Ausführung einer bestimmten Aufgabe erforderlich sind. Diese sind in der Regel darauf ausgelegt, sich wiederholende Operationen oder Verfahren mit hohen Fehlerquoten zu rationalisieren.

S

SAML 2.0

Ein offener Standard, den viele Identitätsanbieter (IdPs) verwenden. Diese Funktion ermöglicht föderiertes Single Sign-On (SSO), sodass sich Benutzer bei den API-Vorgängen anmelden AWS-Managementkonsole oder die AWS API-Operationen aufrufen können, ohne dass Sie einen Benutzer in IAM für alle in Ihrer Organisation erstellen müssen. Weitere Informationen zum SAML-2.0.-basierten Verbund finden Sie unter Über den SAML-2.0-basierten Verbund in der IAM-Dokumentation.

SCADA

Siehe Aufsichtskontrolle und Datenerfassung.

SCP

Siehe Richtlinie zur Dienstkontrolle.

Secret

Interne AWS Secrets Manager, vertrauliche oder eingeschränkte Informationen, wie z. B. ein Passwort oder Benutzeranmeldeinformationen, die Sie in verschlüsselter Form speichern. Es besteht aus dem geheimen Wert und seinen Metadaten. Der geheime Wert kann binär, eine einzelne Zeichenfolge oder mehrere Zeichenketten sein. Weitere Informationen finden Sie unter Was ist in einem Secrets Manager Manager-Geheimnis? in der Secrets Manager Manager-Dokumentation.

Sicherheit durch Design

Ein systemtechnischer Ansatz, der die Sicherheit während des gesamten Entwicklungsprozesses berücksichtigt.

Sicherheitskontrolle

Ein technischer oder administrativer Integritätsschutz, der die Fähigkeit eines Bedrohungsakteurs, eine Schwachstelle auszunutzen, verhindert, erkennt oder einschränkt. Es gibt vier Haupttypen von Sicherheitskontrollen: präventiv, detektiv, reaktionsschnell und proaktiv.

Härtung der Sicherheit

Der Prozess, bei dem die Angriffsfläche reduziert wird, um sie widerstandsfähiger gegen Angriffe zu machen. Dies kann Aktionen wie das Entfernen von Ressourcen, die nicht mehr benötigt werden, die Implementierung der bewährten Sicherheitsmethode der Gewährung geringster Berechtigungen oder die Deaktivierung unnötiger Feature in Konfigurationsdateien umfassen.

System zur Verwaltung von Sicherheitsinformationen und Ereignissen (security information and event management – SIEM)

Tools und Services, die Systeme für das Sicherheitsinformationsmanagement (SIM) und das Management von Sicherheitsereignissen (SEM) kombinieren. Ein SIEM-System sammelt, überwacht und analysiert Daten von Servern, Netzwerken, Geräten und anderen Quellen, um Bedrohungen und Sicherheitsverletzungen zu erkennen und Warnmeldungen zu generieren.

Automatisierung von Sicherheitsreaktionen

Eine vordefinierte und programmierte Aktion, die darauf ausgelegt ist, automatisch auf ein Sicherheitsereignis zu reagieren oder es zu beheben. Diese Automatisierungen dienen als detektive oder reaktionsschnelle Sicherheitskontrollen, die Sie bei der Implementierung bewährter AWS Sicherheitsmethoden unterstützen. Beispiele für automatisierte Antwortaktionen sind das Ändern einer VPC-Sicherheitsgruppe, das Patchen einer Amazon EC2 EC2-Instance oder das Rotieren von Anmeldeinformationen.

Serverseitige Verschlüsselung

Verschlüsselung von Daten am Zielort durch denjenigen AWS-Service , der sie empfängt.

Service-Kontrollrichtlinie (SCP)

Eine Richtlinie, die eine zentrale Steuerung der Berechtigungen für alle Konten in einer Organisation in ermöglicht AWS Organizations. SCPs Definieren Sie Leitplanken oder legen Sie Grenzwerte für Aktionen fest, die ein Administrator an Benutzer oder Rollen delegieren kann. Sie können sie SCPs als Zulassungs- oder Ablehnungslisten verwenden, um festzulegen, welche Dienste oder Aktionen zulässig oder verboten sind. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter Richtlinien zur Dienststeuerung.

Service-Endpunkt

Die URL des Einstiegspunkts für einen AWS-Service. Sie können den Endpunkt verwenden, um programmgesteuert eine Verbindung zum Zielservice herzustellen. Weitere Informationen finden Sie unter AWS-Service -Endpunkte in der Allgemeine AWS-Referenz.

Service Level Agreement (SLA)

Eine Vereinbarung, in der klargestellt wird, was ein IT-Team seinen Kunden zu bieten verspricht, z. B. in Bezug auf Verfügbarkeit und Leistung der Services.

Service-Level-Indikator (SLI)

Eine Messung eines Leistungsaspekts eines Dienstes, z. B. seiner Fehlerrate, Verfügbarkeit oder Durchsatz.

Service-Level-Ziel (SLO)

Eine Zielkennzahl, die den Zustand eines Dienstes darstellt, gemessen anhand eines Service-Level-Indikators.

Modell der geteilten Verantwortung

Ein Modell, das die Verantwortung beschreibt, mit der Sie gemeinsam AWS für Cloud-Sicherheit und Compliance verantwortlich sind. AWS ist für die Sicherheit der Cloud verantwortlich, während Sie für die Sicherheit in der Cloud verantwortlich sind. Weitere Informationen finden Sie unter Modell der geteilten Verantwortung.

SIEM

Siehe Sicherheitsinformations- und Event-Management-System.

Single Point of Failure (SPOF)

Ein Fehler in einer einzelnen, kritischen Komponente einer Anwendung, der das System stören kann.

SLA

Siehe Service Level Agreement.

SLI

Siehe Service-Level-Indikator.

ALSO

Siehe Service-Level-Ziel.

split-and-seed Modell

Ein Muster für die Skalierung und Beschleunigung von Modernisierungsprojekten. Sobald neue Features und Produktversionen definiert werden, teilt sich das Kernteam auf, um neue Produktteams zu bilden. Dies trägt zur Skalierung der Fähigkeiten und Services Ihrer Organisation bei, verbessert die Produktivität der Entwickler und unterstützt schnelle Innovationen. Weitere Informationen finden Sie unter Schrittweiser Ansatz zur Modernisierung von Anwendungen in der AWS Cloud.

SPOTTEN

Siehe Single Point of Failure.

Sternschema

Eine Datenbank-Organisationsstruktur, die eine große Faktentabelle zum Speichern von Transaktions- oder Messdaten und eine oder mehrere kleinere dimensionale Tabellen zum Speichern von Datenattributen verwendet. Diese Struktur ist für die Verwendung in einem Data Warehouse oder für Business Intelligence-Zwecke konzipiert.

Strangler-Fig-Muster

Ein Ansatz zur Modernisierung monolithischer Systeme, bei dem die Systemfunktionen schrittweise umgeschrieben und ersetzt werden, bis das Legacy-System außer Betrieb genommen werden kann. Dieses Muster verwendet die Analogie einer Feigenrebe, die zu einem etablierten Baum heranwächst und schließlich ihren Wirt überwindet und ersetzt. Das Muster wurde eingeführt von Martin Fowler als Möglichkeit, Risiken beim Umschreiben monolithischer Systeme zu managen. Ein Beispiel für die Anwendung dieses Musters finden Sie

unter Schrittweises Modernisieren älterer Microsoft ASP.NET (ASMX)-Webservices mithilfe von Containern und Amazon API Gateway.

Subnetz

Ein Bereich von IP-Adressen in Ihrer VPC. Ein Subnetz muss sich in einer einzigen Availability Zone befinden.

Aufsichtskontrolle und Datenerfassung (SCADA)

In der Fertigung ein System, das Hardware und Software zur Überwachung von Sachanlagen und Produktionsabläufen verwendet.

Symmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der denselben Schlüssel zum Verschlüsseln und Entschlüsseln der Daten verwendet.

synthetisches Testen

Testen eines Systems auf eine Weise, die Benutzerinteraktionen simuliert, um potenzielle Probleme zu erkennen oder die Leistung zu überwachen. Sie können Amazon CloudWatch Synthetics verwenden, um diese Tests zu erstellen.

Systemaufforderung

Eine Technik, mit der einem LLM Kontext, Anweisungen oder Richtlinien zur Verfügung gestellt werden, um sein Verhalten zu steuern. Systemaufforderungen helfen dabei, den Kontext festzulegen und Regeln für Interaktionen mit Benutzern festzulegen.

T

tags

Schlüssel-Wert-Paare, die als Metadaten für die Organisation Ihrer Ressourcen dienen. AWS Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern. Weitere Informationen finden Sie unter Markieren Ihrer AWS -Ressourcen.

Zielvariable

Der Wert, den Sie in überwachtem ML vorhersagen möchten. Dies wird auch als Ergebnisvariable bezeichnet. In einer Fertigungsumgebung könnte die Zielvariable beispielsweise ein Produktfehler sein.

Aufgabenliste

Ein Tool, das verwendet wird, um den Fortschritt anhand eines Runbooks zu verfolgen. Eine Aufgabenliste enthält eine Übersicht über das Runbook und eine Liste mit allgemeinen Aufgaben, die erledigt werden müssen. Für jede allgemeine Aufgabe werden der geschätzte Zeitaufwand, der Eigentümer und der Fortschritt angegeben.

Testumgebungen

Siehe Umgebung.

Training

Daten für Ihr ML-Modell bereitstellen, aus denen es lernen kann. Die Trainingsdaten müssen die richtige Antwort enthalten. Der Lernalgorithmus findet Muster in den Trainingsdaten, die die Attribute der Input-Daten dem Ziel (die Antwort, die Sie voraussagen möchten) zuordnen. Es gibt ein ML-Modell aus, das diese Muster erfasst. Sie können dann das ML-Modell verwenden, um Voraussagen für neue Daten zu erhalten, bei denen Sie das Ziel nicht kennen.

Transit-Gateway

Ein Netzwerk-Transit-Hub, über den Sie Ihre Netzwerke VPCs und Ihre lokalen Netzwerke miteinander verbinden können. Weitere Informationen finden Sie in der Dokumentation unter Was ist ein Transit-Gateway. AWS Transit Gateway

Stammbasierter Workflow

Ein Ansatz, bei dem Entwickler Feature lokal in einem Feature-Zweig erstellen und testen und diese Änderungen dann im Hauptzweig zusammenführen. Der Hauptzweig wird dann sequentiell für die Entwicklungs-, Vorproduktions- und Produktionsumgebungen erstellt.

Vertrauenswürdiger Zugriff

Gewährung von Berechtigungen für einen Dienst, den Sie angeben, um Aufgaben in Ihrer Organisation AWS Organizations und in deren Konten in Ihrem Namen auszuführen. Der vertrauenswürdige Service erstellt in jedem Konto eine mit dem Service verknüpfte Rolle, wenn diese Rolle benötigt wird, um Verwaltungsaufgaben für Sie auszuführen. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter Verwendung AWS Organizations mit anderen AWS Diensten.

Optimieren

Aspekte Ihres Trainingsprozesses ändern, um die Genauigkeit des ML-Modells zu verbessern. Sie können das ML-Modell z. B. trainieren, indem Sie einen Beschriftungssatz generieren,

Beschriftungen hinzufügen und diese Schritte dann mehrmals unter verschiedenen Einstellungen wiederholen, um das Modell zu optimieren.

Zwei-Pizzen-Team

Ein kleines DevOps Team, das Sie mit zwei Pizzen ernähren können. Eine Teamgröße von zwei Pizzen gewährleistet die bestmögliche Gelegenheit zur Zusammenarbeit bei der Softwareentwicklung.

U

Unsicherheit

Ein Konzept, das sich auf ungenaue, unvollständige oder unbekannte Informationen bezieht, die die Zuverlässigkeit von prädiktiven ML-Modellen untergraben können. Es gibt zwei Arten von Unsicherheit: Epistemische Unsicherheit wird durch begrenzte, unvollständige Daten verursacht, wohingegen aleatorische Unsicherheit durch Rauschen und Randomisierung verursacht wird, die in den Daten liegt. Weitere Informationen finden Sie im Leitfaden Quantifizieren der Unsicherheit in Deep-Learning-Systemen.

undifferenzierte Aufgaben

Diese Arbeit wird auch als Schwerstarbeit bezeichnet. Dabei handelt es sich um Arbeiten, die zwar für die Erstellung und den Betrieb einer Anwendung erforderlich sind, aber dem Endbenutzer keinen direkten Mehrwert bieten oder keinen Wettbewerbsvorteil bieten. Beispiele für undifferenzierte Aufgaben sind Beschaffung, Wartung und Kapazitätsplanung.

höhere Umgebungen

Siehe Umgebung.

V

Vacuuming

Ein Vorgang zur Datenbankwartung, bei dem die Datenbank nach inkrementellen Aktualisierungen bereinigt wird, um Speicherplatz zurückzugewinnen und die Leistung zu verbessern.

Versionskontrolle

Prozesse und Tools zur Nachverfolgung von Änderungen, z. B. Änderungen am Quellcode in einem Repository.

VPC-Peering

Eine Verbindung zwischen zwei VPCs , die es Ihnen ermöglicht, den Verkehr mithilfe privater IP-Adressen weiterzuleiten. Weitere Informationen finden Sie unter Was ist VPC-Peering? in der Amazon-VPC-Dokumentation.

Schwachstelle

Ein Software- oder Hardwarefehler, der die Sicherheit des Systems beeinträchtigt.

W

Warmer Cache

Ein Puffer-Cache, der aktuelle, relevante Daten enthält, auf die häufig zugegriffen wird. Die Datenbank-Instance kann aus dem Puffer-Cache lesen, was schneller ist als das Lesen aus dem Hauptspeicher oder von der Festplatte.

warme Daten

Daten, auf die selten zugegriffen wird. Bei der Abfrage dieser Art von Daten sind mäßig langsame Abfragen in der Regel akzeptabel.

Fensterfunktion

Eine SQL-Funktion, die eine Berechnung für eine Gruppe von Zeilen durchführt, die sich in irgendeiner Weise auf den aktuellen Datensatz beziehen. Fensterfunktionen sind nützlich für die Verarbeitung von Aufgaben wie die Berechnung eines gleitenden Durchschnitts oder für den Zugriff auf den Wert von Zeilen auf der Grundlage der relativen Position der aktuellen Zeile.

Workload

Ein Workload ist eine Sammlung von Ressourcen und Code, die einen Unternehmenswert bietet, wie z. B. eine kundenorientierte Anwendung oder ein Backend-Prozess.

Workstream

Funktionsgruppen in einem Migrationsprojekt, die für eine bestimmte Reihe von Aufgaben verantwortlich sind. Jeder Workstream ist unabhängig, unterstützt aber die anderen Workstreams

im Projekt. Der Portfolio-Workstream ist beispielsweise für die Priorisierung von Anwendungen, die Wellenplanung und die Erfassung von Migrationsmetadaten verantwortlich. Der Portfolio-Workstream liefert diese Komponenten an den Migrations-Workstream, der dann die Server und Anwendungen migriert.

WURM

Sehen Sie einmal schreiben, viele lesen.

WQF

Siehe AWS Workload-Qualifizierungsrahmen.

einmal schreiben, viele lesen (WORM)

Ein Speichermodell, das Daten ein einziges Mal schreibt und verhindert, dass die Daten gelöscht oder geändert werden. Autorisierte Benutzer können die Daten so oft wie nötig lesen, aber sie können sie nicht ändern. Diese Datenspeicherinfrastruktur gilt als unveränderlich.

Z

Zero-Day-Exploit

Ein Angriff, in der Regel Malware, der eine Zero-Day-Sicherheitslücke ausnutzt.

Zero-Day-Sicherheitslücke

Ein unfehlbarer Fehler oder eine Sicherheitslücke in einem Produktionssystem. Bedrohungsakteure können diese Art von Sicherheitslücke nutzen, um das System anzugreifen. Entwickler werden aufgrund des Angriffs häufig auf die Sicherheitsanfälligkeit aufmerksam.

Eingabeaufforderung ohne Zwischenfälle

Bereitstellung von Anweisungen für die Ausführung einer Aufgabe an einen LLM, jedoch ohne Beispiele (Schnappschüsse), die ihm als Orientierungshilfe dienen könnten. Der LLM muss sein vortrainiertes Wissen einsetzen, um die Aufgabe zu bewältigen. Die Effektivität von Zero-Shot Prompting hängt von der Komplexität der Aufgabe und der Qualität der Aufforderung ab. Siehe auch Few-Shot-Prompting.

Zombie-Anwendung

Eine Anwendung, deren durchschnittliche CPU- und Arbeitsspeichernutzung unter 5 Prozent liegt. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.