【Azure Key Vault】サービス概要

2025年9月30日

改訂履歴

版数	発行日	改訂内容
第1版	2025年9月30日	初版発行

本資料の内容は 2025/09/30 時点のものです。製品のアップデートにより変更となる場合がございます旨ご了承ください。

Agenda

1. 前提情報
    1. 用語集
2. Azure Key Vault とは
    1. Azure Key Vault とは
    2. Azure Key Vault の利用目的
    3. Azure Key VaultとMicrosoft Azureの連携性
3. セキュリティ課題とAzure Key Vaultによる解決策
    1. 企業のセキュリティ課題と解決策
    2. 課題:機密情報の漏洩リスク
    3. APIキーの重要性とリスク
    4. シークレット管理
    5. キー管理（ソフトウェア/HSM）
    6. シークレット管理とキー管理の違い
    7. 課題:証明書、キー管理の管理負荷による人的資源の枯渇
    8. 証明書管理
4. Azure Key Vault 認証の仕組み
    1. Azure Key Vault におけるアクセス制御の重要性
    2. ２種類のアクセス制御(アクセスポリシー)
    3. ２種類のアクセス制御(Azure RBAC)
    4. アクセス制御、特徴について
5. サービスプランとコスト構成
    1. プラン（Standard / Premium）選定の重要性
    2. Azure Key Vault の課金対象

1. 前提情報

1.1. 用語集

本書で使用する用語及び略称を以下の通り定義します。

No.	用語	説明
1	ハードウェア セキュリティ モジュール（HSM）	暗号鍵などの機密情報を物理的に安全な専用ハードウェアで保管・操作する仕組みです。
2	ハードコーディング	アプリケーションのソースコード内にパスワードやAPIキーなどの機密情報を直接書き込むことです。
3	ゼロトラストセキュリティ	ネットワークの内外にかかわらず、あらゆるアクセス主体（人やデバイス）を信用せず、全てのアクセスに対して「常に検証する」ことを前提とするセキュリティ戦略です
4	クラウドネイティブ	クラウドネイティブとは、クラウド環境やクラウドの特性・メリットを最大限に活用するために、最初からクラウド上で動作することを前提として設計・開発されたシステムやアプリケーション、またソフトウェアアプローチを指します。
5	キーコンテナー	Azureが提供するクラウドサービスで、アプリケーションやサービスで利用するパスワード、APIキー、証明書などの機密情報（シークレット）や暗号化キーを一元的に安全に管理し、厳格なアクセス制御を行うための仮想の金庫です。
6	CSIドライバー	CSI入門/Introduction to CSI - Speaker DeckCSIドライバーとは、Kubernetesなどのコンテナオーケストレーションシステムが、任意のブロックストレージやファイルストレージシステムにアクセスするための標準化されたインターフェース（プラグイン）です。
7	Pod	CSIドライバーにおける「Pod」は、アプリケーションをパッケージ化して管理する最小単位であり、コンテナを実行する環境を指します。

1.1. 用語集

No.	用語	説明
8	カスタマー管理キー	クラウド上のデータを暗号化するための「鍵（キー）」を、クラウド事業者ではなくユーザー自身が管理する方法です。
9	暗号鍵	暗号鍵とは、データを元の「平文」から解読できない「暗号文」に変換する際に使用される情報（文字列や数字など）です。
10	DevSecOps	開発（Development）、セキュリティ（Security）、運用（Operations）の略で、ソフトウェア開発ライフサイクルの全段階にセキュリティを組み込み、継続的なテストと自動化によって問題の早期発見と対応を可能にするアプローチです。
11	FIPS 140-2 Level 2	FIPS 140-2 レベル2とは、FIPS 140-2という米国連邦情報処理標準の暗号モジュールのセキュリティ要件における、物理的な改ざん防止対策と役割ベースの認証に関する追加の要件を指します。
12	GitHub（ギットハブ）	GitHub（ギットハブ）は、プログラマーがソースコードを保存、管理、共有するためのWebベースのプラットフォームです。
13	GDPR	GDPR（一般データ保護規則）は、EU域内で個人データの保護と取り扱いについて定められた法令で、EU域内の居住者の個人データを扱う全ての企業に適用されます。
14	HIPAA	HIPAA（米国の医療保険の携行性と責任に関する法律）は、1996年に制定された米国連邦法で、医療情報のプライバシーとセキュリティを保護することを目的としています。

1.1. 用語集

No.	用語	説明
15	セキュリティストア	Key Vaultの「セキュリティストア」とは、キー、パスワード、証明書などの機密情報を一元管理し、厳重に保護するクラウドサービスです。
16	サーバーレス関数	サーバーレス関数とは、「FaaS (Function as a Service)」とも呼ばれ、特定のイベントをトリガーとして実行される、サーバーの管理を意識せずにコードをデプロイ・実行できるコンピューティングサービスです。
17	コンテナアプリ	コンテナアプリとは、アプリケーションとその実行に必要なライブラリや設定などが一つのパッケージ（コンテナ）にまとめられたアプリケーションのことです。
18	CLI	CLIとは、テキストベースのコマンドを入力してコンピュータを操作するユーザーインターフェースです。
19	Managed HSM	Managed HSM（Hardware Security Module）とは、クラウド上で提供される、FIPS 140-3 レベル3認証を受けたハードウェア暗号モジュールをフルマネージドで利用できる高可用性サービスです。
20	SQL TDE	SQLのTDE（Transparent Data Encryption）とは、データベースの保存データを「透過的に」暗号化する機能です。
21	DevOps	DevOpsとは、「開発（Development）」と「運用（Operations）」を組み合わせた造語で、ソフトウェア開発とIT運用を統合し、連携を強化して迅速なリリースと高品質なサービス提供を目指す概念・文化・手法です。

1.1. 用語集

No.	用語	説明
22	SSL/TLS	SSL/TLSとは、インターネット上でサーバーとウェブブラウザの間でデータを暗号化して送受信するためのプロトコル（通信ルール）です。
23	サービスプリンシパル	サービスプリンシパルとは、人ではなく、アプリケーションやシステムがクラウドサービスのリソースにアクセスするための「ID」です。
24	マネージドID	マネージドIDとは、Azureのようなクラウド環境において、仮想マシンなどのリソースに割り当てられる、Microsoft Entra ID（Azure AD）が自動的に管理するIDです。
25	Azure Disk Encryption	Azure Disk Encryption（ADE）は、Azure仮想マシン（VM）のOSディスクとデータディスクを暗号化する機能です。
26	Azure Confidential Computing	Azure Confidential Computingとは、ハードウェアベースの信頼できる実行環境（TEE）を活用して、クラウド上での使用中のデータを処理中に暗号化し、保護する機能です。
27	Azure Virtual Machines	Azure Virtual Machines（VM）とは、Microsoft Azureが提供するクラウド上の仮想サーバーサービスです
28	Azure App Service	Azure App Serviceとは、WebアプリケーションやRESTful API、モバイルバックエンドなどを、インフラストラクチャの管理を気にせず開発・実行できる、Microsoft AzureのフルマネージドPaaS（プラットフォーム・アズ・ア・サービス）です。
29	Azure Functions	Azure Functions（アジュール ファンクションズ）とは、サーバー管理不要で特定のコードをイベントに応じて実行できるサーバーレスのサービスです。

1.1. 用語集

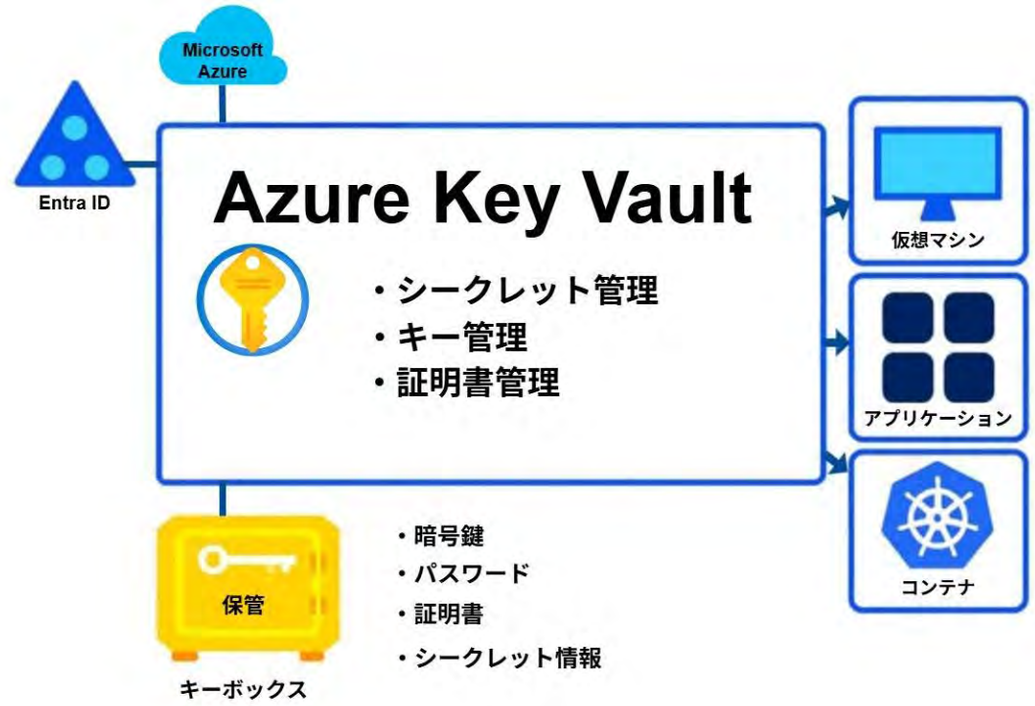
No.	用語	説明
30	Azure Kubernetes Service (AKS)	Azure Kubernetes Service（AKS）とは、マイクロソフトのクラウドプラットフォームであるAzure上で利用できる、マネージドKubernetesサービスです。
31	Azure Storage	Azure Storageとは、Microsoft Azureが提供する、可用性・スケーラビリティ・持続性・安全性に優れたクラウドストレージサービスです。
32	SQL Database	SQLデータベースとは、行と列からなる「テーブル」にデータを整理して格納するシステムで、特にリレーショナルデータベースを指します。
33	Power Automate	Power Automateとは、Microsoftが提供するノーコードRPAツールです。
34	Azure Synapse	Azure Synapse Analyticsは、エンタープライズデータウェアハウスとビッグデータ分析を統合した、Microsoft Azureの無制限な分析サービスです。
35	Azure Data Factory	Azure Data Factory（ADF）とは、Microsoft Azureが提供するクラウドベースの「データ統合サービス」です。
36	APIゲートウェイ	APIゲートウェイとは、クライアントアプリケーションからのリクエストを受け付け、バックエンドの各種サービスにルーティングし、認証、モニタリング、セキュリティなどの共通機能を担当する仲介役となるツールです。
37	Azure Resource Manager(ARM)	Azure Resource Manager（ARM）は、Azure上でリソースの作成、更新、削除を管理するためのサービスです。

2. Azure Key Vault とは

2.1. Azure Key Vault とは

Azure Key Vault とは、Azure上でアプリケーションやサービスが使用する暗号鍵、パスワード、証明書、およびその他のシークレット（秘密情報）を安全に管理・保護するためのクラウドサービスです。
ハードウェア セキュリティ モジュール（HSM）に準拠した環境でこれらの機密情報を格納し、アプリケーション内に直接ハードコーディングすることなく、安全にアクセスし利用することが可能です。
つまり、従来のオンプレミス環境では機密情報の管理・保護を物理層で行っていましたが、Azure Key Vault を活用することで、物理層に機密情報を保存する必要がなくなり、セキュリティレベルが大幅に向上します。
これは、ゼロトラストセキュリティやクラウドネイティブな運用の考え方として重要になるものです。

Azure Key Vault 概要イメージ



Azure Key Vault

✓ 機密情報・暗号資産の安全な保管と管理
✓ セキュリティ運用の効率化と自動化
✓ Azure Key Vault と Entra IDの連携により強固なアクセス制御

2.2. Azure Key Vault の利用目的

近年、クラウドの活用が急速に広がっている中で、クラウド運用時のセキュリティ対策は企業にとってきわめて重要なものとなっています。信頼性の高いクラウドサービスを利用していたとしても、クラウドのセキュリティ対策が不十分なまま利用されているケースは、少なくありません。

Azure Key Vault は、こうした企業や組織のニーズに応えるためのソリューションとして、より強固なセキュリティ体制を提供し、運用効率の向上を実現します。主な利用目的は以下の通りです。

Azure Key Vault の利用目的

■機密情報の安全な保管

パスワード、APIキー、証明書などの機密情報をクラウド上で安全に保管・管理を行います。コード内に直接機密情報を記述せず、Azure Key Vault内にある「キーコンテナー」で保管を行います。アプリケーションは認証を経てアクセスを行うため、情報漏洩のリスクを大幅に削減できます。

■証明書の管理と自動更新

SSL/TLS証明書などを安全に保管・管理することが可能です。また、証明書格納時に設定を行うことで証明機関（CA）と連携して自動的に証明書を更新することが可能となります。これにより更新漏れを防止し、運用を効率化できます。

■暗号鍵の管理と利用

Azure上で暗号鍵、シークレット（パスワードや接続文字列など）、証明書を一元的に管理・保護することが可能です。鍵の作成・制御、アクセス権限の管理、監査ログの記録などを提供し、ハードウェアセキュリティモジュール（HSM）で鍵を保護することで、アプリケーションのセキュリティと信頼性を向上させることができます。

■アクセス制御と監査ログ

Microsoft Entra IDと連携し、細かいアクセス制御と操作履歴の監査ログを提供することで、セキュリティ監査やコンプライアンス対応で活用することができます。

2.3. Azure Key VaultとMicrosoft Azureの連携性

Azure Key Vault（以下、Key Vault）は、Microsoft Azureとの連携性が非常に高く、AWSやGoogle Cloudなど他社クラウドの類似サービスと比較しても、技術的統合度において優位性があります。多くのAzureサービスがKey Vaultをセキュリティストアとして直接利用できるため、シークレット管理や暗号鍵の運用が一元化され、運用負荷の軽減にもつながります。
Key Vaultは、Microsoft Azureのセキュリティサービスの一部として、以下のようなAzureサービスと密接に連携しています。

サービス名	連携内容	利用例
Azure Virtual Machines	VMのディスク暗号化（Azure Disk Encryption）でKey Vaultを使用して暗号鍵を管理。	TLS証明書の自動インストールなど
Azure App Service / Azure Functions	アプリケーション設定でKey Vaultのシークレットを参照可能	Webアプリやサーバーレス関数の構成情報管理
Azure Kubernetes Service (AKS)	アプリケーションが動作するコンテナに、パスワードやAPIキーなどの機密情報を自動で安全に渡すことができます。	コンテナアプリのセキュアな構成管理
Azure Storage / SQL Database	サーバー側暗号化（Server-Side Encryption）でKey Vaultのカスタマー管理キー（CMK）を使用可能	データの暗号化とキー管理の分離
Azure Logic Apps / Power Automate	ワークフロー内でKey Vaultのシークレットを参照	自動化処理での安全な認証情報利用
Azure Synapse / Data Factory	接続情報や認証キーをKey Vaultで管理	データ統合・分析時のセキュリティ強化

技術的統合のメリット

- セキュリティ強化：アプリケーションコードに機密情報を埋め込まずに済む
- 運用効率化：キーやシークレットのローテーションを自動化することが可能
- 一元管理：複数サービスの機密情報をKey Vaultで集中管理
- アクセス制御の統一：Azure RBACを利用してKey Vaultへの権限を一元的に管理可能

3. セキュリティ課題とAzure Key Vaultによる解決策

3.1. 企業のセキュリティ課題と解決策

近年では、クラウド活用の加速に伴い、企業が扱う機密情報の管理がこれまで以上に重要な課題となっています。APIキーやパスワード、証明書などの機密情報が分散し、管理が煩雑化することで、情報漏洩や不正アクセスのリスクが高まっています。特に、開発・運用の現場では、これらの情報がコードや構成ファイルに埋め込まれるケースも多く、セキュリティインシデントの温床となりかねません。現在の組織が直面している主なセキュリティ課題とその解決策を説明します。

組織が抱えるセキュリティ課題

機密情報の漏洩リスク
パスワード、APIキー、証明書などの機密情報が、標的となる攻撃パターンが増加している

証明書やキーの管理負荷による人的資源の枯渇
SSL/TLS証明書や暗号化キーの更新・ローテーションを手動で行うため、管理が煩雑になり、脅威の発見が遅延する

課題解決ソリューション

Azure Key Vaultの活用

・機密情報の漏洩リスク

原因：パスワードやAPIキーがコードに埋め込まれ、攻撃対象になりやすい。

解決策：Key Vaultに機密情報を安全に格納し、アクセス制御と監査ログで漏洩を防止。

・証明書、キー管理の管理負荷による人的資源の枯渇

原因：手動更新による人的ミスや運用負荷の増大。

解決策：証明書・キーの自動ローテーションと一元管理により、運用効率を向上し、人的資源の枯渇を防止。

3.2. 課題:機密情報の漏洩リスク

企業や組織にとって、機密情報の漏洩リスクは、顧客データや業務機密などが外部に流出する重大な脅威です。このリスクは、システムの弱点だけでなく、人為的なミスや運用上の不備からも発生します。
クラウド環境では、単にデータを暗号化するだけでは不十分です。セキュリティの根幹を支える「シークレット」（パスワード、APIキーなど）と、「暗号鍵」（暗号化・署名などに使うキー）は、どちらも厳重に保護する必要があります。
Key Vaultは、この「シークレット」と「暗号鍵」の両方を一元管理できるプラットフォームです。この二つの管理機能を組み合わせることで、企業はより強固な情報セキュリティ体制を築くことができます。

■シークレット管理による漏洩防止
Key Vault のシークレット管理機能は、アプリケーションやサービスが必要とするパスワード、APIキー、接続文字列などの機密情報を安全に保管し、アクセス制御や監査ログによって不正利用や漏洩を防止します。これにより、コードや構成ファイルに機密情報を直接記述する必要がなくなり、セキュリティと運用効率の両面で企業、組織に貢献します。

■キー管理による暗号化の強化
Key Vault のキー管理機能は、企業、組織が扱う暗号化キー（対称鍵・非対称鍵）を安全かつ効率的に保管・運用するための仕組みです。これにより、機密情報の暗号化・復号、署名・検証などの処理を安全に実行でき、情報漏洩リスクの低減に貢献します。

3.3. APIキーの重要性とリスク

前のスライドで述べたように、クラウド環境における機密情報の保護には、暗号化だけでなく、シークレットの適切な管理が不可欠です。中でも、APIキーは外部サービスとの連携に広く利用される一方で、流出リスクが高く、攻撃者にとって非常に魅力的な標的となります。

このスライドでは、APIキーの基本的な役割と、流出によるリスク、そしてなぜ攻撃対象になりやすいのかについて、3つの観点から解説します。

APIキーとは

■外部サービス、アプリケーション、システムと連携するための「認証トークン」

このキーは、利用者を識別し、アクセス権限を制御するために機能します。たとえば、メール送信、クラウドストレージ、決済など、外部の機能をアプリケーションに組み込む際に、APIキーがなければこれらのサービスに安全に接続できません。つまり、APIキーは外部サービスとの通信において、認証と識別の両方を担う重要な役割を果たします。

・例：メール送信、クラウドストレージ、決済サービス、SNS連携など。

※機密情報と同等の扱いが必要

流出した場合のリスク

■主なリスクと影響

APIキーがソースコードに直接書き込まれていると、GitHubのようなコード管理ツールを通じて外部に流出するリスクがあります。一度流出してしまうと、以下のような深刻な影響が発生する可能性があります。

・不正アクセスによるデータ漏洩

・サービスの不正利用・課金

・システムの乗っ取りや改ざん

※分かりやすく例えると、鍵を盗まれた家と同じ状態

なぜ攻撃対象になるのか

■主なリスクと影響

APIキーを狙った攻撃が増えている主な理由は、AIの普及などでAPIの数が増え、管理が複雑になっているからです。認証設定の不備や、使われなくなった古いキーなど、管理が行き届いていない脆弱性が攻撃者にとって格好の標的となっています。

・GitHubなどに誤って公開されるケースが多い

・APIキーだけで認証が通る場合が多い

※攻撃者にとって「見つけやすく、使いやすい」標的

3.4. シークレット管理

シークレット管理を活用すると、アプリケーションコードや設定ファイルにパスワードやAPIキーなどの機密情報を直接記述する必要がなくなります。これにより、情報漏洩のリスクを大幅に低減できます。
例えるなら「金庫の暗証番号をアプリケーションのコードに直接記載するのではなく、鍵付きの安全な金庫に保管するようなものです。
さらに、Entra IDとの連携によるアクセス制御により、「誰が」「どの情報に」アクセスできるかを厳密に管理することができます。これにより、内部不正や誤操作への対策としても非常に有効です。
以下では、Key Vault のシークレット管理が組織にもたらすメリットと、利用にあたっての注意点について説明します。

メリット

■情報漏洩リスクの低減
・機密情報を暗号化して安全に保管し、コードから分離することで漏洩リスクを最小化。

■Entra IDとの連携によるアクセス制御
・Entra IDと連携したAzure RBACにより、最小権限でのアクセス管理を実現。
・インシデント発生時に「誰が・いつ・どのシークレットにアクセスしたか」を追跡でき、原因究明と早期対策の実施が可能。

■運用負荷の軽減
・証明書やキーの自動ローテーションにより、手動管理の手間を削減。

■クラウドサービスとの統合性
・Azureの各種サービスとシームレスに連携し、セキュアな開発・運用を支援。

注意点

■最小権限の原則を徹底
・Azure RBACを適切に設定し、不要なアクセス権を排除。アプリケーションやユーザーに必要最低限の権限のみを付与。

■シークレットの有効期限とローテーション管理
・シークレットに有効期限を設定し、期限切れによる障害を防止。自動ローテーション機能を活用し、人的ミスや更新漏れを回避。

■不要なシークレットの定期的な削除
・使用されていないシークレットは削除し、誤使用や漏洩のリスクを排除。
定期的な棚卸しを実施することが推奨される。

■アクセスログの有効化と定期的な確認
・Key Vaultの操作履歴はAzureのアクティビティログに記録されるが、Azure MonitorやLog Analyticsと連携することで、より詳細な監査が可能。
定期的にログを確認し、不審なアクセスを早期に検知。

3.4. シークレット管理

実際にKey Vaultがシークレット管理機能を用いてどのようにAPIキーを保護するのかについてコードベースで対応例を説明します。コード内に直接APIキーを記述せず、Key Vault（シークレット管理）を挟むことでGitなどのコード管理ツールを通じて外部に流出やAPIキーを狙った攻撃から保護することが可能です。

コード例

■従来のコーディング

```
# APIキーを直接コードに記述
api_key = "01_test_12345abcd"
```

■Key Vault（シークレット管理）を用いたコーディング

```
# Key VaultからAPIキーを取得
api_key = client.get_secret("MyApiKey").value
```

従来のコーディングでは、APIキーをコード内に直接記述（ハードコーディング）する必要がありました。この方法は、ソースコードの公開や共有時にAPIキーが漏洩するリスクを高めるため、セキュリティ上の大きな課題となります。
※APIキー例:"01_test_12345abcd"

Key Vault のシークレット管理機能を活用することで、APIキーをコード内に直接書き込む必要がなくなります。代わりに、アプリケーションから安全な認証を通して Azure Key Vault にアクセスし、APIキー（例："MyApiKey"）を動的に取得する処理を追加します。これにより、機密情報であるAPIキー（例："01_test_12345abcd"）がコードに露出するリスクをなくし、セキュリティを大幅に向上させることができます。

まとめ

企業が所有するアプリケーションの数だけ、APIキーやパスワードといったシークレット（機密情報）が存在します。これらを個別で管理すると、管理が煩雑になるだけでなく、退職者が使っていたキーが放置されたり、誤ってコードに直接書き込まれたりするリスクが生じます。
これらの課題に対処するため、Key Vaultによるシークレットの一元管理が不可欠です。Key Vaultを活用することで、シークレットの安全な保管、アクセス制御、自動ローテーション、監査ログの取得などが可能になり、セキュリティと運用の両面から組織全体のセキュリティレベルを向上させることができます。

3.5. キー管理（ソフトウェア/HSM）

Key Vault のキー管理機能は、暗号化・復号・署名・検証などに使用される暗号鍵（対称鍵／非対称鍵）を、安全かつ効率的に保管・運用するための仕組みを提供します。特に、セキュリティ要件に応じて選択可能な「ソフトウェアキー」と「HSMキー」の2種類の管理方式により、柔軟かつ強固なセキュリティ設計が可能です。以下では、Key Vault のキー管理機能が組織にもたらすメリットと、利用にあたっての注意点について説明します。

メリット

■ キーの生成・保管・使用を一元管理

・暗号鍵をクラウド上で安全に生成し、ハードウェア セキュリティ モジュール（HSM）によって保護された状態で保管できます。これにより、アプリケーションやサービスが必要とする暗号処理を安全かつ効率的に実行することが可能です。

■ キーのライフサイクル管理

・キーの有効期限、ローテーション、廃棄などのライフサイクルを自動化・制御でき、セキュリティポリシーに沿った運用が可能です。

■ アクセス制御と監査ログ

・ Entra ID(旧Azure AD)と連携し、キーへのアクセス権限を細かく制御できます。また、すべての操作はログに記録され、監査やコンプライアンス対応にも有効です。

注意点

■ セキュリティ要件の明確化

・業務内容や法令・業界基準（例：FIPS 140-2、GDPR、HIPAAなど）に応じて、どちらの方式が適切かを事前に確認することが重要です。

■ アクセス制御と監査設定

・キーへのアクセスは Entra IDと連携したAzure RBACで厳密に管理することが重要です。

■ キーのライフサイクル管理

・ 自動ローテーションや期限切れ通知の設定を活用し、運用負荷を軽減しながら安全性を確保することが重要です。

3.5. キー管理（ソフトウェア/HSM）

暗号鍵は、情報セキュリティの根幹を担う重要な要素です。その管理には、技術的な選定と運用設計の両面から慎重な対応が求められます。
Key Vault を利用する際、ユーザーは用途やセキュリティ要件に応じて、適切なキー管理方式を選択する必要があります。これは、例えるなら「金庫の鍵の種類を選ぶ」ようなもので、選択した方式によって安全性や運用の柔軟性が大きく変わります。
このスライドでは、キー管理方式の使い分けと、運用上の留意点について整理します。

ソフトウェアキー方式

ソフトウェアキー方式では、暗号化キーがソフトウェアベースで生成・保管され、Azure のクラウド環境内で安全に管理されます。キーは Microsoft によって保護され、暗号処理はソフトウェアレベルで実行されるため高いセキュリティを提供します。※イメージだとオフィスの鍵付きの引き出しに鍵をしまってある状態です。

■ **主な特徴：**・導入が容易で、Azure ポータルや CLI、API を通じて簡単に操作可能

・コスト効率が高く、開発・テスト環境や一般業務システムに適している

・Azure の各種サービス（Storage、SQL Database、VMなど）と統合しやすい

■ **注意点：**　・普通の業務なら問題ないが、法律や業界ルールでもっと強い保護が必要な場合は注意が必要

HSM キー方式

HSMキー方式では、暗号化キーがハードウェアベースのセキュリティモジュール（HSM）内で生成・保管されます。Azure Key Vault は、FIPS 140-2 Level 2 以上に準拠した Microsoft 管理の HSM を使用しており、より高いセキュリティレベルを提供します。※イメージだと銀行の金庫室に鍵をしまってる状態です。

■ **主な特徴**：・キーが物理的に隔離された環境で保護されるため、セキュリティが非常に高い

・金融、医療、公共機関など、厳格なセキュリティ基準が求められる業界に適している

・Managed HSM を利用することで、専用のHSM環境を構築・運用することも可能

■ **注意点：**　・利用にあたり、料金と運用のルールが厳格に必要。本当に重要な情報なのか見極める必要がある

3.6. シークレット管理とキー管理の違い

クラウド環境において、機密情報の保護は単なる暗号化だけでは不十分です。パスワードやAPIキーなどの「シークレット」と、暗号化・署名・認証に使用される「暗号鍵」は、それぞれ異なる役割を持ちながらも、情報セキュリティの根幹を支える重要な要素です。シークレット管理とキー管理を連携させて運用することで、より強固で効率的なセキュリティ体制を構築することが可能です。Key Vault における「シークレット管理」と「キー管理」は、どちらも機密情報を安全に扱うための機能ですが、目的・対象・利用方法が異なります。以下にその違いを説明します。

比較項目	シークレット管理	キー管理（ソフトウェア／HSM）
対象	パスワード、APIキー、接続文字列などの「認証情報」	暗号化・復号・署名などに使用する「暗号鍵」
目的	アプリケーションやサービスが安全に認証・接続するための情報を保管	データの暗号化・復号、署名・検証などの暗号処理を安全に実行
「保管・運用の仕組み」	バージョン管理、アクセス制御、監査ログ	ソフトウェアキー／HSMキーによる保管、ライフサイクル管理、アクセス制御、監査ログ
セキュリティレベル	中～高（暗号化＋Azure RBAC）	高（特にHSMキーは物理的に保護）
主な利用例	WebアプリのDB接続文字列、API認証キー	Azure Storageの暗号化、トークン署名、SQL TDE など
Azureサービスとの統合	App Service、Function、Logic Apps など	Storage、SQL Database、VM など

運用者目線での違い

・シークレット管理
開発者やDevOps担当者が日常的に利用する機能で、アプリケーションの構成情報や認証情報を安全に保管・参照するために使用されます。

・キー管理（ソフトウェア／HSM）
セキュリティ担当者やインフラ管理者が主に扱う機能で、暗号化処理や署名など、より高度なセキュリティ制御が求められる場面で使用されます。

3.7. 課題:証明書、キー管理の管理負荷による人的資源の枯渇

企業の情報システムにおいて、暗号化キーや証明書の管理はセキュリティの根幹を担う重要な業務です。しかし、証明書の発行・更新・期限管理・失効処理などは手作業が多く、運用担当者にとって大きな負担となる傾向があります。
特に、複数システム・環境にまたがる証明書管理では、人的ミスや更新漏れが発生しやすく、セキュリティリスクの増加や人的資源の枯渇につながる恐れがあります。
このような課題に対して、Key Vault の証明書管理機能は有効な解決策を提供します。

■証明書・キーの自動ローテーションと一元管理により、運用効率を向上し、人的資源の枯渇を防止。

Key Vault では、証明書のライフサイクル（発行、更新、ローテーション、失効）を一元的に管理できるほか、自動更新機能を活用することで、運用者の手作業を大幅に削減できます。さらに、証明書のバックエンドには Key Vault のキー管理機能が連携しており、証明書に紐づく秘密鍵も安全に保管されます。

3.8. 証明書管理

Key Vault では、証明書のライフサイクル（発行、更新、ローテーション、失効）を一元的に管理できるほか、自動更新機能を活用することで、運用者の手作業を大幅に削減できます。さらに、証明書のバックエンドには Key Vault のキー管理機能が連携しており、証明書に紐づく秘密鍵も安全に保管されます。
また、Entra IDとの統合によるアクセス制御や、監査ログの取得により、誰がいつどの証明書にアクセス・操作したかを追跡可能であり、内部統制の強化にも貢献します。
これにより、証明書・キー管理にかかる人的負荷を軽減し、限られたリソースでも高いセキュリティレベルを維持することが可能となります。

メリット

■ 証明書の安全な保管
・証明書（および秘密鍵）は暗号化された状態でKey Vault に格納され、アクセス制御により不正利用を防止。

■ ライフサイクルの自動化
・証明書の発行、更新、ローテーション、失効を自動化できるため、手動管理によるミスや更新漏れを防止。

■ 運用負荷の軽減
・管理者が手動で証明書を更新・配布する必要がなくなり、運用効率が向上。
Azureサービス（App Service、Function、VMなど）と連携することで、証明書の利用も自動化可能。

■コンプライアンス対応の強化
操作履歴は監査ログとして記録され、Azure Monitor や Log Analytics による可視化が可能。金融・医療・公共分野など、厳格なセキュリティ基準に対応しやすくなる。

注意点

■ 証明書発行機関（CA）との連携要件
・自動更新を利用するには、Key Vault が対応する CA（例：DigiCert、GlobalSignなど）との事前設定が必要。CAとの契約やAPI連携の確認を怠ると、自動化が機能しない可能性あり。

■ 利用先との連携設計
・証明書を利用するアプリケーションやサービスとの連携方法（参照方法、更新タイミングなど）を明確にしておく必要がある。

■ アクセス制御の適切な設定
・ 誤ったアクセス権限設定により、証明書や秘密鍵が漏洩するリスクがある。
Azure RBACやポリシー設定を導入初期から厳密に構成することが推奨される。

■ アクセス制御の適切な設定
セキュリティインシデント対応や外部監査に備え、操作ログの取得と保管を必ず有効化しておくこと。

3.8. 証明書管理

証明書の自動発行・自動更新を活用するには、Microsoft が提携する証明書発行機関（CA）から発行された証明書を使用する必要があります。
また、非提携CAから発行された証明書を利用する場合、一部機能が利用できないなどの制約があります。
現在、Key Vault が自動管理に対応している主な提携CAについて説明します。

CA名	対応証明書	主な特徴
DigiCert（CertCentral）	OV/EV SSL/TLS証明書	・CertCentral アカウントとの連携により、Key Vault から直接 SSL/TLS 証明書を発行・更新可能 ・APIキー、アカウントID、組織IDを用いて Key Vault に統合 ・証明書の有効期限が近づくと、Key Vault が自動的に更新処理を実行 ・発行された証明書と秘密鍵は Key Vault に安全に格納され、アプリケーションから透過的に利用可能 ・EV（Extended Validation）や OV（Organization Validation）証明書にも対応
GlobalSign	OV/EV SSL/TLS証明書	・Key Vault から直接証明書を発行・更新できる Managed SSL（MSSL）サービスと連携 ・アカウントID、管理者情報（氏名、メール、電話番号など）を用いて Key Vault に統合 ・自動更新に対応しており、証明書の期限切れによるサービス停止リスクを軽減 ・OV証明書を中心に、企業認証に適した証明書を提供

非提携CAを利用する場合の制約

・Key Vaultにインポートして管理することは可能ですが、自動更新（ローテーション）機能は利用不可
・手動で更新・再インポートが必要

4. Azure Key Vault 認証の仕組み

4.1. Azure Key Vault におけるアクセス制御の重要性

Key Vault は、機密情報、暗号鍵、証明書などを安全に管理するためのサービスです。これらの重要な情報を保護するには、「誰が、どの情報に、どのようにアクセスできるか」を厳密に制御することが不可欠です。特に、セキュリティ上重要な領域では、アクセス制御の設計が情報漏洩や誤操作のリスクを大きく左右します。
Key Vault は、Entra IDと統合することで、IDベースの認証とアクセス制御を実現しています。ユーザー、サービスプリンシパル、マネージドIDなどの認証はすべて Entra IDが担い、Key Vault はその認証結果に基づいて、Azure RBAC方式またはアクセスポリシーを適用します。
この構成により、Key Vault は ID管理を Entra IDに委ねることで、セキュリティの強化とスケーラビリティの向上を両立した運用が可能となります。

アクセスポリシー方式

Key Vault 独自のアクセス管理モデルです。ユーザーやアプリケーションが Key Vault 内のリソース（キー、シークレット、証明書）にアクセスする際に、Entra IDによる認証と、Key Vaultに設定されたアクセスポリシーによる認可の2段階で制御されます。また、多要素認証（MFA）や条件付きアクセス、IP 制限など、Entra ID のセキュリティポリシーをKey Vault へのアクセスにも適用できるため、より強固なセキュリティを実現できます。

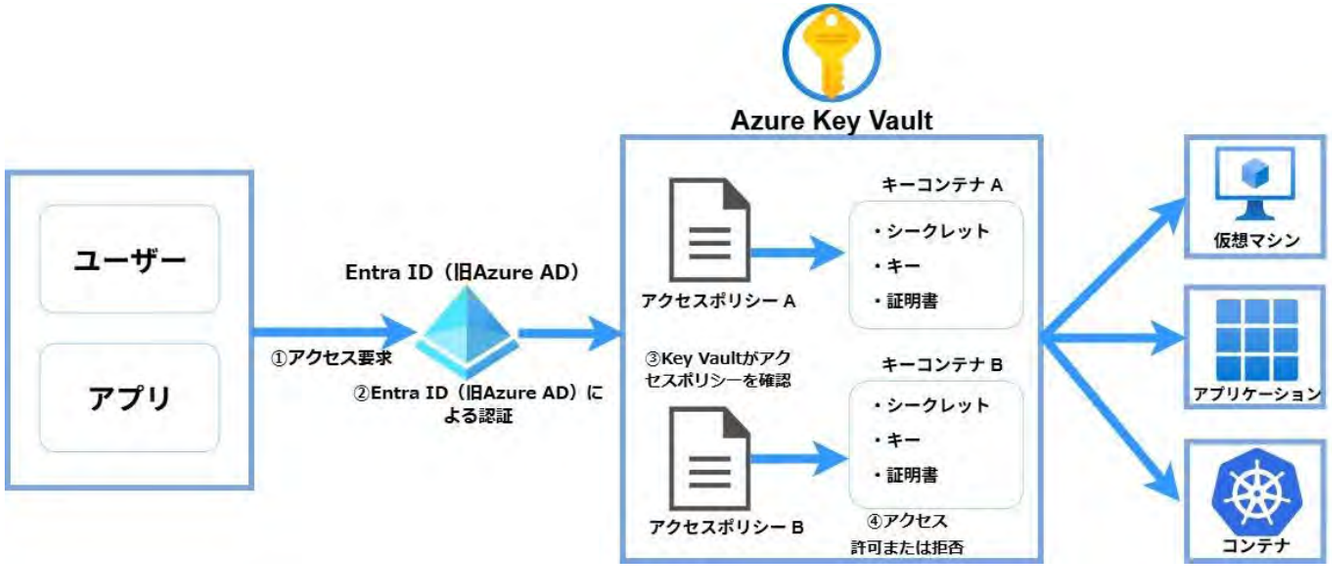
Azure RBAC方式

個々のユーザーに権限を割り当てるのではなく、「管理者」や「閲覧者」といった役割（ロール）に基づいてアクセス権限を管理します。これにより、ユーザーは割り当てられた役割に応じて、必要な権限を自動的に取得できます。この仕組みは、従来のアクセスポリシー方式よりもスケーラブルで、組織全体で統一的なアクセス管理を可能にします。

制御方式	特徴	向いているケース
アクセスポリシー方式	Key Vault独自の設定画面で、誰が何をできるか細かく指定	Key Vault単体で管理したい場合
Azure RBAC方式	Azure全体で使われる「ロールベースのアクセス制御」を使う	他のリソースと一貫した管理をしたい場合

4.2. ２種類のアクセス制御(アクセスポリシー)

アクセスポリシー方式は、Key Vault専用のアクセス管理方法で、以下のような流れで動きます。



■ **アクセスポリシーの認証の流れ**

①アクセス要求

ユーザー、アプリケーションが、Key Vault に保存されたシークレットや証明書などの機密情報にアクセスします。

②Entra ID による認証：Entra IDがユーザーの身元を確認

Key Vaultへのアクセス要求があると、Azureのサービス基盤（APIゲートウェイなど）は、そのユーザーやアプリケーションが正当なIDかどうかを確認するために、Entra IDに認証を依頼します。

認証が成功すると、Entra ID はアクセストークンを発行します。

③Key Vaultがアクセスポリシーを確認

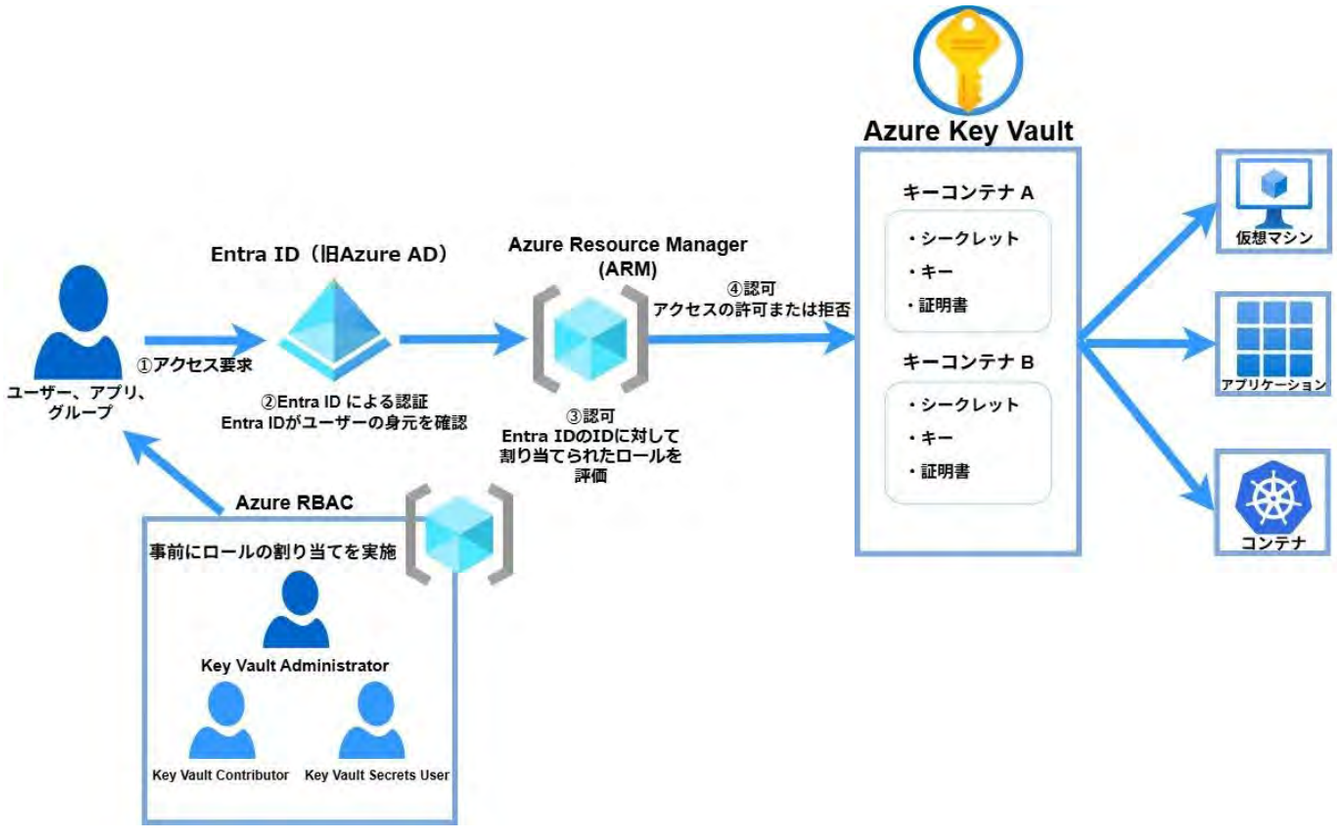
Key Vault は、受け取ったアクセストークンに含まれる ID 情報をもとに、該当するアクセスポリシーを確認します。

④アクセスの許可または拒否

Key Vault では、アクセスポリシーに基づいて操作の許可・拒否が判断されます。指定された操作に対して適切な権限がある場合、Key Vault はアクセスを許可します。権限が不足している場合は、403 Forbidden エラーなどでアクセスが拒否されます。

4.3. ２種類のアクセス制御(Azure RBAC)

Azure RBACは、Azure全体で使える共通のアクセス管理方法で、Key Vaultでも利用できます。
以下は、Azure RBACの認証流れとなります。



■ **Azure RBACの認証の流れ**

①アクセス要求

ユーザー、グループ、またはアプリケーション（マネージドIDなど）が、Key Vault に保存されたシークレットやキー、証明書にアクセスアクセスしようとします。

②Entra ID による認証：Entra IDがユーザーの身元を確認

Key Vaultへのアクセス要求があると、Azureのサービス基盤（APIゲートウェイなど）は、そのユーザーやアプリケーションが正当なIDかどうかを確認するために、Entra IDに認証を依頼します。

認証が成功すると、Entra ID はアクセストークンを発行します。

③認可：Azure Resource Managerがロールを評価

認証されたIDに対して、Azure Resource Manager（ARM）がAzure RBACのロール割り当てを評価します。

ARMは「このIDはこのリソースに対して、どんな操作が許可されているか？」を判断します。

④アクセスの許可または拒否

ARMがロールに基づいてアクセスを許可すれば、リソース操作が実行されます。

許可されていない操作は拒否され、403エラーなどが返されます。

4.3. ２種類のアクセス制御(Azure RBAC)

Azure RBACを効果的に運用するためには、事前に適切なロールを設計・振り分けしておくことが非常に重要です。誰がどの操作を必要とするかを明確にし、それに応じたロールを割り当てることで、不要な権限の付与を防ぎ、セキュリティリスクを最小限に抑えることができます。
以下は、Key Vault において「キー」に関する操作を制御するための代表的な Azure RBACにおけるロールです。

ロール名	説明	用途例
Key Vault Crypto Officer	概要：キーに対して、アクセス許可の管理を除くすべての操作が可能。 主な操作：キーの作成、削除、更新、暗号化・復号、署名・検証など。	暗号鍵のライフサイクル管理を担当するセキュリティ管理者。
Key Vault Crypto User	概要：既存のキーを使った暗号操作が可能。 主な操作：暗号化、復号、署名、検証。	アプリケーションがデータの暗号化・復号を行う際に使用。
Key Vault Crypto Service Encryption User	概要：キーのメタデータの読み取りと、Wrap/Unwrap 操作が可能。 主な操作：WrapKey、UnwrapKey。	Azure Disk Encryption や SQL TDE などのサービス連携。
Key Vault Crypto Service Release User	概要：Azure Confidential Computing などでのリリースキー操作が可能。 主な操作：ReleaseKey。	セキュアなハードウェア環境でのキー使用。
Key Vault Administrator	概要：Key Vault 内のすべてのデータプレーン操作が可能（キー、シークレット、証明書）。 主な操作：キーの管理に加え、シークレット・証明書の操作も可能。	Key Vault 全体の運用管理者。
Key Vault Reader	概要：キーのメタデータの読み取りのみ可能。キーの使用は不可。 主な操作：Get Key Properties。	監査や構成確認を行う担当者。

4.4. アクセス制御、特徴について

ここまでのスライドでは、Key Vault における 2 種類のアクセス制御方式 — アクセスポリシー方式と RBAC方式 — の認証の流れについて説明しました。
このスライドでは、それぞれの方式の主な特徴を整理し、用途や運用規模に応じた使い分けのポイントをまとめます。

比較項目	アクセスポリシー方式	Azure RBAC方式
制御単位	Key Vaultごとに個別設定 Key Vault固有のアクセス制御	Azure全体で統一されたロール管理 Azure全体の権限管理と統合されている
設定対象	ユーザー、アプリ、マネージドID	ユーザー、グループ、サービスプリンシパルなど
権限の粒度	操作単位（Get、List、Signなど）で細かく設定可能	ロール単位（Reader、Contributorなど）で一括管理 ロール名(Key Vault Secrets User、Key Vault Contributor、Reader)
最大設定数	最大1,024件まで	制限なし（Azure RBACのスケーラビリティに依存）
柔軟性	高い（細かい制御が可能）	中程度（ロールに依存）
スケーラビリティ	低（大規模環境では管理が煩雑）	高（複数Vaultを一括管理しやすい）
推奨用途	小規模環境、個別制御が必要な場合 例:スタートアップや小規模プロジェクトでの利用など	大規模環境、統一管理が求められる場合 例:金融機関や医療機関など、厳格なアクセス管理が求められる業界など

補足

・Azure RBAC方式は、Key Vault 以外の Azure リソースとも統一的に管理できるため、クラウド全体のセキュリティポリシーと整合性を取りやすいです。
・アクセスポリシー方式は、より細かい制御が必要な場面や、既存の運用に合わせた柔軟な設定が求められる場合に有効です。

5. サービスプランとコスト構成

5.1. プラン（Standard / Premium）選定の重要性

Key Vaultは、Microsoft Azureが提供する従量課金制のクラウドサービスであり、専用のライセンス製品は存在しません。
ただし、暗号鍵（キー）を操作するための2種類のサービスプラン（SKU）が用意されています。1 つ目は、ソフトウェアキーを使用して暗号化を行う Standard プランです。2 つ目は、専用のハードウェア（HSM）で守られた高セキュリティな鍵を使って暗号化を行うPremiumプラン となっています。プランごとに、各種操作に対する価格設定が異なるため、利用を検討する際には、操作ごとの価格設定を確認する必要があります。

Standard プラン

■特徴

・コストを抑えながら基本的なセキュリティ機能を提供するため、一般的な業務システムやWebアプリケーションに最適です。
・Azureサービスとの連携も容易で、Azure RBACによるアクセス制御や高可用性など、日常的な運用に必要な機能が揃っています。
・すべての操作はトランザクション単位で課金されます。

■利用に際して、注意点

・構築後にPremiumプランへ変更は可能ですが、変更には、CLIやPower Shellを使う必要があります。将来的に高度なセキュリティ（HSMキー）を使う可能性がある場合、最初からPremiumプランで構築を行う方が運用が複雑化せずシンプルとなります。

Premium プラン

■特徴

・HSM（Hardware Security Module）による物理的なキー保護を提供し、金融・医療・公共機関など、法令遵守や高度なセキュリティが求められる業界に適しています。
・顧客管理キー（CMK）によるキー主権の確保や、電子署名、証明書の高度な管理など、セキュリティと運用の両立が必要な場面で選ばれます。

■利用に際して、注意点

・Standardに比べて月額料金・トランザクション料金が高額です。
・すべてのAzureリージョンで利用できるわけではありません。利用可能なリージョンは、公式ドキュメントまたは、Azure Portalで確認が必要です。
・HSM保護キーを利用する場合、月額課金＋操作課金が発生します。

まとめ

一言でまとめると

Standardプラン、通常の業務や一般的な用途で選ばれ、運用において「技術面から誤って高額なサービス（HSMキー）の使用を出来なくするための制限付きプラン」です。
Premiumプラン、「より高度なセキュリティが求められる場面」で選ばれるプランです。

5.2. Azure Key Vault の課金対象

Key Vaultは、サービス自体の利用は無料ですが、Key Vault内での操作に対して料金が発生します。
料金は、利用するプラン（StandardまたはPremium）によって異なります。最新の価格情報は、必ずMicrosoftの公式サイト「Key Vault の価格」で確認してください。

課金項目	課金対象	Standard	Premium
シークレット操作	シークレットの取得・登録・更新・削除	課金対象（トランザクション単位）	課金対象（トランザクション単位）
キー操作（ソフトウェア保護）	暗号鍵の生成・使用（暗号化/復号/署名など）	課金対象（トランザクション単位）	課金対象（トランザクション単位）
キー操作（HSM保護）	HSMで保護された鍵の使用	利用不可	課金対象（月額＋トランザクション）
証明書操作	証明書の取得・登録・更新・削除	課金対象（トランザクション単位）	課金対象（トランザクション単位）
証明書の更新	自動更新リクエスト	課金対象（更新単位）	課金対象（更新単位）
ストレージキーのローテーション	Azure Storageキーの自動更新	課金対象（更新単位）	課金対象（更新単位）
キーのローテーション	スケジュールによる自動ローテーション	課金対象（ローテーション単位）	課金対象（ローテーション単位）