クラウドセキュリティ対策に必須！
シークレット管理のポイントを解説

東京エレクトロン デバイス株式会社

2024年3月12日

中林 稔

本日の内容

- シークレット管理とは
- シークレット管理の課題
- HashiCorp Vaultで課題解決

シークレット管理とは

シークレットとは？

ユーザーやシステムが他のシステムやデータのアクセスできるようにするもの

- ユーザー名/パスワード
- クラウドの認証情報
- APIトークン
- TLS証明書
- SSH鍵、暗号鍵

など

絶対に流出・紛失してはいけないもの

- 悪意を持ったアクセスを許してしまう
- 悪意を持ったアクションを許してしまう

シークレットを管理するとは？

- **パスワード、APIキー、証明書、トークンなどの機密情報を安全に扱い、保管するプロセス**

  シークレットのライフサイクルを適切に管理する

  - 作成
  - 保管
  - アクセス制御
  - 監視
  - ローテーション
  - 廃棄

情報漏えいのリスクを最小限に抑え、システムのセキュリティを保つために重要！

セキュリティ対策とライフサイクル管理

予防的統制

セキュリティインシデントの原因となる問題を取り除く、
もしくは、軽減させるための対策を行うこと

例:
・ユーザーには必要最小限の権限しか与えないことを徹底する
・パスワードを自動変更する
など

＋

発見的統制

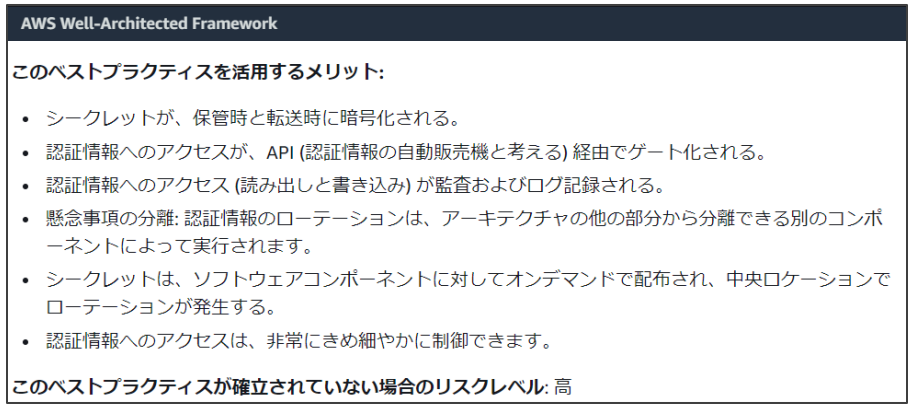
想定したリスクが発生した場合に早期検知と記録を
行い、影響を抑える対策を行うこと

例:
・監査・監視によって不正アクセスを見つける
・通常とは異なるユーザーの異常な振る舞いを分析し、メールで通知
など

ライフサイクル管理

シークレット管理のベストプラクティス

- 定期的なローテーション
- 利用者に応じた細かな権限管理
- 監査記録
- 暗号化

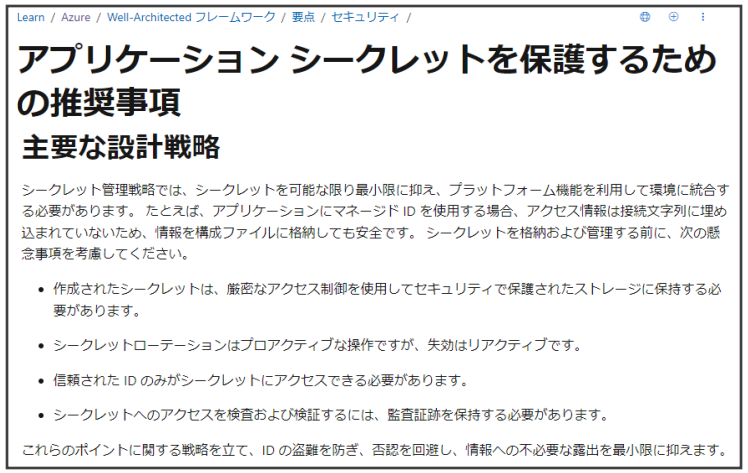


AWS Well-Architected Framework

このベストプラクティスを活用するメリット:

- シークレットが、保管時と転送時に暗号化される。
- 認証情報へのアクセスが、API (認証情報の自動販売機と考える) 経由でゲート化される。
- 認証情報へのアクセス (読み出しと書き込み) が監査およびログ記録される。
- 懸念事項の分離: 認証情報のローテーションは、アーキテクチャの他の部分から分離できる別のコンポーネントによって実行されます。
- シークレットは、ソフトウェアコンポーネントに対してオンデマンドで配布され、中央ロケーションでローテーションが発生する。
- 認証情報へのアクセスは、非常にきめ細やかに制御できます。

このベストプラクティスが確立されていない場合のリスクレベル: 高



Learn / Azure / Well-Architected フレームワーク / 要点 / セキュリティ /

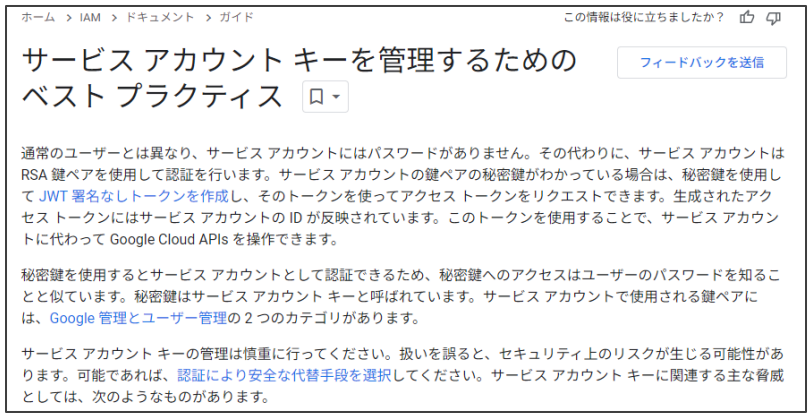
アプリケーション シークレットを保護するための推奨事項

主要な設計戦略

シークレット管理戦略では、シークレットを可能な限り最小限に抑え、プラットフォーム機能を利用して環境に統合する必要があります。 たとえば、アプリケーションにマネージド ID を使用する場合、アクセス情報は接続文字列に埋め込まれていないため、情報を構成ファイルに格納しても安全です。 シークレットを格納および管理する前に、次の懸念事項を考慮してください。

- 作成されたシークレットは、厳密なアクセス制御を使用してセキュリティで保護されたストレージに保持する必要があります。
- シークレットローテーションはプロアクティブな操作ですが、失効はリアクティブです。
- 信頼された ID のみがシークレットにアクセスできる必要があります。
- シークレットへのアクセスを検査および検証するには、監査証跡を保持する必要があります。

これらのポイントに関する戦略を立て、ID の盗難を防ぎ、否認を回避し、情報への不必要な露出を最小限に抑えます。



ホーム > IAM > ドキュメント > ガイド

この情報は役に立ちましたか？

サービス アカウント キーを管理するためのベスト プラクティス

フィードバックを送信

通常のユーザーとは異なり、サービス アカウントにはパスワードがありません。その代わりに、サービス アカウントは RSA 鍵ペアを使用して認証を行います。サービス アカウントの鍵ペアの秘密鍵がわかっている場合は、秘密鍵を使用して JWT 署名なしトークンを作成し、そのトークンを使ってアクセス トークンをリクエストできます。生成されたアクセス トークンにはサービス アカウントの ID が反映されています。このトークンを使用することで、サービス アカウントに代わって Google Cloud APIs を操作できます。

秘密鍵を使用するとサービス アカウントとして認証できるため、秘密鍵へのアクセスはユーザーのパスワードを知ることと似ています。秘密鍵はサービス アカウント キーと呼ばれています。サービス アカウントで使用される鍵ペアには、Google 管理とユーザー管理の 2 つのカテゴリがあります。

サービス アカウント キーの管理は慎重に行ってください。扱いを誤ると、セキュリティ上のリスクが生じる可能性があります。可能であれば、認証により安全な代替手段を選択してください。サービス アカウント キーに関連する主な脅威としては、次のようなものがあります。

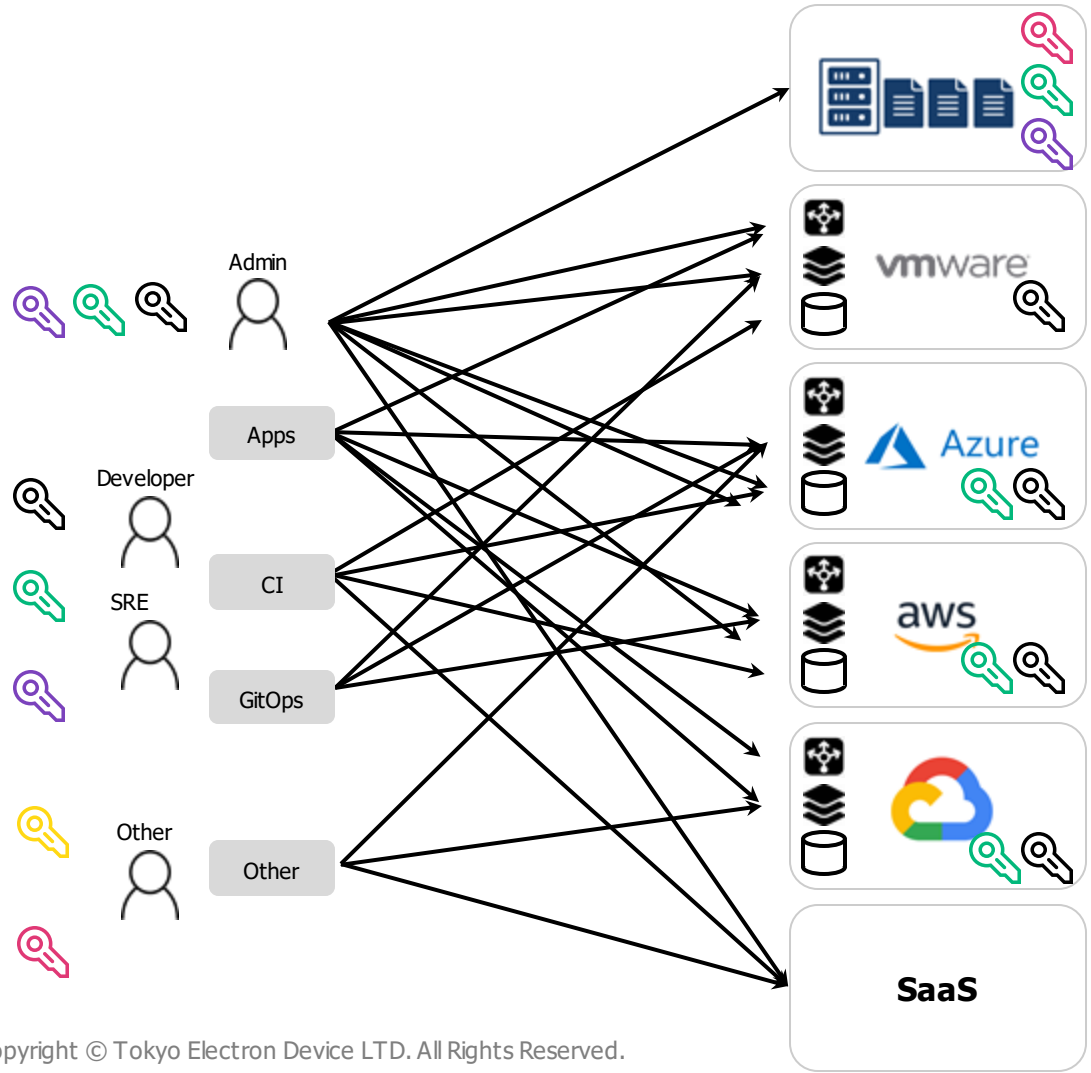
AWS	https://docs.aws.amazon.com/ja_jp/wellarchitected/latest/security-pillar/sec_identities_secrets.html
Azure	https://learn.microsoft.com/ja-jp/azure/well-architected/security/application-secrets
GCP	https://cloud.google.com/iam/docs/best-practices-for-managing-service-account-keys?hl=ja

COMPLIANCE FIRST
東京エレクトロン デバイス

シークレット管理の課題

シークレット管理の問題点

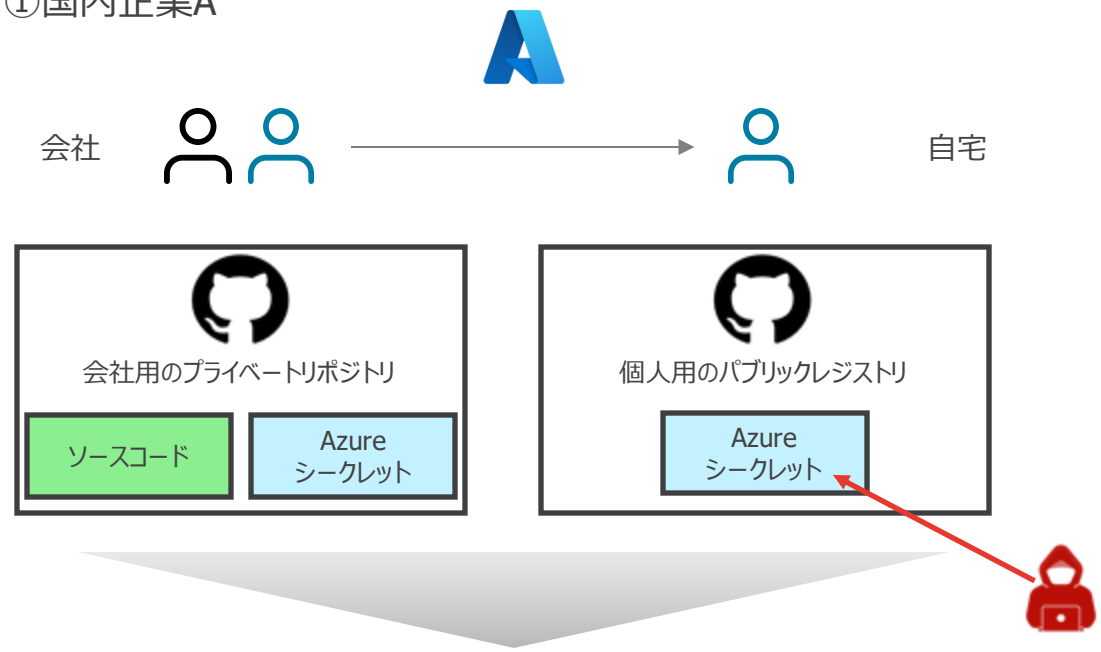
従来型のシークレットの運用管理 ＝ **シークレットの散在（Secret Sprawl）**



- 様々なシークレットが様々な場所に分散して運用管理されて**コントロール不能**
- アクセス制御が**不完全**で**データ漏えい**のリスク増大
- **長期利用**で**使い回し**によるデータ漏えいのリスク増大
- 膨大な量の**手作業**が発生することによる**生産性の低下**

シークレット漏えい事故の例

①国内企業A

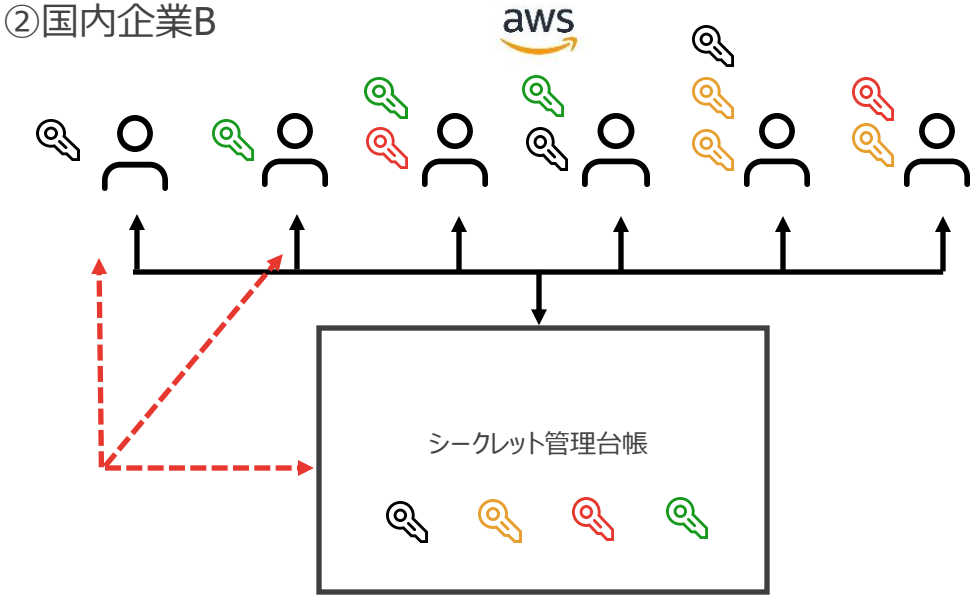


会社アカウントのAzureシークレットを自宅に
持ち帰り、誤ってパブリックリポジトリに
シークレットを公開

↓

攻撃者にシークレットが流出し
数千万の被害額

②国内企業B



社内ルール、プロセスを準拠した状態でエクセルや
メールなどでシークレットを共有、
シークレットが分散配布された状態

↓

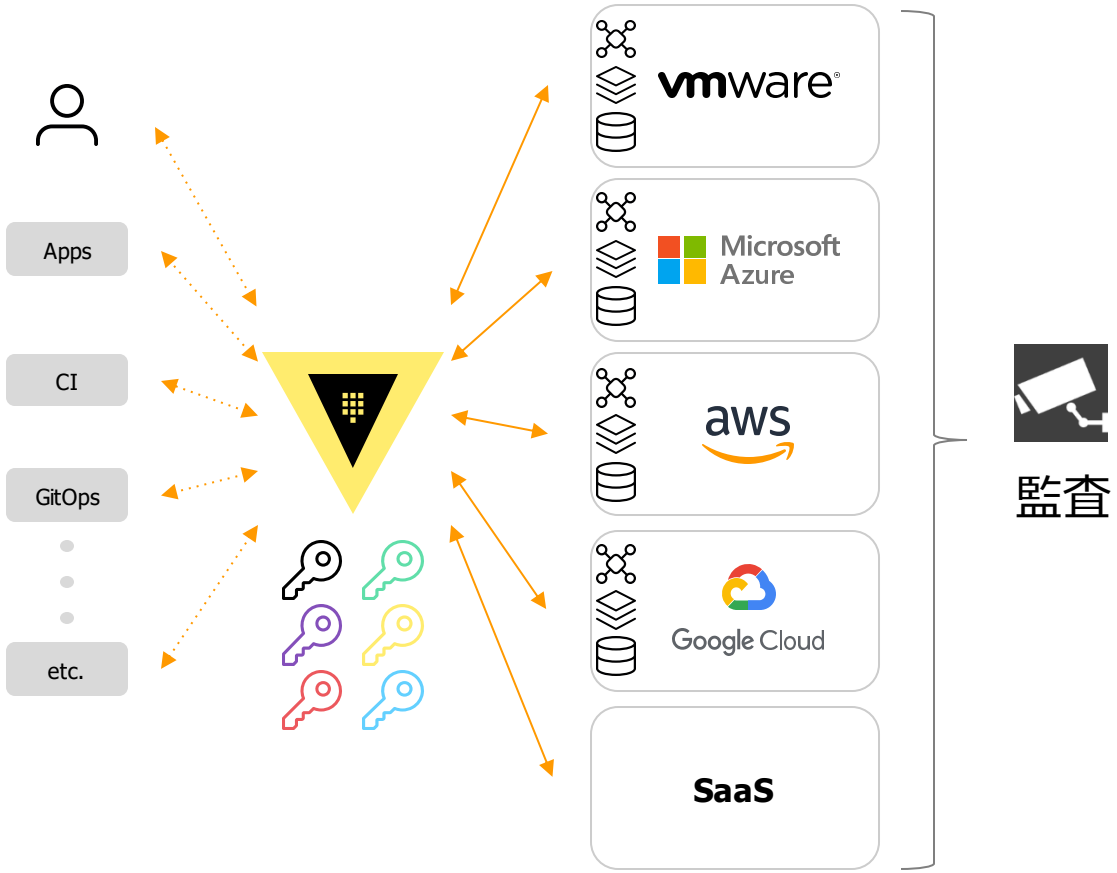
流出経路は不明であるが、
AWSにてアカウントを乱用され
数千万の被害額

HashiCorp
Vault

そこで
Vaultで課題を解決

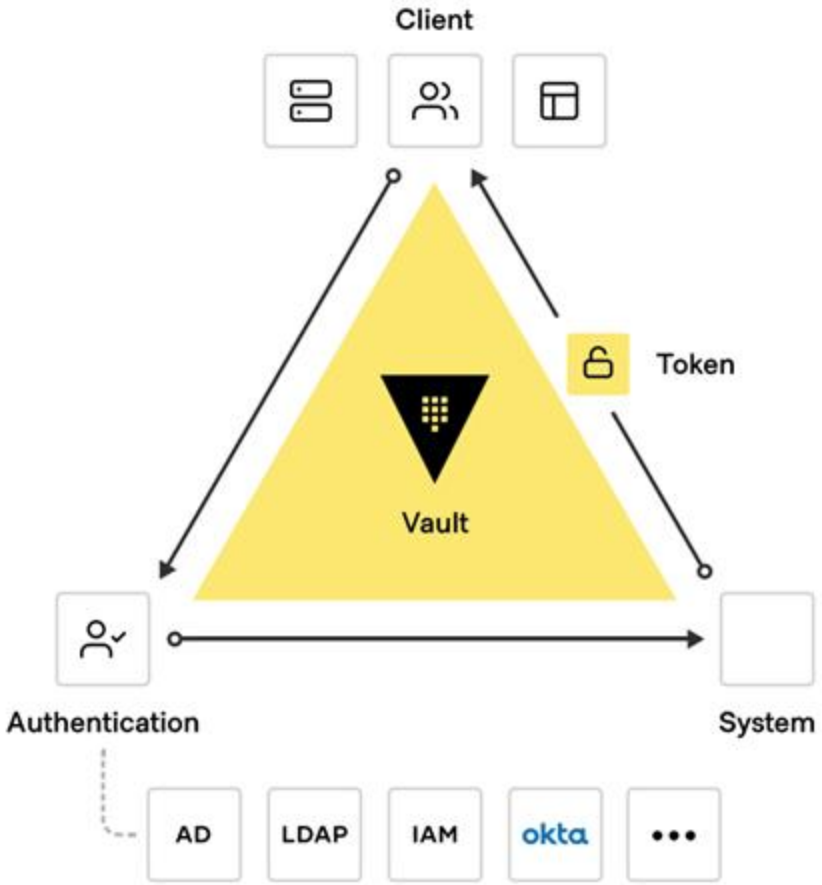
Vaultによるシークレット管理

Vaultによるシークレットの運用管理 ＝ **共通基盤でシークレットを集中的に運用管理**



- クライアント側での恣意的な管理を完全に排除し、Vaultでシークレットを**集中的に管理**
- Vaultへのアクセスは常時**認証・認可**を経由する
- **短期間**で**一意**のシークレット利用で漏えいリスクを軽減
- **ローテーションを自動化**し**生産性を向上**

Vaultとは



信頼できるアイデンティティを活用し、クラウド運用モデルにおける
シークレットやアプリケーションデータを安全に保つクラウドセキュリティ
自働化の基盤

シークレットの運用管理

クラウドやアプリケーションを跨り、インフラで必要なシークレットをGUI/ CUI / API で一元的に運用管理

データ暗号化

環境やワークロードを跨いで、アプリケーションデータを安全に保つ

- 個人情報やクレジットカード番号などの暗号化
- データベース、ストレージの暗号化

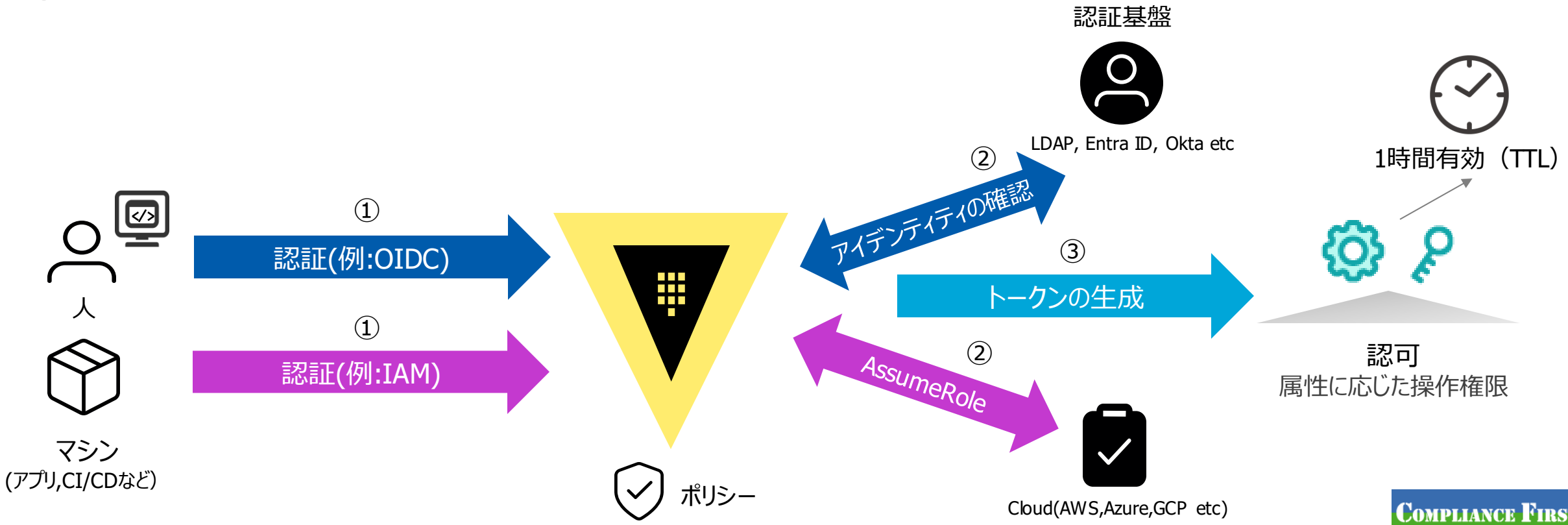
厳格なアクセス制限(1/2)

認証・認可

① クライアントはVaultを利用するために信頼された認証基盤で認証する

② 認証基盤ではクライアントのアイデンティティを確認する

③ 認証に成功するとクライアントの属性に応じたトークンが生成される

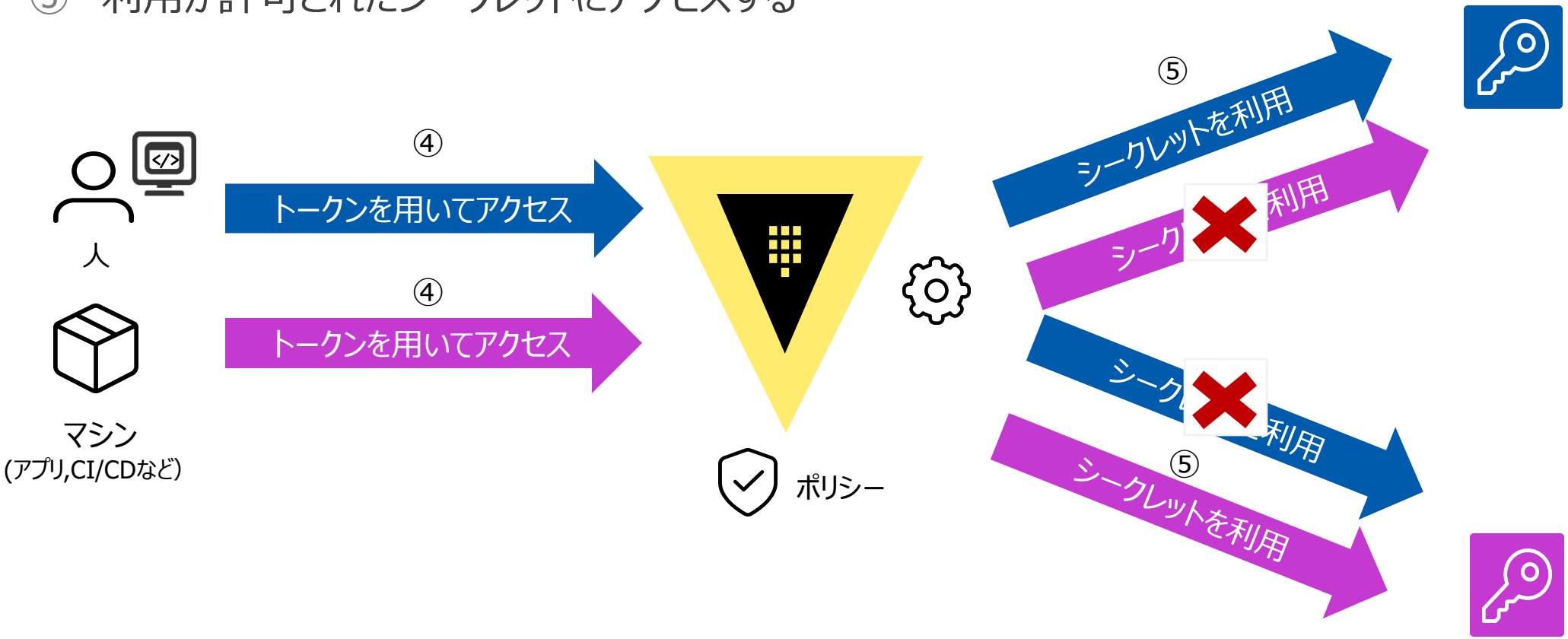


厳格なアクセス制限(2/2)

アクセス制御

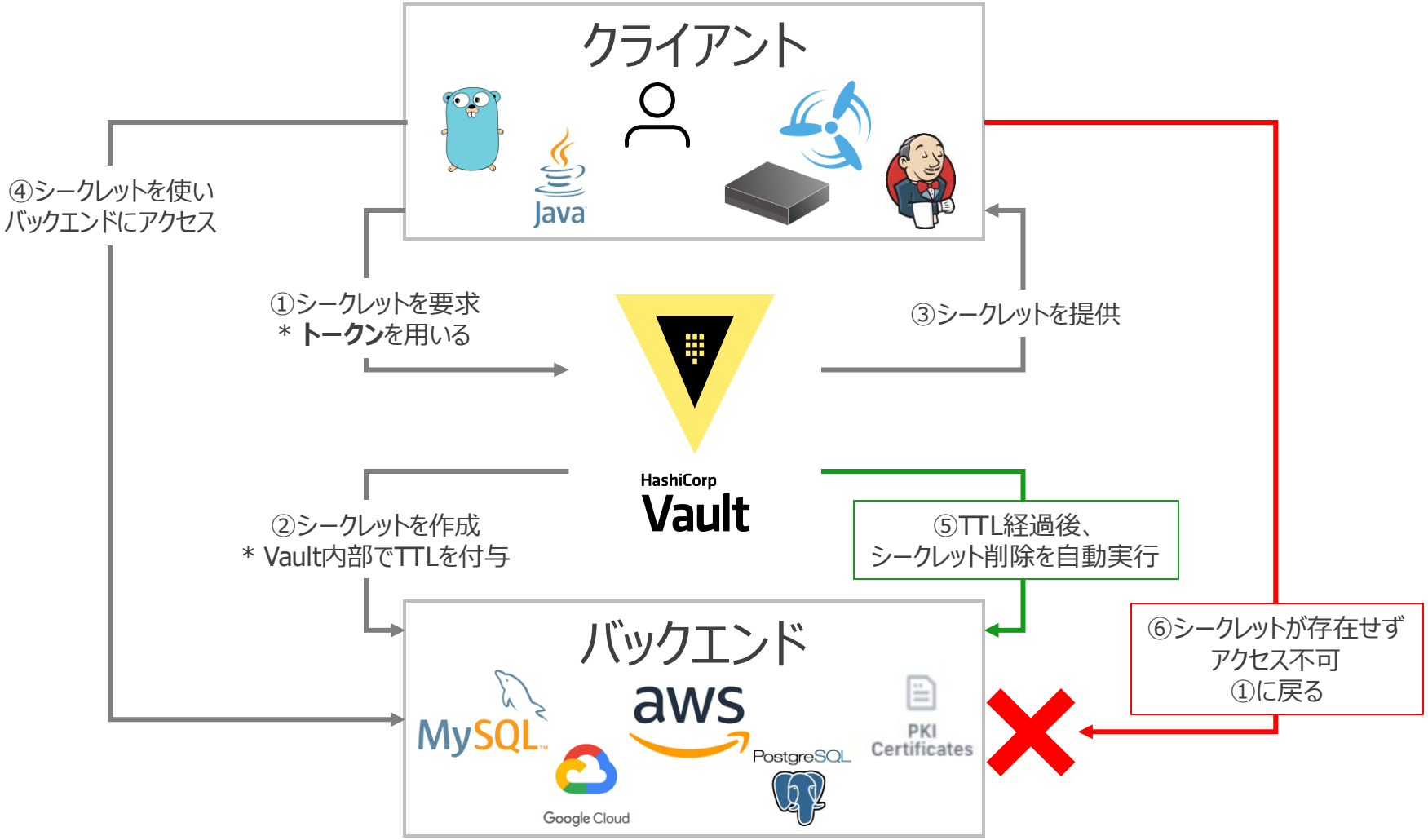
④　クライアントは認証成功時に取得したトークンを用いてVaultにアクセスする

⑤　利用が許可されたシークレットにアクセスする



短期かつ一意なシークレットの利用: 動的シークレット

長期利用、使い回しを抑制。自動ローテーションを委任できる

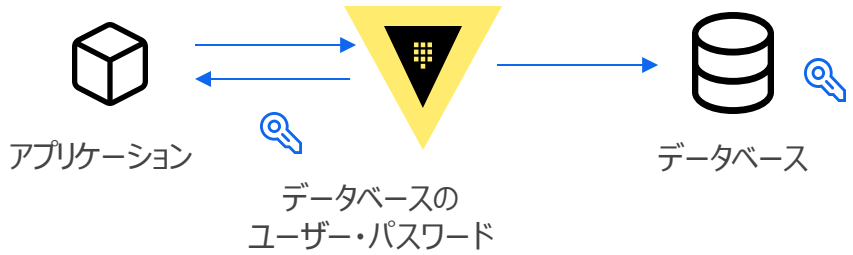


- ユニークかつ最小権限かつ有効期限(TTL)付きシークレットを必要な時のみ生成

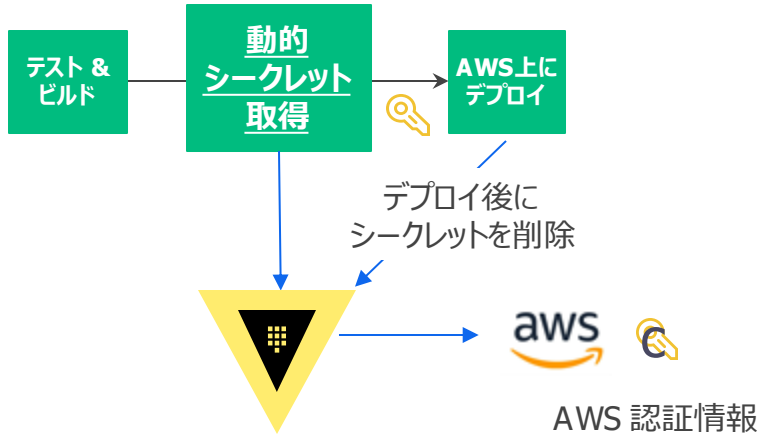
- ファイルへのシークレット記入が不要
- 各クライアントごとにユニークなシークレットを利用可能
- 一定の期間でシークレットをローテーション

動的シークレットの利用シーン

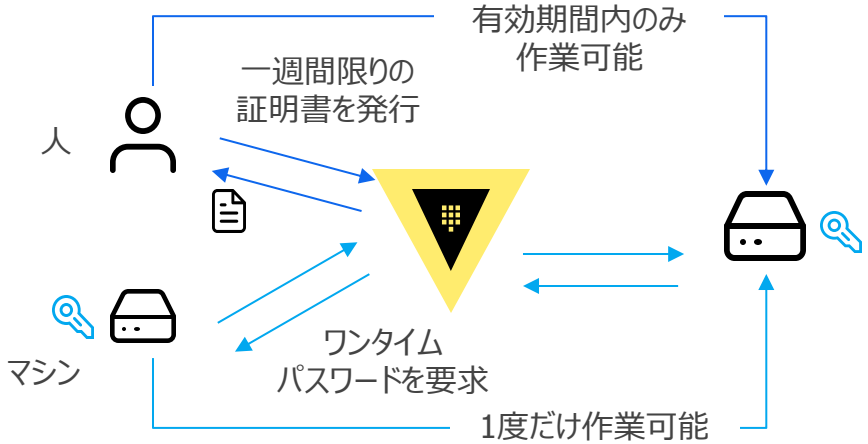
アプリケーションからの利用



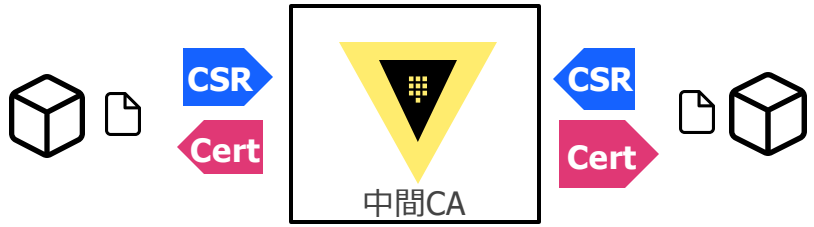
CI/CDからのデプロイ先への認証



SSH ログインでの利用



証明書管理



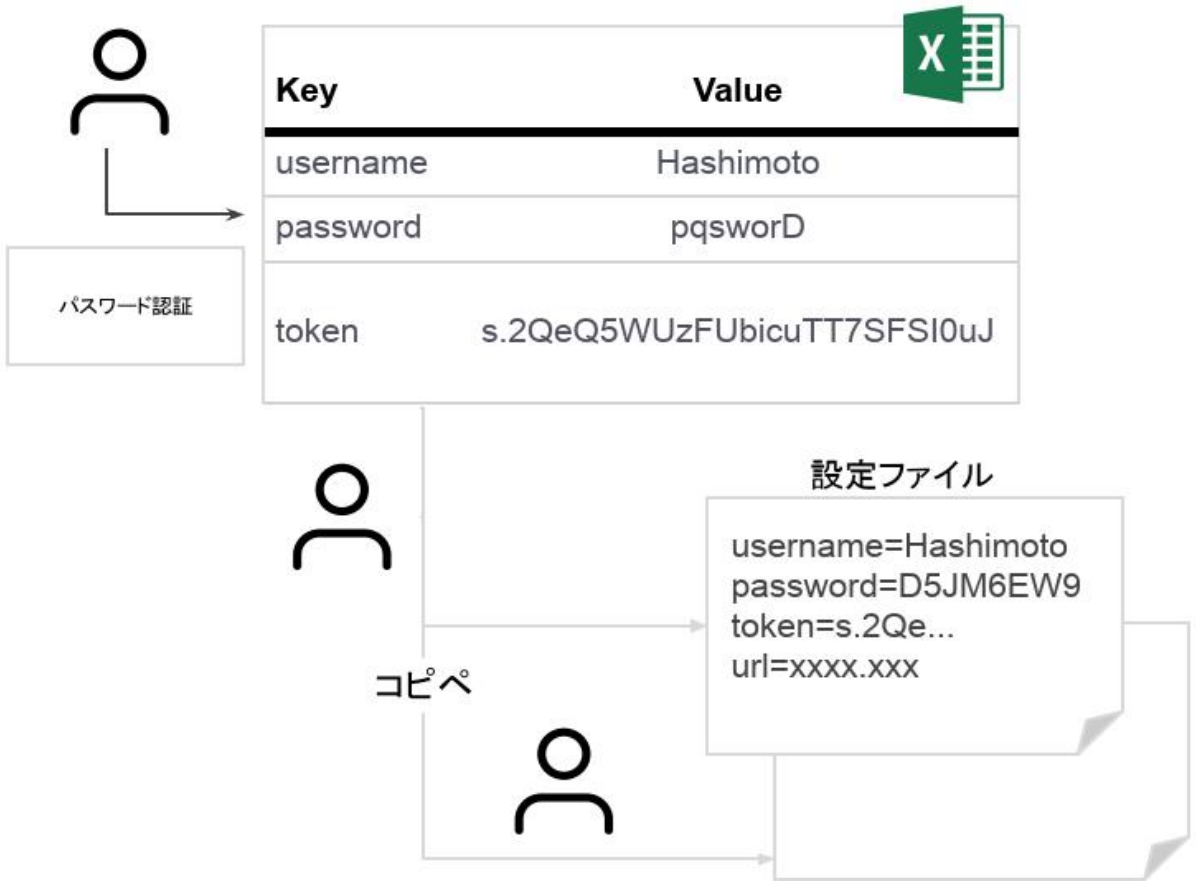
- 証明書の取得
- 証明書の破棄
- 権限管理
- 証明書の期限管理

東京エレクトロン デバイス

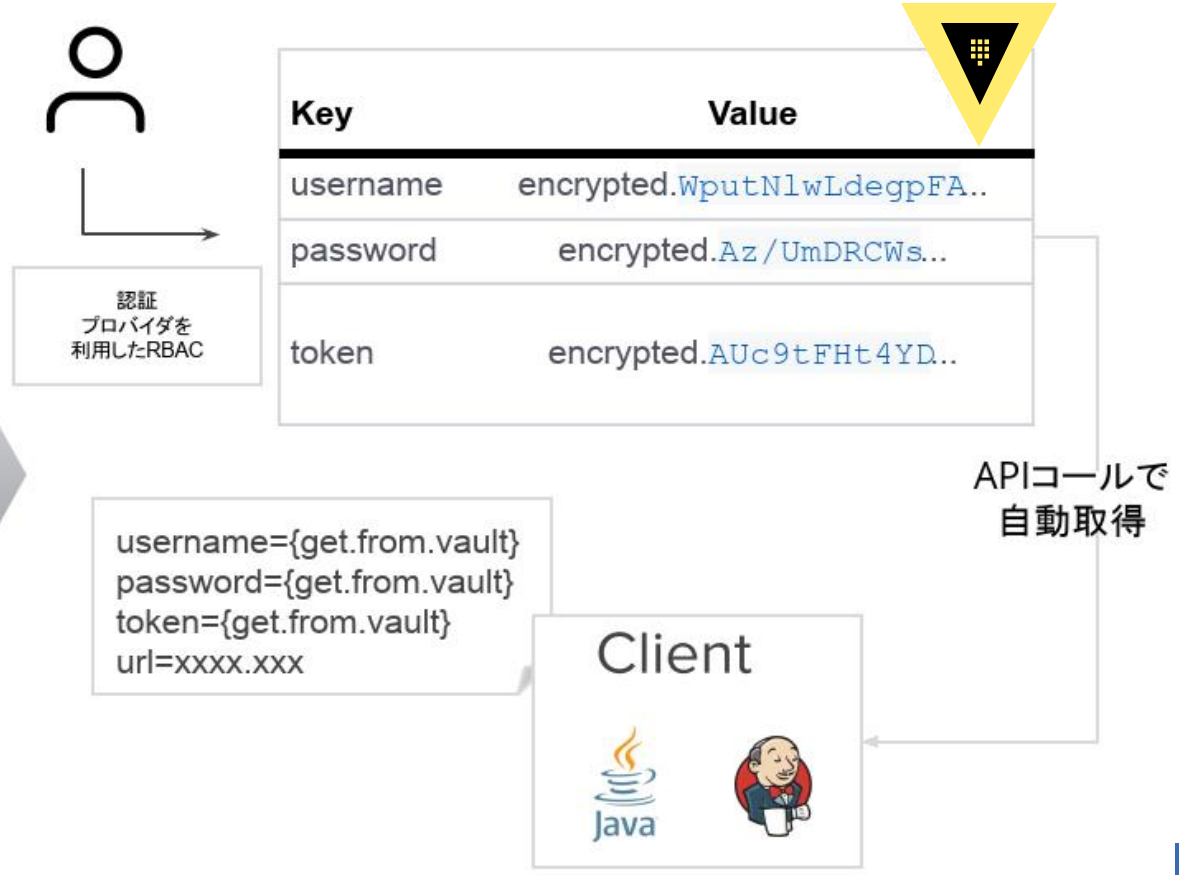
手動作成かつ長期間のシークレットの利用: 静的シークレット

キーバリュー形式によるシークレットの保存・参照。データは自動で暗号化される

従来の運用



Vaultによるシークレット管理



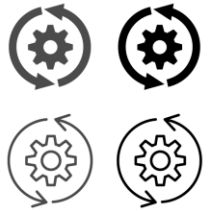
まとめ

従来型の運用

- シークレットが**広範囲**に保存されており、どこで何が起こるか分からない
- **長期利用可能**なシークレットを使い回してしまう
- アクセス制御が**不完全**

シークレット管理の手間や、
長期利用によるリスク

Vaultを使った運用



- シークレットを**集中管理**し、保存場所を狭める
- シークレットを短期間で**ローテーション**し、時間的制限を強める
- シークレットへのアクセスを常時**認証・認可**する

シークレット管理の手間を削減し、
短期利用を実現

Connect Beyond

共に創る 新たな価値を

東京エレクトロン デバイス