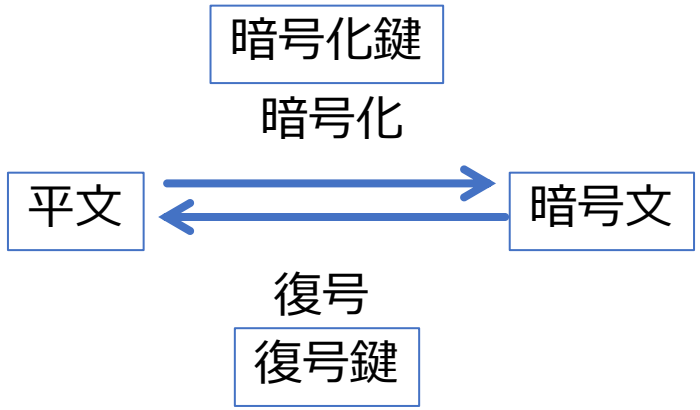
暗号技術の基本

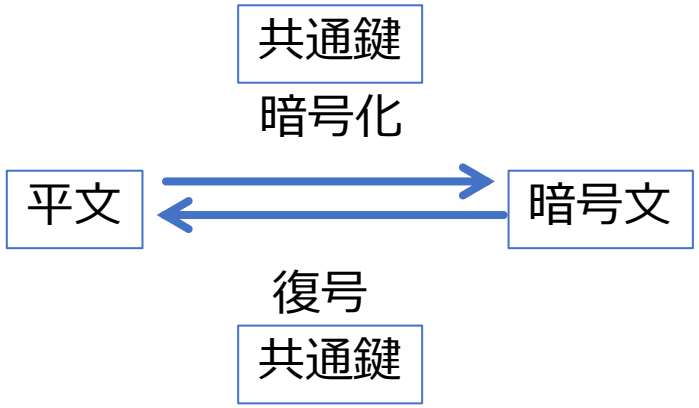
- 平文（ひらぶん）：元のデータ
- 暗号文：暗号化されたデータ
- 鍵：暗号化と復号に使用されるデータ

- 暗号化：暗号化鍵（キー）を使って平文を暗号文に変換する
- 復号：復号鍵（キー）を使って暗号文を平文に変換する



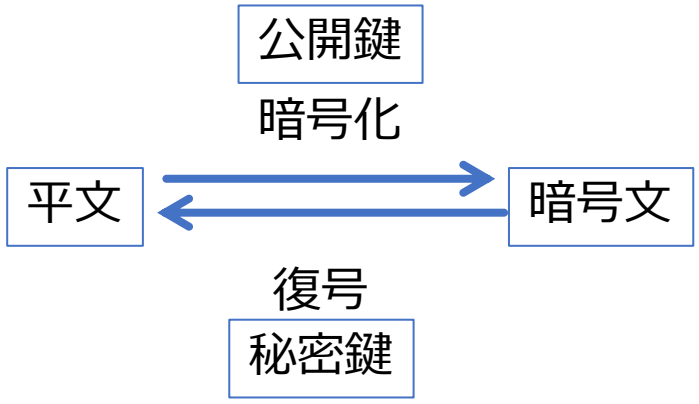
暗号技術の基本

- 対称キー：暗号化と復号の両方で使用される鍵。共通鍵とも。

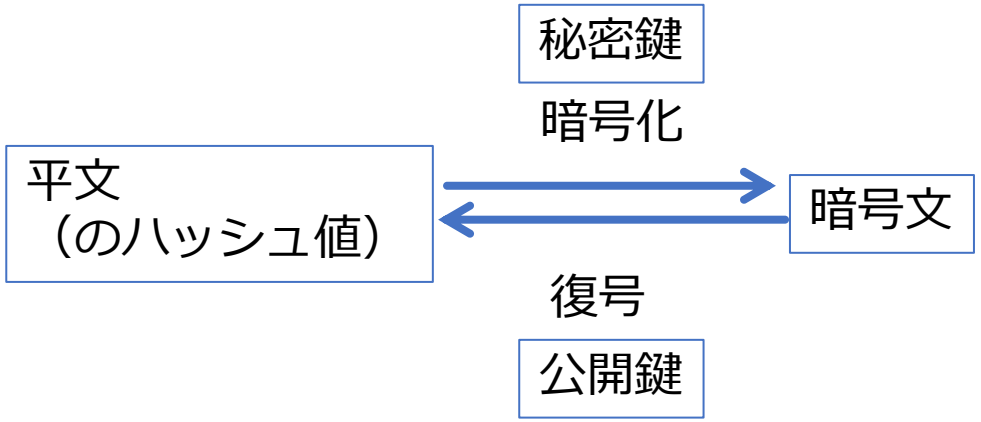


暗号技術の基本

- 非対称キー：公開鍵と秘密鍵のペア
  - 公開鍵で暗号化したデータは、ペアの秘密鍵でのみ復号できる
  - 秘密鍵で暗号化したデータは、ペアの公開鍵でのみ復号できる
  - 公開鍵は公開してよいが、秘密鍵はペアを生成した人だけが持つ

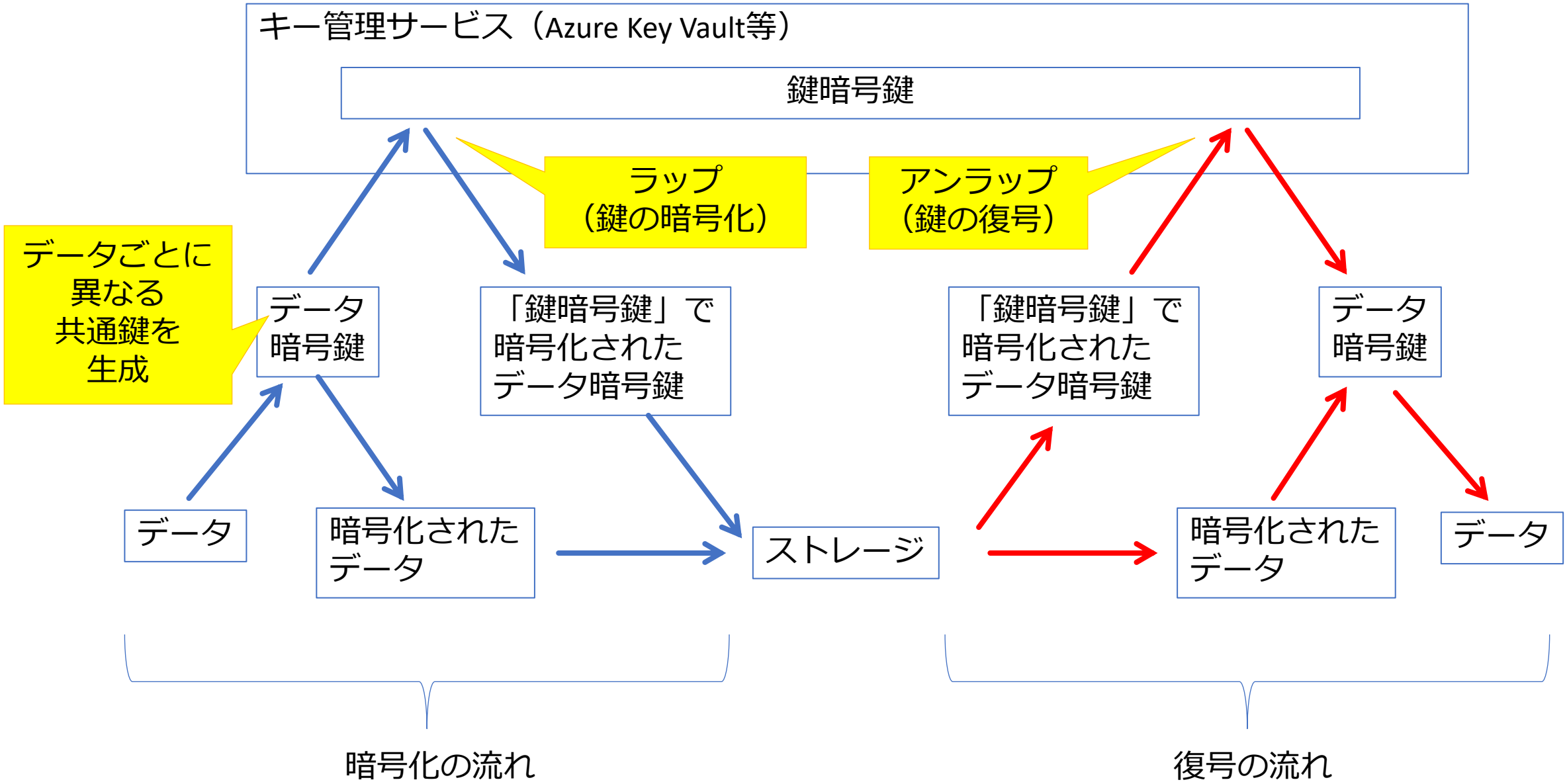


暗号化通信で利用



電子署名・改ざん検出で利用

エンベロープ暗号化



[エンベロープ暗号化 | Cloud KMS ドキュメント | Google Cloud]

エンベロープ暗号化の特徴

危殆化（きたいか）
危うくなること。
鍵が漏洩すること。

- データごとに別の「データ暗号化鍵」を使用する
  →危殆化が発生した際の影響範囲が小さい

- データの暗号化と復号は対称キー（共通鍵）で実行される
  →一般に対称キーのほうが非対称キーよりも計算量が少なくてすむ

- ラップ・アンラップはキー管理サービスで実行される
  →適切な権限をもったユーザーのみサービスを利用できる。
  サービスの利用履歴が残せる。

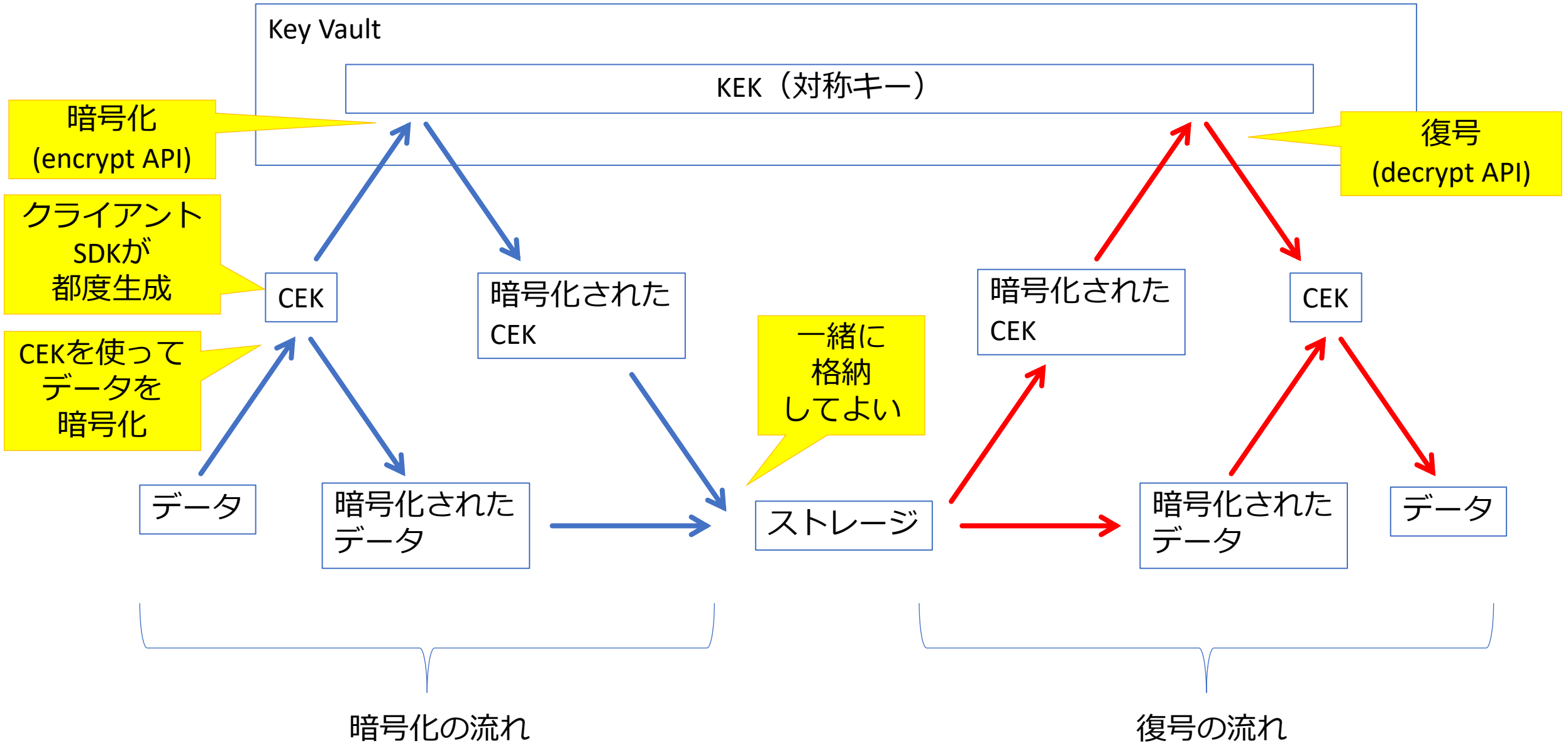
Key Vaultによるクライアント側暗号化

- Azure ストレージ クライアント SDK
  - 暗号化の処理を実行する。
  - 必要な場合はKey VaultのAPIを呼び出す
- CEK（Content Encryption Key）
  - データを暗号化するための鍵
  - Azure ストレージ クライアント SDKが、データごとに生成する。
- KEK（Key Encryption Key）
  - CEKを暗号化・復号するための鍵
  - **対称キー** または **非対称キー**
  - Key Vault内に事前に生成/インポートしておく

※ Azureストレージ クライアントSDKは、キーのラップ・ラップ解除にKey Vaultを使用することも、カスタムプロバイダーを使用することもできる。カスタム プロバイダーを使用する場合は、KEKはKey Vaultの中ではなくカスタム プロバイダー内に置かれることになる。

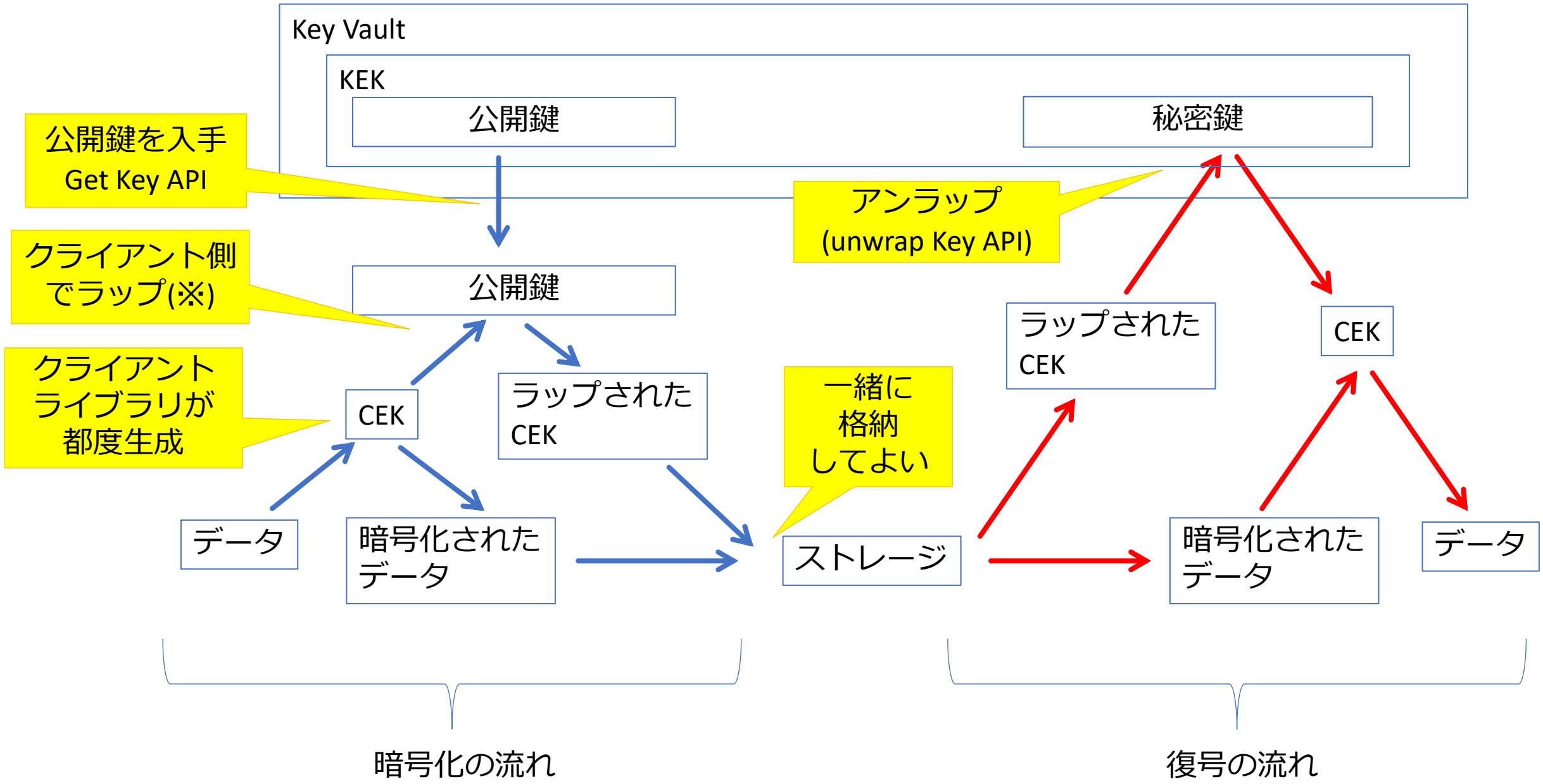
KEKが**対称キー**の場合

※ KEK（対称キー）は常にKey Vaultの中に留まる



KEKが**非対称キー**（公開鍵＋秘密鍵）の場合

※ wrap KeyというAPIも用意されており、
公開鍵へのアクセスがない場合などに使用できる



Azure Key Vaultのエンベロープ暗号化の特徴

- KEKは対称キーまたは非対称キーである

Azure Key VaultのAPIとしては
「キー暗号化」「キー復号」のAPIと
「ラップ」「アンラップ」のAPIは
別のものです。

- KEKが対称キーの場合
  - 暗号化時：**キー暗号化(encrypt key)API**を使用してCEKを暗号化
  - 復号時：**キー復号(decrypt key)API**を使用してCEKを復号

- KEKが非対称キーの場合
  - 暗号化時：**キー入手(get key)API**と**ローカルの処理**によってCEKをラップ
  - 復号時：**アンラップ(unwrap key)API**を使用してCEKをアンラップ