静から動へ：Eventarc ではじめるイベント駆動アーキテクチャ

Google Cloud Next Tokyo

宇都宮 雅彦

Executive IT Specialist, 株式会社NTTデータ

Google Developer Expert (Cloud)

イベント駆動アーキテクチャとは

発生した出来事（イベント）をきっかけにサービスが反応　し、各処理が**非同期に進む**ことで全体のビジネスロジックが構成されるアーキテクチャ

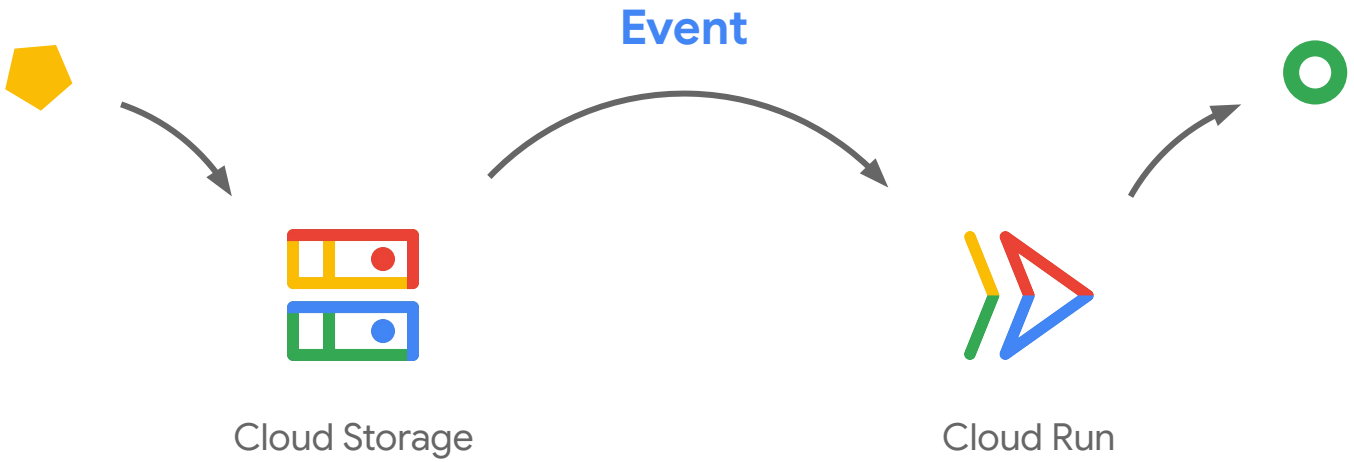
- ✔ 出来事に応じて動く「待ち受け型」のアーキテクチャ
- ✔ 非同期を原則とすることで「疎結合」な構造にしやすい
- ✔ イベント駆動を含むより広い概念に「リアクティブアーキテクチャ」がある

イベント駆動アーキテクチャとは

発生した出来事（イベント）をきっかけにサービスが反応　し、各処理が**非同期に進む**ことで全体のビジネスロジックが構成されるアーキテクチャ

- ✔ 出来事に応じて動く「待ち受け型」のアーキテクチャ
- ✔ 非同期を原則とすることで「疎結合」な構造にしやすい
- ✔ イベント駆動を含むより広い概念に「リアクティブアーキテクチャ」がある

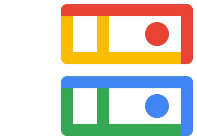

Event
Firestore
Workflows


Event
Cloud Storage
Cloud Run

どんな
イベントを出す？

Event

どうやって
イベントを出す？



Cloud Storage

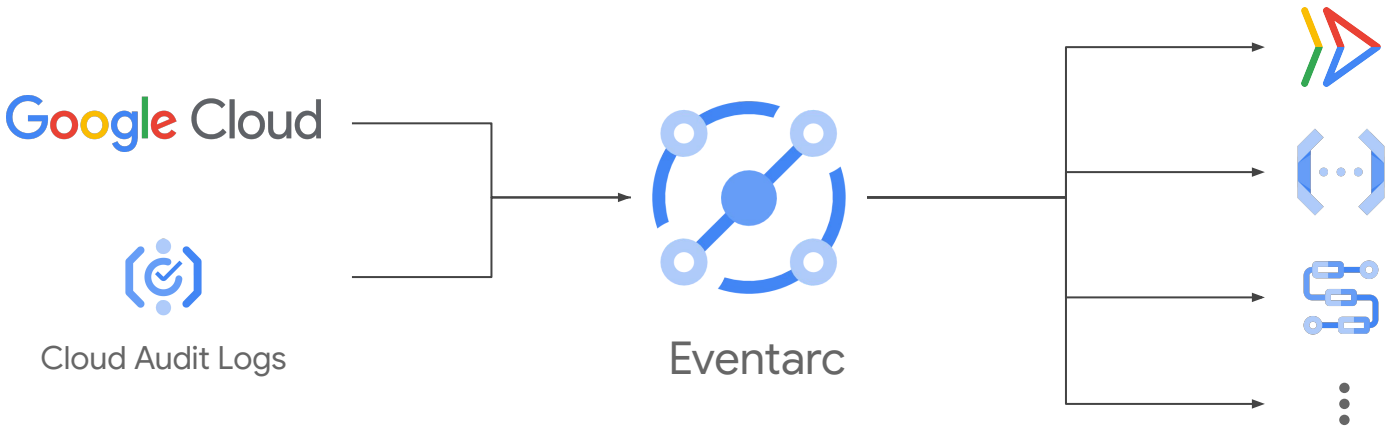
Cloud Run

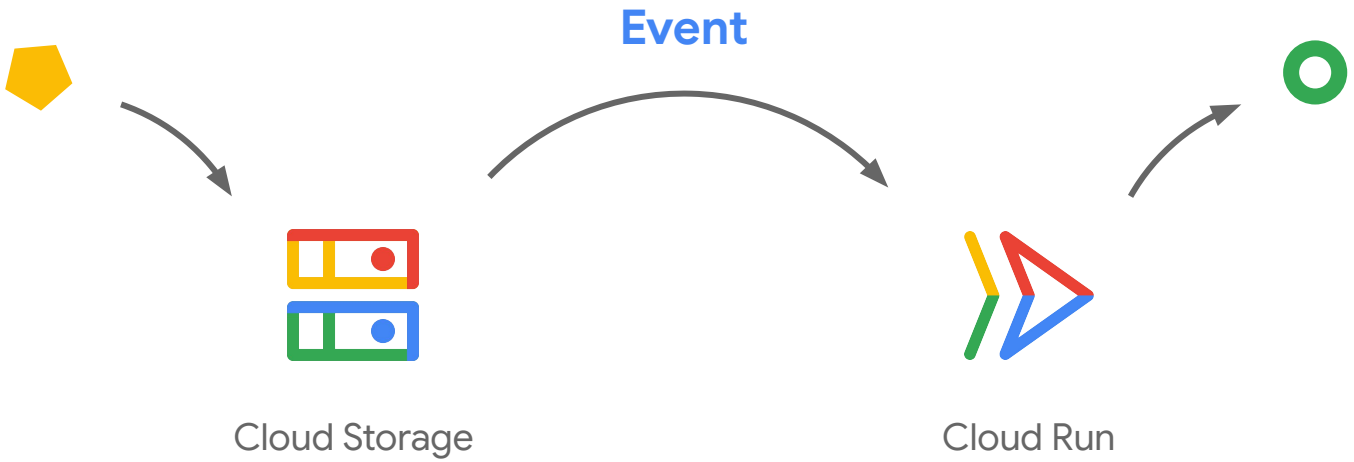
受け取れ
なかったら？

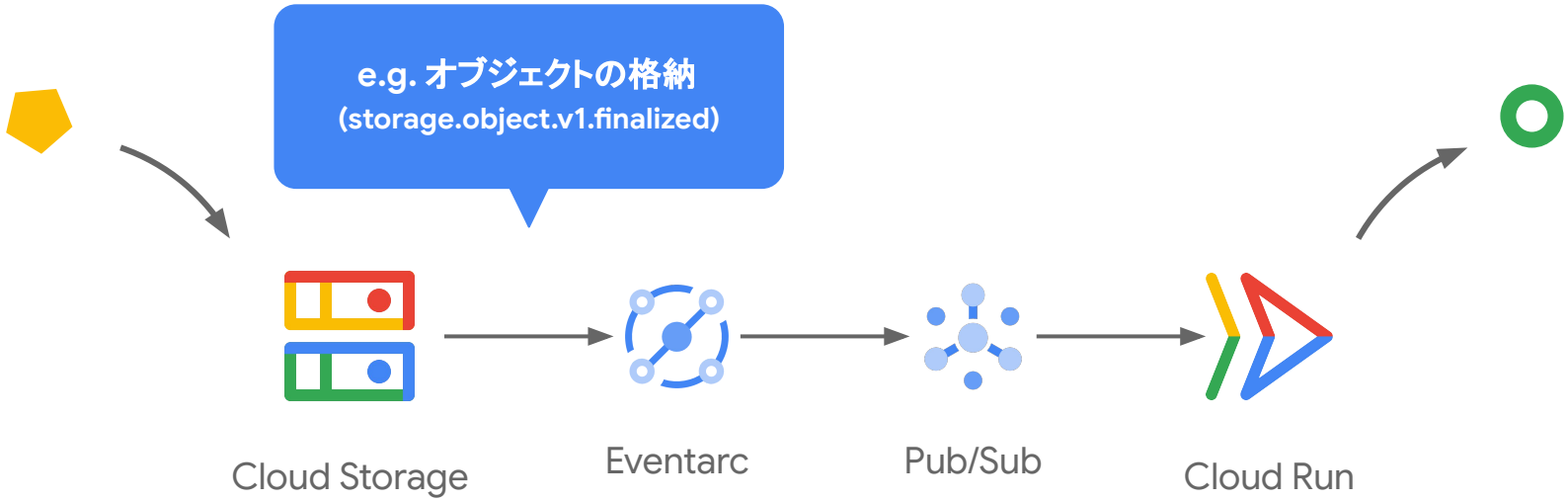
Eventarc

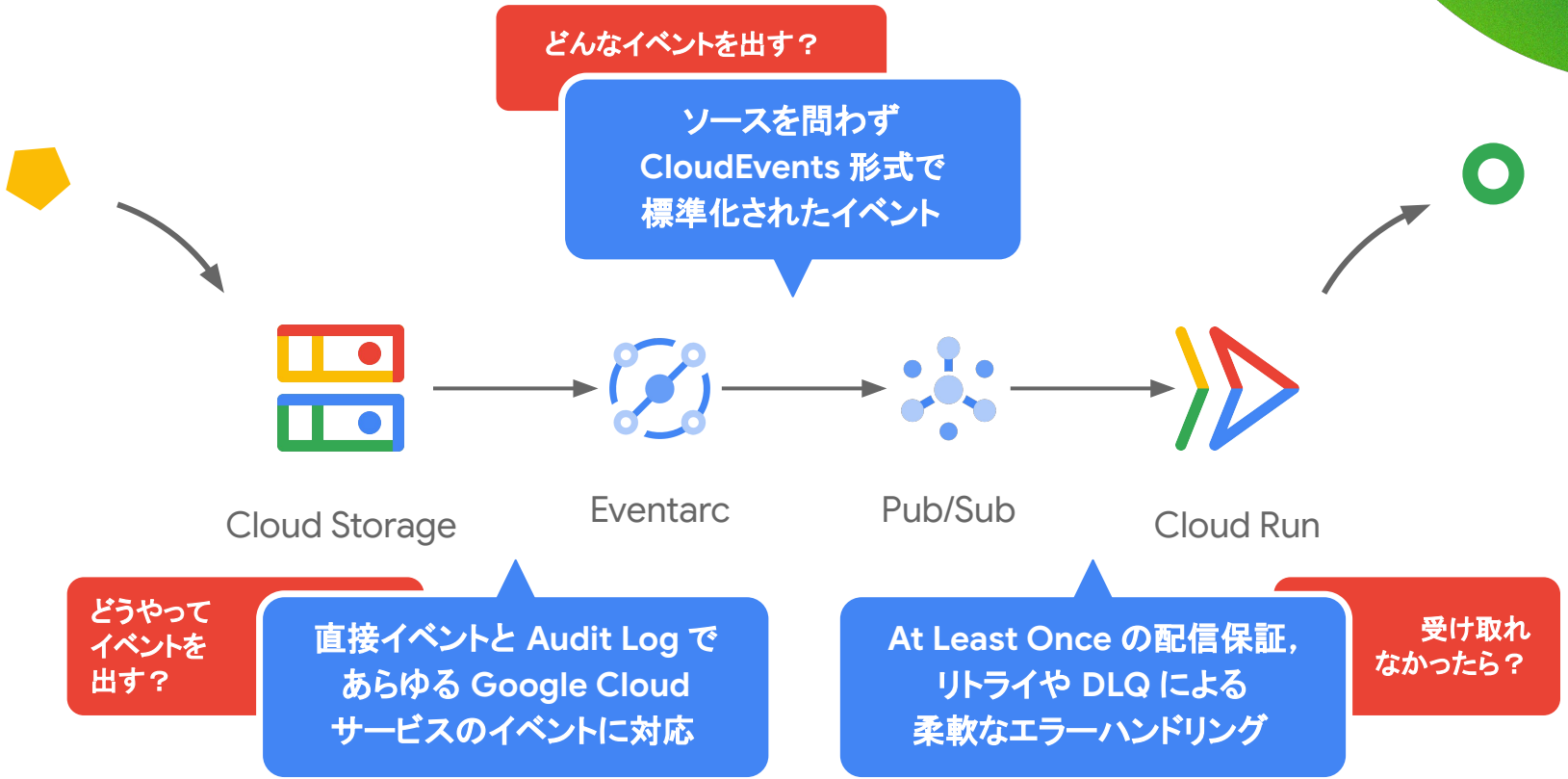
対応サービスの直接イベントや Audit Log（監査ログ）を

CloudEvents 形式に変換し、**イベントソースとターゲットをつなぐ**



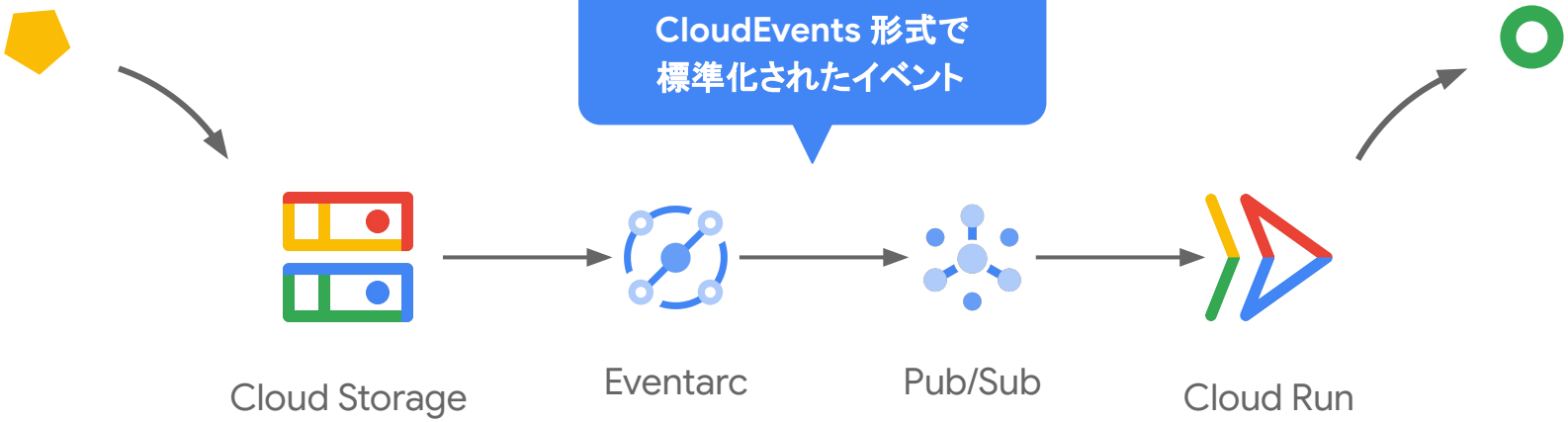

Event
Cloud Storage
Cloud Run


e.g. オブジェクトの格納
(storage.object.v1.finalized)
Cloud Storage
Eventarc
Pub/Sub
Cloud Run


どんなイベントを出す？
ソースを問わず
CloudEvents 形式で
標準化されたイベント
Cloud Storage
Eventarc
Pub/Sub
Cloud Run
どうやって
イベントを
出す？
直接イベントと Audit Log で
あらゆる Google Cloud
サービスのイベントに対応
At Least Once の配信保証，
リトライや DLQ による
柔軟なエラーハンドリング
受け取れ
なかったら？

どんなイベントを出す？

ソースを問わず
CloudEvents 形式で
標準化されたイベント

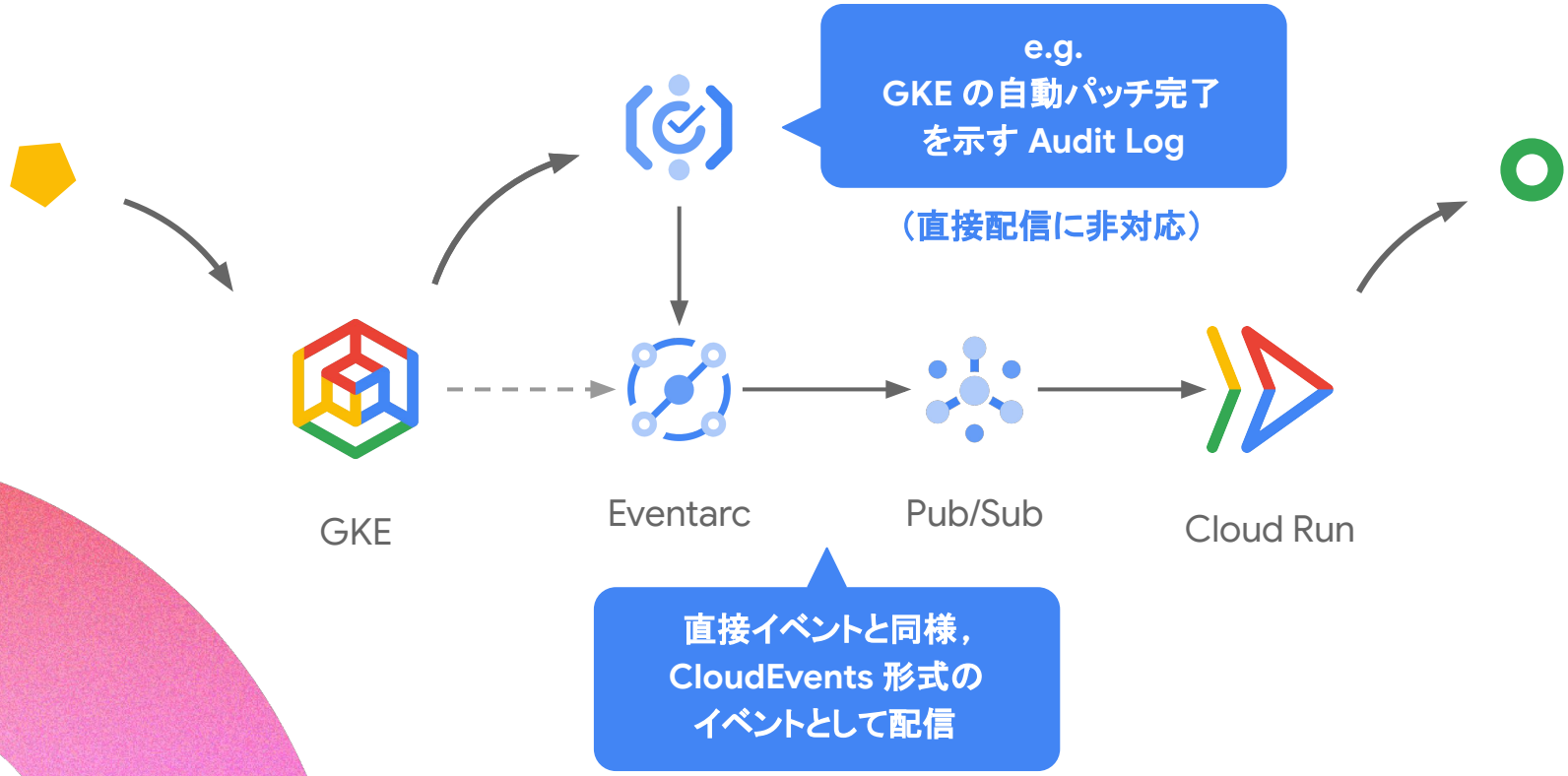


どうやって
イベントを
出す？

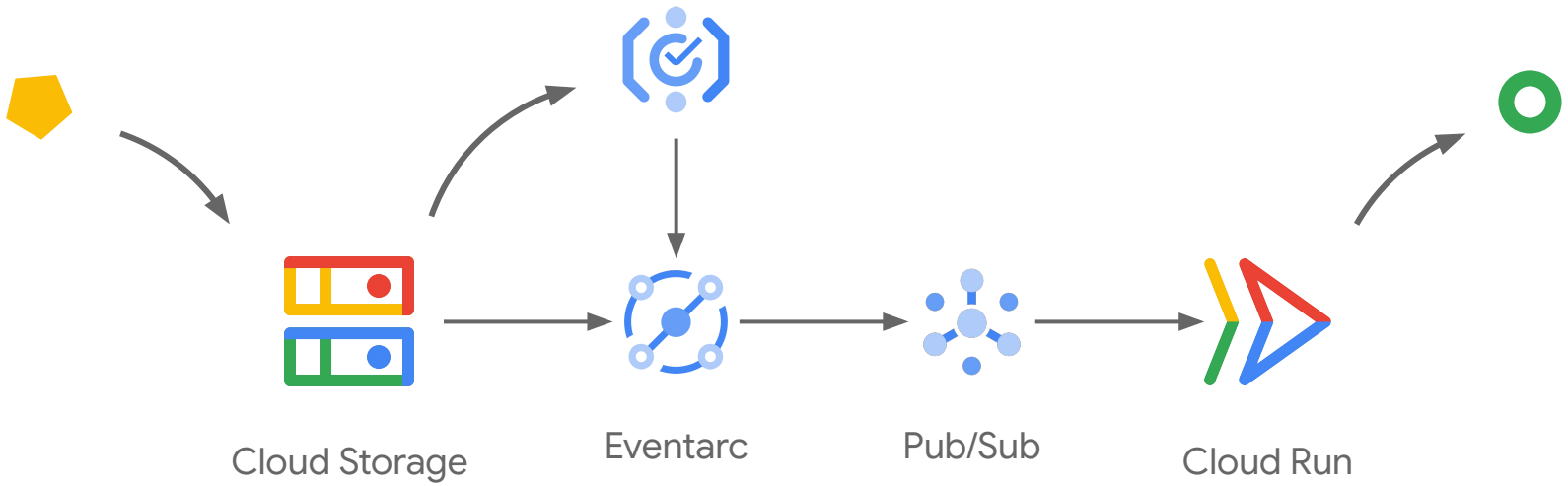
直接イベントと Audit Log で
あらゆる Google Cloud
サービスのイベントに対応

At Least Once の配信保証，
リトライや DLQ による
柔軟なエラーハンドリング

受け取れ
なかったら？

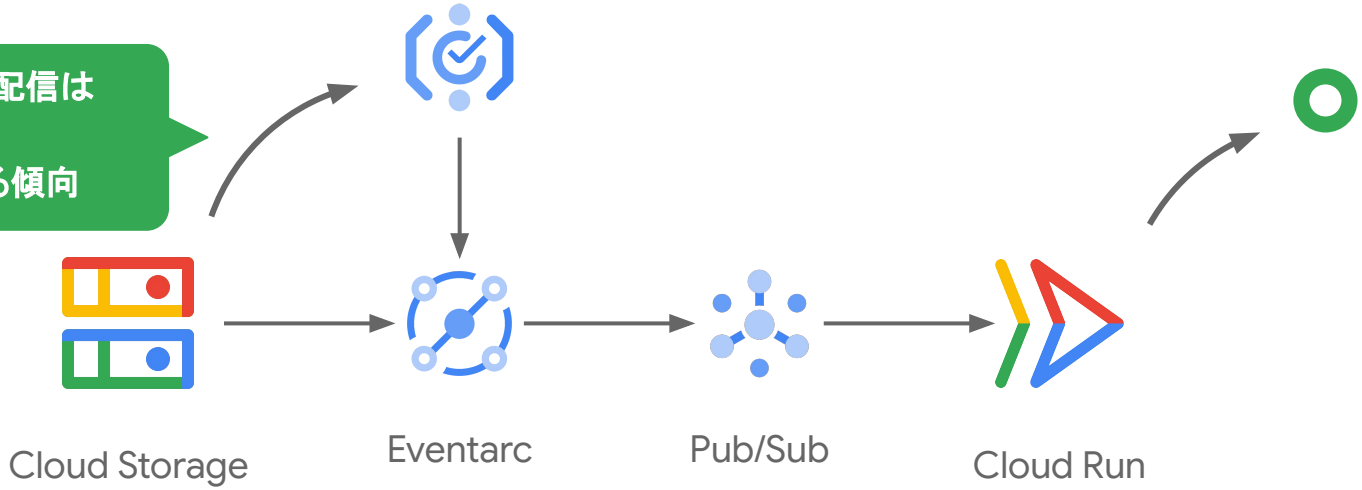

e.g.
GKE の自動パッチ完了
を示す Audit Log
（直接配信に非対応）
GKE
Eventarc
Pub/Sub
Cloud Run
直接イベントと同様，
CloudEvents 形式の
イベントとして配信

設計時の注意点



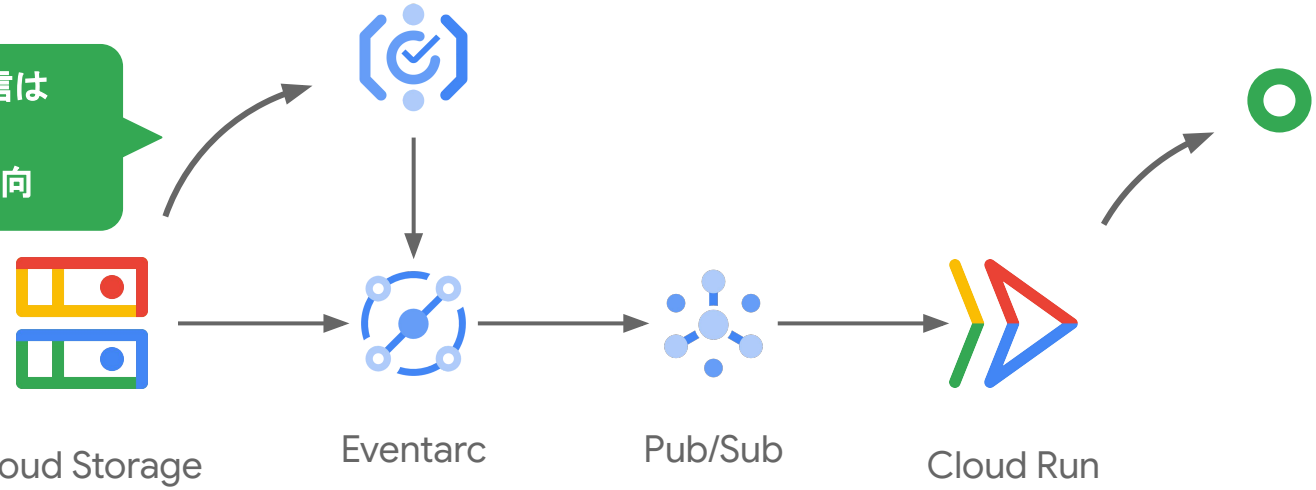
設計時の注意点

1. Audit Log による配信は
   直接イベントよりも
   タイムラグが増える傾向



設計時の注意点

1. Audit Log による配信は
直接イベントよりも
タイムラグが増える傾向

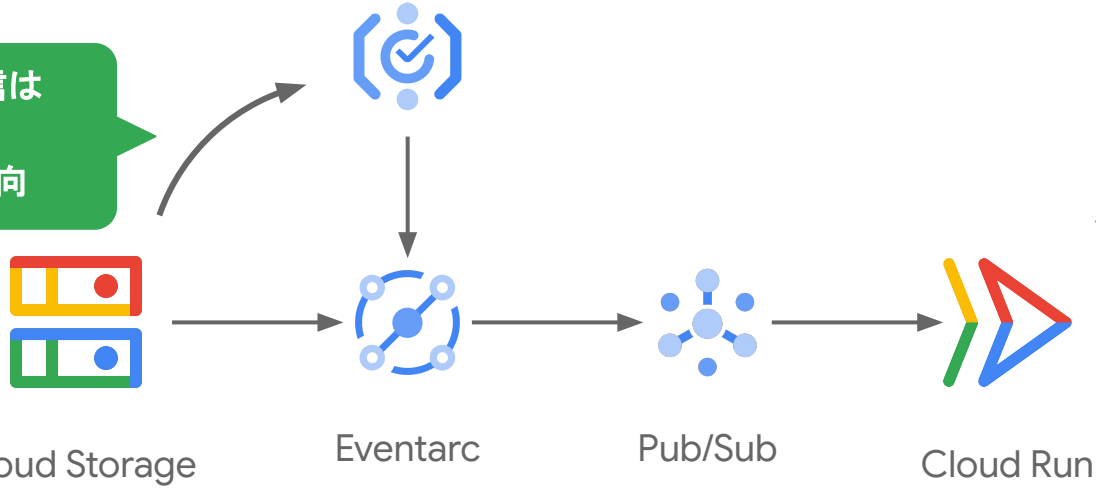


2. ソースからのイベント発行は
多くの場合 At Least Once

3. ターゲットへのイベント配信も
多くの場合 At Least Once

設計時の注意点

1. Audit Log による配信は
直接イベントよりも
タイムラグが増える傾向

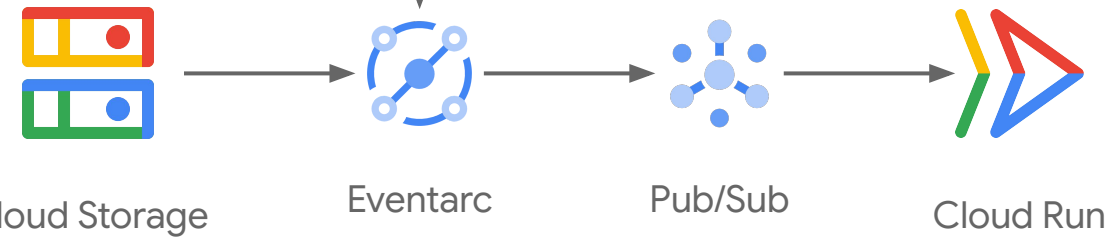


4. 冪等性への
考慮は
引き続き必要

2. ソースからのイベント発行は
多くの場合 At Least Once

3. ターゲットへのイベント配信も
多くの場合 At Least Once

設計時の注意点



1. Audit Log による配信は直接イベントよりもタイムラグが増える傾向

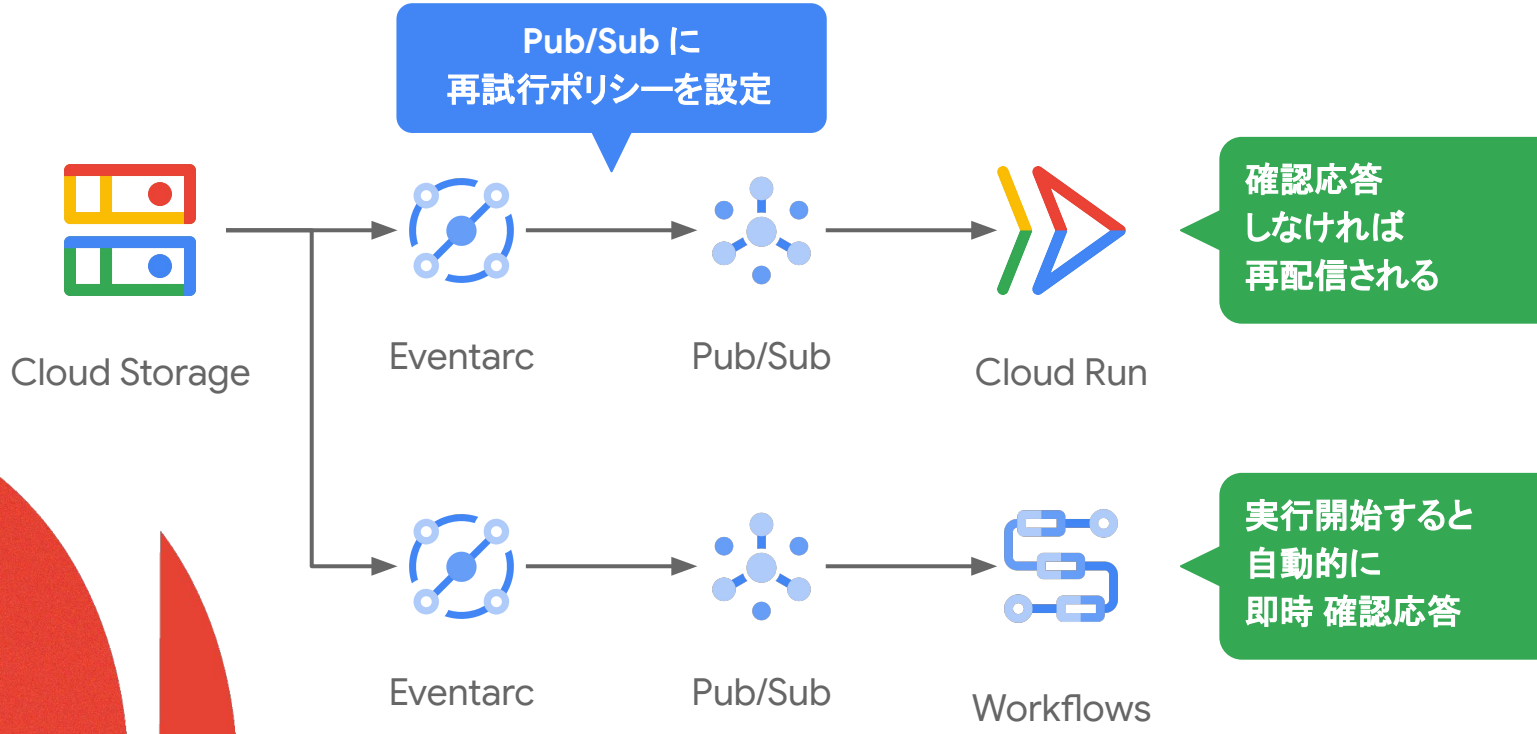
2. ソースからのイベント発行は多くの場合 At Least Once

3. ターゲットへのイベント配信も多くの場合 At Least Once

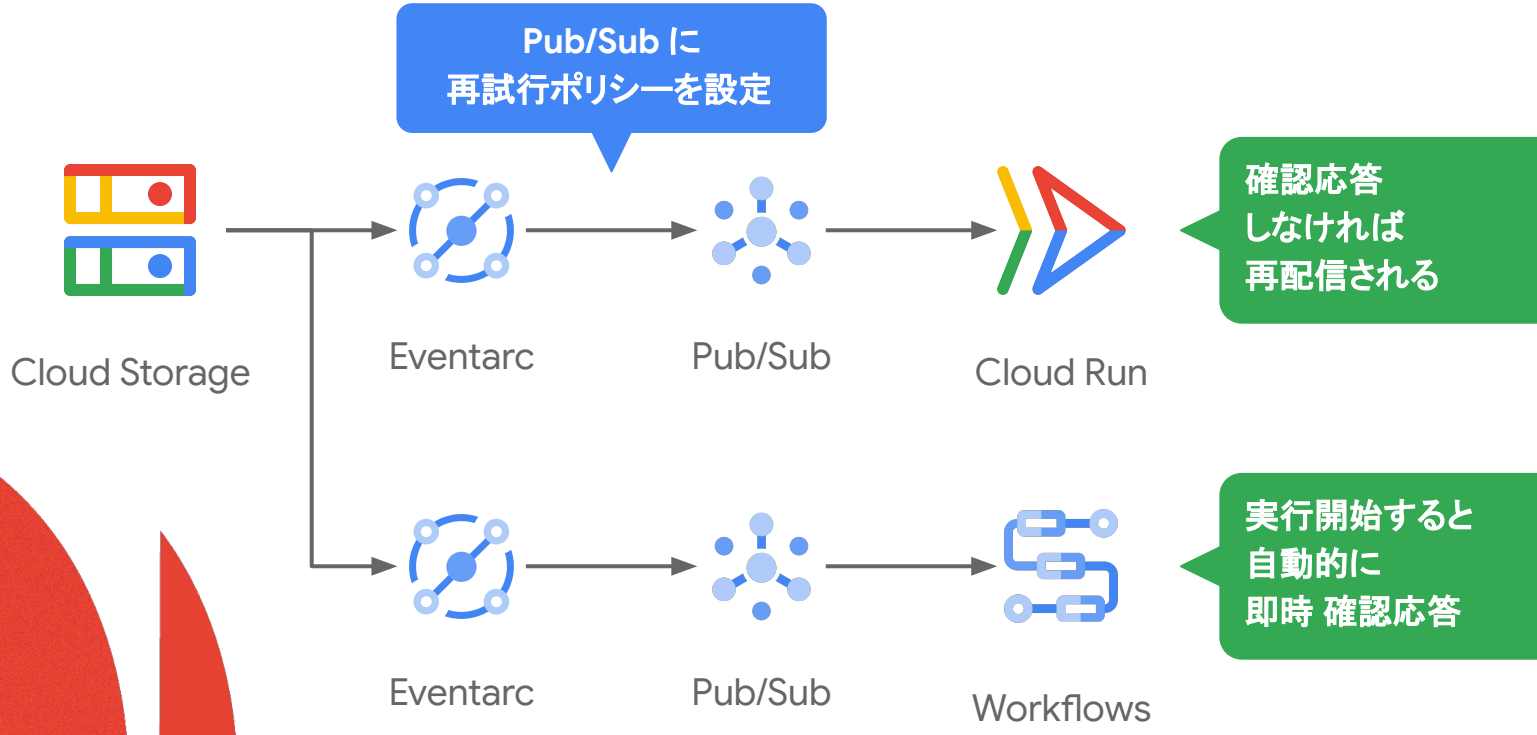
4. 冪等性への考慮は引き続き必要

5. 確認応答のタイミングはターゲットによって異なる

設計時の注意点（確認応答と再試行）



設計時の注意点（確認応答と再試行）



まとめ

Eventarc の最大の特徴は、

イベントを設計せずともイベント駆動を始められる　こと

配信メカニズムとイベント形式を標準化することで、
イベント駆動アーキテクチャ設計時の**ベースラインを容易に確立**できる

→ ユーザにとって、より付加価値のある領域に注力できる

まとめ

Eventarc の最大の特徴は、

イベントを設計せずともイベント駆動を始められる　こと

配信メカニズムとイベント形式を標準化することで、
イベント駆動アーキテクチャ設計時の**ベースラインを容易に確立**できる

→ ユーザにとって、より付加価値のある領域に注力できる

THANK YOU