



任天堂プラットフォーム向け 汎用ゲームサーバー

河原 太介 田中 翔也

任天堂株式会社 技術開発部

スピーカー自己紹介



河原 太介

技術開発部

入社 17 年目

担当: サーバーアプリケーション開発

これまで: データ配信系システム、プッシュ通知システム など



田中 翔也

技術開発部

入社 7 年目

担当: サーバーアプリケーション開発

これまで: ゲームニュース配信システムなど

目次

- 汎用ゲームサーバーの紹介
- 設計コンセプト
- アーキテクチャ
- Google Cloud の利用に際しての課題とその対策



汎用ゲームサーバーの紹介

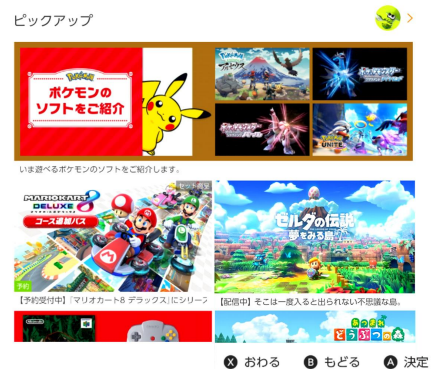
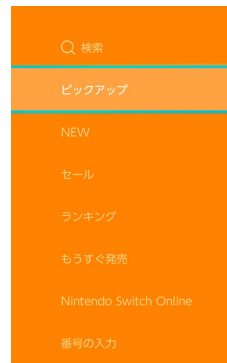
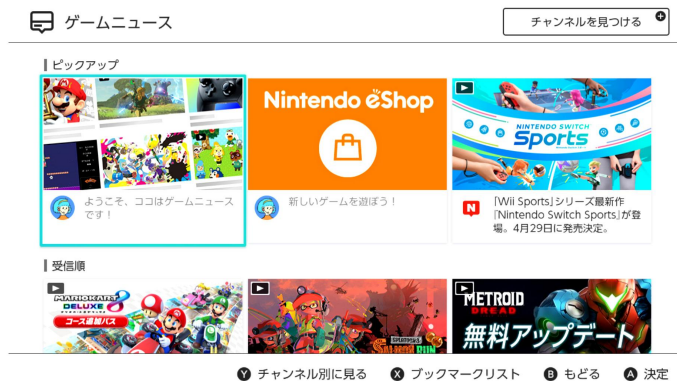
Nintendo Switch™

全世界累計 1 億 354 万台を販売
(2021 年 12 月時点)



Nintendo Switch のオンライン機能

- フレンド
- ニンテンドー e ショップ
- ゲームニュース
- プッシュ通知
- ゲーム内のオンラインマルチプレイ機能
- など



任天堂プラットフォーム向け 汎用ゲームサーバーとは

オンラインゲームの共通機能のためのサーバー

- ユーザ認証・認可
- フレンド
- マッチメイク
- リーダーボード
- メッセージング
- データ交換
- など



NPLN : 汎用ゲームサーバーの再設計プロジェクト

従来システム

- これまでのシステムはニンテンドー 3DS の時代に設計

環境の変化

- 利用タイトルの増加
- 接続するお客様の増加
- 継続的な開発に伴う機能の増加
- 内部開発者の増加
- 新しい技術の普及

NPLN

- 環境変化に対応するため、今までのノウハウを生かしつつ再設計しスクラッチで開発
- 2018 年にプロジェクト始動
- 2021 年より運用を開始



設計コンセプト

NPLN ではこの二つの実現を目標として再設計

マイクロサービス指向

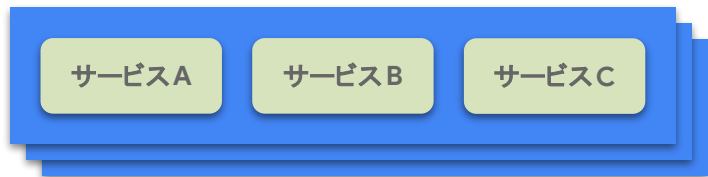
- 開発の大規模化に対応
 - サービスや機能の増加
 - 内部の開発者の増加

マルチテナント構成のサポート

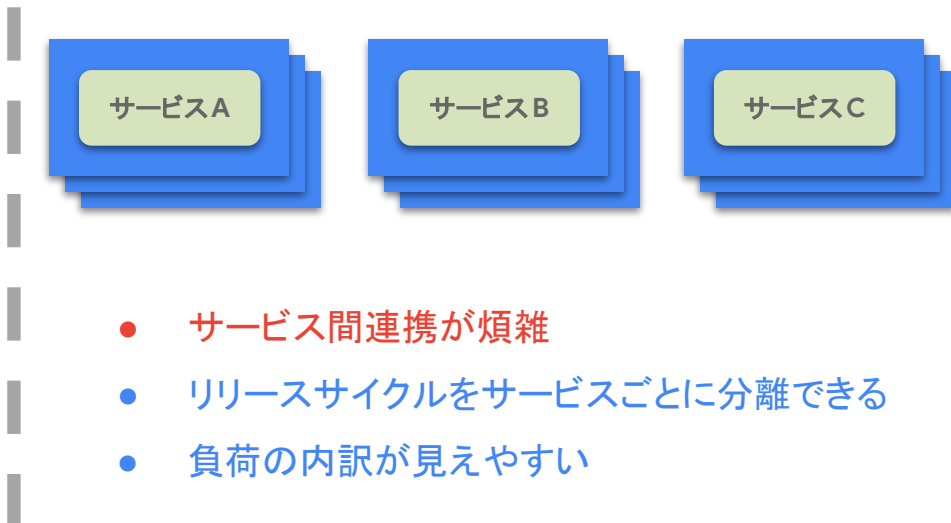
- 利用タイトルの大規模化に対応
 - 運用負担が低い
 - リソース効率が良い

これらを実現するため、従来システムの設計当初には存在しなかったり、
今ほど普及していなかった技術を設計の前提として採用

モノリシック or マイクロサービス



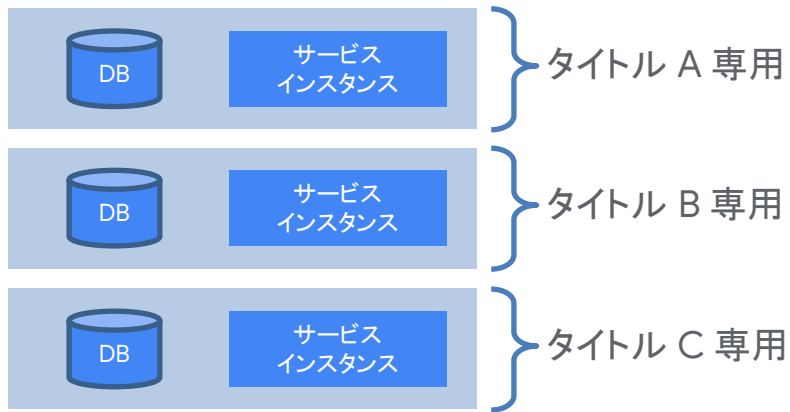
- サービス間連携が容易
- リリースサイクルは全体で共通
- 負荷の内訳が見えにくい



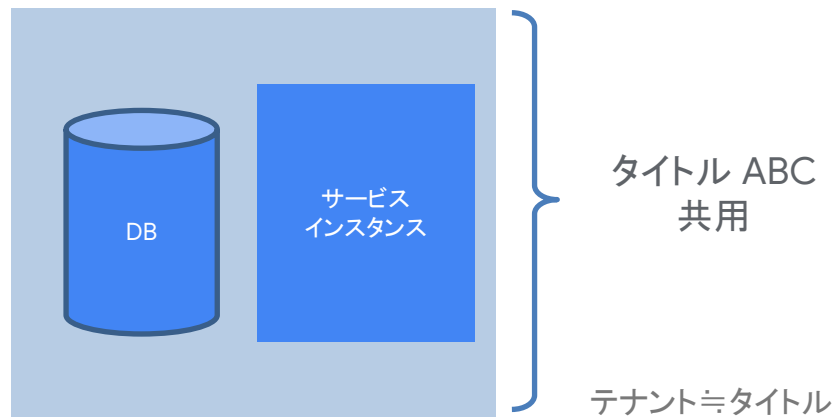
- サービス間連携が煩雑
- リリースサイクルをサービスごとに分離できる
- 負荷の内訳が見えやすい

- NPLN ではサービスごとの開発や運用の独立性を重視してマイクロサービスを採用
- その実現のため、基盤には Kubernetes と Istio を採用し、
Google Kubernetes Engine と Anthos Service Mesh を利用して運用

シングルテナント or マルチテナント



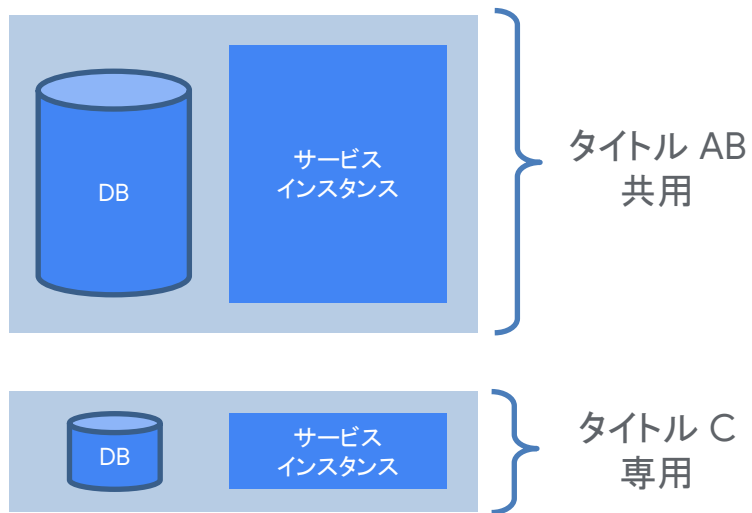
- カスタムしやすい。
- スケーラビリティの上限がタイトル一つ分
- 負荷などの影響を分離しやすい。
- タイトルが増えると管理が大変
- 余剰リソースが非効率



- カスタムしにくい。
- スケーラビリティの上限が複数タイトル合算
- 負荷などの影響が全体に波及
- タイトルが増えても管理が容易
- 余剰リソースが効率的

NPLN でのマルチテナント

マルチテナント構成を基本としつつ、局所的にシングルテナントも可能にするハイブリッド構成

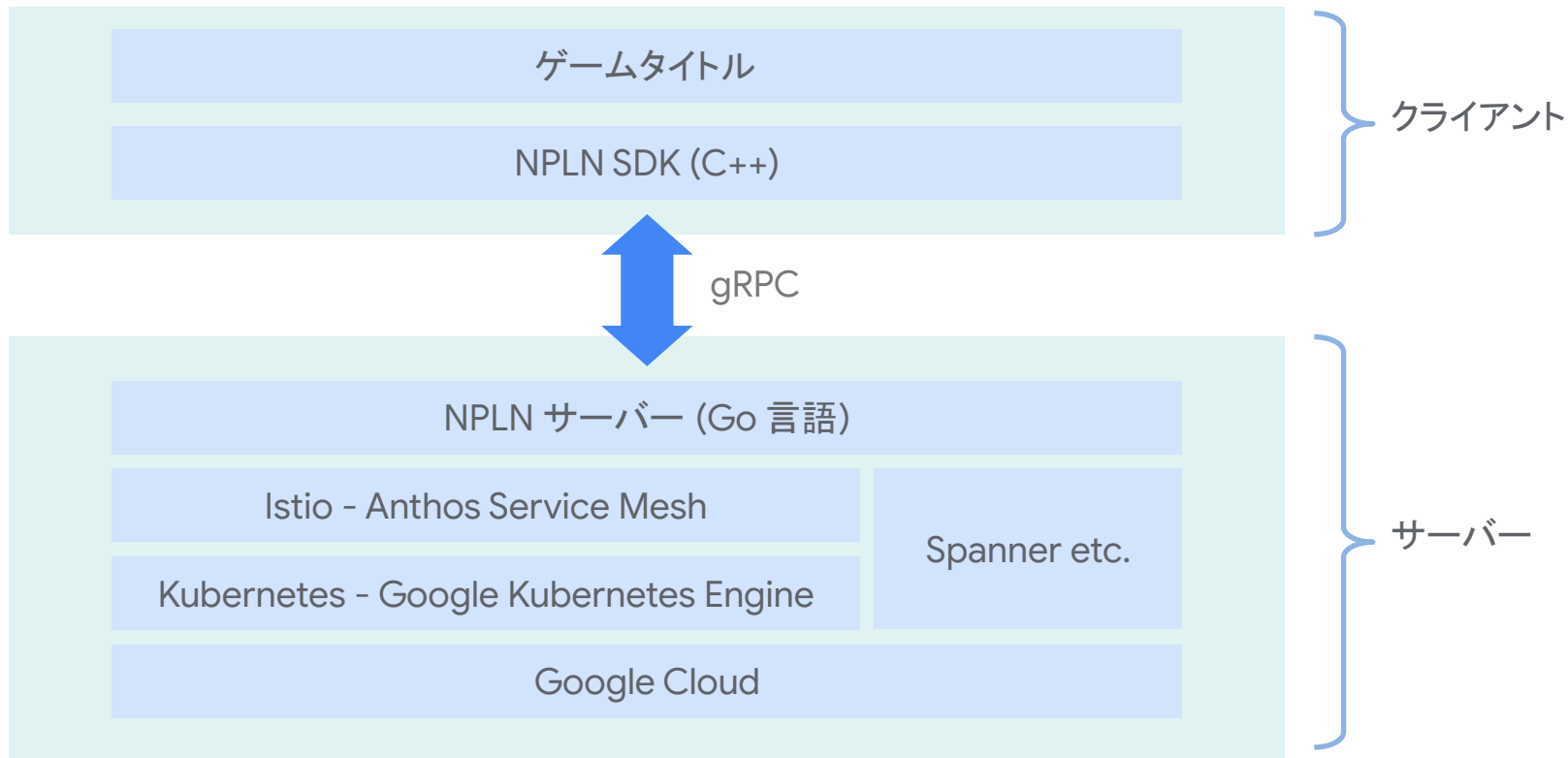


- **カスタムしにくい。**
 - マッチメイク設定などテナント固有のロジックを DSL で記述可能に拡張
- **スケーラビリティの上限が複数タイトル分**
 - DB に Spanner を採用し、大量のタイトルで DB を共有した場合でも対応できる
 - スケーラビリティを確保
- **負荷などの影響が全体に波及**
 - ルーティングを工夫して、一部のタイトルだけインスタンスを分離可能に

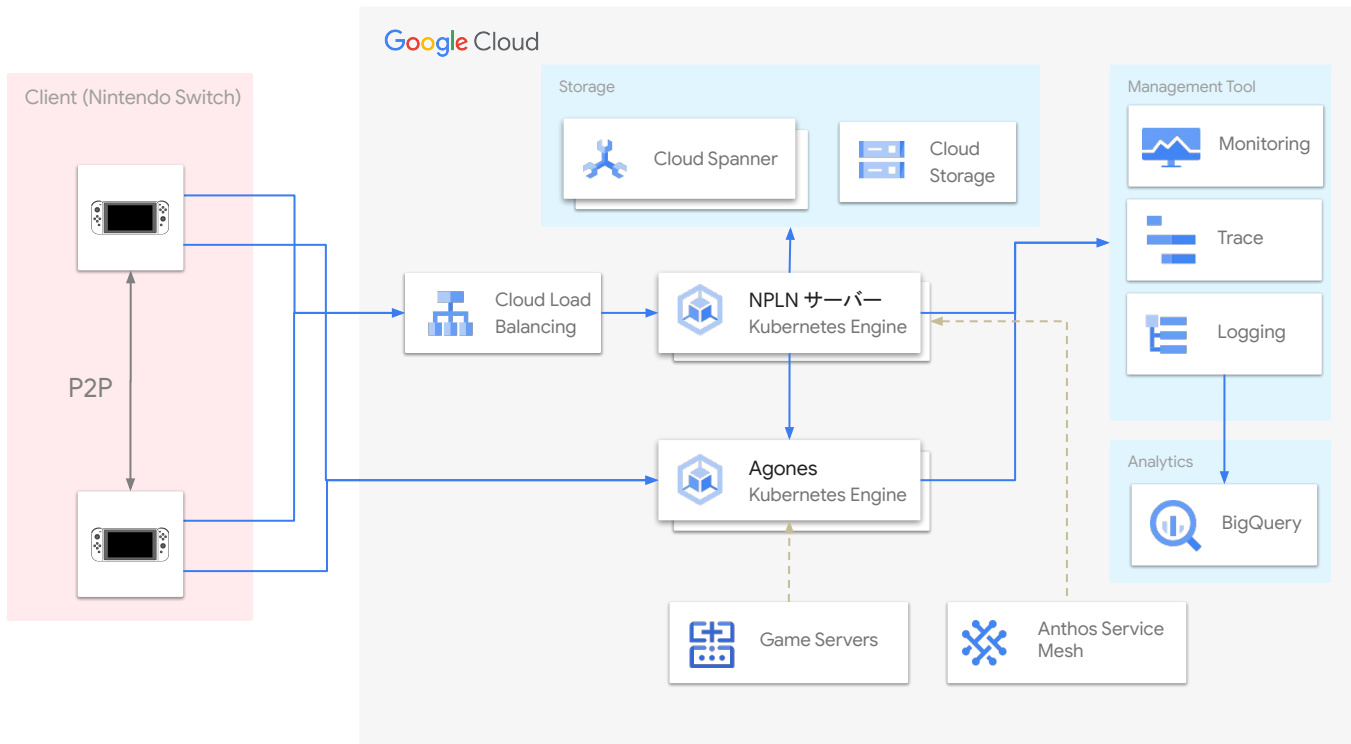


アーキテクチャ

主な構成要素

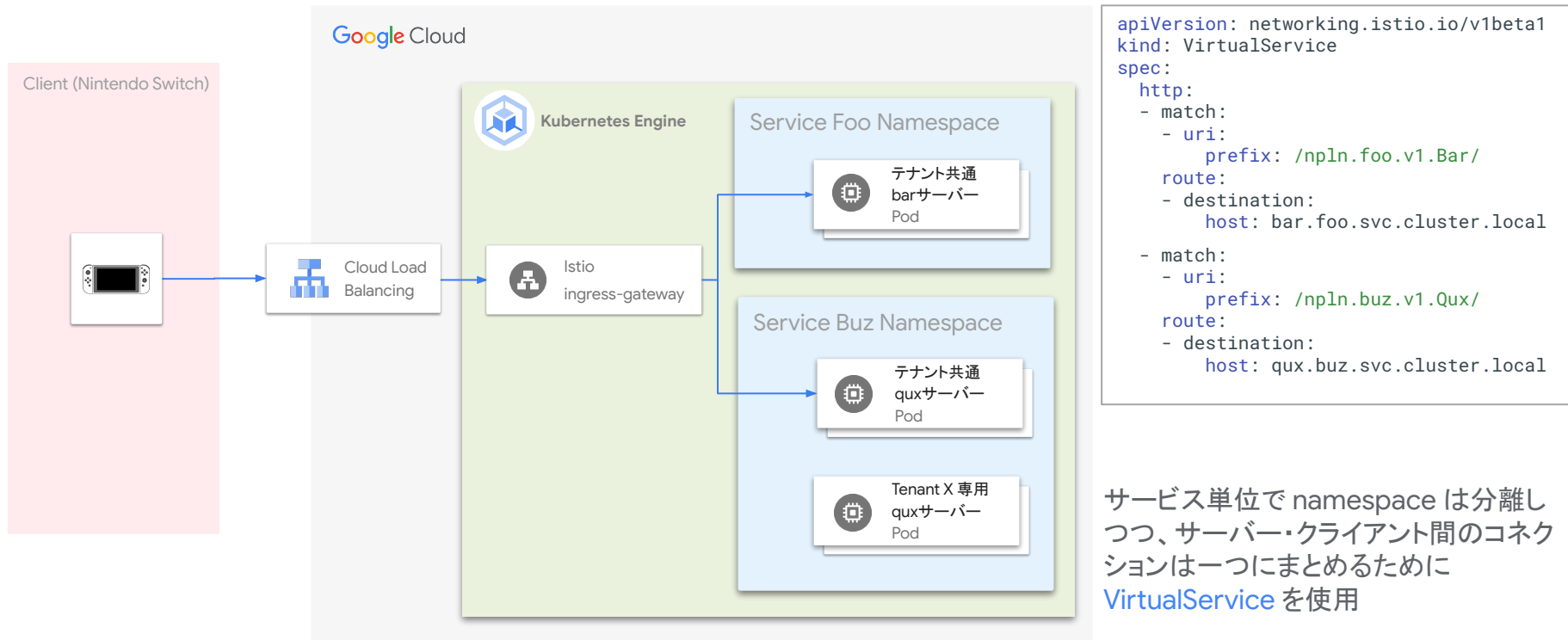


概要図

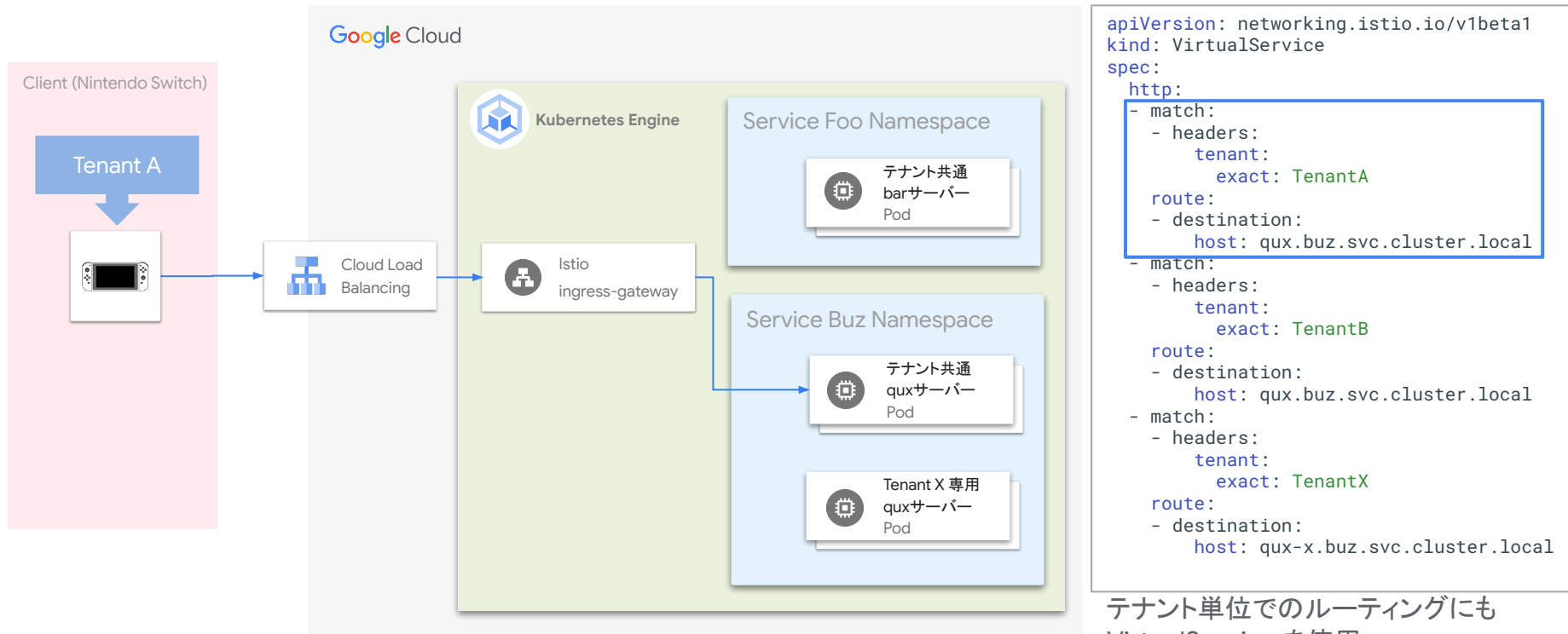


- クライアントは Cloud Load Balancing 経由でNPLN サーバー用の GKE に接続
- Istio は Anthos Service Mesh で管理
- Agones クラスタは Game Servers で管理
- ログの一部は BigQueryに転送して分析

マイクロサービスの実現

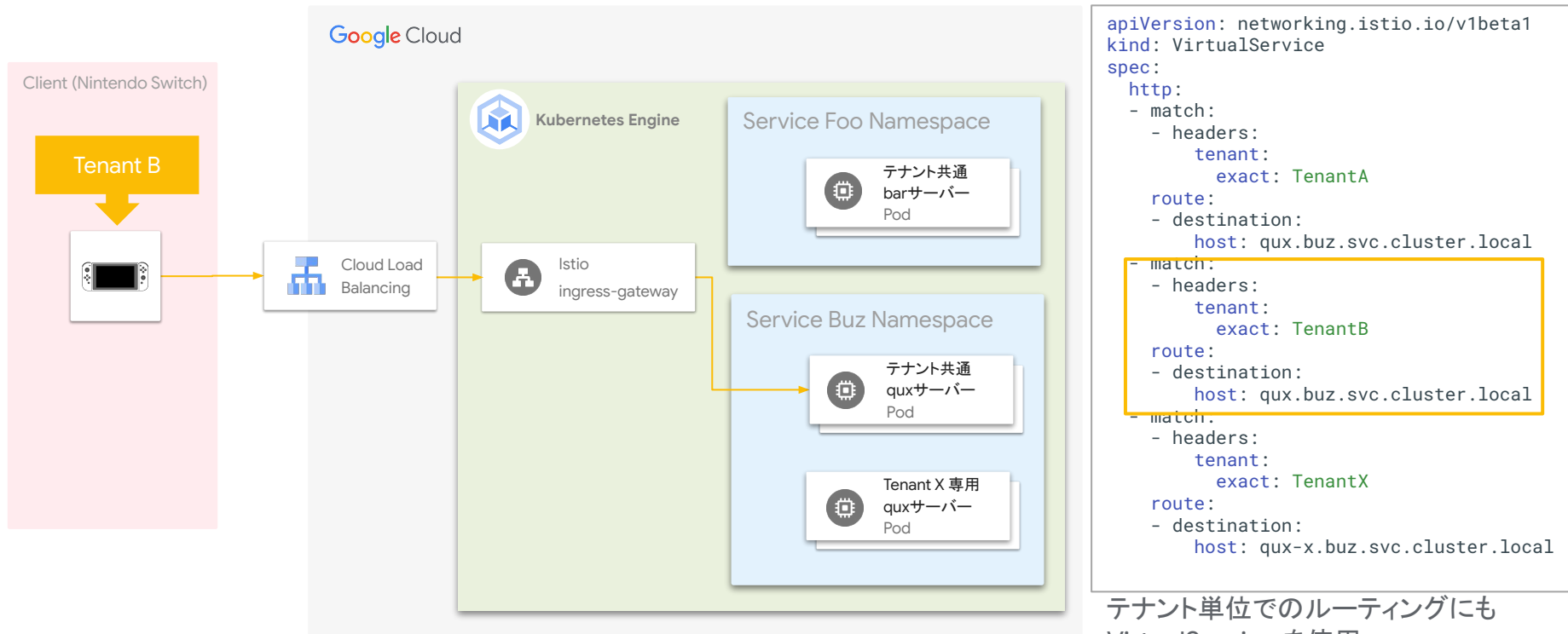


マルチテナント構成の実現



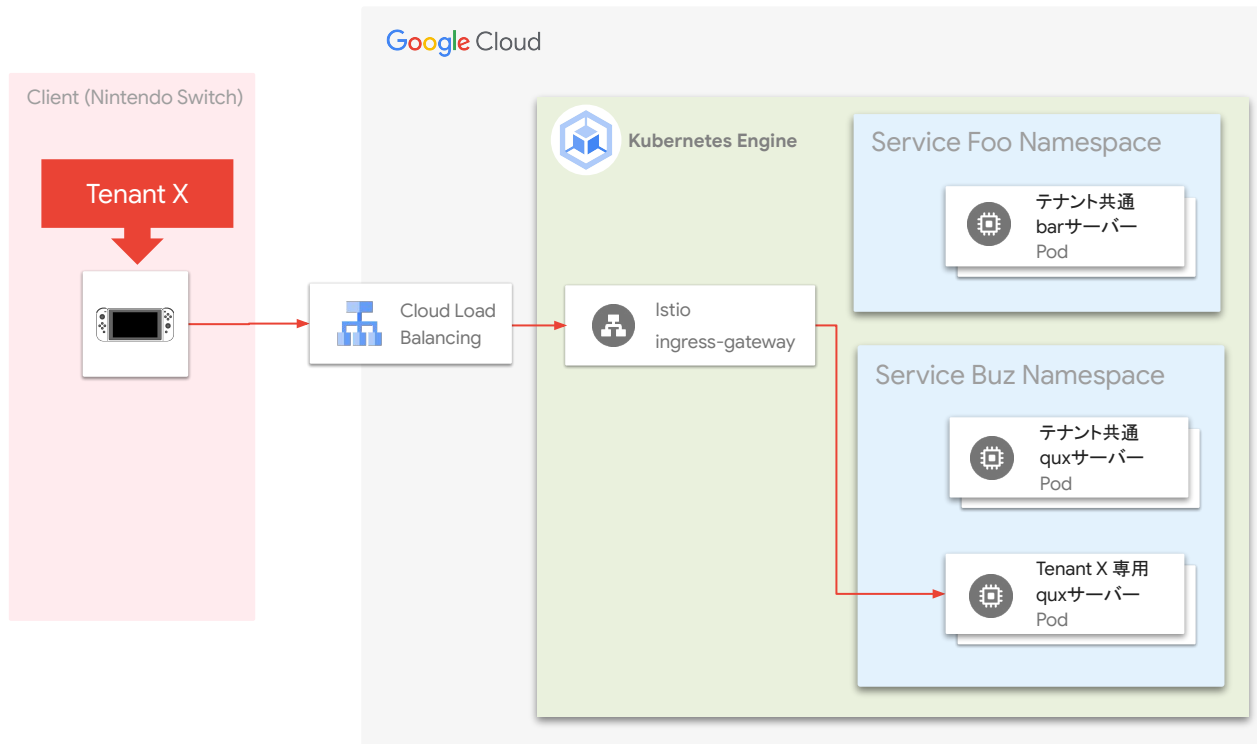
テナント単位でのルーティングにも VirtualService を使用

マルチテナント構成の実現



テナント単位でのルーティングにも VirtualService を使用

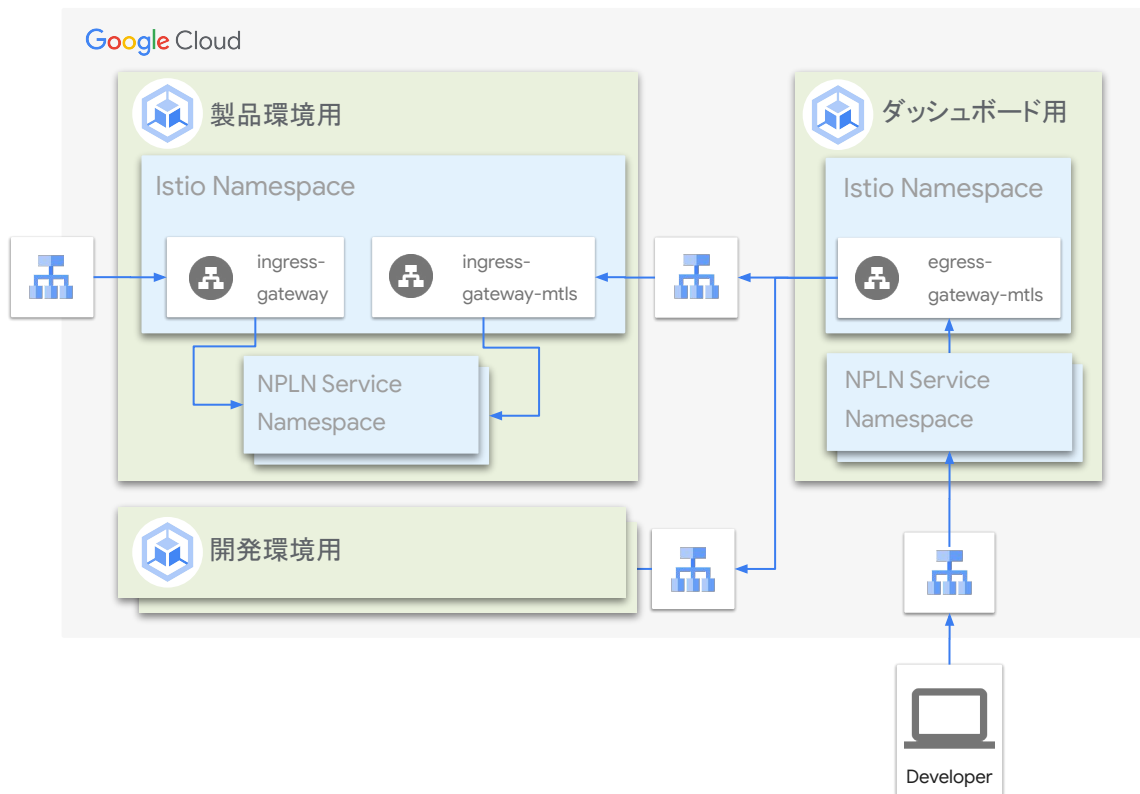
マルチテナント構成の実現



```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
spec:
  http:
  - match:
    - headers:
      tenant:
        exact: TenantA
    route:
    - destination:
        host: qux.buz.svc.cluster.local
  - match:
    - headers:
      tenant:
        exact: TenantB
    route:
    - destination:
        host: qux.buz.svc.cluster.local
  - match:
    - headers:
      tenant:
        exact: TenantX
    route:
    - destination:
        host: qux-x.buz.svc.cluster.local
```

テナント単位でのルーティングにも
VirtualService を使用

マルチクラスタ構成

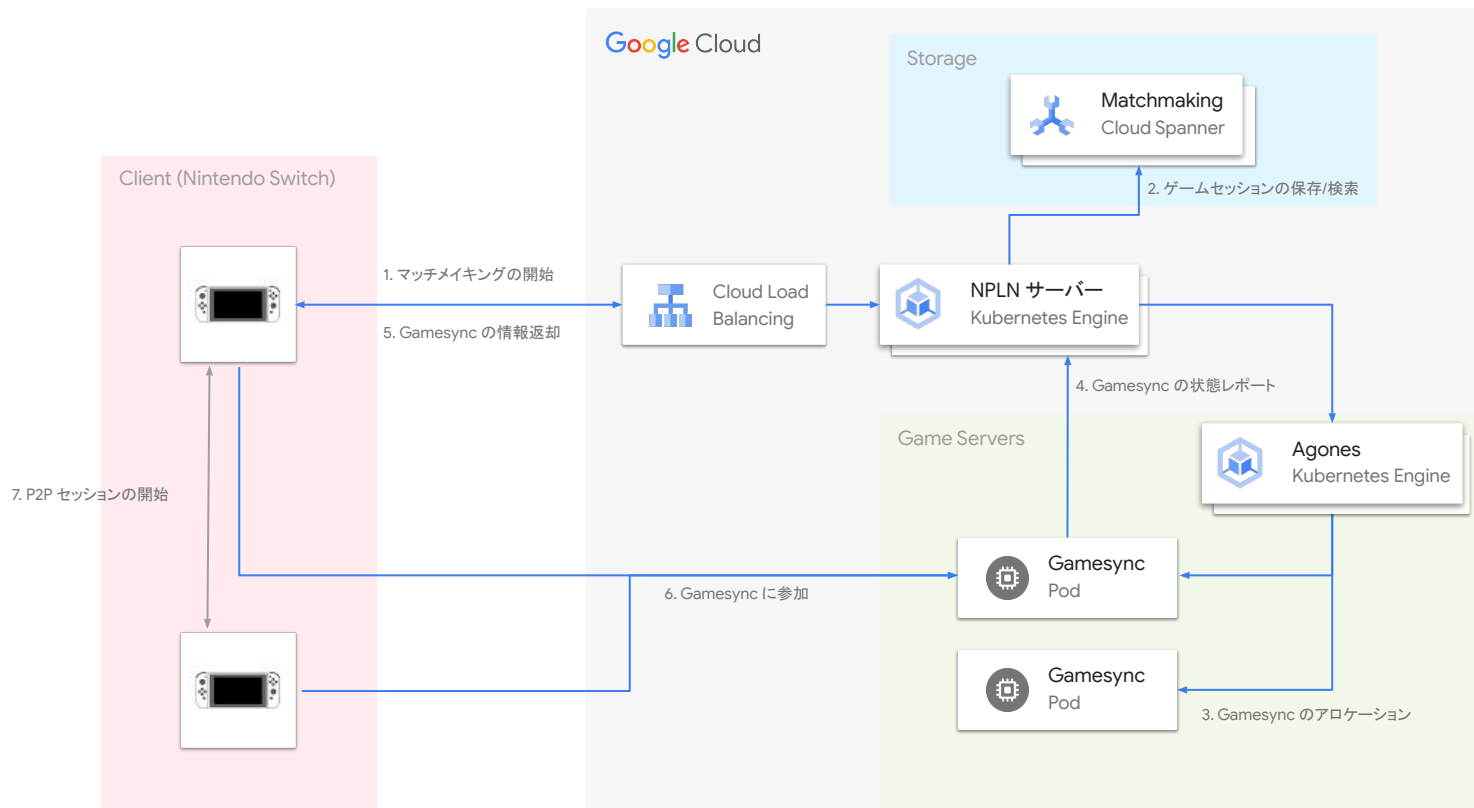


- 用途ごとに GKE クラスタを分離
- サービスやテナントごとの設定値はダッシュボードで集中管理し、各クラスタのサービスに反映
- クラスタ間を mTLS プロキシで接続
- VirtualService、DestinationRule、ServiceEntry、Gateway など Istio の CRD を使用して透過的に別クラスタのサービスへアクセス
- 将来的には ASM の機能への切り替えも検討



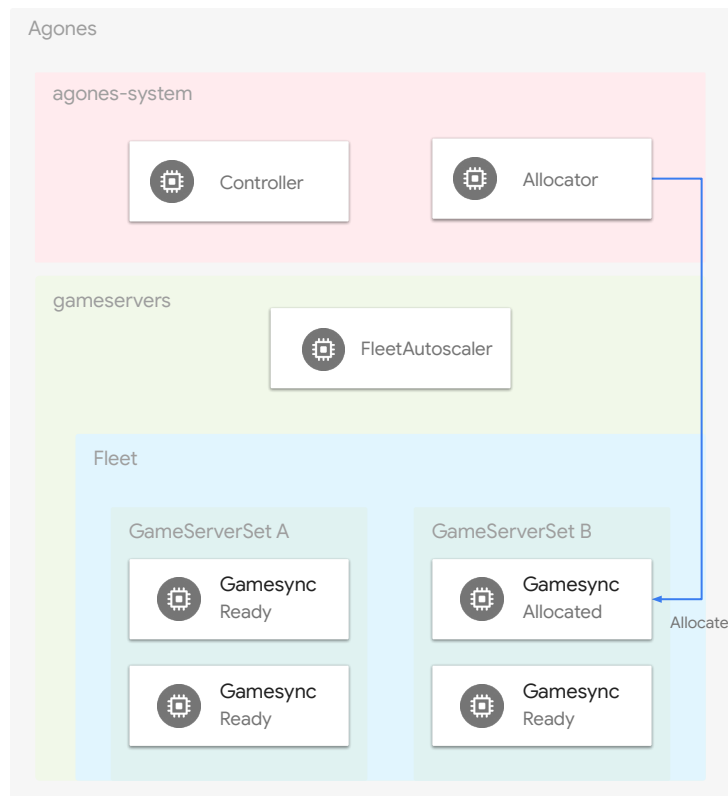
Google Cloud の利用に際しての 課題と対策

マッチメイキングサービスの構成



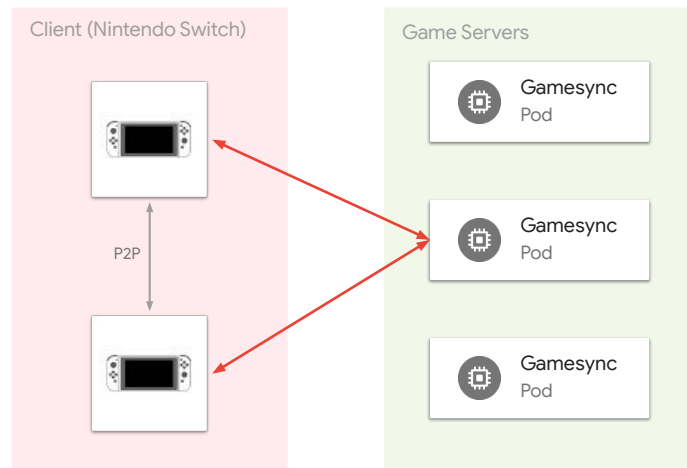
Agones とは

- ゲームサーバーを管理する OSS
- ゲームサーバー群を Fleet として管理
 - ゲームサーバーを Pod として管理 (Gamesync)
- ゲームサーバーは状態を持つ
 - Allocated: 利用中
 - Ready: 利用可能
- FleetAutoscaler は Buffer を管理
 - Ready の最小必要 Pod 数の監視など



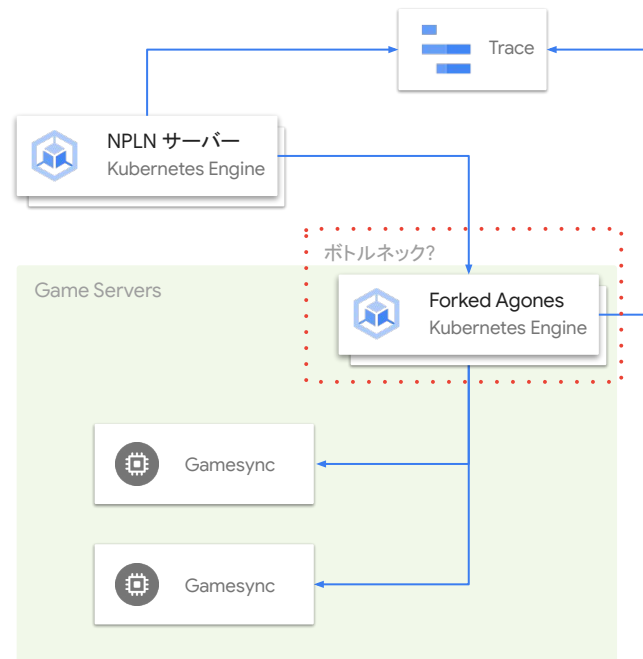
Gamesync とは

- ゲームセッションごとに用意するセッション管理サーバー
- 機能
 - 強い整合性を持ったインメモリ DB
 - リアルタイムの変更通知
- 利用方法
 - P2P では不整合が発生しやすい一部のゲーム情報の状態共有
 - P2P セッションのシグナリング経路



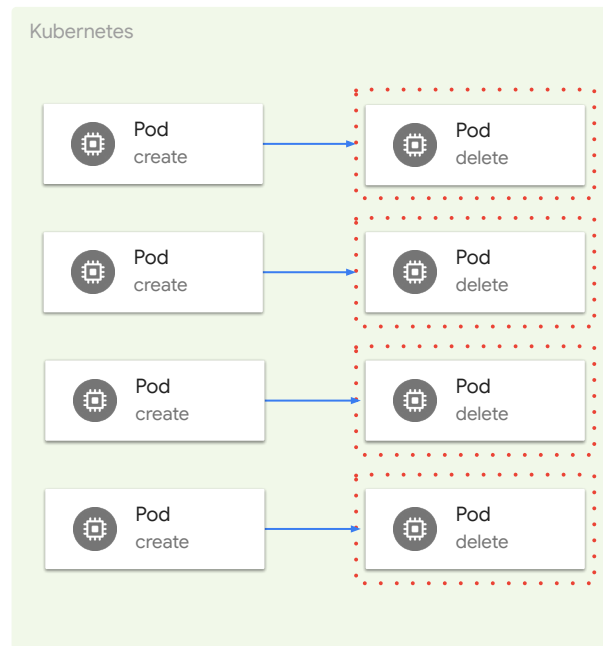
Allocate スループットの課題

- Allocate とは
 - Gamesync の状態を変更する処理
 - Pod の Status が Ready -> Allocated に遷移
- Allocate のスループットが想定より低い
 - Agones をフォークし、Cloud Trace を追加
- GKE コントロールプレーン依存の可能性
 - Cloud Trace で目立ったレイテンシはなし
 - 計測不可なコントロールプレーン部分に注目



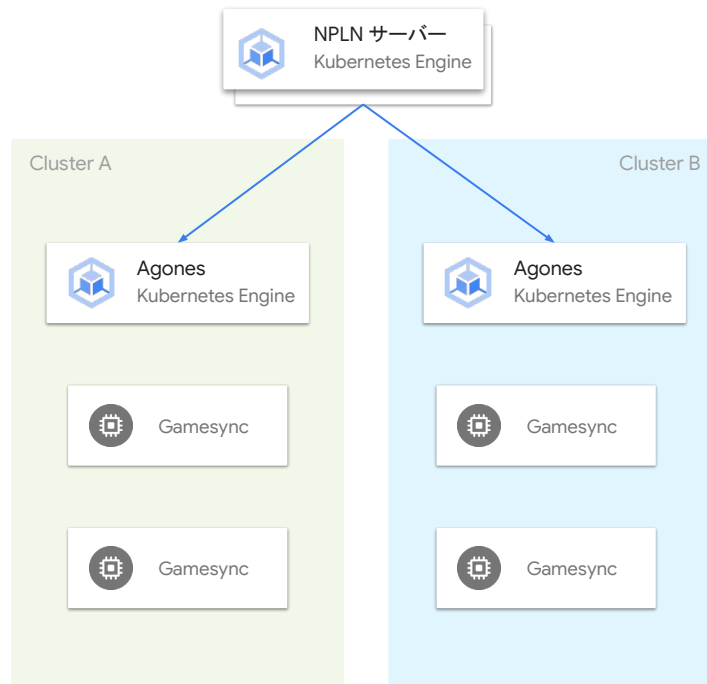
Allocate スループットの対策

- 軽量の Pod の作成および破棄を繰り返す性能試験
 - 期待した性能に届いていないことを確認
- 500 ノードに増やすことで性能改善を確認 (※1)
 - Pod の操作レートが 20 QPS -> 100 QPS に改善
 - Allocate スループットの改善を確認
- Pod を再利用で更に性能改善することを確認
 - デフォルトは 1 回のセッションが終わると削除
 - 削除ではなく Allocated -> Ready に戻す



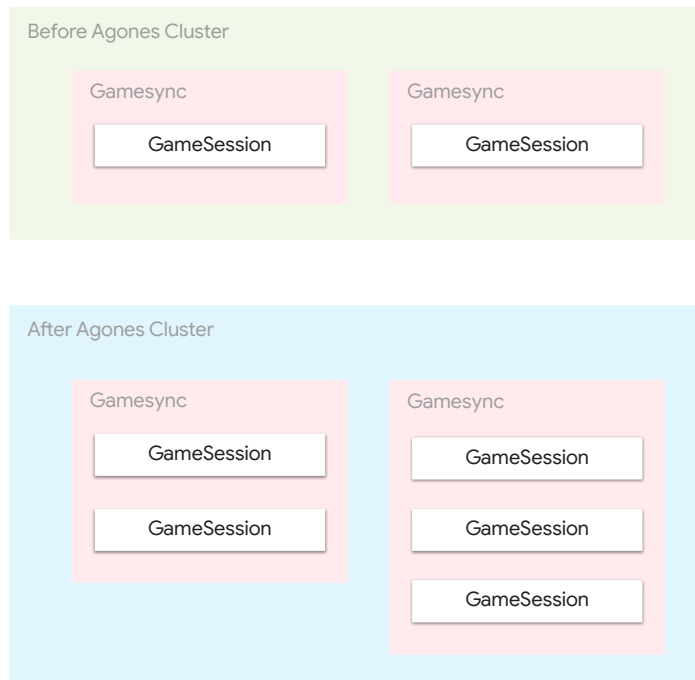
マルチクラスタ化

- 単一クラスタの性能上限
 - コントロールプレーンの性能で Allocate 速度が律速
- スケーラビリティのためにマルチクラスタ化
 - NPLN サーバーで対象クラスタを振り分け
- クラスタ数依存でスケール可能に
 - トレードオフとして運用コストが増加



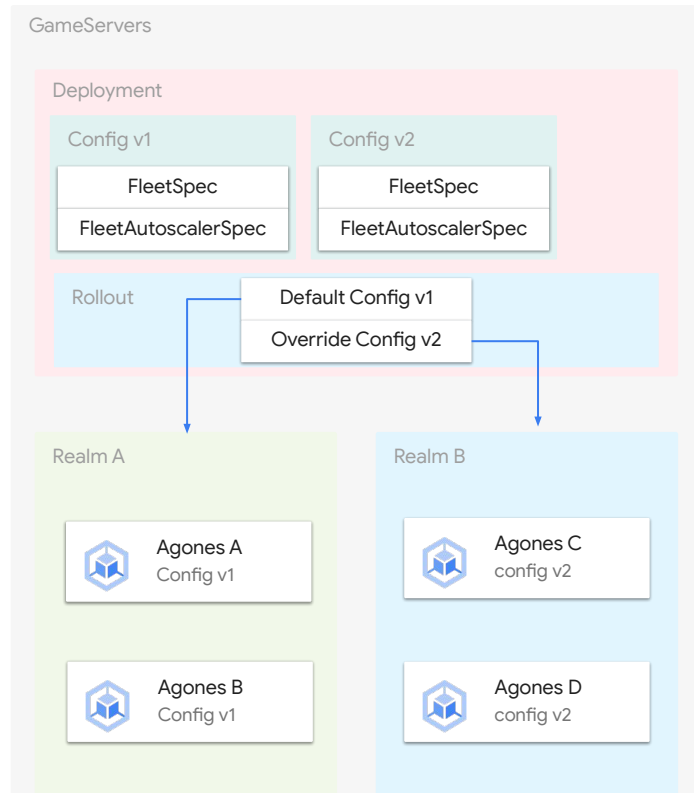
Gamesync の共用

- 課題
 - リージョン単位のクラスタ上限数
 - 多数のクラスタ運用による運用コストの増加
 - 非効率なリソース消費
- 複数のゲームセッションで 1つの Gamesync を共有
 - 論理的に Gamesync を分離
- 効果
 - クラスタの運用数を大幅に削減
 - リソース効率を改善



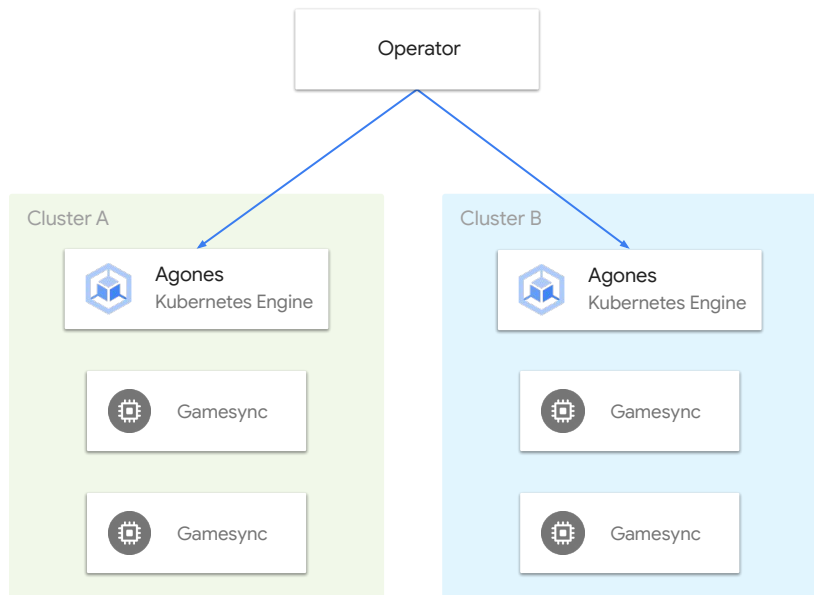
Game Serversの活用事例

- Game Servers
 - 複数の Agones クラスタを管理するための Google Cloud マネージドサービス
- NPLN のユースケース
 - マルチクラスタ化された Agones のリソース管理
 - Fleet のカナリアデプロイ
 - クラスタの Blue / Green デプロイ

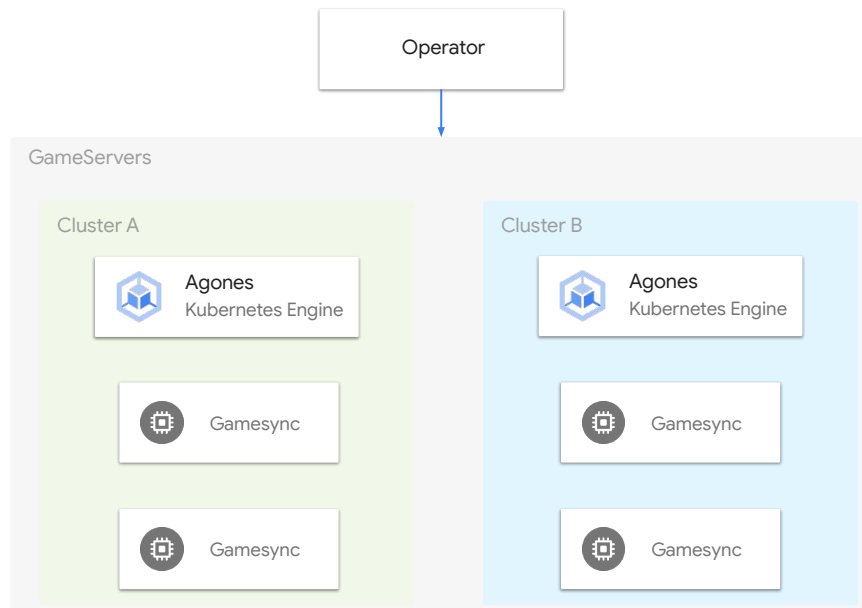


マルチクラスタ化された Agones のリソース管理

Game Servers を利用しない場合

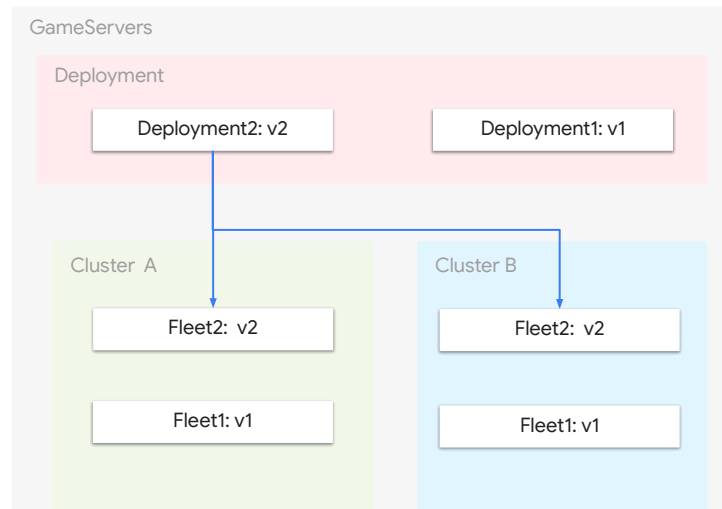


Game Servers を利用する場合



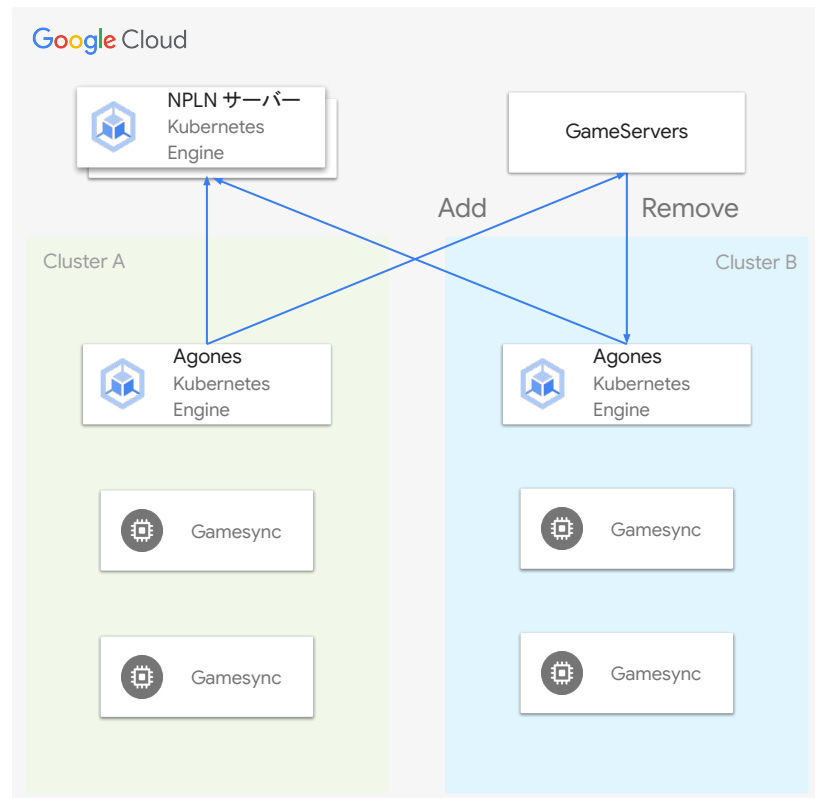
Fleet のカナリアデプロイ

- Deployment を分けてカナリアデプロイ
 - 動作確認後に次の Deployment を反映
- 新しいバージョンが動作しない場合でもサービス影響が出ないように
 - 問題時、起動しないバージョンの Pod を局所化
 - 問題があれば Deployment 単位でロールバック



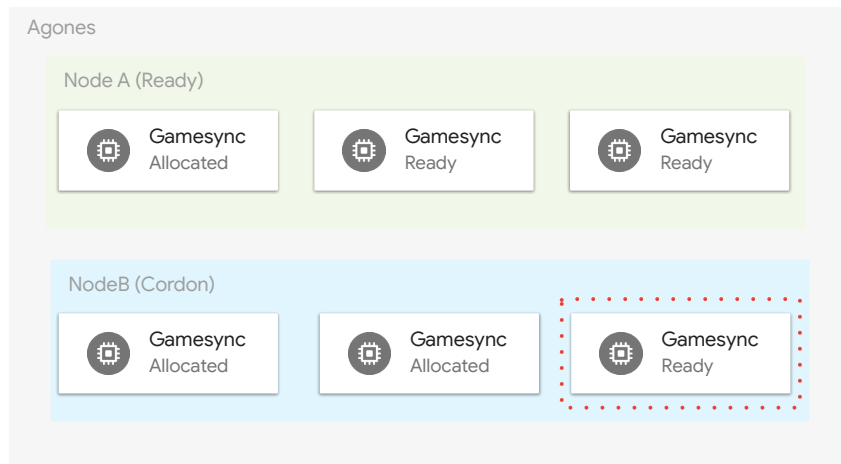
クラスタの Blue / Green デプロイ

- GKE や Agones の更新時に利用
 - 問題があればロールバック
- Game Servers
 - クラスタの追加、除去
- NPLN サーバー
 - 新しいクラスタのエンドポイント追加
 - 古いクラスタのエンドポイント除去
- 古いクラスタ
 - Allocated な Gamesync が 0 になると削除



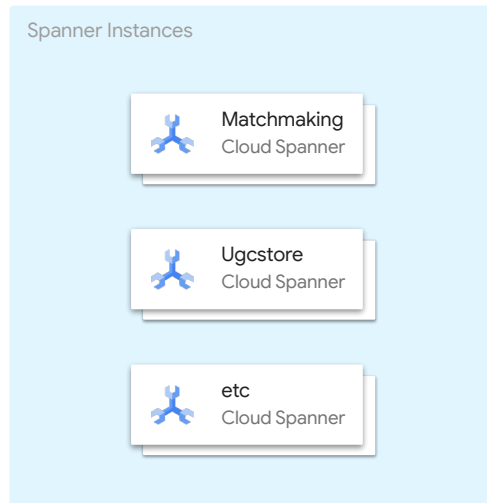
GKE ノードプールの更新

- 更新するノードにユーザーがない保証
 - 対象ノードは新規セッションの作成不可
 - Allocated Pod がある間はノード削除不可
- Gamesync に Annotation を指定
 - Cordon 状態なので Pod は作成不可
 - Allocated Pod があるノードは削除不可
 - ユーザー影響のない更新が可能



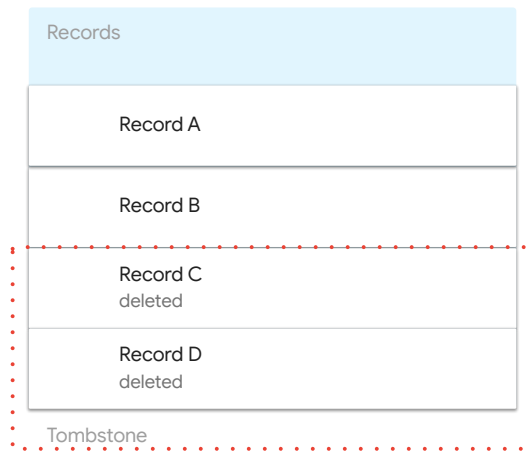
Spanner の活用事例

- ユーザーに直接関わるサービスは原則 Spanner を利用
- マッチメイキングでは
 - ゲームセッションの状態管理に利用
 - 高頻度で Read / Write が発生
- インスタンスはサービス単位で分離



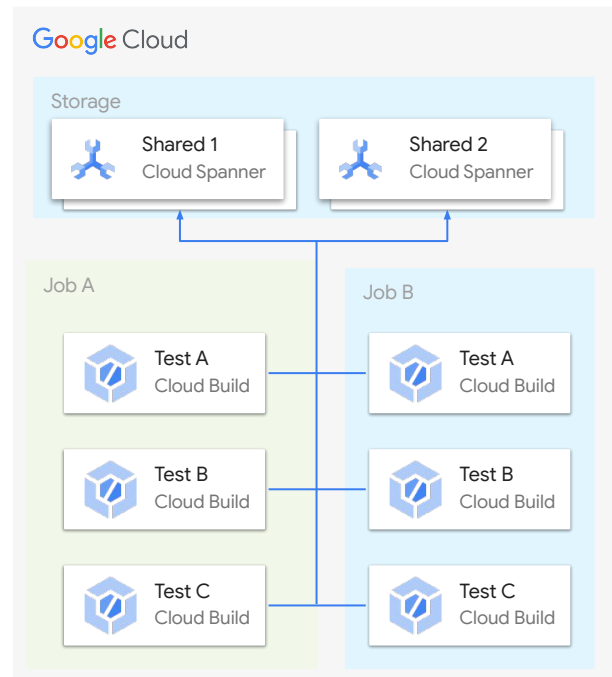
削除済み行の課題

- Spanner は削除済みの行が物理的に一定期間残る
 - 数日から 1 週間程度で実際に削除される
 - 削除されるまで残る行を Tombstone と呼ぶ
- ロングランテストで徐々に検索性能低下
 - 行の削除後、一定期間後にスキャン対象外となるため
- Tombstone を考慮した構成にする必要がある
 - 性能低下を前提としたノード台数の模索
 - ピークタイムがどの程度継続するかを検討



CI での自動テストの課題

- エミュレーターの非対応機能
 - 複数トランザクションの並走、特定のヒント句の利用、実行統計の取得
 - ブランチの重要度に応じてテストを一部スキップ
- 実際の Spanner を用いた自動テスト課題
 - 自動テスト環境は Cloud Build を利用して極力並列化
 - データベースを高頻度で Create / Delete した場合の遅延
 - テストの並列数を上げたときの DB 上限数



まとめ

- 汎用ゲームサーバーの再設計プロジェクト NPLN の紹介
- マイクロサービス化とマルチテナント構造を Kubernetes / GKE と Istio / ASM でどのように実現したかを説明
- Agones / Game Servers や Spanner など Google Cloud のサービスを活用する上での課題と取り組みを紹介

We are hiring!

- サーバーアプリケーション エンジニア
- サイトリライアビリティ エンジニア (SRE)
- サーバーセキュリティ エンジニア
- ネットワークサービス システムエンジニア
- プロジェクト マネージャー (Web サービス・システム開発)

...

「任天堂 キャリア採用」で検索

<https://www.nintendo.co.jp/jobs/career/>

Thank you.

