

GCP を用いた 1日に400億レコードの 視聴データを扱うための データ分析基盤

東 遼平

株式会社 TVer Technologies / Data Architect

森藤 大地

株式会社 TVer Technologies / Chief Data Scientist

東 遼平 (@B1ufe)



株式会社 TVer Technologies
データユニット DataArchitect

TV 視聴などによって送られてきたデータの分析
基盤や分析データを可視化するための BI ツール
の設計／開発／運用／保守に従事。

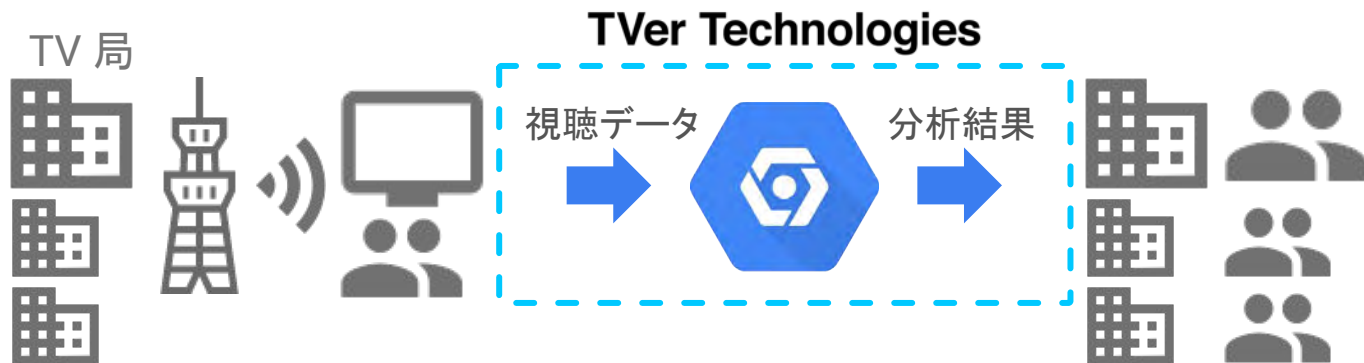
データ量、データの形式、TV 視聴端末による違い、
放送規格、放送局、系列局、放送形態、番組、
コンテンツ、広告代理店、個人情報保護、など
様々な要因を吸収した基盤のベストアンサーを
求めて日々邁進。

株式会社 TVer Technologies

前身の HAROiD から、
TV 局横断での「ID / データプラットフォーム事業」を推進するために新設分割し、
2019 年 9 月に在京民放 5 社が出資し TVer を運営しているプレゼントキャストへの株式譲渡により生まれた。

TV 放送業界共通での ID / データの活用を推進し、ユーザーや放送局、広告主に対して今までにないサービスを提供していく。

2020年7月1日付けで株式会社 TVer Technologies に社名変更



視聴データの収集方法は 3 種類

1. パネル調査

- みんなのよく知る視聴率で採用されている方法

2. テレビメーカー方式

- テレビ自体の機能、例えばリモコン操作などを送る方法

3. BML 方式:各放送局、TVer Technologies

- BML = XML ベースのデータ放送向け言語 (ECMAScript が使える)
- テレビ自体が ECMAScript で HTTP/S によってサーバーに送る方法

テレビが HTTP/S ?

地デジ化によりテレビ内部で放送波からプログラムコンテンツを受け取り、実行し、HTTP/S によりサーバに送信する

ストリーミング配信の視聴のためなどで、インターネットに接続したテレビで利用可能

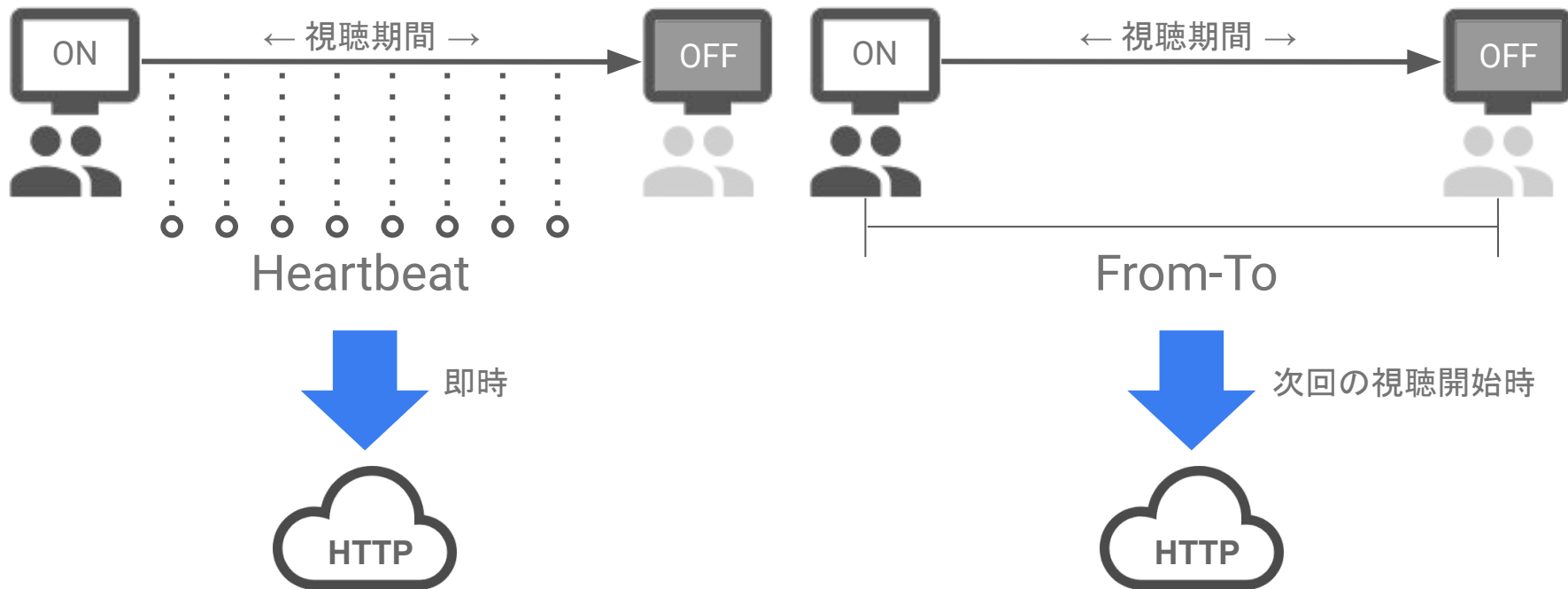
※ただし、ECMA のバージョンはだいぶ古く、かつ、テレビカスタマイズされている

※ HTTP は同期処理

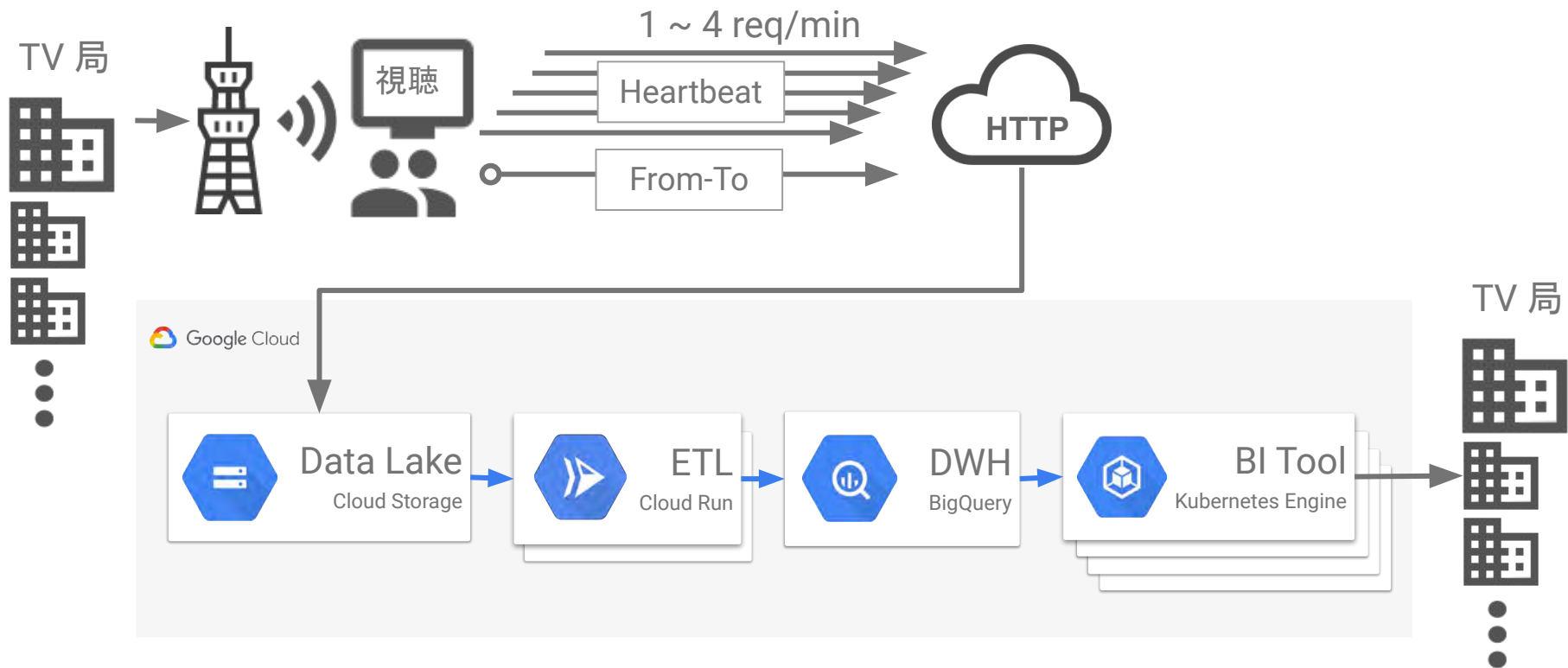


視聴データの形式は 2 種類

視聴データには (大まかに) Heartbeat と From-To の 2 種類の形式がある



視聴データの流れ



Heartbeat と From-To の特徴

	Heartbeat	From-To
(直近の視聴動向の) 送出までの時間	数秒から1分以内	不定 (視聴者による)
視聴期間の正確さ	視聴開始・終了時刻が不定	正確
一定期間のデータ量	膨大	少ない
一定期間の情報量 (2つの方式の比較)	多い	少ない
フェイルソフト (データ破損等の影響)	数分の視聴データのみ欠損	1つの視聴期間全て欠損

Heartbeat と From-To のイイトコ取りを模索してる

規模

(想定しておかないといけないデータ量)

5853 万 (世帯 / 全国)

(https://www.soumu.go.jp/main_content/000633313.pdf の世帯数総計を切り上げ)

およそ 40 %

(HUT: 全日平均視聴率)

<https://news.yahoo.co.jp/byline/fuwarai/20190520-00126604/>

およそ 30 %

(結線率: 全国の受信機がインターネットに接続されている割合)

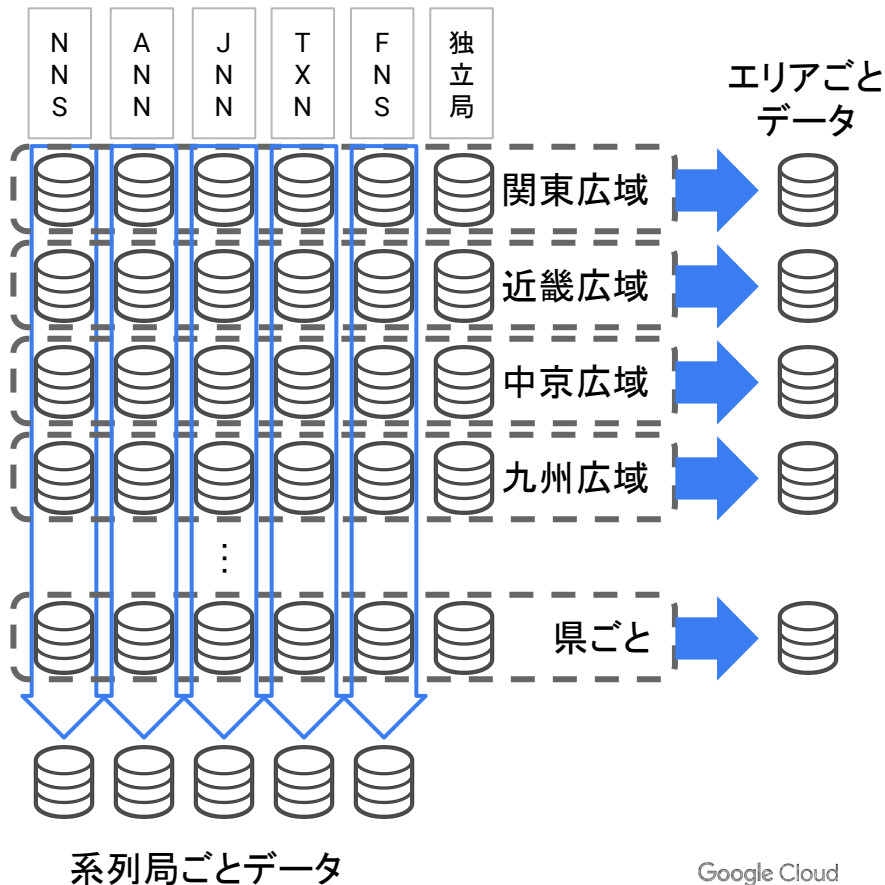
4000 億 (レコード / 日)

5853万 (世帯) x 40% (HUT) x 30% (結線率) x 4 (req / 分) x 60 (分) x 24 (時間)
= 28,094,400 (req / 分) x 60 (分) x 24 (時間)
= 40,455,936,000 (req / 日)

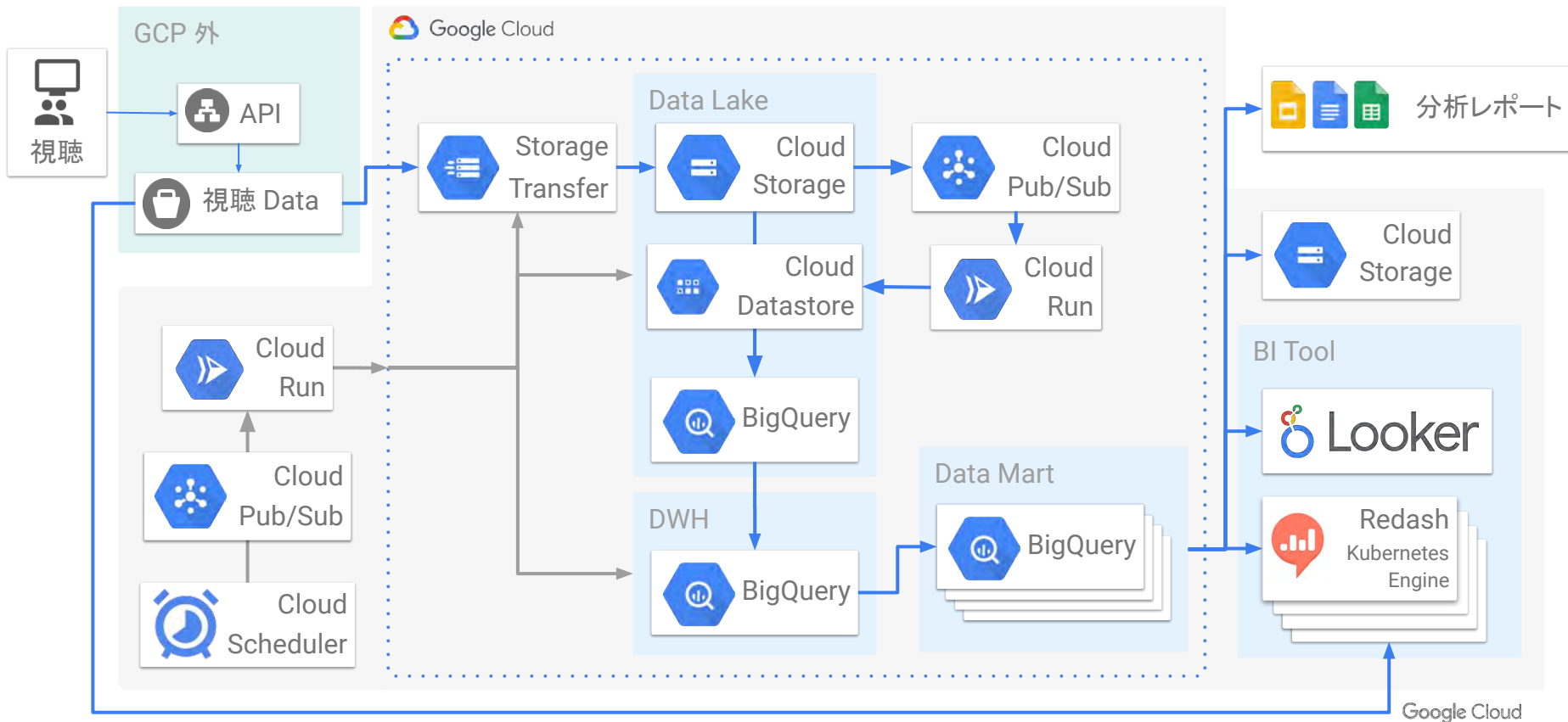
これを GCP で ETL するよ

データの保持の要件

- 要件
 - 複数の地域・複数の系列局のデータを効率的に扱う
 - コストをできる限り抑える
- 取りうるデータ設計手法
 - Wildcard Table
 - Partitioned Table
 - Clustering
- 放送局ごとに混ぜる方向が縦横両方
- 日付で絞り込みたい(全走査 NG)



ETL の構成全体 (Lambda Architecture)



Redash の構成



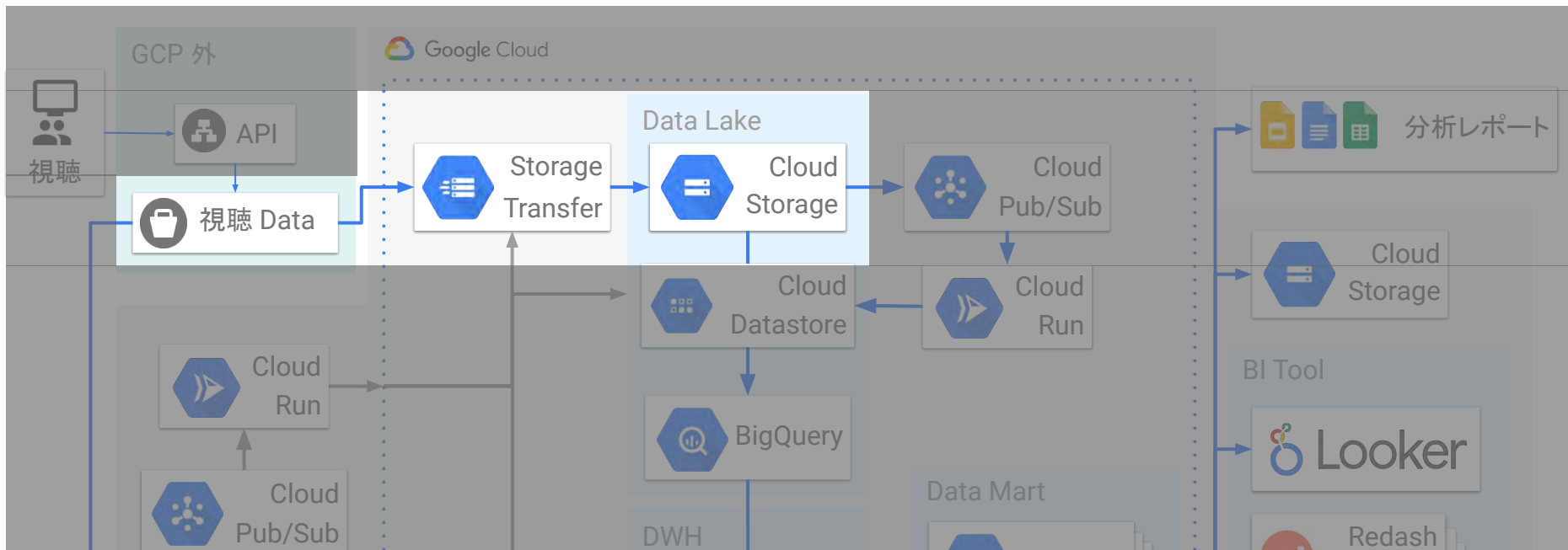
- GKE 上で 1 クラスタ上に TV 局ごと・系列マルチテナント
 - クラスタは有償になるので極力数を減らした
 - TV 局ごとにメンテしたいから node selector で分けてる
- Terraform を使って構築
 - 構築は手軽に
 - Redash を提供する TV 局の設定を用意して、
terraform apply するとその TV 局の Redash が立ち上がる
- Serverless VPC Access も登場したし、
CloudRun や CloudFunctions で Redash を色々操作できそうで夢は広がる

(´・ω・)つ



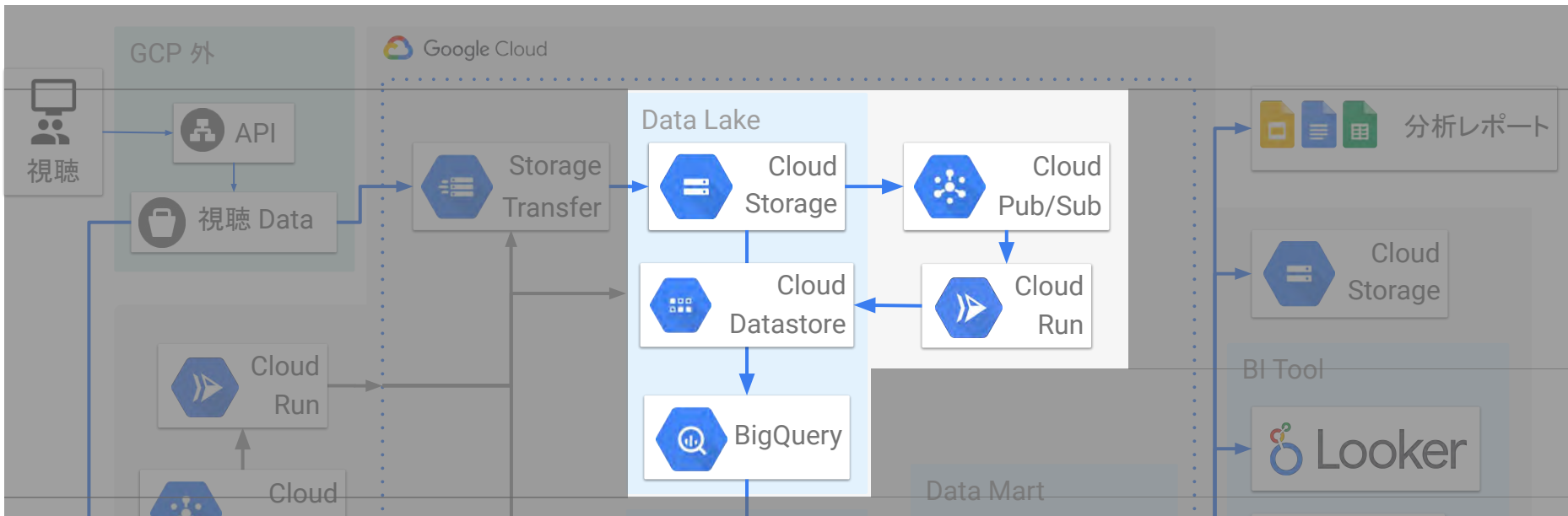
(´ω´) {今回はご縁がなk(ry

視聴データの転送



ネットワーク転送料金を抑えるために、非圧縮の視聴データをBigQueryにStreaming Insert するのではなく、Parquet + gzip に圧縮した視聴データファイルを一旦GCSに転送することにした。

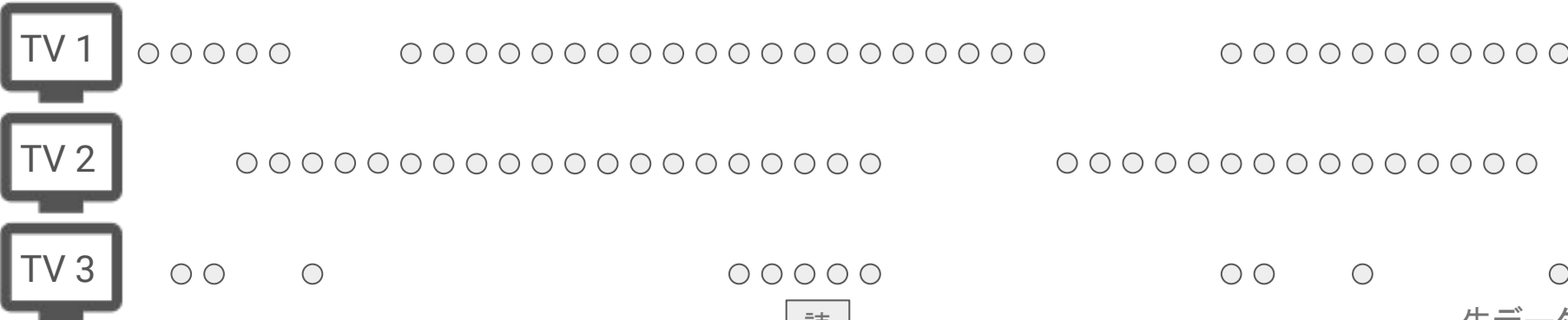
BigQuery へのデータロード



1 ロードジョブあたり1,000 万ファイルという制限があるので分刻みで読み込む。
非同期にファイルが転送される中で、コストを抑えつつ最低1 回はファイルが BigQuery に読み込まれることを保証するために、Datastore で転送されたファイルの読み込みステータス管理をしている。

(○視聴データ)

視聴時刻



読込

生データ

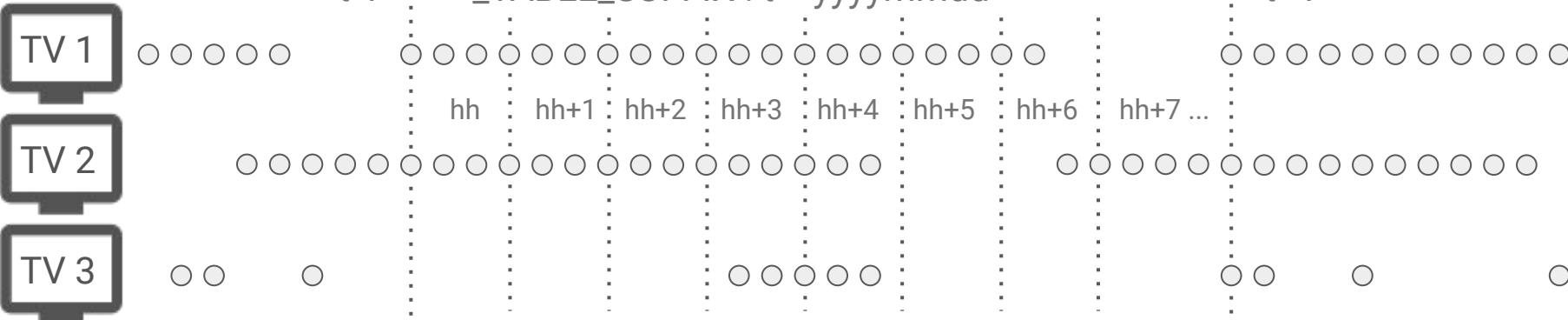
ロード後

t-1

_TABLE_SUFFIX : t = yyyyymmdd

t+1

hh hh+1 hh+2 hh+3 hh+4 hh+5 hh+6 hh+7 ...

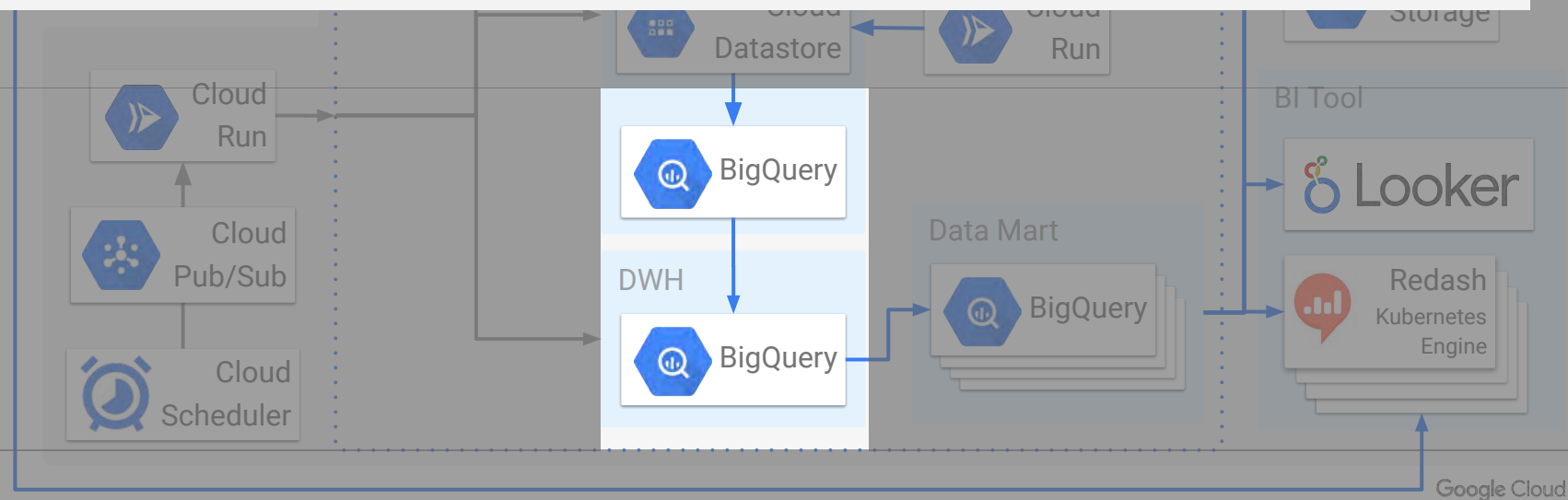


日付で Wildcard Table しつつ、時間を Clustering して Block Pruning

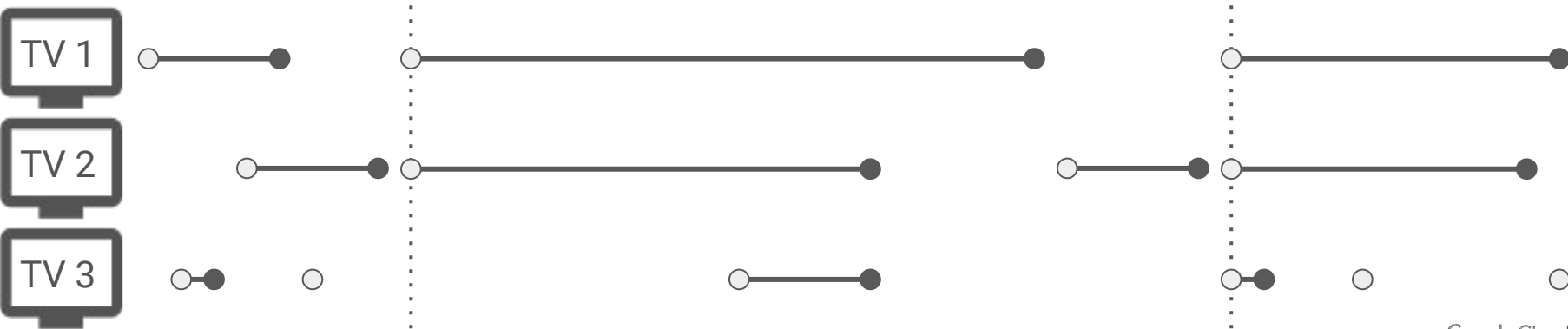
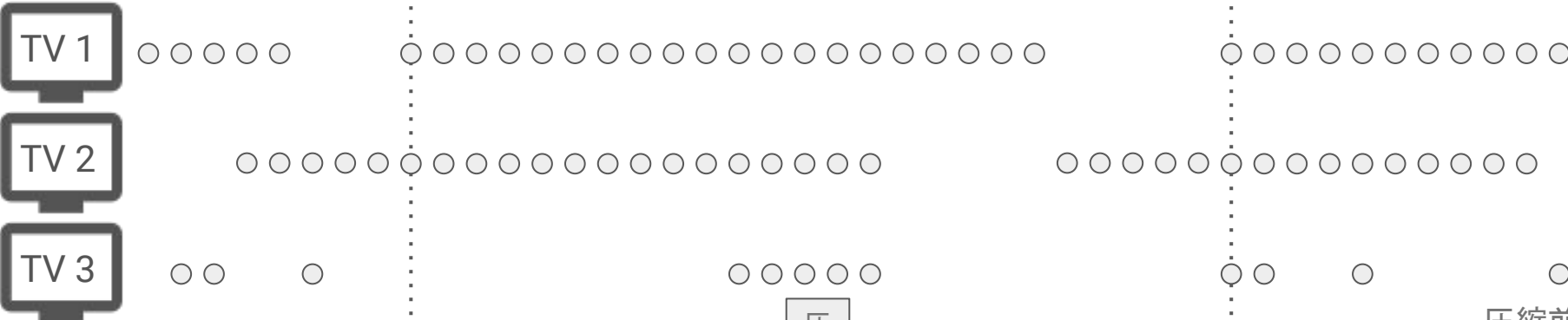
データの圧縮と分割と加工

視聴データは膨大な量が送られてくるけど定型なのでデータの圧縮処理をしている。

また実際提供する時は基本TV局ごとなのでTV局を識別するIDを元にデータを分割している。



(○視聴データ) t-1 TABLE_SUFFIX : t = yyyyymmdd t+1 視聴時刻



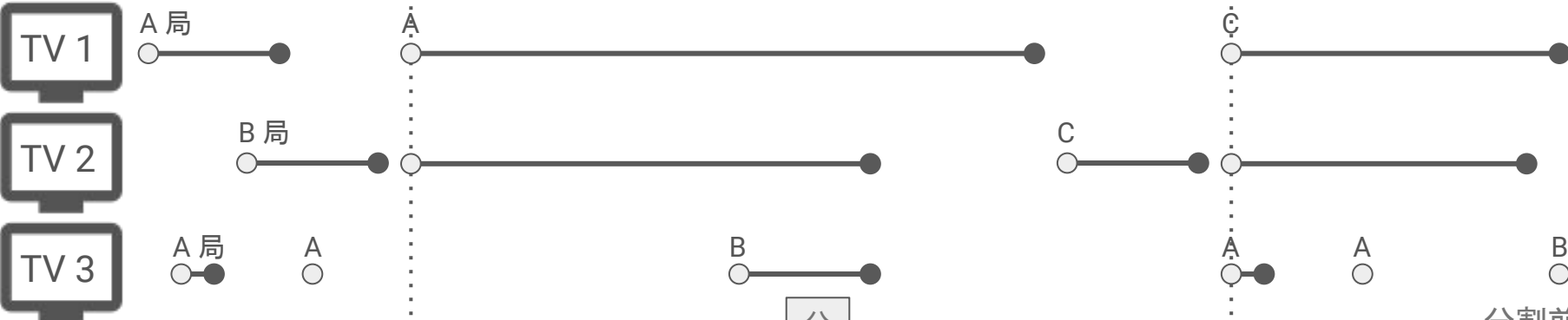
(○視聴データ)

t-1

_TABLE_SUFFIX : t = yyyyymmdd

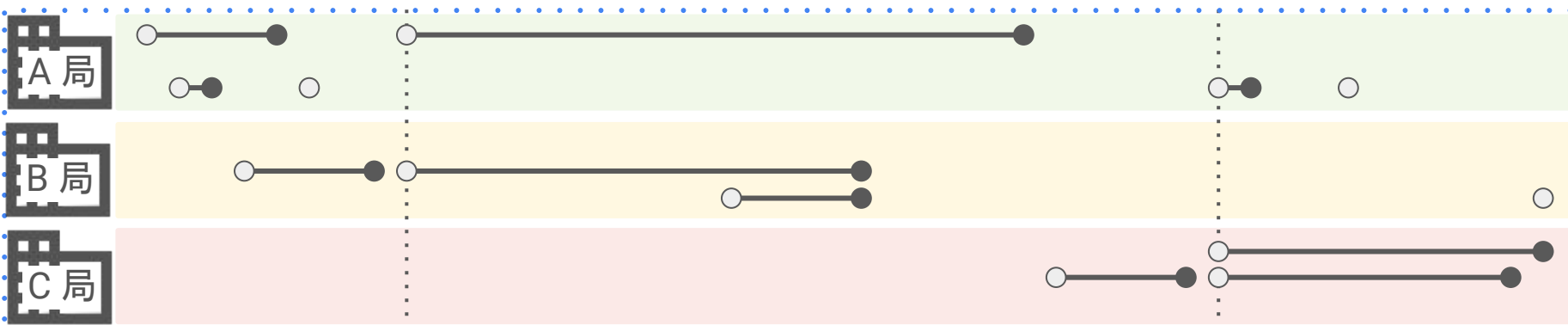
t+1

視聴時刻



分割前

分割後



TV 局の ID (0x0000 ~ 0xFFFF) を Integer Range Partition できるように変換

その他の加工

- クラスタリング用途
 - 視聴データの時刻 (0~23 時)
 - 受信機ごとの ID (16 進数) の 1 文字目をカラムに抽出
- クレンジング目的
 - 重複データや情報の欠損したデータの削除
 - 異常値 (長過ぎ、古過ぎ、何故か未来、アベコベなデータ) の除外
 - 桁数不足や表記揺れの補正
 - 追加情報の処理 (JSON 等で送られてくる情報など)

利用者 (TV 局) に応じたデータの抽出と加工

GCP 外

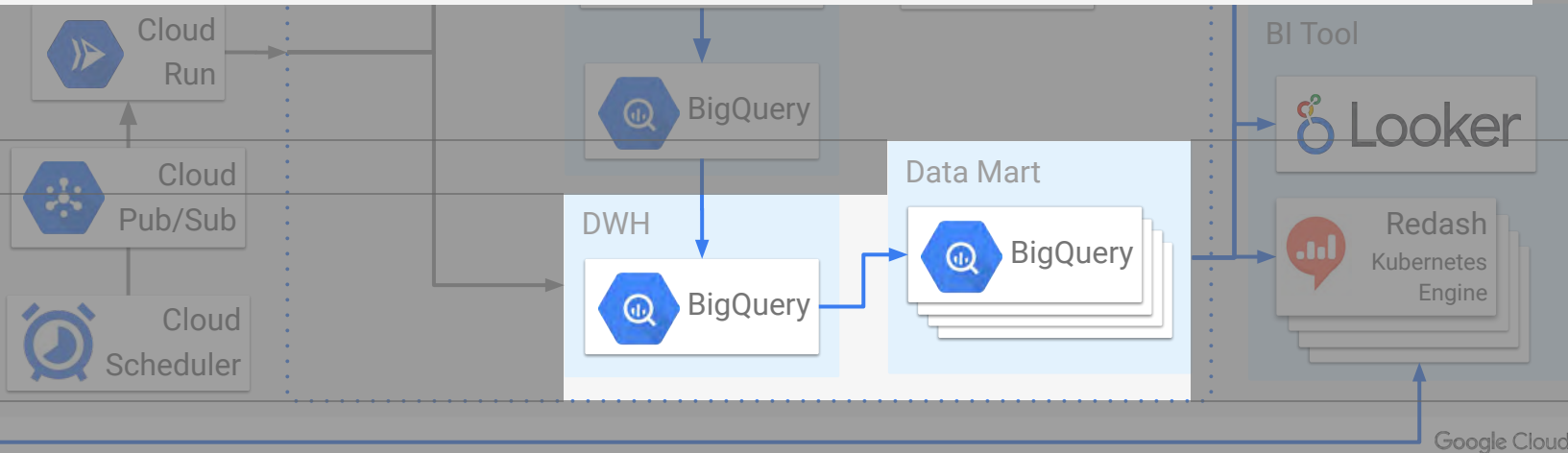
Google Cloud

視聴

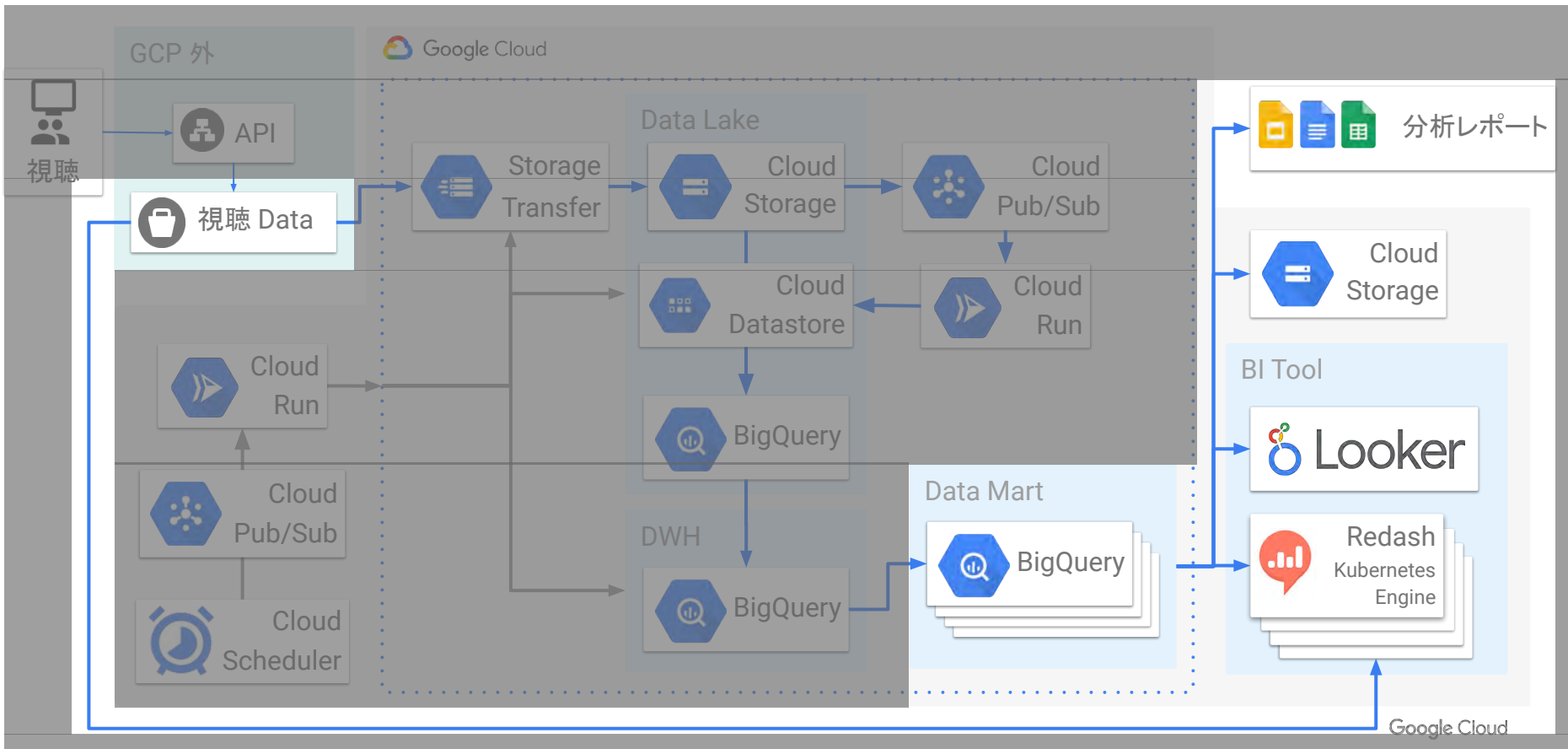
欲しい情報も番組や分析内容の変化で次第に変わりにえるため、極力View を使うことで柔軟性を持たせてる。

View では要件を満たせない場合のためにユースケースに応じた二次的加工もできるようにしている。

ポート



データの提供



ETL で利用している GCP 機能のおまとめ

- Storage Transfer: クラウドベンダ跨ぎのデータ転送が楽にできる
- Cloud Pub/Sub: Error Tolerance があるので安心して連携に使える
- Cloud Scheduler: バッチサーバー保守が要らないのありがたい
- BigQuery
 - 言わずもがな。View も有用。_TABLE_SUFFIX ちょっとツライ
 - コストを抑える方法が日々増えているのが頼りたくなる要因かな
- Cloud Run: 実行環境自分で選定できるのが嬉しい
- GKE: 言わずもがな
- Container Registry: 同上

実際の視聴データの 利活用イメージ

森藤 大地 (@muddydixon)

株式会社 TVer Technologies
データユニット Chief Data Scientist

データエンジニアリング・データ分析・
統計処理・機械学習・可視化・課題抽出・
レポート作成まで、データにまつわる業務を手広
く扱っています。

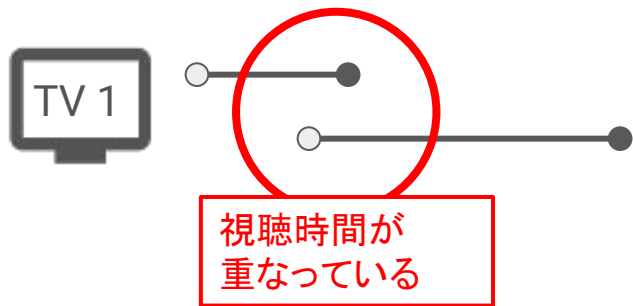
個人情報保護法に関して、個人特定にならない
範囲で非個人情報をいかに分析するかが最近の
主な関心事です。



前処理

- メーカーによる動作の差異を調整するために、下記のような前処理が必要
 - ログの不正值の除外・補正
 - 視聴時刻の除外・補正
 - 視聴データとしての除外・補正
- 上記処理に大規模データに対する分析関数が必要
 - BigQuery は大規模な分析関数を効率的に処理できるの実時間で結果を得られる

前処理 サンプル



一方、または両方を消すか、3つ以上重なっていたときの処理などは決め
現状は重なったら全部消す

```
SELECT
  tvid,
  `from`,
  `to`,
FROM (
  SELECT
    tvid,
    `from`,
    `to`,
    (
      LAG(`to`) OVER(PARTITION BY tvid ORDER BY `from`, `to`) IS NOT NULL AND
      TIMESTAMP_DIFF(`from`, LAG(`to`) OVER(PARTITION BY tvid ORDER BY `from`, `to`), SECOND) < 0
    )
  ) OR (
    LEAD(`from`) OVER(PARTITION BY tvid ORDER BY `from`, `to`) IS NOT NULL AND
    TIMESTAMP_DIFF(LEAD(`from`) OVER(PARTITION BY tvid ORDER BY `from`, `to`), `to`, SECOND) < 0
  ) AS is_overlap
FROM
  audience_logs
)
WHERE
  NOT is_overlap
```

大規模なレコードに対して分析関数を利用しても実時間で十分動作する！

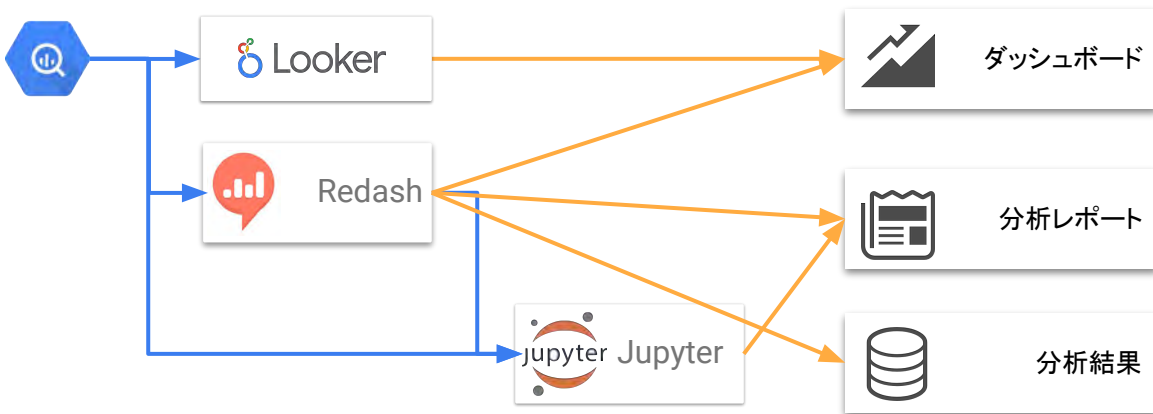
SQL サンプル

分析の種類と流れ

基本的にはBQ -> 各種 BI ->
詳細・複雑な分析は
Jupyter (Python 3)で実施

成果物は現状下記

1. ダッシュボード
2. 分析レポート
3. 分析結果提供



「データの民主化」という程には進んでいないのが現状

顧客がデータで何を解決したいか、を明文化できていないという前提での
課題抽出とアウトプットのインピーダンスマッチングが非常に重要

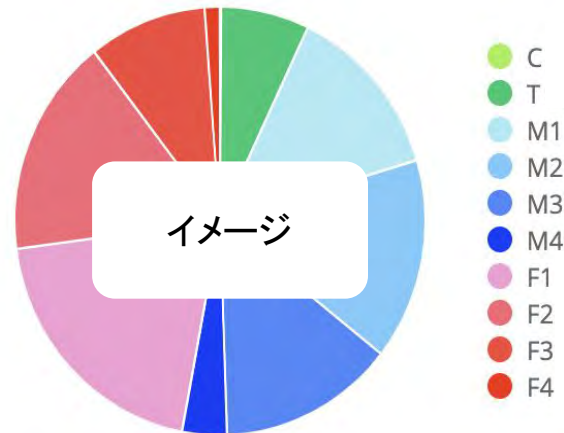
一般的な観点とテレビ特有の観点

一般的な観点

- 通常の状態をいつも把握できるようにする
 - パネルの特性(視聴者セグメント)の比率など
- 番組ごと・時間帯ごと・曜日ごとにパネルとどれくらい違うか、という比較が重要

テレビ特有の観点

- テレビは買い換えられる
 - パネルデータは賞味期限がある
- 常に新しい視聴データを元にした集計を実施する必要がある
- 視聴データはサイズが大きいので都度実行したくない



➡ Materialized View の効率的な活用

<https://cloud.google.com/bigquery/docs/materialized-views-intro?hl=ja>

Redash と Looker

- 使い分け
 - Redash は主に Adhoc な分析やデータ集計/分析において内部で利用
 - Looker は分析軸の深掘りや第三者意見を求める目的で利用
 - ダッシュボードとしての機能は両方とも備わっている
- GCP の Service Account を適切に分けて Redash / Looker に登録することで、費用計算がかんたんに行える
 - 各放送局に請求を行うため
- BigQuery Audit Logs として出力してどのテーブル・誰が、なども見ている
 - Audit Logs 、STRUCT / REPEATED が多く読み解く・クエリ化はツライ

統計処理・機械学習

- 統計処理・機械学習等、最後のところまではBigQuery ML をフル活用
 - LogisticRegression / TensorFlow モデルのインポート
 - 視聴率予測、視聴者嗜好の分類など
- ➡ GCP Notebooks や Sage なんちゃらだと、結局インスタンス管理をしないといけないので、現在の人員では無理ゲー
BigQuery ML はフルマネージドなので非常に使いやすいし管理リソースがかからない！
- ➡ もちろん、いかにしてBQに寄せるか、は知恵の絞りどころ

分析時の BigQuery の機能のおまとめ

- 分析ツール・用途に応じた Service Account の使い分けで費用計算がかんたん
- AuditLogs は重要だけど、クエリ作成ツライ
- Materialized View は基本値の記録に有用
- 分析関数ゴリゴリ回して動作するのがすごい(LAG/LEAD だけではなく SUM/AVG/FIRSTVALUE/LASTVALUE など使っています)
- データエンジニア・機械学習インフラエンジニアが不足している弊社は、マネージドの統計処理・機械学習環境として非常に効果的



We are hiring
TVer Technologies
Innovate TVision

Google Cloud

Thank you