

aws **DEV DAY**
ONLINE JAPAN

DEV DAY

20-22.10.2020

In Partnership with **intel**

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

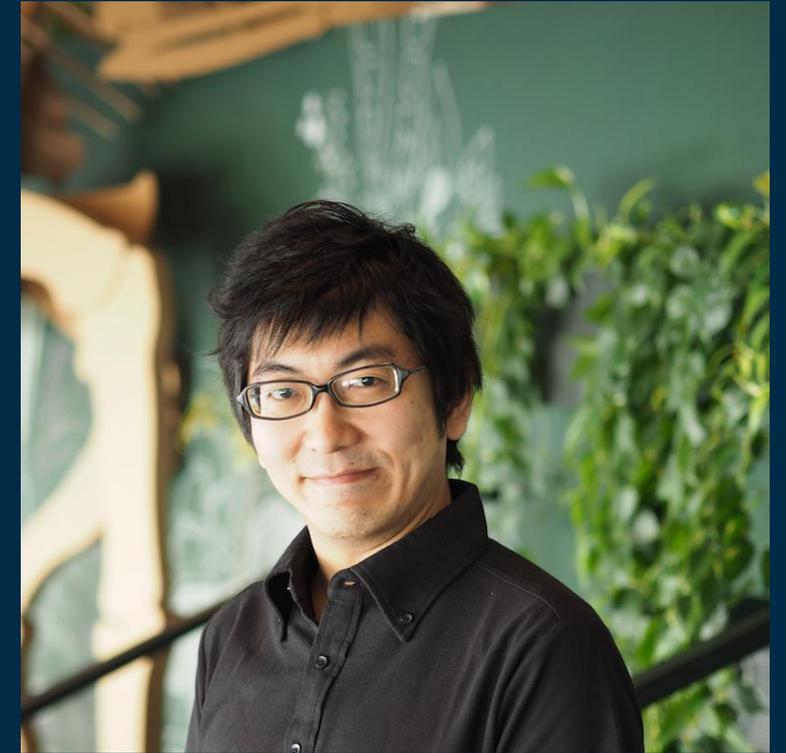
Purpose-Build Database Advanced Design Pattern

Solutions Architect
Akihiro Kuwano (twitter: @kuwa_tw)

自己紹介

桑野 章弘(くわの あきひろ)

- ソリューションアーキテクト
- 主にメディア系のお客様を担当しております
- 好きなAWSのサービス
 - Amazon Route53, Amazon S3, Amazon Aurora
 - Amazon DocumentDB



本セッションについて

対象の方

- AWSには触っていただいている、データベースサービスもちょっと知っている
- どのデータベースを使い分けたいかわからない

今日持ち帰っていただくこと

- AWSのデータベースサービスの使い分け方の指針について理解できる

話さないこと

- RDBMSのお話
- サービス毎の細かいお話



Agenda

AWS の Purpose-Built Database

適切なデータベースを選択する方法

Working backwards...クエリを想定したデザイン

アプリケーションによるデータベースの選定例

まとめ



AWS の Purpose-Built Database

AWSのデータベースサービスをご存知でしょうか?

Aurora?



AWSのデータベースサービスをご存知でしょうか?

Aurora?

DynamoDB!



AWSのデータベースサービスをご存知でしょうか?

Aurora?

ElastiCache!

DynamoDB!



AWSのデータベースサービスをご存知でしょうか?

DocumentDB

QLDB

Aurora?

Keyspaces

ElastiCache!

Timestream

Neptune

DynamoDB



AWSのデータベースサービスをご存知でしょうか?

DocumentDB

QLDB

Aurora?

Keyspaces

ElastiCache!

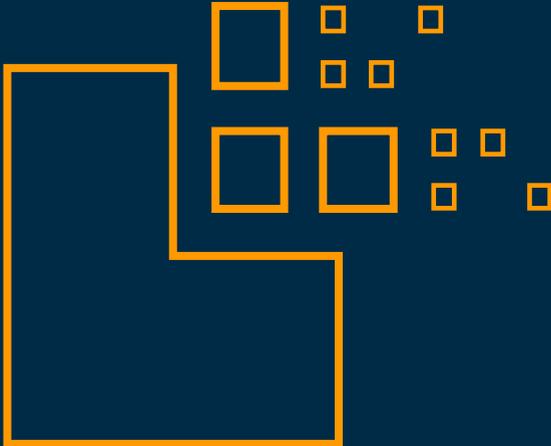
実はいっぱいあります

Timesreams

DynamoDB!

Neptune

Purpose built



Purpose-built databases

Relational

Key-value

Document

In-Memory

Graph

Time-Series

Ledger

Wide Column

リレーショナル

ドキュメント

グラフ

台帳

キーバリュー

インメモリー

時系列

ワイドカラム



Purpose-built databases

Relational

Key-value

Document

In-Memory

Graph

Time-Series

Ledger

Wide Column



Amazon Aurora



Amazon RDS



Amazon DynamoDB



Amazon DocumentDB



Amazon ElastiCache



Amazon Neptune



Amazon Timestream



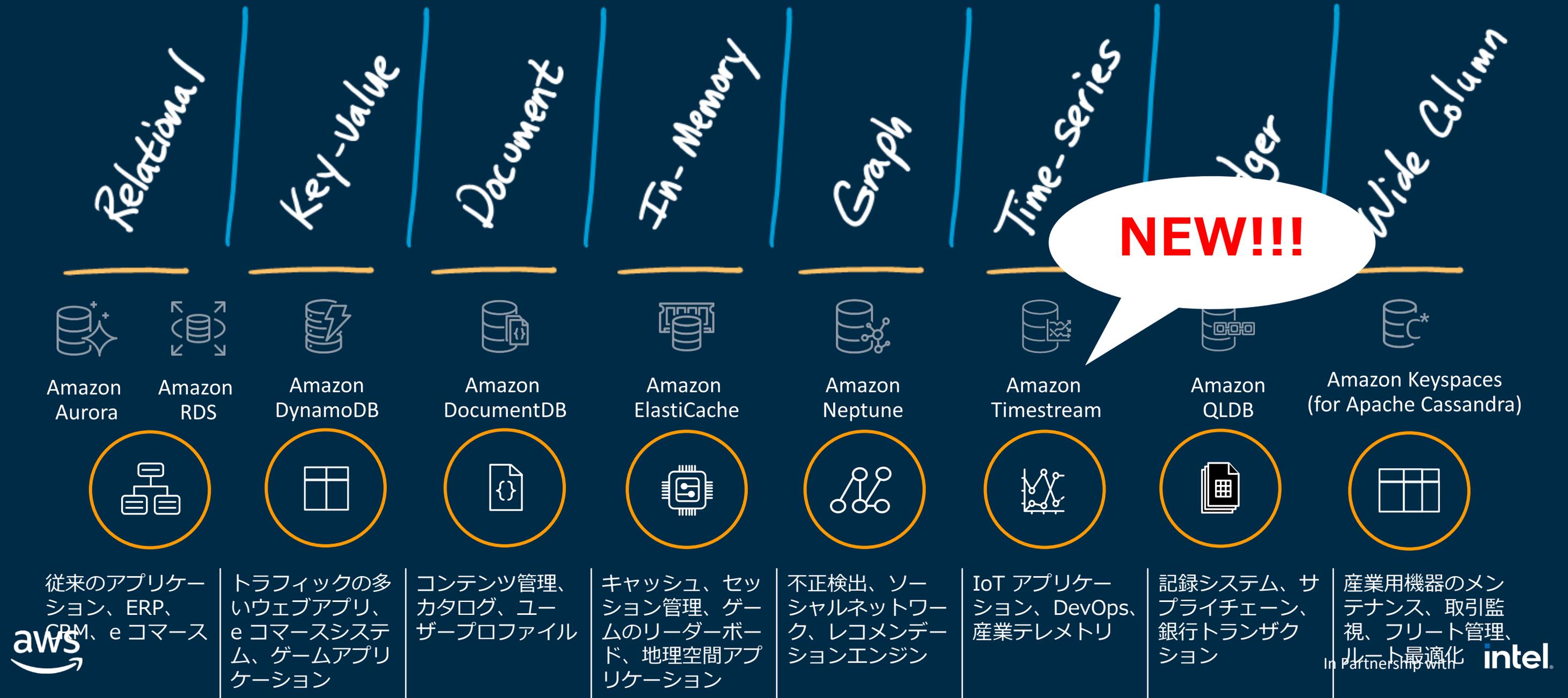
Amazon QLDB



Amazon Keyspaces
(for Apache Cassandra)



Purpose-built databases

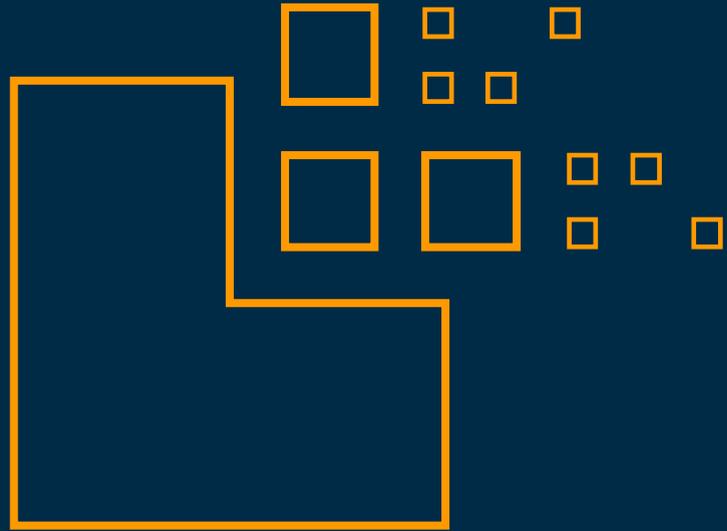


In Partnership with 

When You're a Hammer, Everything Looks Like a Nail



データベースの選択



Purpose built

The right tool for
the right job

AWS では多様な
データベースの選択肢

ワークロードに応じて
最適な選択が可能

適材適所の選択

<https://www.allthingsdistributed.com/2018/06/purpose-built-databases-in-aws.html>



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with 

Amazon.com はどのような選択をしたのか

Oracle databases



AWS Purpose-Built Databases



Relational



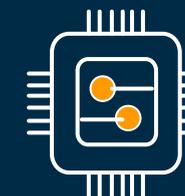
Document



Graph



Key-value



In-memory



Data warehouse

コスト削減、パフォーマンス向上、より速い革新を実現するために、2016年からすべてのOracleをAWSに移行開始

Purpose-Built Databases によってワークロードに最適なDBエンジンを提供でき、コストとユーザー体験の最適化を実現

適切なデータベースを選択する方法

適切なデータベースを選択する方法

まず最初に作るアプリケーションにどのような要件が存在するのかをブレイクダウンする

項目の例

- アクセスパターン
- データ量
- スケールパターン
- ユースケース
- グラフDBや、台帳DBの様なユースケースが明確なものは決めやすい
- 機能要件
- エンジニアのスキルセット
- などなど

あるアプリケーションの要件その1



Social media

ユーザー: 1,000 以上

データ量: GB

場所: リージョナル

パフォーマンス: ミリ秒

リクエスト: 数千/sec

アクセス: ウェブ

拡張: スケールアップ/ダウン

開発者アクセス: ドライバアクセス

あるアプリケーションの要件その2



Ride hailing



Media streaming



Social media



Dating

ユーザー: 1,000,000 以上

データ量: PB

場所: グローバル

パフォーマンス: ミリ秒

リクエスト: 数百万/sec

アクセス: ウェブ/モバイル

拡張: スケールアウト/イン

開発者アクセス: APIアクセス

そしてもう一つ大事なことは、、、



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with  intel.

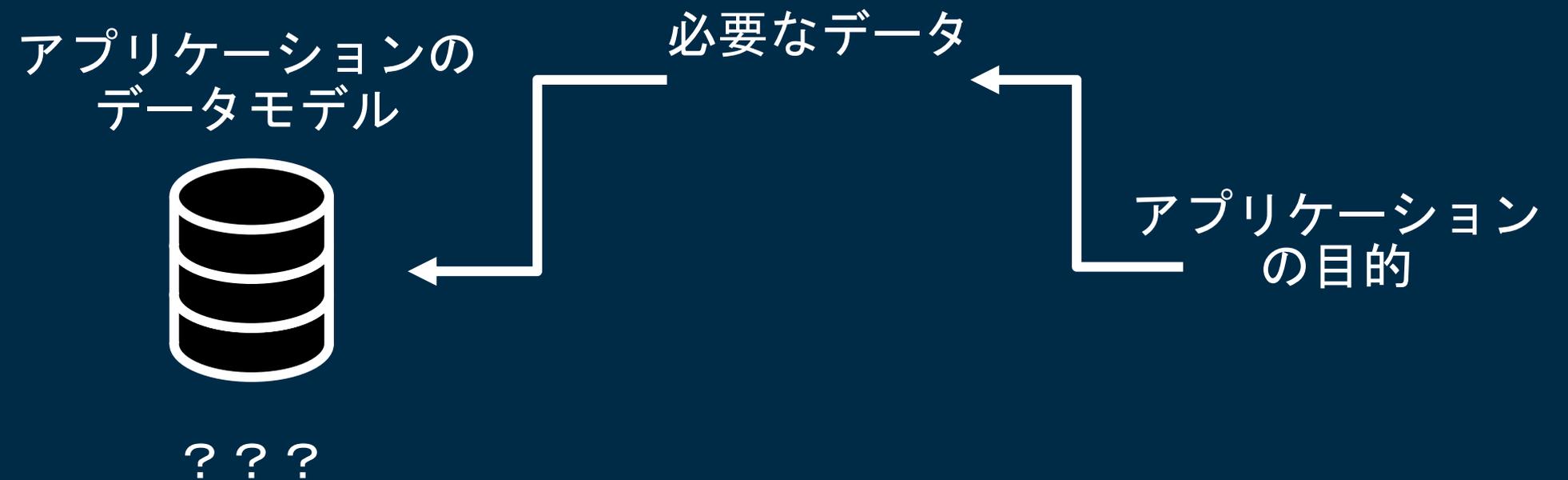
Working backwards ...クエリを想定したデザイン

Working backwards – クエリを想定したデザイン

アプリケーション
の目的

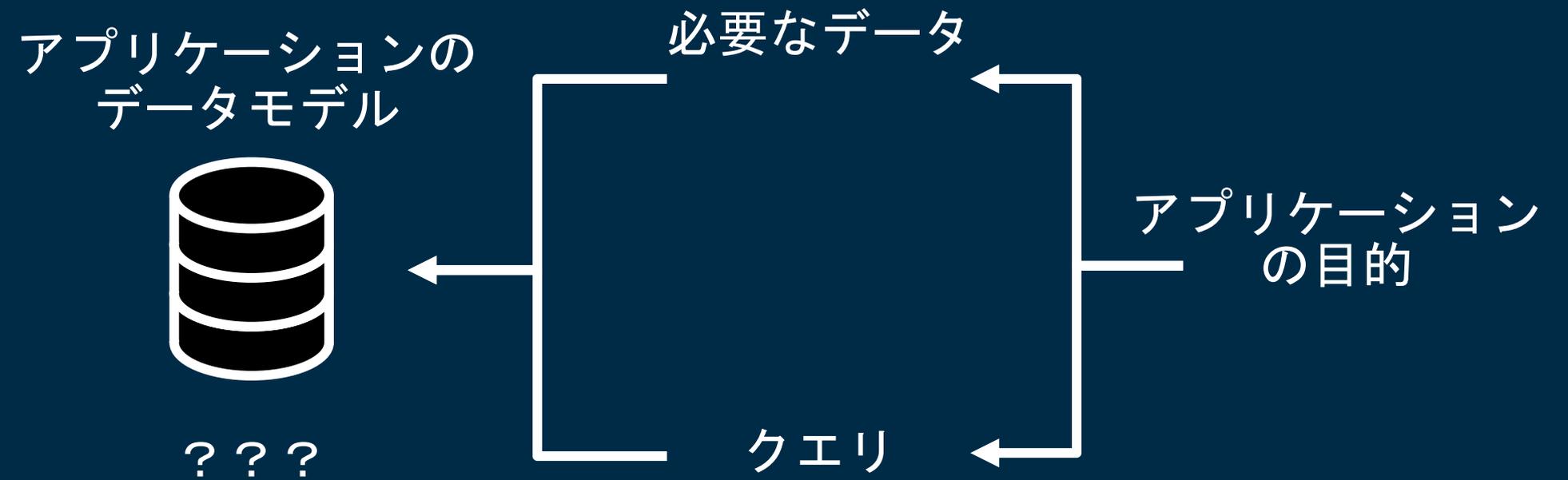


Working backwards – クエリを想定したデザイン



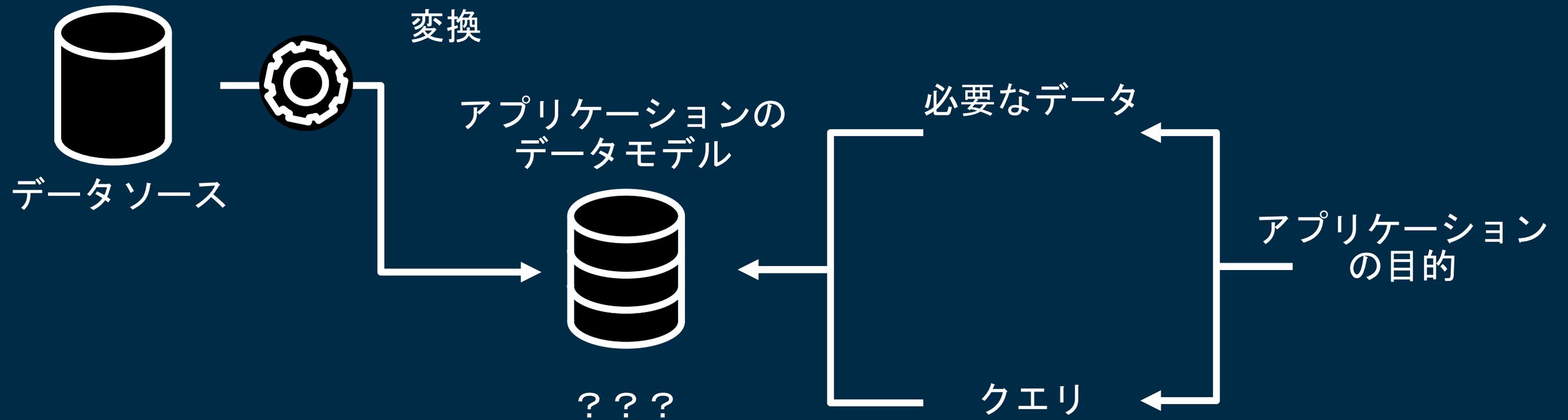
目的を達成するためにはどのようなデータが必要なのか？

Working backwards – クエリを想定したデザイン



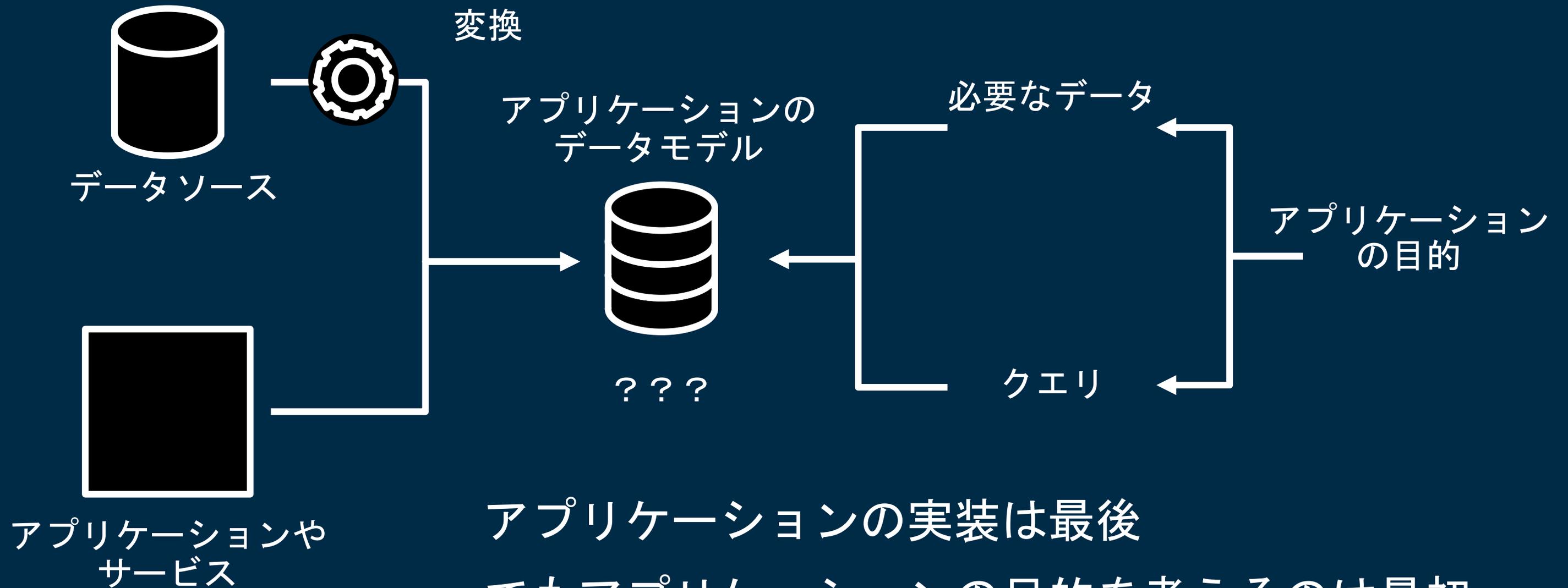
そしてどのようなクエリが必要となるのか？

Working backwards – クエリを想定したデザイン



最後にデータソースからの変換を考える

Working backwards – クエリを想定したデザイン



アプリケーションの実装は最後までアプリケーションの目的を考えるのは最初

アプリケーションによる データベースの選定例

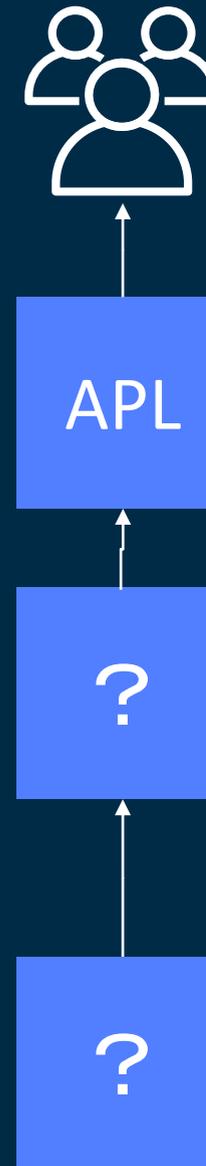
アプリケーションとデータベース

では実際にアプリケーションの要件を決めつつ選定を行ってみましょう



コンテンツ管理システム

- ニュースサイトなど、文書などのコンテンツ管理を行うシステム
- 不定形なデータ構造
- コメントなどをつけることができる
- タグで分類
- 著者名も管理



Amazon Web Services ブログ

新機能 – Redis 6 互換の Amazon ElastiCache

by Channy Yun | on 13 OCT 2020 | in [Open Source](#) | [Permalink](#) | [Share](#)

最新の [Redis 5.0 互換の Amazon ElastiCache](#) 以降、[5.0.6](#) でのアップストリームサポートを含め、[Amazon ElastiCache for Redis](#) では多くの改善が行われました。

今年の初め、1 つのリージョンのクラスターを最大 2 つの他のリージョンのクラスターにレプリケートできる [Datastore for Redis](#) 機能を発表しました。最近では、[18 個のエンジンおよびノードレベルの CloudWatch](#) を追加することで、Redis フリートをモニタリングする機能が向上しました。[リソースレベルのアクセス許可](#) のサポートも追加され、特定の ElastiCache リソースに [AWS Identity and Access Management \(IAM\)](#) ルアクセス許可を割り当てることができるようになりました。

そして本日、[Redis 6 互換の Amazon ElastiCache for Redis](#) を発表することになりました。今回のリリースで、[Amazon ElastiCache for Redis](#) にいくつかの新機能や重要な機能が追加されています。

- [マネージドロールベースのアクセス制御](#) – Amazon ElastiCache for Redis 6 では、Redis コマンドベースの [アクセス制御 \(RBAC\)](#) の設定に使用できるユーザーとユーザーグループを作成および管理できるようになりました。複数のアプリケーションで、互いのデータにアクセスすることなく、同じ Redis クラスターを共有する

アプリケーション要件の整理

- 条件を列挙し、まずは右のようなアクセス要件を考える

ユーザー: 1,000,000 以上

データ量: TB

場所: リージョナル

パフォーマンス: ミリ秒

リクエスト: 数万/sec

アクセス: ウェブ/モバイル

拡張: スケールアップ/ダウン、スケールアウト/イン

開発者アクセス: APIアクセス



アプリケーション要件の整理

- 条件を列挙し、まずは右のようなアクセス要件を考える
- 更に以下の様な要件も考えてみる
 - 不定形なデータを入れられる
 - ピークはあるが予期しないスパイキーなアクセスはない

ユーザー: 1,000,000 以上

データ量: TB

場所: リージョナル

パフォーマンス: ミリ秒

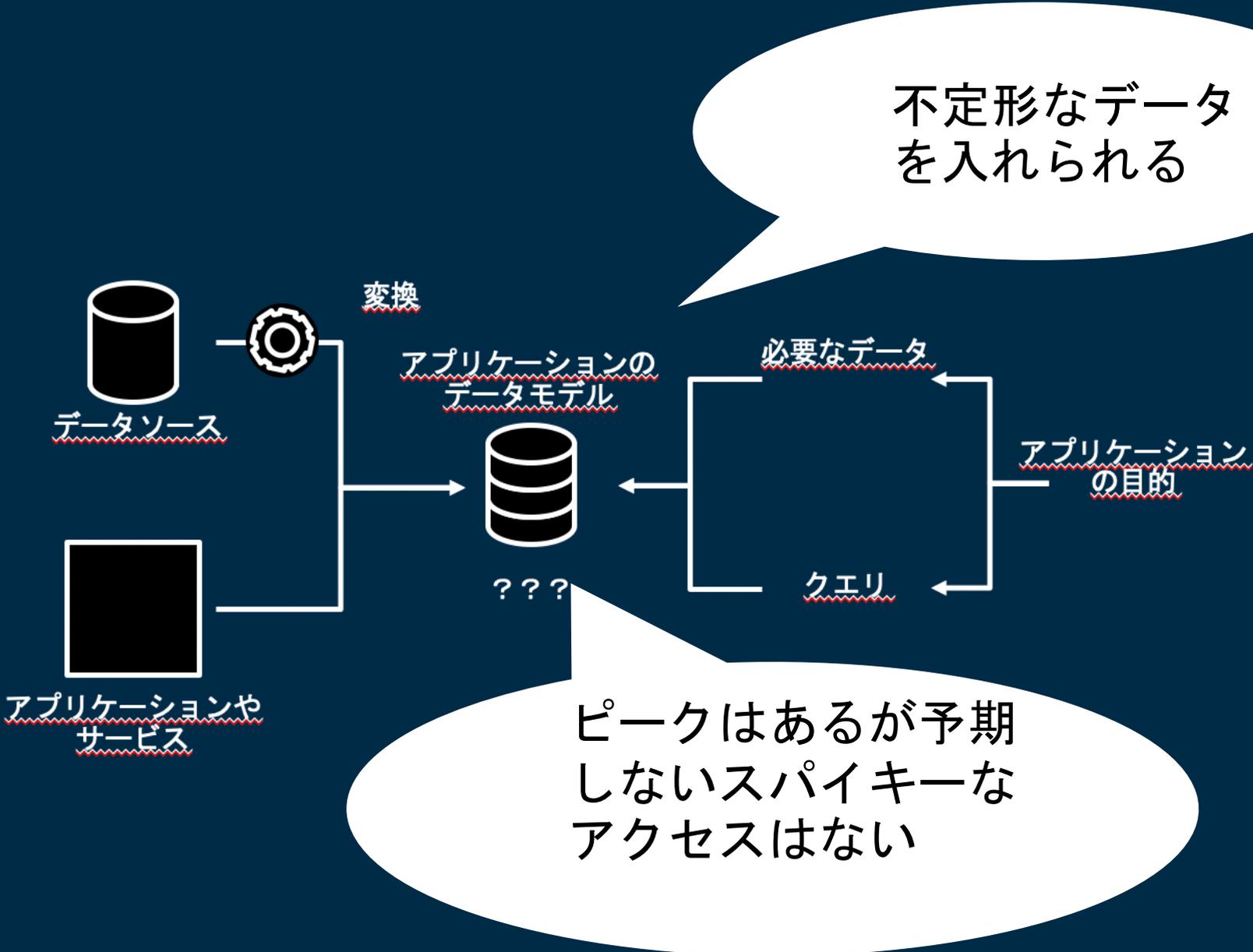
リクエスト: 数万/sec

アクセス: ウェブ/モバイル

拡張: スケールアップ/ダウン、スケールアウト/イン

開発者アクセス: APIアクセス

Working backwards してみる



ユーザー: 1,000,000 以上

データ量: TB

場所: リージョナル

パフォーマンス: ミリ秒

リクエスト: 数万/sec

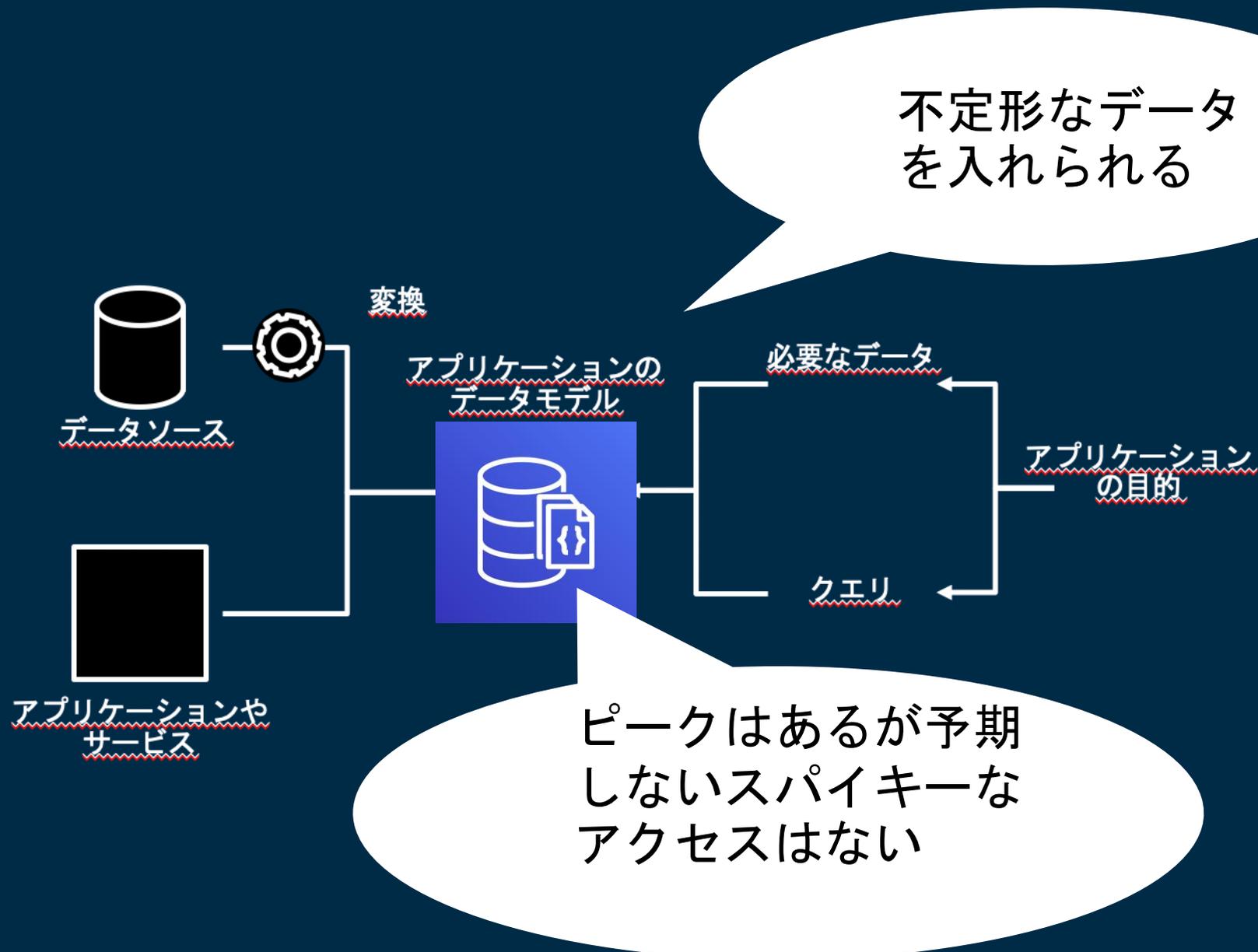
アクセス: ウェブ/モバイル

拡張: スケールアップ/ダウン、スケールアウト/イン

開発者アクセス: APIアクセス



Working backwards してみる



ユーザー: 1,000,000 以上

データ量: TB

場所: リージョナル

パフォーマンス: ミリ秒

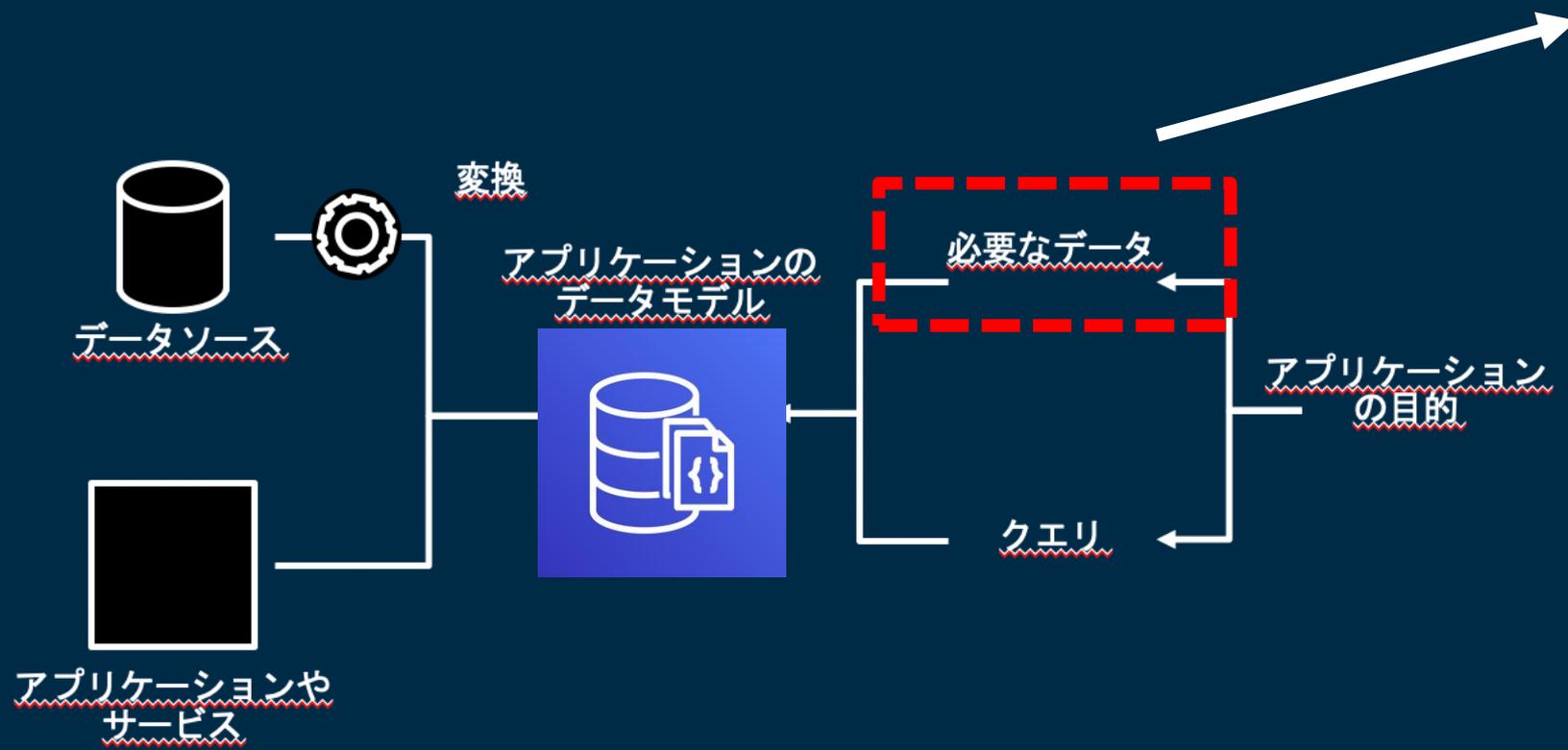
リクエスト: 数万/sec

アクセス: ウェブ/モバイル

拡張: スケールアップ/ダウン
、スケールアウト/イン

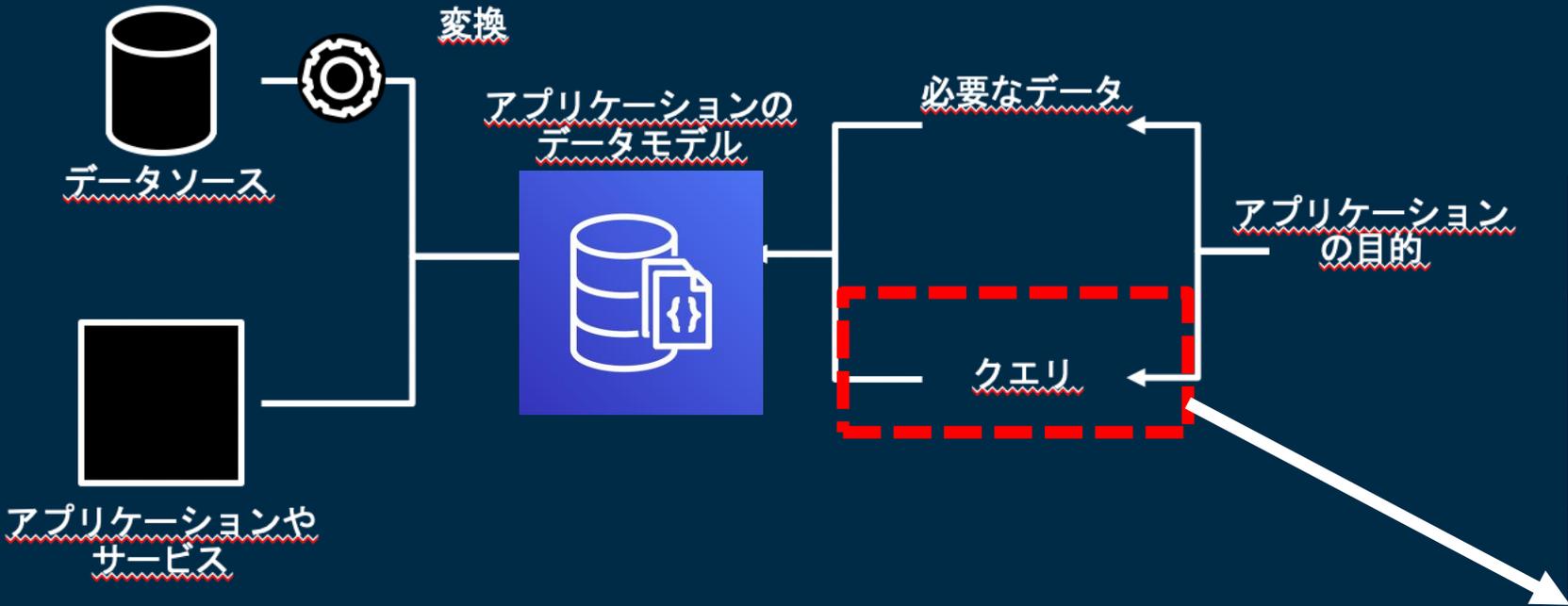
開発者アクセス: APIアクセス

Working backwards してみる



項目	内容
記事ID	abcde
作成日時	2020/10/21 00:00:00
執筆者	Akihiro Kuwano
タグ	blog
本文	<テキスト>
(コメント)	<テキスト>

Working backwards してみる



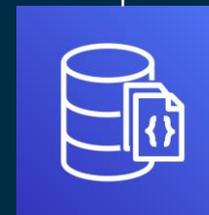
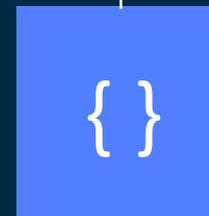
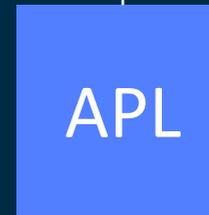
項目	内容
記事ID	abcde
作成日時	2020/10/21 00:00:00
執筆者	Akihiro Kuwano
タグ	blog
本文	<テキスト>
(コメント)	<テキスト>

- ## クエリ
- 記事IDにマッチした記事を取得する
 - 作成日時でソートしてXX件表示
 - 執筆者で検索
 - タグで検索
 - 記事IDについてのコメント一覧の取得



結果：コンテンツ管理システム

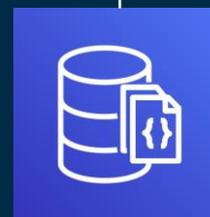
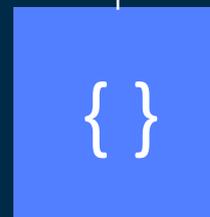
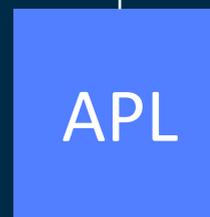
- ニュースサイトなど、文書などのコンテンツ管理を行うシステム
- 不定形なデータ構造でも対応可能
- 文書ごとの関係性や、タグ付け、コメントなども表現しやすい
- 文書の簡易検索は \$regex などでも利用可能
- スケールも可能
- DocumentDBの開発効率の良さが活きる



```
{  
  _id: abcde,  
  title: "My Blog",  
  created: ISODate("2020-08-26..."),  
  author: { _id: "kuwanoa", name: 'Akihiro  
Kuwano' },  
  tags: [ "blog", ... ],  
  detail: { text: "テキスト..." }  
}
```

結果：コンテンツ管理システム

- ニュースサイトなど、文書などのコンテンツ管理を行うシステム
- 不定形なデータ構造でも対応可能
- 文書ごとの関係性や、タグ付け、コメントなども表現しやすい
- 文書の簡易検索は \$regex などにも利用可能
- スケールも可能
- DocumentDBの開発効率の良さが活きる



```
{
  _id: abcde,
  title: "My Blog",
  created: ISODate("2020-08-26..."),
  author: { _id: "kuwanoa", name: 'Akihiro Kuwano' },
  tags: [ "blog", ... ],
  detail: { text: "テキスト..." },
  comment: [
    {
      author: { name: "Takashi" },
      created: ISODate("2020-08-26..."),
      detail: { text: "コメント..." }
    },
    {
      author: { name: "Zabbio" },
      created: ISODate("2020-08-26..."),
      detail: { text: "コメント..." }
    }
  ]
}
```

ちょっとまった！



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with  intel.

ここで条件がもし変わったらどうなるでしょうか
同じアプリケーションだけどちょっとだけ条件を変えてみましょう



アプリケーション要件の整理：再び

- アクセス要件は一箇所を除いて同じです

ユーザー: 1,000,000 以上

データ量: TB

場所: リージョナル

パフォーマンス: ミリ秒

リクエスト: 数百万/sec

アクセス: ウェブ/モバイル

拡張: スケールアップ/ダウン、スケールアウト/イン

開発者アクセス: APIアクセス

アプリケーション要件の整理：再び

- アクセス要件は一箇所を除いて同じです
- 更に以下の様な要件も考えてみる
 - 不定形なデータを入れられる
 - **ピークはあるが予期しないスパイキーなアクセスがある**
 - **結果リクエスト数も多い**

ユーザー: 1,000,000 以上

データ量: TB

場所: リージョナル

パフォーマンス: ミリ秒

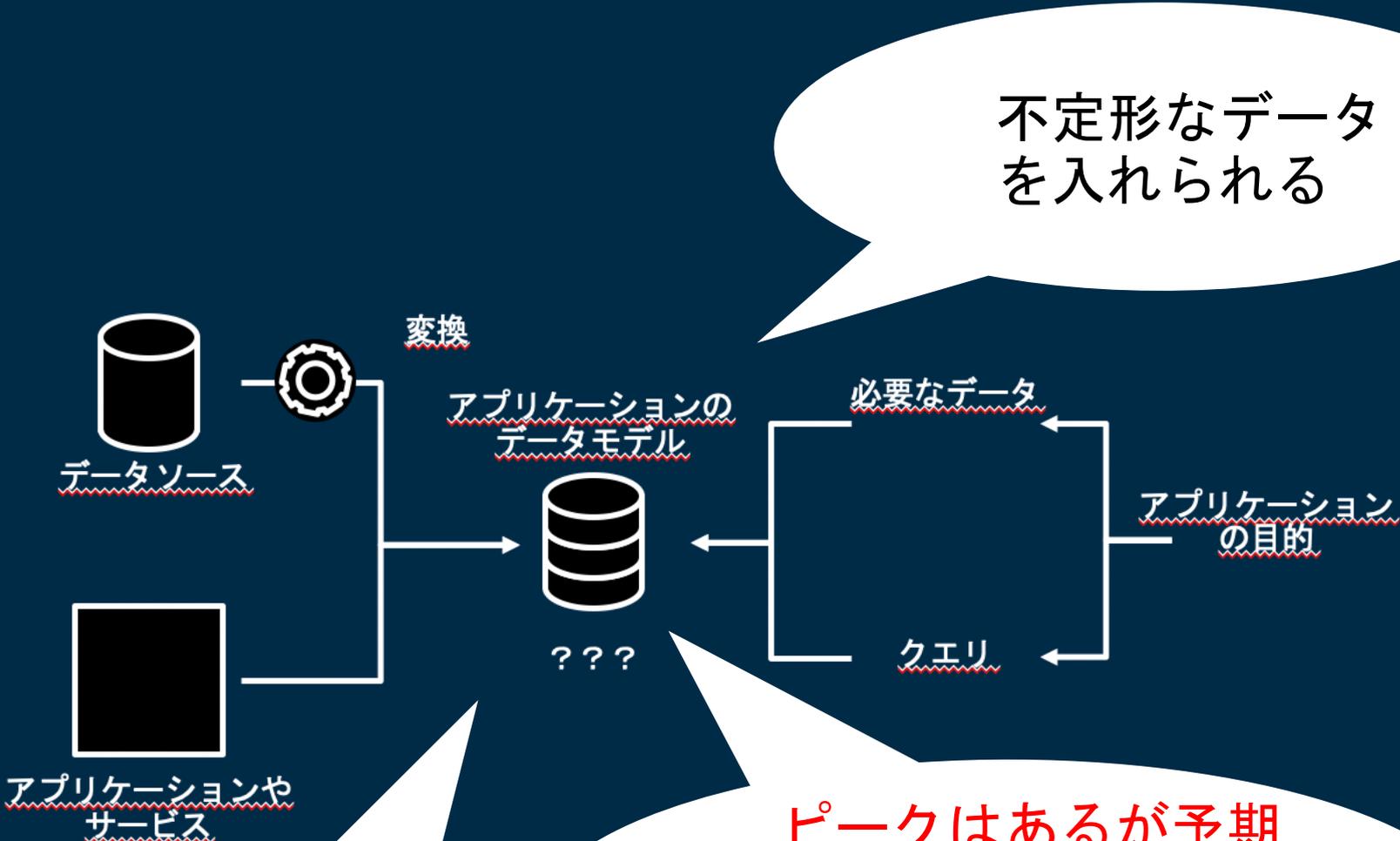
リクエスト: 数百万/sec

アクセス: ウェブ/モバイル

拡張: スケールアップ/ダウン、スケールアウト/イン

開発者アクセス: APIアクセス

Working backwards してみる : 再び



不定形なデータ
を入れられる

ユーザー: 1,000,000 以上

データ量: TB

場所: リージョナル

パフォーマンス: ミリ秒

リクエスト: 数万/sec

アクセス: ウェブ/モバイル

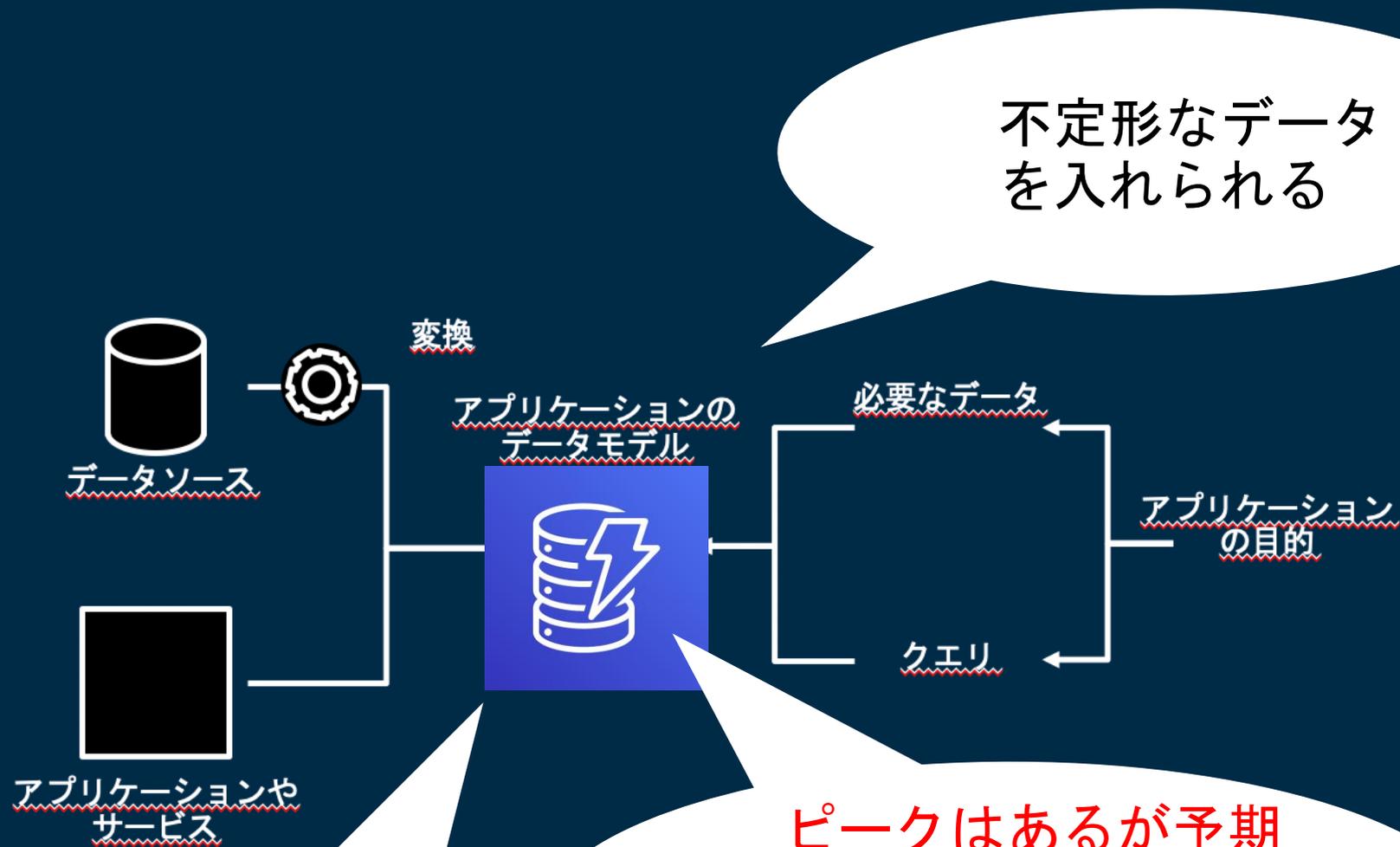
拡張: スケールアップ/ダウン
、スケールアウト/イン

開発者アクセス: APIアクセス

結果リクエスト
数も多い

ピークはあるが予期
しないスパイクな
アクセスがある

Working backwards してみる : 再び



ユーザー: 1,000,000 以上

データ量: TB

場所: リージョナル

パフォーマンス: ミリ秒

リクエスト: 数万/sec

アクセス: ウェブ/モバイル

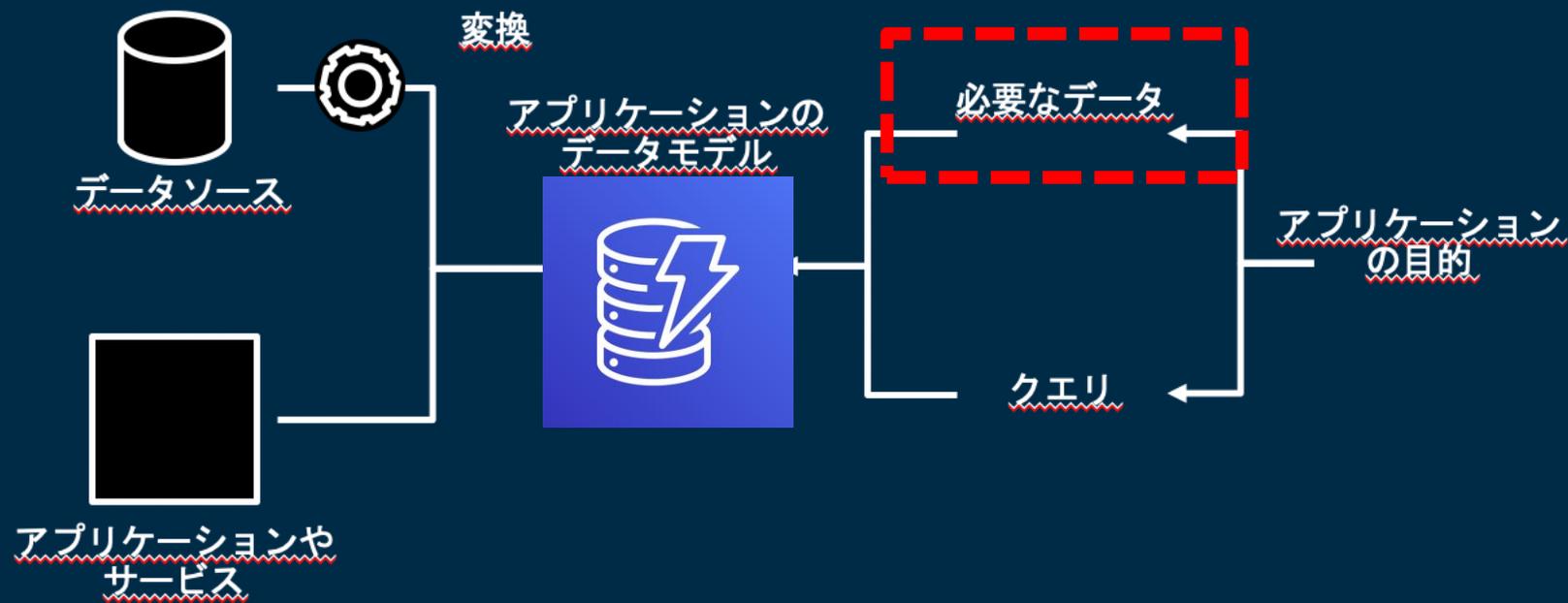
拡張: スケールアップ/ダウン
、スケールアウト/イン

開発者アクセス: APIアクセス

結果リクエスト
数も多い

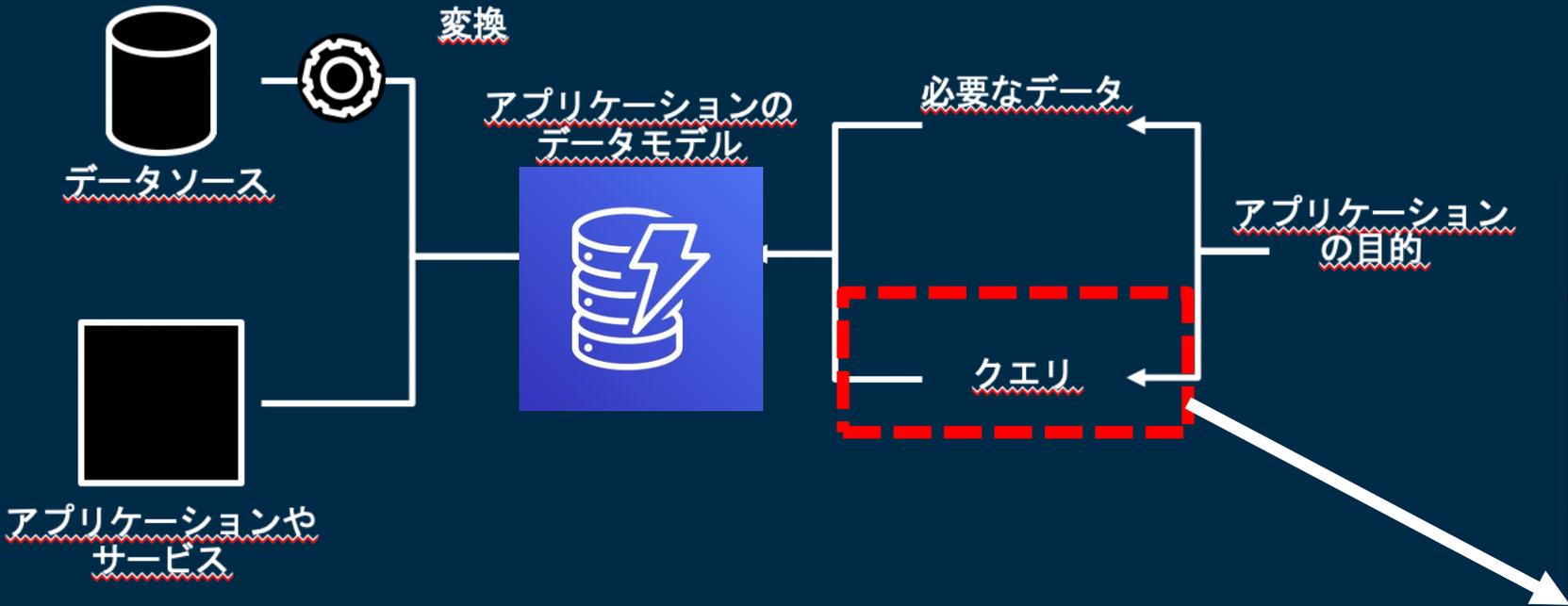
ピークはあるが予期
しないスパイキーな
アクセスがある

Working backwards してみる : 再び



項目	内容
記事ID	abcde
作成日時	2020/10/21 00:00:00
執筆者	Akihiro Kuwano
タグ	blog
本文	<テキスト>
(コメント)	<テキスト>

Working backwards してみる : 再び



項目	内容
記事ID	abcde
作成日時	2020/10/21 00:00:00
執筆者	Akihiro Kuwano
タグ	blog
本文	<テキスト>
(コメント)	<テキスト>

クエリ

記事IDにマッチした記事を取得する

作成日時でソートしてXX件表示

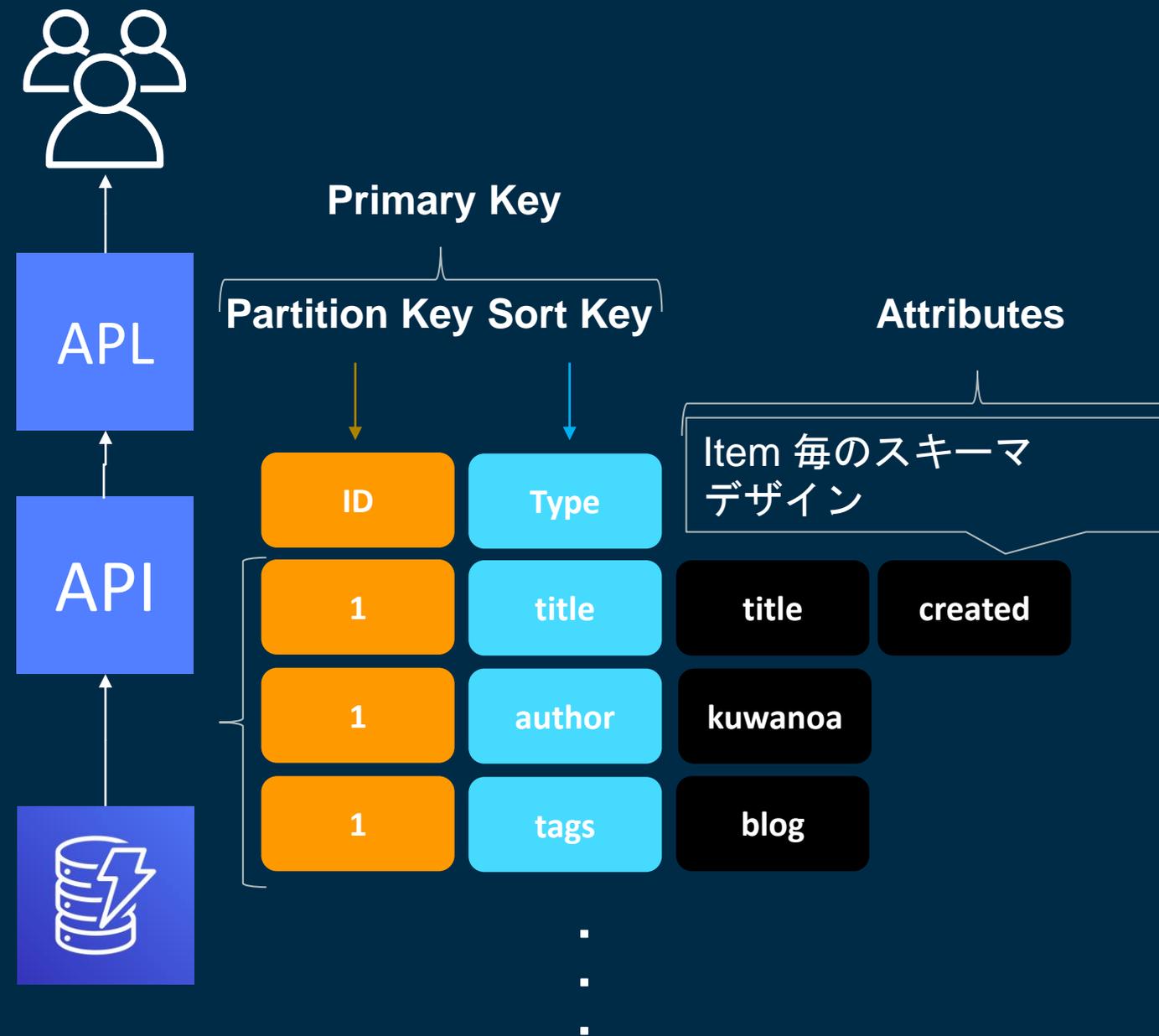
執筆者で検索

タグで検索

記事IDについてるコメント一覧の取得

コンテンツ管理システム

- ニュースサイトなど、文書などのコンテンツ管理を行うシステム
- 不定形なデータ構造でも対応可能
 - モデリングの手法は変わる
- スケールなどについての考慮事項が減る
- 検索などはDynamoDB StreamsでElasticsearchなどに振ってもいい



データベース選定のまとめ

- 要件をブレイクダウンしていき最終的に一番メリットを享受できるデータベースを選択する
 - 高い開発効率、低い運用負荷など
- 同じアプリケーションほぼ同じ要件でも何を重視するかで選択肢は変わる

まとめ

まとめ

- AWS の Purpose-Built Database の考えかたの基本は “適材適所”
- 最適なデータベースの選定にはまず “何を実現したいのか” から逆に考えていく
- 最適なデータベースを選択することで “高い開発効率” や “低い運用負荷” を実現する事が可能になる

このあたりの話を詳しく知りたい！

Purpose-Built Database Weekのアーカイブも御覧ください！

The screenshot shows the AWS website header with navigation links: お問い合わせ, サポート, 日本語, アカウント, and a button for 今すぐ無料サインアップ. Below the header is a navigation bar with links for 製品, ソリューション, 料金, ドキュメント, 学ぶ, パートナーネットワーク, AWS Marketplace, カスタマー支援, イベント, and さらに詳しく見る. The main content area features a large banner with the AWS logo and the text "PURPOSE-BUILT DATABASES WEEK ONLINE". Below the banner is the text "AWS の NoSQL データベースサービス総集編" and a "Watch On-Demand" button. At the bottom of the banner, there is a "FREE ONLINE EVENT" section with the text "AWS NoSQL Databasesに焦点を当てた1週間に渡る全5日間のイベントのアーカイブ動画を公開中。" and a box indicating the event dates "August 24(Mon) - August 28(Fri)" and "本セミナーは終了しました". The footer of the screenshot shows a file download bar with "[EXTERNAL]_(F...zip)" and a "Show All" button.



Thank you!

Speaker Name

Contact information



Please complete the session
survey in the mobile app.