Amazon Cognito
Amazon Mobile Analytics

AWS Black Belt Tech Webinar 2014 (旧マイスターシリーズ)

アマゾンデータサービスジャパン株式会社

今井雄太　西谷圭介

Agenda

Agenda

- AWSモバイルサービスのご紹介
- クライアントSDKのおさらい
- Amazon Cognito
  - Amazon Cognitoとは？
  - Amazon Cognito Identity Broker
  - Amazon Cognito Sync
  - Webアプリケーションからの利用
  - 料金
- Amazon Mobile Analytics
  - Amazon Mobile Analyticsとは？
  - レポート
  - カスタムイベント
  - 料金
- まとめ
- Q&A

Agenda

Agenda

- AWSモバイルサービスのご紹介
- クライアントSDKのおさらい
- Amazon Cognito
  - Amazon Cognitoとは？
  - Amazon Cognito Identity Broker
  - Amazon Cognito Sync
  - Webアプリケーションからの利用
  - 料金
- Amazon Mobile Analytics
  - Amazon Mobile Analyticsとは？
  - レポート
  - カスタムイベント
  - 料金
- まとめ

モバイルアプリ開発の課題

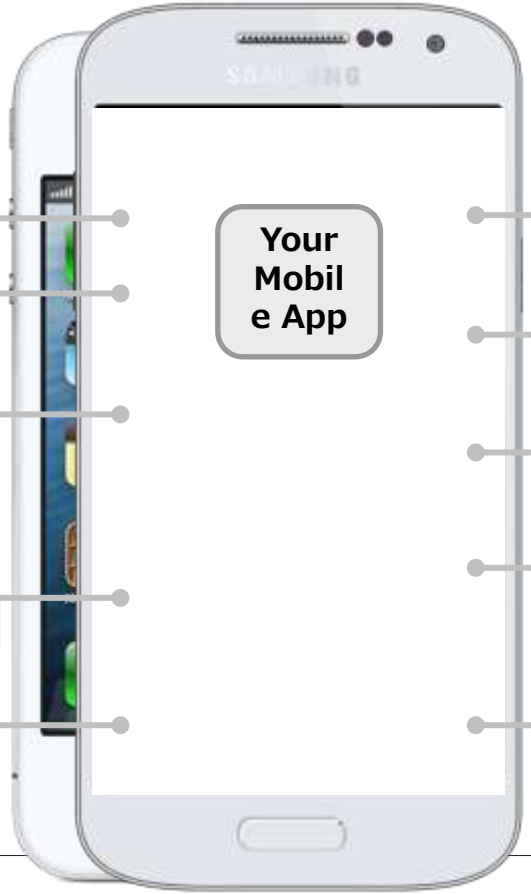
ユーザ認証
ユーザの管理や
IDプロバイダとの連携

アクセスの認可
クラウドリソースへの
セキュアなアクセス

データの同期
ユーザ設定等の複数
デバイス間での同期

ユーザの行動分析
アクティブユーザや
エンゲージメントの追跡

保持率の追跡
ファンネルやキャペーン
効果の管理



メディアの管理
ユーザが投稿した写真やその他
メディアの保存と共有

メディアの配信
モバイルデバイスの自動識別と
素早く、グローバルなコンテンツの配信

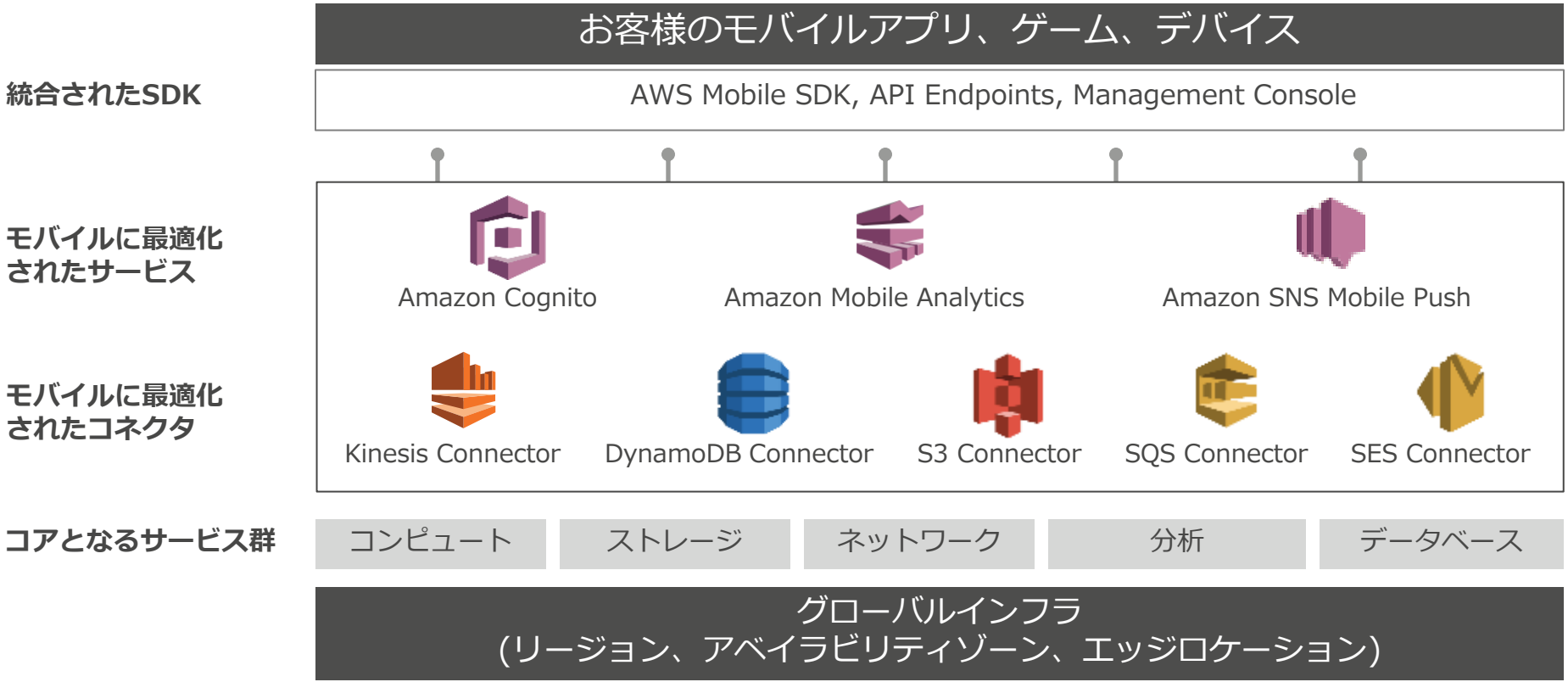
プッシュ通知の送信
ユーザをアクティブに保つための信頼性の高い
メッセージ送信

共有データの保存
複数ユーザ、デバイス間における共有データの
保存と高速な検索

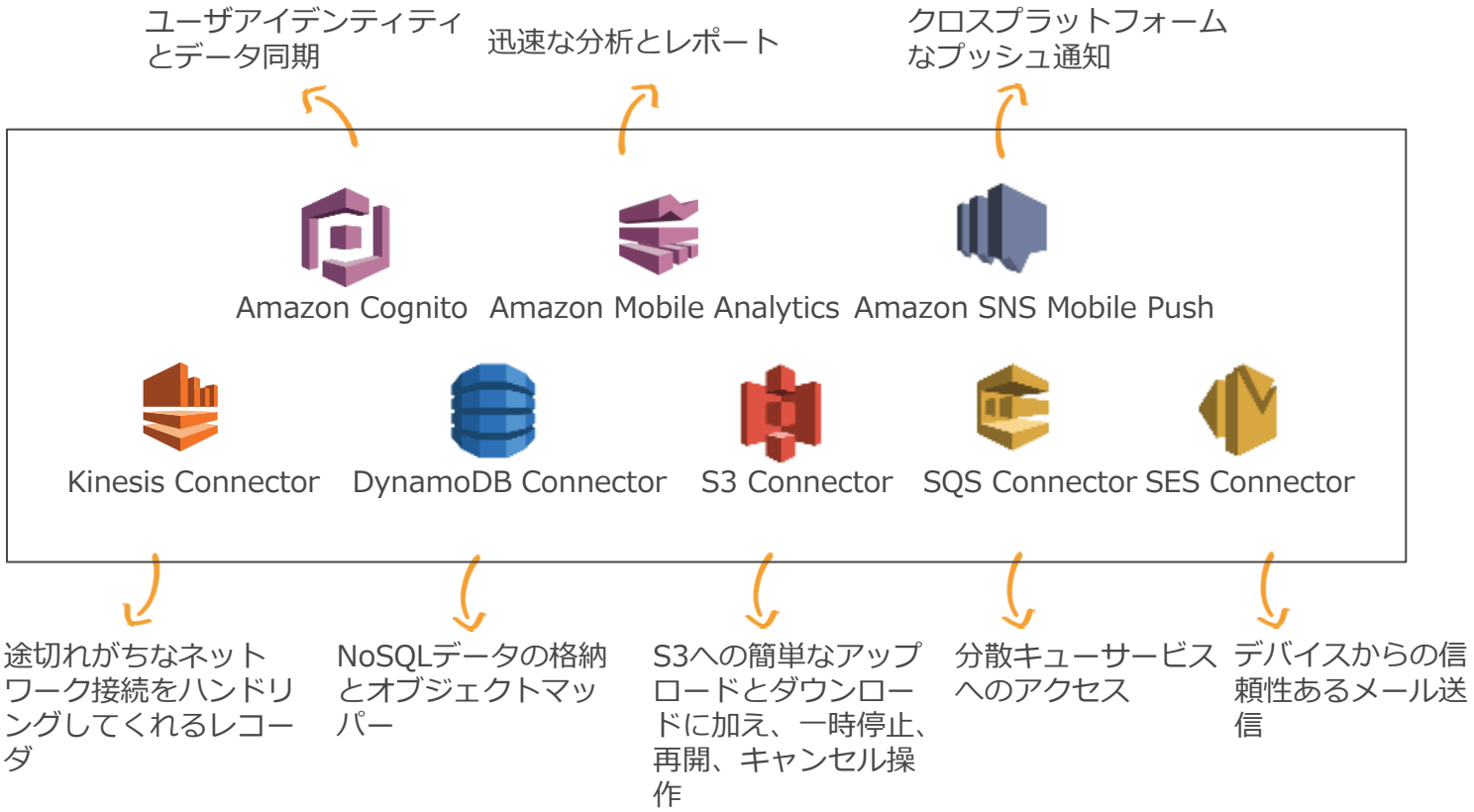
データのリアルタイム解析
リアルタイムなクリックストリームやログ等の
収集とそれに対する素早いアクションの実施

amazon web services

AWSモバイルサービスのご紹介

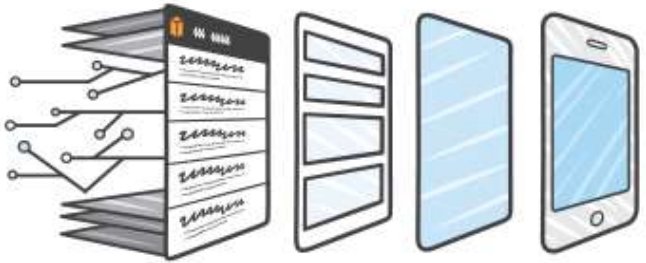


モバイルへの最適化、クロスプラットフォーム

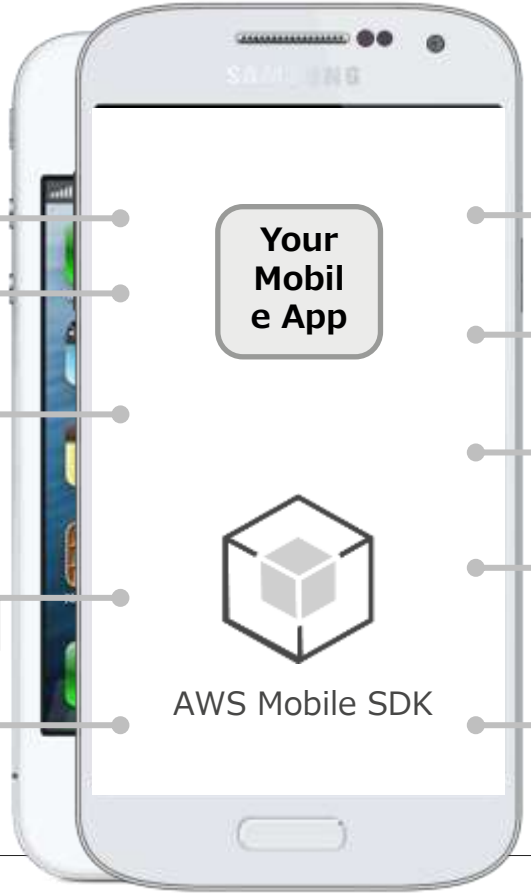


統合されたAWS Mobile SDK

- 全てのサービスに共通の認証機構
- オンライン・オフラインを自動でハンドリング
- クロスプラットフォームのサポート：Android, iOS, Fire OS (今後の予定： Unity, PhoneGap, Cordova)
- Mobile OSへの最適化
  - 例： ローカルオフラインキャシュを利用するアーキテクチャ
- メモリフットプリントの削減
  - 同梱するパッケージの選択も可



モバイルアプリ開発の課題



ユーザ認証
Amazon Cognito
(Identity Broker)

アクセスの認可
AWS Identity and
Access Management

データの同期
Amazon Cognito
(Sync)

ユーザの行動分析
Amazon Mobile
Analytics

保持率の追跡
Amazon Mobile
Analytics

メディアの管理
Amazon S3
Transfer Manager

メディアの配信
Amazon CloudFront
(Device Detection)

プッシュ通知の送信
Amazon SNS
Mobile Push

共有データの保存
Amazon DynamoDB
(Object Mapper)

データのリアルタイム解析
Amazon Kinesis
(Recorder)

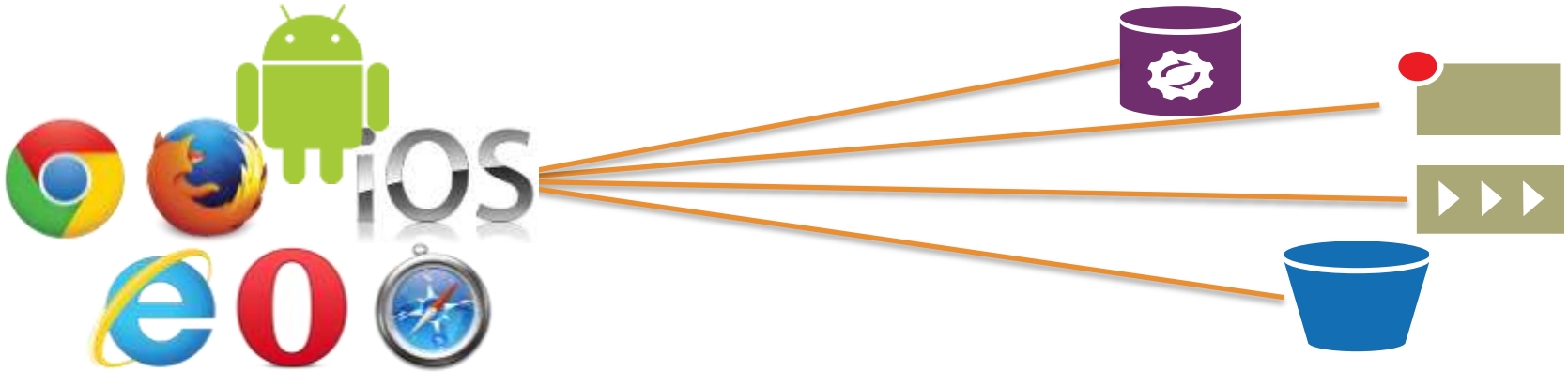
Agenda

Agenda

- AWSモバイルサービスのご紹介
- クライアントSDKのおさらい
- Amazon Cognito
  - Amazon Cognitoとは？
  - Amazon Cognito Identity Broker
  - Amazon Cognito Sync
  - Webアプリケーションからの利用
  - 料金
- Amazon Mobile Analytics
  - Amazon Mobile Analyticsとは？
  - レポート
  - カスタムイベント
  - 料金
- まとめ

AWSのマネージドサービスを活用した 2-tierのアーキテクチャが組める

- モバイル端末やブラウザから直接AWSの各種サービスを呼ぶ
  - AWSのマネージドサービスを組み合わせてバックエンドに
  - しかもプラットフォーム横断で連携！



クライアントSDK活用のメリット

- アプリの開発に多くのメリット：
  - バックエンド側の開発コストを最小化
  - バックエンド側の運用コストを最小化
  - スケーラビリティの心配なし
  - バックエンドのEC2を減らせるため金額面でもローコスト

- 必要に応じてEC2も導入できる安心感
  - 後からバックエンド側にロジックを入れてシステムの最適化

クライアントSDKを使う際の認証情報の扱い

- AWSの各種サービスはあくまでバックエンド
- エンドユーザは必ずしもAWSユーザではない
  - アプリは開発者のアカウントで認証・認可を受ける必要
- アプリに開発者アカウントのアクセスキー等を埋め込んだら
  - アクセスキーが広範にばら撒かれることに
  - アクセスキーの不正利用を止めるためにキーを無効化したら
    → 全ユーザへのサービスが停止！！
- アクセスキーの定期的な更新で対処するにしても
  - 更新のたびにバージョンアップは非現実的
  - 更新前のアプリからはサービス利用不可に

セキュアなAWSアクセスを提供するには

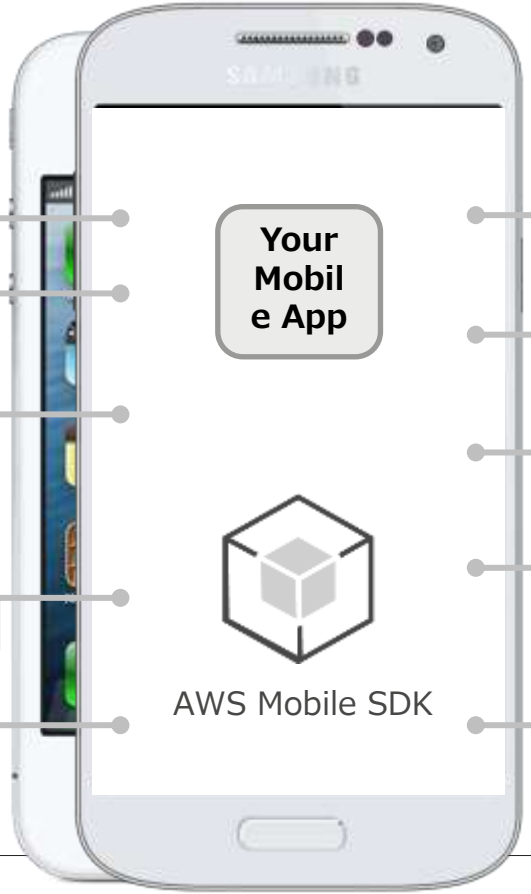
- アプリに認証情報を埋め込むべきではない
  - アクセスキーが広範囲に配布されてしまう
  - アクセスキーの更新はアプリのアップデートを伴うため非現実的
- エンドユーザ/端末ごとに異なる認証情報を提供すべき
  - ユーザごとに必要最小限の権限を与えるのは重要
  - 不正利用発覚時に不正ユーザのみ権限を停止
- 認証情報は期限が来たら無効化されるべき
  - 不正ユーザの影響も期限付きに

Agenda

Agenda

- AWSモバイルサービスのご紹介
- クライアントSDKのおさらい
- Amazon Cognito
  - Amazon Cognitoとは？
  - Amazon Cognito Identity Broker
  - Amazon Cognito Sync
  - Webアプリケーションからの利用
  - 料金
- Amazon Mobile Analytics
  - Amazon Mobile Analyticsとは？
  - レポート
  - カスタムイベント
  - 料金
- まとめ

モバイルアプリ開発の課題



ユーザ認証
Amazon Cognito (Identity Broker)

アクセスの認可
AWS Identity and Access Management

データの同期
Amazon Cognito (Sync)

ユーザの行動分析
Amazon Mobile Analytics

保持率の追跡
Amazon Mobile Analytics

メディアの管理
Amazon S3 Transfer Manager

メディアの配信
Amazon CloudFront (Device Detection)

プッシュ通知の送信
Amazon SNS Mobile Push

共有データの保存
Amazon DynamoDB (Object Mapper)

データのリアルタイム解析
Amazon Kinesis (Recorder)

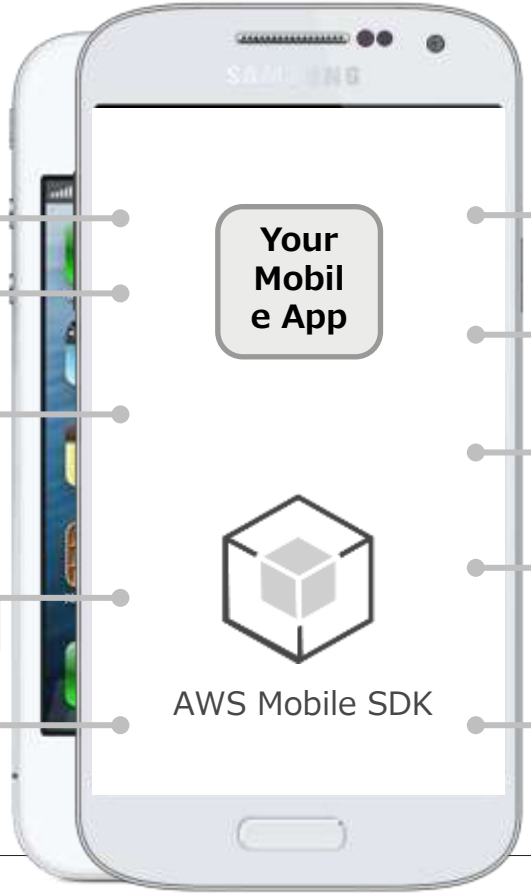
Amazon Cognitoとは？

- アプリのデータをセキュアに、オフラインでも参照可能な形で保存し、かつ、デバイス間でSync可能にするサービス

- Identity Broker
  - IDとアクセスの管理
    - 複数のIDプロバイダとの連携
    - パブリックなIDプロバイダを利用することで認証基盤の実装が不要
      当然、パスワードの保管も気にしなくていい
  - ユニークIDの作成と管理、識別

- Sync
  - デバイスをまたいだデータ同期
  - アプリケーションはオフラインでも機能
  - あらゆるデータをKey/Value形式で保存可能
    - アプリケーションの設定、ゲームにおける状態など
  - 手間のかかるサーバサイドの実装と運用が不要

モバイルアプリ開発の課題



ユーザ認証
Amazon Cognito (Identity Broker)

アクセスの認可
AWS Identity and Access Management

データの同期
Amazon Cognito (Sync)

ユーザの行動分析
Amazon Mobile Analytics

保持率の追跡
Amazon Mobile Analytics

メディアの管理
Amazon S3 Transfer Manager

メディアの配信
Amazon CloudFront (Device Detection)

プッシュ通知の送信
Amazon SNS Mobile Push

共有データの保存
Amazon DynamoDB (Object Mapper)

データのリアルタイム解析
Amazon Kinesis (Recorder)

Amazon Cognito Identity Broker

- 複数のIDプロバイダとの連携
- 未認証ユーザのゲストアクセス
- AWS認証情報の保護

特徴1: 複数のIDプロバイダをサポート

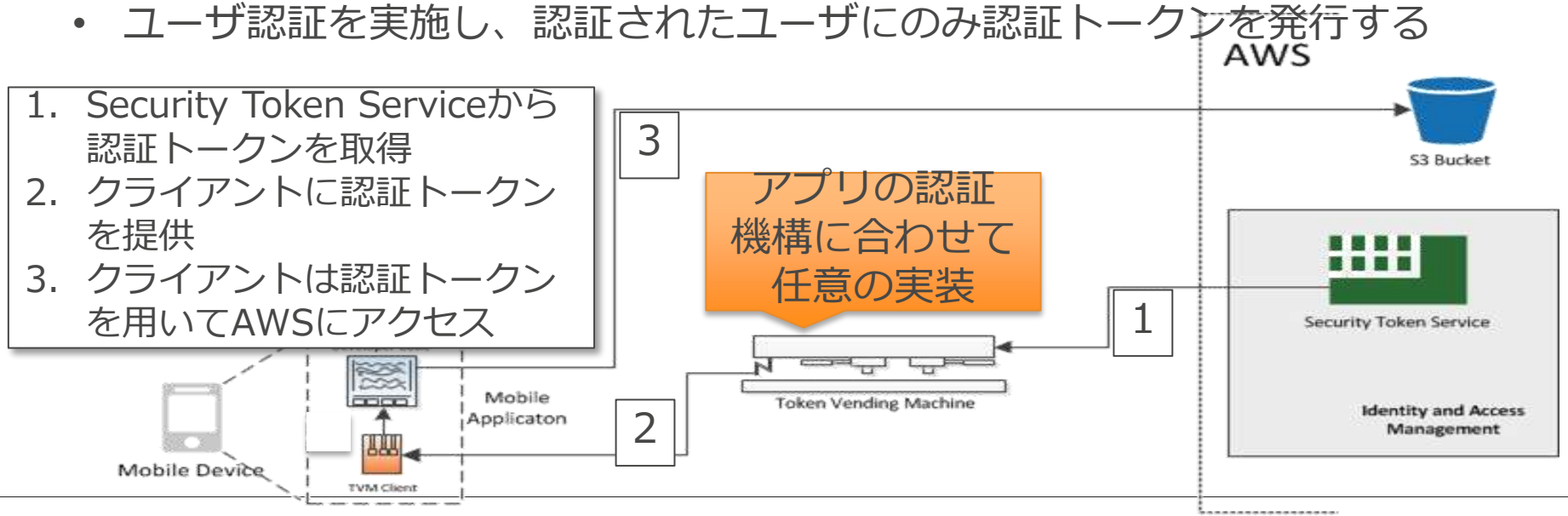
- 主要なIDプロバイダを利用した認証とのインテグレーションが可能
  - 2014年8月27日時点ではFacebook, Google, Amazonをサポート
  - IDプロバイダによる認証済ユーザに対してユニークIDを割り当て
  - アクセス権として事前に定義したIAM Roleが割り当てられる
  - ユーザの認証情報は保存されず、IDプロバイダから受信したトークンのみが保存される
- デバイスやプラットフォームをまたがったユニークユーザの認識と管理
- 従来のSTSを利用したWeb Identity Federationと同様の機能
  - ただし、独自の認証基盤を用いたい場合は従来通りSTS＋TVMの実装が必要

（参考）独自の認証機構の実装

Token Vending Machine (TVM)を導入

- ユーザ/端末の認証とトークンの発行を実施
- アプリケーションごとの認証とSTSを結びつけるための仕組み
- ユーザ認証を実施し、認証されたユーザにのみ認証トークンを発行する

1. Security Token Serviceから認証トークンを取得
2. クライアントに認証トークンを提供
3. クライアントは認証トークンを用いてAWSにアクセス

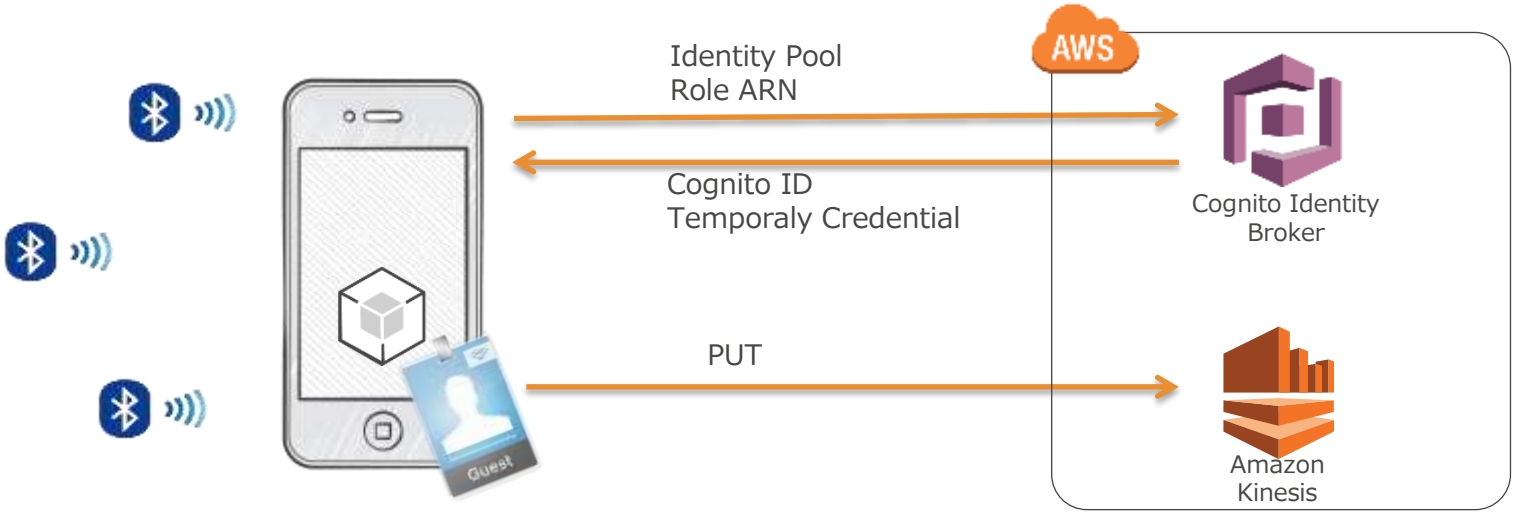


特徴2: 未認証ユーザのゲストアクセス

- IDプロバイダで認証をしていないユーザをゲストユーザとしてユニークIDの付与と管理
  - 未認証の場合、IDはデバイスと紐づくので同一デバイスからのアクセスの場合に同一ユーザとして認識される
  - 未認証によるアクセスを許可しない設定も可能
- アプリやAWSリソースへのアクセスにアカウントの作成や認証が不要
  - 別のログイン画面が表示されて抵抗感を与えるというようなことなくAWSリソースへアクセスさせることが可能
  - アクセス権限は未認証ユーザに対して割り当てたIAM Roleのポリシーに基づく
  - 従来、Anonymous TVMとしてサーバサイドで実装が必要だったものと同等
  - センサーデバイス等のスクリーンや入力装置のないデバイスに対してもユニークIDの付与と管理が可能
- データはクラウド上に保存され後からログインした場合は自動でマージ

（例）ゲストアクセスのユースケース

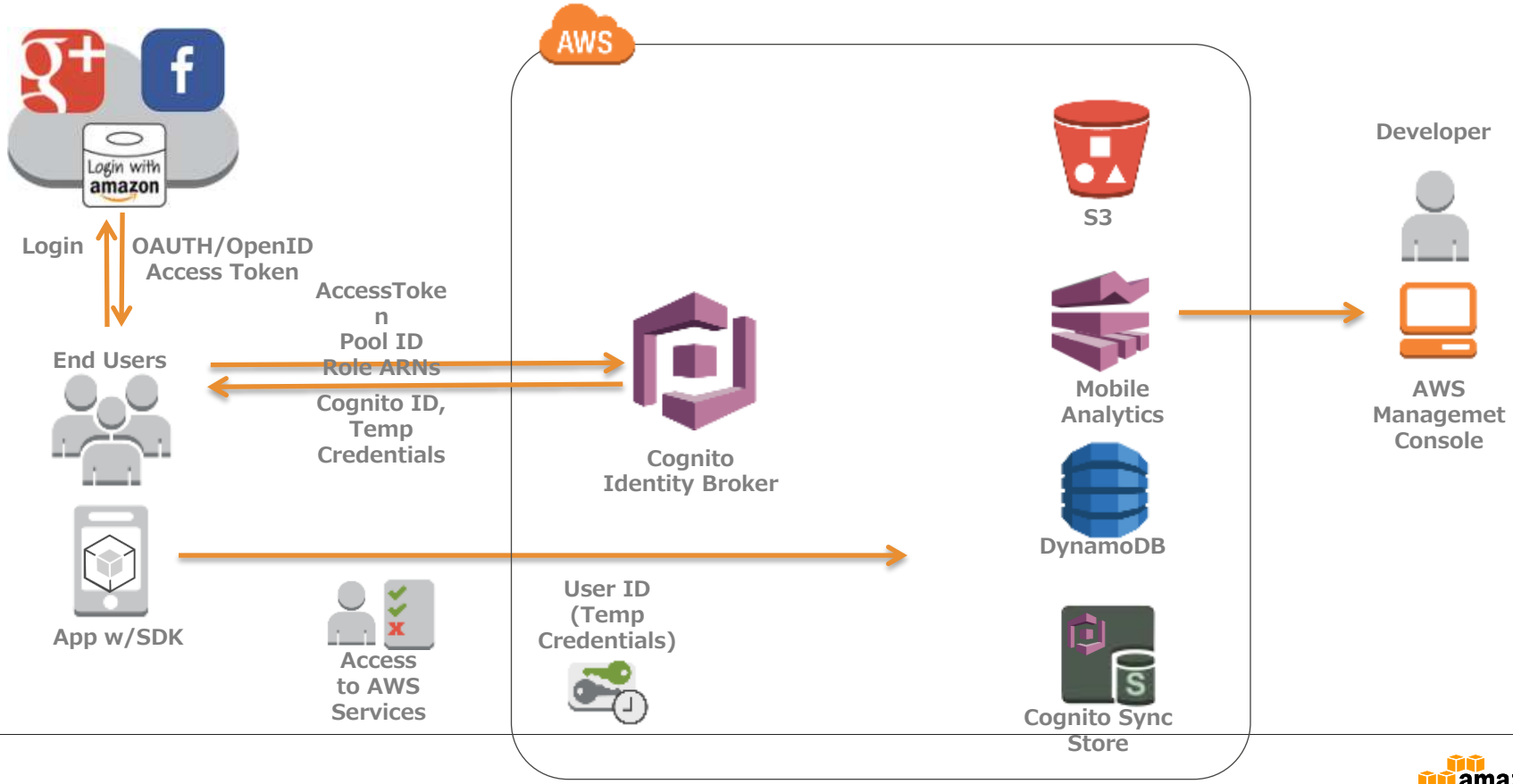
1. イベント会場でBLE（Bluetooth Low Energy）を利用したBeacon端末を配置
2. ユーザ登録不要なイベント公式アプリを配布し、ゲストユーザとして認証情報を取得
3. アプリはBeaconを拾って位置情報等を直接KinesisへPUT
   - ゲストユーザはKinesisへのPUTだけを許可
4. Kinesisは受け取ったデータを元にリアルタイムに処理を実施（ヒートマップ作成等）



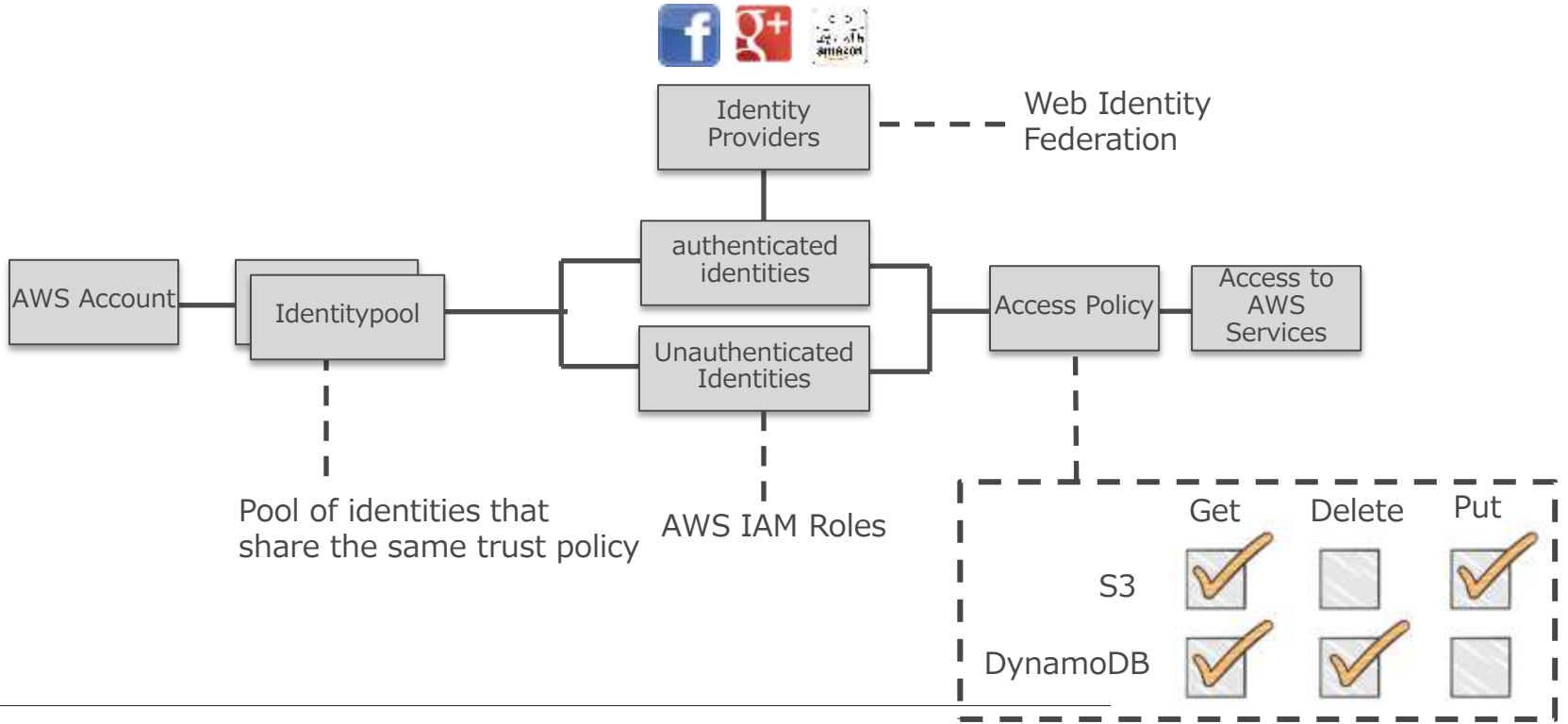
特徴3: AWS認証情報の保護

- 認証情報をアプリ内に埋め込む必要なし
  - 認証情報を埋め込んだアプリを配布することによるセキュリティリスク軽減
- AWSの各種リソースへのアクセスをきめ細やかに設定可能
  - 細かいアクセス権の設定はIAM Policyを用いて行う
- セキュリティのベストプラクティスの実装が容易
  - 従来、STSとTVMを用いて行っていた実装が簡単にサーバーレスで行えるように

Amazon Cognito セキュリティアーキテクチャ



Amazon Cognito (Identity Broker)



（参考）IAM Role用のアクセスポリシーの作成

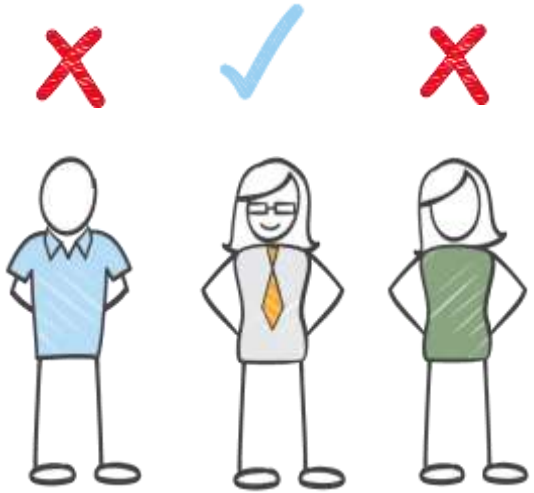
Allow
Actions:
All S3, Sync store Operations
Resource:
All resources within these services

Deny
Actions:
All DDB Operations
Resource:
All resources

```
{
  "Effect":"Allow",
  "Action":["s3:*"],
  "Resource":"*"
}

{
  "Effect": "Deny",
  "Action": ["dynamodb:*"],
  "Resource": "*"
}

{
  "Effect": "Allow",
  "Action": ["cognito-sync:*"],
  "Resource": "*"
}
```



（参考）アクセスポリシーによる制限

Allow
Actions:
Certain operations
Resource:
One bucket, table ..

```
{
  "Effect":"Allow",
  "Action":["s3:PutObject","s3:GetObject","s3:DeleteObject",
            "s3:ListMultipartUploadParts","s3:AbortMultipartUpload"],
  "Resource":"arn:aws:s3:::BUCKET_NAME/*"
}
{
  "Effect":"Allow",
  "Action":["s3:ListBucket","s3:ListBucketMultipartUploads"],
  "Resource":"arn:aws:s3:::BUCKET_NAME"
}
{
  "Effect": "Allow",
  "Action": ["dynamodb:GetItem", "dynamodb:Query", "dynamodb:PutItem"],
  "Resource" : [ "arn:aws:dynamodb:REGION:123456789:table/TABLE_NAME",
                 "arn:aws:dynamodb:REGION:123456789:table/TABLE_NAME/
              index/INDEX_NAME" ]
}
```

（参考）アクセスポリシーによる制限

Allow
Actions:
Certain operations
Resource:
Within a bucket with specific prefix (user)

```
{
  "Effect":"Allow",
"Action"
["s3:PutObject","s3:GetObject","s3:DeleteObject","s3:ListMultipartUploadParts","s3:AbortMultipartUpload"],
  "Resource":"arn:aws:s3:::BUCKET_NAME/Bob/*"
}
{
  "Effect":"Allow",
  "Action":"s3:ListBucket",
  "Resource":"arn:aws:s3:::BUCKET_NAME",
  "Condition":{"StringLike":{"s3:prefix":"Bob/"}}
}
{
  "Effect":"Allow",
  "Action":["s3:ListBucketMultipartUploads"],
  "Resource":"arn:aws:s3:::BUCKET_NAME"
}
```

アクセスポリシーによる制限 (Policy Variables)

Allow
Actions:
All sync operations
Resource:
Only to that identity

```
{
"Effect":"Allow",
"Action":"cognito-sync:*",
"Resource":["arn:aws:cognito-sync:us-east-1:
        123456789012:identitypool/
        ${cognito-identity.amazonaws.com:aud}/identity/
        ${cognito-identity.amazonaws.com:sub}/*"]
}
```

Allow
Actions:
S3 Get/Put operations
Resource:
Only to a specific part
of bucket to that identity

```
{
"Effect": "Allow",
"Action": ["s3:GetObject","s3:PutObject"],
"Resource": ["arn:aws:s3:::
        myBucket/amazon/snakegame/
        ${cognito-identity.amazonaws.com:sub}"]
}
```

Step 1: New Identity Pool

Identity pools are used to store end user identities. To declare a new identity pool, enter a unique name.

Identity Pool Name

Example: My App Name

Configure Identity Providers

Amazon Cognito recognizes tokens from these public identity providers. You need to specify the application provider identifier you plan to support in your application. You can always change your selection of providers after your identity pool is created. Learn more: Android, iOS.

Amazon App ID Optional

Example: amzn1.application.188a56d827a7d6555a8b67a5d

Facebook App ID Optional

Example: 7346241598935555

Google Client ID Optional

Example: 123456789012.apps.googleusercontent.com

事前に取得したIDプロバイダのApp ID等を指定

Unauthenticated Identities

Amazon Cognito can support unauthenticated identities by providing a unique identifier and AWS credentials for users who do not authenticate with an identity provider. If your application allows users who do not log in, you can enable access for unauthenticated identities. Learn more: Android, iOS.

Note: If you do not enable access for unauthenticated users, the Amazon Cognito API will not respond to any unauthenticated user requests.

Enable Access to Unauthenticated Identities

未認証ユーザによるアクセスを許可する場合はチェック

cel Create Pool

 Privacy Policy Terms of Use Feedback

Step 1. New Identity Pool
Step 2. Access Management
Step 3. Start Coding

Step 2: Identity and Access Management

Assigning a role to your application end users helps you restr... you select specific roles for both your authenticated and una... about IAM.

By default, Amazon Cognito creates a new role with limited p... access to other AWS resources, such as S3 or DynamoDB, y...

- 認証ユーザ、未認証ユーザそれぞれに割り当てるIAM Roleを指定
- 事前に用意したものを割り当てることも新たに作成することも可能

Assign Role to Authenticated Identities

This role is used for end users who have logged in through a public identity provider.

IAM Role Create a new IAM role

Role Name Cognito_sampleAuth_DefaultRole (preview IAM role policy)

Assign Role to Unauthenticated Identities

Unauthenticated end users automatically assume this role.

IAM Role Create a new IAM role

Role Name Cognito_sampleUnauth_DefaultRole (preview IAM role policy)

Cancel Skip Update Roles

 Privacy Policy Terms of Use

Feedback

Step 1. New Identity Pool
Step 2. Access Management
Step 3. Start Coding

Step 3: Start Using Amazon Cognito

Congratulations, your identity pool has been created! Here is some code to help you get started with the AWS Mobile SDK. We have pre-populated the identity pool and role variables for you, all you need to do is copy and paste. Please refer to the Getting Started Guides to access your identity pool ID and role ARNs in the future.

Note: For security reasons, the pre-populated code below will not be saved. We recommend you download the code by clicking the button below and keep it as reference.

Download Starter Code (Android and iOS)

Android Starter Code | iOS Starter Code

If you haven't already, download and install the AWS Mobile SDK.

Step 1: Import the Cognito libraries into your class:

```
import com.amazonaws.android.auth.CognitoCredentialsProvider;
```

Step 2: Initialize the Cognito client:

```
CognitoCredentialsProvider cognitoProvider = new CognitoCredentialsProvider(
    myActivity.getContext(), // get the context for the current activity
    [redacted],
    "us-east-1:[redacted]",
    "arn:aws:iam::[redacted]:role/Cognito_sampleUnauth_DefaultRole",
    "arn:aws:iam::[redacted]:role/Cognito_sampleAuth_DefaultRole"
);
```

Step 3: Retrieve your identity ID:

```
cognitoProvider.getIdentityId();
```

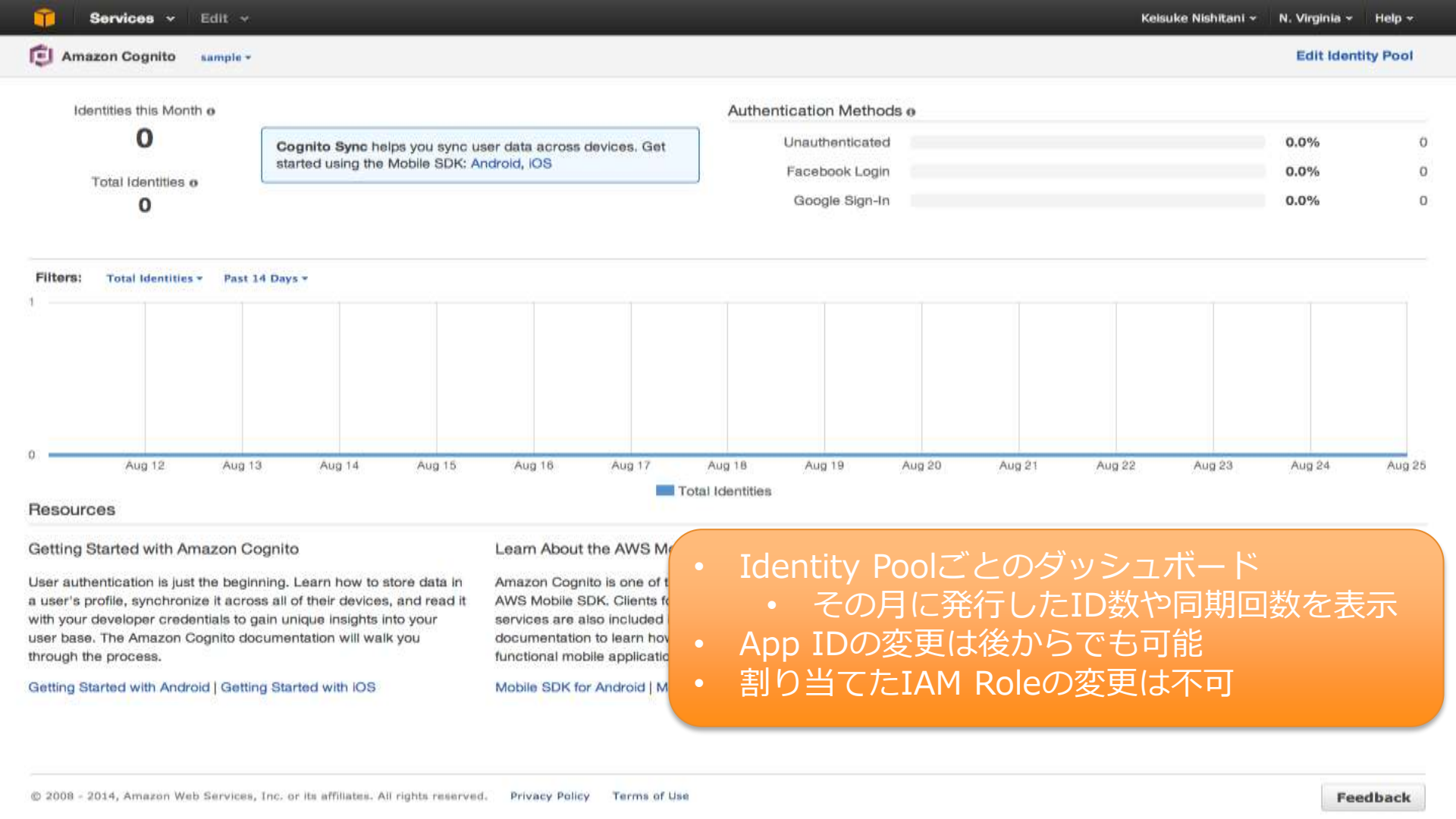
Learn more about Amazon Cognito: Getting Started Guide for Android.

サンプルはそのままコピー&ペーストして利用可能

Done

 Privacy Policy Terms of Use

Feedback


Services
Edit
Keisuke Nishitani
N. Virginia
Help
Amazon Cognito
sample
Edit Identity Pool
Identities this Month
0
Total Identities
0
Cognito Sync helps you sync user data across devices. Get started using the Mobile SDK: Android, iOS
Authentication Methods
Unauthenticated 0.0% 0
Facebook Login 0.0% 0
Google Sign-In 0.0% 0
Filters:
Total Identities
Past 14 Days
1
0
Aug 12
Aug 13
Aug 14
Aug 15
Aug 16
Aug 17
Aug 18
Aug 19
Aug 20
Aug 21
Aug 22
Aug 23
Aug 24
Aug 25
Total Identities
Resources
Getting Started with Amazon Cognito
User authentication is just the beginning. Learn how to store data in a user's profile, synchronize it across all of their devices, and read it with your developer credentials to gain unique insights into your user base. The Amazon Cognito documentation will walk you through the process.
Getting Started with Android | Getting Started with iOS
Learn About the AWS M
Amazon Cognito is one of t
AWS Mobile SDK. Clients f
services are also included
documentation to learn hov
functional mobile applicatio
Mobile SDK for Android | M
• Identity Poolごとのダッシュボード
• その月に発行したID数や同期回数を表示
• App IDの変更は後からでも可能
• 割り当てたIAM Roleの変更は不可
© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Feedback

IAMポリシー

- 最低限以下のポリシーが必要

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action":"cognito-sync:*",
    "Resource":["arn:aws:cognito-sync:us-east-
1:123456789012:identitypool/${cognito-
identity.amazonaws.com:aud}/identity/${cognito-identity.amazonaws.com:sub}/*"]
  }]
}
```

IAMポリシー

- Identity Pool全体へのアクセスを許可する場合は以下のポリシーを設定

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action":"cognito-sync:*",
    "Resource":["arn:aws:cognito:us-east-1:123456789012:identitypool/*"]
  }]
}
```

基本的な流れ（Android/iOS共通）

- マネージメントコンソール上でIdentity Poolを作成する
- AWS SDK for Android/iOSをプロジェクトに追加する
- Amazon Cognito credentials providerを初期化する
- IDプロバイダの認証情報を渡す
  ※未認証ユーザの場合は不要
- Cognitoの認証情報を利用して、その他のAWSサービスのクライアントを初期化する

コード例（Android）

```
com.amazonaws.android.auth.CognitoCredentialsProvider;

// Credential Providerの初期化
CognitoCredentialsProvider credentialsProvider = new CognitoCredentialsProvider(
    getContext(), // 処理実行時にコンテキストを指定
    “1234567890”, // AWSアカウントID
    “us-east-1:XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXX”, // 作成したIdentity PoolのID
    “arn:aws:iam::XXXXXXXXXX:role/YourRoleName”,       // 未認証ユーザに割り当てるIAM RoleのARN
    “arn:aws:iam::XXXXXXXXXX:role/YourRoleName”        // 認証済ユーザに割り当てるIAM RoleのARN
);

// Facebookユーザで認証する場合
Map logins = new HashMap();
logins.put("graph.facebook.com", Session.getActiveSession().getAccessToken());
credentialsProvider.withLogins(logins);
```

※上記例では省略していますが、事前にIDプロバイダを利用した認証処理の実装が必要です
※実際にはLoaderやAsyncTask等を用いた非同期タスクとして実装すること

コード例（iOS）

iOS

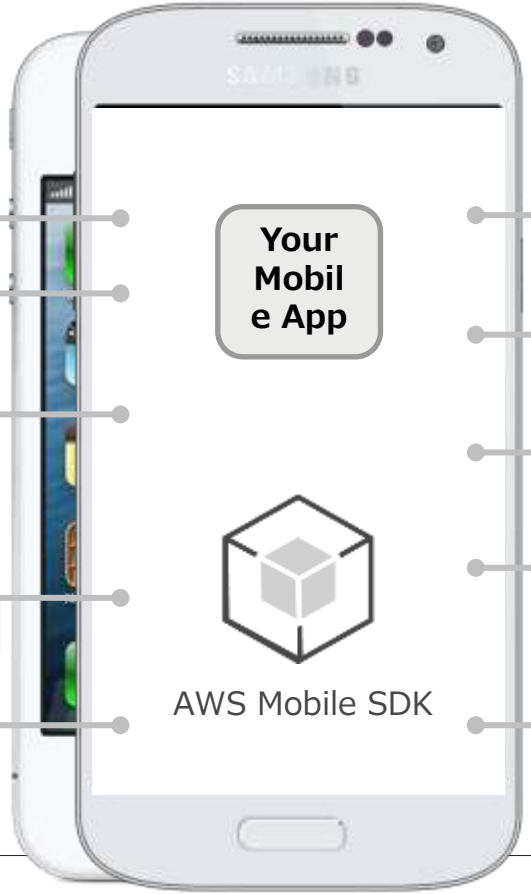
```
#import <AWSiOSSDK/AWSCore.h>

// Credential Providerの初期化
AWSCognitoCredentialsProvider *credentialsProvider = [AWSCognitoCredentialsProvider
    credentialsWithRegionType:AWSRegionUSEast1
    accountId:@"1234567890", // AWSアカウントID
    identityPoolId:@"us-east-1:XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXX", // 作成したIdentity PoolのID
    unauthRoleArn:@“arn:aws:iam::XXXXXXXXXX:role/YourRoleName”,  //未認証ユーザのIAM Role ARN
    authRoleArn:@“arn:aws:iam::XXXXXXXXXX:role/YourRoleName” //認証済ユーザのIAM Role ARN
];

// Facebookユーザで認証する場合
NSString *token = FBSession.activeSession.accessTokenData.accessToken;
credentialsProvider.logins = @{ AWSCognitoLoginProviderKeyFacebook: token };
```

※上記例では省略していますが、事前にIDプロバイダを利用した認証処理の実装が必要です

モバイルアプリ開発の課題



ユーザ認証
Amazon Cognito
(Identity Broker)

アクセスの認可
AWS Identity and
Access Management

データの同期
Amazon Cognito
(Sync)

ユーザの行動分析
Amazon Mobile
Analytics

保持率の追跡
Amazon Mobile
Analytics

メディアの管理
Amazon S3
Transfer Manager

メディアの配信
Amazon CloudFront
(Device Detection)

プッシュ通知の送信
Amazon SNS
Mobile Push

共有データの保存
Amazon DynamoDB
(Object Mapper)

データのリアルタイム解析
Amazon Kinesis
(Recorder)

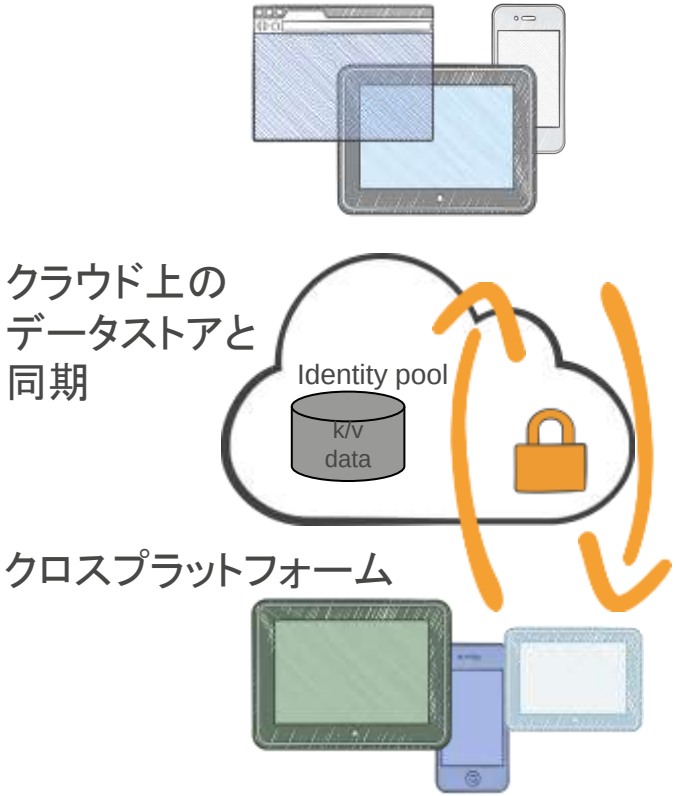
amazon web services

Amazon Cognito Syncの特徴

- 異なるデバイス、OS間でのデータ同期
- オフライン機能
- インテリジェントな同期と柔軟なコンフリクトの解消

特徴1:異なるデバイス、OS間でのデータ同期

- 一行のコードでユーザデータや設定をデバイス間で同期
  - サーバサイドの実装は不要
  - アプリの設定やゲームの状態などを簡単にクラウド上に保存可能
- データの読み書きはローカルのSQLiteに対してのみ行うため高速



特徴2: オフライン機能

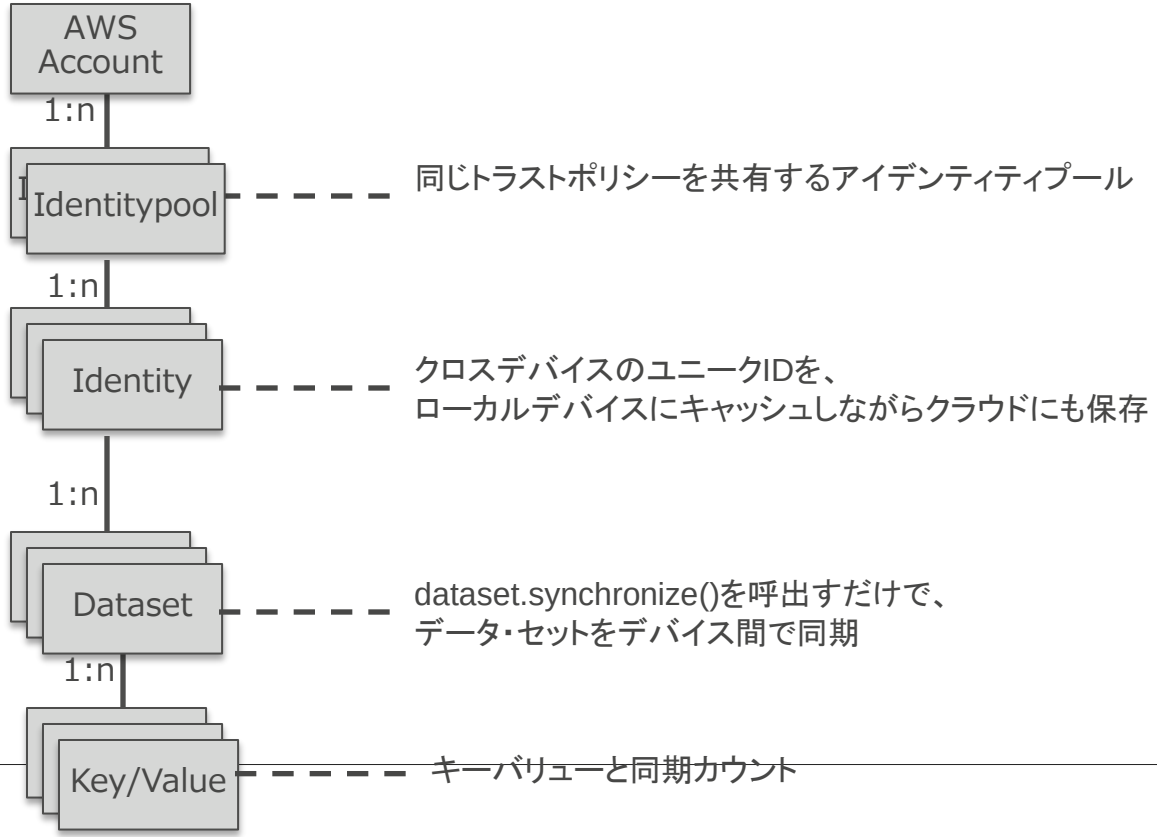
- SDKがローカルのSQLiteデータストアを管理
  - キャッシュとしての動作及び、全ての読み書きの受け口となる
- 書き込みはまずローカルのSQLiteに行われる
  - インターネット接続が不安定または切断状態でもアプリが機能する

Local SQLite Cache

特徴3: インテリジェントな同期と柔軟なコンフリクト解決

- 同期処理を行うメソッドはローカルのデータとクラウド上にあるデータのバージョンを比較
- 差分があった場合、クラウドにプッシュすると同時に新しい変更を取得
  - 最初にクラウド上の変更を読み取り、その後ローカルの変更をクラウド上へと書き込む
- 競合時は最後の書き込みを適用（デフォルト）
  - 開発者がコンフリクトの解決方法を独自に実装することも可能

Amazon Cognito Sync データモデル



Dataset

- 各Identityは最大20MBまでデータ保存可能
- 各Datasetは1MBまでKey/Value形式のデータを保存可能
  - Key/Valueともに英数字の文字列
  - 保存可能なキーの数は1024個まで
  - 容量内であれば文字数の制限はなし
  - バイナリデータはbase64でエンコードして保管する
- 保管されるデータはすべて暗号化され、通信はHTTPSで暗号化されている
  - ただし、ローカルキャッシュはユーザ側で暗号化を実装しなければ暗号化はされない

コード例（Android）

```
//CredentialsProviderとCognitoSyncClientの初期化
provider = new CognitoCredentialsProvider(context, AWS_ACCOUNT_ID,
                COGNITO_POOL_ID, COGNTIO_ROLE_UNAUTH,
                        COGNITO_ROLE_AUTH);

cognito = new DefaultCognitoSyncClient(context, COGNITO_POOL_ID, provider);

//Datasetを作成もしくはオープンし、Key/Value形式でデータを追加
cognito.openOrCreateDataset(datasetName);
dataset.put(key, value);

//同期処理の実行
dataset.synchronize(new SyncCallback(){..});
```

コード例（iOS）

iOS

```
// Credential Providerの初期化
AWSCognitoCredentialsProvider *credentialsProvider = [AWSCognitoCredentialsProvider
    credentialsWithRegionType:AWSRegionUSEast1
    accountId:@"1234567890", // AWSアカウントID
    identityPoolId:@"us-east-1:XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXX", // 作成したIdentity PoolのID
    unauthRoleArn:@“arn:aws:iam::XXXXXXXXXX:role/YourRoleName”,  //未認証ユーザのIAM Role ARN
    authRoleArn:@“arn:aws:iam::XXXXXXXXXX:role/YourRoleName” //認証済ユーザのIAM Role ARN
];

AWSCognitoSyncClient *syncClient = [[AWSCognitoSyncClient alloc]       initWithConfiguration:
configuration];


//Datasetを作成もしくはオープンし、Key/Value形式でデータを追加
DataSet *dataset = [syncClient openOrCreateDataSet:@"myDataSet"];
NSString *value = [dataset readStringForKey:@"myKey"];
[dataset putString:@"my value" forKey:@"myKey"];

//同期処理の実行
[dataset synchronize];
```

2種類のSyncronize

- synchronize
  - 接続が不安定な場合などエラー時の処理は自分で実装する必要がある
  - コールされるとクラウド上の変更がpullされ、ローカルの変更はpushされる
- synchronizeOnConnectivity
  - 実行時に接続可能であれば通常のsynchronizeメソッドと同様の振る舞いをする
  - 接続できなかったときは接続状態を監視し可能になったら同期される
  - 複数回呼び出した場合は最後のオペレーションがキープされる

Webアプリケーションでの利用

- モバイルアプリケーションと同様にWebアプリケーションからの利用も可能
  - AWS SDKから利用
  - 現時点では以下のSDKで利用可能
    - Java、.NET、PHP、NodeJS、JavaScript（Client Side）
- Mobile SDKと同様にIDプロバイダと連携した認証やデータ同期が可能
  - モバイルアプリでSyncしたデータにもアクセス可能
  - 認証済ユーザはプラットフォームを超えてユニークに認識される

JavaScriptサンプル

```
// AWSアカウントID、Identity Pool ID、認証/未認証時に割り当てるIAM Role(ARN)を指定してパラメータ作成
// 以下ではFacebookユーザを認証済ユーザとして使用
AWS.config.region = ‘us-east-1’;  // リージョンの指定（必須）
AWS.config.credentials = new AWS.CognitoIdentityCredentials(
    AccountId: "YOUR_AWS_ACCOUNT_ID",
    RoleArn: "arn:aws:iam::6157xxxxxxxx:role/a_valid_aws_role_arn",
    IdentityPoolId: "YOUR_COGNITO_IDENTITY_POOL_ID”,
    Logins: {
        graph.facebook.com : facebookResponse.authResponse.accessToken
    }
);

// Cognitoから付与されたIDの取得
AWS.config.credentials.get(function(err) {
    if (!err) {
        console.log("Cognito Identity Id: " + AWS.config.credentials.identityId);
    }
});
```

JavaScriptサンプル（Sync）

```
// Sync
var cognitoSyncClient = new AWS.CognitoSync();
cognitoSyncClient.listDatasets({
    IdentityId: AWS.config.credentials.identityId,
    IdentityPoolId: "YOUR_COGNITO_IDENTITY_POOL_ID"
}, function(err, data) {
    if ( !err ) {
        console.log(JSON.stringify(data));
    }
});
```

料金

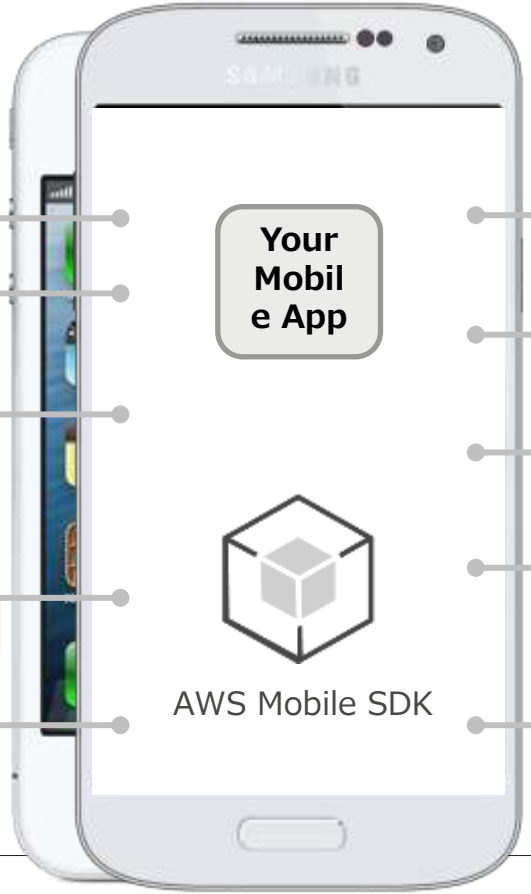
- 無料利用枠
  - 月あたり100万回の同期オペレーション
  - 月あたり10GBのデータストア
  - 最初の12ヶ月のみ
- それ以降
  - 同期オペレーション
    10000回の同期オペレーションあたり$0.15
  - 同期用データストアの容量
    $0.15/GB

Agenda

Agenda

- AWSモバイルサービスのご紹介
- クライアントSDKのおさらい
- Amazon Cognito
  - Amazon Cognitoとは？
  - Amazon Cognito Identity Broker
  - Amazon Cognito Sync
  - Webアプリケーションからの利用
  - 料金
- Amazon Mobile Analytics
  - Amazon Mobile Analyticsとは？
  - レポート
  - カスタムイベント
  - 料金
- まとめ

モバイルアプリ開発の課題



ユーザ認証
Amazon Cognito
(Identity Broker)

アクセスの認可
AWS Identity and
Access Management

データの同期
Amazon Cognito
(Sync)

ユーザの行動分析
Amazon Mobile
Analytics

保持率の追跡
Amazon Mobile
Analytics

メディアの管理
Amazon S3
Transfer Manager

メディアの配信
Amazon CloudFront
(Device Detection)

プッシュ通知の送信
Amazon SNS
Mobile Push

共有データの保存
Amazon DynamoDB
(Object Mapper)

データのリアルタイム解析
Amazon Kinesis
(Recorder)

Amazon Mobile Analyticsとは？

- アプリの利用状況に関するデータを収集、可視化
- グラフィカルなレポートとCSV形式のデータダウンロードを提供
  - アプリケーション側はAmazon Mobile SDKを組み込むだけ
  - CognitoのIdentity Poolもしくは IAMユーザを用意するだけで利用可能
- 高速かつスケーラブル
  - データを受け取ったら60分以内にレポート反映
  - 数百万のユーザーからの数十億イベント/日を収集・処理可能
- データの所有
  - 収集したデータは共有、集計や再利用はされない
- クロスプラットフォーム
  - 各種デバイス、OS（Android, iOS, FireOS）からデータを送信可能

レポート

- 一般的に計測することの多いメトリクスはイベントデータ受信時、自動的に計算・更新
  - Daily Active Users (DAU), Monthly Active Users (MAU), 新規ユーザ
  - Sticky Factor (DAU ÷ MAU)
  - Session数と DAU当たりの平均セッション数
  - Average Revenue per Daily Active User (ARPDAU)
  - Average Revenue per Paying Daily Active User (ARPPDAU)
  - 1, 3, 7日のRetention
  - 1, 2, 3週の Retention

Amazon Mobile Analytics | Demo App | Filters: 06/11/14 - 06/17/14 | All Platforms

Overview | Active Users | Sessions | Revenue | Retention | Custom Events

Lifetime Users

505,616 iOS: 84,547 Android: 382,456 Fire OS: 38,613

Lifetime Value Per User

$0.65 USD iOS: $2.15 USD Android: $0.38 USD Fire OS: $0.05 USD

Daily Active Users (DAU)

Avg. **2,914** | Change ▲4.7%

Monthly Active Users (MAU)

Avg. **5,577** | Change ▲0.6%

New Users

Avg. **1,108** | Change ▲28.1%

Sticky Factor

Wt. Avg. **0.52** | Change ▲4.1%

Total Sessions

Avg. **4,112** | Change ▲5.7%

Day 1 Retention

Avg. **140** | Change ▲2.9%

カスタムイベント

- 開発者が定義するアプリケーション固有のイベント
  - アプリケーション固有のユーザ行動を把握可能
  - 属性とメトリクスを指定してコンテキストを追加可能
- カスタムイベントの例
  - ニュースアプリなどにおける記事ごとのLike/Share数
  - インアプリ商品の購入
  - 音楽アプリにおける曲の再生数

IAMポリシー

- Analyticsを利用するには以下のポリシーが必要

```
{
  "Statement": [{
      "Effect": "Allow",
      "Action": "mobileanalytics:PutEvents",
      "Resource": "*"
  }]
}
```

コード例（Android）

- AndroidManifest.xmlに以下のパーミッションを追加

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

- 以下のクラスをimport
  サンプルではLogクラスも利用するのでこれもimport

```
import com.amazonaws.android.mobileanalytics.*;
import com.amazonaws.android.auth.CognitoCredentialsProvider;
import android.util.Log;
```

コード例（Android）

- Mobile Analytics Clientへの参照としてStatic変数を追加
  あわせて、後ほどカスタムイベントとして使用する定数も追加しておく

```
private static AmazonMobileAnalytics analytics;
private static final int STATE_LOSE = 0;
private static final int STATE_WIN = 1;
```

コード例（Android）

```
// アクティビティのonCreate()でCognitoクライアントを初期化し、その後AmazonMobileAnalyticsインスタンスを作成。
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    CognitoCredentialsProvider cognitoProvider = new CognitoCredentialsProvider(
        getApplicationContext(),
        AWS_ACCOUNT_ID,
        COGNITO_IDENTITY_POOL,
        "arn:aws:iam::AWS_ACCOUNT_ID:role/UNAUTHENTICATED_ROLE",
        "arn:aws:iam::AWS_ACCOUNT_ID:role/AUTHENTICATED_ROLE"
    );

    try {
        AnalyticsOptions options = new AnalyticsOptions();
        options.withAllowsWANDelivery(true);
        analytics = new AmazonMobileAnalytics(
                cognitoProvider,
                getApplicationContext(),
                "yourCompany.yourAppName",
                options
        );
    } catch(InitializationException ex) {
            Log.e(this.getClass().getName(), "Failed to initialize Amazon Mobile Analytics", ex);
    }
}
```

コード例（Android）

```
// セッションイベントを記録するためにonPause()とonResume()をoverrideし、Analyticsのコードを記述する
// アプリケーション内の各アクティビティのonPause()とonResume()で同様のことをする必要がある

@Override
protected void onPause() {

    super.onPause();
    if(analytics != null) {
        analytics.getSessionClient().pauseSession();
            //Attempt to send any events that have been recorded to the Mobile Analytics service.
        analytics.getEventClient().submitEvents();
    }
}

@Override
protected void onResume() {
    super.onPause();
    if(analytics != null)  {
        analytics.getSessionClient().resumeSession();
    }
}
```

カスタムイベントのコード例（Android）

- プレイヤーのレベルがあがった時にその情報をカスタムイベントとして記録

```
// プレイヤーのレベルがあがったら呼び出されるメソッド
public void onLevelComplete(String levelName, String difficulty, double timeToComplete, int playerState) {

    // Level Completeというカスタムイベントの作成
    // attributeとしてLevelNameとDifficultyの2つ、metricsとしてTimeToCompleteを定義
    AnalyticsEvent levelCompleteEvent = analytics.getEventClient().createEvent("LevelComplete")
            .withAttribute("LevelName", levelName)
            .withAttribute("Difficulty", difficulty)
            .withMetric("TimeToComplete", timeToComplete);

    //attributeとmetricsはaddステートメントを使用して追加することも可能
    if (playerState == STATE_LOSE)
        levelCompleteEvent.addAttribute("EndState", "Lose");
    else if (playerState == STATE_WIN)
        levelCompleteEvent.addAttribute("EndState", "Win");

    //Record the Level Complete event
    analytics.getEventClient().recordEvent(levelCompleteEvent);
}
```

カスタムイベントのコード例（Android）

```
// onCreate()の最後に先ほどのカスタムイベントを呼び出す処理を追加
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    CognitoCredentialsProvider cognitoProvider = new CognitoCredentialsProvider(
        getApplicationContext(),
        AWS_ACCOUNT_ID,
        COGNITO_IDENTITY_POOL,
        "arn:aws:iam::AWS_ACCOUNT_ID:role/UNAUTHENTICATED_ROLE",
        "arn:aws:iam::AWS_ACCOUNT_ID:role/AUTHENTICATED_ROLE"
    );

    try {
        AnalyticsOptions options = new AnalyticsOptions();
        options.withAllowsWANDelivery(true);
        analytics = new AmazonMobileAnalytics(
                cognitoProvider,
                getApplicationContext(),
                "yourCompany.yourAppName",
                options
        );
    } catch(InitializationException ex) {
            Log.e(this.getClass().getName(), "Failed to initialize Amazon Mobile Analytics", ex);
    }

    this.onLevelComplete("Lower Dungeon", "Very Difficult", 2734, STATE_WIN);
}
```

料金

- 収集するイベント数による課金

- 無料枠
  - 月あたり1億イベントまで

- それ以降
  - 月あたり100万イベントにつき$1

Agenda

Agenda

- AWSモバイルサービスのご紹介
- クライアントSDKのおさらい
- Amazon Cognito
  - Amazon Cognitoとは？
  - Amazon Cognito Identity Broker
  - Amazon Cognito Sync
  - Webアプリケーションからの利用
  - 料金
- Amazon Mobile Analytics
  - Amazon Mobile Analyticsとは？
  - レポート
  - カスタムイベント
  - 料金
- まとめ

まとめ

Summary

- Amazon Cognito
  - ユーザの認証、セキュアなAWSサービスリソースへのアクセスを簡単に実現！
  - クロスプラットフォーム、クロスデバイスなデータ同期！
- Amazon Mobile Analytics
  - モバイルアプリの利用に関する情報の収集と可視化！
  - 標準のメトリクスに加え、カスタムイベントも収集可能！

参考資料

- Twitter: @awsformobile
- ブログ
  - http://mobile.awsblog.com/
- ドキュメント
  - Amazon Cognito: https://aws.amazon.com/documentation/cognito/
  - Amazon Mobile Analytics: https://aws.amazon.com/documentation/mobileanalytics/

Webinar資料の配置場所

- AWS クラウドサービス活用資料集
  - http://aws.amazon.com/jp/aws-jp-introduction/

プロダクト別：				
Amazon S3		AWSマイスターシリーズ Re:Generate Amazon Simple Storage Service (S3)	Slideshare	PDF
Amazon Glacier		AWSマイスターシリーズ Reloaded Amazon Glacier Amazon Glacierのご紹介 機能編	Slideshare（Reloaded） Slideshare（機能編）	PDF（Reloaded） PDF（機能編）
Amazon Route 53		AWSマイスターシリーズ Re:Generate	Slideshare	PDF

ご参加ありがとうございました。