

C - 5

# Slack も採用する Amazon Chime SDK とは？

Kazuki Matsuda  @mats16k

Startup Solutions Architect  
Amazon Web Services Japan K.K.

**Kazuki Matsuda**  @mats16k

Startup Solutions Architect  
Amazon Web Services Japan K.K.

---



インターネット広告代理店（のアドサーバ作っている部門）でインフラエンジニアに

➡ アドテク系スタートアップ2人目のエンジニア（SRE、アプリ開発、情シス、データエンジニア、採用）

➡ AWS でスタートアップ支援

 AWS Fargate /  AWS Lambda あたりが好きです

最近では  AWS Amplify と  Amazon Chime SDK がイチオシ



# はじめに

## 本セッションで想定する聴講者

- プレスリリースを見て Amazon Chime SDK に興味を持った方
- 自社サービスにリアルタイムコミュニケーション機能を組み込みたいと考えているフロントエンドエンジニア / ネイティブアプリエンジニア

## 本セッションのゴール

- Amazon Chime SDK の概要、仕組みを理解する
- Amazon Chime SDK を自社サービスに組み込むイメージをもつ



# アジェンダ

- リアルタイムコミュニケーションの機運と課題
- Amazon Chime SDK とは
- Amazon Chime SDK の主な API
- サンプルアプリケーションの紹介
- まとめ



# リアルタイムコミュニケーションの 機運と課題

# リアルタイムコミュニケーションへの期待

遠隔医療



銀行



保険



遠隔教育



小売



セールス・サポート



企業ミーティング




ゲーム



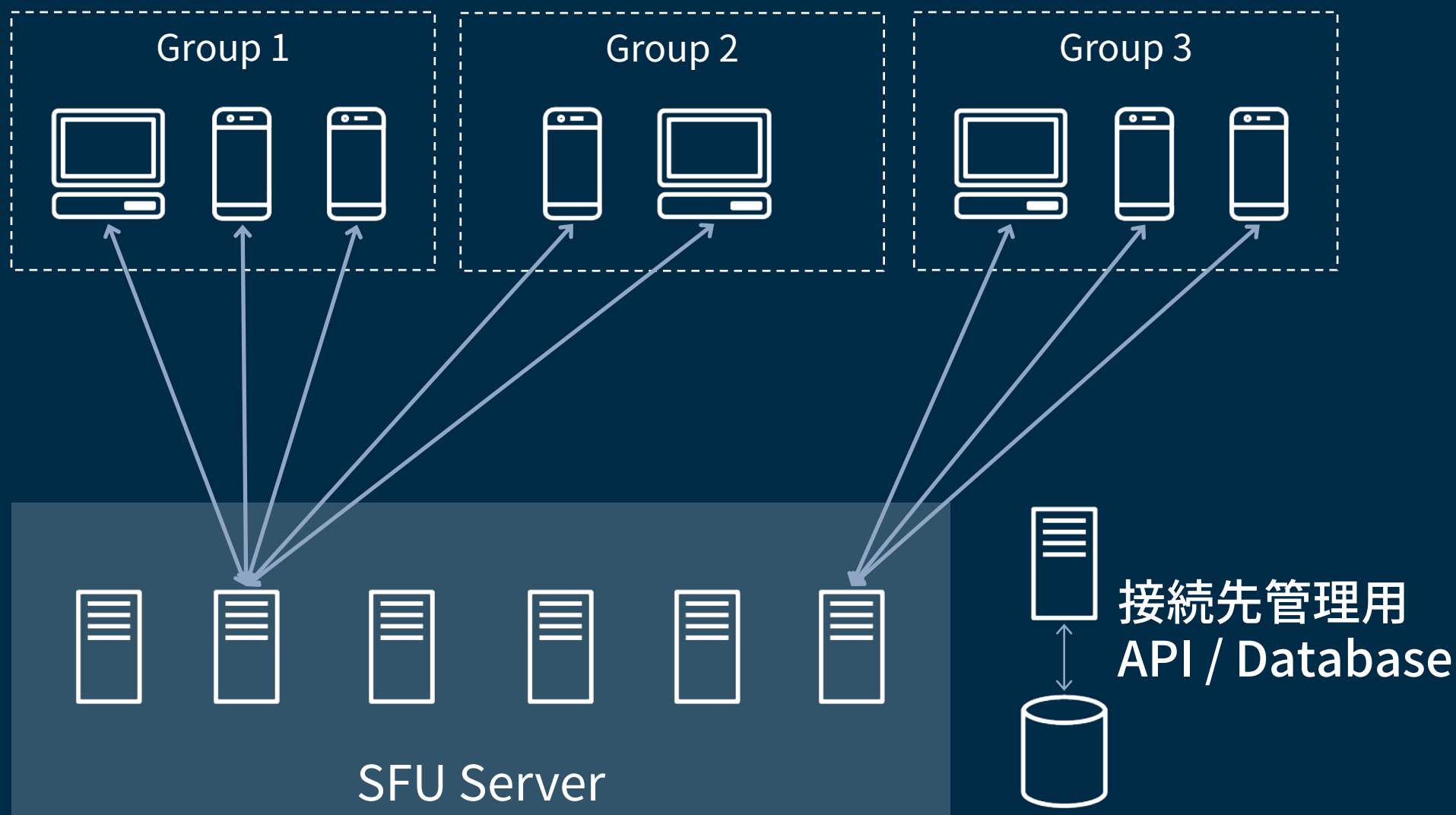
リアルタイムコミュニケーションにより、カスタマーエンゲージメントをさらに深める



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with  intel.

# WebRTC SFU (Selective Forwarding Unit)



## ■ 課題

- SFU サーバの運用に関する知識
- 構築、運用の工数
- 接続数などに応じたチューニング
- 接続先の管理

# リアルタイムコミュニケーション機能のチャレンジ



## 開発のオーバーヘッド

ゼロから構築する工数の増大  
長い納期  
コアコンピテンシーでないスキル



## 複雑なオペレーション

インフラレイヤーの設計と管理  
品質と信頼性の期待に応える  
コストの最適化



## 限られた拡張性

複雑な他のサービスとの統合  
AI/ML ソリューションに追随  
サービス間のセキュリティ

**市場投入までの時間、コスト、機能制限の増加**

# Amazon Chime SDK とは

# Amazon Chime SDK とは



## Amazon Chime SDK

### ■ Amazon Chime SDK ≠ Amazon Chime

- Amazon Chime を制御するための SDK ではない
- Chime SDK はビデオ通話機能を実装するのに必要な **フロントエンド向けの SDK** (JavaScript, iOS, Android) と、**バックエンドの Chime Media Services** で構成されたフルマネージドサービス
- 会議セッションや参加者の作成や管理は AWS SDK を利用し Chime API から行う

### ■ 音声、ビデオ、画面共有、電話網通話

- 主要なオペレーティングシステムとブラウザをサポート
- SDK は JavaScript / iOS / Android で利用可能

### ■ RTC の運用に関する専門知識は不要

- 140カ国、14 のリージョンで利用可能

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.





# Amazon Chime SDK とは



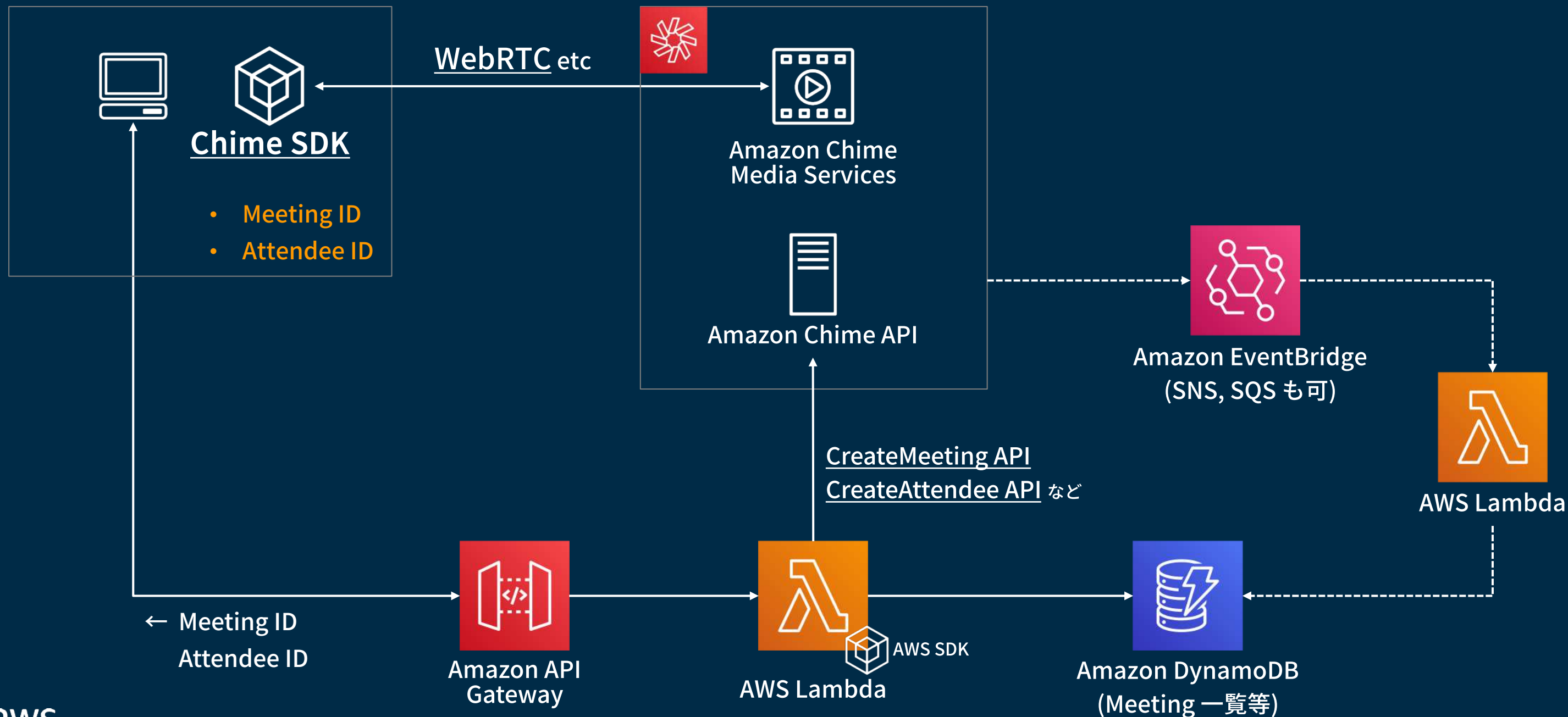
## ■ Amazon Chime SDK で行うこと

- 音声通話、ビデオ通話の接続と制御
- 画面共有の制御
- オーディオ、ビデオデバイスの制御（ミュートなど）
- 会議セッション内のイベントのハンドリング など

## ■ AWS SDK で行うこと

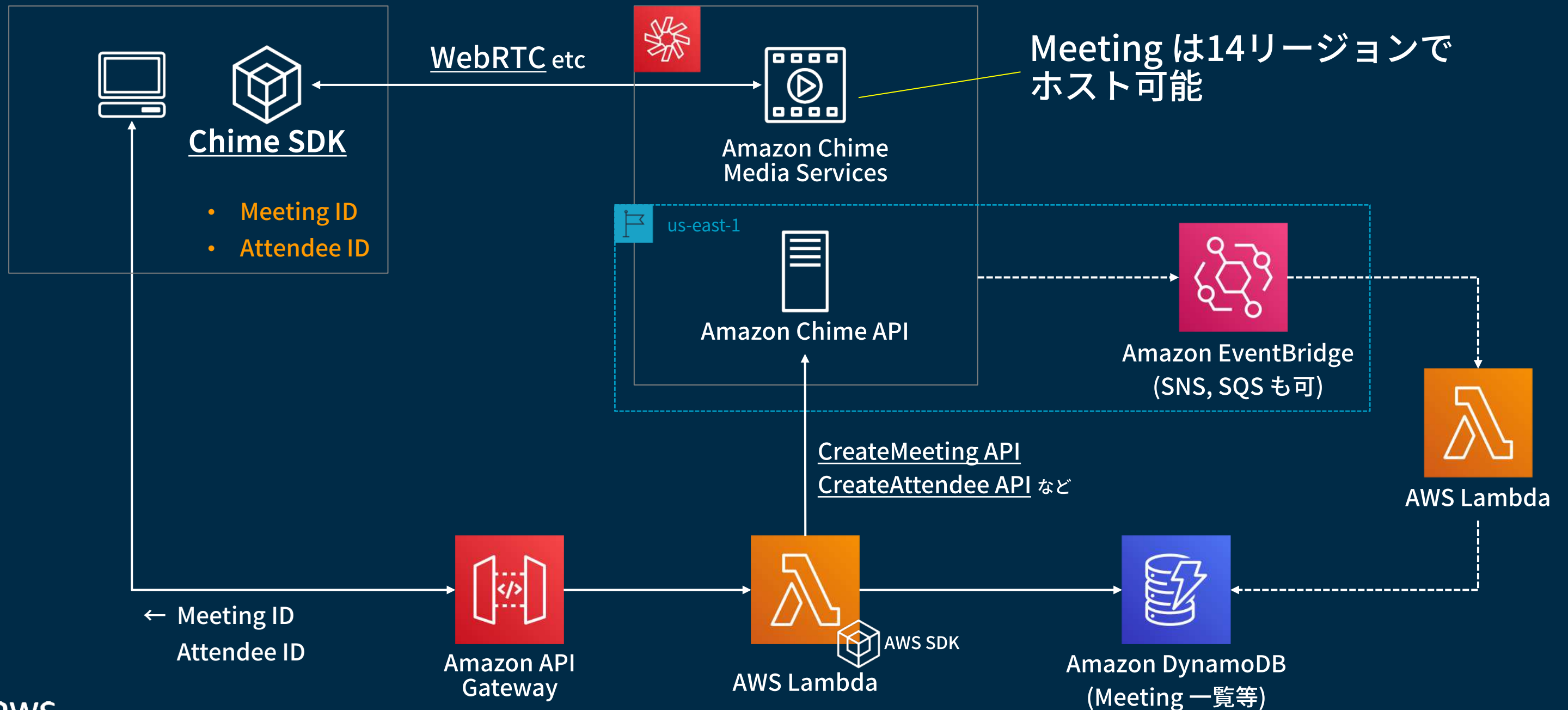
- 会議セッションの作成、終了、参照
- 参加者の追加、削除、参照
- 会議セッションや参加者へのタグの付与 など

# Amazon Chime SDK を利用した構成例

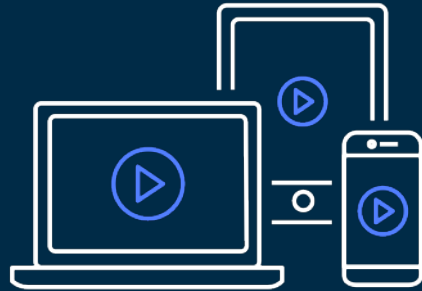




# Amazon Chime SDK を利用した構成例

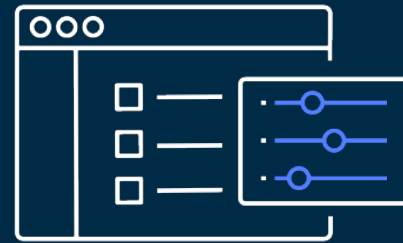


# リアルタイムコミュニケーション機能を迅速に導入



## 素早い構築

SDK を組み込むだけ  
WebRTC の複雑さを抽象化  
迅速な市場投入



## 運用のシンプル化

14の AWS リージョンで利用可能  
エンタープライズのニーズに対応  
事前構成済み  
デプロイ不要



## 既存アプリの拡張

公衆電話網のコール管理  
AI/ML 機能の追加  
イベント駆動のワークフロー

市場投入までの時間、コスト、オーバーヘッドを削減

# Amazon Chime SDK の料金

各「参加者」はどのメディアモダリティ (画面共有、画面表示、双方向音声、一方向音声、双方向ビデオ、または一方向ビデオ) に関係なく、会議セッションに接続される時間で課金されます。

	1分あたりの料金 (各参加者)
Amazon Chime SDK 会議セッションへの接続	0.0017 USD (※)

下記条件に合致した場合は自動的に会議セッションは終了されます

- 音声接続が無い状態で5分以上経過した場合
- 音声接続が2つ未満の状態でも30分以上経過した場合
- 画面共有の閲覧者が30分以上非アクティブの場合
- 会議が開始して24時間経過した場合

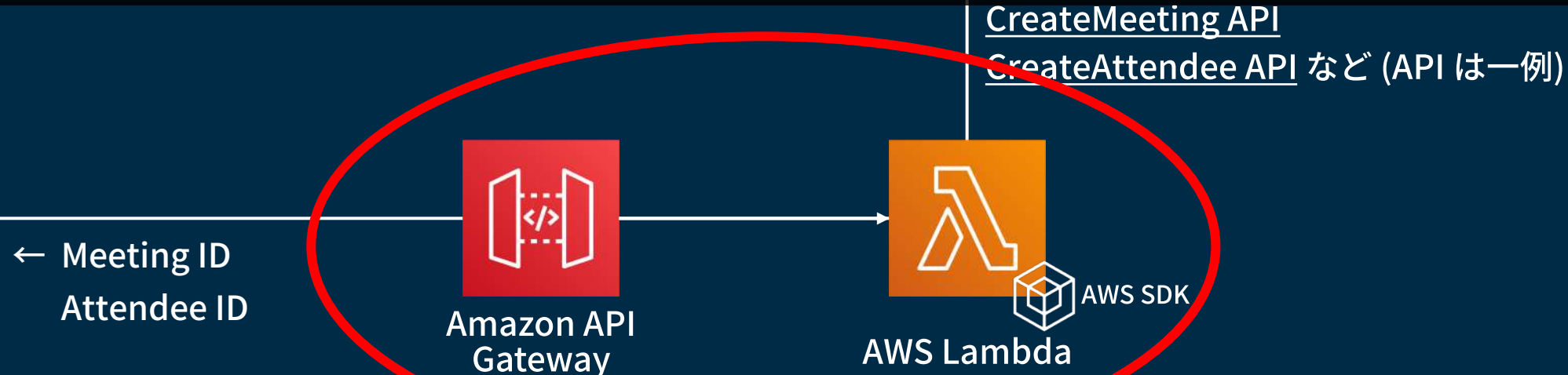
(※) 正確な料金は料金ページを参照してください → <https://aws.amazon.com/jp/chime/pricing/>



# Amazon Chime SDK を利用した構成例

サーバサイドの部分の開発や運用がめんどくさい？

簡単に開発/運用する方法知りたいですよね？



# フロントエンド開発に集中するために




AWS Amplify



CHAL<sup>λ</sup>LENCE



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with  intel.

# フロントエンド開発に集中するために

## AWS Amplify

```
$ amplify add api
? Please select from one of the below mentioned services:
  GraphQL
> REST
? Provide a friendly name for your resource to be used as a
label for this category in the project: ChimeAdminAPI
? Provide a path (e.g., /book/{isbn}): /meeting
? Choose a Lambda source (Use arrow keys)
> Create a new Lambda function
? Provide a friendly name for your resource to be used as a
label for this category in the project: ChimeAdminFunction
? Provide the AWS Lambda function name: ChimeAdminFunction
? Choose the runtime that you want to use: (Use arrow keys)
  .NET Core 3.1
  Go
  Java
> NodeJS
  Python
```

## AWS Chalice (only Python)

```
from chalice import Chalice

app = Chalice(app_name='chime')

@app.route('/')
def index():
    return {'chime': 'saiko'}
```

```
$ chalice new-project chime
```

```
$ chalice deploy
```



# フロントエンド開発に集中するために

AWS DevDay Online Japan 関連セッションのご紹介

## A-3 : フロントエンドエンジニアのための AWS アーキテクチャ

講師：水馬 拓也（アマゾンウェブサービスジャパン株式会社）

セッション概要：本セッションでは主にフロントエンドエンジニアに向けて Single Page Application (SPA) や Server Side Rendering (SSR)、JAMstack などのトピックを中心に AWS をどのようにご活用いただけるかについて紹介します。

## B-1 : AWS Amplify 実践編

講師：Jaga (Daisuke Nagayama)（アマゾンウェブサービスジャパン株式会社）

セッション概要：AWS Amplify 実践編と称し、より実践的な Amplify を用いた開発についてお話しします。少人数で MVP をスピーディに開発するのに留まらず、事業が成長してチームが大きくなってきたときにどう Amplify を使えばよいか？ビジネスの敏捷性をエンジニアリングで支えるために、Amplify はどのような効率的なツール群を提供するか？広範なビジネスニーズに答えるための柔軟性を、Amplify と AWS の175以上のサービス群で実現するには？こうした質問に答える形で、Amplify の魅力をお伝えします。



# Amazon Chime SDK の主な API



# Setup

```
const logger = new ConsoleLogger('MyLogger', LogLevel.INFO);
const deviceController = new DefaultDeviceController(logger);

// You need responses from server-side Chime API. See below for details.
const meetingResponse = /* The response from the CreateMeeting API action */;
const attendeeResponse = /* The response from the CreateAttendee API action */;
const configuration = new MeetingSessionConfiguration(meetingResponse, attendeeResponse);

// In the usage examples below, you will use this meetingSession object.
const meetingSession = new DefaultMeetingSession(
    configuration,
    logger,
    deviceController
);
```

# Mute / Unmute

```
// Mute
meetingSession.audioVideo.realtimeMuteLocalAudio();

// Unmute
const unmuted = meetingSession.audioVideo.realtimeUnmuteLocalAudio();
if (unmuted) {
    console.log('Other attendees can hear your audio');
} else {
    // See the realtimeSetCanUnmuteLocalAudio use case below.
    console.log('You cannot unmute yourself');
}
```

# Attendees

```
const attendeePresenceSet = new Set();

const callback = (presentAttendeeId, present) => {
  console.log(`Attendee ID: ${presentAttendeeId} Present: ${present}`);
  if (present) {
    attendeePresenceSet.add(presentAttendeeId);
  } else {
    attendeePresenceSet.delete(presentAttendeeId);
  }
};

meetingSession.audioVideo.realtimeSubscribeToAttendeeIdPresence(callback);
```



# サンプルアプリケーション (demo)

# サンプルアプリケーション (JavaScript)

**Serverless Demo** (SPA + API Gateway + Lambda + DynamoDB)

<https://github.com/aws/amazon-chime-sdk-js/tree/master/demos/serverless>

**Browser Demo** (SPA only)

<https://github.com/aws/amazon-chime-sdk-js/tree/master/demos/browser>

**Virtual Classroom Application** (Electron)

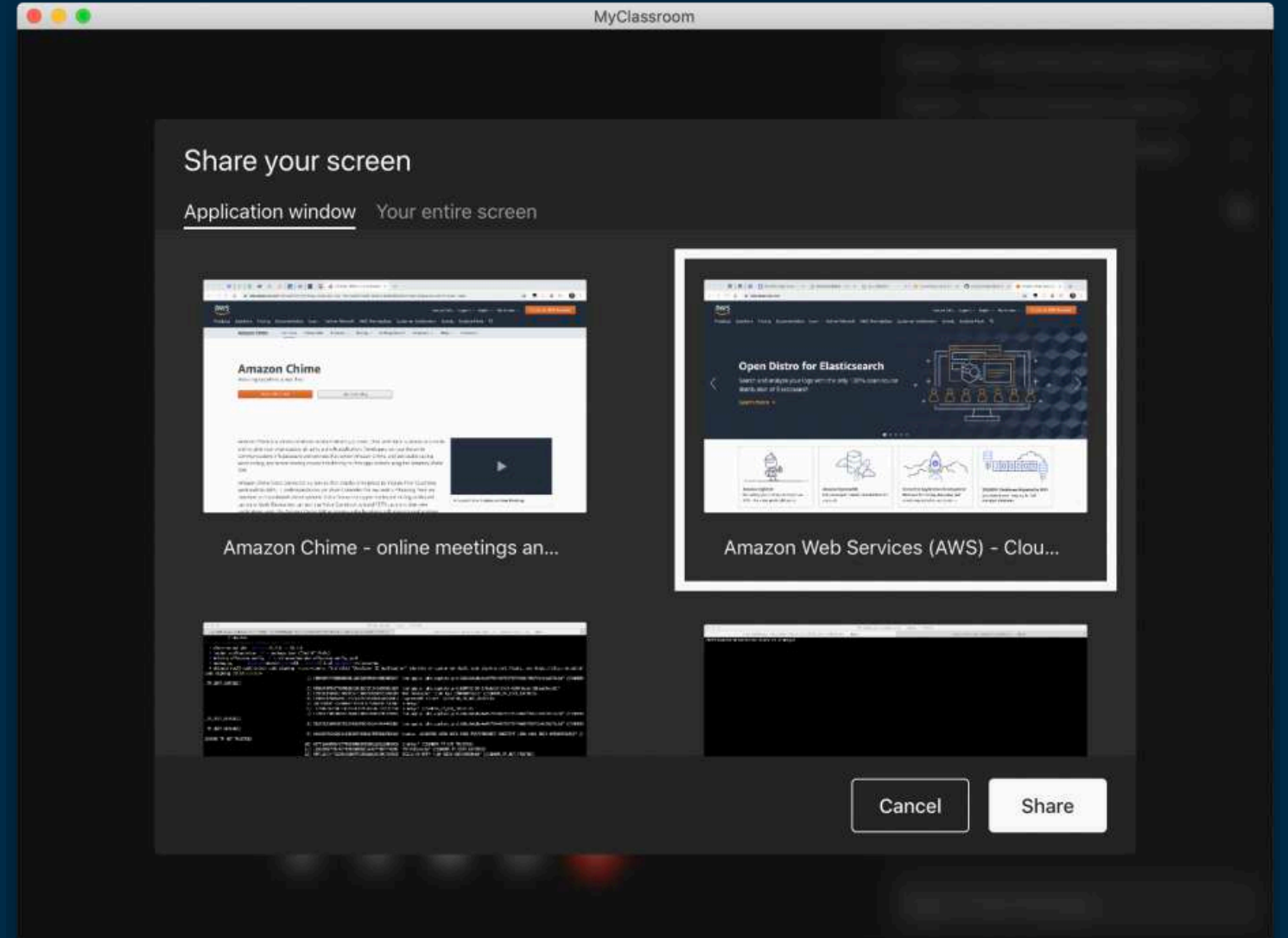
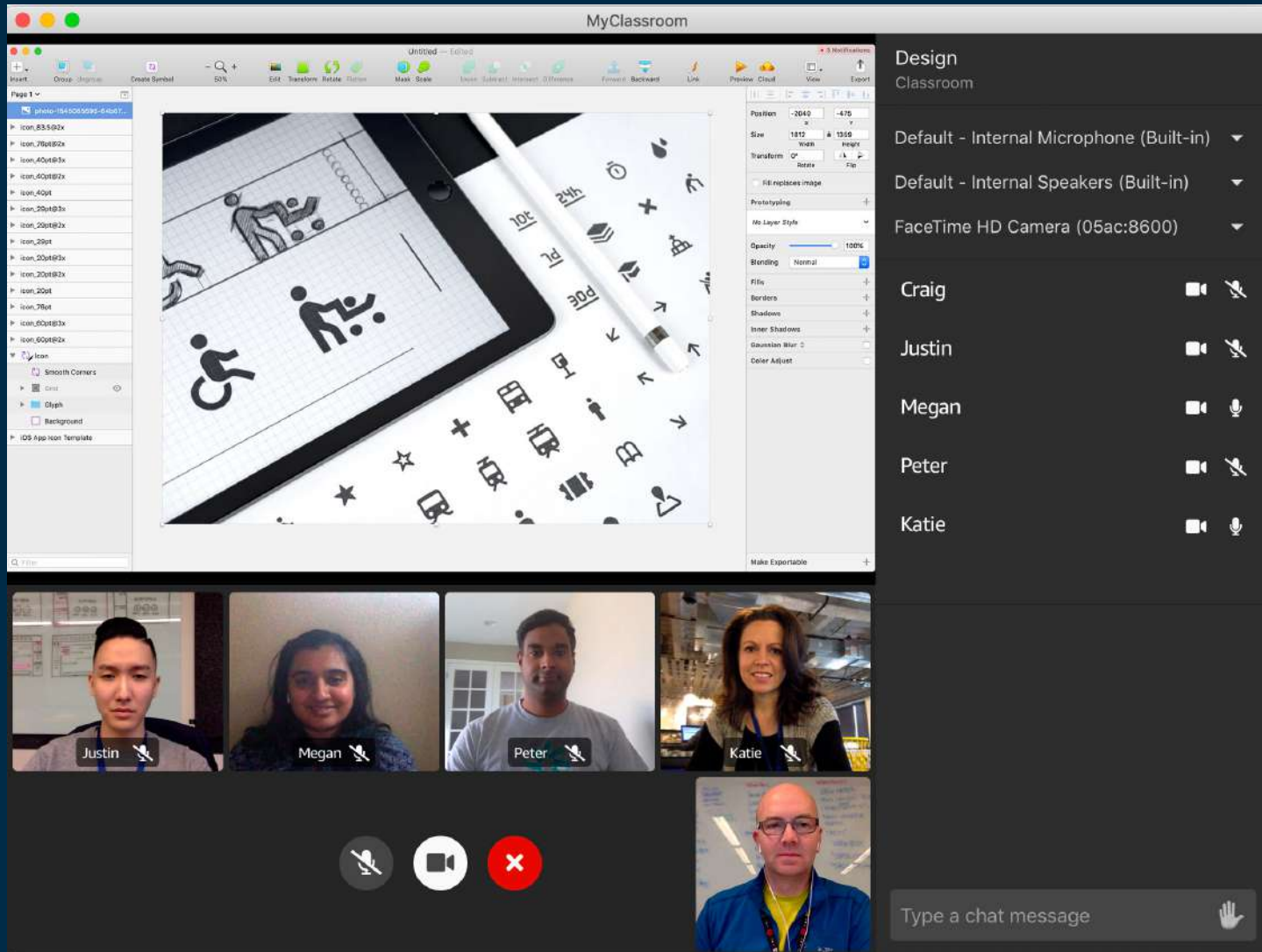
<https://github.com/aws-samples/amazon-chime-sdk-classroom-demo>

Blog: [Building a Virtual Classroom Application using the Amazon Chime SDK](#)



# サンプルアプリケーション (JavaScript)

## Virtual Classroom Application (Electron)



# サンプルアプリケーション (iOS/Android)

## iOS Demo (Swift)

<https://github.com/aws/amazon-chime-sdk-ios#running-the-demo-app>

Blog: [Building a Meeting Application on iOS using the Amazon Chime SDK](#)

## Android Demo (Kotlin)

<https://github.com/aws/amazon-chime-sdk-android#running-the-demo-app>

Blog: [Building a Meeting Application on Android using the Amazon Chime SDK](#)

## React Native Demo

<https://github.com/aws-samples/amazon-chime-react-native-demo>



# まとめ



# まとめ

- Amazon Chime SDK を利用することで、インフラストラクチャの管理をすること無く、リアルタイムコミュニケーション機能を自社サービスに簡単に追加することが出来る
  - Amazon Chime と同じ通信基盤
- Meeting の作成や参加については自社サービスに合わせて柔軟に設計可能
  - AWS SDK を利用して API 等を作成する必要がある
  - AWS Amplify や AWS Chalice を活用することで、サーバサイドもより簡単に
- サンプルが豊富に用意されている
  - ユースケースにあったものを探して Try it now!!



# Appendix

# Amazon Chime SDK for JavaScript

<https://github.com/aws/amazon-chime-sdk-js>

- npm で利用可能なオープンソース JavaScript SDK
- Windows および Mac での WebRTC 対応ブラウザ (Chrome、Edge、Firefox、Safari) および electron アプリのサポート
- カスタマイズ可能なコンポーネントモデル-アクティブスピーカー検出器、デバイスコントローラー、ロガー
- ハードウェアアクセラレーションによるエンコード/デコード
- 名簿、ボリューム、接続状態、および帯域幅変更インジケータのリアルタイムイベント

Web アプリケーション

JavaScript APIs

カスタマイズ可能なコンポーネント

Amazon Chime JavaScript SDK  
(オープンソース)

ブラウザ  
判定

Proto  
BufJs

ブラウザ



# Amazon Chime SDK for iOS/Android

<https://github.com/aws/amazon-chime-sdk-android>

<https://github.com/aws/amazon-chime-sdk-ios>

- Swift および Kotlin ベースの API
- Android > 5.0 and iOS > 10.0
- カスタマイズ可能なオープンソースコンポーネント- アクティブスピーカー、デバイスコントローラー
- オーディオ品質、アクティブスピーカーイベント、強化されたエコーキャンセレーション、ハードウェアアクセラレーションによるエンコード/デコードとレンダリング、帯域調整のためのリアルタイムシグナリング



# Thank you!

Kazuki Matsuda