

# Cloud Streaming Services

## ライブイベントストリーミング（超低 遅延ライブストリーミング）

### 製品ドキュメント



Tencent Cloud

## Copyright Notice

©2013-2022 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

## カタログ：

ライブイベントストリーミング（超低遅延ライブストリーミング）

概要

ユースケース

クイックスタート

SDKへのアクセスの説明

ライブイベントストリーミングと標準ライブストリーミングの違い

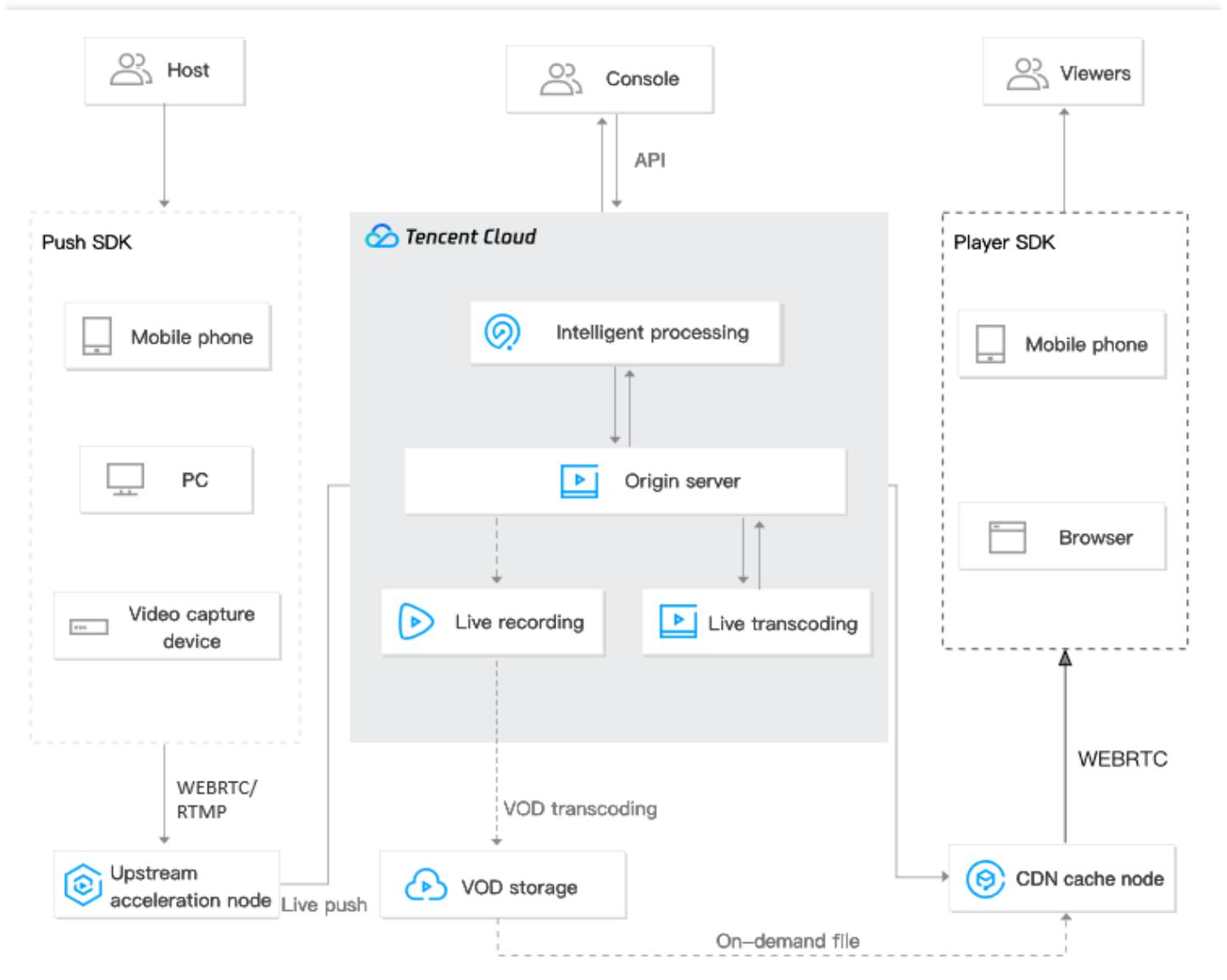
# ライブイベントストリーミング（超低遅延ライブストリーミング）

## 概要

最終更新日：：2022-05-05 15:57:35

ライブイベントブロードキャスト（Live Event Broadcasting、LEB）（超低遅延ライブブロードキャスト）は、超低遅延の再生シナリオでのCSSの拡張です。標準ライブブロードキャストよりもディレーが低く、視聴者にミリ秒レベルのライブブロードキャスト視聴体験を提供します。eラーニング、スポーツイベントのライブブロードキャスト、オンライン質疑応答など、より高いディレーパフォーマンスを必要とする特定のシナリオのニーズを満たすことができます。

## 製品アーキテクチャ



## 製品の特徴

### ミリ秒レベルの超低遅延再生

UDPプロトコルを使用し、従来のライブブロードキャストでの3~5秒のディレイを1秒以内に短縮させます。また、インスタントブロードキャストやラグ率などのコア指標も考慮して、ユーザーに超低遅延のライブブロードキャスト体験を提供します。

### パーフェクトな機能、スムーズな互換性

プッシュ、トランスコード、レコーディング、スクリーンキャプチャ、ポルノ検出、再生といったすべての機能を含む、標準ライブブロードキャストと互換性があり、お客様が既存の標準ライブブロードキャストサービスからスムーズに移行できるようにサポートします。

## シンプルで使いやすく、安全かつ高い信頼性

標準プロトコルが採用されており、ドッキングは簡単で、ChromeやSafariブラウザではプラグインなしで再生できます。再生プロトコルはデフォルトで暗号化されており、より安全で信頼性が高くなっています。

## 製品料金

ライブイベントブロードキャスト課金項目には、基本サービス料金と付加価値サービス料金が含まれています。基本サービス料金は、ライブイベントブロードキャストサービス使用時にアップリンクプッシュおよび再生によって発生するトラフィック/帯域幅料金です。付加価値サービス料金は、ライブブロードキャストのトランスコード、レコーディング、スクリーンキャプチャおよびポルノ検出などの付加価値機能によって発生するリソース消費料金です。課金に関するドキュメントは、[価格一覧](#)をご参照ください。

## 製品へのアクセス

ライブイベントブロードキャストへのアクセスには、Demo体験やクイックアクセスなどの紹介が含まれています。詳細については、[ライブイベントブロードキャストクイックスタート](#)をご参照ください。

# ユースケース

最終更新日：：2021-09-02 10:50:51

## スポーツイベント

ライブイベントストリーミングは、スポーツイベントに超低遅延のライブストリーミング機能を提供し、試合の結果をライブストリーミングで素早くユーザーに届け、視聴者はリアルタイムで状況を知る楽しみを味わうことができます。



## eコマースライブストリーミング

eコマースライブストリーミングでは、商品のオークションやプロモーションの争奪戦などの取り引きのフィードバックにライブストリーミングのリアルタイム性が強く求められます。ライブイベントストリーミングの超低遅延機能により、キャスターや視聴者は取り引きのフィードバックを速やかに得ることができ、「見ながら買う」とい

う体験を向上させます。



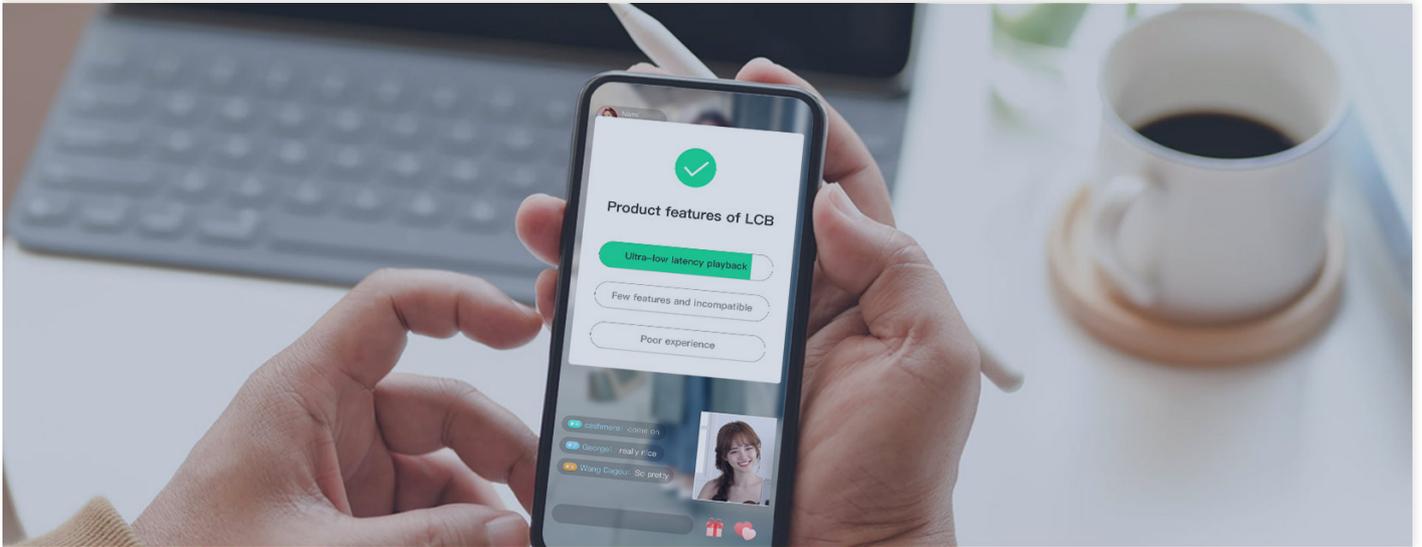
## オンライン授業

教師と学生がライブストリーミングを使って行うオンラインクラスの授業では、ライブイベントストリーミングの超低遅延性能が役立ちます。ライブイベントストリーミングでは教師と学生が双方のライブストリーミング画面でリアルタイムに相手側と同期を取ることができますので、オンラインクラスでも対面しているかのように自然です。



## オンライン解答

従来のクイズなどのオンライン解答は遅延があるため、場合によっては、視聴者側はフレームを補足しないと視聴者とキャスターの両方を同時に表示できませんでした。超低遅延なライブイベントストリーミングであれば、この問題を見事に解決し、双方がリアルタイムに解答画面を視聴できるようになります。実現の難度は下がり、視聴体験もより滑らかです。



## ショーでのインタラクション

ライブイベントストリーミングはライブショーブロードキャストシーンに適していて、視聴者がギフトを贈るなど、画面のリアルタイム性への要求が高いライブストリーミングインタラクティブシーンで、視聴者のインタラクティブ体験を大幅に最適化します。



# クイックスタート

最終更新日：2022-08-29 12:06:25

このチュートリアルにより、ライブイベントストリーミングサービスをすばやく理解することができます。ライブイベントストリーミングサービスをご利用になる前に、ライブイベントストリーミングの[価格一覧](#)をご覧ください。誤解を避けるために、**課金項目**と\***価格**を確認しておくことをお勧めします。

## 準備作業

1. [Tencent Cloudアカウント](#)の登録を行います。
2. [Tencent Cloud CSSサービスアクティブ化ページ](#)に入り、『Tencent Cloudサービス契約』の同意にチェックを入れ、**アクティブ化申請**をクリックすればCSSサービスをアクティブ化できます。

説明：

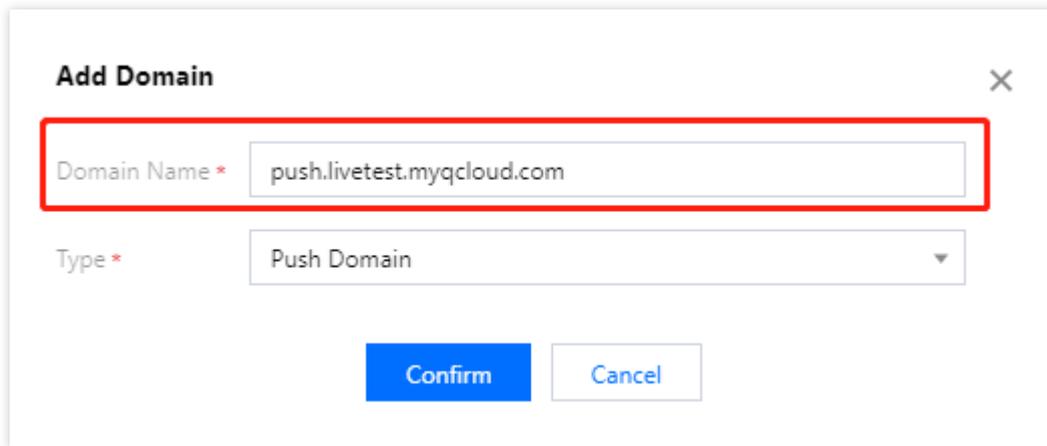
- CSSサービスのアクティブ化申請後、中国本土の再生トラフィック20GBの無料体験がご利用になります。
- ドメイン名の設定は、標準ライブストリーミングと同様です。標準ライブストリーミングにアクセスしている場合は、[ステップ4：再生アドレスの取得](#)部分でライブイベントストリーミングプルについて直接学ぶことができます。

## ステップ1：ドメイン名の追加

CSSサービスを使用するには、少なくとも**2つ**のドメイン名が必要です。1つは**プッシュドメイン名**、もう1つは**再生ドメイン名**です。プッシュと再生に同じドメイン名を使用することはできません。

**独自ドメイン名**を使用して、ICP登録済みの独自ドメイン名を追加することができます。

1. 独自ドメイン名を準備します。中国大陸（中国本土）で使用する場合、ドメイン名のICP登録を済ませている必要があります。
2. CSSコンソールにログインし、[ドメイン名管理](#)に進みます。
3. 独自ドメイン名の追加ページに進み、**ドメイン名の追加**をクリックします。



**Add Domain** ✕

Domain Name \*

Type \*

説明：

- CSSは、デフォルトでテストドメイン名 `xxxxx.livepush.myqcloud.com` を提供しています。このドメイン名でプッシュテストを行うことはできますが、正式なサービスでこのドメイン名をプッシュドメイン名として使用することはお勧めしません。
- ドメイン名の追加に成功すると、**ドメイン名管理**のドメイン名リストでドメイン名情報を確認することができます。追加したドメイン名の管理を行いたい場合は、**ドメイン名管理**をご参照ください。
- その他のCSSドメイン名関連情報については、[CSS基本関連のご質問](#)をご参照ください。

4. ドメイン名の追加に成功すると、システムが自動的にCNAMEドメイン名（`.tlivecdn.com` または `.tlivepush.com` を拡張子とする）を割り当てます。CNAMEドメイン名は直接アクセスできません。ドメイン名サービスプロバイダのところでCNAME設定を完了させる必要があり、設定が有効になるとCSSサービスをご利用いただけます。例として、DNSサービスプロバイダがTencent Cloudとなる場合のCNAMEレコード追加の操作手順は次のようになります。

- [ドメイン名サービスコンソール](#)にログインします。
- CNAMEを追加する必要があるドメイン名を選択し、**解決**をクリックします。
- ドメイン名解決のページに入り、**レコードの追加**をクリックします。
- 新しく追加した列で、ホストレコードとしてドメイン名プレフィックスを入力し、レコードタイプはCNAMEを選択し、レコード値としてCNAMEドメイン名を入力します。
- 保存**をクリックするとCNAMEレコードが追加できます。

注意：

- CNAMEに成功した後は、通常、有効になるまでに一定の時間を要します。CNAMEに成功しない場合はCSSを使用できません。

- ドメイン名のCNAMEに成功すると、CSSコンソールの [ドメイン名管理](#) リストでドメイン名CNAME



のアドレスステータスの記号が  に変更されていることが確認できます。

- CNAME操作後の検出で失敗が続く場合は、ドメイン名登録サービスプロバイダに問い合わせることをお勧めします。
- 他のDNSサービスプロバイダを使用する場合、その他の操作については、[ドメイン名CNAME設定](#) をご参照ください。

## ステップ2：プッシュアドレスの取得

1. CSSツールボックス>[アドレスジェネレーター](#)を選択します。
2. アドレスジェネレーターのページに入り、次の設定を行います。
  - i. 生成タイプは**プッシュドメイン名**を選択します。
  - ii. ドメイン名管理で追加済みのプッシュドメイン名を選択します。
  - iii. カスタマイズされたStreamNameを入力します（例：liveteststream）。
  - iv. アドレス期限を選択します（例： `2021-05-25 23:59:59` ）。
  - v. **アドレスの発行**をクリックすると、プッシュアドレスを発行することができます。

The screenshot shows the 'Address Generator' interface. It includes the following fields and options:

- Domain Type \***: A dropdown menu set to 'Push Domain'. Below it is a text box containing a domain name. A note states: 'If you select push domain, a push address will be generated; and if you select playback domain, a playback address will be generated. If there is no available domain, please [Add Domain](#)'.
- AppName \***: A text box containing 'live'. A note states: 'Use "live" by default. Only letters, digits, and symbols are supported.'
- StreamName \***: A text box containing 'liveteststream'. A note states: 'Only support letters, digits, and symbols.'
- Expiration Time**: A date/time picker set to 'UTC-9' and '2021-07-09 19:07:04'. A note states: 'The expiration time of playback address is the setting timestamp plus the playback authentication expiration time, and the push address expiration time is the setting time.'

At the bottom, there are two buttons: 'Generate Address' (in blue) and 'Address Resolution Sample' (in blue).

説明：

- プッシュアドレスの構成は、 `live` がデフォルトのAppName、 `txSecret` が再生プッシュの署名、 `txTime` がプッシュアドレスの有効時間となります。

- 上記方法以外にも、CSSコンソールの **ドメイン名管理** でプッシュドメイン名を選択して、**管理** をクリックし、**プッシュ設定** を選択して、プッシュアドレスの期限とカスタマイズされたStreamNameを入力し、**プッシュアドレスの発行** をクリックすると、プッシュアドレスを発行することができます。
- 実際の業務のニーズに応じて、プッシュアドレスの発行前に対応する **機能テンプレート** を設定して作成し、プッシュドメイン名に関連付けることができます。付加価値機能の料金については、**価格一覧** をご参照ください。

## ステップ3：CSSプッシュ

業務シナリオに応じて、生成したプッシュアドレスに対応するプッシュソフトウェアに入力することができます。

- PC端末プッシュには、OBSプッシュを使用することをお勧めします。まず、**OBSプラグイン** の設定を行います。プラグインの設定完了後の操作は、**OBSプッシュ** をご参照ください。
- Web端末プッシュには、**Webプッシュ** を使用することをお勧めします。プッシュしたいドメイン名を選択して、カスタマイズしたStreamNameを入力し、アドレスの有効期限を選択してから、カメラをオンにして、**プッシュ開始** をクリックすればOKです。
- モバイルプッシュの場合、TCToolkit Appをダウンロードしてインストールし、**CSSプッシュ** > **CSSプッシュ** を選択して、2次元コードを手入力、またはスキャンしてプッシュアドレスをアドレス編集枠内に取り込み、**プッシュ開始** をクリックすればプッシュが完了します。

説明：

- カスタマイズされたAppは、Tencent Cloudが提供するMLVB SDKと統合して、プッシュ機能を実装することができます。
- ライブイベントストリーミングWebソリューションは、Bフレームのデコードと再生をサポートしていません。詳細については、**Bフレームについて** をご参照ください。

## ステップ4：再生アドレスの取得

1. プッシュに成功した後、**ストリーム管理** > **オンラインストリーム** を選択して、プッシュアドレスの状態を確認し、**プレビュー** をクリックしてオンライン再生を視聴します。
2. **CSSツールボックス** > **アドレスジェネレーター** を選択して再生アドレスを取得し、このページで次のように設定します。

- i. 生成タイプを選択します（例：再生ドメイン名）。
- ii. ドメイン名管理で追加済みの再生ドメイン名を選択します。
- iii. プッシュアドレスと同一のStreamNameを入力します。対応するストリームを再生するためには、再生アドレスのStreamNameとプッシュアドレスのStreamNameを一致させる必要があります。
- iv. アドレス期限を選択します（例： 2021-05-25 23:59:59 ）。
- v. 実際のニーズに応じて、バインドするトランスコードテンプレートを選択します。（トランスコードテンプレートを選択した場合、生成される再生アドレスはトランスコード後のライブストリーミング再生アドレスになります。元のCSSストリームを再生する必要がある場合は、トランスコードテンプレートの選択とアドレスの生成は不要です）
- vi. **アドレスの発行**をクリックすると、再生アドレスを発行することができます。ライブイベントストリーミングのURL形式は、 `webrtc://domain/path/stream_id` です。

Domain Type \* Playback Domain [Dropdown]

If you select push domain, a push address will be generated; and if you select playback domain, a playback address will be generated. If there is no available domain, please [Add Domain](#)

AppName \*

Use "live" by default. Only letters, digits, and symbols are supported.

StreamName \*

Only support letters, digits, and symbols.

Expiration Time UTC-9 2021-07-09 19:07:04 [Calendar]

The expiration time of playback address is the setting timestamp plus the playback authentication expiration time, and the push address expiration time is the setting time.

Transcoding Template [Dropdown] [Cancel Transcoding](#)

If you select a transcoding template, the generated playback address will be the live streaming address after transcoding. If you want to play the original live stream, you don't need to select a transcoding template to generate the address.

[Generate Address](#) [Address Resolution Sample](#)

3. 業務シナリオに応じて、次の方式を用いてCSSストリームが正常に再生できるかどうかテストすることができます。

- **Web端末CSSストリームテスト**： [TCPlayer Demo](#) ツールを使用して再生を体験することをお勧めします。

説明：

- WebRTC Live Demoは複数の解像度機能をサポートします。CSSコンソールの [機能設定](#) > [CSSトランスコード](#) で高精細度-HD、標準-SDのトランスコードテンプレートを設定でき、トランスコードテンプレートを有するWebRTCストリームアドレスをDemo中の対応するカテゴリーに入力した

後、再生をテストします（この機能のテストが不要な場合は、Demo中にWebRTCオリジナルストリームを一つ入力します）。

- CSSトランスコードの操作ガイドとトランスコード料金の内容については、ドキュメント[CSSトランスコード](#)をご参照ください。

- **モバイル端末CSSストリームテスト**：TCToolkit Appをダウンロードしてインストールし、それを開いて**ライブイベントストリーミング再生**を選択し、**プッシュ体験**で取得したライブイベントストリーミング再生アドレスを手入力、またはスキャンして再生ボタンをクリックし、視聴することをお勧めします。

説明：

Appでストリーミングをプッシュする必要がある場合は、MLVB SDKを統合してライブイベントストリーミングサービスを使用することができます。使用中に問題が発生した場合は、[よくあるご質問](#)をご参照ください。

## ステップ5：ライブイベントストリーミング製品へのアクセス

**モバイル端末ソリューション**：Bフレームデコード、AACオーディオ形式をサポートしており、現在、MLVB SDKに統合されています。

## よくあるご質問

### プルURLの生成

ライブイベントストリーミングプルURLとTencent Cloud CSSプルURLは、基本的に同じです。Tencent Cloud CSSプルURLの前にある `rtmp` を `webrtc` に置き換えるだけで済みます。

ライブイベントストリーミングプルURL形式は、`webrtc://domain/path/stream_id` \*\*です。[リンク不正アクセス防止認証](#)が必要な場合、プルURL形式は、`webrtc://domain/path/stream_id?`

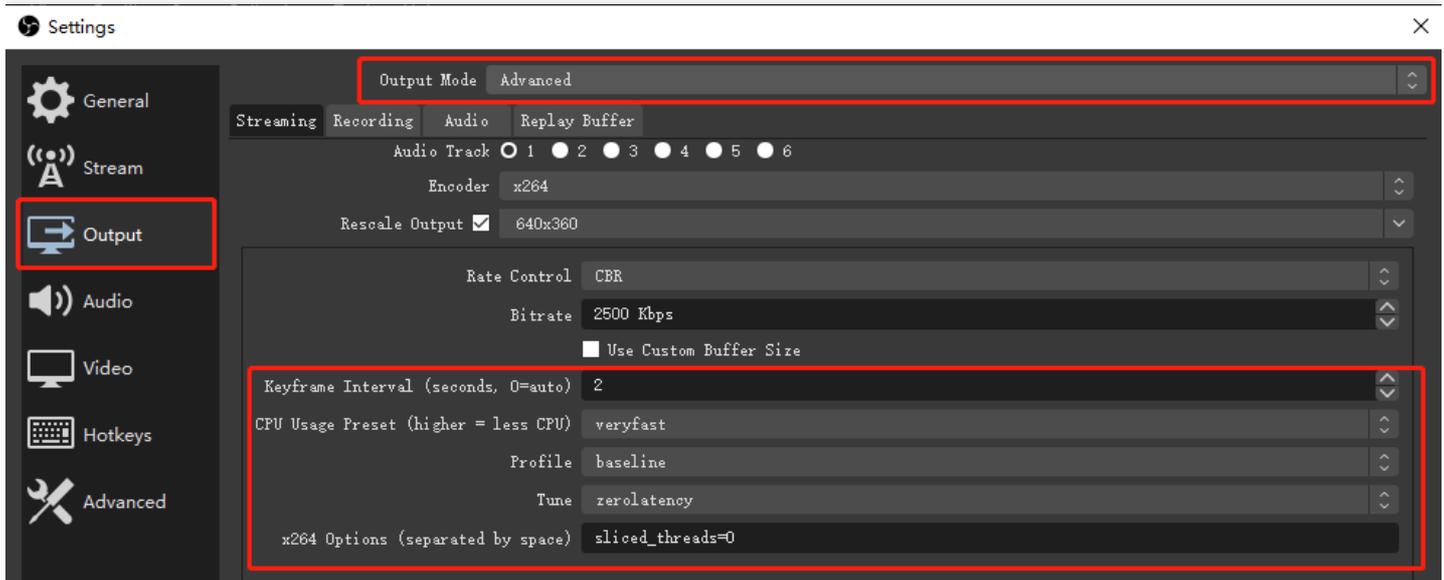
`txSecret=xxx&txTime=xxx` \*\*となります。Tencent Cloud CSSプルURLの生成については、[再生アドレスの取得](#)をご参照ください。

説明：

異なる解像度とビットレートのストリームをプルする必要がある場合は、トランスコードされたストリームをプルできます。トランスコードされたストリームのURL生成については、[CSSカプセル化とトランスコーディング](#)をご参照ください。

## Bフレームについて

ライブイベントストリーミングWebソリューションは、**Bフレームのデコードと再生をサポートしていません**。そのため、元のストリームにBフレームがある場合、バックエンドは自動的にトランスコードしてBフレームを削除しますが、これにより予定外のトランスコード遅延が発生し、**トランスコード料金が発生**します。ですので、Bフレームを含むストリームは、できるだけプッシュしないことをお勧めします。ユーザーは、プッシュストリーミングソフトウェア（OBSなど）のビデオコーデックパラメータを調整することによって、Bフレームを削除することができます。OBSを使用してプッシュする場合は、設定によってBフレームをオフにすることができます。下図のとおりです。



## オーディオトランスコードテンプレートについて

Web端末のブラウザが標準WebRTCプロトコルのみサポートし、AACオーディオ形式をサポートしていない場合は、プッシュされたAACオーディオ形式をOPUSオーディオ形式に変換して再生する必要があります。これにより、**オーディオトランスコード料金**が発生します。

# SDKへのアクセスの説明

最終更新日：：2022-12-23 15:11:47

ライブイベントブロードキャスト(LEB)は、超低遅延の再生シナリオでの標準ライブブロードキャストの拡張です。従来のライブブロードキャストプロトコルよりも低遅延で、視聴者にミリ秒レベルの最高ライブブロードキャストの視聴体験を提供します。

ライブイベントストリーミングサービスをご利用になる前に、[ライブイベントストリーミングのサービス料金](#)をご覧ください、誤解を避けるために、課金項目と価格を確認しておくことをお勧めします。

注意：

ライブイベントストリーミングはWebRTCプロトコルの低遅延特性を使用しているため、デフォルトではBフレームをサポートせず、かつオーディオコーデックはopusコーデックとしています。ライブイベントストリーミングのストリーム再生が可能なことを保証するため、プッシュ時に、Bフレームがある場合またはオーディオコーデックがopusでない場合は、CSSバックエンドが自動的にトランスコーディングを開始してBフレームを削除し、opusエンコードに変換します。これにより[標準トランスコード料金](#)が発生します。

## APP導入

### 導入の説明

iOS、Androidのアプリの場合、ライブブロードキャストSDKを統合することで、App端末でのCSSプッシュ/再生機能を実装できます。

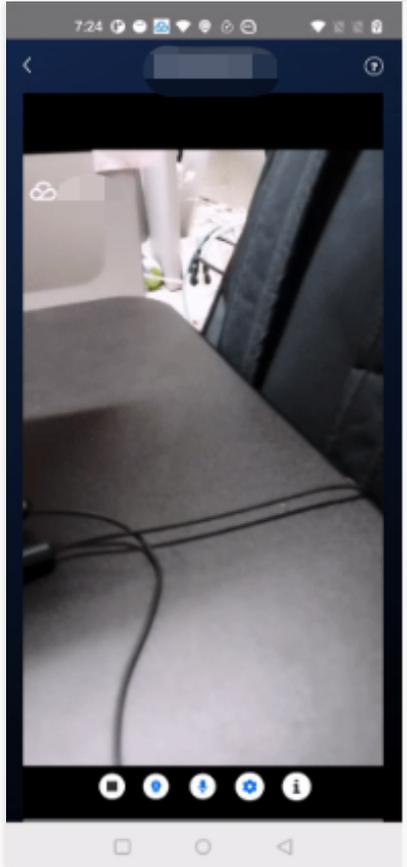
- **App端末のCSSプッシュ**：カメラ画面のキャプチャまたはスマホ画面のキャプチャをサポートし、RTMPプロトコルによって迅速にCSSサービス上にプッシュ可能です。詳細については、[カメラプッシュ](#) および [スクリーンキャプチャプッシュ](#)をご参照ください。
- **App端末CSS再生**：WebRTC再生プロトコルをサポートし、ライブイベントストリーミングサービスと組み合わせて低遅延ライブストリーミング体験を素早く構築します。詳細については、[ライブイベントストリーミングプル](#)をご参照ください。

説明：

ライブブロードキャストSDKはCSS、IM、TRTCなどのサービスの力を借りることで、多人数オーディオビデオの低遅延な相互接続・相互通信を実現することが可能です。多人数コラボ配信によるインタラクティブな効果を実現し、コラボ配信に参加しない視聴者もCSSのサービスを介して視聴することができます。詳細については、[CSSコラボ配信インタラクション](#)をご参照ください。

## Demo体験

ビデオクラウドツールキットは、Tencent Cloudがオープンソース化したオーディオとビデオサービスの完全なソリューションです。ビデオクラウドツールキットを使用すれば、ライブイベントブロードキャストのミリ秒レベルの低遅延プル機能を体験できます。

開発端末	体験版インストール	プッシュデ(Android)	再生デモ(Android)
Android			
iOS	メンテナンス中		

## Webへのアクセス

### 導入の説明

ウェブサイトでCSSプッシュと再生を行う必要がある場合は、次の方式によるアクセスを推奨します：

- **Web端末CSSプッシュ**：ブラウザの一般的なWebRTCの規格をベースに設計とカプセル化を行い、コードスニペットを導入することで、ブラウザの中でCSSプッシュを実現させます。詳細については、[WebRTCプッシュ](#)をご参照ください。

注意：

- WebRTCプッシュの時は、オーディオコーデック方式はopusによるコーデックとなります。標準ライブストリーミングの再生プロトコル(RTMP、FLV、HLS)を使用して再生を行う場合は、正常な視聴を確保するため、CSSバックエンドはオーディオコーデックを自動的に開始してaacに変換し、これによりオーディオトランスコード料金が発生します。詳細については、[オーディオトランスコード料金の説明](#)をご参照ください。（ライブイベントストリーミングのみをご使用の場合、オーディオトランスコードは開始されません）
- WebRTCプロトコルを使用してストリームをプッシュします。各プッシュドメイン名は、デフォルトで**1000パス同時**プッシュ数に制限されています。このプッシュ制限を超える必要がある場合は、[チケット](#)を提出してお申し出ください。

- **Web端末CSS再生**：プレーヤーSDKの[TCPlayer](#)を選択することを推奨します。スマホブラウザとPCブラウザでのライブイベントストリーミングWebRTCプロトコルによるCSSストリームの再生をサポートしています。従来のライブストリーミングプロトコルに比べて、更に遅延が少なく、ミリ秒クラスの究極のライブストリーミング視聴体験を視聴者にご提供いたします。

注意：

WebRTCをサポートしていないブラウザ環境では、プレーヤーに渡されたWebRTCアドレスは、メディア再生をより適切にサポートするために自動的にプロトコル変換されます。デフォルトでは、モバイル端末はHLSに、PC端末はFLVに変換されます。

## Demo体験

- **Web端末CSSプッシュ**：CSSコンソール>[Webプッシュツール](#)により、Web端末プッシュ機能をテストします。
- **Web端末CSSプル**：[TCPlayer](#)ツールを使用して再生を体験いただけます。

説明：

- Web端末CSSプッシュとプルはいずれも標準WebRTCプロトコルを使用します。Web端末プッシュ時はBフレームを含まず、またオーディオコーデックはOPUSオーディオ形式のため、オーディオトランスコードおよびBフレームトランスコード料金は発生しません。
- WebRTC Live Demoは複数の解像度をサポートし、CSSコンソールの[機能設定](#)>[CSSトランスコード](#)でハイビジョン-HD、標準-SDのトランスコードテンプレートを設定し、トランスコードテンプレート

付きのWebRTCストリームアドレスをDemo中の該当する項目に入力した後、再生をテストできます（この機能のテストが不要な場合は、DemoにWebRTCオリジナルストリームを入力します）。

- CSSトランスコードの操作ガイドとトランスコード料金については、ドキュメント[CSSトランスコード](#)をご参照ください。

## OBS WebRTCプロトコルプッシュへのアクセス

WebRTCプロトコルプッシュは、主にビデオクラウドのライブイベントブロードキャスト（超低遅延ライブブロードキャスト）のプッシュに使用され、キャプチャされたオーディオとビデオ画面またはビデオファイルを、WebRTCプロトコルを介してライブブロードキャストサーバーにプッシュする役割を担います。下記の内容では主に、OBSツールを使用してWebRTCプロトコルプッシュ機能を実現する方法について説明します。

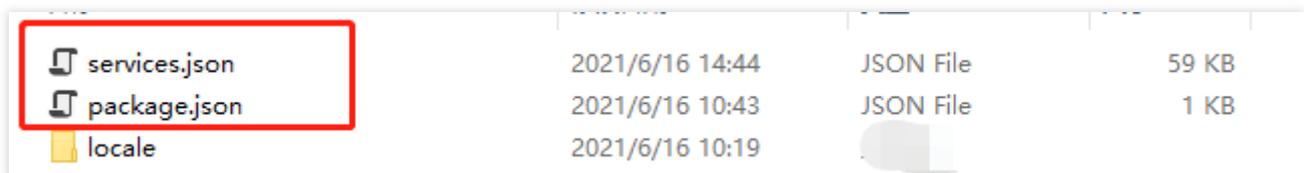
### 注意事項

- 現在、OBS用のバージョンは、26以降のバージョンが必要です。
- WebRTCプロトコルプッシュは現在、OBS用のWindowsプラグインのみです。MacにWebRTCストリーミングを実装する場合は、[Webアクセス](#)を使用できます。

### OBSプラグインの設定

#### 1. プラグインデータを設定します。

- OBSプラグイン**をダウンロードし、dataファイルの中の2つの `services.json` および `package.json` ファイルを、対応する **data > obs-plugins > rtmp-services**ディレクトリに移動して上書きします。（`obs-studio` はデフォルトではCドライブにインストールされており、対応するディレクトリは `C:\Program Files\obs-studio\data\obs-plugins\rtmp-services` です。実際の状況に応じて設定してください。）



 <code>services.json</code>	2021/6/16 14:44	JSON File	59 KB
 <code>package.json</code>	2021/6/16 10:43	JSON File	1 KB
 <code>locale</code>	2021/6/16 10:19		

- これら2つのJSONファイルを `C:\Users\&lt;コンピュータ名>\AppData\Roaming\obs-studio\plugin_config\rtmp-services` ディレクトリにコピーして上書きします。（`&lt;コンピュータ名>` は実際の状況に応じて記入してください）。

#### 2. プラグイン動的ライブラリを設定します。

`obs-plugins\64bit` のdllファイルを、対応する **obs-studio > obs-plugins > 64bit**ディレクトリに移動します。（`obs-studio` はデフォルトではCドライブにインストールされており、対応するディレクトリ

は `C:\Program Files\obs-studio\obs-plugins\64bit` です。実際の状況に応じて設定してください。)

 <code>obs-webrtc.dll.pdb</code>	2021/6/28 17:05	Program Debug...	94,276 KB
 <code>obs-webrtc.dll</code>	2021/6/28 17:05		8,587 KB
 <code>websocketclient.dll</code>	2021/6/16 14:15		582 KB
 <code>libcrypto-1_1-x64.dll</code>	2019/4/28 18:16		2,757 KB
 <code>libssl-1_1-x64.dll</code>	2019/4/28 18:16		667 KB

## プッシュリンクの設定

1. WebRTCプッシュアドレスを発行します。
2. Tencent Cloudライブブロードキャストコンソールにログインし、[ライブブロードキャストツールボックス](#)>[\\*\\*アドレスジェネレーター](#)でプッシュアドレスを発行します。具体的な操作については、[アドレスジェネレーター](#)をご参照ください。
3. 発行したアドレスのプレフィックス `rtmp` を `webrtc` に変更します。具体的な使用説明については、[自身でのライブストリーミングURLの結合](#)をご参照ください。

Domain Type \* Push Domain XXXXXXXXXXXXXXX

If you select push domain, a push address will be generated; and if you select playback domain, a playback address will be generated. If there is no available domain, please [Add Domain](#)

AppName \*

Use "live" by default. Only letters, digits, and symbols are supported.

StreamName \*

Only support letters, digits, and symbols.

Expiration Time UTC+8 2021-09-02 15:25:00

The expiration time of playback address is the setting timestamp plus the playback authentication expiration time, and the push address expiration time is the setting time.

[Generate Address](#) [Address Resolution Sample](#)

**Generation Result** (Generate the following address according to the above settings)

Type	Push Domain
Expiration Time	2021-09-02 15:25:00 (UTC+8) <a href="#">reference documentation</a> <b>Replace rtmp:// with webrtc://</b>
Push Address	<b>rtmp://</b> <span>XXXXXXXXXXXXXXX</span> /live/WebRTC?txSecret= <span>XXXXXXXXXXXXXXX</span> &txTime=61307C4C
OBS Push Address	rtmp:// <span>XXXXXXXXXXXXXXX</span> /live/
OBS Push Name	WebRTC?txSecret= <span>XXXXXXXXXXXXXXX</span> &txTime=61307C4C

#### 4. OBSプッシュサービスを設定します。

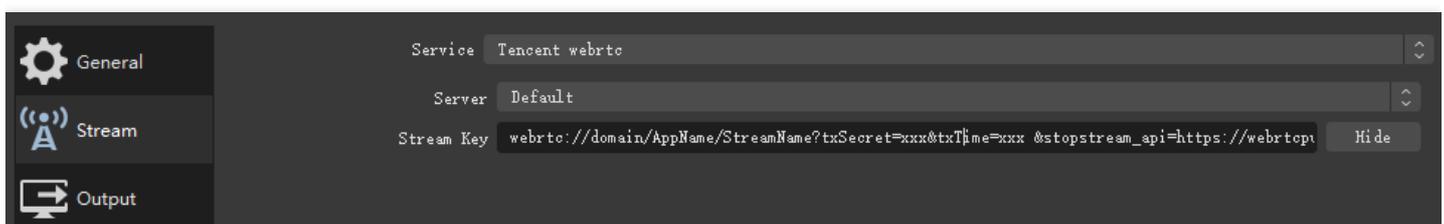
5. OBSを開いて、1番下のツールバーの**コントロール**>**設定**ボタンから設定画面に入ることができます。

6. **プッシュ**をクリックしてストリーム設定タブに入り、サービスタイプは `Tencent webrtc`、サーバーは `Default` を選択し、ストリームキーに上記で発行した**WebRTCプッシュアドレス**を入力し、その後で `&stopstream_api=https://webrtcpush.myqcloud.com/webrtc/v1/stopstream` を結合します。

#### ストリームキーの例：

```
webrtc://domain/AppName/StreamName?txSecret=xxx&txTime=xxx&stopstream_api=https://webrtcpush.myqcloud.com/webrtc/v1/stopstream
```

下図に示すように：



## ライブイベントストリーミングのプル再生

ライブイベントストリーミングSDKを統合してプル再生を行います。具体的には[ライブイベントストリーミングプル](#)をご参照ください。

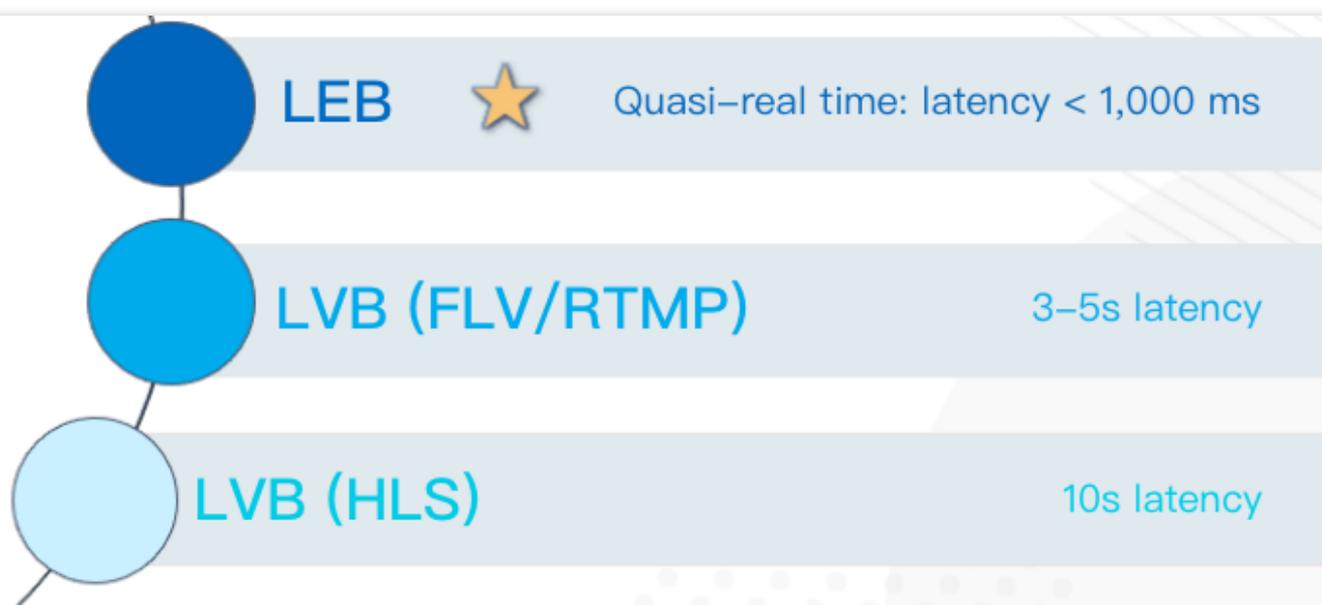
# ライブイベントストリーミングと標準ライブストリーミングの違い

最終更新日： : 2022-05-05 15:57:35

ライブイベントブロードキャストは、超低遅延の再生シナリオでの標準ライブブロードキャストの拡張です。従来のライブブロードキャストプロトコルよりもディレーが低く、視聴者にミリ秒レベルの最高ライブブロードキャスト視聴体験を提供します。この製品は、eコマースのライブブロードキャストとeラーニングに加えて、スポーツイベントのライブブロードキャストやゲームのライブブロードキャストなど、リアルタイムのインタラクションを統合できるさまざまなライブブロードキャストシナリオも含む、より高いディレーパフォーマンスを必要とする特定のシナリオのニーズを満たすことができます。

## プロトコルの比較

現時点では、標準ライブブロードキャストは通常、RTMP、FLV、HLSといった形式の再生プロトコルを使用します。また、これらの再生プロトコルには、いずれもTCPプロトコルに基づいているという共通点があります。TCPには、遅延確認とピギーバック応答があります。例えば、データが送られてくると直ちにそれぞれのデータに対してACKで応答するのではなく、データがある程度溜まってから応答するため、遅延が検知されます。脆弱なネットワーク環境では、TCPなどのメカニズムによってデータが滞留し、データの伝送中に閉塞が起これ、数秒から数十秒の遅延が発生することがあります。



ある調査によると、業界での低遅延ライブブロードキャストのプロトコルには、QUIC、SRT、WebRTCおよびORTCがあることが判明しました。比較すると、QUICにはストリーミングメディア機能がないため、遅延は比較的大きく、SRT、WebRTCおよびORTCにはストリーミングメディア機能があるため、遅延はすべてミリ秒レベルです。このうちSRTとORTCの使用量が少なく、WebRTCエコシステムがよく使われているため、ライブイベントブロードキャストではWebRTCを使用して超低遅延を実現し、WebRTCの基盤にはUDPプロトコルを使用しています。

## UDP Is Essential for Low-Latency Live Streaming

Tencent Cloud Audio/Video Solution 2021

UDP	Latency	Streaming Media Characteristics	Ecosystem
<a href="#">quic</a>	0.5–10s	No	Supported by most browsers
<a href="#">srt</a>	Millisecond	Yes	Immature ecosystem, supported by few browsers
WebRTC	Millisecond	Yes	Mature ecosystem, supported by most browsers
ORTC	Millisecond	Yes	Immature ecosystem, supported by few browsers



More than 93% of browsers have supported WebRTC.

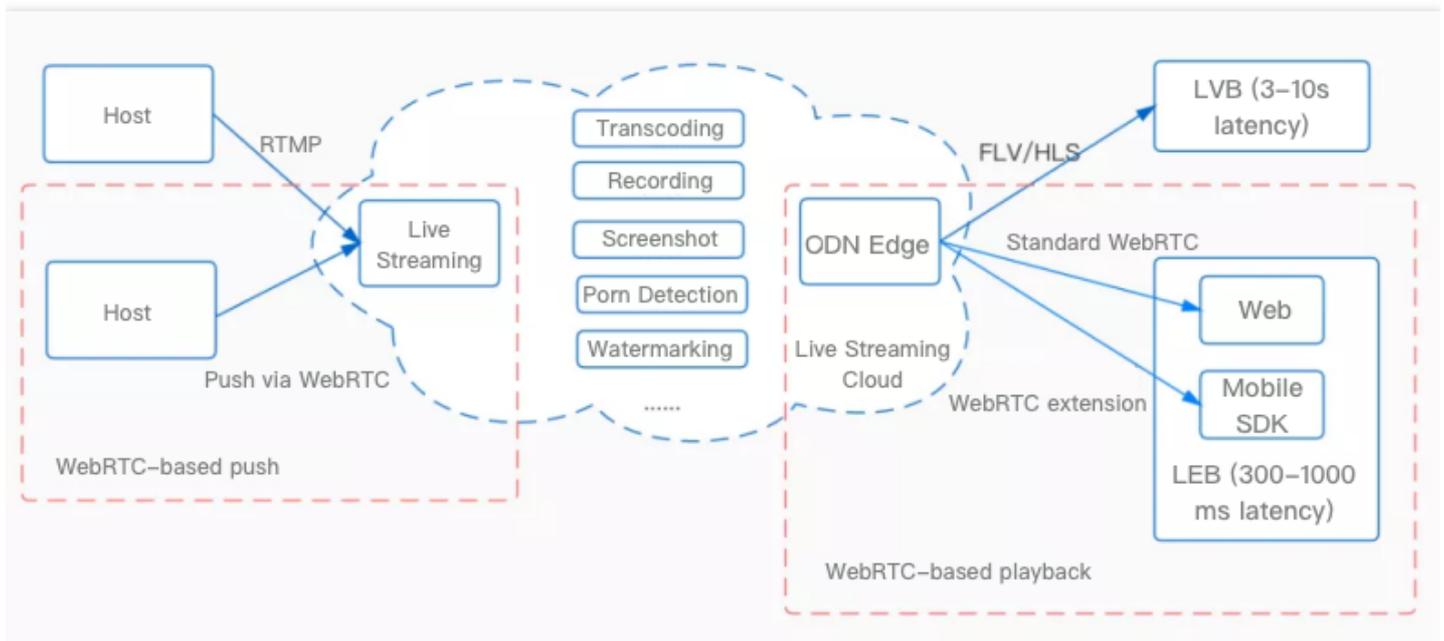
**WebRTC is the future of low-latency live streaming.**

## 遅延の比較

現在、標準ライブブロードキャストのFLVプロトコルの遅延は通常約2秒～10秒であり、その遅延要因は主に、GOPサイズとTCPの脆弱なネットワーク伝送による滞留です。HLSの遅延は更に大きく、通常数秒～数十秒で、その遅延要因は主に、GOPサイズとTSサイズです。HLSはファイル形式でインデックスおよびダウンロードされるため、1ファイルあたりのサイズによって遅延が制限され、多くのプレーヤーは再生するために3つのTSを待たなければなりません。3つのTSは数十秒かかる場合があるため、HLSは標準ライブブロードキャストの中でも最も遅延が大きくなります。

ライブイベントブロードキャストでWebRTCを使用して低遅延改良を行う場合に考慮すべき重要な事項です。ChromeやSafariなど人気のあるブラウザのほとんどは、WebRTC標準をサポートしています。また、成熟したオープンソースのWebRTC SDKでは、最適化とカスタマイズを簡単に行えます。これにより、ブラウザを通じて標準のWebRTCライブブロードキャスト機能を提供することに加えて、カスタマイズされたSDKを通じてアップグ

レードされパーフェクトな低遅延ライブブロードキャスト機能（ライブイベントブロードキャストの遅延は通常、300ms~1000ms）を提供することもできます。



## ライブイベントブロードキャストの優位性

- グローバルな配信と広範なカバレッジを持つライブイベントブロードキャストのスーパーアクセラレーションノード（2100以上のノードをサポート、25か国をサポート）。
- 超広帯域幅の容量（100T以上の帯域幅をサポート）。
- 高品質、低コスト、パーフェクトな機能（30%のパケットロス防止）。
- 簡単にアクセス（SDKを再生するだけで改良が完了、パーフェクトな機能、スムーズな互換性）。
- インスタントブロードキャスト、超低遅延、以下のテスト結果図から、現在のライブイベントブロードキャストで実現可能な遅延は通常約300ミリ秒であり、限界は43ミリ秒であることがわかります。

