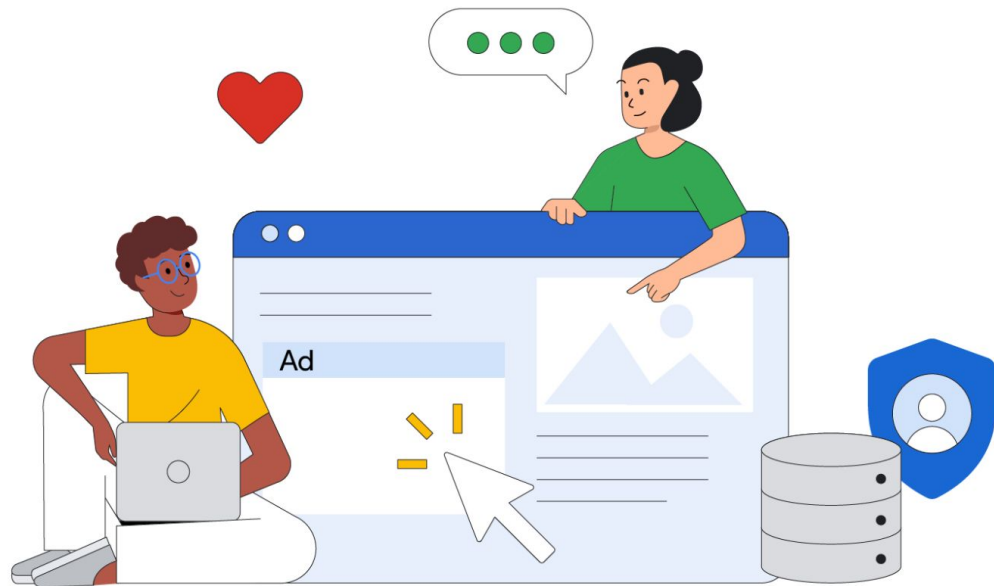# Google consent management solutions

Google

Agenda

1 Yêu cầu về quản lý sự đồng ý của Google và các mốc thời gian quan trọng

2 Giới thiệu qua về giải pháp Consent Management của Google

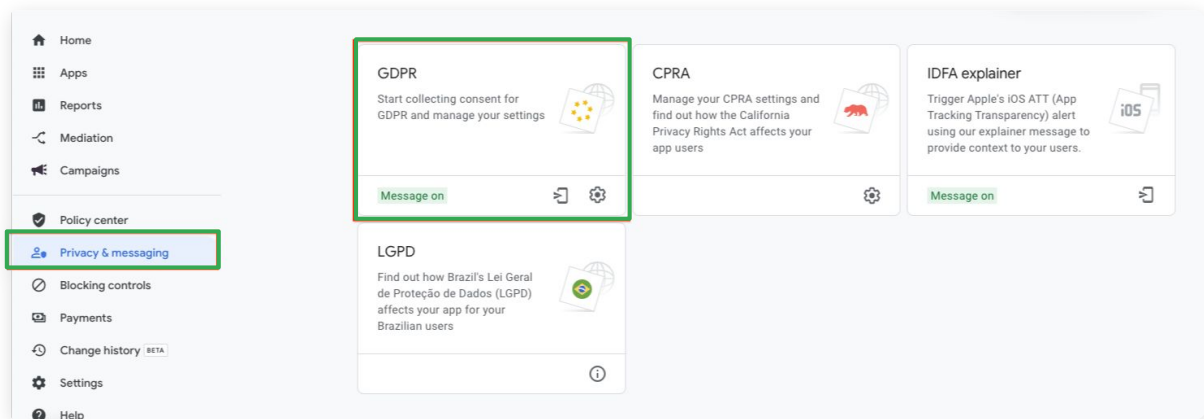**3** Cách tích hợp giải pháp Consent Management của Google

4 Đánh giá ảnh hưởng

Google

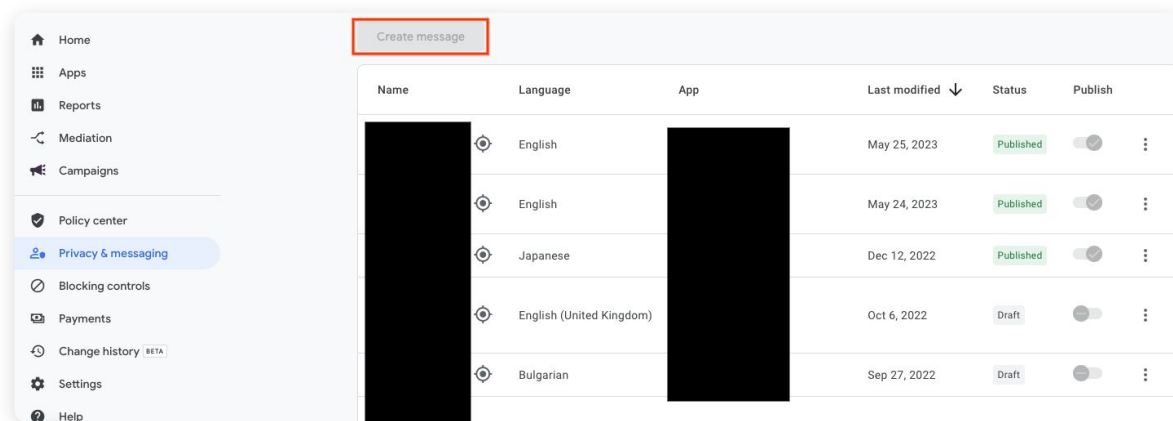# Configure GDPR messaging

Setting up a GDPR message in AdMob (1/4)

1. Click **Privacy & messaging**

2. Click **GDPR**

Google

Setting up a GDPR message in AdMob (2/4)

1. Click **Create message**. The **GDPR message page** opens.

2. **Select the apps** in which this message will be displayed:

   - Click **Select apps**

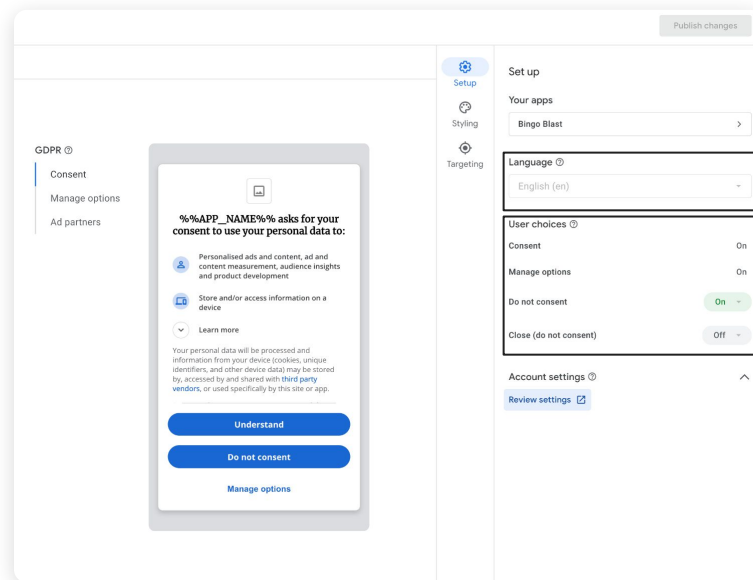   - Select the **desired apps**

   - Click **Save**

Setting up a GDPR message in AdMob (3/4)

1. **Select the languages** in which this message will be displayed:

    - **Default language:** The message is shown in the default language when the user's device language cannot be determined, or when the user's device language doesn't adequately match the selected "Additional languages."

    - **Additional languages:** The message can be shown to users in any of the "Additional languages" that match the user's device language. Languages you select here should match the languages in which your app is available.

2. In the **User choices section**, **select one** of the following:

    - **Consent or Manage options:** This message presents your users with a choice to consent to your ad partners and your ad partners' purposes, or to customize their consent selections.

    - **Consent, Do not consent, or Manage options:** This message presents your users with a choice to consent to your ad partners and your ad partners' purposes, not consent to your ad partners and your ad partners' purposes, or to customize their consent selections.

NOTE:
Work with your legal team to determine how GDPR apply to your business and impact your compliance strategy

Google

Setting up a GDPR message in AdMob (4/4)

1. In the **Targeting section**, **select** one of the following options:

   - **Countries subject to GDPR (EEA and UK):** Display the GDPR message to app visitors in the EEA and UK

   - **Everywhere:** Display the GDPR message to all app visitors
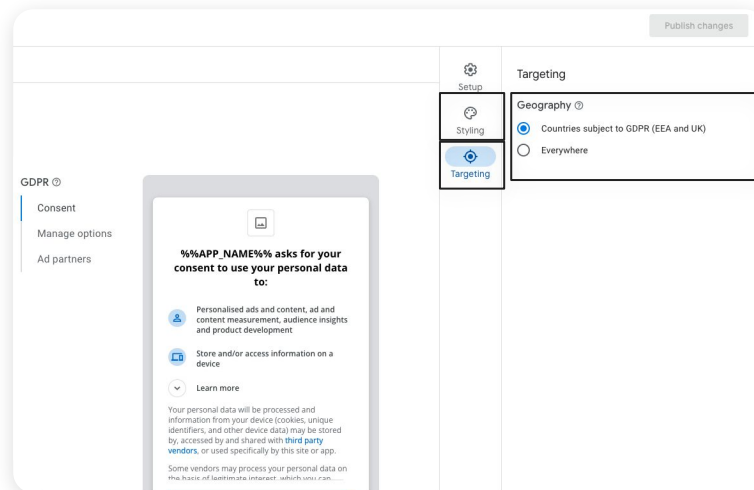
2. Click **Continue**. The **Edit message page** opens.

3. *(Optional)* **Edit and format** **the message** to match your app's editorial and visual standards

4. **Add a privacy policy URL**

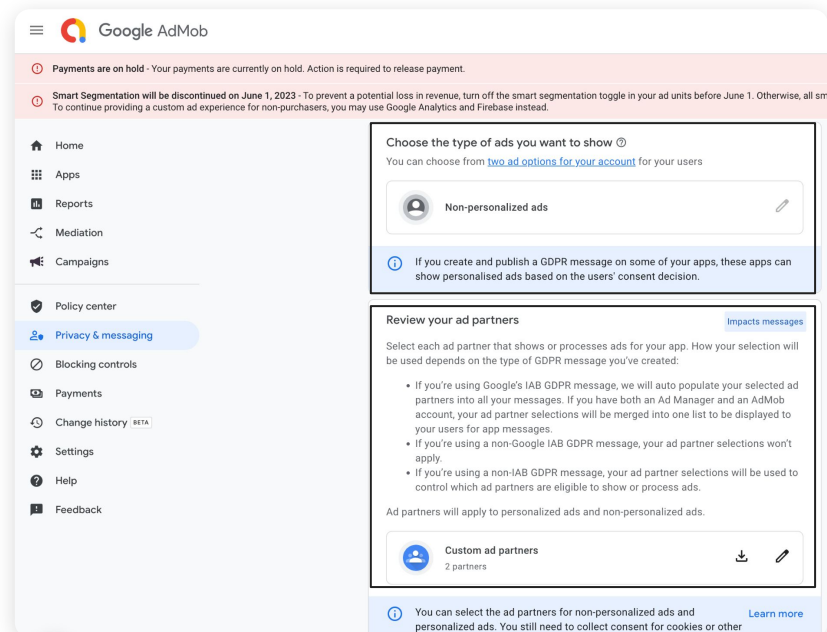5. Click **Save**

6. When you're ready, **publish the message**

NOTE:
Work with your legal team to determine how GDPR apply to your business and impact your compliance strategy

Google

Settings and controls

Select ad technology providers (for personalized ads):

- To help with compliance under Google's updated EU User Consent Policy, you have the option to select your preferred ad technology providers from a list of companies that have provided us with information about their compliance with the GDPR—all of whom also have to comply with our data usage policy to ensure publisher data is protected. If you select these ad technology providers (including Google and other bidders and vendors), they may use data about your users for the purposes of ads personalization and measurement.
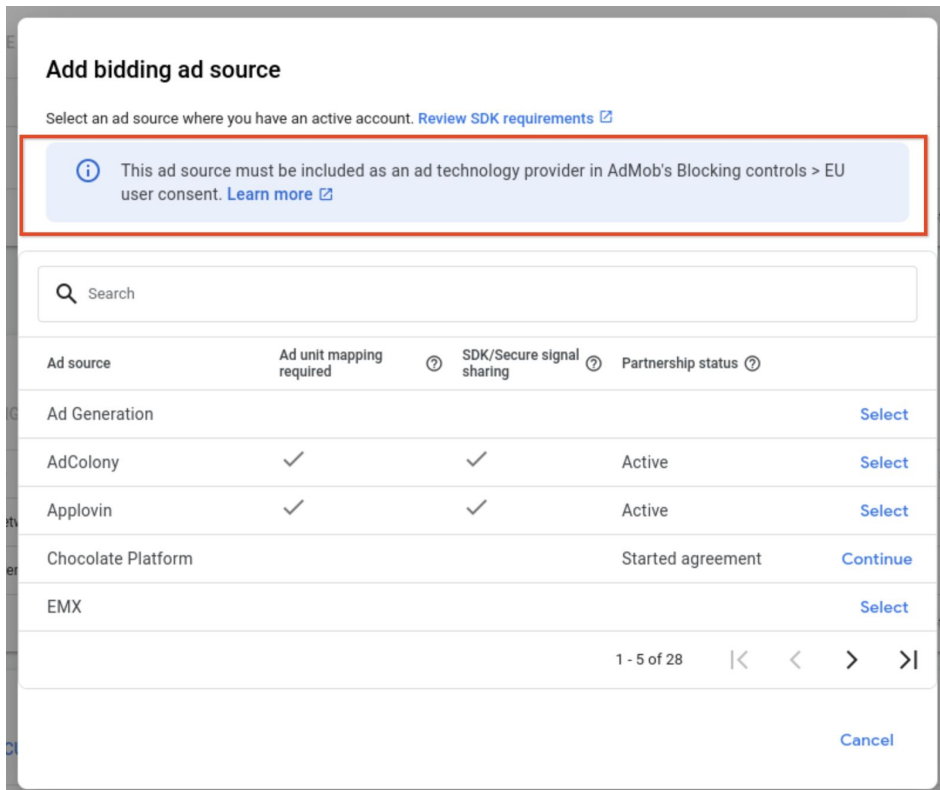

NOTE:
Work with your legal team to determine how GDPR apply to your business and impact your compliance strategy

Settings and controls

- Some biddings like Pangle, Mintegral are not yet in the common partner list
- There is a notification to inform publisher about this when adding bidding ad source

Google User Messaging Platform (UMP) SDK implementation

Google

Requirements to begin

- GDPR consent message created in the Privacy & messaging tab on the AdMob or Ad Manager dashboard

- Google User Messaging Platform (UMP) SDK
  - On Android:
    - Install with Gradle: Add the dependency for the Google User Messaging Platform SDK to your module's app-level Gradle file, normally app/build.gradle:

```
dependencies {
    implementation 'com.google.android.ump:user-messaging-platform:2.1.0'
}
```

  - On iOS:
    - Install using CocoaPods: The UMP SDK is included as a dependency of the Google Mobile Ads SDK pod starting with Google Mobile Ads SDK 7.64.0.
    - The easiest way to import the UMP SDK into an iOS project is to use CocoaPods. Open your project's Podfile and add this line to your app's target, then update the repo:

```
pod 'Google-Mobile-Ads-SDK'
```

Then, run the following command:

```
$ pod install --repo-update
```

Request Latest Consent Information

```swift
// Create a UMPRequestParameters object.
let parameters = UMPRequestParameters()
// Set tag for under age of consent. false means users are not under age
// of consent.
parameters.tagForUnderAgeOfConsent = false

// Request an update for the consent information.
UMPConsentInformation.sharedInstance.requestConsentInfoUpdate(with: parameters) {
  [weak self] requestConsentError in
  guard let self else { return }

  if let consentError = requestConsentError {
    // Consent gathering failed.
    return print("Error: \(consentError.localizedDescription)")
  }

  // TODO: Load and present the consent form.
}
```

iOS

```java
// Set tag for under age of consent. false means users are not under age
// of consent.
ConsentRequestParameters params = new ConsentRequestParameters
    .Builder()
    .setTagForUnderAgeOfConsent(false)
    .build();

consentInformation = UserMessagingPlatform.getConsentInformation(this);
consentInformation.requestConsentInfoUpdate(
    this,
    params,
    (OnConsentInfoUpdateSuccessListener) () -> {
      // TODO: Load and show the consent form.
    },
    (OnConsentInfoUpdateFailureListener) requestConsentError -> {
      // Consent gathering failed.
      Log.w(TAG, String.format("%s: %s",
          requestConsentError.getErrorCode(),
          requestConsentError.getMessage()));
    });
```

Android

You should request an update of the user's consent information at every app launch using **requestConsentInfoUpdate()**. This determines whether your user needs to provide consent if they haven't done so already, or if their consent has expired.

Load & Show Form if Required

```java
UserMessagingPlatform.loadAndShowConsentFormIfRequired(
    this,
    (OnConsentFormDismissedListener) loadAndShowError -> {
        if (loadAndShowError != null) {
            // Consent gathering failed.
            Log.w(TAG, String.format("%s: %s",
                loadAndShowError.getErrorCode(),
                loadAndShowError.getMessage()));
        }

        // Consent has been gathered.
    }
)
```

Android

```swift
UMPConsentForm.loadAndPresentIfRequired(from: self) {
    [weak self] loadAndPresentError in
    guard let self else { return }

    if let consentError = loadAndPresentError {
        // Consent gathering failed.
        return print("Error: \(consentError.localizedDescription)")
    }

    // Consent has been gathered.
}
```

iOS

Call **loadAndShowConsentFormIfRequired()** on the ConsentForm class to load a consent form. If the consent status is required, the SDK loads a form and immediately presents it from the provided activity.

The callback is called after the form is dismissed. If consent is not required, the callback is called immediately.

Consent revocation

Consent revocability is a requirement of the Privacy & messaging user consent program. You must provide a link in your app's menu that allows users who want to revoke consent to do so, then present the consent message to those users again.

To accomplish this:

1. Load a form every time the user launches your app, so that the form is ready to display in case the user wishes to change their consent setting.
2. Present the form when the user selects the link in your app's menu.

⚠️ We support 23 languages across EEA. Please ensure to add a consent revocation link on your app

Testing - Force a geography

The UMP SDK provides a way to test your app's behavior as though the device was located in the EEA or UK using the **setDebugGeography** method on **ConsentDebugSettings.Builder**.

You must provide your test device's hashed ID in your app's debug settings to use the debug functionality. If you call **requestConsentInfoUpdate()** without setting this value, your app logs the required ID hash when run.

```java
ConsentDebugSettings debugSettings = new ConsentDebugSettings.Builder(this)
    .setDebugGeography(ConsentDebugSettings
        .DebugGeography
        .DEBUG_GEOGRAPHY_EEA)
    .addTestDeviceHashedId("TEST-DEVICE-HASHED-ID ✏")
    .build();

ConsentRequestParameters params = new ConsentRequestParameters
    .Builder()
    .setConsentDebugSettings(debugSettings)
    .build();

consentInformation = UserMessagingPlatform.getConsentInformation(this);
consentInformation.requestConsentInfoUpdate(
    this,
    params,
```

Android

```swift
let parameters = UMPRequestParameters()
let debugSettings = UMPDebugSettings()
debugSettings.testDeviceIdentifiers = ["TEST-DEVICE-HASHED-ID"]
debugSettings.geography = UMPDebugGeography.EEA
parameters.debugSettings = debugSettings
UMPConsentInformation.sharedInstance.requestConsentInfoUpdate(
    with: parameters,
```
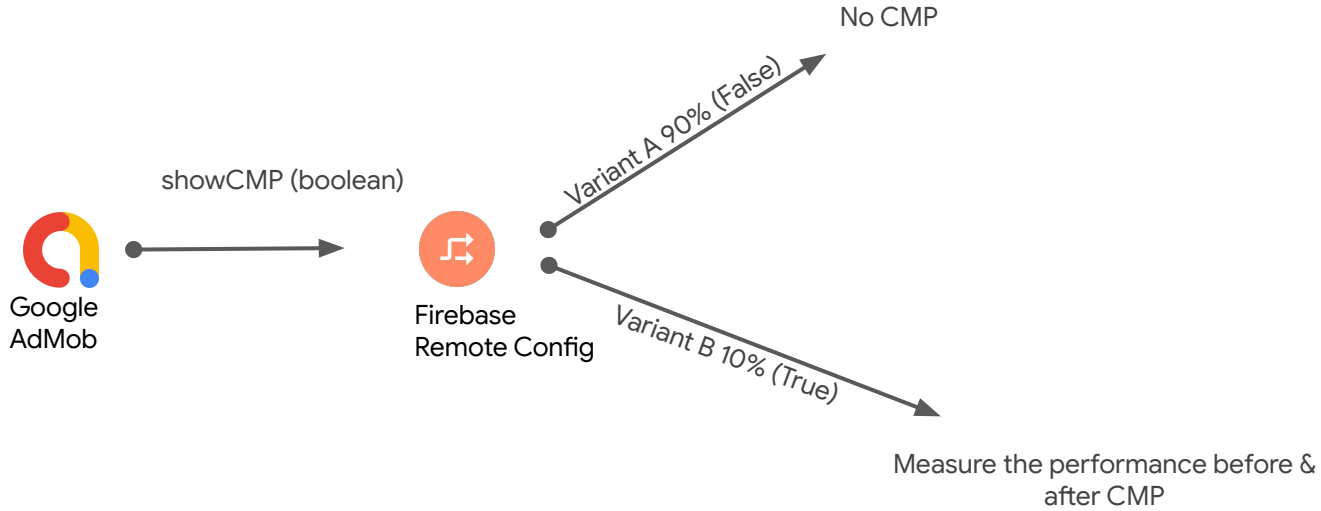
iOS

Section 4

Evaluating Impact

Google

A/B testing & percentage rollout (Recommended)

Google AdMob

showCMP (boolean)

Firebase Remote Config

Variant A 90% (False)

No CMP

Variant B 10% (True)

Measure the performance before & after CMP

Consent rate

- Can be found under Privacy & messaging > GDPR
- Data are aggregated over last 30 days

Impact evaluation with Timed Test

Look at following metrics for versions before / after CMP then check whether the gap between them match non-consent rate?

- DAV%
- Impression / DAU
- Impression / DAV (optional)
- Match rate%

Build comparison report

A	B	C	D
	1.1.2 (26 Aug - 03 Sep)	**1.1.4** (17 Sep - 25 Sep)	**Diff**
DAV	84.11%	74.66%	-11.24%
Earning / DAU	???	???	0.00%
Impression / DAU	56.41	49.2	-12.78%
Match Rate	95.56%	52.21%	-45%

Align with do not consent rate

Implementation issue