**Нативный CMP (Провайдер управления согласием)**

**Требования**

Издателям должна быть предоставлена библиотека, позволяющая интегрировать CMP в нативные приложения. Усилия со стороны издателей для интеграции должны быть сведены к минимуму.

CMP является Javascript-приложением и выполняется в Webview. При определенных условиях, Webview отображается как модальное всплывающее окно в приложении, с которым может взаимодействовать пользователь, и которое закрывается снова при нажатии кнопки.

Всплывающее окно управляется только CMP, приложение не может осуществлять управление. CMP управляет всплывающим окном через Javascript-интерфейс, который встраивается с помощью Webview. CMP может выполнять следующие функции через интерфейс Javascript:

- getConsent: Запрашивать согласие на обработку данных, находящееся в хранилище (SharedPreferences)

- setConsent: Сохранять/обновлять согласие в системе хранения данных

- showUi: Показывать всплывающие окна

- hideUi: Закрывать всплывающие окна

Библиотека должна работать на всех устройствах от уровня API 24.

В комплект поставки входит прототип, состоящий из демо-приложения и будущей библиотеки. Прототип был создан с помощью Android Studio (последняя версия) и демонстрирует основные принципы работы. Затем прототип должен быть адаптирован под все требования.

**Структура**

Основные компоненты:

- Javascript-интерфейс (класс: JsProxy): Все функции, доступные CMP для связи с приложением. Соответствующие методы описаны в коде.

- OpenCmp: Этот класс берет на себя инициализацию библиотеки и служит одновременно обработчиком для JsProxy.

- OpenCmpContext: Объект конфигурации, с помощью которого издатель запускает библиотеку

Исключения:

В некоторых случаях могут возникать технические ошибки, затрагивающие функциональность.

Они не должны мешать использованию приложения, поэтому необходимо отслеживать исключения в библиотеке. Приложение должно реагировать на них должным образом.

**Установка**

Чтобы использовать библиотеку, издатель должен интегрировать ее в свое приложение как зависимую часть, а затем настроить ее. Для использования библиотеки требуются следующие данные:

- Домен

- Ключ активации (опционально)

Дополнительно издатель может настроить экземпляр кнопки на повторное открытие CMP, если он был закрыт.

Техническая конфигурация должна быть максимально простой, чтобы всплывающее окно могло отображаться в любой Activity в любое время. В демоверсию включены ApplicationContext и Activity, если есть возможность более простой реализации, то, будет ещё лучше.

В идеале издателю нужно сделать всего один запрос, как в демоверсии:

OpenCmp.setup(context);

**Логика**

**Запуск приложения:**

Создается невидимый Webview, который загружает CMP. CMP получает настройки из SharedPreferences через JS-интерфейс и затем решает, нужно ли отображать всплывающее окно.

Webview постоянно активен во время работы приложения, так что всплывающее окно может отображаться без задержки в любое время, а также не прерываются запросы отслеживания, встроенные в CMP.

**Отображается всплывающее окно:**

****

Требования:

* Взаимодействие пользователя с приложением вне всплывающего окна должно быть исключено.
* Кроме того, полупрозрачный слой должен быть размещен над приложением, чтобы пользователь сконцентрировался на всплывающем окне. Здесь было бы идеально, если бы Webview использовался как слой, если это возможно. Просто используйте его на весь экран, слой уже встроен в CMP. Если не хватает только прозрачности, можно ли сделать прозрачным фон Webview, чтобы через него было видно приложение?
* Пользователь может свободно взаимодействовать с Webview.

Когда пользователь нажимает на одну из кнопок сохранения во всплывающем окне, CMP через JsProxy.setConsent() отправляет согласие на обработку данных приложению, которое сохраняет его в SharedPreferences.

После этого CMP скрывает всплывающее окно с помощью JsProxy.hideUi().

**Всплывающее окно скрывается:**

В этом случае Webview должен оставаться невидимым. Для повторного отображения всплывающего окна, чтобы внести изменения и сохранить их, пользователь может использовать опциональную кнопку. В демоверсии это кнопка в правом нижнем углу.

В демоверсии также доступна кнопка в нижнем левом углу, с помощью которой можно снова удалить настройки, чтобы при следующем запуске приложения всплывающее окно автоматически отображалось на экране.