Automotive Push Notifications With UnifiedPush

BMW
11th September 2023
Automotive Push Notifications

Goal
Agree on an industry standard service/protocol for 3rd party push notifications in the automotive space to enable core user experiences like

- Receiving VoIP calls (e.g., Zoom, Webex)
- Receiving messages

A possible Solution: UnifiedPush

- COVESA members only have to agree on the specifications
  - Each OEM has full freedom to either use an existing push server/distributor combination, develop their own or collaborate within COVESA
  - Compatibility is ensured by a testing suite
- UnifiedPush is Apache-2.0 licensed, same as AOSP
- UnifiedPush is compatible with web push
  - 3rd party app backends might already support it
UnifiedPush

- **Application server**: This is the server that hosts the application – the 3rd party backend.

- **Automotive application**: The (3rd party) application receiving the push notifications. The application gets started by the distributor on incoming push notifications if it is not running.

- **Push server**: This is the server that listens for incoming push messages and forwards them to the connected Push Distributor running in the car.

- **Distributor**: This is the application that forwards push messages to the registered end user application. It is the application which is connected to the Push Server and must be running at all times.

- **UP API & UP Protocol**: The only thing COVES needs to standardize.

Source: [https://unifiedpush.org/](https://unifiedpush.org/)
POC 1
Existing solution for OEM part / simulate 3rd party application

1. Use nfty.sh as the push server and distributor
   - Easily installable on the head unit as an .apk
   - Provides a free push server we can use for the POC

2. Use the UnifiedPush example app to simulate a 3rd party application receiving push notifications
   - Easily installable on the head unit as an .apk

3. Make an HTTP post request to the URL displayed in the UnifiedPush example app simulating the 3rd party backend

→ Get to know UnifiedPush

Time requirements: <1 hour

---

1 https://f-droid.org/de/packages/io.heckel.ntfy/
2 https://f-droid.org/packages/org.unifiedpush.example/
POC 1 Walkthrough
POC 2
Existing solution for OEM part / 3rd party application POC

1. Use ntfy.sh as the push server and distributor
   - Easily installable on the head unit as an .apk¹
   - Provides a free push server we can use for the POC

2. Install the 3rd party application POC that implements UnifiedPush

→ Evaluate the end-user experience

Time requirements: <1 hour once the 3rd party application POC is available

¹ https://f-droid.org/de/packages/io.heckel.ntfy/
3rd Party application POC
Add UnifiedPush support to your app

1. Evaluate whether your backend already supports web push
   - Yes: probably works out of the box 🎉
   - No: add support for UnifiedPush/web push

2. Integrate the UnifiedPush connector library into your app or implement the UnifiedPush protocol yourself

3. Easily simulate the OEM part using any Android device with an installed UnifiedPush distributor (e.g., ntfy.sh)
   - The ntfy.sh backend is open-source and might be a valuable reference implementation
   - The UnifiedPush example app is open-source and a reference implementation for the UnifiedPush connector library

**Time requirements:** 1-7 days
Thank you!