UnifiedPush

Push notifications. Decentralized and Open Source

FOSS on Mobile • FOSDEM 2026

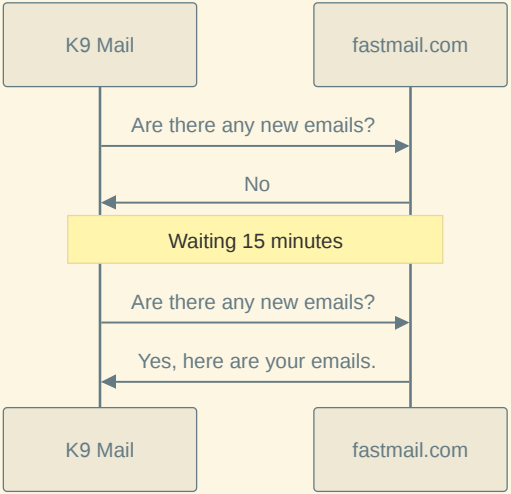
Simon Gougeon, Daniel Gultsch

unifiedpush.org codeberg.org/UnifiedPush @unifiedpush@fosstodon.org

Question

How do notifications actually get to my phone?

Periodic Polling

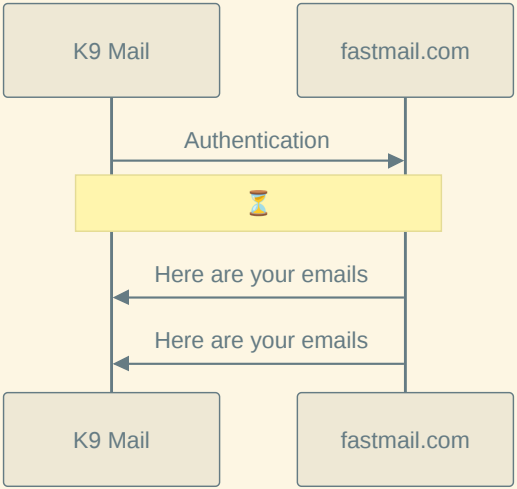


Periodic Polling: Pros & Cons

- Easy to implement

- Expensive (TLS, Login, …)
- High latency
- Majority of connections are unnecessary

Constant connection (or long polling)

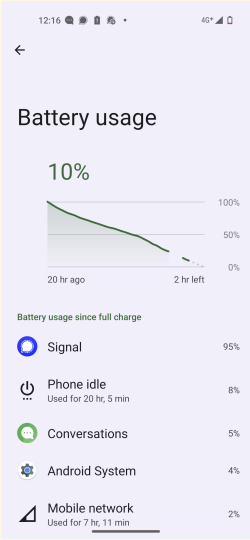


Constant connection: Theory vs Practice

- A network connection that does nothing uses no power.
- Management
  - Periodic pings
  - Re-establishing connection after a network change or disconnection
  - Not synchronized across apps

    Transmit as rarely as possible, and when you do, send as much as possible at once.

Constant connection: Theory vs Practice



Problem

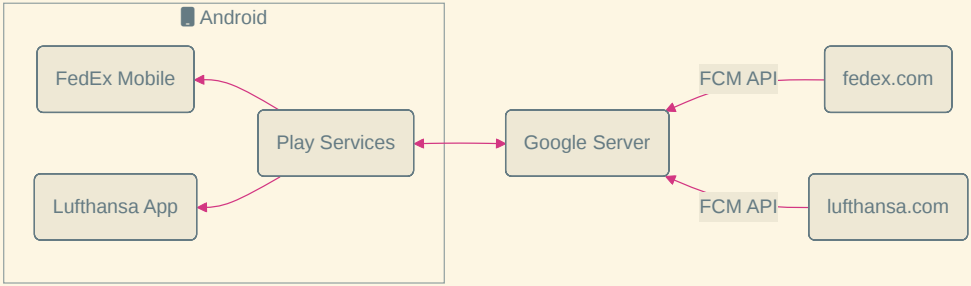
Not very simple in practice

Disclaimer: Battery usage attribution is hard

Question

Can’t we combine that in one app?

Firebase Cloud Messaging



FCM: Mode of operation

- Play Services maintain a persistent connection to Google servers

FCM: Mode of operation

- Play Services maintain a persistent connection to Google servers
- Simple constant connection / push protocol. No magic involved
  - Well managed (ping, re-connect)
  - Relatively efficient

FCM: Mode of operation

- Play Services maintain a persistent connection to Google servers
- Simple constant connection / push protocol. No magic involved
  - Well managed (ping, re-connect)
  - Relatively efficient
- Service provides a ’token’ on push for correlation with app

FCM: Mode of operation

- Play Services maintain a persistent connection to Google servers
- Simple constant connection / push protocol. No magic involved
  - Well managed (ping, re-connect)
  - Relatively efficient
- Service provides a ’token’ on push for correlation with app
- By default, few things work without ’Google Push’.
  - Apps need explicit user consent to maintain a persistent connection to the server.
  - Additional restrictions for apps distributed on Google Play

FCM: Criticism

- *Google just wants total control*
  Implementing push notifications yourself isn't easy for the average developer.
- Google doesn't necessarily see the content. Just: Which services, which app, how often.
- Google can see which Google Account the messages are sent to.

FCM: Much more criticism

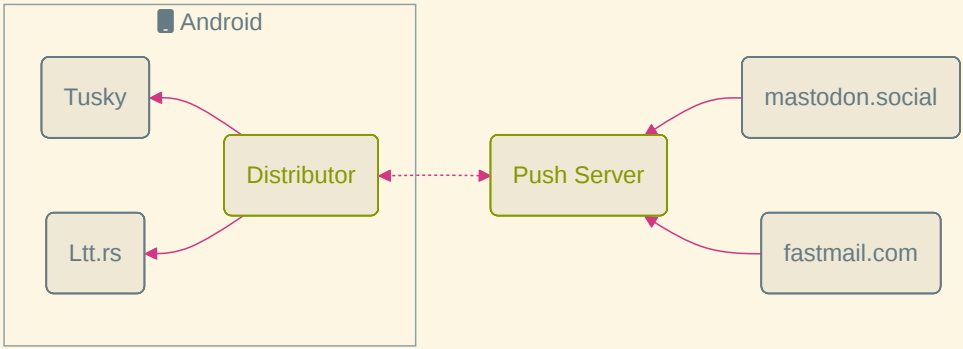
- Play Services installed with system privileges
- Play Services provide unwanted *features* like Advertising ID.
- Tracking (Resulting in unnecessary traffic and battery drain)
- Google Account often tied to real world identity
- Law enforcement regularly requests push token from services (tied to Google Account)

Using microG as a replacement can alleviate some of those problems.

Question

Can’t that be open source and decentralized?

UnifiedPush



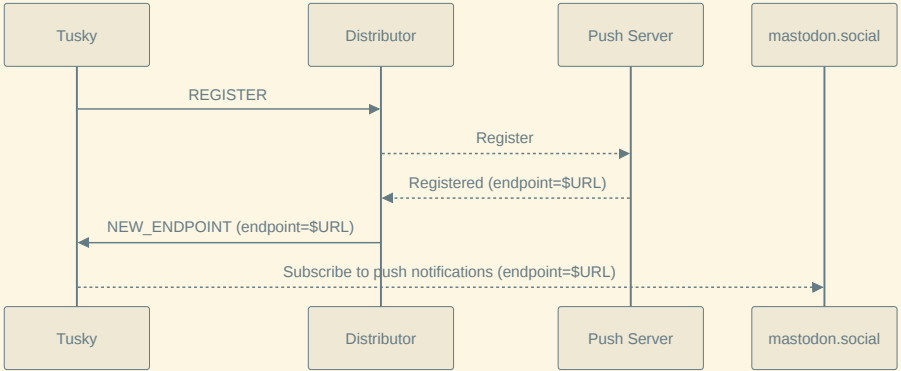
UP: Mode of operation

- App ↔ Distributor
  - Android IPC via *Broadcast Intents*
- Service ↔ Push Server
  - WebPush (RFC 8030, RFC 8292, RFC 8291)
  - Web standard supported by many services (Mastodon, JMAP, IRC, …)
  - Most likely candidate for future support (IMAP, …)
- **Not standardized:** Distributor ↔ Push Server

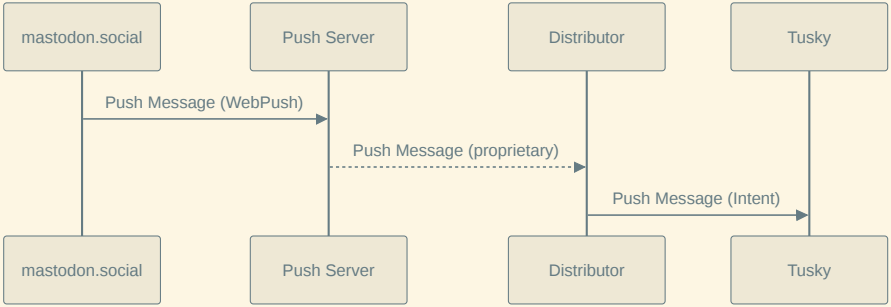
UP: Distributor ↔ Push Server

- Distributor establishes connection to *Push Server*
  - Regular Pings (NAT)
  - Re-establishing connection after a network change
- Competition between different protocols
  - WebSocket
  - Server-sent events (EventSource)
  - XMPP

Register with UnifiedPush



Push with UnifiedPush



Distributors

- Quick and easy
  - Sunup
- For self-hosting
  - ntfy
  - NextPush
- Jabber/XMPP Users
  - Conversations

Apps with UnifiedPush support

- Tusky
- Fedilab
- DAVx^5
- Ltt.rs
- Fennec
- Molly (Signal client)
- …

unifiedpush.org/users/apps/

UP: Coming to your app soon?

- No complicated, permanent connection
- Very easy to implement
  - *connector* library for Android IPC
  - WebPush libraries for server $\rightarrow$ push server

UP: Linux on Mobile

- D-Bus API
- Not many implementations. Knowledge is transferable.

Demo

Questions?

unifiedpush.org · #unifiedpush:matrix.org