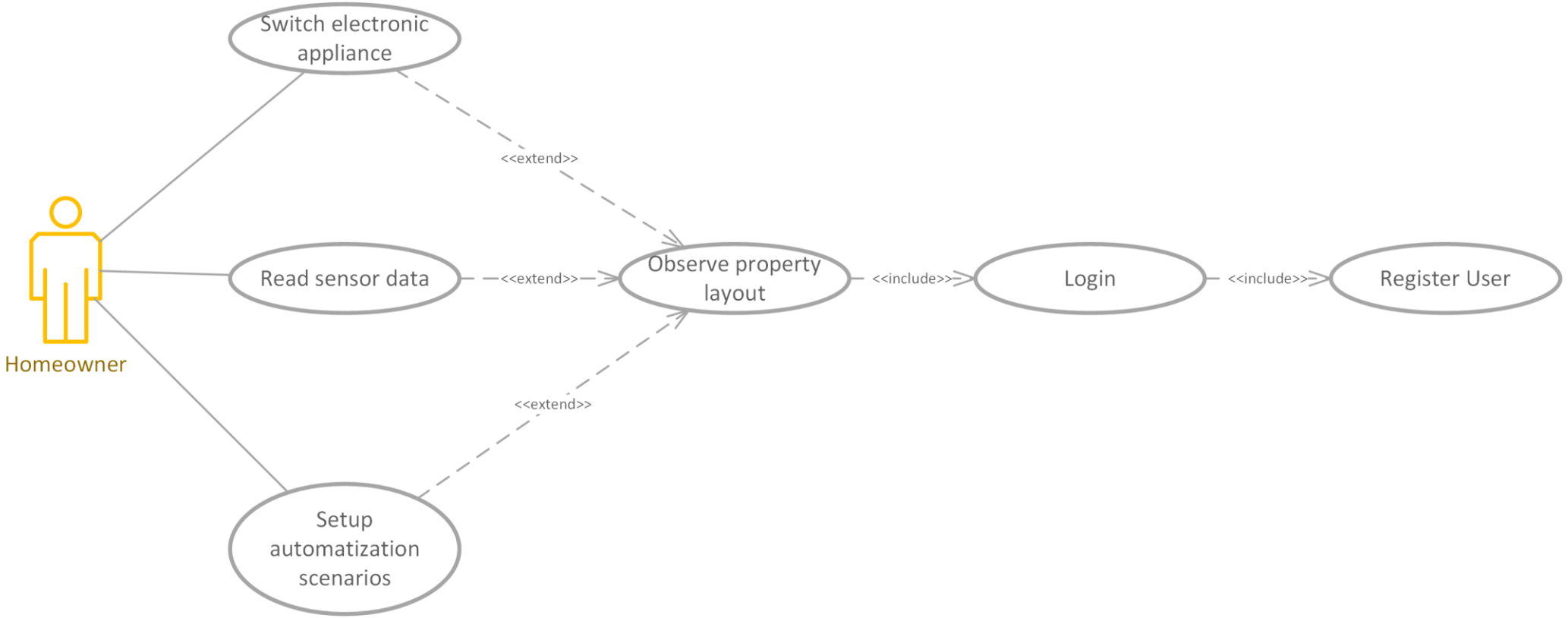
Adaptation Concepts

Anastasia Lykhtar, Dmitriy Monakhov

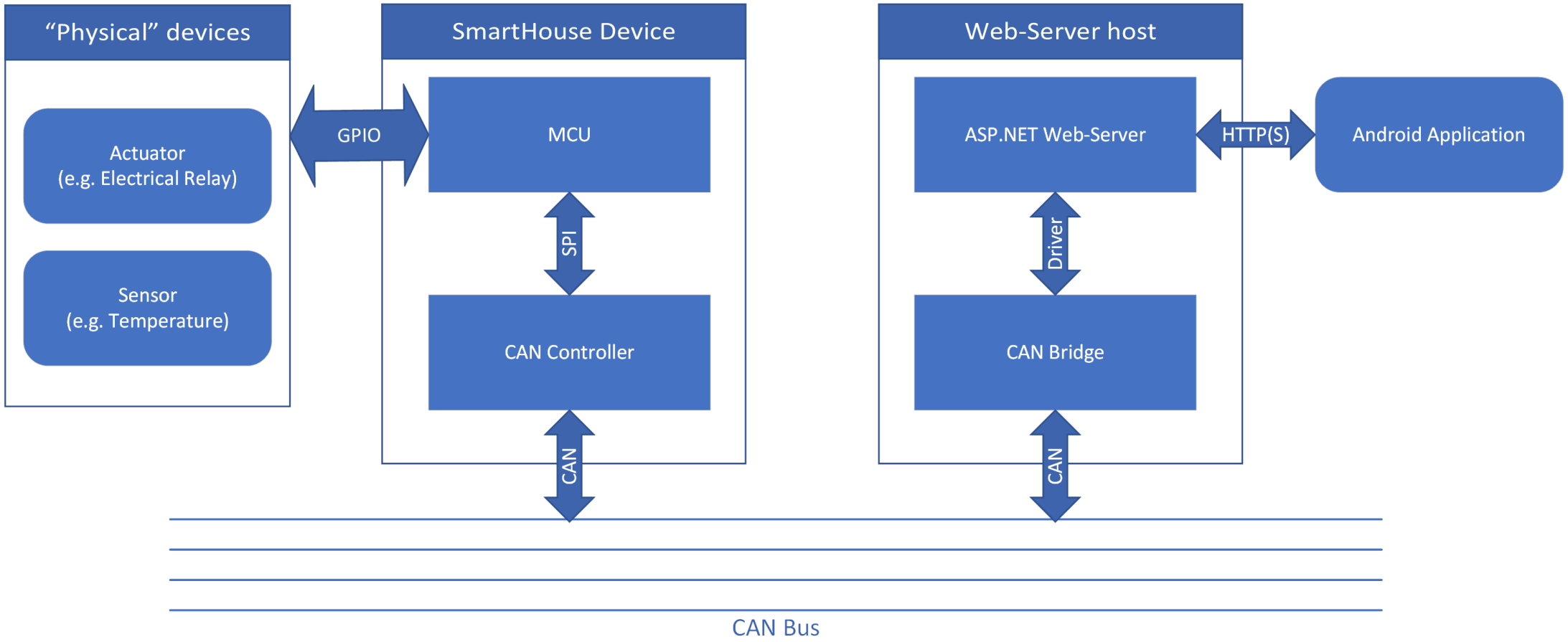
Key functions

- Light control
- Reading Appliance state
- Remote control of appliance
- Automation scenarios
- Authorization. Separation of roles
- Reading sensor data

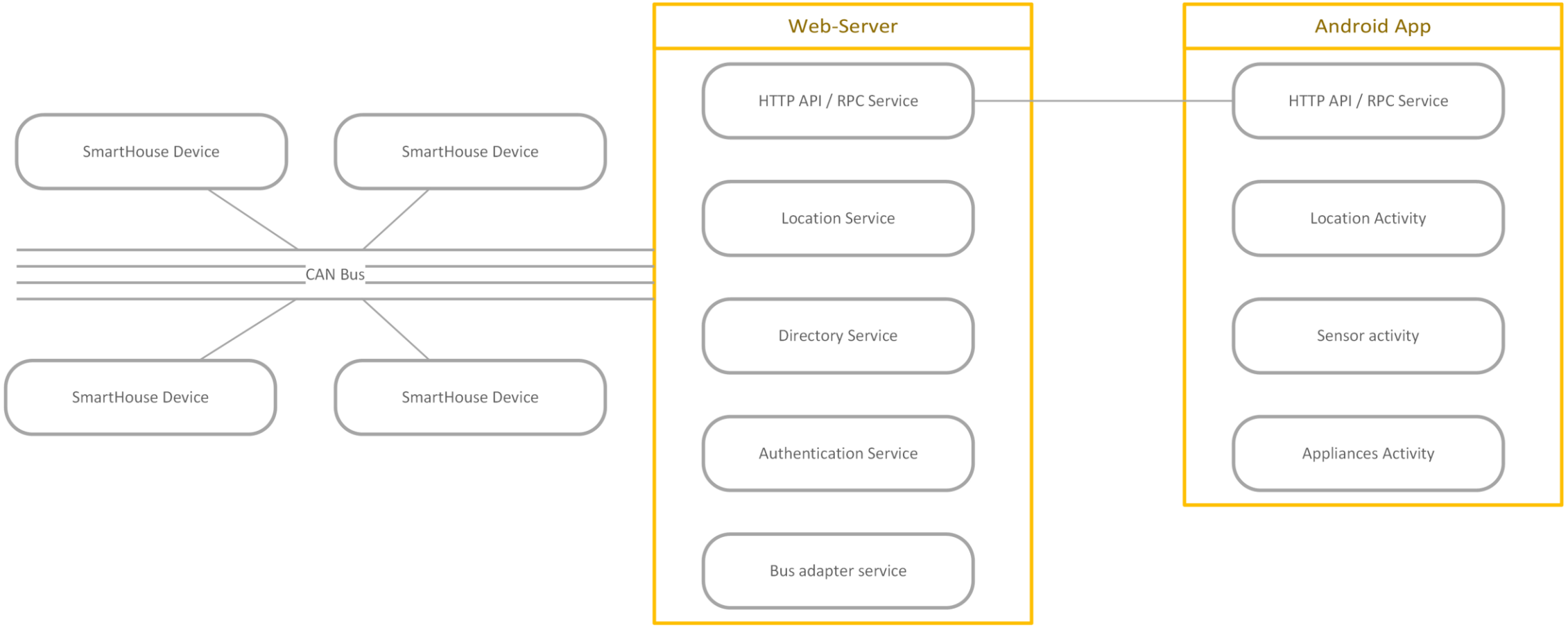
Use Cases


Switch electronic appliance
<<extend>>
Homeowner
Read sensor data
<<extend>>
Observe property layout
<<include>>
Login
<<include>>
Register User
<<extend>>
Setup automatization scenarios

Hardware Structure diagram



Software Component Structure diagram



Adaptation – Wi-Fi location tracking

- Get network state
- Check if Wi-Fi SSID is in location tracking Data Base
- Measure signal strength and determine the current room
- Adaptation: select active room and update screen info depending on location

Adaptation – energy saving

- Get battery state
- If battery has low charge, then disable server polling for SmartHouse states
- User can still update SmartHouse states manually and push updates manually

Problem – user leaves the apartment

- We can detect that user is not at home via checking Wi-Fi state
- If it is the case, we should switch to WAN connection mode or to offline mode
- If we are using WAN mode, the app cannot use adaptation mechanism of location monitoring, but still provided the main features of SmartHouse
- If we are using offline mode, the app can only show last known states of SmartHouse equipment and queue user commands for future pushing to the server
- When user enters the apartment again, we switch to LAN mode

Problem – battery low

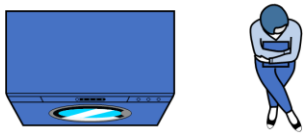
- By default, the mobile app periodically polls the server for updates
- This requires a permanent connection, which uses lots of battery energy
- If the battery is low, we should switch polling off and make the user update state information manually
- We also can use Push-notifications, but this technology depends on Google services and thus is unavailable on some devices

UX Storyline: User leaves the apartment

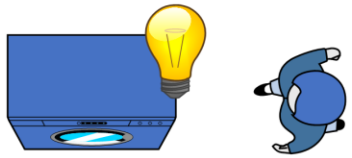
Storyboard Persona: Student Scenario: user leaves the apartment

User wants to go rest while the washing machine is working

User wants to return exactly when the machine stops



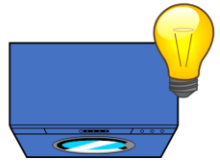
SmartHouse notifies user about state change



Problematic situation: user leaves the apartment
The application cannot use trusted LAN connection to fetch updates

SmartHouse changes network mode to WAN
Enables untrusted connection features, e.g. authentication and encryption

SmartHouse notifies user about state change



User returns home, network mode is LAN again

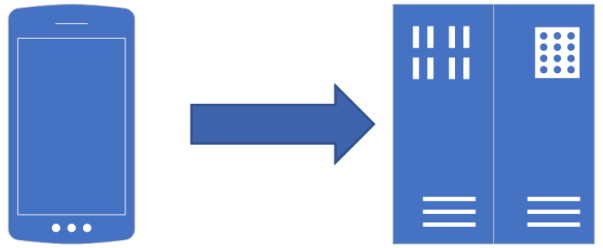
UX Storyline: Battery low

Storyboard Persona: Student Scenario: smartphone battery is low

User actively uses the smartphone for the whole day

Problematic situation: Now the battery is low
The app is draining battery power to fetch updates regularly

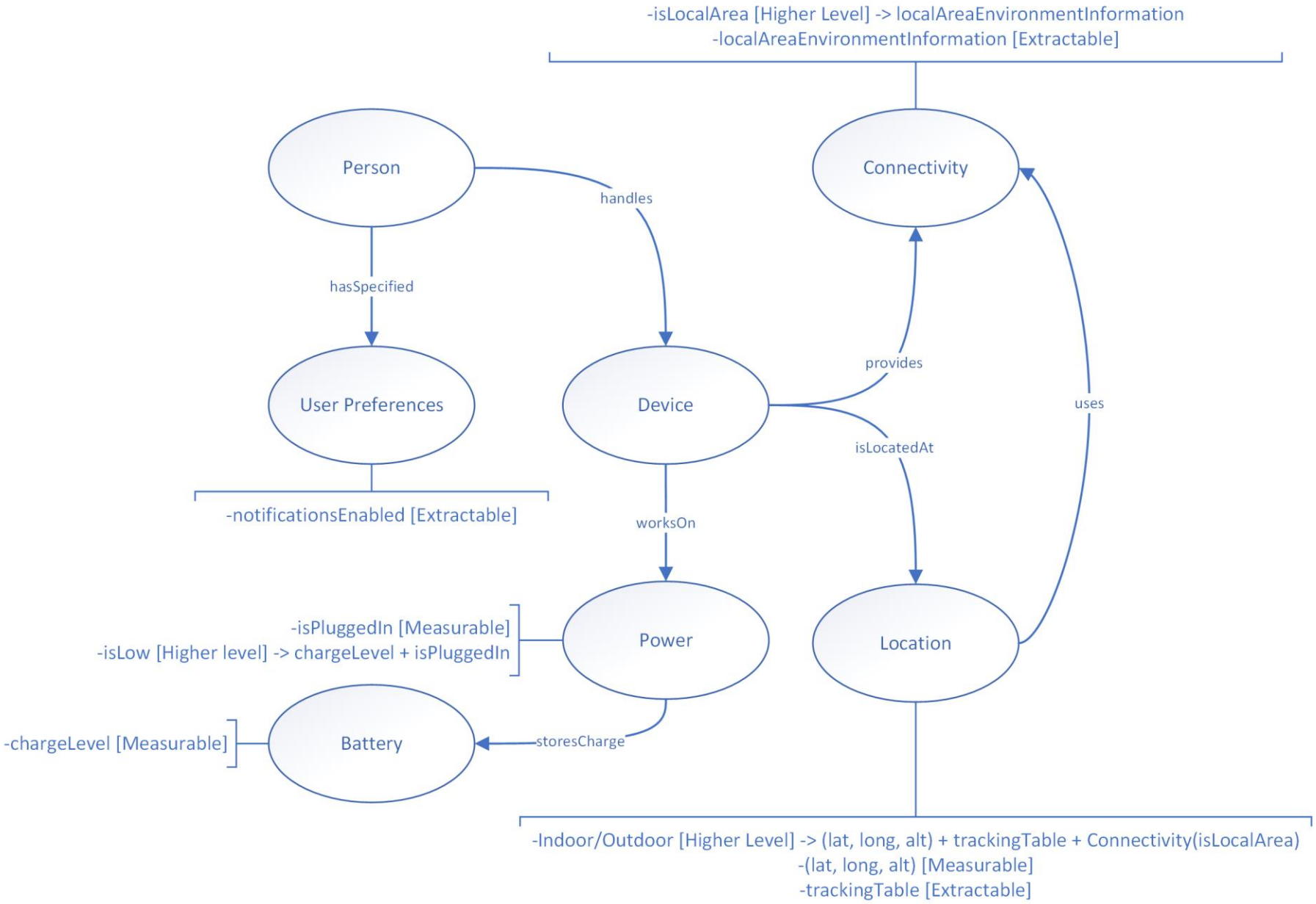
SmartHouse changes interaction scheme to push-only



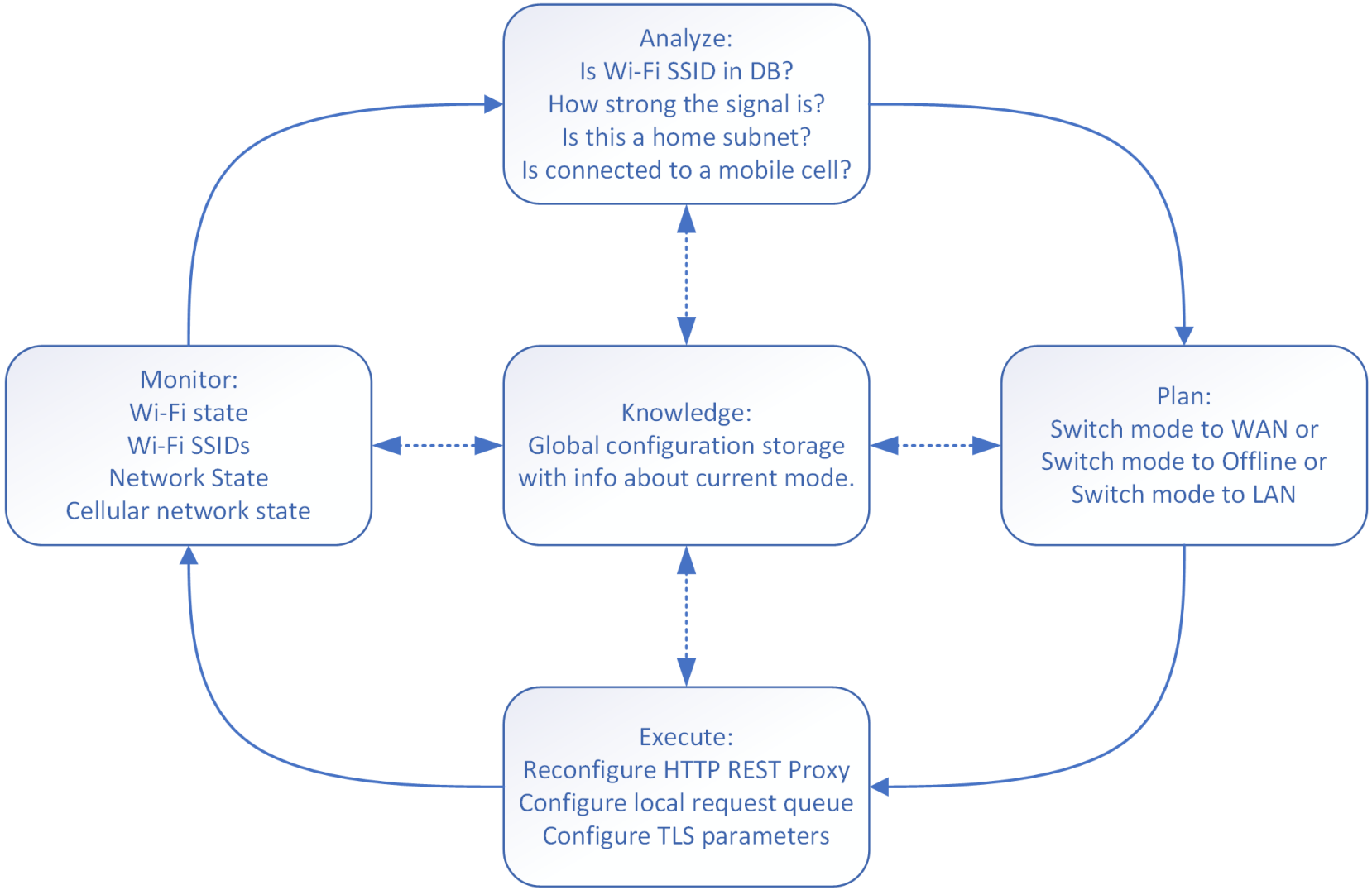
Battery life is extended

User plugs the phone in and SmartHouse is pull-push again

Context Model



MAPE-K for Connectivity and Offline Challenge



MAPE-K for Energy Challenge