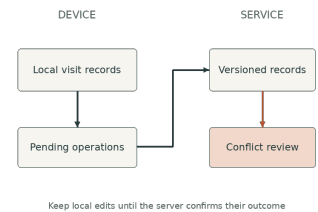


## APPLICATION ENGINEERING

# Offline service notes

Finish a service visit without a connection and resolve changes when it returns.



## PROPOSED ARCHITECTURE

Offline workflow brief; record ownership to define

## THE PROBLEM

A technician needs asset details and a checklist in a room with poor reception. Notes should remain available after a normal app restart. Reconnecting must not silently overwrite another technician's edit or submit the same visit twice.

## SCOPE

- Build an installable web app with IndexedDB storage and an explicit outbox.
- Sync through a REST API using operation IDs and record versions; surface conflicting edits.
- Cache assigned visits and asset details; keep the first release to text and checklist data.

## PLANNED DELIVERABLES

- Offline web client, API, and synchronization protocol
- Two-client conflict tests and interrupted-sync scenarios
- Local deployment, sample visits, and storage-recovery instructions

## CONSTRAINTS

### Users

Assigned technicians and a dispatcher

### Conflicts

Review version conflicts; no silent last-write-wins

### Storage

Show pending work and provide export; browser storage may be evicted

### Access

Server authorization applies to every synchronized record

## VERIFICATION PLAN

### Disconnection

Save offline, restart normally, and reconnect; the note appears once.

### Concurrent edits

Edit one record on two devices; neither change disappears silently.

### Revoked access

Reject an unauthorized sync and preserve an exportable local draft.

## NEXT STEP

Define visit states, record ownership, and the conflict-resolution screen.

Offline drafts depend on retained device storage. Export and sync status need to stay visible; photo uploads are outside the first release.