**Sprint 8.5 – Precision Watt *stubs* + short-session guard (🧩)  
Kontekstblokk  
• Fra S5: Indoor/outdoor + GPS/Wind stabil, CLI-felter (wind\_rel, v\_rel, calibrated, status) på plass.  
• Fra S6: Rapport/observabilitet levert; strukturert JSON-logging og metrikker fungerer.  
• Fra S7: Dynamisk DoD oppdatert (schema\_version, ≥30 samples, edge-case-dekning).  
• Fra S8: FE scaffold ferdig, ENV-switch mock↔live, schema-version-guard og HR-only fallback i UI.  
• Avhengigheter: Ingen nye tredjepartstjenester; bygger kun videre på eksisterende FE-kode (typer/mocks/visning).  
• Neste steg (Frontlog): S9 – Økt-kort & nøkkelmetrikker.**

**Sprintbeskrivelse  
Oppgaver:**

1. **Utvid SessionReport (typer): legg til *valgfrie* felter precision\_watt, precision\_watt\_ci, sources, cda, crr, reason.**
2. **Oppdater mockSession (mocks): små dummy-serier for precision\_watt og precision\_watt\_ci.**
3. **Dev-sanity (SessionView): vis antall PW/CI samples *kun i DEV*.**
4. **Short-session guard: egen guard-funksjon + kontrollert info-kort ved <30 samples.  
   DoD (må alle bestås):  
   • FE bygger grønt med nye felter (ingen UI-regress).  
   • Dev-sanity viser counts i DEV når PW-data finnes.  
   • Kort-økt (<30 samples) gir kontrollert beskjed, ingen crash.  
   Estimert tid: 2–2.25h**

**Arbeidsmetode (for Copilot) – “halvfast default” (justert)**

1. **PLAN (5–7 trinn)  
   • Hvilke filer, hvorfor, forventet output, risikopunkter. Vent på “OK PLAN”.**
2. **Kjør trinnvis  
   • Hvert trinn: diff → kjør npm run type-check/npm run build → vis kort output. Vent på “OK <trinn>”.**
3. **Stop-regler (justert for FE-mini-sprint)  
   • Stopp hvis: forslag endrer core/CLI/backend; alt arbeid skal være i frontend/\*\* + mocks.  
   • Tillatt: ≤6 filer totalt (typer, mocks, én ny lib-fil, én rute-endring).  
   • Ingen nye libs/rammeverk.**
4. **Schema-bump sjekkliste (ny)  
   • Ingen bump nå; behold eksisterende schema\_version-guard.  
   • Hvis bump blir nødvendig: utsett til S9 og følg schema-dokumentasjonsløp.**
5. **Debug-partner  
   • Bruk ChatGPT ved TypeScript-typer/paths/ENV-issues.**
6. **Ferdigkriterium  
   • DoD oppfylt (bygg grønt, DEV-sanity synlig, short-session beskjed).  
   Svarformat (Copilot):  
   • PLAN → vent på “OK PLAN”.  
   • TRINN 1 … → vent på “OK 1”, “OK 2”, osv.**

**Sprint-delta (mini-kontekst)  
• 8.5 innfører kun datakontrakt-stubs + guard; ingen graf/kalibrerings-UI. Reduserer risiko og tid i S9–S12.**

**Tekniske rammer (kontrakt)  
• Input: eksisterende analyze\_session()-JSON; nye felter er *valgfrie*.  
• Output: UI som fortsatt rendrer mock/live uten crash; DEV ser PW/CI counts; <30 samples viser kontrollert info.  
• ENV: uendret (mock↔live).**

**Endringsgrenser  
• Tillatt: frontend/src/types/session.ts, frontend/src/mocks/mockSession.ts, frontend/src/lib/guards.ts (ny), frontend/src/routes/SessionView.tsx.  
• Ikke rør: core/src/\*\*, CLI/backend, docs (utenom eksisterende).**

**Tester / kommandoer  
npm run type-check  
npm run build  
npm run dev  
npx serve -s dist**

**Verifikasjonsplan (DoD → test)**

1. **Types: npm run type-check passerer etter utvidelser.**
2. **Dev-sanity: i DEV vises “PW samples: N, CI: M” når mock har PW/CI.**
3. **Short-økt: sett mock til <30 samples → vis kontrollert info-kort.**
4. **Build: npm run build fullfører uten feil; prod-serve fungerer.**

**Forventet output (filer/endringer)  
• frontend/src/types/session.ts — legg til *valgfrie* felter:**

* **precision\_watt?: number[] | null**
* **precision\_watt\_ci?: [number, number][] | null**
* **sources?: string[] | null**
* **cda?: number | null**
* **crr?: number | null**
* **reason?: string | null  
  (Eksisterende felter urørt.)  
  • frontend/src/mocks/mockSession.ts — legg til små dummy-serier (f.eks. 10 punkter):**
* **precision\_watt: [210,212,209,211,213,212,210,209,211,212]**
* **precision\_watt\_ci: [[190,230],[192,234],[189,231],[191,233],[193,235],[192,234],[190,232],[189,231],[191,233],[192,234]]**
* **sources: ["powermeter","weather","profile"]**
* **cda: null, crr: null, reason: null  
  • frontend/src/lib/guards.ts — ny fil med:**
* **isShortSession(n:number, min=30): boolean**
* **guessSampleLength(d: Partial<SessionReport>): number (bruk lengden av watts/precision\_watt/ev. hr\_series hvis finnes)  
  • frontend/src/routes/SessionView.tsx — utvid:**
* **Importér guessSampleLength/isShortSession.**
* **Etter setData(d): beregn n = guessSampleLength(d) og short = isShortSession(n).**
* **Renderer kontrollert info-kort når short === true.**
* **Legg inn DEV-sanity under kortet: vis PW samples/CI counts *kun når* import.meta.env.DEV og data finnes.**

**Frontlog (kommende oppgaver)  
• S9 – Økt-kort & nøkkelmetrikker (📊): UI for NP/IF/VI/Pa:Hr/W/slag/CGS/PrecisionWatt-verdi; no-watt varsel; formatter-tester; kort-økt-håndtering videreføres.  
• S10 – Precision Watt UI (🎛️): Graf + usikkerhetsbånd; tooltip (datakilde, vær, profil); performance på 1 Hz/2**

**Gi meg så detaljerte detaljer trinn for trinn at jeg og chat gpt kan gjøre det sammen mens du styrer roamdmap og framdrift mens Chat og jeg står for kodeinhold og de faktiske leveransene i sprinten.**

Lag en Sluttrapport etter Følgende format

✅ Sprint: [ID – Navn]

Branch: [navn]

Commits: [hash – message]

Endrede filer: [paths]

Tester: [pytest + cargo test resultater]

Observasjoner: [korte punkter]

Status: [Ferdig / Delvis ferdig / Feil gjenstår]

**Sluttføring av Workflow (ChatGPT v2)**

1. **Sluttrapport:** lim inn Copilot-rapport i template.
2. **Delta-sammendrag:** skriv 2–3 linjer etter komprimeringsmal:
   * Hva ble gjort.
   * Teststatus.
   * Observasjon.
3. **Evaluering & Oppdatering:** oppdater DoD og Frontlog basert på nye funn/innsikter.
   * Klassifiser funn som A (DoD), B (Frontlog), C (Observasjon).
   * Marker ferdige sprinter, flytt avhengigheter.
   * Juster Sprintplan om nødvendig.
4. **Synk Masterplan:** oppdater statuslinje eller legg til ny rad.
5. **Oppdater Sprint Log:** kort notat med status og carry-over.
6. **Template neste sprint:** sjekk om “halvfast default” trenger justering og fyll ut template basert på neste sprint.