
⚡ GridAware – Smart EV Charging Dashboard

Product Requirements Document (PRD)

1. Product Vision

GridAware is a modern, intelligent dashboard that empowers EV owners in Germany to make cost-optimal charging decisions using real-time electricity prices. By integrating the Awattar market API, advanced EV configuration, and a clean, data-driven interface, GridAware delivers actionable, transparent insights — enabling users to maximize savings, efficiency, and confidence in their energy decisions.

2. Target Audience

- Private and fleet EV owners in Germany
 - Energy consultants and smart-home enthusiasts
 - Engineers, researchers, and students in energy informatics
 - Anyone interested in energy price transparency and smart charging
-

3. Core Features

A. Live Market Prices

- **On-demand fetch** of the latest hourly electricity prices from Awattar API
- Data is **never auto-fetched**; only updated at user's request
- **Responsive UI**: Loading spinner and clear status banner during/after fetch
- **Persistent data**: Last valid price data stored and shown until user fetches new data
- **Graceful failure**: If live fetch fails, user sees a warning and the most recent data, or a blank chart with error
- **Data visualization**:
 - Interactive bar chart of current and next hours
 - X-axis: local hour of day, Y-axis: price in €/kWh
 - Clear chart titles, color cues, and price axis

- **Full data transparency:** Display data source, last updated timestamp, and error messages if relevant

B. EV Charging Optimization

- **Expanded configuration options** for professional-level accuracy:
 - Vehicle preset (choose from popular models or manual entry)
 - Battery capacity (kWh)
 - Current state of charge (SoC, %)
 - Target SoC (%)
 - Max charging power (kW)
 - Charging efficiency (%)
 - Charging mode (Cheapest, Eco, Fast)
 - Preferred charging window (fine-grained: select start/end to the minute)
 - Charge deadline (by when charging must be complete, optional)
- **Inputs are persistent** until user changes or resets them
- **Smart recommendation engine:**
 - Calculates best charging start time based on config and latest prices
 - Simulates charging at all valid hours within selected window and deadline
 - Considers efficiency, vehicle limits, and price volatility
- **Clear results and insights:**
 - Summary card: optimal window, kWh needed, duration, estimated cost, finish time
 - Visual overlay on price chart to highlight recommended window
 - Table/summary: compare cost for other possible start times
 - Alert if no valid window is available
- **UI always tells user** if data/config is missing or incomplete

C. About & Help

- Explains:
 - **What the dashboard does**
 - **How it works:** data flow, recommendations, and session storage

- **APIs used:** Awattar, with link to API docs
 - **Tech stack:** Dash, Plotly, Pandas, Requests (with versions)
 - **Design philosophy:** Clarity, transparency, user control
 - **Planned roadmap:** Auto-scheduling, CO₂ intensity optimization, solar integration, notifications
 - Credits, authorship, and open-source license
-

4. User Workflow

1. **App loads:** User sees header and three tabs (Market Prices, EV Charging, About)
 2. **Market Prices tab:**
 - Chart is blank; status message invites user to fetch prices
 - User clicks **Fetch Latest Prices**
 - Spinner and banner indicate loading; chart and banner update with results or errors
 - If data fetched, chart and data stored for session
 3. **EV Charging tab:**
 - User enters/selects vehicle config, charging window, efficiency, etc.
 - Inputs update in-session and are retained
 - When prices and config both available, user clicks **Save & Analyze**
 - Dashboard displays:
 - Charging recommendation summary card
 - Visual cost chart and optimal window highlight
 - Breakdown table/card for start/end time, kWh, hours, cost
 - Notices for errors or config gaps
 4. **About tab:** User reads about features, design, and tech stack
-

5. UI & UX Principles

- **Modern, clean look:** Large header, rounded cards, subtle box shadows, logical grouping

- **Clear separation:** Input forms, actions, and results are visually distinct
 - **Professional input UX:**
 - Input rows/groups, sensible spacing, instant validation (where possible)
 - Dropdowns and time inputs for accuracy and ease
 - **Action feedback:** Loading spinner for long fetches, live status banners, all errors shown in plain language
 - **Responsive layout:** Works on desktop and tablet; mobile roadmap noted
-

6. Technical Architecture

- **Dash/Plotly:** UI and charts
 - **Requests:** API communication
 - **Pandas:** Data processing (future extensibility)
 - **Session storage:** Dash's dcc.Store for config and data persistence (per browser session)
 - **Local disk:** Fallback to last stored market data as JSON
 - **Callback-driven state:** All interactivity, recommendation logic, and UI updates managed via Dash callbacks
-

7. Data Management

- **Awattar API:**
 - GET <https://api.awattar.de/v1/marketdata>
 - Read marketprice (EUR/MWh) and start_timestamp (UNIX ms)
 - Convert to €/kWh and local hour for display
 - On fetch failure:
 - Show last stored data (with time of retrieval)
 - If none, keep chart blank and show error
- **EV Config:**
 - All form fields saved to session store on "Save & Analyze"
 - Config used for all further recommendations until changed

8. Error & Edge-Case Handling

- **No prices loaded:** Chart blank, status “Please fetch prices”
- **API error:** Banner shows reason, last known data remains if available
- **Incomplete EV config:** Results panel stays empty, notice prompts user for missing fields
- **No valid charging window:** Notice “No valid charging window found for your criteria”

9. Roadmap (Future Enhancements)

- CO₂-aware charging (use grid emission factors)
- User authentication and persistent profile
- Solar production integration (PVGIS API)
- Smart auto-scheduling, notification integration
- Multi-language support (German/English)
- Export recommendations as PDF/CSV
- Mobile-optimized layout

10. Tech Stack

- **Frontend/UI:** Dash (v2.16+), Plotly (v5.22+), CSS3
- **Backend:** Python 3.9+, Requests, Pandas
- **Data/APIs:** Awattar Germany (<https://www.awattar.de/services/api>)
- **DevOps:** Local run via python app.py, pip requirements for easy install
- **Open Source License:** MIT

11. Acceptance Criteria

- Market prices only fetched on user action, always visible with loading feedback and status
- EV charging config is rich, realistic, and easy to use
- Charging recommendations are clearly shown and updated in real-time

- All states (loading, error, incomplete) are handled and communicated to the user
 - App is visually modern, structured, and pleasant to use
-

12. Author / Credits

Developed by **Sheri** as a next-generation, open-source dashboard for energy transparency and EV charging intelligence.
