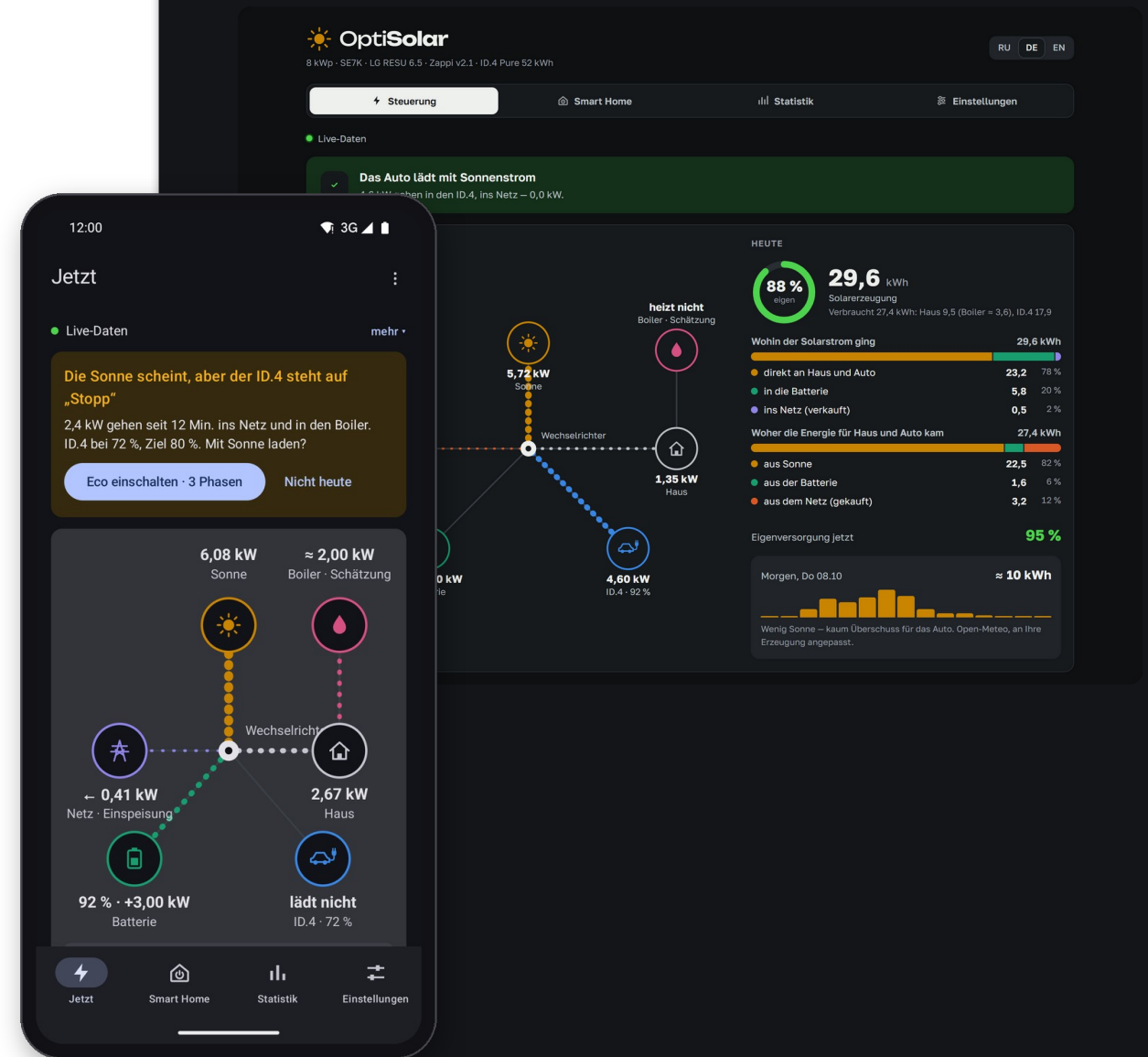


OptiSolar

Eine Plattform für Sonnenstrom, E-Auto und Smart Home

Vom Alltagsproblem zum produktiven System in 10
Tagen – konzipiert als Product Owner, entwickelt
mit Claude Code.

Roman Krol · Product Owner



Das Problem: fünf Apps für ein Zuhause

SolarEdge PV-Anlage 8 kWp & Batterie

myenergi Zappi Wallbox

Volkswagen ID.4 – Ladestand

Smart Life Licht, Steckdosen, Rollläden

Meross · Xiaomi weitere Geräte

- **Kein Gesamtbild:** Wohin fließt der Sonnenstrom gerade – ins Haus, die Batterie, das Auto oder ins Netz?
- **Verschenkter Sonnenstrom:** Überschuss geht für 8 ct/kWh ins Netz, statt für 36 ct/kWh das Auto zu laden – weil man vergisst, die Wallbox umzuschalten.
- **Ständiges App-Hopping:** fünf Logins, fünf Bedienlogiken, keine Verbindung zwischen den Geräten.
- **Familie & unterwegs:** keine sichere, gemeinsame Steuerung für alle im Haushalt.

Die Idee: eine Plattform, ein Blick

„Ich will nicht fünf Apps bedienen – ich will, dass mein Haus mit der Sonne lebt.“

01

Sehen

Live-Energiefluss, Prognose und Statistik für Sonne, Batterie, Netz, Auto und Boiler – in Echtzeit.

02

Entscheiden

Hinweise statt starrer Automatik: das System schlägt vor, der Mensch entscheidet mit einem Tap.

03

Steuern

Wallbox und Smart Home aus einer App – zuhause im WLAN oder sicher von unterwegs.

Use Cases aus dem Alltag

„Morgen früh brauche ich 90 %.“

Smart Boost in Prozent: „bis 90 % um 07:00“ statt Kilowattstunden rechnen.

„Ich warte auf die Sonne.“

Solar-Hinweis + Push: Sonne da, Auto auf „Stopp“ → „Jetzt mit Eco laden?“ – ein Tap.

„Ich brauche das Auto jetzt.“

Eco oder Fast per Fingerabdruck, Fortschritt live, Ladeende-Prognose für 80 / 100 %.

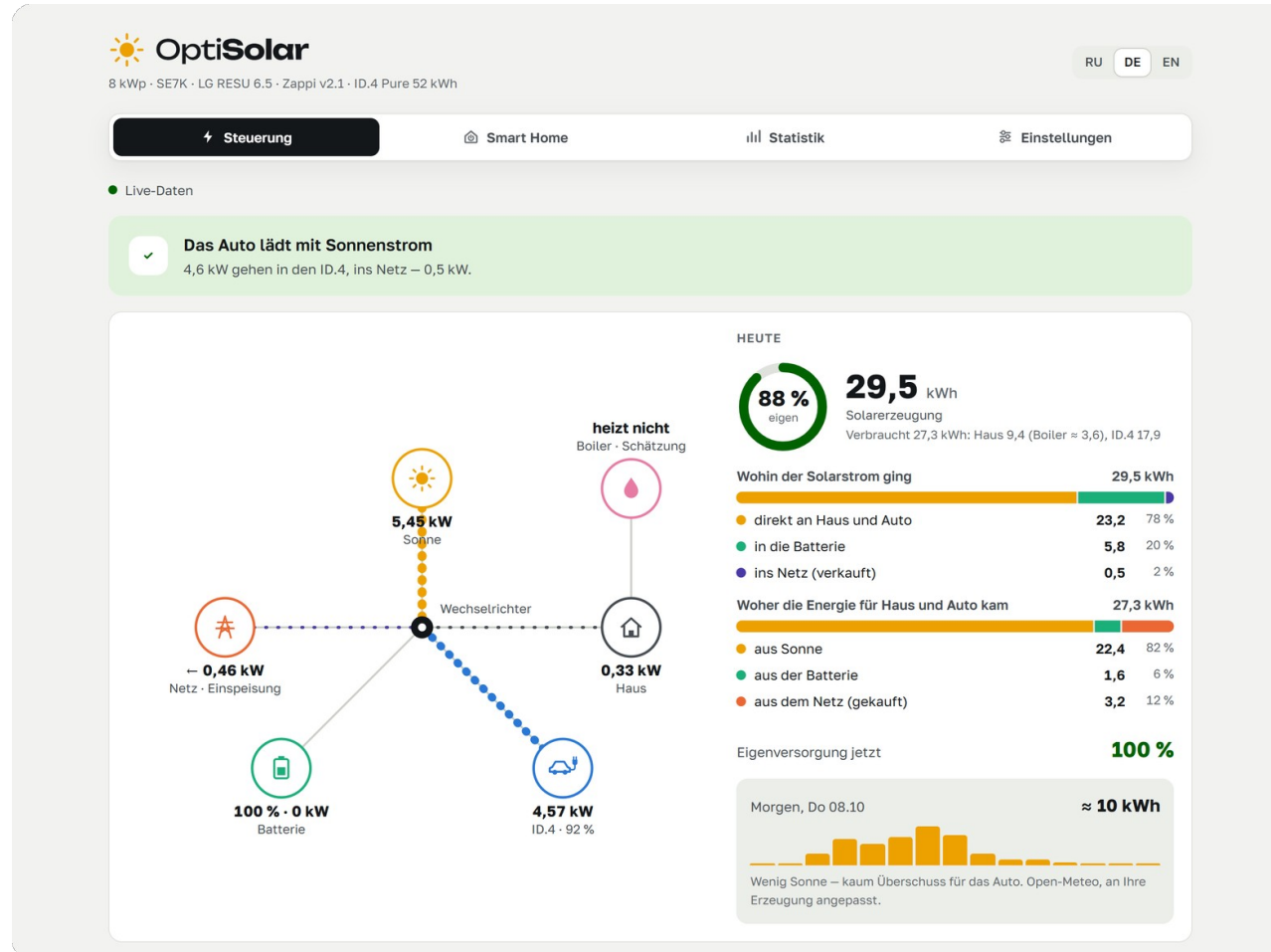
„Licht aus, Rollläden zu.“

Smart Home nach Räumen, Favoriten und Reihenfolge pro Person.

„Hat sich das gelohnt?“

Statistik: heute 18 kWh geladen – davon 98 % aus der eigenen Sonne.

Live: Energiefluss in Echtzeit



- **Ein Bild statt fünf Apps:** Sonne, Haus, Batterie, Netz, ID.4 und Boiler – Animation zeigt Richtung und Stärke.
- **Klartext statt Zahlen:** „Das Auto lädt mit Sonnenstrom – 4,1 kW gehen in den ID.4.“
- **Heute im Blick:** 88 % Eigenversorgung, Prognose für morgen (Open-Meteo, an die eigene Anlage angepasst).
- **Steuerung:** Zappi-Modus, Phasen, Boost und Ladeziel – mit Bestätigung und Live-Fortschritt.

Prognose: wann das Auto mit Sonne lädt



7 Tage

Vorschau

24 h

pro Tag, stündlich

≈ 10 kWh

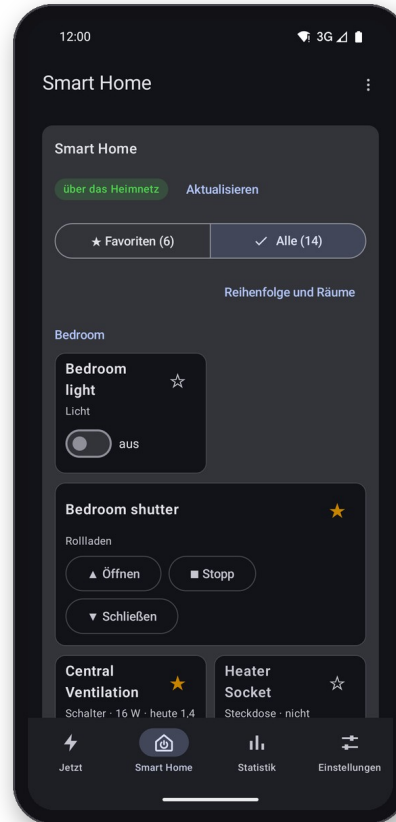
fürs Auto, z. B. 12–15 Uhr

- **7 Tage voraus:** Wetter, Höchst- und Tiefstwerte und erwarteter Solarertrag pro Tag.
- **Stunde für Stunde:** Sonne, Wolken, Regen (mm und Wahrscheinlichkeit), Temperatur und Erzeugung in kW.
- **Beste Ladezeit:** „Am besten laden 12:00–15:00 – ≈ 10 kWh aus der Sonne“; an trüben Tagen der Hinweis auf den sonnigsten Tag der Woche.
- **Daten:** Open-Meteo, an die eigene Anlage angepasst, stündlich aktualisiert – auf Web und Android.

Mobil: dieselbe Plattform in der Tasche



Solar-Hinweis: ein Tap, und das Auto lädt mit Sonne

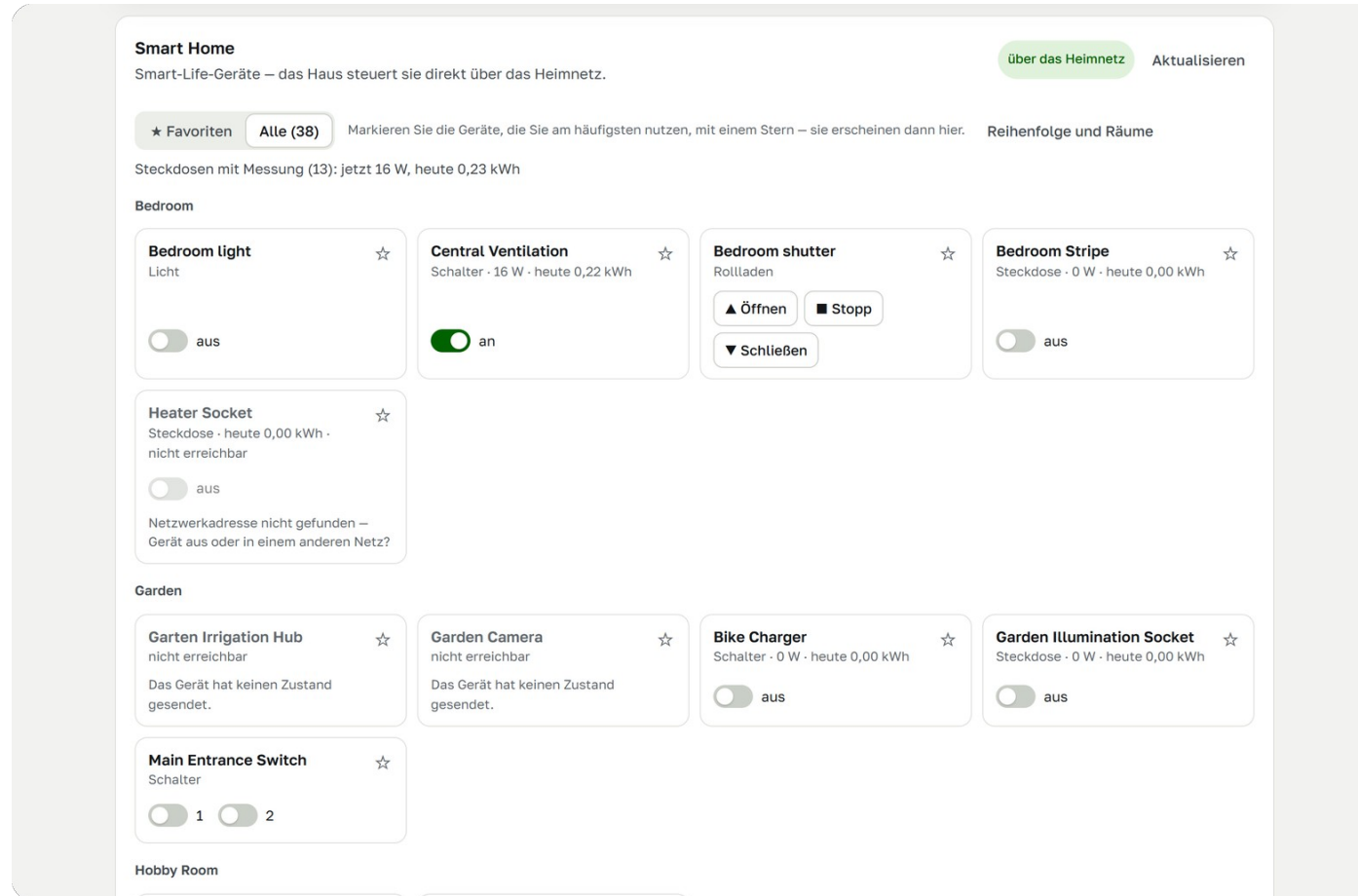


Smart Home nach Räumen, zwei Kacheln pro Reihe



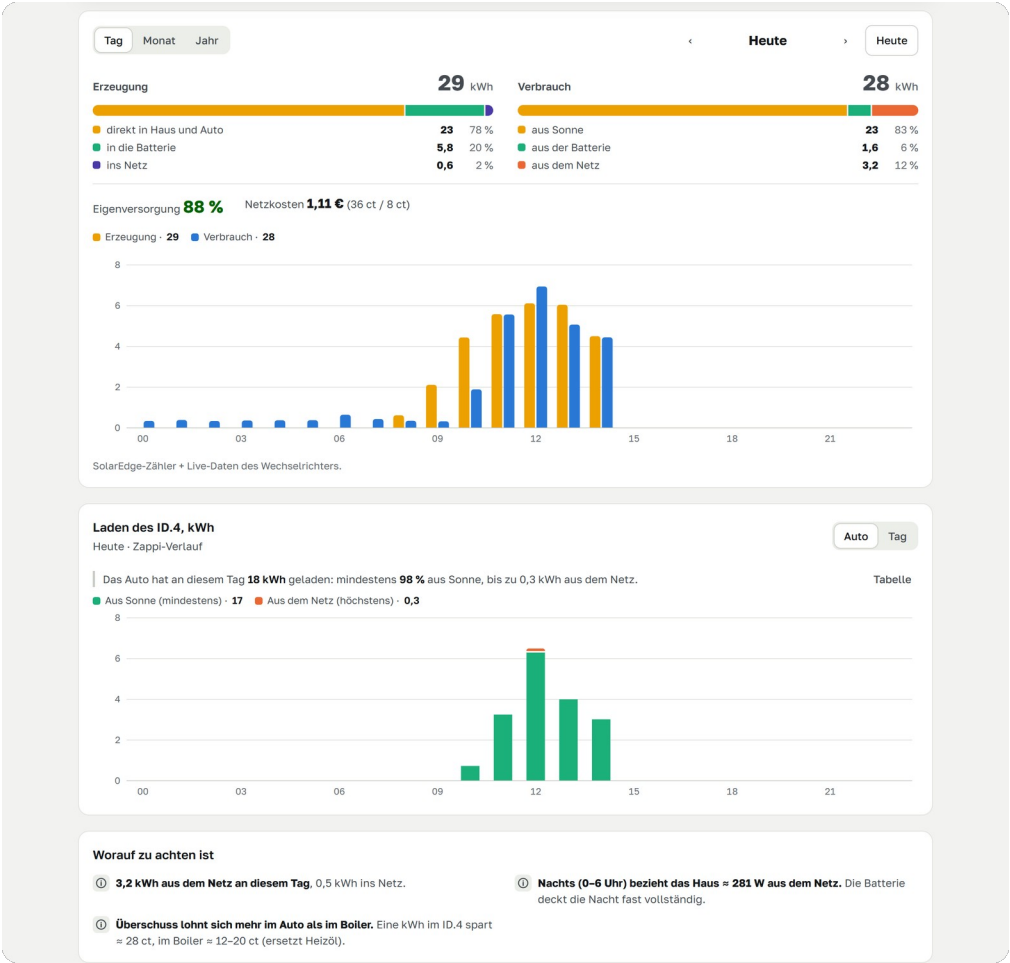
Statistik: Laden des ID.4 nach Herkunft

Smart Home – lokal, nach Räumen, pro Person



- **Direkt im Heimnetz:** 38 Geräte über das lokale Tuya-Protokoll – schnell, ohne Hersteller-Cloud.
- **Räume aus Smart Life, Favoriten und Reihenfolge pro Person (Drag & Drop).**
- **Rollläden und Schalter** mit mehreren Kanälen; Steckdosen mit Verbrauchsmessung.
- **Bewusste UX-Entscheidung:** Licht ohne Fingerabdruck, Wallbox nur mit.

Ergebnis: Sonnenstrom im Auto statt im Netz



88 %

Eigenversorgung
heute

18 kWh

in den ID.4 geladen

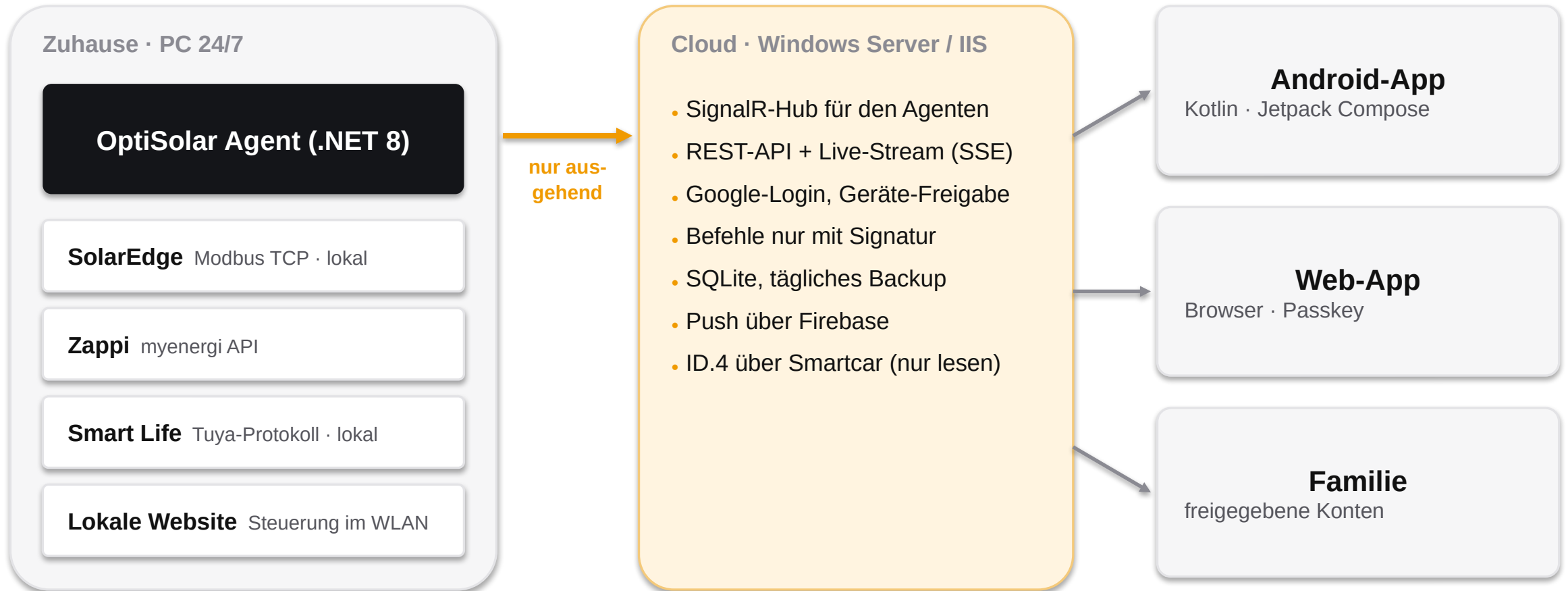
98 %

davon aus der Sonne

1 App

statt fünf

Architektur: das Haus öffnet keinen einzigen Port



Der Agent baut die Verbindung selbst auf (WebSocket) – zuhause bleibt die Firewall zu. Das Haus hat das letzte Wort: Limits, Trockenlauf, Freigaben.

Security by Design

Anmeldung

Google-Login nur für freigegebene Konten; jedes neue Gerät muss vom Admin bestätigt werden.

Befehle

Wallbox-Befehle sind kryptografisch signiert – per Fingerabdruck (Android Keystore) oder Passkey im Browser.

Haus hat das letzte Wort

Befehls-Limits, Trockenlauf-Modus, Whitelist erlaubter Aktionen – auch eine kompromittierte Cloud kann wenig tun.

Netz

Zuhause keine offenen Ports; Server nur 80/443, HTTPS mit SSL-Labs-Bewertung A+, HSTS.

Betrieb

Tägliches Datenbank-Backup + Backup vor jeder Migration; Audit-Log aller Befehle.

Wie sicher ist das System?

BEDROHUNG	SCHUTZ	UMGESETZT MIT
Fremder will sich anmelden	Nur freigegebene Google-Konten; jedes neue Gerät bestätigt der Admin	Google-ID-Token-Prüfung (Signatur, Audience), Allowlist, Geräte-Status
Token wird gestohlen	Kurzlebige Zugangstoken, Refresh-Token rotiert; Wiederverwendung beendet die Sitzung	JWT 10 Min., Refresh-Rotation 30 Tage, Audit-Eintrag
Befehl wird abgefangen oder gefälscht	Jeder Wallbox-Befehl einzeln signiert; Wiederholung unmöglich	ECDSA P-256 im Android-Keystore + Fingerabdruck, WebAuthn-Passkey; ±60 s, Einmal-Nonce
Cloud wird kompromittiert	Das Haus entscheidet selbst: nur erlaubte Befehle, Limits, Trockenlauf	Befehls-Whitelist, Zappi max. alle 30 s, Auto max. 20/Tag; Wechselrichter wird nur gelesen
Angriff aufs Heimnetz	Keine offenen Ports zuhause; Agent verbindet sich nur ausgehend	WebSocket (SignalR) über TLS, eigener Agent-Schlüssel
Mitlesen, Brute Force, fremde Websites	Nur HTTPS mit Forward Secrecy; Ratenbegrenzung; Schutz vor Cross-Site-Anfragen	TLS 1.2/1.3, HSTS, Rate-Limit pro IP, CSRF-Header, CSP, Host-Prüfung
Datenverlust	Automatische Sicherungen und lückenlose Nachvollziehbarkeit	Tägliches Backup + vor jeder Migration; Audit-Log aller Befehle

Geprüft – nicht nur behauptet

A+

SSL-Labs-Bewertung

nur TLS 1.2 / 1.3, nur Cipher mit Forward Secrecy, HSTS

2

offene Ports am Server

Port-Scan von außen: nur 80 und 443;
zu Hause: keiner

30+

automatisierte Sicherheitstests

laufen bei jeder Änderung mit

Beispiele aus den Sicherheitstests

- Gefälschter Google-Token wird abgelehnt und protokolliert
- Abgefangener Befehl kann nicht wiederholt werden
- Befehl ohne gültige Signatur erreicht das Haus nicht
- Widerrufenes Gerät verliert sofort den Zugriff
- Wiederverwendeter Refresh-Token beendet die Sitzung
- Fremde Website kann zu Hause nichts auslösen

Produktentscheidungen – und warum

Automatik raus, Hinweis rein

Ein „Plan für morgen“ hat Entscheidungen des Nutzers überschrieben und verwirrt. Ersetzt durch einen Solar-Hinweis mit einem Tap – der Mensch behält die Kontrolle.

Akku vor Features

Feedback „Handy-Akku leert sich“: Hintergrund-Timer entfernt, Animation auf 30 fps begrenzt. Im Hintergrund macht die App jetzt nichts.

Opt-in statt Spam

Push-Benachrichtigung schaltet jede Person selbst ein – höchstens einmal am Tag.

Sicherheit vs. Komfort

Licht und Steckdosen ohne Fingerabdruck (Alltag), Wallbox und Auto nur signiert (Kosten, Risiko).

Weniger ist mehr

Leere Karten ausgeblendet, „Pause“-Knopf entfernt, Ladeziel in % statt kWh – jedes Detail aus echter Nutzung.

So entstanden: Product Owner + Claude Code

Meine Rolle

- Vision, Use Cases und Priorisierung
- Akzeptanzkriterien und Abnahme im echten Haus
- Produkt-, UX- und Sicherheitsentscheidungen
- Risiken steuern: Freigaben vor jeder Änderung an Geräten und Server

Claude Code

- Architekturvorschläge und Umsetzung (TDD)
- Code für Backend, Cloud, Android, Web
- Automatisierte Tests, Doku, Installationsskripte
- Analyse fremder Protokolle und Fehlerdiagnose

10

Tage

119

Commits

31.500

Zeilen Code

≈ 500

automatisierte Tests

Was schiefging – und was ich daraus gelernt habe

Server gesperrt

Eine TLS-Härtung hat alle Verschlüsselungen deaktiviert – HTTPS und Fernzugriff weg. Über den Recovery-Modus des Hosters in 30 Min. repariert, danach A+.

Learning: Änderungen an Produktion nur mit Trockenlauf, Prüfungen vorher, gemeinsamer Review.

Rollläden stumm

Hersteller-Doku sagte „open/close“, das Gerät wollte „on/off“ – per kontrolliertem Live-Test gefunden.

Learning: Annahmen am echten Gerät validieren, nicht in der Doku.

Akku leer

Ein Timer lief im Hintergrund weiter, Animationen mit 120 fps.

Learning: Nicht-funktionale Anforderungen gehören in jede Abnahme.

Technologie

Backend & Agent

.NET 8 · C# · ASP.NET Core Minimal API · EF Core · SQL Server

Cloud

Windows Server · IIS · SignalR · SQLite · Firebase Cloud Messaging

Mobile

Android · Kotlin · Jetpack Compose · Biometrie / Android Keystore

Web

HTML/JS ohne Framework · Server-Sent Events · Passkeys (WebAuthn)

Geräte & Daten

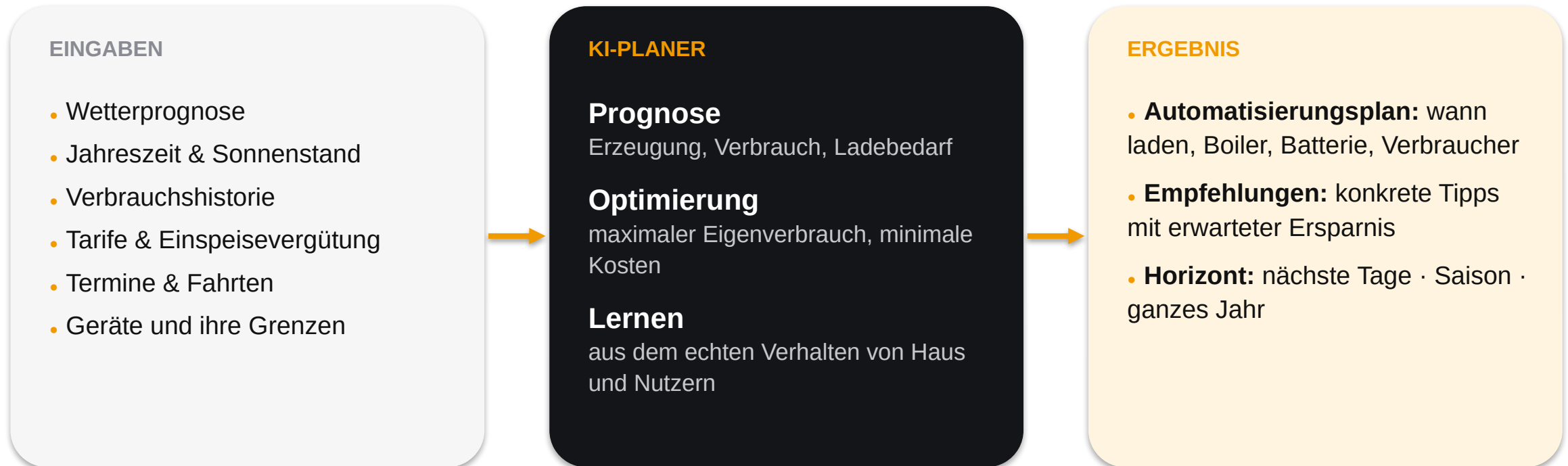
Modbus TCP (SolarEdge) · myenergi API · Tuya lokal · Smartcar · Open-Meteo

Qualität

Test-Driven Development · xUnit · JUnit · drei Sprachen (DE / EN / RU)

Nächster Schritt: KI-gestützte Energieplanung

Vom Reagieren zum Planen: Das System plant den Energiehaushalt für Tage, Seasons und ein ganzes Jahr voraus.



Für Haushalte, Unternehmen und Produktionsbetriebe. Gelernt aus Version 1: Der Plan ist ein Vorschlag – freigegeben vom Menschen, nicht über seinen Kopf hinweg.

Danke!

Fragen – oder eine Live-Demo direkt am System?

Roman Krol · Product Owner · OptiSolar