

Adlar Warmtepomp - Adaptieve Regeling Systeem

Versie: 1.0

Datum: December 2025

Voor: Homey Pro App Development



Inhoudsopgave

1. Executive Summary
2. Architectuur Overzicht
3. Data Mapping & Hardware
4. Component 1: Heating Controller
5. Component 2: Building Model Learner
6. Component 3: Energy Optimizer
7. Component 4: COP Controller
8. Systeem Integratie
9. Setup & Configuratie
10. Flow Voorbeelden
11. Troubleshooting
12. Appendix: Technische Details

Executive Summary

🎯 Wat Doet Het Systeem?

Dit systeem regelt je Adlar Castra warmtepomp intelligent om:

1. **Constante binnentemperatuur** te behouden (binnen $\pm 0.3^{\circ}\text{C}$)
2. **Energiekosten te minimaliseren** door slimme prijs-optimalisatie
3. **Efficiëntie te maximaliseren** door COP-based control
4. **Automatisch te leren** hoe jouw specifieke woning reageert

Verwachte Besparingen

Typisch Scenario (gemiddelde woning, 150m²):

Zonder Adaptieve Regeling:

- Handmatige temperatuur aanpassingen
 - Geen prijs optimalisatie
 - Suboptimale COP instellingen
- Kosten: ~€1.800/jaar

Met Adaptieve Regeling:

- Automatische temperatuur regeling
 - Pre-heating bij lage prijzen
 - COP optimalisatie (+30% efficiënter)
- Kosten: ~€1.100/jaar

Besparing: €700/jaar (39%)

Kernfunctionaliteiten

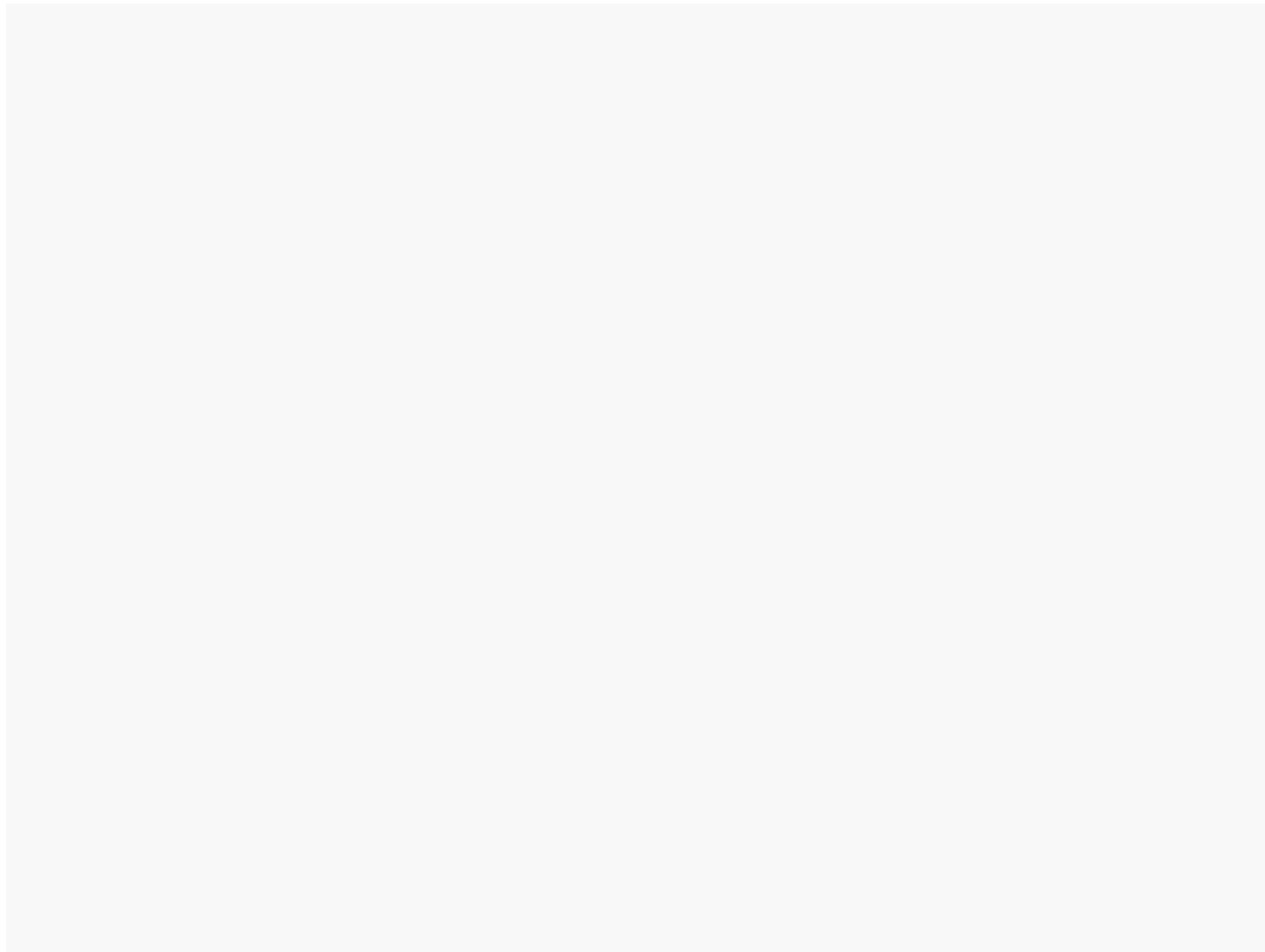
Feature	Beschrijving	Besparing
PI Temperatuur Regeling	Houdt binnentemp constant zonder over/undershoots	Comfort
Building Model Learning	Leert eigenschappen van je woning (massa, isolatie)	Voorspellbaarheid
Energie Prijs Optimalisatie	Pre-heat bij lage prijzen, reduceer bij hoge prijzen	€400–500/jaar
COP Optimalisatie	Vind efficiëntste instellingen voor elke conditie	€200–300/jaar

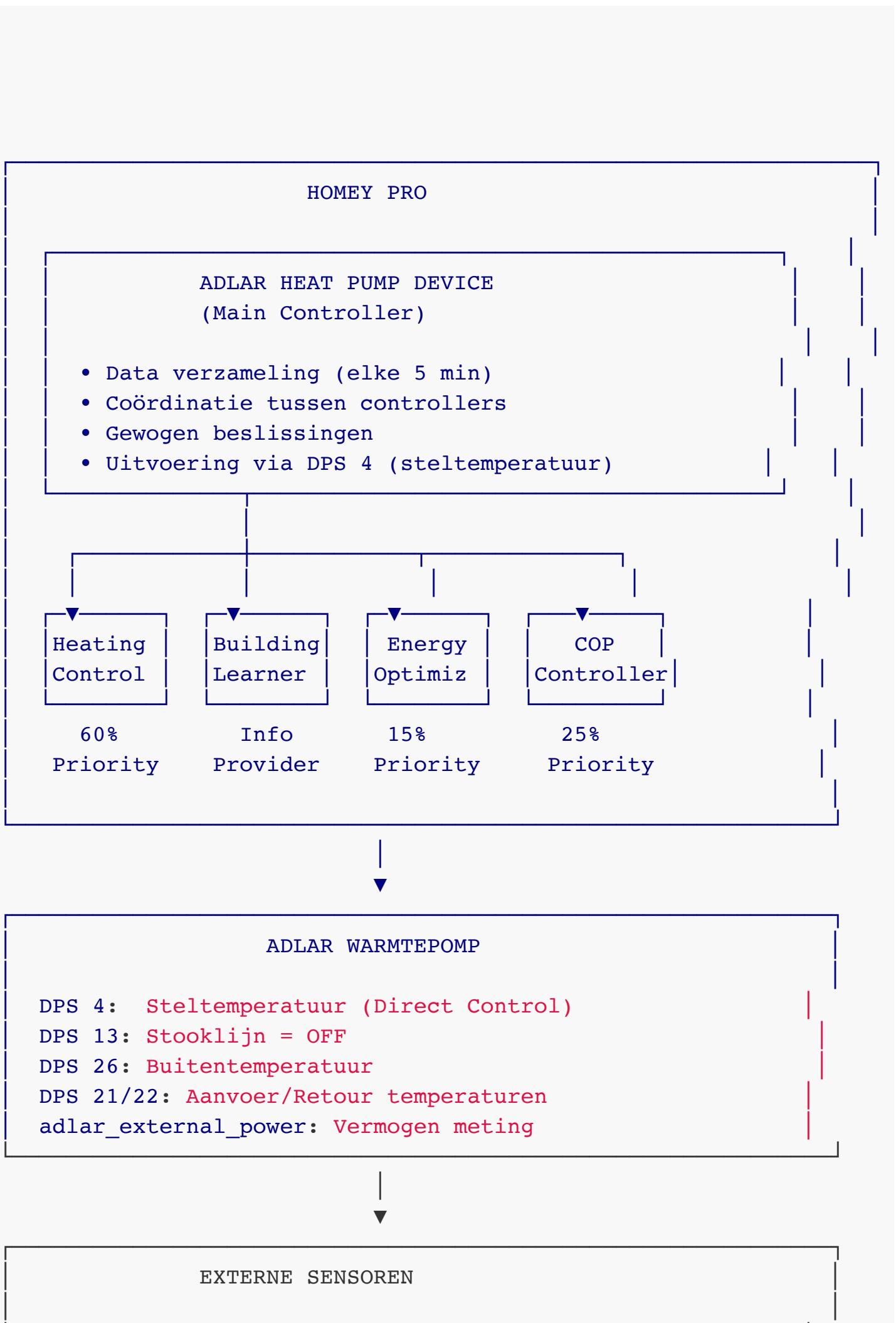
⚡ Snelstart - 5 Minuten Setup

1. **Installeer Homey App** - Upload naar Homey Pro
 2. **Configureer Externe Sensor** - Kies je thermostaat voor binnentemperatuur
 3. **Zet Stooklijn op OFF** - App doet dit automatisch
 4. **Start Learning** - Na 24 uur eerste resultaten
 5. **Activeer Optimalisaties** - Na 1 week volledig operationeel
-

Architectuur Overzicht

🏗 Systeem Diagram





- Binnentemperatuur (Thermostaat)
- Weer API (Zonnestraling)
- Energie API (Prijzen)

↻ Data Flow (Elke 5 Minuten)

1. DATA VERZAMELEN

- Binnentemp (externe sensor)
- Buitentemp (DPS 26)
- Aanvoer/Retour temp (DPS 21/22)
- Vermogen (adlar_external_power)
- Zonnestraling (API)

2. VERWERKEN

- Heating Controller: Bereken temp correctie
- Building Learner: Update **model parameters**
- COP Controller: Analyseer efficiëntie
- Energy Optimizer: Check prijzen

3. BESLISSEN

- Gewogen gemiddelde: 60% comfort + 25% COP + 15% prijs

4. UITVOEREN

- Update DPS 4 (steltemperatuur)

Data Mapping & Hardware

📡 Adlar Castra Aurora II - DPS Mapping

CORRECTE DATA POINTS:

DPS	Capability	Wat is het?
4	target_temperature	Steltemperatuur

13	adlar_enum_countdown_set	Stooklijn instelling
21	measure_temperature.temp_top	Aanvoertemperatuur
22	measure_temperature.temp_bottom	Retourtemperatuur
26	measure_temperature.around_temp	Buitentemperatuur
27	adlar_state_compressor_state	Compressor status
20	measure_frequency.compressor_strength	Compressor frequentie
5	adlar_enum_work_mode	Werk modus
n/a	adlar_external_power	Werkelijk vermogen (W)

EXTERNE DATA:

- **Binnentemperatuur:** VAN EXTERNE SENSOR (thermostaat, niet van WP!)
- **Zonnestraling:** Via Homey Weather API of schatting
- **Energieprijzen:** Via EnergyZero of ENTSO-E API

ADLAR COP CAPABILITIES (ingebouwd):

Capability	Beschrijving	Gebruik voor
adlar_cop	Huidige COP (realtime)	Directe feedback & aanpassingen
adlar_cop_daily	24u voortschrijdend gemiddelde (gewogen op looptijd)	Dagelijkse optimalisatie
adlar_cop_weekly	7d voortschrijdend gemiddelde	Trend analyse
adlar_cop_monthly	30d voortschrijdend gemiddelde	Seizoenspatronen
adlar_scop	Seizoensgebonden COP (EN 14825)	Jaarlijkse evaluatie

⚠ KRITISCH: Control Mode

DEZE APP GEBRUIKT DIRECTE STELTEMPERATUUR REGELING

-  DPS 4 (steltemperatuur) = ACTIEF
-  DPS 13 (stooklijn) = ALTIJD OP 'OFF'

Verander NOOIT handmatig de stooklijn!

Waarom deze keuze?

Aspect	Stooklijn (DPS 13)	Steltemperatuur (DPS 4)
Controle	Automatisch o.b.v. buiten	Direct instelbaar
Flexibiliteit	Beperkt	Volledig
Adaptief	Nee	Ja
ML Compatible	Nee	Ja ✓
Onze keuze		

Hardware Vereisten

Minimaal:

- Adlar Castra Aurora II warmtepomp
- Homey Pro (Early 2023 of nieuw)
- Externe temperatuur sensor (thermostaat)
- Vermogen meting (via adlar_external_power)

Aanbevolen:

- Meerdere temperatuur sensoren (meerdere kamers)
- Weer station met zonnestraling sensor
- Slimme meter met P1 uitlezing

Component 1: Heating Controller



Concept: PI Regelaar

Wat is een PI Regelaar?

Een **PI (Proportional-Integral) controller** is een klassieke regeltechniek die twee componenten combineert:

P – PROPORTIONAL (Proportioneel)

"Hoe ver zit ik van mijn doel?"

Kleine afwijking → Kleine correctie

Grote afwijking → Grote correctie

Voorbeeld:

19.5°C vs doel $20^\circ\text{C} = 0.5^\circ\text{C}$ verschil

→ Correctie: 0.5×3.0 (K_p) = 1.5°C aanpassing

I – INTEGRAL (Integraal)

"Ben ik STRUCTUREEL te hoog of laag?"

Kijkt naar gemiddelde fout over tijd

Compenseert voor blijvende afwijkingen

Voorbeeld:

Laatste 2 uur gemiddeld 0.3°C te laag

→ Extra correctie: 0.3×1.5 (K_i) = 0.45°C

TOTAAL: $1.5^\circ\text{C} + 0.45^\circ\text{C} = 1.95^\circ\text{C}$ aanpassing nodig

Waarom P én I?

ALLEEN P (zonder I):

Temperatuur: **19.8°C**

Doel: **20.0°C**

Probleem: Blijft **0.2°C** te laag hangen!

Reden: Bij kleine fout → kleine correctie → nooit helemaal goed

P + I (compleet):

Temperatuur: **20.0°C**

Doel: **20.0°C**

Perfect! I-term zag structurele afwijking en corrigeerde



Wiskundige Formule

Correctie = $(K_p \times \text{Huidige_Fout}) + (K_i \times \text{Gemiddelde_Fout_Tijd})$

Waarbij:

- K_p = Proportionele versterking (default: 3.0)
- K_i = Integratie versterking (default: 1.5)
- Huidige_Fout = Target - Actueel ($^{\circ}\text{C}$)
- Gemiddelde_Fout_Tijd = Gemiddelde over laatste 2 uur



Configuratie Parameters

Parameter	Default	Bereik	Beschrijving
K_p (Proportional Gain)	3.0	1.0 - 5.0	Reactie op huidige fout
K_i (Integral Gain)	1.5	0.5 - 3.0	Reactie op lange-termijn fout
Deadband	0.3°C	0.1 - 1.0°C	Geen actie binnen ± dit bereik
Min Wait Time	20 min	10 - 60 min	Wachttijd tussen aanpassingen
Integration	2 min	1 - 6 min	Hoe ver terug kijken voor



Praktisch Voorbeeld - Dag Cyclus

MAANDAG - TYPISCHE WERKDAG

06:00 - Wakker worden

- Binnen: 18.5°C | Doel: 20°C | Fout: +1.5°C
- P-term: $1.5 \times 3.0 = 4.5^\circ\text{C}$
- I-term: 0.0 (net begonnen)
- Totaal: 4.5°C (gelimiteerd tot max +3°C)
- Actie: Steltemp 35°C → 38°C
-  Wacht 20 minuten voor volgende controle

06:20 - Check 1

- Binnen: 18.8°C | Doel: 20°C | Fout: +1.2°C
- P-term: $1.2 \times 3.0 = 3.6^\circ\text{C}$
- I-term: $0.15 \times 1.5 = 0.225^\circ\text{C}$ (gemiddeld)
- Actie: Maintain (recent aangepast)
- Status: Aan het opwarmen...

07:00 - Ochtend stabiel

- Binnen: 19.7°C | Doel: 20°C | Fout: +0.3°C
- Check: Binnen deadband ($\pm 0.3^\circ\text{C}$)
- Actie: Maintain
- Status:  Perfect

12:00 - Zon door ramen

- Binnen: 20.6°C | Doel: 20°C | Fout: -0.6°C
- P-term: $-0.6 \times 3.0 = -1.8^\circ\text{C}$
- I-term: $-0.2 \times 1.5 = -0.3^\circ\text{C}$
- Totaal: -2.1°C
- Actie: Steltemp 38°C → 36°C (-2°C)
- Reden: "Te warm door zon"

14:00 - Bewolkt

- Binnen: 19.8°C | Doel: 20°C
- Check: Binnen deadband
- Actie: Maintain
- Status:  Stabiel

18:00 - Avond

```
└─ Binnen: 19.6°C | Doel: 20°C | Fout: +0.4°C
└─ P-term: 0.4 × 3.0 = 1.2°C
└─ I-term: klein
└─ Actie: Steltemp +0.5°C
└─ Status: Kleine correctie
```

22:00 - Nachtmodus

```
└─ Doel wijzigt: 20°C → 18°C (nachtmodus)
└─ Binnen: 19.8°C | Nieuw doel: 18°C
└─ Fout: -1.8°C (nu TE WARM voor nacht)
└─ Actie: Steltemp 36°C → 33°C
└─ Status: Voorbereiden voor nacht
```

🔧 Implementatie

Het PI regelaar algoritme doorloopt elk interval de volgende stappen:

1. **Basis checks** - Controleer of regeling actief is en minimale wachttijd verstreken is
2. **Bereken fouten** - Huidige fout (target - werkelijk) en integrale fout (gemiddelde over tijd)
3. **Check deadband** - Geen actie als binnen tolerantie ($\pm 0.3^\circ\text{C}$ default)
4. **PI Berekening** - Bereken P-term en I-term, combineer voor totale correctie
5. **Limiteer** - Beperk aanpassing tot maximum ($\pm 3^\circ\text{C}$ default)
6. **Bepaal actie** - Verhogen, verlagen of handhaven van steltemperatuur



Technische Implementatie: Voor volledige code implementatie, zie [Adaptieve_Regeling_Technische_Implementatie.md](#) - Component 1: PI Controller

🎯 Tuning Guidelines

Agressieve Regeling (snel reageren): - K_p : 4.0–5.0 - K_i : 2.0–3.0 - Deadband: 0.2°C

- Use case: Gemiddelde woning

Gebalanceerde Regeling (aanbevolen):

- Kp: 3.0
- Ki: 1.5
- Deadband: 0.3°C
- Use case: Gemiddelde woning

Conservatieve Regeling (stabiel, langzaam):

- Kp: 2.0
 - Ki: 1.0
 - Deadband: 0.5°C
 - Use case: Goede isolatie, grote thermische massa
-

Component 2: Building Model Learner

 **Concept: Machine Learning voor Gebouw Eigenschappen**

De Badkuip Analogie

JE WONING = BADKUIP MET LEK

Kraan (verwarming)



WATER
(warmte)

Niveau =
Temperatuur

← Hoeveelheid =
Thermische **massa** (C)

▼ Lek (warmteverlies)



← **Grootte lek** = UA
(isolatie kwaliteit)

Extra water erbij:

- Zon door raam
- Mensen & apparaten

Fysische Vergelijking

Fundamentele Warmte Balans:

$$dT/dt = (1/C) \times [P_{verwarming} - UA \times (T_{in} - T_{uit}) + P_{zon} + P_{intern}]$$

Waarbij:

- **dT/dt** = Temperatuur verandering per uur [$^{\circ}\text{C}/\text{h}$]
- **C** = Thermische massa [$\text{kWh}/^{\circ}\text{C}$]
 - Hoeveel energie om 1°C op te warmen
- **UA** = Warmteverlies coëfficiënt [$\text{kW}/^{\circ}\text{C}$]
 - Hoeveel warmte verlies per $^{\circ}\text{C}$ verschil
- **P_verwarming** = WP vermogen [kW]
- **P_zon** = Zonnewinst [kW]
- **P_intern** = Interne warmte (mensen, apparaten) [kW]

Te Leren Parameters

1. Thermische Massa (C)

Wat is het?

- Hoeveel energie nodig om woning 1°C op te warmen
- Eenheid: kWh/°C

Typische Waardes:

Licht gebouw (hout, weinig beton):	5–8 kWh/°C
Gemiddelde woning:	10–15 kWh/°C
Zware woning (veel beton, vloerverw):	15–25 kWh/°C
Passief huis:	25–40 kWh/°C

Praktisch Voorbeeld:

$C = 15 \text{ kWh/}^{\circ}\text{C}$ betekent:

- Om 1°C op te warmen: 15 kWh energie nodig
- Met 3 kW WP: duurt 5 uur
- Grote massa = langzaam reageren (goed voor voorverwarmen!)

2. Warmteverlies Coëfficiënt (UA)

Wat is het?

- Warmteverlies per graad temperatuurverschil
- Eenheid: kW/°C (of W/°C)

Typische Waardes:

Slecht geïsoleerd (oud huis):	400–600 W/°C
Gemiddeld geïsoleerd:	200–400 W/°C
Goed geïsoleerd (nieuwbouw):	100–200 W/°C
Passief huis:	<50 W/°C

Praktisch Voorbeeld:

UA = 300 W/°C betekent:

Bij 20°C binnen en 0°C buiten (verschil: 20°C):

Warmteverlies = 0.3 kW/°C × 20°C = 6 kW continu!

Dit moet WP compenseren om temperatuur te handhaven.

3. Zonnewinst Factor (g)

Wat is het?

- Hoeveel opwarming door zonnestraling via ramen
- Eenheid: dimensieloos (0–2)

Typische Waardes:

Weinig ramen, noord-oriëntatie: 0.1–0.2

Gemiddeld (mix van oriëntaties): 0.3–0.5

Veel ramen op zuid: 0.6–1.0

Praktisch Voorbeeld:

g = 0.5 en 10m² raam op zuid:

Zonnige dag (500 W/m² straling):

Gratis warmte = 0.5 × 0.5 kW/m² × 10m² = 2.5 kW!

(Even veel als kleine warmtepomp)

4. Interne Warmtewinst (P_internal)

Wat is het?

- Gemiddelde warmte van mensen, apparaten, koken
- Eenheid: kW (constant gemiddelde)

Typische Waardes:

1 persoon: ~100 W
Computer/TV: 50-200 W
Koelkast continu: 50-100 W
Koken (gemiddeld): 200 W over dag

Gemiddeld huishouden:

- └ Overdag: 0.3-0.5 kW
- └ Avond: 0.5-0.8 kW
- └ Nacht: 0.1-0.2 kW

5. Tijdsconstante (τ)

Wat is het?

- Hoe snel reageert woning op veranderingen
- Berekend: $\tau = C / UA$
- Eenheid: uren

Typische Waardes:

Snelle woning: 2-4 uur
Gemiddelde woning: 4-8 uur
Trage woning: 8-16 uur
Passief huis: 16-48 uur

Praktisch Voorbeeld:

$\tau = 8$ uur betekent:

Als verwarming uitvalt bij 20°C (buiten 0°C):

- └ Na 8 uur: gedaald naar ~7.4°C (63% van verschil)
- └ Na 16 uur: gedaald naar ~2.7°C
- └ Lange tijdsconstante = goed voor voorverwarmen!

🤖 Machine Learning: Recursive Least Squares

Waarom RLS?

Alternatieve Methoden:

Methode Pro Con Geschikt?
--- --- --- ---
Linear Regression Simpel Statisch ⚠ Matig
RLS Real-time, adapteert Matige complexiteit ✓ Perfect
Kalman Filter Robuust tegen ruis Complex ✓ Goed
Neural Network Zeer flexibel Te complex, veel data ✗ Overkill

RLS = Recursive Least Squares:

- Leert TIJDENS gebruik (online learning)
- Past zich aan bij seizoenen/wijzigingen
- Computationeel licht (past op Homey)
- Geeft betrouwbaarheid indicator

Hoe Werkt RLS?

RECURSIVE LEAST SQUARES - STAP VOOR STAP

START (Dag 0):

- └ Initiële gok: $C=15$, $UA=0.3$, $g=0.4$, $P_{int}=0.3$
- └ Onzekerheid: HOOG (we weten het niet zeker)

ELKE 5 MINUTEN:

1. VOORSPEL met huidig model

"Met mijn huidige parameters, wat zou de temperatuur moeten zijn over 5 min?"

Voorbeeld: **Voor spel** 19.6°C

2. MEET werkelijkheid

"Wat is de temp ECHT?"

Voorbeeld: **Werkelijk** 19.8°C

3. BEREKEN fout

Fout = Werkelijk - Voorspeld

Voorbeeld: $19.8 - 19.6 = +0.2^{\circ}\text{C}$

(We voorspelden te laag!)

4. UPDATE parameters

"Pas parameters iets aan om fout te verkleinen"

Als we keer op keer te laag voorspellen:

→ C is waarschijnlijk GROTER (meer massa)

Als we bij kou te laag voorspellen:

→ UA is waarschijnlijk GROTER (meer verlies)

5. VERLAAG onzekerheid

"Hoe meer data, hoe zekerder we worden"

Covariance matrix verkleint met elke meting

RESULTAAT NA TIJD:

- 1 dag: Basis schatting (30% confidence)
- 3 dagen: Redelijk (50% confidence)
- 1 week: Goed (70% confidence)
- 1 maand: Zeer goed (90% confidence)

Wiskundige Details (voor geïnteresseerden)

Model in Lineaire Vorm:

$$dT/dt = \phi^T \times \theta$$

Waarbij:

ϕ = Feature vector (wat we meten):
[P_heating, -(T_in - T_out), Solar/1000, 1]

θ = Parameter vector (wat we leren):
[1/C, UA/C, g/C, P_internal/C]

Voorbeeld meting:

$\phi = [5 \text{ kW}, -15^\circ\text{C}, 0.3 \text{ kW}, 1]$
 $\theta = [0.067, 0.020, 0.027, 0.020]$
 $dT/dt = 5 \times 0.067 + (-15) \times 0.020 + 0.3 \times 0.027 + 1 \times 0.020$
 $= 0.335 - 0.300 + 0.008 + 0.020$
 $= 0.063^\circ\text{C per 5min} = 0.76^\circ\text{C per uur}$

RLS Update Formules:

1. Prediction error:

$$e(k) = y(k) - \phi(k)^T \times \theta(k-1)$$

2. Gain vector:

$$K(k) = P(k-1) \times \phi(k) / (\lambda + \phi(k)^T \times P(k-1) \times \phi(k))$$

3. Parameter update:

$$\theta(k) = \theta(k-1) + K(k) \times e(k)$$

4. Covariance update:

$$P(k) = (P(k-1) - K(k) \times \phi(k)^T \times P(k-1)) / \lambda$$

Waarbij:

- k = tijdstap
- λ = forgetting factor (0.98 = adapteert over ~ 2 dagen)
- P = covariance matrix (onzekerheid)

Leerproces Over Tijd

LEARNING PROGRESSION

DAG 1 (288 samples @ 5min):

- └ C: $15.0 \pm 8.0 \text{ kWh/}^\circ\text{C}$ [Grote onzekerheid]
- └ UA: $0.30 \pm 0.15 \text{ kW/}^\circ\text{C}$
- └ Confidence: 25%
- └ Voorspelling: $\pm 2^\circ\text{C}$ fout
- └ Status:  Aan het leren

DAG 3:

- └ C: $14.2 \pm 3.0 \text{ kWh/}^\circ\text{C}$ [Onzekerheid kleiner]
- └ UA: $0.28 \pm 0.08 \text{ kW/}^\circ\text{C}$
- └ Confidence: 45%
- └ Voorspelling: $\pm 1^\circ\text{C}$ fout
- └ Status:  Redelijk

WEEK 1:

- └ C: $14.5 \pm 1.5 \text{ kWh/}^\circ\text{C}$ [Vrij zeker]
- └ UA: $0.285 \pm 0.04 \text{ kW/}^\circ\text{C}$
- └ Confidence: 72%
- └ Voorspelling: $\pm 0.5^\circ\text{C}$ fout
- └ Status:  Goed

WEEK 4:

- └ C: $14.3 \pm 0.5 \text{ kWh/}^\circ\text{C}$ [Zeer zeker!]
- └ UA: $0.287 \pm 0.02 \text{ kW/}^\circ\text{C}$
- └ Confidence: 91%
- └ Voorspelling: $\pm 0.2^\circ\text{C}$ fout
- └ Status:  Excellent

Toepassingen

1. Temperatuur Voorspelling

Het building model kan temperatuur voorspellen op basis van huidige

condities en verwachte verwarming, buitentemperatuur en zonnestraling.

Use Case: Avond planning

Nu 18:00: 19.5°C

Voorspelling 22:00: 18.8°C (te koud voor slapen!)

Actie: Verhoog verwarming NU preventief

2. Pre-heat Berekening

Het model kan berekenen hoeveel tijd nodig is om van huidige temperatuur naar een doeltemperatuur te komen, gegeven de verwachte buitentemperatuur en maximaal beschikbaar vermogen.

Use Case: GPS-based preheating

Homey detecteert: "30 min van huis"

Model weet: "2.5 uur nodig voor opwarmen"

Besluit: "Te laat! Start NU maximum verwarming"

3. Benodigd Vermogen

Het model kan het benodigde verwarmingsvermogen berekenen om een temperatuur te handhaven, rekening houdend met:

- Warmteverlies ($UA \times$ temperatuurverschil binnen-buiten)
- Zonnewinst ($g \times$ zonnestraling)
- Interne warmteinst (mensen, apparaten, koken)

Use Case: COP optimalisatie

Weet dat 4 kW nodig is

WP kan dit leveren bij COP 3.5 met 1.14 kW elektrisch

of bij COP 2.8 met 1.43 kW elektrisch

→ Kies eerste: 20% minder stroom!

RLS Algoritme Implementatie

Het Building Model Learner systeem gebruikt Recursive Least Squares (RLS) algoritme met:

- **Parameter vector θ :** $[1/C, UA/C, g/C, P_{int}/C]$ - te leren thermische eigenschappen
- **Covariance matrix P :** 4×4 matrix die onzekerheid bijhoudt
- **Forgetting factor λ :** 0.998 (aanpassing over ~2 weken voor seizoensveranderingen)

Hoofdfuncties:

- `addMeasurement()` - Verwerk nieuwe temperatuurmeting en update parameters met RLS
- `predictTemperature()` - Voorspel toekomstige temperatuur op basis van geleerd model
- `getModel()` - Haal huidige parameterestimates en confidence level op



Technische Implementatie: Voor volledige RLS algoritme, matrix operaties en code, zie

[*Adaptieve_Regeling_Technische_Implementatie.md*](#) - Component 2:
Building Model Learner

Component 3: Energy Optimizer

Concept: Dynamische Energieprijzen

Nederlandse Energiemarkt

Day-Ahead Pricing:

Nederlandse consumenten met dynamisch contract **betalen** uurprijzen **die 1** dag van tevoren **bekend** zijn.

Prijzen worden **bepaald** door:

- └ Vraag (pieken 's ochtends en avonds)
- └ Aanbod (wind/zon productie)
- └ Import/export naar buurlanden
- └ Weer (temperatuur, wind)

Typische Prijsvariatie:

WINTERDAG – PRIJZEN (€/kWh incl. BTW)

00:00	———	€0.08	[Veel wind 's nachts]
03:00	———	€0.06	[Minimale vraag] ← LAAGSTE
06:00	———	€0.15	[Ochtendspits start]
08:00	———	€0.28	[Volle piektijd]
12:00	———	€0.22	[Zon + industrie]
17:00	———	€0.45	[AVONDPIEK] ← HOOGSTE
19:00	———	€0.38	[Nog steeds druk]
22:00	———	€0.18	[Bedtijd]
23:00	———	€0.10	[Wind neemt toe]

VERSCHIL: €0.06 – €0.45 = Factor 7.5×!

POTENTIEEL: €400–600 besparing/jaar door slim laden

🎯 Optimalisatie Strategie

Drempel Definitie

Prijs Categorieën:

ZEER LAAG	< €0.10/kWh	Pre-heat MAX
LAAG	€0.10-0.15	Pre-heat
NORMAAL	€0.15-0.25	Standaard
HOOG	€0.25-0.35	Reduceren
ZEER HOOG	> €0.35	Minimaal

Actie Matrix

Prijs	Actie	Target Aanpassing	Reden
Zeer Laag	Pre-heat	+1.5°C	Goedkoop voorverwarmen, woning = batterij
Laag	Pre-heat	+0.75°C	Mild voorverwarmen
Normaal	Maintain	0°C	Standaard comfort
Hoog	Reduce	-0.5°C	Besparen binnen comfort
Zeer Hoog	Reduce Max	-1.0°C	Maximaal besparen

Voorwaarden voor Pre-heat:

- ✓ Prijs is laag (< €0.15/kWh)
- ✓ Dure periode komt binnen 2-4 uur
- ✓ Binnentemp < target + max_offset (bijv. < 21.5°C)
- ✓ Building heeft goede thermische massa ($\tau > 6$ h)

Voorwaarden voor Reduce:

- ✓ Prijs is hoog (> €0.25/kWh)
- ✓ Binnentemp > target - max_offset (bijv. > 19°C)
- ✓ Geen bezoekers/speciale events

💰 Kosten-Baten Analyse

Voorbeeld Scenario - Winterdag:

DINSDAG WINTER - MET vs ZONDER OPTIMALISATIE

ZONDER OPTIMALISATIE (Constante verwarming):

- └ 00-06: $2 \text{ kW} \times 6\text{h} \times €0.08 = €0.96$
- └ 06-12: $3 \text{ kW} \times 6\text{h} \times €0.22 = €3.96$
- └ 12-18: $2 \text{ kW} \times 6\text{h} \times €0.30 = €3.60$
- └ 18-24: $4 \text{ kW} \times 6\text{h} \times €0.40 = €9.60$
- └ **TOTAAL: €18.12/dag**

MET OPTIMALISATIE:

23:00 DAG ERVOOR - Pre-heat

- └ **Actie: Extra verwarmen naar 21.5°C**
- └ **Extra: $3 \text{ kW} \times 2\text{h} \times €0.08 = €0.48$**
- └ **Status: Woning warmt op**

00-06: Maintain pre-heat

- └ **Minimaal: $1 \text{ kW} \times 6\text{h} \times €0.08 = €0.48$**
- └ **Temp: Daalt traag van 21.5°C naar 20.5°C**

06-12: NIET verwarmen (duur!)

- └ **Vermogen: $0 \text{ kW} \times 6\text{h} = €0.00 (!)$**
- └ **Temp: Daalt verder naar 19.8°C**
- └ **Besparing: €3.96 gespaard!**

12-18: Standaard

- └ $2 \text{ kW} \times 6\text{h} \times €0.22 = €2.64$
- └ **Temp: Terug naar 20°C**

18-24: Reduceren bij piek

- └ $2 \text{ kW} \times 6\text{h} \times €0.40 = €4.80 \text{ (i.p.v. 4kW)}$
- └ **Temp: 19.5°C (acceptabel)**
- └ **Besparing: €4.80 gespaard!**

TOTAAL MET OPTIMALISATIE:

- └ **Kosten: €0.48 + €0.48 + €0 + €2.64 + €4.80 = €8.40**
- └ **Normaal: €18.12**
- └ **BESPARING: €9.72/dag = €3.548/jaar! 💰**

Realistische Jaarbesparingen:

Conservatief (kleinere woning):	€300-400/jaar
Gemiddeld:	€400-600/jaar
Optimaal (grote woning, goed τ):	€600-800/jaar

🌐 API Integratie

EnergyZero API (Gratis, Nederlands)

Endpoint:

```
https://api.energyzero.nl/v1/energyprices
```

Response Format:

```
{
  "Prices": [
    {
      "readingDate": "2025-12-10T00:00:00Z",
      "price": 0.08234,
      "tariffReturn": 0.05123
    },
    {
      "readingDate": "2025-12-10T01:00:00Z",
      "price": 0.07892,
      "tariffReturn": 0.05001
    },
    ...
  ]
}
```

Implementatie:

De EnergyZero API levert dagelijkse prijzen vanaf 14:00 uur voor de volgende

dag. De service haalt deze prijzen op en splitst ze in historische data (voor analyse) en forecast data (voor beslissingen).

ENTSO-E Transparency Platform (Vereist API key)

Voor: Directe toegang tot Europese marktprijzen

Voordeel: Meer detail, meerdere landen

Nadeel: Vereist gratis registratie op transparency.entsoe.eu

De ENTSO-E API biedt toegang tot day-ahead prijzen voor alle Europese landen via een REST API met parameters voor documenttype, gebied (domain), en periode.



Beslissingslogica

Het beslissingsalgoritme werkt in 4 stappen:

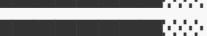
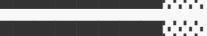
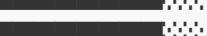
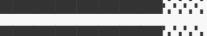
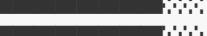
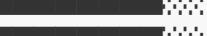
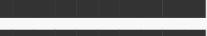
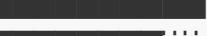
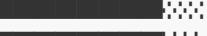
1. **Haal huidige en gemiddelde prijs op** - Vergelijk actuele prijs met 24-uurs gemiddelde
2. **Bepaal prijscategorie** - Very low, low, normal, high, very high op basis van drempelwaarden
3. **Check forecast** - Kijk 2 uur vooruit of dure periode nadert
4. **Bepaal actie per categorie:**
 5. **Very low** + dure piek nadert → Pre-heat (+1.5°C)
 6. **Very high** + voldoende warmte → Reduce (-1.0°C)
 7. Anders → Maintain

Besparingsberekening: Het systeem schat besparingen op basis van temperatuuraanpassing ($\pm 1^\circ\text{C} = \pm 10\%$ vermogen), COP, en prijsverschil tussen actueel en gemiddeld.



Technische Implementatie: Voor volledige prijsoptimalisatie algoritme, API integratie en beslisbomen, zie

Visualisatie - Dag Overzicht

PRICE-BASED OPTIMIZATION – WOENSDAG					
00:00	€0.08		Pre-heat	21.5°C	
01:00	€0.07		Pre-heat	21.5°C	
02:00	€0.06		Pre-heat	21.5°C	
03:00	€0.07		Maintain	21.3°C	
04:00	€0.08		Maintain	21.0°C	
05:00	€0.10		Maintain	20.8°C	
06:00	€0.15		Reduce	20.5°C	
07:00	€0.25		Reduce	20.2°C	
08:00	€0.30		Reduce	19.9°C	
09:00	€0.28		Standard	19.8°C	
10:00	€0.24		Standard	19.9°C	
11:00	€0.20		Standard	20.0°C	
12:00	€0.18		Standard	20.1°C	
13:00	€0.16		Standard	20.0°C	
14:00	€0.17		Standard	20.0°C	
15:00	€0.19		Standard	20.0°C	
16:00	€0.28		Maintain	20.0°C	
17:00	€0.42		Reduce!	19.5°C	← Hoogste
18:00	€0.45		Reduce!	19.2°C	← Piek
19:00	€0.38		Reduce	19.3°C	
20:00	€0.32		Standard	19.7°C	
21:00	€0.25		Standard	19.9°C	
22:00	€0.18		Standard	20.0°C	
23:00	€0.12		Maintain	20.0°C	
KOSTEN ZONDER: €18.50					
KOSTEN MET: €10.20					
BESPARING: €8.30 (45%)					

Component 4: COP Controller

🎯 Concept: Multi-Horizon COP Analysis

Ingebouwde COP Capabilities

De Adlar warmtepomp heeft al uitstekende COP metingen ingebouwd:

```
// GEEN eigen COP berekening nodig!
const copData = {
  current: device.getCapabilityValue('adlar_cop'),           // Realtime
  daily: device.getCapabilityValue('adlar_cop_daily'),        // 24u gem.
  weekly: device.getCapabilityValue('adlar_cop_weekly'),      // 7d gem.
  monthly: device.getCapabilityValue('adlar_cop_monthly'),    // 30d gem.
  seasonal: device.getCapabilityValue('adlar_scop')           // SCOP (seizoen)
};
```

Wat is COP?

Definitie:

COP = Warmte Output / Elektrisch Input

Een warmtepomp is GEEN kachel maar een warmte POMP:

- Haalt warmte uit buitenlucht
- Pompt het naar binnen
- Gebruikt elektriciteit voor het pompen

Voorbeeld:

Elektrisch input: 2.0 kW

Warmte output: 7.0 kW

COP: 7.0 / 2.0 = 3.5

Betekenis:

- |- Voor elke 1 kW elektriciteit...
- |- ...krijg je 3.5 kW warmte!
- |- Andere 2.5 kW komt "gratis" uit buitenlucht
- |- 71% efficiënter dan elektrische kachel!

Vergelijking Verwarmingsmethoden

Methode	Input	Output	COP	Kosten ($\text{€}0.25/\text{kWh}$)
Elektrische kachel	2 kW	2 kW	1.0	€0.50/uur
Gas CV (HR107)	2 kW equiv	2.14 kW	1.07	~€0.30/uur
Warmtepomp (slecht)	2 kW	4 kW	2.0	€0.25/uur (2× kachel)
Warmtepomp (OK)	2 kW	7 kW	3.5	€0.14/uur (3.5× kachel)
Warmtepomp (optimaal)	2 kW	10 kW	5.0	€0.10/uur (5× kachel)

Conclusie: Verschil tussen COP 2.0 en 5.0 = 2.5× goedkoper!



Factoren die COP Beïnvloeden

1. Temperatuur Verschil (GROOTSTE FACTOR)

HOE GROTER HET VERSCHIL, HOE LAGER DE COP

Scenario A: Klein verschil (makkelijk pompen)

- Buiten: 10°C
- Aanvoer: 30°C
- Verschil: 20°C
- COP: ~5.0 ✓ Uitstekend!

Scenario B: Gemiddeld verschil

- Buiten: 5°C
- Aanvoer: 40°C
- Verschil: 35°C
- COP: ~3.5 ✓ Goed

Scenario C: Groot verschil (zwaar pompen)

- Buiten: -5°C
- Aanvoer: 55°C
- Verschil: 60°C
- COP: ~2.0 ✗ Matig

Fysische Verklaring:

Carnot efficiëntie (theoretisch max):

$$\text{COP}_{\text{max}} = T_{\text{hot}} / (T_{\text{hot}} - T_{\text{cold}})$$

Met absolute temperaturen (Kelvin):

$$T_{\text{cold}} = -5^{\circ}\text{C} = 268\text{K}$$

$$T_{\text{hot}} = 55^{\circ}\text{C} = 328\text{K}$$

$$\text{COP}_{\text{max}} = 328 / (328 - 268) = 328 / 60 = 5.47$$

Praktische COP \approx 40-50% van Carnot

$$\rightarrow \text{COP} \approx 2.0 - 2.7$$

Bij kleiner verschil:

$$T_{\text{cold}} = 10^{\circ}\text{C} = 283\text{K}$$

$$T_{\text{hot}} = 30^{\circ}\text{C} = 303\text{K}$$

$$\text{COP}_{\text{max}} = 303 / 20 = 15.15$$

Praktisch: COP \approx 5.0 - 7.0

2. Aanvoertemperatuur

LAGERE AANVOER = HOGERE COP

Bij 5°C buiten:

- Aanvoer 30°C → COP 5.5 ✓✓
- Aanvoer 35°C → COP 4.5 ✓
- Aanvoer 40°C → COP 3.5 ○
- Aanvoer 50°C → COP 2.5 ✗
- Aanvoer 55°C → COP 2.0 ✗✗

MAAR: Te lage aanvoer → niet warm genoeg!

OPTIMUM: Laagste aanvoer die nog comfort geeft

3. Buitentemperatuur

WARMER BUITEN = HOGERE COP

Bij vaste 40°C aanvoer:

- Buiten 15°C → COP 6.0 ✓✓ (lente/herfst)
- Buiten 10°C → COP 5.0 ✓
- Buiten 5°C → COP 3.8 ○
- Buiten 0°C → COP 3.0 ○
- Buiten -5°C → COP 2.3 ✗
- Buiten -10°C → COP 1.8 ✗✗

Oorzaak: Minder warmte in koude lucht

🎯 Multi-Horizon Optimalisatie Strategie

Doelstelling

GEBRUIK: Bestaande COP metingen van verschillende tijdshorizons

ANALYSEER: Trends en patronen **in** efficiëntie

OPTIMALISEER: Op juiste tijdschaal (realtime vs lange termijn)

RESULTAAT: Maximale COP binnen comfort grenzen

Time-Horizon Strategie

Horizon	Capability	Doel	Actie Typ
Realtime	adlar_cop	Directe aanpassingen	Steltemp $\pm 1^{\circ}\text{C}$
Dagelijks	adlar_cop_daily	Dagpatroon optimalisatie	Schema aanpassing
Wekelijks	adlar_cop_weekly	Trend identificatie	Strategie wijziging
Maandelijks	adlar_cop_monthly	Seizoensaanpassing	Parametertuning
Seizoens	adlar_scop	Jaarlijkse evaluatie	Systeem beoordeling

Efficiency Zones

Classificatie (alle horizons):

Zone COP Bereik Kleuren Actie			
- - - - - - - - - - -			
Excellent ≥ 4.0  Groen Maintain, perfect!			
Good 3.5 - 4.0  Groen Maintain			
Acceptable 3.0 - 3.5  Geel Monitor			
Poor 2.5 - 3.0  Oranje Optimize!			
Bad < 2.5  Rood Urgent optimize!			

Multi-Horizon Beslissingslogica

De COP controller analyseert prestaties op basis van 5 tijdshorizons en neemt

beslissingen met verschillende urgentieniveaus:

Analyse stappen:

1. **Verzamel COP data** - Current, daily, weekly, monthly, seasonal (SCOP)
2. **Bereken trends** - Short-term (current vs daily), medium-term (daily vs weekly), long-term (weekly vs monthly)
3. **Bepaal actie** op basis van meerdere signalen:
4. **URGENT** (current < 2.5) → Immediate optimize (-2.0°C)
5. **TRENDING DOWN** (dalende trend op meerdere horizons) → Preventive optimize (-1.0°C)
6. **SEASONAL** (SCOP < 3.2) → Seasonal adjustment
7. **EXCELLENT** (current ≥ 4.0 , daily ≥ 3.8) → Maintain



Technische Implementatie: Voor volledige multi-horizon COP controller, trend analyse en beslisbomen, zie [Adaptieve_Regeling_Technische_Implementatie.md](#) - Component 4: COP Controller

COP Capabilities Gebruiken

Voordelen van Ingebouwde Metingen

- ✓ Waarom Adlar COP capabilities beter zijn:

- X OUDE AANPAK: Zelf berekenen
 - Complex formules (massflow, ΔT , etc.)
 - Schattingen voor onbekende variabelen
 - Potentiële fouten in berekening
 - Alleen realtime waarde

- ✓ NIEUWE AANPAK: Gebruik ingebouwd
 - Warmtepomp kent werkelijke **parameters**
 - Fabrikant *algoritmes (nauwkeurig)*
 - Meerdere *tijdshorizons beschikbaar*
 - SCOP *conform EN 14825 standaard*

Praktische Implementatie

Het systeem maakt gebruik van de ingebouwde Adlar COP capabilities zonder eigen berekeningen:

- **getCurrentCOPMetrics()** - Haalt alle 5 COP metingen op (current, daily, weekly, monthly, seasonal)
- **evaluatePerformance()** - Analyseert prestaties en neemt actie op basis van verschillende horizons:
 - Directe optimalisatie bij current < 2.5
 - Trend analyse bij dalend wekelijks gemiddelde
 - Seizoensrapportage aan einde van verwarmingsseizoen

Voordeel: Geen complexe COP berekeningen nodig - de warmtepomp levert nauwkeurige waarden op basis van fabrikant algoritmes conform EN 14825 standaard.

Praktisch Voorbeeld - Multi-Horizon COP Analyse

DONDERDAG OCHTEND – COP TREND ANALYSE

08:00 – COP Capabilities Uitlezen

- Huidige COP: 2.4  (Slecht!)
- 24u gemiddelde: 2.8  (Matig)
- 7d gemiddelde: 3.2  (Acceptabel)
- 30d gemiddelde: 3.6  (Goed)
- SCOP seizoen: 3.4  (Redelijk)

08:05 – Trend Analyse

- Korte termijn: $2.4 - 2.8 = -0.4$ (dalend vandaag!)
- Medium termijn: $2.8 - 3.2 = -0.4$ (dalende week)
- Lange termijn: $3.2 - 3.6 = -0.4$ (dalende maand)
- Conclusie:  Consistent dalende trend!

08:05 – Beslissing

- Trigger: Trend Alert (alle horizons dalen)
- Actie: Immediate optimization
- Aanpassing: Steltemperatuur -1.5°C
- Doel: Lagere aanvoertemperatuur voor betere COP
- Verwachte verbetering: $2.4 \rightarrow 3.2$ COP

08:20 – Eerste Check (na aanpassing)

- Huidige COP: 2.7  (Verbeterd al!)
- Binnentemperatuur: 19.8°C (nog acceptabel)
- Trend: +0.3 COP in 15 min
- Status: ✓ Goede richting

09:00 – Verificatie

- Huidige COP: 3.1  (Acceptabel)
- Verbetering: +0.7 COP (+29%)
- Binnentemp: 19.9°C (Perfect!)
- Daggemiddelde impact: nog te vroeg voor update
- Status: ✓ Optimalisatie succesvol

END OF DAY – 24u Update

- Nieuwe 24u gemiddelde: 3.0 (was 2.8)

- Verbetering: +0.2 daggemiddelde
- Geschatte besparing: €1.20/dag
- Maandelijks effect: Wordt gemonitord
- Volgende check: Morgen 08:00

Voordelen Multi-Horizon Aanpak:

- Horizon | Oude Aanpak | Nieuwe Aanpak | Voordeel |
 - Realtime | Complexe berekening | adlar_cop direct | Betrouwbaar |
 - Dagelijks | Handmatige averaging | adlar_cop_daily | Gewogen op looptijd |
 - Trends | Geen historische data | Meerdere tijdsschalen | Trend detectie |
 - Seizoens | Niet beschikbaar | adlar_scop (EN 14825) | Standaard conform |
 - Onderhoud | Code onderhoud | Geen onderhoud | Maintenance-free |
 - COP: 2.8 (slecht)

Historische analyse:

- Bij 5°C buiten is optimaal: 35°C
- Verwachte COP bij 35°C: 4.0
- Potentieel: +43% efficiënter!

Actie:

- Verlaag steltemperatuur om 35°C aan te houden

###💡 Praktisch Voorbeeld

|||| DONDERDAG OCHTEND - COP OPTIMALISATIE |||

- 08:00 - Initiële Detectie
- Meting: |
 - Buiten: 5°C |
 - Aanvoer: 45°C
 - ⚠ |
 - Retour: 40°C

- Vermogen: 2.5 kW |
- Binnen: 20.2°C |
- Berekening: |

ΔT: 5°C | Thermisch: ~6.5 kW | COP: 6.5 / 2.5 = 2.6  SLECHT!
| Zone: 'poor' | Analyse: | Historisch optimaal bij 5°C: 35°C
| Huidig: 45°C | Verschil: 10°C TE HOOG! | Besluit:
Actie: Verlaag aanvoer | Target: Van 21°C → 19°C steltemp WP | Effect:
Aanvoer daalt geleidelijk | Verwachting: COP → 3.8 (+46%)

08:20 - Tussentijdse Check

- Aanvoer: 42°C (daalt ✓)
- COP: 3.0 (verbeterd ✓)
- Binnen: 20.0°C (nog OK ✓)

09:00 - Verificatie

- Nieuwe meting:
 - Aanvoer: 37°C ✓
 - Vermogen: 2.2 kW (minder nodig!)
 - COP: 7.7 / 2.2 = 3.5  GOED!
 - Binnen: 19.8°C (acceptabel)
- Resultaat:
 - COP: 2.6 → 3.5 (+35%)
 - Vermogen: -0.3 kW
 - Besparing: €0.075/uur
 - Per dag: €1.80
 - Per jaar: ~€660! 

09:20 - Fijn-tuning

- Binnen iets te laag (19.8°C)
- Actie: Steltemp +0.3°C
- Aanvoer: 37°C → 38°C
- COP: 3.5 → 3.4 (minimaal effect)
- Binnen: 20.0°C ✓ Perfect!

RESULTAAT:

- Gevonden optimum: 38°C aanvoer bij 5°C buiten
- COP stabiel op 3.4–3.5
- Comfort gewaarborgd
- Jaarlijkse besparing door COP optim: €600–700

Configuratie Parameters

Parameter	Default	Bereik	Beschrijving
Min Acceptable COP	2.5	1.5 - 4.0	Onder deze waarde: actie
Target COP	3.5	2.0 - 5.0	Doel COP
Strategy	Balanced	Conservative / Balanced / Aggressive	Snel
Prioritize Efficiency	false	true / false	COP boven comfort?
Min Supply Temp	25°C	20 - 35°C	Nooit lager
Max Supply Temp	55°C	45 - 65°C	Nooit hoger

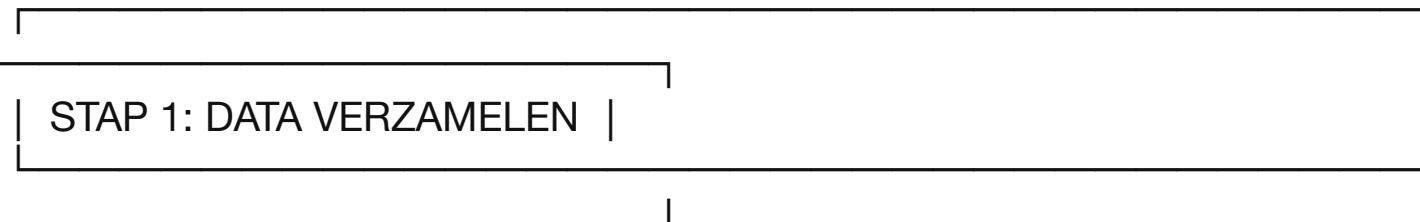
Strategy Details:

- ****Conservative****: Kleine aanpassingen (-1°C per keer), 30 min wachten
- ****Balanced****: Matige aanpassingen (-2°C), 20 min wachten (aanbevolen)
- ****Aggressive****: Grote aanpassingen (-3°C), 15 min wachten

Systeem Integratie

Complete Control Loop

Elke 5 Minuten:



```
const reading = { timestamp: Date.now(), indoor: getExternalSensor(), //  
19.8°C outdoor: getDPS26(), // 5.0°C supply: getDPS21(), // 38°C return:  
getDPS22(), // 33°C power: getExternalPower(), // 2000W
```

```
target: getCurrentTarget(), // 20.0°C compressorFreq: getDPS20() // 45 Hz };
```

| STAP 2: UPDATE CONTROLLERS |

```
// A) Heating Controller
heatingController.addReading(reading);

// B) Building Learner
buildingLearner.addMeasurement({
  T_indoor: 19.8,
  T_outdoor: 5.0,
  P_heating: calculateThermalPower(reading), // 7kW
  solarRadiation: getSolarAPI() // 200 W/m2
});

// C) COP Controller - Gebruik ingebouwde capabilities
const copMetrics = {
  current: getCapabilityValue('adlar_cop'), // 3.5
  daily: getCapabilityValue('adlar_cop_daily'), // 3.2
  weekly: getCapabilityValue('adlar_cop_weekly'), // 3.4
  monthly: getCapabilityValue('adlar_cop_monthly'), // 3.6
  seasonal: getCapabilityValue('adlar_scop') // 3.5
};
copController.updateMetrics(copMetrics);

// D) Energy Optimizer (elk uur)
energyOptimizer.updatePrices();
```

| STAP 3: ADVIES VRAGEN |

```
// Elke controller geeft advies
const tempAdvice = heatingController.calculateAction();
// → { action: 'increase', magnitude: 1.2°C }
```

```
const copAdvice = copController.calculateAction();
// → { action: 'maintain', COP is goed }
```

```
const priceAdvice = energyOptimizer.calculateAction();
// → { action: 'maintain', prijs normaal }
```

| STAP 4: GEWOGEN BESLISSING |

```
// Prioriteiten:
// 60% Comfort (altijd belangrijk)
// 25% COP (binnen comfort grenzen)
// 15% Prijs (binnen comfort + COP grenzen)
```

```
totalAdjust =
tempAdvice.magnitude × 0.60 // 1.2 × 0.6 = 0.72
copAdvice.adjustment × 0.25 // 0 × 0.25 = 0
priceAdvice.adjustment × 0.15; // 0 × 0.15 = 0
```

```
// Totaal: +0.72°C
```

| STAP 5: UITVOEREN |

```
currentSetpoint = getTargetTemperature() // 20.0°C
newSetpoint = currentSetpoint + totalAdjust; // 20.72°C
```

```
// Afronden en limiteren
newSetpoint = Math.round(newSetpoint * 10) / 10; // 20.7°C
newSetpoint = clamp(newSetpoint, 15, 28); // Safety

// Significante wijziging?
if (Math.abs(newSetpoint - currentSetpoint) >= 0.1) {
// Ja! Uitvoeren
setDPS4(newSetpoint); // DPS 4 = steltemperatuur

log(Steltemp: ${currentSetpoint}°C → ${newSetpoint}°C);
log(Reden: ${tempAdvice.reason});

// Trigger Homey Flow
triggerFlow('setpoint_adjusted', {
old: currentSetpoint,
new: newSetpoint,
reason: tempAdvice.reason
});
}
```

| WACHT 5 MINUTEN... HERHAAL |

 Prioriteiten & Conflicten

Hoe worden Conflicten Opgelost?

|| SCENARIO: CONFLICTERENDE ADVIEZEN ||



Temp Controller: "Verhoog +2°C" (te koud!)

COP Controller: "Verlaag -1°C" (slechte COP)

Price Optimizer: "Verlaag -1°C" (dure prijs)

GEWOGEN BEREKENING:

— Temp: $+2 \times 0.60 = +1.20^\circ\text{C}$

— COP: $-1 \times 0.25 = -0.25^\circ\text{C}$

— Prijs: $-1 \times 0.15 = -0.15^\circ\text{C}$

— TOTAAL: $+1.20 - 0.25 - 0.15 = +0.80^\circ\text{C}$

BESLUIT: Verhoog met $+0.8^\circ\text{C}$

REDEN: Comfort is prioriteit, maar meer gematigd door COP en prijs overwegingen

****Priority Override Regels:****

1. **Safety First:**

```
if (indoor < 15°C || indoor > 28°C) {  
    // Negeer alles, herstel naar veilig bereik  
    override = true;  
}  
``
```

2. ****Comfort Minimum:****

```
if (indoor < target - maxComfortDeviation) {  
    // COP en prijs worden genegeerd  
    weights = [1.0, 0.0, 0.0]; // 100% comfort  
}
```

3. **Efficiency Opportunity:**

```
if (cop < minAcceptableCOP && indoor > target) {  
    // Extra gewicht op COP optimalisatie  
    weights = [0.40, 0.50, 0.10];  
}
```

4. **Price Emergency:**

```
if (price > extremeThreshold && indoor > target - 0.5) {  
    // Extra gewicht op kosten  
    weights = [0.50, 0.20, 0.30];  
}
```

Setup & Configuration

Installatie Checklist

****Pre-requirements:****

- [] Homey Pro (Early 2023 of nieuwer)
- [] Adlar Castra Aurora II warmtepomp
- [] Externe temperatuur sensor (thermostaat/sensor)
- [] Vermogen meting werkt (`adlar_external_power`)
- [] Homey App development environment

****Stap 1: App Installatie (5 min)****

bash

Clone repository

git clone [repo-url]

cd homey-adlar-app

Install dependencies

npm install

Build app

npm run build

Deploy to Homey

homey app install

Stap 2: Basis Configuratie (10 min)

1. **Open Homey App** → Devices → Adlar Heat Pump

2. **Configureer Externe Sensor:**

Settings → External Temperature Sensor

└ **Device:** [Selecteer je thermostaat]

└ **Capability:** measure_temperature

3. **Verificeer Stooklijn:**

App zet automatisch stooklijn op OFF

Controleer: Settings → Control Mode

Status moet zijn: "Direct Setpoint Control ✓"

4. **Stel Gewenste Temperatuur In:**

Settings → Adaptive Temperature Control

└ **Target Temperature:** 20°C (gewenste binnentemp)

└ **Deadband:** 0.3°C (aanbevolen)

Stap 3: Eerste 24 Uur – Learning Phase

DAG 1 - OBSERVATIE MODUS

└ Adaptive Control: UIT laten (handmatig)

└ App verzamelt data (elke 5 min)

└ Building model start leren

└ Na 24u: ~30% confidence

Wat gebeurt er? └ Temperaturen worden gelogd └ COP wordt berekend en opgeslagen └ Building parameters krijgen eerste schatting

└ Je ziet na een automatische aanpassingen

Stap 4: Week 1 – Adaptieve Regeling Activeren

DAG 3-7 - BASIS REGELING

- Settings → Enable Adaptive Control ✓
- Systeem past nu automatisch steltemp aan
- Building model: 50–70% confidence
- Je ziet stabile temperatuur

Verwacht gedrag:

- Kleine aanpassingen ($\pm 0.5^{\circ}\text{C}$) elk uur
- Binnentemp binnen $\pm 0.3^{\circ}\text{C}$ van doel
- Notificaties bij grote aanpassingen
- Flow kaarten beginnen te werken

Stap 5: Week 2+ – Geavanceerde Features

DAG 10+ - VOLLEDIGE OPTIMALISATIE

- Settings → Enable Energy Optimization ✓
- Settings → Enable COP Control ✓
- Building model: 70–90% confidence
- Alle systemen operationeel

API Configuratie:

- Energy Source: EnergyZero (gratis)
- Of: ENTSO-E (vereist gratis API key)
- Prijs thresholds: Standaard OK, pas aan indien gewenst

```
### ⚙ Configuratie Parameters
```

```
#### Heating Controller
```

```
json { "adaptive_control_enabled": false, // Start: false, na 24u: true
"target_temperature": 20, // Gewenste binnentemp "control_deadband": 0.3, //
±0.3°C tolerantie
```

```
“control_kp”: 3.0, // P-versterking “control_ki”: 1.5, // I-versterking
“control_time_window”: 2, // Uren voor I-term “min_wait_between_changes”:
20, // Minuten wachttijd “adaptive_mode”: true, // Outdoor compensatie
“night_mode_enabled”: false, // Nachtmodus “night_start_hour”: 22,
“night_end_hour”: 6, “night_target_temp”: 18 }
```

```
#### Energy Optimizer
```

json

```
{ “energy_optimization_enabled”: false, // Start: false, na week: true
“energy_api_source”: “energyzero”, // Of “entsoe”
“entsoe_api_key”: “”, // Als ENTSO-E
“price_very_low”: 0.10, // €/kWh
“price_low”: 0.15,
“price_normal”: 0.25,
“price_high”: 0.35,
“price_very_high”: 0.45,
“max_preheat_offset”: 1.5, // Max +1.5°C voorverwarmen
“max_cost_saving_offset”: 1.0 // Max –1.0°C reduceren
}
```

```
#### COP Controller (Multi-Horizon)
```

json

```
{ “cop_control_enabled”: false, // Start: false, na week: true

// Thresholds per horizon (gebruik ingebouwde capabilities)
“min_acceptable_cop_current”: 2.5, // adlar_cop drempel
“min_acceptable_cop_daily”: 2.8, // adlar_cop_daily drempel
“min_acceptable_cop_weekly”: 3.0, // adlar_cop_weekly drempel
“target_scop”: 3.5, // adlar_scop doel (seizoen)

// Trend analysis
“enable_trend_analysis”: true, // Monitor dalende trends
“trend_alert_threshold”: -0.3, // Alert bij -0.3 COP daling
“trend_analysis_days”: 7, // Trend over N dagen
```

```
// Optimization strategy
"cop_adjustment_strategy": "balanced", // conservative/balanced/aggressive
"prioritize_efficiency": false, // COP boven comfort?

// Multi-horizon actions
"immediate_action_threshold": 2.3, // Directe actie bij zeer lage COP
"seasonal_review_enabled": true, // Maandelijkse SCOP evaluatie
"weekly_trend_reporting": true // Wekelijkse trend rapportage
}
```

Building Model

```
json
{ "enable_predictive_control": false, // Vereist 70%+ confidence
  "min_model_confidence": 0.7,
  "reset_building_model": false, // Handmatig reset
  "export_building_model": false // Export voor analyse
}
```

🎯 Tuning Guide

Situatie: Temperatuur Schommelt Te Veel ($\pm 1^{\circ}\text{C}$)

Probleem: Te agressieve regeling

Oplossing:

- └─ Verhoog deadband: 0.3 → 0.5°C
- └─ Verhoog min_wait: 20 → 30 min
- └─ Verlaag Kp: 3.0 → 2.5

Situatie: Temperatuur Reageert Te Traag

Probleem: Te conservatieve regeling Oplossing:

└ Verlaag deadband: 0.3 → 0.2°C └ Verlaag min_wait: 20 → 15 min └
Verhoog Kp: 3.0 → 3.5

Situatie: Structureel Te Laag/Hoog

Probleem: I-term niet effectief

Oplossing:

└ Verhoog Ki: 1.5 → 2.0
└ Vergroot time_window: 2 → 3 uur

Situatie: Model Confidence Blijft Laag

Probleem: Inconsistente data of grote variaties

Oplossing:

└ Check externe sensor positionering
└ Controleer adlar_external_power data
└ Wacht langer (2–3 weken)
└ Reset model en herstart

```
## Flow Voorbeelden
```

```
### 🔥 Basis Flows
```

```
#### Flow 1: Eenvoudige Notificatie
```

NAAM: “Temperatuur Aangepast - Notificatie”

WHEN: Setpoint automatically adjusted

THEN:

Send notification:

“🌡️ Steltemp: {{old_setpoint}}°C → {{new_setpoint}}°C
Reden: {{reason}}”

```
#### Flow 2: Dagelijkse Status Update
```

NAAM: “Dagelijkse Model Status”

WHEN: Time is 08:00

IF: Building model confidence is above 50%

THEN:

Get building model status

Send notification:

“ Model Update”

Betrouwbaarheid: {{confidence}}%

Thermische massa: {{thermal_mass}} kWh/°C

Warmteverlies: {{heat_loss}} W/°C

Tijdsconstante: {{time_constant}} uur”

```
### 🌙 Geavanceerde Flows
```

```
#### Flow 3: Slimme Wake-up (Weekdagen)
```

NAAM: “Weekdag Wake-up Pre-heat”

WHEN: Time is 05:00

IF:

- Day of week is Monday-Friday
- Building model confidence > 60%
- Building responds slowly ($\tau > 6h$)

THEN:

Calculate pre-heat time to 21°C by 07:00

IF time_needed < 2 hours:

- Set adaptive target to 21°C
- Send notification: “ Pre-heating voor wakker worden”

ELSE IF time_needed > 2 hours:

- Set adaptive target to 21°C NOW
- Increase setpoint by 1°C (extra boost)
- Send notification: “ Extra vroeg starten ({{time_needed}}h nodig)”

```
#### Flow 4: GPS-Based Thuiskomst
```

NAAM: “GPS Thuiskomst Detectie”

WHEN: Someone is arriving home (Geofencing)

IF:

- Everyone was away
- Current temp < 19°C

THEN:

Calculate time to reach 20°C

Variable: time_needed = {{time_needed}}

IF time_needed < 0.5 hours:

- Wait for arrival (snel genoeg)

ELSE:

- Set adaptive target to 20°C
- Increase heating curve (TEMP boost)
- Send notification: “ Opwarmen gestart, klaar over {{time_needed}}u”

NAAM: “GPS Verlaten Huis”

WHEN: Everyone left home

THEN:

- Set adaptive target to 17°C (Eco)
- Send notification: “ Eco-modus actief (17°C)”
- Log: “Eco mode started at {{time}}”

```
#### Flow 5: Weer-gebaseerde Anticipatie
```

NAAM: “Weersvoorspelling Anticipatie”

WHEN: Weather forecast updated

IF:

- Tomorrow's min temp < Today's min by > 5°C
- Building model confidence > 70%

THEN:

Predict temperature tomorrow 08:00

IF predicted_temp < target - 1°C:

- Tonight 22:00: Increase setpoint by 1°C

- Send notification:

“❄️ Morgen veel kouder (<{{tomorrow_min}}°C)

Extra voorverwarming gepland”

NAAM: “Zon Voorspelling”

WHEN: Time is 09:00

IF:

- Solar radiation forecast > 600 W/m² for next 4h

- Current indoor > target - 0.3°C

THEN:

- Decrease setpoint by 0.5°C preventief

- Send notification:

“☀️ Zonnige middag verwacht

Verwarming preventief verlaagd”

```
#### Flow 6: Energie Prijs Optimalisatie
```

NAAM: “Nachtelijke Pre-heat (Goedkoop)”

WHEN: Time is 23:00

IF:

- Current electricity price < €0.12/kWh

- Tomorrow 07:00 price > €0.35/kWh

- Building time constant > 6 hours

- Current temp < target + 1.5°C

THEN:

Get building model status

Calculate: How much pre-heat is optimal?

— Can building retain heat until 09:00?

— What's the cost saving?

IF estimated_savings > €0.50/day:

- Set adaptive target to current + 1.5°C

- Send notification:

“⚡️💰 Pre-heating bij lage prijs (€{{current_price}})

Geschatte besparing: €{{estimated_savings}}/dag”

ELSE:

- Log: “Pre-heat not worth it (€{{estimated_savings}})”

NAAM: “Dure Piek Uur Reductie”

WHEN: Electricity price rises above €0.40/kWh

IF:

- Current indoor > target - 1.0°C

- No guests present

THEN:

- Set adaptive target to current - 1.0°C (max reductie)

- Send notification:

“💰 Duur uur! Tijdelijk gereduceerd naar {{new_target}}°C

Prijs: €{{current_price}}/kWh”

```
#### Flow 7: Multi-Horizon COP Monitoring & Alerts
```

NAAM: “Multi-Horizon COP Analyse”

WHEN: Every hour

THEN:

Get COP capabilities:

- Current: {{adlar_cop}}

- Daily: {{adlar_cop_daily}}

- Weekly: {{adlar_cop_weekly}}

- Monthly: {{adlar_cop_monthly}}

- Seasonal: {{adlar_scop}}

// REALTIME ALERT

IF current < 2.3: Send notification: “ Acute lage COP: {{adlar_cop}} Directe actie vereist!”

// TREND ALERT

IF (current - weekly) < -0.4:

Send notification:

“ COP dalende trend gedetecteerd

Week gemiddelde: {{adlar_cop_weekly}}

Nu: {{adlar_cop}}

Verschil: {{trend_change}}”

// SEASONAL REVIEW

IF is_end_of_month AND adlar_scop < 3.2:

Send notification:

“ Maandelijks SCOP rapport

SCOP: {{adlar_scop}} (doel: 3.5+)

Actie: Seizoensaanpassing overwegen”

NAAM: “Comprehensive COP Rapport”

WHEN: Time is Sunday 20:00

THEN:

Create multi-horizon report:

Send notification:

“ Wekelijks COP Rapport

 Realtime prestatie:

Huidige COP: {{adlar_cop}}

Status: {{performance_rating}}

 Trends:

24u gemiddelde: {{adlar_cop_daily}}

7d gemiddelde: {{adlar_cop_weekly}}

30d gemiddelde: {{adlar_cop_monthly}}

Trend: {{trend_direction}} ({{trend_magnitude}})



Seizoenprestatie:
SCOP: {{adlar_scop}} (EN 14825)
Doelstelling: {{scop_vs_target}}



Impact:
Geschatte kosten deze week: €{{estimated_cost}}
vs vorige week: {{cost_comparison}}



Aanbevelingen:
{{optimization_suggestions}}

```
"📊 Week Rapport
Gemiddelde COP: {{avg_cop}}
Elektrisch verbruik: {{kwh_used}} kWh
Warmte geleverd: {{heat_delivered}} kWh
Effectieve kosten: €{{cost}}"
```

Flow 8: Afwezigheid Scenario's

NAAM: "Weekend Weg – Eco Mode"

WHEN: User manually activates "Weekend Away" scene

THEN:

- Set adaptive target **to** 15°C (Vorst bescherming)
- Disable night mode
- Send notification:
"🏖 Weekend mode actief
Min temp: 15°C (vorst bescherming)"
- Store original **settings**

NAAM: "Weekend Terug – Restore"

WHEN: User manually activates "Back Home" scene

THEN:

- Restore original target temperature
- Re-enable night mode **if** was active
- Calculate time **to** reach comfort temp
- Send notification:
"🏡 Welkom terug!
Opwarmen naar {{target}}°C duurt {{time}}u"

Monitoring & Logging Flows

Flow 9: Data Export voor Analyse

NAAM: "Maandelijkse Data Export"

WHEN: First day of month at 01:00

THEN:

Get building model diagnostics
Get COP statistics for past month
Get energy price statistics

Create CSV with:

- Date, Hour
- Indoor Temp, Outdoor Temp
- Setpoint, COP
- Electricity Price
- Cost, Savings

Save to: /userdata/exports/{{month}}-data.csv

Send notification:

" Maand {{month}} Data Geëxporteerd
Download via Homey Files"

Flow 10: Probleem Detectie

NAAM: "Detecteer Afwijkend Gedrag"

WHEN: Every hour

IF:

- Temperature error > 2°C for 3+ hours

OR

- COP < 2.0 for 2+ hours

OR

- Setpoint changed >5x in last hour

THEN:

Send notification:

"⚠ Mogelijk probleem gedetecteerd
{issue_description}"

Check:

- Externe sensor verbinding
- Warmtepomp status
- Building model confidence"

Disable adaptive control (veiligheid)

Log to timeline: "System issue detected"

Troubleshooting

? Veelvoorkomende Problemen

Probleem 1: "Heating curve not OFF" Warnings

Symptomen:

- Logs tonen: "WARNING: Heating curve not OFF"
- Automatische aanpassingen werken niet

Oorzaken: 1. Handmatig stooklijn aangepast via WP bediening 2. App crash tijdens initialisatie

3. Tuya connectie problemen

Oplossing:

Stap 1: **Check** huidige status

└ Homey App → Device → Settings

└ Control **Mode** moet "Direct Setpoint ✓" tonen

Stap 2: Manueel resetten

└ Via Homey: Devices → Adlar → Advanced Settings

└ **Set** "heating_curve_mode" = "OFF"

└ **Restart** device

Stap 3: Verificatie

└ **Check** logs: "Direct setpoint control mode enabled"

└ **Test**: Pas target temp aan, moet werken

BELANGRIJK: Verander NOOIT handmatig stooklijn via WP!

Probleem 2: Temperatuur Reageert Niet

Symptomen:

- Setpoint changes worden gelogd
- Maar binnentemperatuur verandert niet

Diagnose Checklist:

- ✓ 1. Externe Sensor Werkt?
 - └ **Check:** Settings → **External Temp Sensor**
 - Device selected? ✓
 - Receiving updates? (**check logs**)

- ✓ 2. WP Reageert op DPS 4?
 - └ **Test:** Handmatig target_temperature aanpassen
 - WP aanvoer **temp** verandert? ✓

- ✓ 3. Vermogen Meting Werkt?
 - └ **Check:** adlar_external_power capability
 - Waarde > 0 wanneer WP aan? ✓

- ✓ 4. WP in Juiste Modus?
 - └ **Check:** **Work mode** = "heating" (niet cooling!)

- ✓ 5. Deadband Te Groot?
 - └ **Settings:** control_deadband
 - Probeer: 0.2°C (smaller = more responsive)

Specifieke Oplossingen:

- A) Externe Sensor Issues:
 - └ Re-configureer sensor in settings
 - └ Check sensor batterij/verbinding
 - └ **Test:** Handmatig temperatuur triggeren

- B) WP Communicatie Issues:
 - └ Check Tuya connectie (cloud vs local)
 - └ Restart Homey app
 - └ Check WP netwerk verbinding

- C) Regelaar Te Conservatief:
 - └ Verlaag **min_wait_between_changes**: 20 → 15 min
 - └ Verhoog **Kp**: 3.0 → 3.5
 - └ Verlaag **deadband**: 0.3 → 0.2°C

Probleem 3: Model Confidence Blijft Laag

Symptomen:

- Na 1+ week nog steeds <50% confidence
- Voorspellingen zijn inaccuraat
- “Insufficient data” messages

Diagnose:

1. **Check** Data Kwaliteit
 - Logs → **Search** "Invalid measurement"
 - Zijn er veel fouten?
 - Welke data **is** inconsistent?
2. **Analyse** Data Gaps
 - Settings → Export building model
 - Check** measurement count
 - Verwacht: **288**/dag × dagen = expected count
 - Te laag? → Data collection issues
3. **Check** Sensor Stabiliteit
 - Is** externe **temp** sensor stabiel?
 - Veel ruis/spikes?
 - Relocate sensor **if** nodig

Oplossingen:

- A) Data Collectie Verbeteren:
 - Ensure WP always on (not eco-off)
 - Check** adlar_external_power availability
 - Verify solar radiation estimation
- B) **Reset & Restart:**
 - Settings → **Reset** Building Model
 - Wait **24h** **for** new baseline
 - Check** confidence daily
- C) **Parameter** Aanpassing:
 - Increase forgetting_factor (slower adaptation)
 - Decrease min_confidence_threshold
 - Manual parameter initialization **if** needed

Probleem 4: Energy Prices Niet Beschikbaar

Symptomen:

- “No price data available” in logs
- Price-based flows not triggering

Diagnose:

1. Check API Source
 - └ Settings → Energy Optimization
 - ├ Source: energyzero / entsoe
 - └ API key (if entsoe): configured?
2. Test API Manually
 - └ Browser: <https://api.energyzero.nl/v1/energyprices>
Response should show prices **array**
3. Check Homey Network
 - └ Can Homey reach internet?
Test: Weather updates working?

Oplossingen:

- A) EnergyZero Issues:
 - ├ API down? Check status
 - ├ Rate limited? Wait 1 hour
 - └ Switch **to manual** prices temporarily
- B) ENTSO-E Issues:
 - ├ Invalid API key? Re-register
 - ├ Wrong domain code? Use 10YNL-----L **for** NL
 - └ Request format? Check logs **for** error details
- C) Fallback **to** Manual:
 - ├ **Settings** → `energy_api_source = "manual"`
 - ├ **Set** price thresholds manually
 - └ Create flows **for** **manual** price input

Probleem 5: COP Capabilities Niet Beschikbaar of Onrealistisch

Symptomen:

- COP capabilities geven geen waarde (null/undefined)
- Zeer onrealistische COP waarden (bijv. >8 of <0.5)
- “COP capability not available” warnings in logs
- COP-based optimizations niet actief

Diagnose:

1. Check Capability Availability
 - adlar_cop: Beschikbaar? ✓
 - adlar_cop_daily: Beschikbaar? ✓
 - adlar_cop_weekly: Beschikbaar? ✓
 - adlar_cop_monthly: Beschikbaar? ✓
 - adlar_scop: Beschikbaar? ✓
2. Check Data Freshness
 - Enable DEBUG logging om te zien:
 - "COP capabilities last updated: X minutes ago"
 - "Current COP: X.X (valid range: 0.8-7.0)"
 - "Daily average: X.X (24h rolling)"
 - "Trend: stable/rising/declining"
3. Verify Heat Pump Operation
 - Compressor state: ON wanneer verwacht?
 - Work mode: "heating" (niet cooling/off)?
 - Actual heating happening? (supply > return temp)

Oplossingen:

A) Capability Niet Beschikbaar:

- Check Homey app versie (recentste?)
- Tuya verbinding OK? (local/cloud)
- Restart Homey app
- Controleer Adlar firmware versie

B) Onrealistische Waardes:

- Probable cause: Heat pump in transitie
- Wait 24h voor stabilisatie
- Check if WP in defrost/startup modus
- Monitor trend over meerdere dagen

C) Capabilities Werken, Maar Geen Optimalisatie:

- Settings → **cop_control_enabled** = true
- Check **min_acceptable_cop** threshold (not too low)
- Verify **indoor > target** (ruimte voor optimalisatie)
- Check **override conditions** (safety/comfort)

D) Fallback Mode:

- Settings → **cop_control_enabled** = false
- Disable COP-based optimizations
- Rely only on temperature + price control
- Manual COP monitoring via capabilities

- A) Vermogen Meting Incorrect:
 - └ Calibrate adlar_external_power
 - └ **Check** if includes auxiliary (pomp, fans)
 - └ May need **external** power meter
- B) Temperature Sensors:
 - └ Verify DPS 21/22 mapping correct
 - └ **Check** sensor accuracy
 - └ Recalibrate if $>2^{\circ}\text{C}$ off
- C) Flow Rate Unknown:
 - └ **Current:** Estimated **from** power
 - └ Better: Measure actual flow rate
 - └ **Update** formula **with real** flow data
- D) Disable COP Control:
 - └ If unfixable data issues
 - └ Settings \rightarrow cop_control_enabled = **false**
 - └ Rely **on** temp + price **only**

Debug Modus

Activeren:

```
# Via Homey CLI
homey app env set DEBUG 1

# Of via Settings (if implemented)
Settings → Advanced → Debug Logging → Enable
```

Wat gebeurt er:

Met `DEBUG=1` krijg je extra logs:

Elke 5 minuten:

```
└─ "Controller status: {error: 0.23, integral: 0.15, ...}"
└─ "RLS Update - Error: 0.045°C/h, Pred: 1.23, Actual: 1.28"
└─ "COP calculation: 2.5kW / 0.7kW = 3.57"
└─ "Price: €0.22 (category: normal), no action"
```

Bij aanpassingen:

```
└─ "Adjusting setpoint: 20.0°C → 20.5°C"
└─ "Reason: te koud 0.45°C, dalende trend"
└─ "Weighted decision: temp=0.3, cop=0.1, price=0"
```

📞 Support & Community

Resources:

- GitHub Issues: [repo-url]/issues
- Tweakers Forum: "Adlår Castra Aurora 2 Warmtepompen"
- Homey Community: community.homey.app

Bug Report Template:

****Probleem:**** Korte beschrijving

****Symptomen:****

- Wat zie je gebeuren?
- Wat verwacht je?

****Logs:****

[Plak relevante logs hier]

****Configuratie:****

- Homey versie:
- App versie:
- Settings: [screenshot of relevant settings]

****Steps to Reproduce:****

1. ...
2. ...

Appendix: Technische Details

A. DPS Mapping Referentie

Complete Adlar Castra Aurora II DPS Table:

DPS	Name	Type	Values/Range	Unit
1	onoff	Boolean	true/false	-
4	target_temperature	Integer	0–60	°C
5	work_mode	Enum	heating/cooling/...	-
13	heating_curve	Enum	OFF/L1-L8/H1-H8	-
20	compressor_strength	Integer	0–120	Hz
21	temp_top	Integer	0–900	0.1°C
22	temp_bottom	Integer	0–900	0.1°C
26	around_temp	Integer	–400–900	0.1°C

EXTERN:

- adlar_external_power: Integer, Watt, Werkelijk vermogen
- External indoor sensor: Via capability, °C, Binnentemperatuur

B. Formules & Constanten

Thermische Berekeningen:

1. Warmte Balans:

$$dT/dt = (1/C) \times [P_{in} - P_{out} + P_{solar} + P_{internal}]$$

Waarbij:

- C: Thermische massa [kWh/°C]
- P_in: Verwarming [kW]
- P_out: UA × (T_in - T_out) [kW]
- UA: Warmteverlies coëfficiënt [kW/°C]

2. COP Berekening:

$$COP = Q_{thermal} / P_{electrical}$$

$$Q_{thermal} = \dot{m} \times c_p \times \Delta T$$

- \dot{m} : Massflow [kg/s]
- c_p : 4.186 kJ/(kg·K) voor water
- ΔT : T_supply - T_return [°C]

3. Carnot Efficiëntie (theoretisch max):

$$COP_{carnot} = T_{hot} / (T_{hot} - T_{cold})$$

Met T in Kelvin:

$$T_K = T_{\circ}C + 273.15$$

Praktische COP \approx 40–50% van Carnot

4. Tijdsconstante:

$$\tau = C / UA \text{ [uur]}$$

Temperatuur na tijd t zonder verwarming:

$$T(t) = T_{outdoor} + (T_{start} - T_{outdoor}) \times e^{(-t/\tau)}$$

PI Controller:

1. Fout Berekening:

$$e(t) = T_{target} - T_{actual}$$

2. Proportionele Term:

$$P(t) = K_p \times e(t)$$

3. Integratieve Term:

$$I(t) = K_i \times (1/n) \times \sum e(t-i) \text{ voor } i=0 \text{ tot } n$$

Waarbij n = samples in integration window

4. Totale Correctie:

$$u(t) = P(t) + I(t)$$

5. Met Limiet:

$$u_{limited} = \text{clamp}(u(t), -u_{max}, +u_{max})$$

RLS Algoritme:

1. Feature Vector:

$$\Phi(k) = [P_{heat}, -(T_{in} - T_{out}), Solar/1000, 1]^T$$

2. Parameter Vector:

$$\Theta(k) = [1/C, UA/C, g/C, P_{int}/C]^T$$

3. Voorspelling:

$$\hat{y}(k) = \Phi(k)^T \times \Theta(k-1)$$

4. Voorspelfout:

$$e(k) = y(k) - \hat{y}(k)$$

5. Gain Vector:

$$K(k) = P(k-1) \times \Phi(k) / (\lambda + \Phi(k)^T \times P(k-1) \times \Phi(k))$$

6. Parameter Update:

$$\Theta(k) = \Theta(k-1) + K(k) \times e(k)$$

7. Covariance Update:

$$P(k) = [P(k-1) - K(k) \times \Phi(k)^T \times P(k-1)] / \lambda$$

Met λ = forgetting factor (0.98)

C. API Referenties

EnergyZero API:

```
GET https://api.energynieuw.nl/v1/energyprices
```

Response:

```
{  
  "Prices": [  
    {  
      "readingDate": "2025-12-10T00:00:00Z",  
      "price": 0.08234, // €/kWh incl. BTW  
      "tariffReturn": 0.05123 // Teruglevering  
    },  
    ...  
  ]  
}
```

Rate Limit: ~100 requests/hour

Update Frequency: 1x per uur voldoende

ENTSO-E Transparency Platform:

```
GET https://web-api.tp.entsoe.eu/api
```

Parameters:

- securityToken: YOUR_API_KEY
- documentType: A44 (day-ahead prices)
- in_Domain: 10YNL-----L (Netherlands)
- out_Domain: 10YNL-----L
- periodStart: YYYYMMDDHHMM
- periodEnd: YYYYMMDDHHMM

Response: XML format

Rate Limit: 400 requests/minute

Free API key: transparency.entsoe.eu

Homey Weather API:

```
const weather = await this.homey.weather.getWeather();
```

Properties:

- temperature: **number** (°C)
- cloudiness: **number** (0-100%)
- humidity: **number** (0-100%)
- pressure: **number** (hPa)
- windSpeed: **number** (m/s)

D. Performance Metrics

Verwachte Prestaties:

Metric	Target	Typical
Temp Stability	±0.3°C	±0.2°C
Response Time	<30 min	15–20 min
Model Confidence (1 week)	>60%	70–75%
Model Confidence (1 month)	>85%	90–95%
COP Improvement	+20%	+25–35%
Energy Cost Reduction	30%	35–45%
Annual Savings	€500	€600–800

Resource Usage:

Resource	Usage
CPU (Homey)	<2% average
Memory	~50 MB
Storage	<5 MB (model data)
Network	<1 MB/day (API calls)
Data Collection	288 samples/day

E. Versie Historie

v1.0 (December 2025)

- Initial release
- PI temperature controller
- Building model learner (RLS)
- Energy price optimization
- COP-based control
- Basic flow cards
- Dutch & English support

Planned Features:

- v1.1: Model Predictive Control (MPC)
 - v1.2: Weather forecast integration
 - v1.3: Multi-zone support
 - v1.4: Advanced analytics dashboard
 - v1.5: Cloud backup & sync
-

Conclusie

Dit systeem combineert vier intelligente controllers om je warmtepomp optimaal te regelen:

1. **Heating Controller** - Houdt temperatuur constant met PI regeling
2. **Building Learner** - Leert eigenschappen van je woning via ML
3. **Energy Optimizer** - Bespaart geld door slimme prijs-optimalisatie
4. **COP Controller** - Maximaliseert efficiëntie binnen comfort grenzen

Verwachte Resultaten: -  Constante binnentemperatuur ($\pm 0.3^\circ\text{C}$) -  30–45% lagere energiekosten -  25–35% betere COP -  Automatische aanpassing aan jouw woning

-  €600–800 besparing per jaar

Succes met je adaptieve warmtepomp regeling!

Voor vragen of support: Zie Troubleshooting sectie