

# Automatic Dimensioning and Efficiency Optimization of Heating Systems

Matteo Ludwig

RWTH Aachen University

# Residential Heating

~25% of national energy consumption

Energy costs rising



Need for increases in efficiency

Subsidies & regulation



**Hydronic balancing**  
**Supply Temperature Reduction**

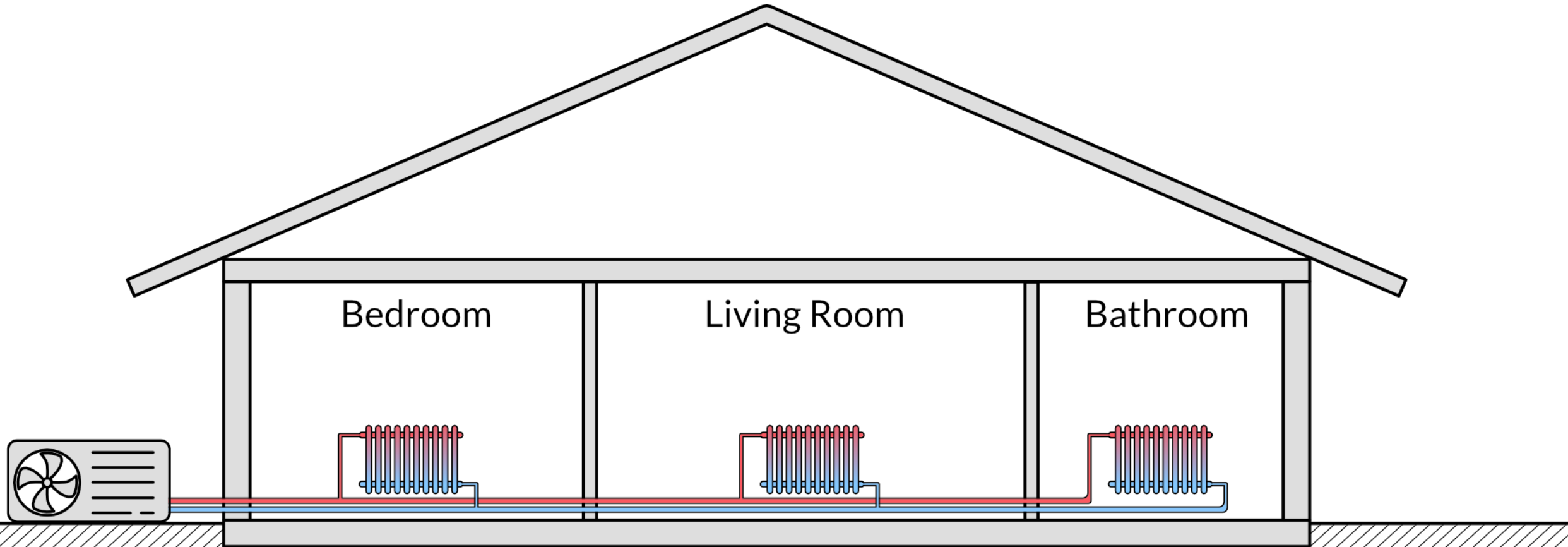
§60c GEG, KfW & BAFA subsidies



Develop Library

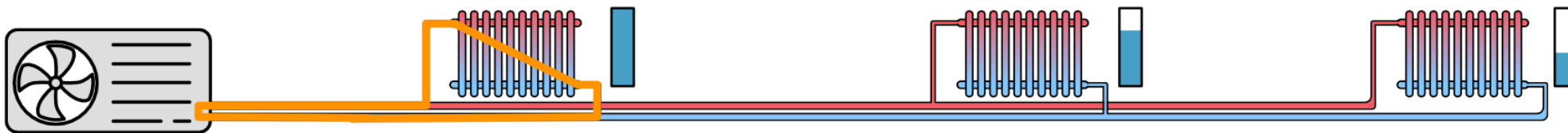
# Hydronic Central Heating

- Hydronic = Heating with water
- Two Pipes: **Supply** & **Return**



# Hydraulic Circuits

- One per emitter
- Pressure loss
  - Pipes & elements cause friction
- Uneven flow distribution



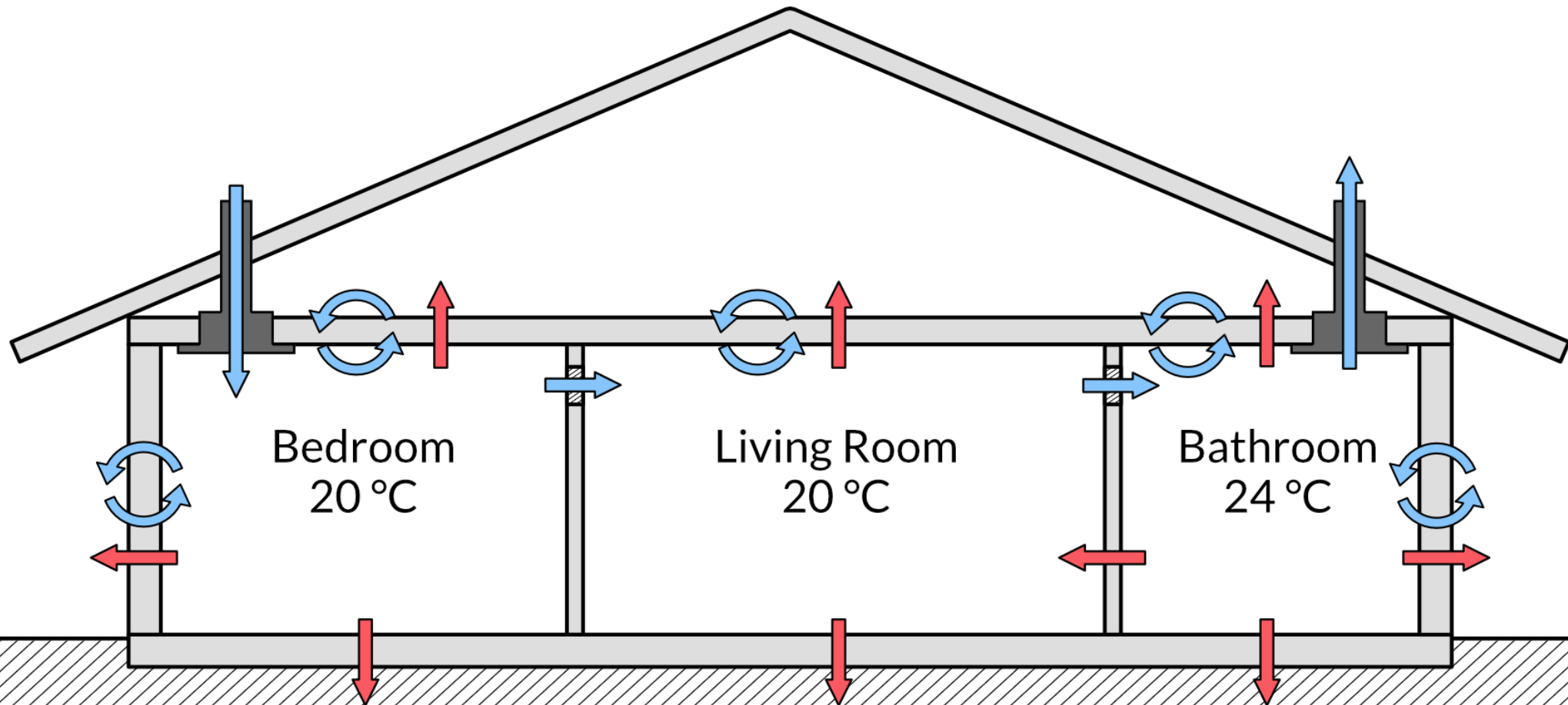
# Radiator Flow Rate

$$f(\vartheta_S, \Phi_N, \Phi) \rightarrow q_m$$

- Mass flow  $q_m$  [kg/s]
- Supply temperature  $\vartheta_S$  [°C]
  - Heat generator
- Nominal power output  $\Phi_N$  [W]
  - Radiator Model
- Required power output  $\Phi$  [W]
  - Heat load

# Heat Load

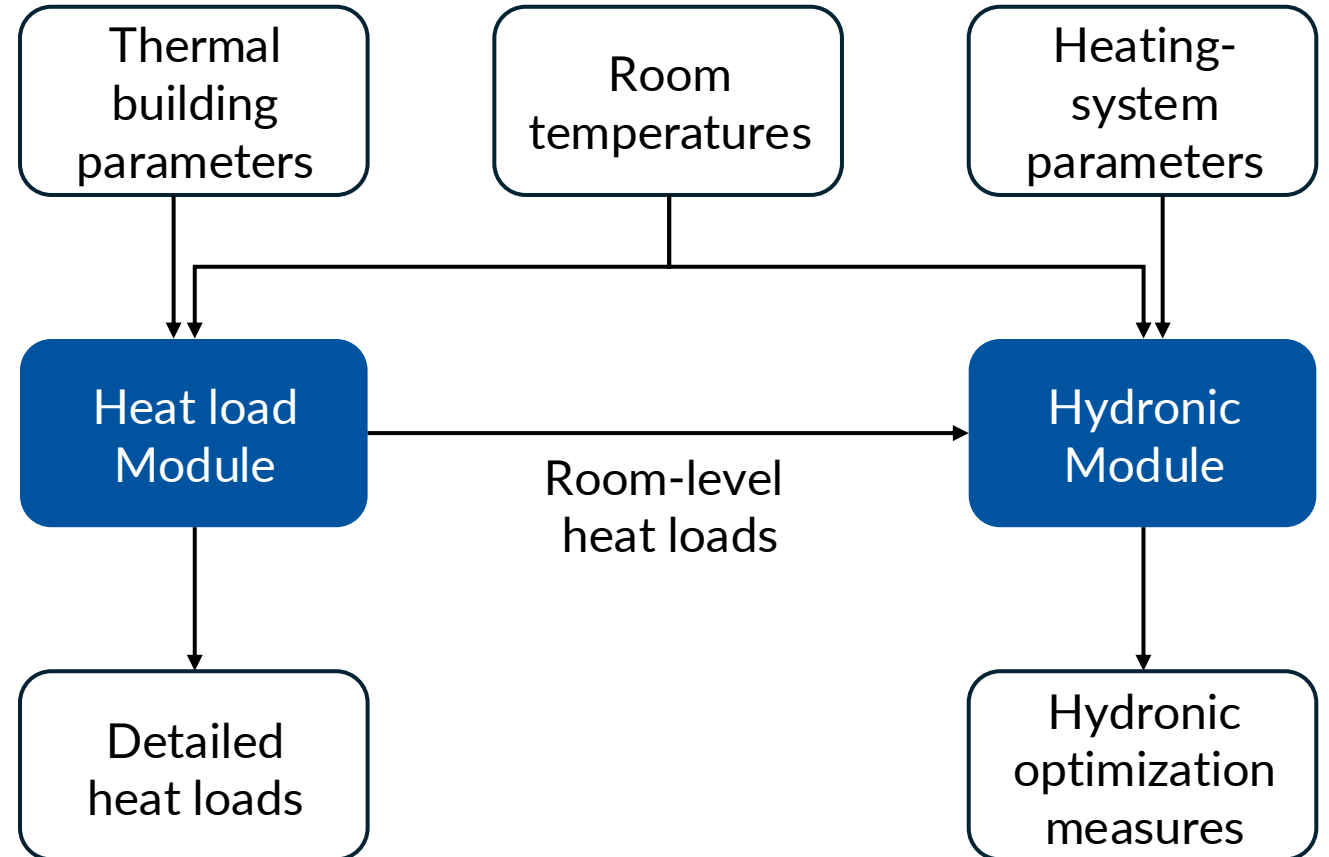
- Heat Loss: **Transmission** & **Ventilation**





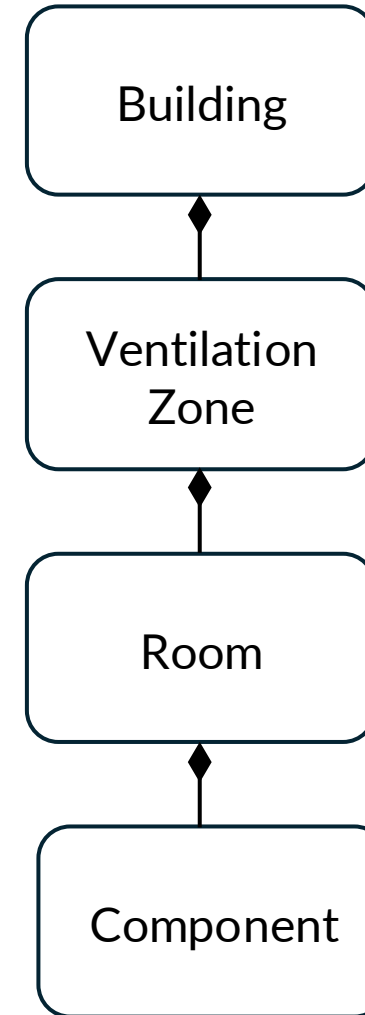
# Library Design

- 3 Problems
  - Heat load calculations
  - Hydronic balancing
  - Supply temperature reduction
- 2 Modules
  - Heat load
  - Hydronic



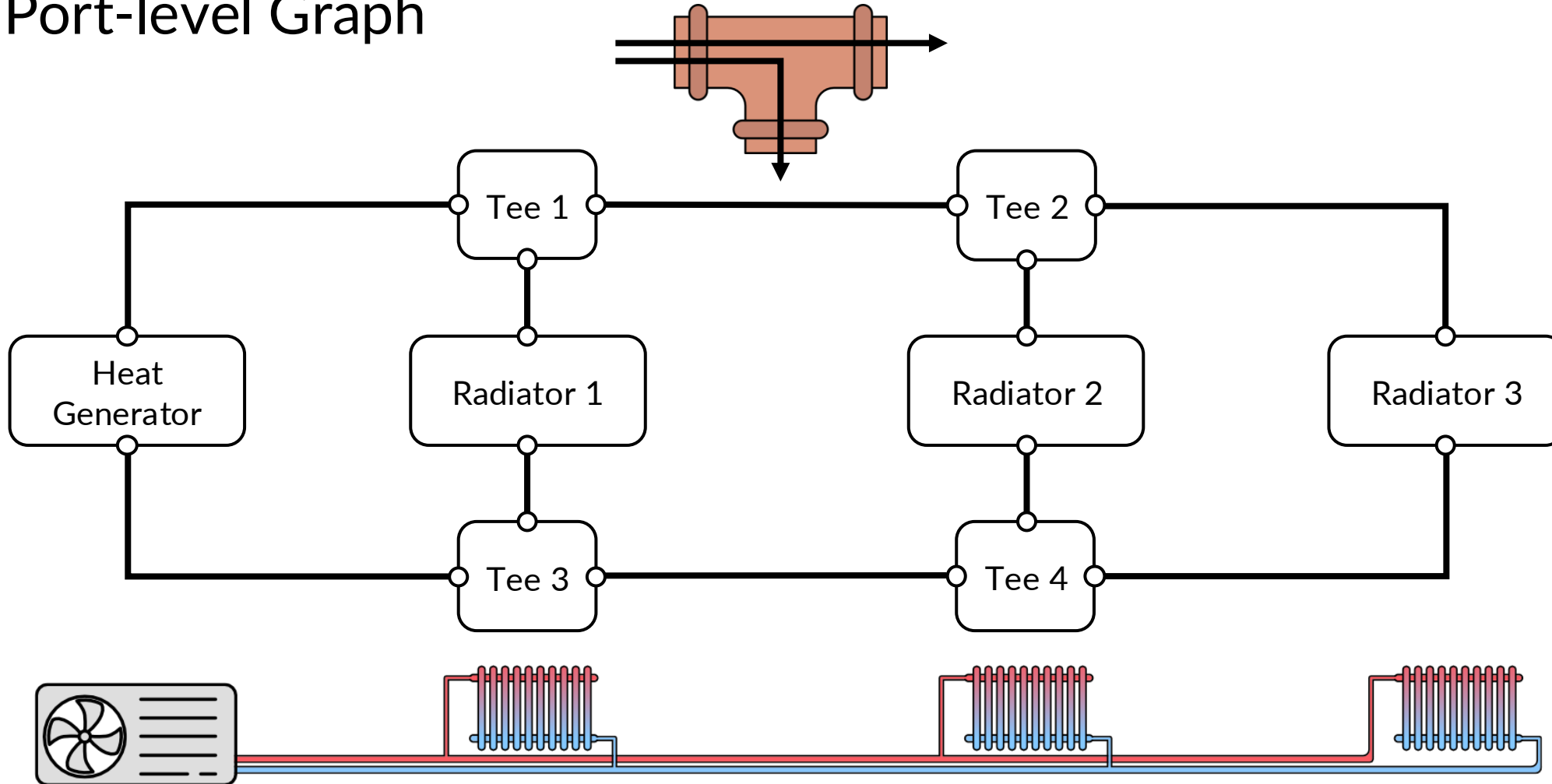
# Heat Load Module

- Implements DIN/TS 12831-1
- 3 Procedures
  - Standard
  - Simplified room-level
  - Simplified building-level



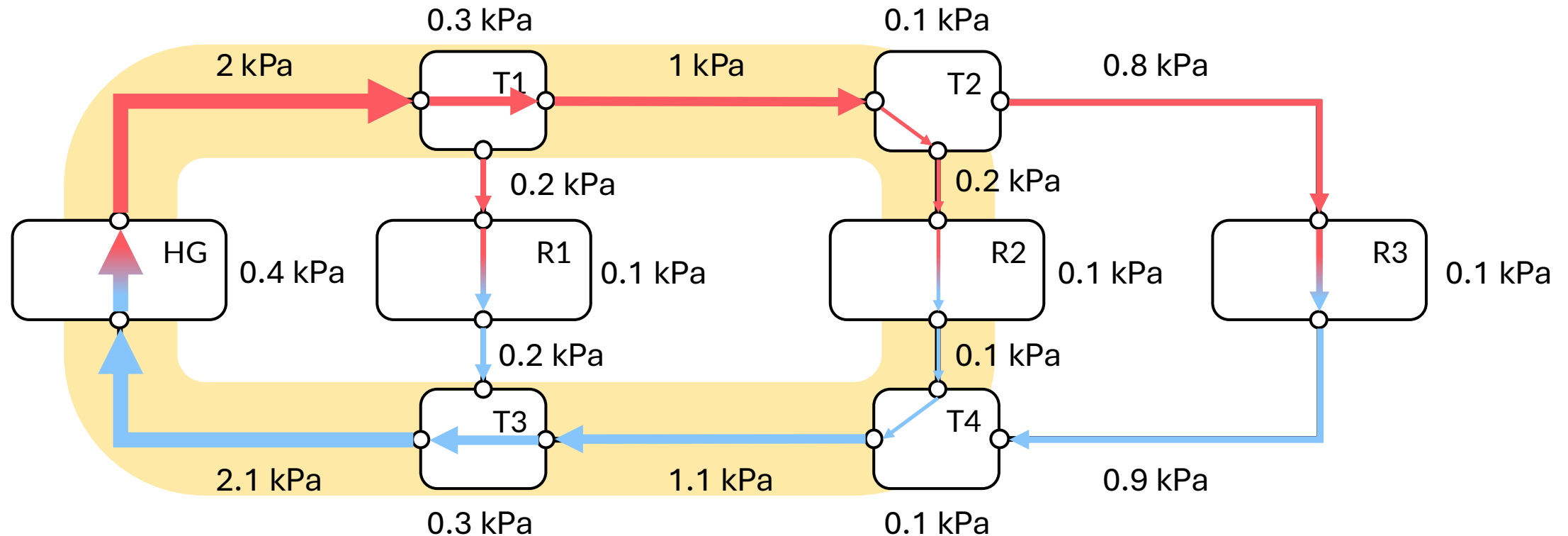
# Hydronic Module

## Port-level Graph



# Pipe Network Calculation

Radiator flows → Find circuits → Pipe flows → Pressure losses



→ Circuit pressure loss is sum

# Hydronic Balancing

## Static

- Adjustable valves
- Full load only
- Cheaper hardware
- Valve authority
  - 30-70%
- Discrete/continuous
- Branch valves

↓  
MILP

## Dynamic

- Pressure independent valves
- Full & partial load
- Easy adjustment

↓  
Pipe network calculation

# Hydronic Balancing

## Static

- Adjustable valves
- Full load only
- Cheaper hardware
- Valve authority
  - 30-70%
- Discrete/continuous
- Branch valves



MILP

$$\begin{aligned}
 &\text{lex min} \quad \left( \Delta p_{\max}, -\Delta p_{\min}, -\sum_{e \in E} (\Delta p_e^{\text{bal}} + \Delta p_e^{\text{br}} + \Delta p_{e,\text{th}}) \right) \\
 &\text{s.t.} \quad \Delta p_e^{\text{bal}} + \Delta p_e^{\text{br}} \leq \Delta p_{\max} && \forall e \in E \\
 &\quad \Delta p_e^{\text{bal}} + \Delta p_e^{\text{br}} \geq \Delta p_{\min} && \forall e \in E \\
 &\quad (1 - a_v^{\max}) \cdot \Delta p_e^{\text{bal}} \leq a_v^{\max} \cdot \Delta p_e^{\text{br}} && \forall e \in E \\
 &\quad (1 - a_v^{\min}) \cdot \Delta p_e^{\text{bal}} \geq a_v^{\min} \cdot \Delta p_e^{\text{br}} && \forall e \in E \\
 &\quad \sum_{s \in S_{e,r}} x_{e,r,s} = 1 && \forall (e,r) \in (E \times R) \cap \mathcal{D} \\
 &\quad \sum_{s \in S_b} y_{b,s} = 1 && \forall b \in B \cap \mathcal{D} \\
 &\quad x_{e,r,s} \in \{0, 1\} && \forall (e,r) \in (E \times R) \cap \mathcal{D}, \forall s \in S_{e,r} \\
 &\quad y_{b,s} \in \{0, 1\} && \forall b \in B \cap \mathcal{D}, \forall s \in S_b \\
 &\quad \tilde{p}_{e,r} \in [\Delta p_{e,r}^{\min}, \Delta p_{e,r}^{\max}] && \forall (e,r) \in (E \times R) \cap \mathcal{K} \\
 &\quad \tilde{p}_b \in [\Delta p_b^{\min}, \Delta p_b^{\max}] && \forall b \in B \cap \mathcal{K} \\
 &\quad \Delta p_{\max} \in \mathbb{R}_{\geq 0} \\
 &\quad \Delta p_{\min} \in \mathbb{R}_{\geq 0}
 \end{aligned}$$

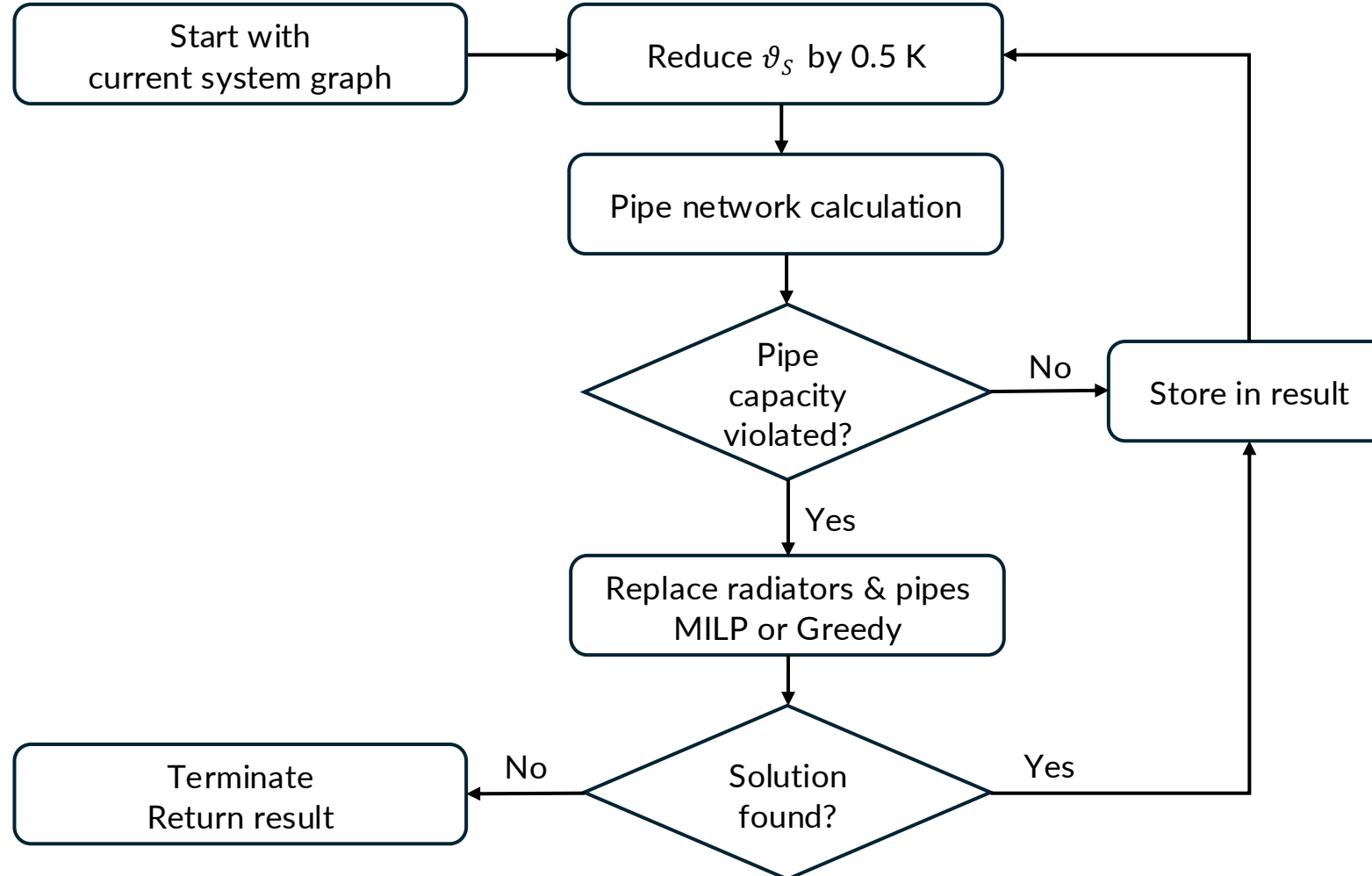
# Supply Temperature Reduction

$$f(\vartheta_S, \Phi_N, \Phi) \rightarrow \text{qm}$$

↓   ↑   -   =

- Pipe network calculation
  - Flow capacities violated?
- Replacements
  - Upsize Radiators
  - Upsize Pipes
  - Cost

# Supply Temperature Algorithm



# Implementation

- Idiomatic Swift
  - Value types
  - Thread safety
- MILP Solver
  - Ip\_solve
  - Swift ↔ C
  - Declarative API

```
var model = LPModel()

let x1 = model.addVariable("x1", kind: .integer)
let x2 = model.addVariable("x2", kind: .continuous)

model.optimizationDirection = .maximize
model.objective = 6*x1 + 9*x2

model.addConstraints {
    2*x1 + x2 <= 42
    2*x1 + 3*x2 <= 161
}

let solution = try model.solve()
print("objective:", solution.objective)
print("x1:", solution.value(of: x1))
```

# Heat Load Evaluation

Reference	Transmission	Ventilation	Building	Conclusion
E DIN SPEC 12831-1 w/o mechanical ventilation	All rooms within $\pm 0.5$ W	All rooms within $\pm 0.8$ W	-	✓ Pass
E DIN SPEC 12831-1 w/ mechanical ventilation	All rooms within $\pm 0.5$ W	11/16 rooms: $\pm 0.5$ W 5/16: up to 231 W*	-	✓ Pass*
EVEBI	All rooms within $\pm 1.0$ W	All rooms within $\pm 15$ W*	4742W (EVEBI) 4730 W (Module)	✓ Pass*
wbs	All rooms within $\pm 0.8$ W	All rooms within $\pm 0.5$ W	5696 W (wbs) 5696 W (Module)	✓ Pass

\* Errors found in reference data

# Hydronic Evaluation

Max. rel. Error	Radiator Flow Rates	Pipe Friction	Pipe Network Calculation	Static Hydronic Balancing
VDI 2073-2	-	3 %**	0.6 %*	-
E DIN 94679-1	0.7 %	0.8 %	0.8 %*	✓ Pass
ZVPlan	2 %	11 %**	1 %***	✓ Pass

\* Errors found in reference data

\*\* Viscosity not stated in reference

\*\*\* Excludes pipe friction

# Performance

	Average Execution Time
Heat Load - Standard Procedure	167 $\mu$ s
Heat Load - Simplified Building	1 $\mu$ s
Heat Load - Simplified Room	5 $\mu$ s
Pipe Network Calculation	532 $\mu$ s
Static Hydronic Balancing	8.3 ms
Supply Temperature Optimization - Greedy	0.8 s
Supply Temperature Optimization - MILP	2.5 s → Worse for large buildings

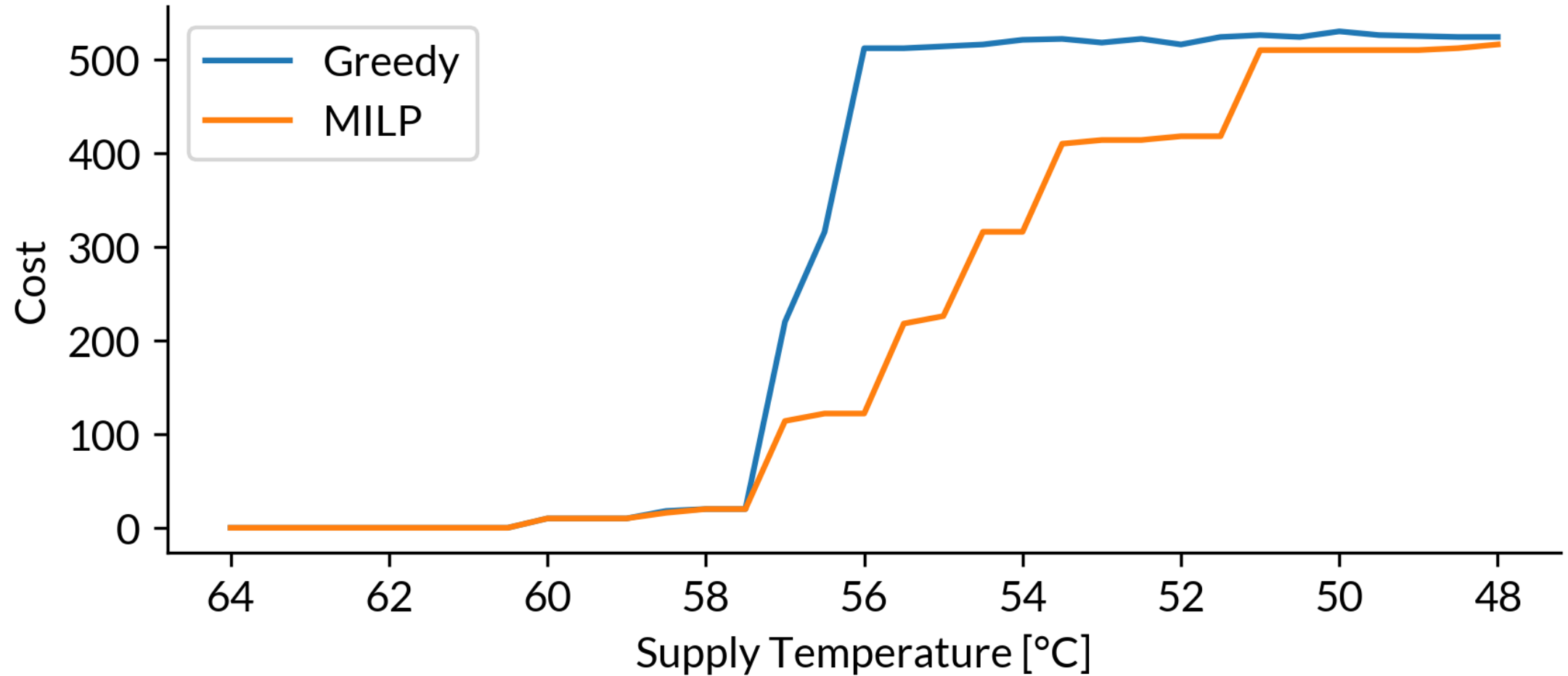
MacBook Pro 16-inch, M2 Max, 32GB RAM, on cable power, macOS 26.3

# Conclusion

- Library → Ready for GEG (Heat Pumps), KfW & BAFA
  - Heat load calculations (DIN/TS 12831-1, DIN EN12831-1)
  - Hydronic balancing (E DIN 94679-1, VDI 2073-2)
  - Supply temperature reduction
- Validated
  - Poor reference data quality → Further validation
- Future Work
  - MILP performance
  - ROI on replacements

Backup

# Supply Temperature MILP vs Greedy I



# Supply Temperature MILP vs Greedy II

