AMC 2026 · DATA CENTER COOLING CASE STUDY

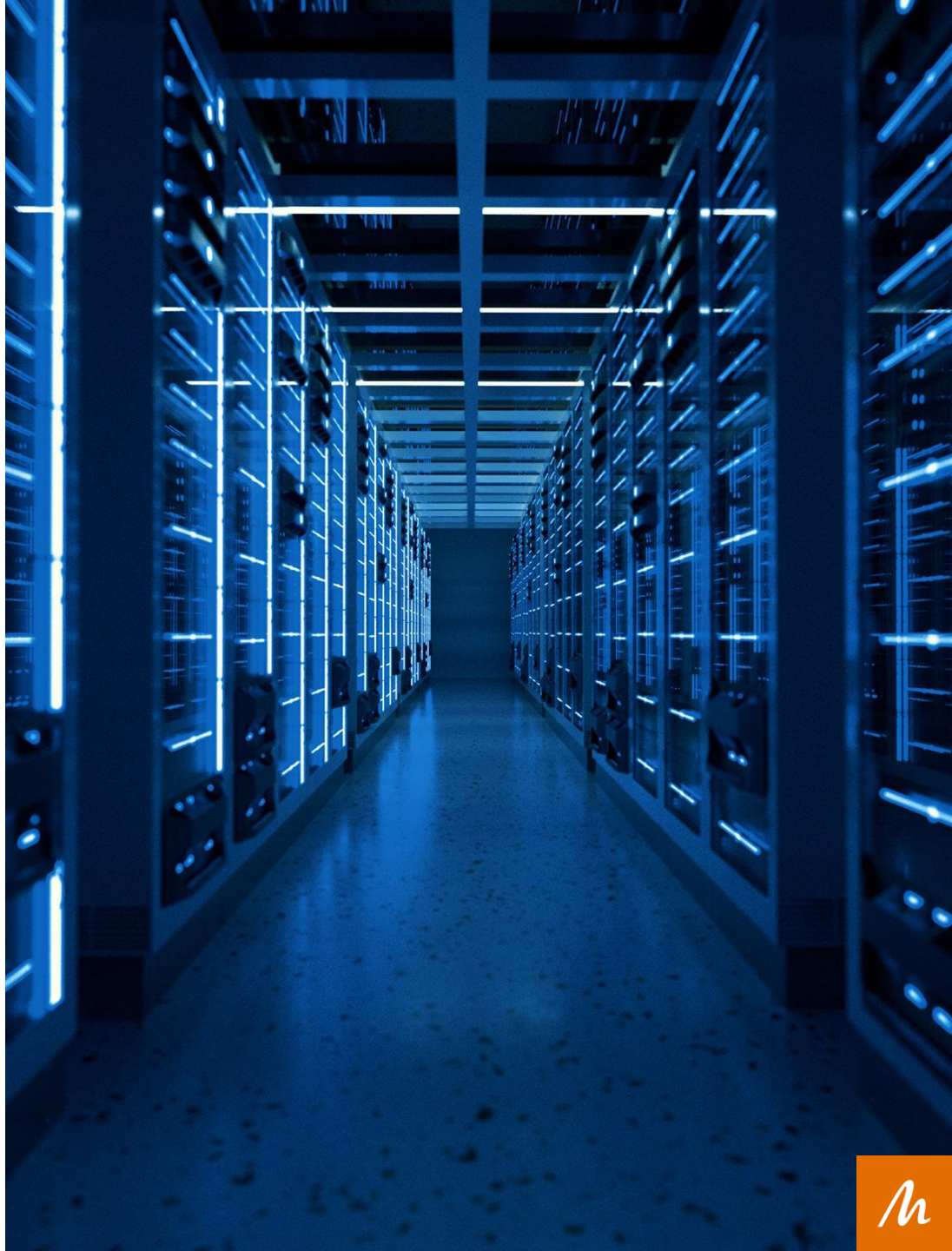
Agent-Assisted Construction of Modelica System Models from Validated Components

From a vendor PDF to a validated 7.4 MW dual-loop cooling model, through an MCP-connected AI agent

Nithish Selvan · Modelon Engineering, India

Johan Andreasson · Modelon AB, Sweden

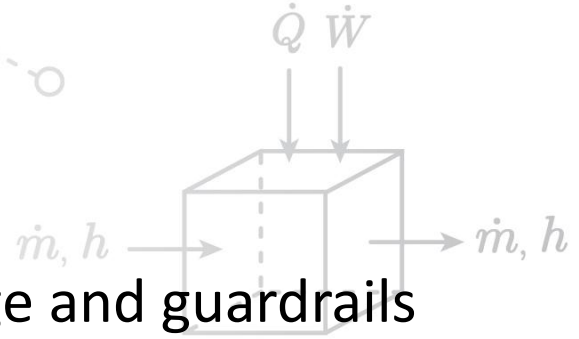
Pratyush Agarwal · Modelon Inc., United States

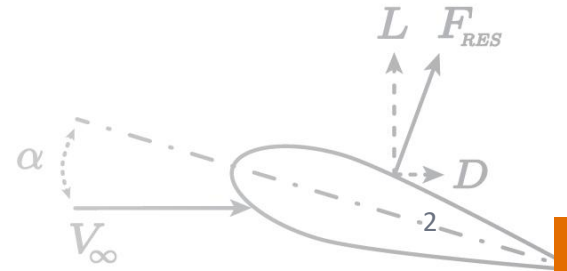


©2026 Modelon. All Rights Reserved.

Outline

1. Motivation
2. Agent setup: architecture, MCP tools, knowledge and guardrails
3. Workflow: from the 1 MW reference to the agent loop
4. Case study: first run, diagnosis and result
5. Where humans come in
6. Discussion, future work and takeaways





From a vendor PDF to a validated model in one session

AI compute is pushing facilities toward high-density liquid cooling. Designers must evaluate more architectures, faster — and most of the spec-to-model work is systematic: read, derive, select, wire.

10 pages

Schneider Electric RD108 PDF as the only input

405 lines

of Modelica written for a 7.4 MW dual-loop plant

19 °C

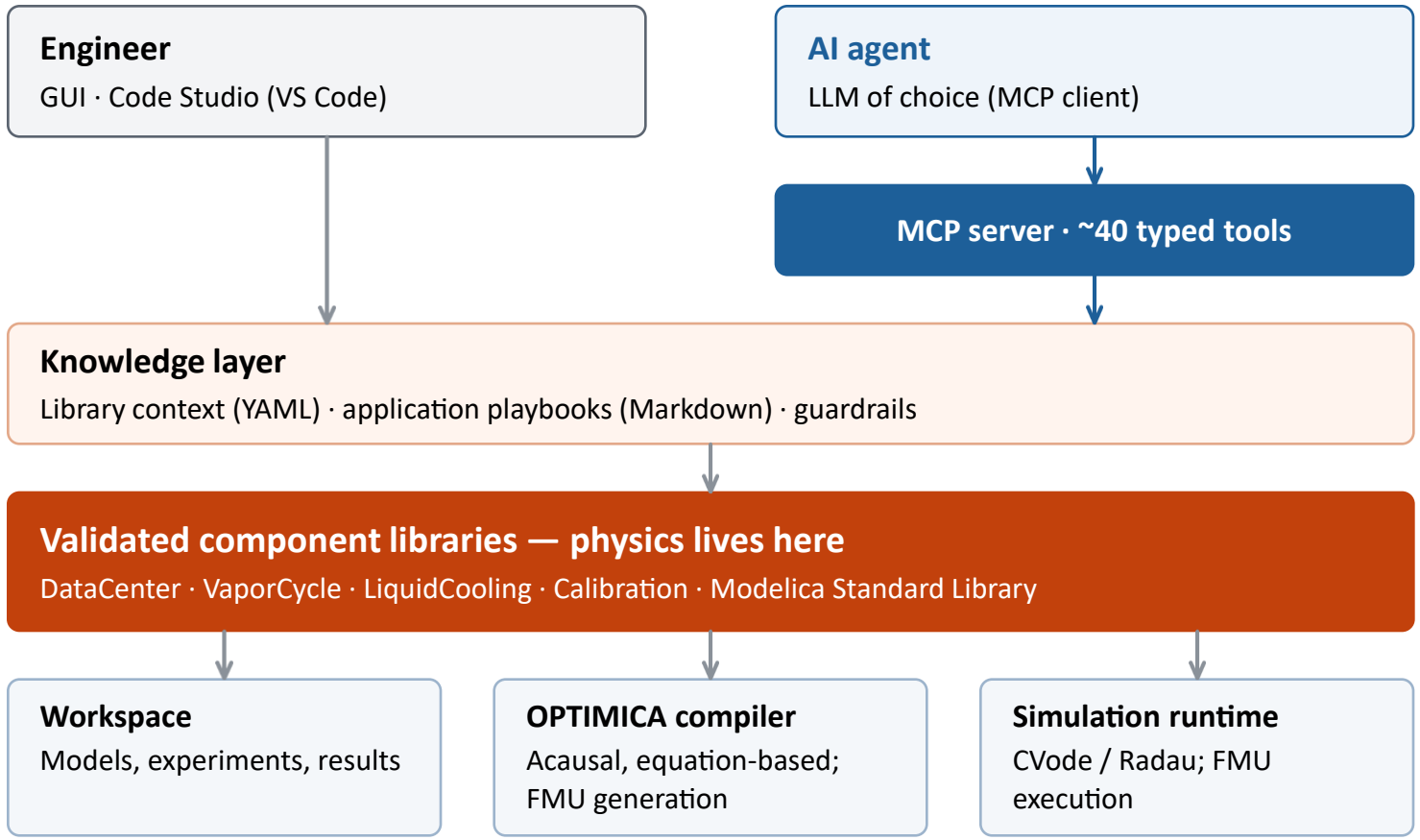
error found on the first run, fixed by a library re-query

≈ 4 h

wall-clock, vs. an estimated 3–4 working days by hand

One user prompt. Two write_model calls, two compilations, two 24 h simulations, one agent-initiated correction loop, no manual coding.

Architecture: the agent on top, physics underneath



Engineer and agent share the same Modelon Impact platform services

Every model change is a typed tool call — logged and attributable

Domain knowledge is plain text that experts can write and review

The library carries the physics; the agent selects and composes

Trajectories stay server-side; only KPI scalars enter the LLM context

MCP turns the platform into typed tools

Model Context Protocol — an open JSON-RPC standard for exposing typed tools to an LLM agent.

- The server wraps the Modelon Impact API as ~40 tools, one per logical operation
- Groups: orientation, library access, model read/write, compile & run, results, DoE, diagram, quality
- Every call returns structured JSON; the whole session is logged
- Building, editing and debugging models all go through the same typed calls

Tool	Returns
get_workspace_libraries	Loaded libraries and versions
list_project_models	Model classes in the project
read_model	Full Modelica source of a model
write_model	Write or overwrite Modelica source
compile_model	Compile to FMU; log and status
run_experiment	Launch dynamic simulation; return ID
get_scalar_results	Named scalar KPIs (JSON)
get_case_results	Time-series trajectories
get_component_interface	Parameters, ports, type info

A knowledge base layered by lifetime

Schemas	Bill-of-materials format, KPI specification, failure taxonomy
Methods M1–M5	Physics-agnostic procedures: sensitivity, DoE, sweeps, Pareto, reporting
Applications A1–A4	Domain playbooks; only one loads — A2 data centres here
Physics P1–P4	First-principles references the agent can cite
Workspace	Findings, open hypotheses, model and simulation status for this project
User preferences	Per-engineer choices the agent applies by default

Organised by **lifetime**: a finding moves up a layer without dragging session noise with it. A Navigator file is read first.

Promoted from this study

LT chiller sizing rule · dual-loop topology pattern · Kv scaling · RD108 HT CDU record — tagged *confirmed*

Three guardrails, enforced as tools

1 Parameter transparency

A structured engineering summary is shown before model assembly. Every extracted and assumed value is visible before any Modelica is written.

2 Verification tagging

Every knowledge-base entry carries a provenance tag: *observed*, *confirmed* or *verified* — what was checked vs. assumed.

3 Provenance enforcement

An MCP tool blocks any report containing a quantitative claim not cited to the experiment that produced it.

All three are **enforced through tool calls**, so execution stops until the condition is met. They apply regardless of domain.

The LLM is one replaceable component

Plans

chooses which tool call comes next

Interprets

recognises patterns in the JSON it gets back

Writes

Modelica source, summaries and the report

The model decides how reliably these are done. The architecture decides whether there is anything trustworthy to plan, interpret or write about.

Modelica · FMI

equation-based models, compile-time checks, vendor-neutral exchange

Python

automation and agent runtime

JSON · YAML

structured tool data; readable library context

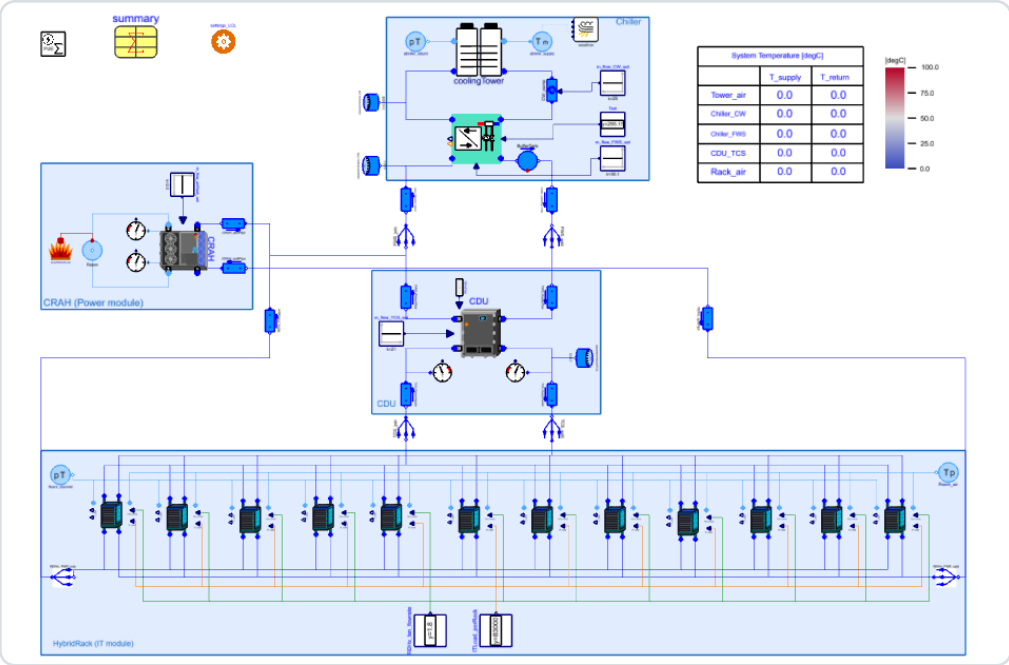
Markdown

domain knowledge experts can review

Case study run with Claude Code and claude-sonnet-4-6. The MCP interface and guardrails do not rely on model-specific behaviour.

CASE STUDY

The task: 1 MW reference to 7.4 MW dual loop



Start: **DataCenter1MW**, validated against RD48 — PUE 1.148 vs 1.12 target; IT load, supply temperatures and CDU duty pass.

RD48 · 1 MW · single loop

- 996 kW IT (12 racks × 83 kW), FWS at 22 °C
- One chiller + water-side economizer, one tower

RD108 · 7.4 MW · two independent loops

- HT loop, 37 °C: 48 AI racks × 132 kW via CDU pods
- LT loop, 23 °C: 48 networking racks × 22 kW
- Tier III, NVIDIA GB200 reference design

Not a scale-up: new loop boundaries, two chiller records, separate flows and valve sizes — derived from the PDF alone.

WORKFLOW

Same engineering steps, different executor

Conventional workflow

- **Engineer** read the PDF, pick out parameters
- **Engineer** derive flows, capacities, Kv by hand
- **Engineer** search library docs for matching records
- **Engineer** build in the GUI or write Modelica
- **Engineer** compile, chase errors
- **Engineer** simulate, post-process, check targets
- **Engineer** diagnose, fix, re-run
- **Engineer** document assumptions

Est. 3–4 working days, experienced engineer

Agent workflow (this study)

- **Engineer** one prompt + the RD108 PDF
- **Agent** extract 27 parameters, tag D / C / A
- **Engineer** review the pre-build table
- **Agent** query library, write 405 lines
- **Agent** compile, run 24 h Berlin TMY
- **Agent** compare KPIs, diagnose, swap record
- **Agent** write confirmed findings to knowledge
- **Engineer** interpret deviations, sign off

≈ 4 h wall-clock · 1 prompt · no manual coding

WORKFLOW

The agent loop, as MCP tool calls

1 Read specification

PDF text layer → 27 parameters tagged D / C / A

2 Query the validated library

`search_components(…)`
`get_component_interface(…)`

3 Assemble model

`write_model(…) → 405 lines, HT + LT loops`

4 Compile and simulate

`compile_model(…)`
`run_experiment(…) → 24 h Berlin TMY`

5 Compare to KPI targets

`get_scalar_results(…) → 5 of 6 met; LT 42.2 °C vs 23 °C`

6 Diagnose and repair

re-query library, swap record, re-run → back to step 2

A failure drives a **library re-query**, not manual debugging. Only KPI scalars enter the context; trajectories stay server-side.

Before any Modelica, every value is tagged

Parameter (excerpt)	Value	Src
Total IT load	7,392 kW	D
PUE target	1.10	D
Cooling budget (PUE − 1)	740 kW	C
TCS medium	Ethylene glycol 25 %	A
AI rack liquid mass flow	3.0 kg/s per rack	C
Networking rack air ΔT	12 K	A
HT chiller capacity	5,520 kW	C
LT chiller capacity	1,872 kW	C
Kv, HT condenser valve	224	C
Chiller record, LT	1,881 kW library record	C

D **Datasheet** — read directly from RD108

C **Calculated** — derived from first principles

A **Assumed** — where the spec is silent

A wrong assumption gets corrected **here**, not after a failed simulation.

Note the last row. It returns on the next slides.

First run: most KPIs pass, one fails clearly

KPI	RD108	Result	Status
PUE	1.10 (annual)	1.153	Close
IT load total	7,392 kW	7,392 kW	Pass
AI racks	6,336 kW	6,336 kW	Pass
Networking racks	1,056 kW	1,056 kW	Pass
T_FWS,HT (CDU supply)	37 °C	41.1 °C	Close
T_FWS,LT (networking)	23 °C	42.2 °C	Fail
Q_CDU (3 pods)	5,512 kW	≈ 5,500 kW	Pass

42.2 °C

LT supply vs. a 23 °C target — a 19 °C miss with a clear thermal signature



Diagnosis: a record-selection error, not physics

LT load: 1,056 kW networking racks + chiller-side heat rejection

↓

Required LT chiller capacity ≈ 1.88 MW

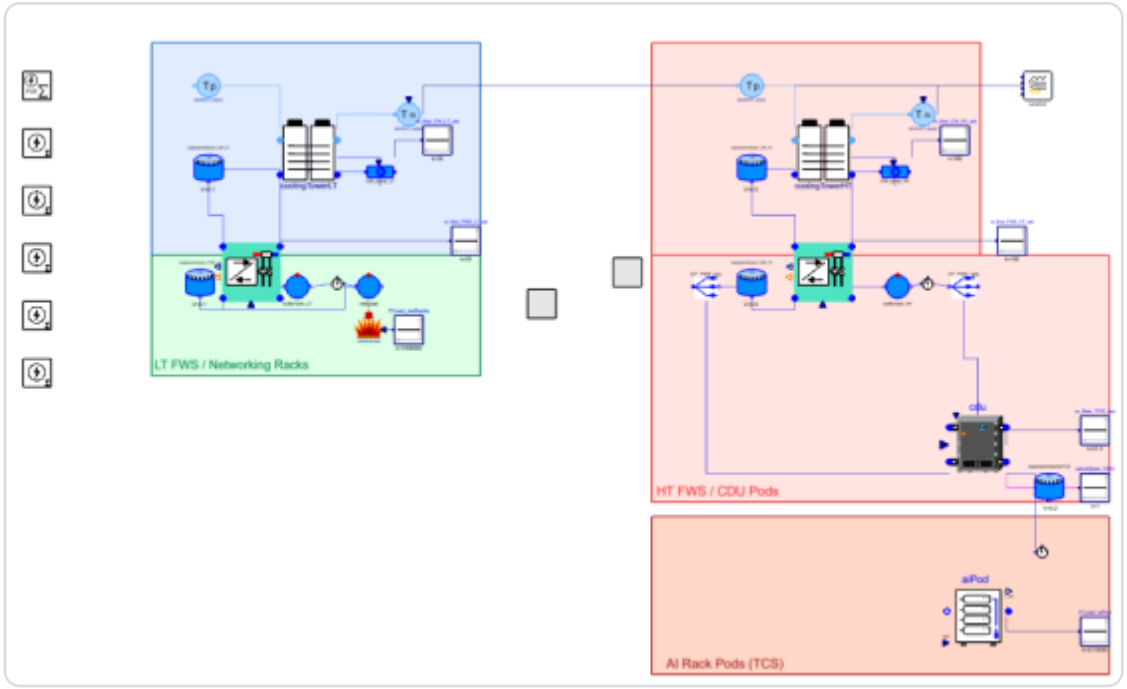
↓

Instantiated record: 1,125 kW — short by ≈ 1.7×

↓

Confirm the 1,881 kW record via get_component_interface, swap it, re-run → **23.0 °C**

The 1,881 kW record was found in the pre-build query but not used when the model was written.



DataCenter7MW. Topology unchanged between runs; one record substituted; 405 → 366 lines after removing redundant instantiations.

Final result, with the deviations explained

KPI	RD108 target	Result	Delta	Status
PUE	1.10 (annual)	1.153	+0.053	Close [a]
IT load total	7,392 kW	7,392 kW	0	Pass
T_FWS,HT (CDU supply)	37 °C	41.1 °C	+4.1 °C	Close [b]
T_FWS,LT (networking)	23 °C	23.0 °C	0	Pass
Q_CDU (3 pods)	5,512 kW	≈ 5,500 kW	−12 kW	Pass

[a] PUE gap

Single winter day (Berlin TMY), not an annual figure. Winter is normally the most favorable season, so this result is not yet comparable to RD108's annual 1.10. An overhead breakdown and an 8,760 h run are needed before quoting PUE.

[b] HT supply offset

Settles 4.1 K above setpoint while the WSE carries 5,494 kW and the HT chiller is idle (2.8 kW). Likely a plant-control setting (chiller trim / tower setpoint), flagged for engineering review.

The engineer owns intent, assumptions and sign-off

Intent
One prompt and the spec; scope agreed

→

Pre-build review
Check assumed (A) values before assembly

→

Agent loop
Derive, assemble, compile, simulate, diagnose

→

Interpret
PUE gap; HT offset — fix or document?

→

Sign-off
Topology fit for the customer; Tier III staging

Automate
Repetitive derivation and assembly: reading specs, computing flows and Kv, selecting records, wiring loops.

Keep human
Decisions that need context the agent does not have: customer use case, site-specific staging, what a deviation means.

Budget for review
The agent writes code faster than an engineer can read it. Review effort grows with productivity.

The library guarantees the physics

The agent never wrote a heat-exchanger equation or derived a chiller efficiency curve.

Correction needs the library too: without validated records there is nothing to re-query — only unverified physics with no calibration history.

Data Center Library 1.3

Built on LiquidCooling 4.0, VaporCycle 5.0 and Calibration 1.4

7 component categories

Chillers at two fidelity levels, towers, CDU pods, CRAH, pumps, fans, racks

15+ CDU records

800 kW – 2 MW, from public manufacturer specs, each with a calibration workflow

9 reference systems

Including DataCenter1MW, validated against RD48

Datasheet-parameterised CDU records for direct-to-chip cooling are not in community libraries such as Buildings v13.

Reasoning versus retrieval

Retrieval

Reliable because the source is validated

- Naming, diagram wiring and port style — read from the 1 MW reference
- CDU capacity, chiller record, rack type — resolved by library query

Reasoning

Not look-up tasks

- A dual-loop topology that does not exist in the reference
- Flows, Kv values and heat-rejection budgets for two circuits
- Tracing the 19 °C error to an undersized record

Conflating the two **overstates** what the agent generates and **understates** what the library provides.

Where the approach does not apply

When a precondition fails

No validated library for the domain

The agent must generate components; the validation guarantee disappears

Poorly specified reference design

Missing flows, undefined media, ambiguous ratings: uncertainty accumulates

Novel physics

The bottleneck moves back to expert modelling; the agent supports, not leads

Gaps seen in this study

Truncated compile log

Limits equation tracing; a full log and dump_equations tool would help

Ambiguous record naming

York YK vs. YT families — the kind of mis-selection behind the first failure

Diagram layout is manual

Placement, icons and connection annotations after write_model

One simulated day

24 h Berlin TMY is not an annual PUE

FUTURE WORK

From research workflow to product

Graphical completeness

Extend MCP to component placement, icons, connection annotations, media assignment and connector orientation.

Deeper diagnosis surface

Full compile logs and an equation-dump tool for structural debugging.

Broader intake

Vendor datasheet to validated model across more data center architectures, with less per-project setup.

Into the product

These skills — playbooks, knowledge layers and guardrails — are being integrated into the AI agent in Modelon Impact.

The same MCP toolchain and guardrails are already in use with thermal management, powertrain, HVAC and vehicle dynamics libraries. The library content changes; the framework does not.

TAKEAWAYS

What this study shows

1. Validated library + typed MCP tools took a vendor PDF to a validated 7.4 MW model in one session.
2. The agent selects and composes. The library guarantees the physics.
3. Engineers own intent, assumptions and sign-off.

Questions?

Nithish Selvan
Modelon Engineering, India
nithish.selvan@modelon.com

Johan Andreasson
Modelon AB, Sweden
johan.andreasson@modelon.com

Pratyush Agarwal
Modelon Inc., United States
pratyush.agarwal@modelon.com

LinkedIn · P. Agarwal