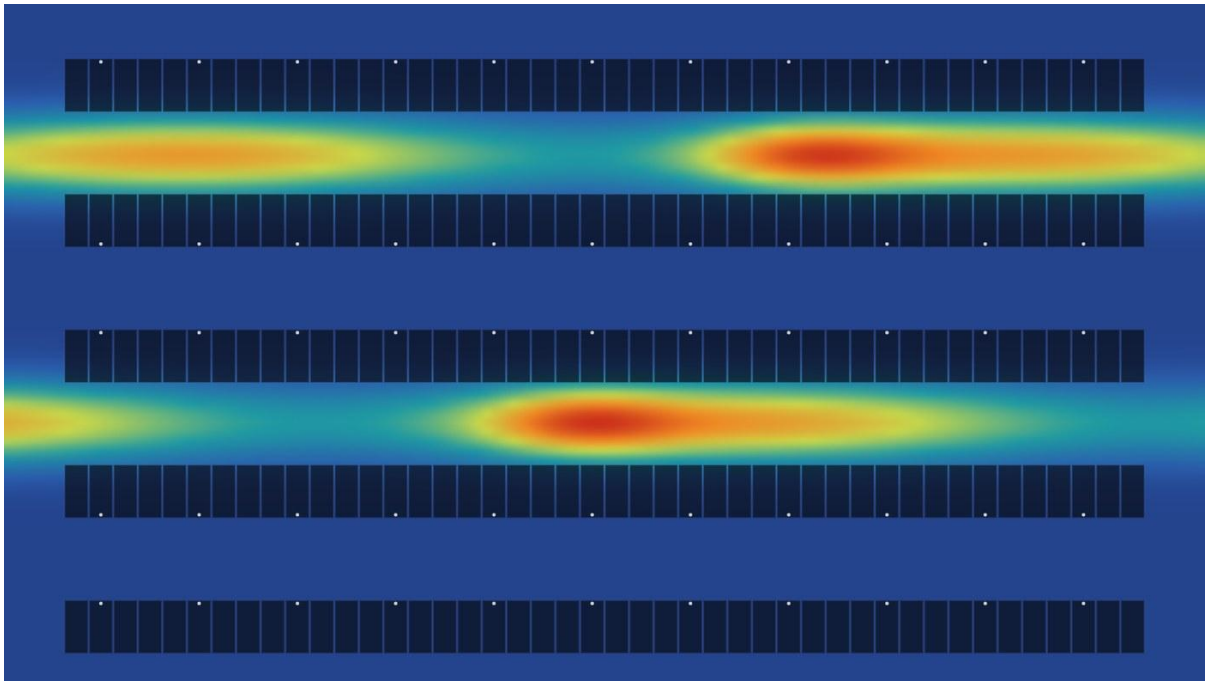


Illustrative temperature field of a data hall, with hot aisles between rack rows.

The Data Center Digital Twin

A practitioner's guide to building, calibrating and operating a living model of power, cooling and compute.

Anomny: AI | IoT | Datacenter

Website: anomny.com

Presales: presales@anomny.com

Contents

Preface **3**

Part I. Foundations **5**

1. What a digital twin is 6
2. Why the data center is an ideal subject 9
3. Anatomy of a twin 11

Part II. Building the twin **14**

4. The data foundation 15
5. Modeling the physical facility 18
6. Thermal and airflow modeling 20
7. The IT and workload layer 23

Part III. Using the twin **25**

8. Capacity planning and placement 26
9. Operations, resilience and training 29
10. Energy, carbon and water 32
11. Designing for AI: density and liquid cooling 35

Part IV. Delivering the twin **38**

12. Roadmap, team and economics 39
13. Security, governance and risk 42
14. The road ahead 45

Appendices **47**

Appendix A. Glossary 48
Appendix B. Interface cheat sheet 50
Appendix C. A sample data model 51
Appendix D. Pilot checklist 52
Appendix E. Standards and further reading 53

About Anomny **54**

Preface

A modern data center is one of the most heavily instrumented buildings ever built, and yet the people running it often cannot say, with confidence, what will happen to the room if a cooling unit trips at 3 a.m. A digital twin is the discipline of closing that gap.

This book is for the people who design, build and run data centers, and for the data, software and finance colleagues who work beside them. It assumes you know what a rack, a UPS and a chilled-water loop are. It does not assume you have built a twin before, or that you know which of the many products and buzzwords around the term are worth your time.

What you will get

The book is organised in four parts. **Part I** defines the term, explains why data centers are unusually good candidates, and lays out the architecture of a working twin. **Part II** covers the building blocks: the data foundation, the facility model, thermal modeling, and the IT and workload layer. **Part III** turns to applications: capacity planning, operations and resilience, energy and carbon, and the special demands of AI infrastructure. **Part IV** covers delivery: roadmap, team and economics, security and governance, and where the field is heading.

Read straight through if you are new to the subject. If you already run a facility, go to the chapter that matches your most expensive current problem and follow the cross-references back to the foundations as needed.

Conventions

- Temperatures are in °C and power in kW or MW unless a figure comes from a source that uses other units.
- “Twin” means a data center digital twin. “Hall” means a data hall, the room or module where IT equipment lives.
- Worked examples use round, illustrative numbers. They are there to show the reasoning, not to benchmark your site.
- Where the book names a standard, it names it so you can look up the current edition. Standards change, and this book will age faster than they do.

> **A note on claims.** Digital twins attract enthusiastic marketing. Wherever this book cites a benefit, it tries to say what had to be true for the benefit to appear: what was instrumented, what was modeled, and who acted on the output. Treat any claimed saving that does not come with those conditions as a hypothesis to test.

Part I. Foundations

1. What a digital twin is
2. Why the data center is an ideal subject
3. Anatomy of a twin

1. What a digital twin is

A digital twin is not a 3D picture of your building. It is a model that stays synchronized with the real facility closely enough to answer questions you would otherwise have to answer by experiment.

A working definition

Definitions vary by standards body and vendor, but they converge on three tests. A system is a digital twin if it has all three:

1. **Representation.** It models a specific physical thing, not a generic one. It is your Hall 3 with your 212 racks, not a typical hall.
2. **Synchronization.** It is kept up to date with the real system, at a stated frequency and to a stated level of fidelity. A model that was accurate at commissioning and has not been touched since is a historical document.
3. **Purpose.** It exists to support a decision: place this load, change this setpoint, schedule this maintenance, approve this design. A twin without a decision attached is an expensive screensaver.

The third test matters most in practice. It sets the required fidelity, the update rate and the data you need to collect, and it is the only reliable defense against building a beautiful twin nobody uses.

What a twin is not

Data centers already have many tools that get called twins. Each is useful, and each is missing something.

Tool	Live data	Predicts	Answers “what if?”	Main use
CAD or BIM model	No	No	No	Design and construction reference
Offline simulation (CFD, electrical)	No, inputs are entered by hand	Yes, for one scenario	Yes, slowly	Design validation, troubleshooting
Monitoring dashboard (BMS, EPMS)	Yes	No	No	Awareness and alarming
DCIM inventory	Partly	Limited	Limited, mostly by arithmetic	Asset and capacity records
Digital twin	Yes	Yes	Yes, on current state	Decisions in design, planning and operation

A twin usually contains or connects to all of these. It is the integration that changes what you can do: a simulation initialized from live data, a dashboard that shows predicted values next to measured ones, an inventory that knows the thermal and electrical consequences of a change.

Scope: from component to campus

Twins nest. A **component twin** models one device, such as a chiller with its efficiency curve or a UPS with its battery state. A **system twin** models a subsystem such as a chilled-water loop or a power path. A **facility twin** covers a hall or building and the interactions between systems. A **portfolio twin** spans multiple sites and is usually much less detailed. Most successful programs start with a system or a single hall and grow outward.

The maturity ladder

Twins are often described as moving through stages of capability. The stages are useful because each one demands more of your data, your models and your organization than the last.



Figure 1.1. The maturity ladder. Most operating twins today sit in the middle steps, with humans approving actions.

Descriptive twins mirror current state: a live, spatially organized view of temperatures, loads and equipment status. **Diagnostic** twins explain deviations by comparing measured behavior with expected behavior. **Predictive** twins forecast what will happen under current conditions or a proposed change. **Prescriptive** twins search the space of possible actions and recommend one. **Autonomous** twins carry out actions themselves within defined limits.

Notice that climbing the ladder is not the goal. Plenty of facilities get most of their value from the second and third steps and rationally stop there. Automation belongs where the decision is frequent, the model is trustworthy and the cost of error is bounded.

Fidelity is a choice, not a virtue

The right level of detail is the lowest one that answers your question reliably. A capacity-planning twin that updates daily and models each rack as a heat source and an airflow demand can be more useful than a fluid-dynamics model that takes hours to run. An operator's early-warning twin needs minutes or seconds of latency but may only need a coarse thermal model. Chapter 3 returns to this trade-off as a design tool.

> **Rule of thumb.** Write the decision, the person who makes it, and the time they have to make it on one line before you choose a model. If you cannot fill in all three, you are not ready to build.

2. Why the data center is an ideal subject

Almost every industry could benefit from a twin. Data centers benefit more than most because their physics is coupled, their instrumentation is dense, and their mistakes are expensive.

Five properties that make data centers good candidates

Tightly coupled systems. Electricity becomes heat almost instantly and almost entirely. Every kilowatt delivered to a rack must be removed by the cooling system, so a change in the IT load is a change in the thermal load, the electrical load on upstream equipment and the efficiency of the plant, all at once. These couplings are exactly what a whole-facility model captures and what separate monitoring systems miss.

Rich instrumentation. Compared with a typical commercial building, a data center already has meters at multiple points in the power chain, temperature sensors at racks, control points on every cooling unit and management controllers in nearly every server. The raw material for a twin usually exists. It just sits in separate systems that do not talk to each other.

High cost of being wrong. Downtime is expensive for the operator or its customers. Overcooling is expensive every hour of every year. Building capacity ahead of demand ties up capital, and building it late loses business.

Constant change. Racks are added, moved and re-densified. Firmware, workloads and equipment generations change the power profile under your feet. Static documents go stale within months, while a twin that is updated through the same change process stays useful.

Scarce headroom. Grid interconnection, transformer lead times and cooling capacity increasingly limit how much compute a site can host. When power is the scarcest resource, wasting it as stranded capacity or excess overhead is costly in a way it was not when power was cheap and abundant.

Where the pain shows up

Problem	What it looks like	What a twin contributes
Stranded capacity	Power is available but cooling or space is not, or the reverse	Shows which constraint binds and what would release the most capacity
Overcooling	Supply air is set colder than needed because nobody is sure where the hot spots are	Predicts rack inlet temperatures, so setpoints can rise safely
Hot spots	A few racks run hot while the room average looks fine	Locates the airflow cause, whether bypass, recirculation or blocked tiles

Problem	What it looks like	What a twin contributes
Slow deployment	Placement decisions require a specialist and days of analysis	Checks candidate locations against power, cooling and failure cases in seconds
Untested failure modes	The team has procedures for failures it has never seen	Rehearses failures against the model before they happen
Lost knowledge	Behavior of the site lives in a few people's heads	Encodes topology, curves and lessons in a reusable model

The energy argument, in one calculation

Power usage effectiveness, PUE, is the ratio of total facility energy to IT equipment energy. Industry surveys have put the fleet average in the mid-1.5s, while the best hyperscale sites report values near 1.1 to 1.2. The gap is largely cooling and power conversion overhead.

Take a 10 MW IT load and electricity at $80 per MWh. One tenth of a point of PUE is 1 MW of continuous overhead, which over 8,760 hours is 8,760 MWh, or about $700,000 a year. That is before counting carbon or the power capacity you free for revenue-earning IT. Chapter 10 has a worked version of this calculation.

Honest limits

A twin cannot fix a poor design; it can only show you the poor design more clearly. It cannot compensate for missing instrumentation: if you cannot measure rack inlet temperatures, you will be modeling them blind. Very small sites, or sites with a single simple load, may not justify the effort. And a twin adds a new system that itself needs maintenance, security and ownership. Chapter 12 treats these costs directly.

> **Field note.** The most common reason a data center twin fails is not technical. It is that nobody owns keeping it current. Decide who owns synchronization before you decide which product to buy.

3. Anatomy of a twin

Every working twin, whatever its vendor or scale, has the same layers. Knowing what each layer is for makes it much easier to judge what you have, what you are missing and what to buy.

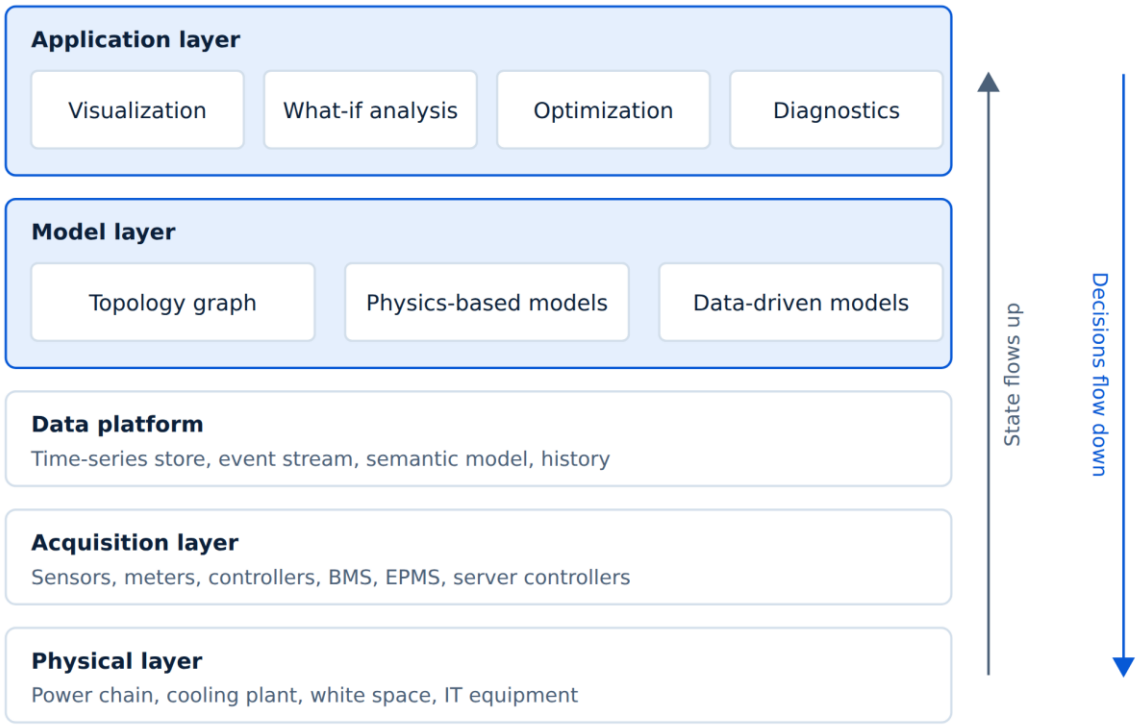


Figure 3.1. The layered architecture of a data center twin. State flows up from the facility; decisions and, in mature systems, bounded control actions flow back down.

Physical and acquisition layers

The physical layer is the facility itself. The acquisition layer is everything that observes it: sensors, meters, programmable controllers, building and electrical management systems and the management controllers inside servers. Chapter 4 covers this layer in depth because it is where most projects meet their first surprises. Sensors drift, clocks disagree, point names are inconsistent and some data that everyone assumes exists does not.

Data platform

The data platform holds the streams and history. It typically combines a time-series store for measurements, an event stream for near-real-time processing, and a semantic model that says what each point means: which asset it belongs to, what it measures, in which units and at which

location. The semantic model is the unglamorous heart of the system. Without it, every application has to rediscover that `AI-0342` is the supply temperature of CRAH 7.

Model layer

Three kinds of model coexist:

- The **asset and topology graph** records what exists and how it connects: which racks hang off which PDU, which PDU off which UPS, which cooling unit serves which aisle. It changes when the facility changes.
- **Physics-based models** encode known laws: airflow and heat transfer, electrical behavior, equipment efficiency curves. They extrapolate sensibly to situations never seen before, but they need calibration and can be slow.
- **Data-driven models** learn behavior from history: anomaly detectors, load forecasts, fast surrogates of slower simulations. They are fast and adapt to quirks of the real site, but they behave poorly outside the conditions they were trained on.

The strongest twins combine all three. The graph tells the models what is connected to what, physics provides structure and safe extrapolation, and learned components correct the residual error.

Application layer

Applications are where humans and other systems meet the twin. Visualization gives a shared picture of the site. What-if analysis answers “what happens if I add this row, lose this unit, or raise this setpoint?” Optimization searches for better operating points. Diagnostics compares expected with observed behavior and flags the difference. Notice that a 3D viewer is only one application. It is often the first one built because it demos well, and it is rarely the one that pays back.

The feedback boundary

The arrow that goes back down is the most consequential design decision. In an advisory design, outputs go to people, who decide and act through the normal change process. In a closed-loop design, the twin writes setpoints to the control system. Between those two lies a range of designs with approvals, limits and fallbacks. Chapter 10 and Chapter 13 describe how to draw that boundary safely. For now, hold the principle that the twin should earn authority in steps and that the facility’s existing control system should always retain the final say.

Synchronization and latency

How fast does the twin need to track reality? Different questions want different answers.

Question	Typical update rate	Typical model
Is a rack about to overheat?	Seconds to a minute	Sensor-driven, fast reduced-order or learned model
Can I raise the chilled-water setpoint?	Minutes	Plant model plus thermal predictions
Where can I place these 12 racks?	Daily or on demand	Topology graph with capacity and failure checks
Should we build another hall?	Weekly to monthly	Forecasts plus scenario models

This is the fidelity trade-off from Chapter 1 in practical form. Detail, speed and data availability pull against each other, and a good architecture lets different applications draw on the same underlying asset graph and data platform at different levels of fidelity.

One twin or many?

A common mistake is to try to build one universal twin. In practice, you will maintain a family: a capacity twin, a thermal twin, an electrical twin, a plant twin. The requirement is not that they be one program but that they share identities and a semantic model, so that “rack A12” means the same thing everywhere and a change entered once shows up in all of them.

Part II. Building the twin

4. The data foundation

5. Modeling the physical facility

6. Thermal and airflow modeling

7. The IT and workload layer

4. The data foundation

A twin is only as good as its data. Most of the effort in the first year of a project goes into finding it, cleaning it and agreeing on what it means, and that effort is well spent.

Where the data lives

The signals a twin needs are scattered across systems built by different vendors, at different times, for different purposes. The table lists the usual sources.

Source	What it provides	Common interfaces	Typical rate
Building management system	Cooling units, chillers, pumps, valves, room and plant sensors	BACnet, Modbus, vendor gateways	10 s to 1 min
Electrical power monitoring	Switchgear, UPS, PDU and busway meters, breaker status	Modbus TCP, SNMP, BACnet	1 s to 1 min
Rack sensors and intelligent PDUs	Inlet temperature and humidity, per-outlet power	SNMP, Modbus, HTTP with JSON	10 s to 1 min
Servers and accelerators	Power, inlet and outlet temperature, fan speed, utilization, GPU telemetry	Redfish, IPMI, vendor agents	1 to 60 s
Network equipment	Port status, optics temperature, throughput	SNMP, streaming telemetry	10 s to 1 min
Leak, smoke and environmental detection	Discrete alarms and states	Dry contacts, BACnet	Event-driven
DCIM, CMDB and ticketing	Asset locations, models, owners, planned changes	REST APIs, database exports	On change or daily
Building models	Geometry, equipment placement, as-built drawings	IFC, CAD and BIM exports, point clouds	On change
Schedulers and orchestrators	Jobs, placement, queue depth, planned load	Scheduler and cluster APIs	1 s to 1 min
Weather and grid	Outdoor temperature and humidity, grid price and carbon intensity	External APIs	5 to 60 min

Middleware such as MQTT brokers and OPC UA servers often sits between the operational systems and the data platform. Their job is to decouple the two, so that the twin reads from a stable, documented interface instead of reaching into the control network directly.

Meaning before volume

The first hard problem is not throughput but semantics. A raw point called `AI-0342` or `CRAH7_SAT` means nothing to an application until it is tied to an asset, a measured quantity, a unit and a location. Open metadata schemas such as Brick and Project Haystack exist to describe exactly this. You do not have to adopt one wholesale, but you should use one consistent vocabulary, and record for every point at least:

- the **asset** it belongs to, by stable identifier;
- the **quantity** measured, such as supply air temperature or active power;
- the **unit**, normalized at ingestion, because a plant with three vendors will report in three ways;
- the **location**, ideally by reference to the spatial model rather than free text;
- the **source and sampling behavior**: polled or pushed, and whether values are reported on change or on a fixed interval.

Appendix C shows a small example.

Time is data too

Correlating a fan speed change with a temperature response requires timestamps that agree. Synchronize every collector to a common time source and record both the source timestamp and the ingestion timestamp. Decide how to handle intermittent devices that report only on change, because filling gaps by carrying the last value forward is correct for a setpoint and wrong for a temperature that has silently stopped updating.

Quality is a feature you build

Assume some of your data is wrong at any moment. Common faults include sensors that drift or stick, meters with wrong scaling, units swapped between vendors, points mapped to the wrong asset and communication gaps that leave holes in history. Build checks into ingestion instead of discovering faults during an incident:

- **Range and rate checks:** a rack inlet temperature of −40 °C, or one that jumps 15 °C in a second, is a fault, not a reading.
- **Flatline detection:** a value that has not changed to the last decimal in hours is suspicious.
- **Cross-checks:** the sum of branch-circuit power should roughly match the parent meter, and heat removed by a coolant loop should roughly match the electrical load it serves. Energy balance is one of the strongest validators a twin has.
- **Quality flags:** carry a per-point quality score forward so that applications can ignore or down-weight suspect data instead of silently trusting it.

Sensor placement

The most important measurement in a data hall is the temperature of air entering IT equipment, because that is what the equipment and the thermal guidelines care about. Room-level return temperatures are a poor proxy. Guidance from industry bodies generally recommends measuring at rack inlets at more than one height, since stratification means the top of a rack can run several degrees warmer than the bottom. If the budget allows only a subset of racks, prioritize the ends of rows, the highest-density racks and the racks farthest from cooling supply, and choose them so the model can interpolate the rest.

Storage and history

Most implementations use three stores working together: a time-series database for measurements, a stream or message bus for near-real-time use, and a graph or relational store for assets and relationships. Retain raw data long enough to recalibrate and to replay incidents, then downsample. Keep at least a few years of hourly or better history for seasonal analysis, because weather and load patterns repeat annually and models must be validated across them.

> **In practice.** Run a “data audit” before choosing a platform. For every use case you care about, list the required signals, check whether each exists, is trustworthy and can be read programmatically, and score the result. The gaps you find define the real project scope.

5. Modeling the physical facility

The facility model answers three questions: where things are, how they connect, and how they behave. Most twins need all three, but they are built and maintained by different methods.

Spatial model: where things are

The spatial model places every rack, cooling unit, tile, cable tray and structural element in three dimensions. It usually starts from design drawings or a building information model, and is checked against the real room with a walk-down or a laser scan, because as-built conditions differ from drawings more often than anyone likes to admit. Level of detail should follow use: rack positions and heights are enough for capacity planning, while airflow modeling needs tile openings, containment gaps, cable cutouts and under-floor obstructions.

Topology model: how things connect

Data centers are networks of dependencies, and a graph is the natural way to represent them. Nodes are assets; edges are physical or logical connections such as “feeds power to” and “removes heat from.” With this graph the twin can answer questions that spreadsheets make painful:

- Which racks lose power if this UPS output breaker opens?
- Which cooling units can reach this row, and how much do they share?
- If Path A is under maintenance, what does Path B carry, and is that within rating?

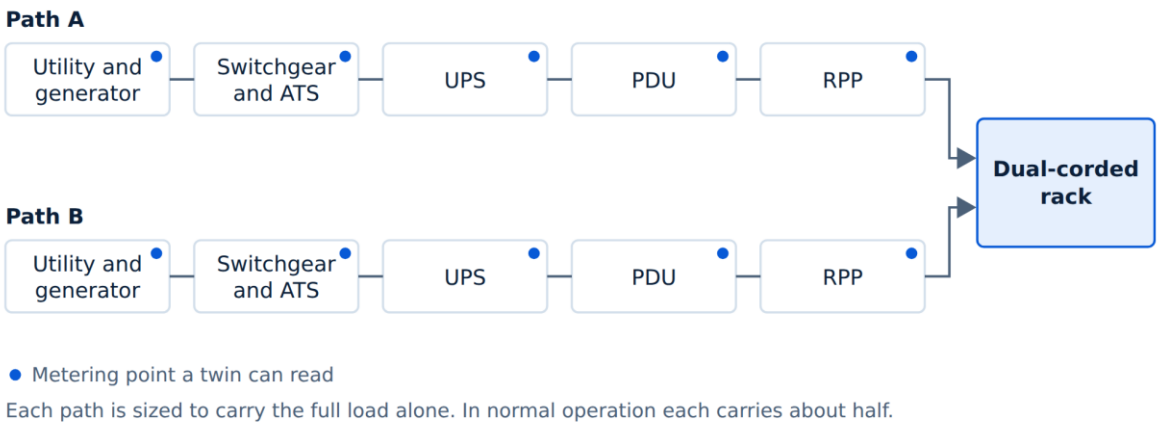


Figure 5.1. A 2N power topology, simplified. Redundancy is a property of the graph: a twin can test it for every single failure instead of trusting the design intent.

Redundancy schemes such as N+1 and 2N are properties of topology plus loading, and they degrade silently as racks are added. A topology graph combined with live load lets the twin run an **N−1 analysis**: for each component that could fail, redistribute its load along the remaining paths and check whether any surviving component would exceed its rating. Doing this every time the load changes turns redundancy from a design claim into a monitored quantity.

Behavioral model: how things respond

Equipment does not behave linearly, and the twin must know the shape of its curves:

- **Fans and pumps** follow affinity laws: flow scales with speed, and power scales roughly with the cube of speed. A small speed reduction across many fans saves disproportionately.
- **Chillers** vary in efficiency with load, with the temperature of the water leaving and entering, and with outdoor conditions. The efficiency map is a core input to any energy optimization.
- **UPS systems** are less efficient at light load, which matters in a hall that is only partly filled or in a 2N design where each unit carries half.
- **Cooling coils and heat exchangers** have capacity limits that depend on flow, temperature difference and fouling.

Manufacturers' curves are the starting point. Real equipment ages and installs differently, so the twin should fit each curve to measured data and store the fitted curve with its date, its error and the range of conditions it covers.

Authoring and keeping it current

The largest source of twin failure is **drift between the model and the room**. Racks get moved, breakers get re-labeled, a cooling unit is swapped for a different model, and nobody updates the model. Three practices help:

1. Attach twin updates to the **existing change process**. A move, add or change ticket should not close until the model reflects it.
2. Run **automated reconciliation**. Compare what the model says is on each PDU outlet with what metering and server controllers report, and flag mismatches for a person to resolve.
3. Schedule a **periodic physical audit** of a sample of racks, and record the mismatch rate. A rising mismatch rate is an early warning that process discipline is slipping.

6. Thermal and airflow modeling

If there is one model that gets people to say “digital twin” in a data center, it is the thermal model. Air is invisible, it takes surprising paths, and a few degrees of error at the wrong rack is the difference between comfortable and constrained.

What you are trying to predict

The quantity that matters is the **temperature of air entering the IT equipment**. Industry thermal guidelines from ASHRAE’s data center technical committee define recommended and allowable envelopes by equipment class. For the commonly used classes the recommended range for inlet air is 18 to 27 °C, with wider allowable ranges for tolerant equipment. Confirm current values and your own equipment’s specifications, but the shape of the goal is clear: keep every inlet inside its envelope with as little cooling energy as possible.

The twin’s thermal job is therefore to predict inlet temperatures across the hall, for the current state and for proposed changes, quickly enough to be useful.

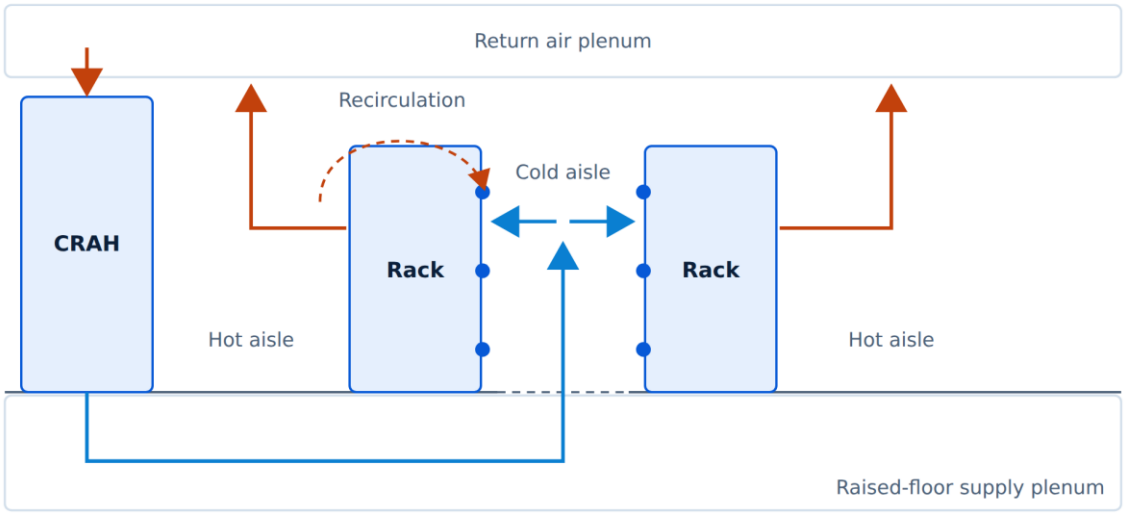


Figure 6.1. Air path in a raised-floor hall. Blue is supply air and orange is exhaust. Dots mark inlet sensors. The dashed arrow shows recirculation, where exhaust air finds its way back to an inlet.

Why airflow goes wrong

- **Bypass:** cold air returns to the cooling unit without passing through any IT equipment, wasting fan energy and lowering return temperatures so units appear to be doing less work than they are.
- **Recirculation:** hot exhaust air wraps around or over racks and re-enters an inlet. It causes hot spots even when the room’s cooling capacity is ample.

- **Supply imbalance:** tiles or nozzles deliver more air to some racks than they need and less to others, because under-floor pressure is uneven or because a rack's demand changed.
- **Blockages and leaks:** missing blanking panels, cable cutouts, open containment doors and under-floor obstructions all change the pattern.

Indices such as the Rack Cooling Index and the Return Temperature Index summarize how well a room is doing on these dimensions. They are useful for tracking a hall over time and comparing before and after an improvement.

Choosing a model family

Approach	Fidelity	Speed	Needs	Best for
Full CFD	High, resolves flow paths in 3D	Minutes to hours per case	Detailed geometry, calibration	Design, retrofit studies, root cause of stubborn hot spots
Zonal or potential-flow models	Medium, captures dominant airflow paths	Seconds	Geometry, tile and unit data	Interactive what-if and placement checks
Learned surrogates	Medium to high inside the trained range	Milliseconds	Many simulation runs or long sensor history	Real-time prediction and optimization loops
Sensor interpolation	Low, describes only what sensors see	Instant	Dense inlet sensing	Live situational awareness

A practical pattern is to use full CFD offline to generate a library of scenarios, then train or fit a fast surrogate on it. The surrogate powers real-time features, and CFD is re-run when the geometry changes materially. Sensor data anchors both, by initializing boundary conditions and by supplying the residual that tells you the model is drifting.

Calibration and validation

A thermal model that has not been checked against measurements is a hypothesis. A sound workflow looks like this:

1. **Capture** a steady, well-measured snapshot: rack power, cooling unit airflow and temperatures, tile flow where measurable, and inlet temperatures.
2. **Initialize** the model with those boundary conditions and run it.
3. **Compare** predicted and measured inlet temperatures rack by rack. Examine the pattern of error, not only the average.
4. **Tune** the uncertain parameters. In practice these are usually leakage paths, unit airflow versus fan speed, tile flow coefficients and the effective airflow demand of each rack.

5. **Hold out** a second snapshot, or a deliberate test such as a controlled change of setpoint, and confirm the tuned model predicts it.

Set an explicit accuracy target tied to the decision. A team that intends to raise supply temperature by a degree or two needs inlet predictions accurate to well under that margin. A team screening placement options may accept a coarser error if the model ranks locations correctly. Report uncertainty alongside every prediction. A twin that says “27.4 °C” is less useful, and less honest, than one that says “27.4 °C, plus or minus 1.5.”

Time and transients

Steady-state answers are not enough. What happens in the first minutes after a cooling unit fails or a chilled-water pump trips determines whether IT equipment throttles or shuts down. Thermal mass in air, water and structure sets the ride-through time, and it varies widely between sites. Modeling transients is harder than modeling steady state, but it is the model behind the most valuable operational question: how much time do the operators have?

> **Field note.** Do not begin with the most elaborate model. Begin with sensor-based awareness and a coarse model, measure how often they mislead you, and add fidelity only where the errors cost money. Teams that start with a hall-scale CFD often spend months calibrating a model whose first use is something a simpler one would have answered.

7. The IT and workload layer

A facility model with no understanding of what is happening inside the racks is only half a twin. IT equipment is where electricity becomes heat, and it is also the part of the system that changes fastest.

From utilization to power to heat

The twin needs to know how much power each device draws now, how much it might draw soon, and where the resulting heat goes. Three sources of information combine:

- **Measured power** at rack, PDU outlet or server level. This is the ground truth where it exists.
- **Nameplate and specification data** from the asset inventory. Nameplate ratings are safe upper bounds and poor estimates of real draw; using them alone is a leading cause of stranded capacity.
- **Power models** that estimate consumption from utilization, configuration and workload for devices that are not individually metered.

Server power is not proportional to utilization. Idle draw can be a substantial fraction of peak, and the relationship is non-linear and depends on the processor, memory, fans and power supply. Fit models per hardware generation, and re-fit after firmware or configuration changes.

Inventory as a model input

The twin inherits the quality of the asset inventory. For each device it needs a stable identity, model and configuration, rack and position, power feeds, airflow direction, and owner. Reconcile the inventory against discovery data, such as what management controllers and network switches report, and treat the discrepancies as a standing to-do list rather than a one-off cleanup. A rack that the inventory says holds ten servers and the metering says draws the power of twenty is exactly the kind of thing a twin is supposed to catch.

Accelerated computing changes the rules

Racks built around GPUs and other accelerators draw far more power in less space, and their behavior is different in kind:

- **Density:** rack power that was once measured in single-digit kilowatts now reaches tens or, for the newest systems, well over a hundred kilowatts.

- **Synchronized load:** large training jobs can drive thousands of devices through compute and communication phases together, producing fast, large swings in facility power that stress electrical and cooling equipment designed for smoother loads.
- **Power capping and telemetry:** accelerators and CPUs expose controls and counters for limiting and reading power. A twin can use these to test whether a cap on a few racks releases capacity or removes a hot spot, and at what performance cost.

Chapter 11 returns to these systems in detail. For now, note that the twin should model workloads as first-class objects, not as an afterthought to the hardware.

Connecting workloads to the facility

Schedulers and orchestrators decide where jobs run, and therefore where heat appears. When the twin has access to the scheduler's queue and placement, it can forecast the power and thermal impact of planned jobs, warn before a placement would overload a circuit or a cooling zone, and in more advanced setups feed constraints back to the scheduler. This is one of the highest-value integrations because it aligns the decisions of two teams who often work in ignorance of each other.

Workloads differ in flexibility. Latency-critical services must run now, where they are. Batch and training jobs can often wait, move or slow down. A twin that knows which is which can shift flexible load to cooler times, cooler halls or lower-carbon hours without touching the critical workloads.

The network and the small stuff

Switches, optics and cabling contribute both heat and constraints. High-speed optical modules generate concentrated heat at the front or rear of equipment, and dense cabling affects airflow. In large clusters, network topology limits where workloads can be placed together, which in turn constrains where power and cooling must be delivered. Include the network in the topology graph, even if only coarsely, so placement checks can consider it.

> **In practice.** The IT team and the facilities team usually own different systems, budgets and vocabularies. Getting the two data sets into one model is often more of a relationship project than an engineering one. Start with a single, concrete shared benefit, such as fewer surprises when a new cluster lands, and the rest follows.

Part III. Using the twin

8. Capacity planning and placement

9. Operations, resilience and training

10. Energy, carbon and water

11. Designing for AI: density and liquid cooling

8. Capacity planning and placement

Capacity is the most immediately profitable use of a twin. Every hall has more usable capacity than its spreadsheets admit, and the twin's job is to show where it is and how to release it safely.

Four resources, one binding constraint

A rack position is usable only if **space, power, cooling and connectivity** are all available together, and often weight, structural loading and airflow direction as well. The resource that runs out first sets the limit, and the others are then stranded: paid for, installed and idle.

Consider a hall with 2,000 kW of available power, 1,600 kW of cooling, 180 rack positions and an average deployment of 10 kW per rack. Power supports 200 racks, cooling supports 160 and space supports 180. Cooling binds. The hall can take 160 racks, and 400 kW of power and 20 positions sit unused. Adding power would achieve nothing. Adding cooling, or making the existing cooling reach the racks better, would release capacity immediately.

Worked example: which constraint binds?

The web edition of this book includes an interactive version of this calculation. The table shows the same hall at two average rack densities.

	10 kW per rack	20 kW per rack
Available power	2,000 kW	2,000 kW
Available cooling	1,600 kW	1,600 kW
Rack positions	180	180
Racks supported by power	200	100
Racks supported by cooling	160	80
Racks supported by space	180	180
Racks the hall can take	160	80
Binding constraint	Cooling	Cooling
Power left unused	400 kW	400 kW
Positions left unused	20	100

Try raising the density to 20 kW. Power and cooling now support only 100 and 80 racks, so the hall fills less than half of its 180 positions and cooling still binds. Adding more positions would do nothing. This is why capacity plans built on position counts break as hardware gets denser.

What a twin adds to a spreadsheet

A spreadsheet can run the arithmetic above. It cannot see the constraints that live in the interactions:

- **Locality.** Cooling capacity is not a single pool. A hall may have plenty of total cooling and no capacity at the far end of the row where the free positions are.
- **Failure cases.** Capacity that exists only when everything works is not capacity in a redundant design. The twin must check every placement against the N−1 cases from Chapter 5, both electrical and thermal.
- **Actual draw.** Planning on nameplate wastes headroom. Planning on last week's average risks a breaker trip at peak. The twin uses measured peaks, with an explicit margin, and shows the distribution.
- **Interactions.** Adding a dense rack at one end of a row changes airflow for its neighbors. Only a thermal model sees that.

The placement question

The daily use of a capacity twin is answering: *where can I put this?* Given a request for some number of racks with a stated power, cooling direction, network and redundancy requirement, the twin scans candidate positions and returns those that pass every check, ranked by an objective such as least stranded capacity, best airflow or lowest cable length. A good answer takes seconds and cites its reasons, so the requester can see why a location was rejected.

The steps are straightforward to state:

1. Enumerate candidate positions that satisfy space, weight and connectivity.
2. Check electrical headroom on every upstream element in normal operation and under each single failure.
3. Predict inlet temperatures for the new racks and their neighbors, normal and degraded.
4. Score the survivors and present the top options with the constraint that would be closest to binding.
5. On approval, **reserve** the capacity and record the change so later requests see it.

Forecasting and reservation

Lead times for power and cooling equipment can be long, so planning looks months or years ahead. Combine committed orders, pipeline probabilities and historical growth into a demand forecast, then run it through the twin to find *when* and *which* resource will first bind. That is the trigger for procurement. Treat reservations with care: capacity held for a customer who never arrives is another form of strand, so give reservations an expiry.

Releasing stranded capacity

When cooling binds, the cheapest fix is often not new equipment but better use of what exists. Sealing bypass paths, adding blanking panels, rebalancing tiles, raising the supply temperature within the envelope, and moving a few high-density racks toward strong supply can each free capacity. The twin lets you test each option before touching anything and rank them by capacity gained per dollar and per risk.

> **Why this pays first.** Capacity built is very expensive per megawatt, so recovering even a small share of stranded capacity can be worth more than years of energy savings. It is also relatively low-risk: it is decision support, with a human choosing and the normal change process in force.

9. Operations, resilience and training

Operations is where the twin earns trust. It is also where a bad recommendation is most visible, so the emphasis is on catching problems early, explaining them clearly and rehearsing responses before they are needed.

Anomaly detection and early warning

Threshold alarms fire when a value crosses a fixed line. They are necessary and they are late. A twin lets you compare each measurement with what the model *expected* given current conditions, and alert on the residual. A cooling unit whose supply temperature is within limits but 2 °C warmer than the model predicts for its load and flow is telling you something long before a threshold trips.

Residual-based detection has two virtues. It adapts to load and weather, so it flags fewer false alarms, and it points at causes, because the pattern of residuals across neighboring assets distinguishes a failing unit from a room-wide change. Its weakness is that it is only as good as the model. Track the detector's precision, and let operators mark alerts as useful or not, so the model improves.

Predictive maintenance

Fans, pumps, bearings, batteries and compressors leave signatures in the data before they fail: rising vibration, growing power for the same flow, slower response to commands, falling battery capacity or rising internal resistance. A twin that knows the expected behavior of each asset can estimate remaining useful life and schedule maintenance for when it is cheapest and least risky. Its value is largest for assets that are expensive to replace, hard to access, or that would put redundancy at risk if they failed unexpectedly.

Alarm rationalization

During a real event, control rooms are often flooded with alarms, most of them consequences of one root cause. Because the twin has the topology graph, it can group alarms by upstream cause, show operators the likely origin, and suppress the dependent noise. This is one of the least glamorous and most operationally valuable applications.

Rehearsing failures

Most sites write procedures for events they have never experienced. The twin lets teams run those events in advance: what happens if CRAH 4 fails during a hot afternoon? If Path B goes into maintenance while Path A has a battery fault? If a chilled-water pump trips at high load?

The output is not just “it survives” or “it does not,” but the timeline: how quickly inlet temperatures rise, which racks cross a limit first, and how long operators have to respond.

Use these runs to test **methods of procedure** before executing them on live equipment. A planned maintenance activity can be checked for hidden single points of failure, and the safest sequence chosen. Simulated rehearsal does not replace testing on the real system, but it makes the real test safer and shorter.

Commissioning and integrated systems testing

Commissioning verifies that a new facility performs as designed. The final and most demanding stage, integrated systems testing, exercises power and cooling together under simulated failures. A twin built during design can be compared with measured behavior during these tests. Where they diverge, either the model or the installation is wrong, and both are worth knowing. The calibrated model that results is a running start for operations, instead of a blank slate.

Incident replay and learning

After an incident, replay the recorded data through the twin. Compare what the model predicted with what happened, identify where operators had information they could not see, and record the lessons in the model: a failure mode to add, an alarm to regroup, a curve to correct. Over time the twin becomes an institutional memory that new staff can query.

Training

Operators learn fastest from realistic scenarios. A twin can drive a training environment that reproduces a chosen incident with plausible sensor data, allowing new staff to practice diagnosis and response without risk. Score responses against expected actions and time-to-decision.

Summary of operational use cases

Use case	Key data	Model type	Typical maturity
Residual anomaly detection	Cooling, power and rack temperatures	Physics or learned expected-value model	Diagnostic
Predictive maintenance	Vibration, current, efficiency drift, battery metrics	Degradation and remaining-life models	Predictive
Alarm grouping	Alarm streams, topology graph	Graph reasoning	Diagnostic
Failure rehearsal	Live state, topology, transient thermal model	Dynamic physics model	Predictive
Procedure checking	Topology, live load	N−1 electrical and thermal analysis	Predictive to prescriptive

Use case	Key data	Model type	Typical maturity
Incident replay and training	Recorded history	Any of the above	Descriptive to diagnostic

Shadow mode first

Before any operational twin output influences a decision, run it in **shadow mode** for a meaningful period: it makes predictions and recommendations, humans ignore them, and you score what would have happened. Shadow mode builds evidence for or against trust, and often reveals model errors that would have been embarrassing in production.

10. Energy, carbon and water

Energy is where the twin's coupled physics pays off continuously. The same model that predicts a hot spot can also find the setpoint at which cooling costs the least while every inlet stays comfortable.

Measuring efficiency honestly

The key metrics are standardized. **Power usage effectiveness** (PUE) is total facility energy divided by IT equipment energy. **Water usage effectiveness** (WUE) is site water use per unit of IT energy. **Carbon usage effectiveness** (CUE) is emissions per unit of IT energy. Definitions and measurement boundaries are set out in the ISO/IEC 30134 series and by The Green Grid. Whichever you use, state where you measure, over what period and whether the figure is instantaneous, annualized or averaged.

PUE is useful and easy to overrate. It says nothing about whether the IT equipment is doing useful work, and it can be improved in ways that increase total energy. For example, raising supply air temperature lowers chiller energy but can make server fans spin faster. The fan energy is counted as IT load, so PUE improves even if the total energy does not. Use PUE as one of several indicators and always track absolute energy alongside it.

Worked example: what does overhead cost?

The web edition includes an interactive version. The table uses a 10 MW IT load, power distribution losses of 8 percent of IT power, other overhead of 4 percent, and electricity at $80 per MWh. It compares cooling overhead of 30 percent and 20 percent of IT power.

	Cooling overhead 30%	Cooling overhead 20%
PUE	1.42	1.32
Overhead power	4.2 MW	3.2 MW
Overhead energy per year	36,792 MWh	28,032 MWh
Overhead cost per year	$2.94 million	$2.24 million

At this load and price, each 0.1 of PUE is worth about $700,800 a year.

The levers

Most of the recoverable overhead in a typical facility sits in the cooling system, so the levers concentrate there:

- **Supply air or water temperature.** Higher supply temperatures allow more hours of free or economized cooling and make chillers more efficient, within the inlet envelope and subject to the server-fan trade-off above.
- **Airflow matching.** Cooling fans should deliver what the IT needs, not more. Because fan power varies roughly with the cube of speed, modest reductions pay off strongly.
- **Plant sequencing.** Which chillers, pumps and towers run, and at what temperatures, changes with load and weather. A plant model can find combinations that rules of thumb miss.
- **Economizer strategy.** Deciding when to use outside air or water-side free cooling depends on forecast weather, humidity limits and load.
- **Containment repair.** Fixing bypass and recirculation is the enabling step for everything else. Higher setpoints are only safe when inlets are predictable.
- **Load placement.** Putting flexible IT load where cooling is efficient and headroom is largest reduces overhead without changing any equipment.

Optimization with a safety envelope

Given a model that predicts both inlet temperatures and cooling energy, optimization becomes a search: find the setpoints that minimize energy subject to constraints on inlet temperatures, equipment limits and rate of change. The constraints are the important part. An optimizer will exploit every weakness in a model, so limits must be set on the inputs it can choose, not only on the outputs it predicts.

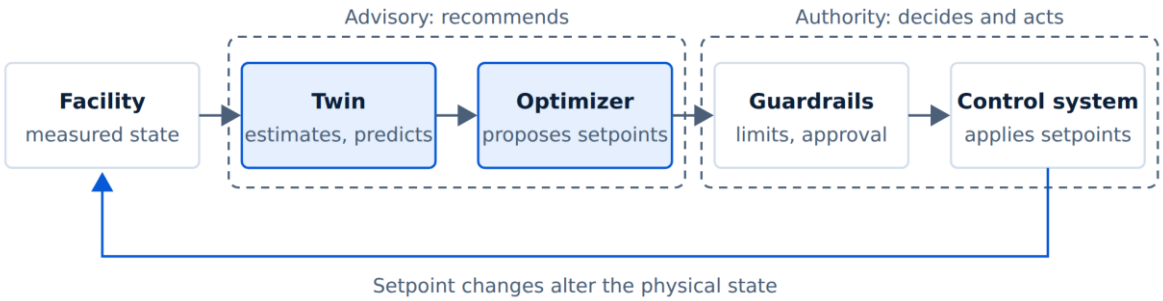


Figure 10.1. A guarded optimization loop. Recommendations come from the twin and optimizer, but the authority to act sits behind explicit limits, and often a human approval.

Sensible guardrails include hard bounds on each setpoint, a maximum rate of change, a requirement that predicted inlet temperatures stay a stated margin inside the envelope, automatic fallback to the standard control strategy when data quality drops or the model's uncertainty exceeds a threshold, and a watchdog that returns control if the twin stops responding.

What has been achieved

Google reported in 2016 that a machine-learning system trained on its operating data reduced the energy used for cooling in its data centers by around 40 percent in its tests. The system began by recommending settings and was later allowed to control them under operator supervision. Results like this depend on rich instrumentation, a consistent facility design and a large fleet to learn from. Many operators will see smaller gains, and how much you gain depends heavily on how far your site sits from a well-tuned baseline today. Measure your own baseline before promising anything.

Carbon and water

Electricity is not equally clean at all hours. With a forecast of grid carbon intensity and price, the twin can advise on when to shift flexible load, when to precool thermal storage and when to favor a technology with a different carbon or water profile. Water-side economizing and evaporative cooling trade electricity for water, so the right choice depends on local scarcity, and the twin should let you see the trade explicitly rather than optimizing one metric at the expense of another.

Regulators increasingly require reporting. In the European Union, for instance, operators of larger data centers must report on energy and related indicators. A twin that already holds metered, validated data can produce those reports with far less manual effort.

11. Designing for AI: density and liquid cooling

Dense accelerated computing is pushing air cooling to its limits and rewriting the assumptions many facilities were designed around. It is also where a twin is close to indispensable, because intuition built on lower densities stops working.

What changes

When a rack draws tens of kilowatts or more, moving enough air becomes impractical. The volume required, the fan energy and the pressure needed all climb steeply. Facilities increasingly turn to liquid, because liquid carries far more heat per unit volume than air. The common approaches are:

- **Direct-to-chip cooling:** cold plates on processors and accelerators carry heat into a liquid loop. Some heat still goes to the air from memory, power supplies and other components, so most designs are hybrid.
- **Rear-door heat exchangers:** liquid-cooled doors on the back of racks capture exhaust heat before it enters the room.
- **Immersion cooling:** equipment sits in a dielectric fluid, either single-phase or two-phase.

Each approach introduces new elements for the twin to model: coolant distribution units, secondary loops, manifolds, flow and pressure at each rack, coolant temperature, leak detection, and the interaction with the primary facility water system.

Modeling the liquid side

The physics is different from airflow. Liquid loops are governed by flow, pressure drop and heat exchanger effectiveness, and they are far easier to model accurately than turbulent airflow. The uncertainties move elsewhere: partial blockage, air in the loop, degraded flow on a single branch, fouling of heat exchangers, and the behavior of many parallel cold plates sharing a manifold.

Useful twin outputs for liquid systems include the predicted coolant temperature and flow at each rack, remaining thermal margin before a throttle point, which racks a unit failure would affect, and the effect of changing coolant supply temperature on both chip temperature and plant efficiency.

Transients are faster

A liquid loop holds much less thermal mass than a large air volume, and high-density racks generate heat in a small space. When cooling is interrupted, temperatures can rise within seconds to a few minutes. Ride-through arithmetic that worked for air-cooled halls does not carry over. The twin should model transients explicitly, and its results should feed design decisions such as thermal storage, redundant pumps, and how quickly the IT layer must be told to throttle.

Power behavior

Large accelerator clusters can produce rapid, correlated changes in load. A training job that pauses to synchronize can drop facility power sharply and then ramp it back up, and thousands of devices may do this together. The effects reach upstream equipment such as UPS systems, generators and the grid interface, and can also disturb cooling controls that assume slower changes. Model these swings using measured or vendor-provided workload profiles and test them against the electrical model, including under redundancy failures.

The twin as a design tool

Because equipment for these facilities has long lead times and high cost, mistakes are expensive to fix after construction. A twin built at design time can simulate layouts, test redundancy against failure scenarios, size cooling and distribution for planned and future densities, and establish the phasing of deployment. Several vendors and platform providers now offer simulation-ready templates and reference designs for large accelerated computing facilities, which can shorten the start of a project. Treat them as starting points to be calibrated against the actual equipment and site.

Retrofits

Many operators will convert existing halls rather than build new ones. A twin of the existing hall, calibrated with real measurements, can show what densities a given area can support with and without added liquid cooling, where structural or floor-loading limits appear, and how to phase a conversion so the remaining air-cooled equipment stays within its envelope during the work.

Operational hazards to model

- **Leaks:** where would fluid go, which equipment would it affect, and how quickly would sensing detect it?

- **Loss of flow:** how long until each rack reaches its throttle point, and what does the sequence of alarms look like?
- **Mixed environments:** how do air-cooled and liquid-cooled equipment interact through shared power, shared facility water and shared heat rejection?

> **Field note.** The equipment in these halls changes generation quickly, and vendors change specifications between revisions. Store the source and version of every performance figure you put in the twin, and re-validate against measurements when hardware changes.

Part IV. Delivering the twin

12. Roadmap, team and economics

13. Security, governance and risk

14. The road ahead

12. Roadmap, team and economics

Technology is the smaller half of a twin program. The larger half is choosing the right first problem, staffing it properly and showing, in money terms, that it worked.

A staged roadmap

The stages below are ordered by dependency, not by calendar. Durations depend on the site, the state of its data and how many people can work on it, so treat any timeline as an illustration.

1. **Frame.** Pick one or two use cases, name the person who makes each decision, record the current baseline and define success in measurable terms. Do this before selecting any technology.
2. **Instrument and connect.** Run the data audit from Chapter 4. Fill the most important sensor gaps, establish read-only connections and build the semantic model for the assets in scope.
3. **Model.** Build the spatial and topology models for the chosen area and reconcile them with reality. Deliver a descriptive view that operators find genuinely useful.
4. **Calibrate and validate.** Fit and test the physical and learned models against measurements. Run in shadow mode. Publish the measured accuracy.
5. **Advise.** Put recommendations into the workflows where the decisions are made: the placement request, the maintenance plan, the setpoint review. Measure adoption and outcomes.
6. **Automate selectively.** Where evidence supports it and the cost of error is bounded, grant limited authority with the guardrails from Chapter 13.

Choosing the first use case

A good first use case has five properties: the decision recurs often enough to measure, the data already exists or is cheap to add, the value can be measured in money or risk, the consequences of a wrong answer are contained, and a named owner wants it. Capacity placement (Chapter 8) and thermal-aware setpoint review (Chapter 10) usually satisfy all five. Whole-site autonomous control almost never does.

The team

A small team can start, but it needs a range of skills:

- a **product owner from operations**, who represents the people who will use the twin and has authority over their priorities;
- a **mechanical engineer and an electrical engineer** who understand how the facility really works and can sanity-check every model;
- a **data engineer** who owns ingestion, quality and the semantic model;
- a **simulation or modeling engineer** for physics-based and surrogate models, and where useful a **machine-learning engineer**;
- a **representative from IT infrastructure**, so workload and inventory data is in the room;
- a **security and OT specialist**, involved from the start rather than at sign-off.

Build, buy or blend

Approach	Strengths	Risks	Fits when
Commercial twin or DCIM platform	Fast start, packaged models and visualization, vendor support	Lock-in, limited customization, integration work still required	Fairly standard facilities and limited in-house data skills
Build on a general data and simulation platform	Flexibility, control and reuse across use cases	Needs a skilled team; slow to first value	Large operators with strong engineering depth
Blend: buy specialist tools, own the data and semantic layer	Balances speed and portability	Integration effort between components	Most operators

Whatever you choose, **own your semantic model and your data**. Simulation engines and dashboards can be replaced. The mapping between your points, assets and locations took months to build and is very expensive to rebuild. Insist on export in an open format as a contract requirement.

Economics

Costs come in five kinds: sensors and instrumentation, integration and data engineering, software licenses or platform costs, modeling and calibration labor, and the continuing cost of keeping the twin current. That last one is the one most often left out of business cases.

Benefits fall into a few categories:

- **Capacity released or deferred.** Building capacity is expensive, often in the range of several million to well over ten million dollars per megawatt depending on market and design. Freeing even a small fraction of stranded capacity can outweigh other benefits.

- **Energy and carbon.** Recurring, easily measured against a baseline, and growing more valuable as power gets scarcer and more expensive.
- **Avoided incidents.** Real, but hard to prove. Count near-misses caught and time to resolution, and be conservative about avoided-outage claims.
- **Labor and speed.** Faster deployment approvals, less time in troubleshooting and audits, shorter onboarding for new staff.

Build the case around a baseline you measured before the project started, and attribute benefits to specific decisions the twin informed. Be wary of savings that could equally have come from changes made anyway.

Common failure patterns

- **Visualization first.** A beautiful 3D view with nothing behind it is a demo, not a twin.
- **Boiling the ocean.** Trying to model every system in every site before delivering anything.
- **No owner.** The model decays because nobody is responsible for keeping it current.
- **Uncalibrated trust.** Presenting model outputs as fact, without stating accuracy, so the first visible error destroys credibility.
- **Ignoring change management.** Operators who were not consulted quietly stop using the tool.
- **Lock-in by accident.** Letting the vendor's proprietary tags become the only description of your assets.

13. Security, governance and risk

A twin sits between IT systems and the operational technology that keeps the lights on. Done carelessly it becomes a new way in for attackers and a new source of confident mistakes. Done well it can make both more visible.

Architecture for safety

The default posture should be **read-only**. Data flows from control systems into the twin through a controlled gateway, and nothing flows back unless a specific, reviewed pathway exists. Segment networks using established industrial-security models, with the twin and its cloud or enterprise components in a zone separated from the control network, and use dedicated, monitored, minimum-privilege interfaces for any write path. The IEC 62443 series is the widely used framework for industrial control system security, and its concepts of zones, conduits and security levels apply directly.

Design the twin so that **its failure never harms the facility**. If the twin, its network link or its data feed goes down, the control system must continue on its own, unchanged. This is easiest to guarantee when the control system never depends on the twin for its basic operation.

Earning authority in steps

Define explicit authority levels and the evidence needed to move between them:

Level	What the twin can do	Evidence to advance
0. Observe	Read data, display state and history	Data quality and coverage meet agreed thresholds
1. Advise	Recommend actions to humans	Shadow-mode predictions match outcomes within stated accuracy
2. Approve	Prepare changes that a human approves and applies	Recommendations accepted and successful over a sustained period
3. Act within bounds	Change specified setpoints within hard limits and rates	Independent review, a tested fallback, and a demonstrated safe failure mode

Demotion should be as easy as promotion. Define in advance the conditions that automatically drop a system back a level: degraded data quality, model uncertainty above a threshold, a mismatch between predicted and measured behavior, or an operator override. Provide a clearly labeled manual stop that any operator can use.

Model governance

Treat models as controlled artifacts:

- **Version everything:** the model, its parameters, the data it was calibrated on and the software that ran it.
- **Record validity ranges.** Every model works within conditions. State them, and make the twin say so when it is asked about a case outside them.
- **Monitor drift.** Track prediction error over time. Rising error is a signal to recalibrate or retire the model.
- **Keep an audit trail:** who or what recommended what, on which inputs, and what was done.
- **Review after incidents.** Any incident where the twin was involved, or absent, is a chance to improve it.

Data governance

Twins collect detailed data about power, workload and location. In multi-tenant facilities, a customer's power draw and equipment can be commercially sensitive, so define who sees what, and enforce it in the platform rather than in a document. Limit retention to what use cases need, minimize any personal data, and be clear about where data is stored and processed, especially when using hosted services in a different jurisdiction from the site.

Suppliers and software

The twin adds software components, many from third parties. Keep an inventory of them, apply patches on a defined schedule, control remote vendor access and review the security posture of any hosted service. When choosing a provider, ask how they isolate customers, how they handle credentials for your control systems, what they log and how you can export your data if you leave.

Human factors

People trust automation too much when it is usually right and too little after it is visibly wrong. Present predictions with uncertainty, show the reasoning behind recommendations, and make it easy to disagree. Train operators on the twin's limits as well as its features, and design alerts to be few and actionable. An operator who has learned to ignore the twin is a failure that no dashboard will reveal.

Certification and contracts

Changes to how a facility operates can affect certifications, insurance conditions and customer service-level agreements. Involve the people who own those commitments early, and document how the twin's recommendations are validated and approved. If you operate to a certified tier

or a contractual availability target, make sure the twin supports the evidence you must provide, not just the outcomes you hope to achieve.

14. The road ahead

The direction of travel is clear even if the pace is not: more automation, more integration with computing and the grid, and interfaces that let more people ask questions of the facility. Each brings both promise and new ways to go wrong.

From advice to autonomy, unevenly

Some functions will move to bounded autonomy first: cooling setpoints and plant sequencing where the physics is well understood, the constraints are clear and the cost of a bad move is small. Others, such as decisions affecting redundancy, will remain human-approved for a long time. The winning pattern is a twin that knows which is which and enforces it.

Language interfaces

Large language models make it plausible to ask a facility questions in plain words: “which racks in Hall 2 would exceed their inlet limit if we lost CRAH 6?” or “why did the alarm storm start at 02:10?” The reliable way to build this is to let the language model translate a question into calls to the twin’s deterministic tools, such as its graph queries and simulations, and then explain the results. Do not treat a language model as a source of physical truth. It can be fluent and wrong, and its output should always be traceable to a model run or a measurement. Agent-style systems that propose or execute changes need the same guardrails and authority levels as any other automation, and arguably stricter ones.

The grid at the other end

Data centers are becoming significant loads on power systems, and grid operators increasingly care about how flexibly they can behave. A twin that knows which workloads can shift, how much thermal storage the facility has and what its on-site batteries and generators can do is the natural basis for participating in demand response, providing ancillary services, or coordinating with a nearby renewable source. Doing this safely depends on the electrical and thermal models of Part II being trustworthy under unusual operating patterns.

Campus and fleet twins

Operators with many sites want to compare them, place workloads between them and plan capacity across them. Federated twins, where each site maintains its own detailed model and shares a standard summary, may scale better than a single central model. The prerequisite is a common semantic vocabulary, which is why early attention to open schemas pays dividends later.

Lifecycle and embodied impact

A twin that begins at design can follow a facility through construction, operation, upgrade and decommissioning. That continuity allows accounting for embodied carbon in materials and equipment alongside operational emissions, and it preserves knowledge about what is behind each wall. Expect increasing pressure to report on the whole lifecycle, not only operating energy.

Better sensing

Cheaper wireless sensors, thermal imaging, and mobile platforms that survey rooms can fill gaps in coverage and check the model against the room. Where these sources are trustworthy and well-calibrated, they reduce the cost of keeping a twin honest.

Standards and interoperability

Progress depends on shared formats: common semantic models for assets and points, standard ways to exchange simulation models and results, and open interfaces for control and data. Buyers have real leverage here. Specifying open standards in procurement, and refusing to accept proprietary lock-in for core descriptions of your own assets, will shape the market.

What to do on Monday

1. Write down the single most expensive decision your team makes repeatedly, and who makes it.
2. List the data you would need to make it better, and check whether that data exists and can be trusted.
3. Choose one hall or system and give one person ownership of its model.
4. Start the semantic model, even in a spreadsheet, with stable IDs and units.
5. Run a small pilot in shadow mode, publish its accuracy, and let the results decide what comes next.

Appendices

Appendix A. Glossary

Appendix B. Interface cheat sheet

Appendix C. A sample data model

Appendix D. Pilot checklist

Appendix E. Standards and further reading

Appendix A. Glossary

Short definitions of the terms used in this book.

ATS (automatic transfer switch)

Switches a load between power sources, typically utility and generator, when one fails.

BACnet

A communication protocol widely used in building automation and cooling plant controls.

BMS (building management system)

The control and monitoring system for mechanical plant and building services.

CDU (coolant distribution unit)

Equipment that transfers heat between a facility water loop and a secondary loop serving racks, and controls that loop's flow and temperature.

CFD (computational fluid dynamics)

Numerical simulation of fluid flow and heat transfer, used to predict airflow and temperatures in a hall.

CRAH and CRAC

Room-level cooling units. A CRAH uses chilled water; a CRAC uses a refrigerant-based compressor.

CUE (carbon usage effectiveness)

Emissions associated with a facility's energy use, per unit of IT energy.

DCIM

Data center infrastructure management: software for asset, capacity, power and environment records.

EPMS

Electrical power monitoring system covering switchgear, UPS, PDUs and meters.

Fidelity

How closely a model reproduces reality in the ways that matter for its purpose.

N+1, 2N

Redundancy schemes. N+1 provides one spare unit beyond the number needed; 2N provides two fully independent sets, each able to carry the whole load.

N−1 analysis

Testing whether the system stays within its ratings after the loss of any single component.

PDU and RPP

Power distribution unit and remote power panel: equipment that distributes power from the UPS toward racks.

PUE (power usage effectiveness)

Total facility energy divided by IT equipment energy.

RCI and RTI

Rack Cooling Index and Return Temperature Index: metrics for how well a room delivers cooling and manages airflow.

Redfish

A standard REST-based interface for managing servers and related equipment.

Reduced-order model

A simplified model that keeps the dominant behavior of a complex system and runs much faster.

Residual

The difference between a measured value and the model's prediction of it.

Semantic model

A structured description of what each data point means: its asset, quantity, unit and location.

Shadow mode

Running a model's recommendations alongside real operations without acting on them, in order to score them.

Stranded capacity

Installed capacity of one resource that cannot be used because another resource has run out.

Surrogate model

A fast approximation, often learned from many runs of a slower simulation.

UPS (uninterruptible power supply)

Equipment that conditions power and bridges short interruptions, usually with batteries.

WUE (water usage effectiveness)

Site water use per unit of IT energy.

Appendix B. Interface cheat sheet

A quick reference to the interfaces you are most likely to meet when connecting a data center to a twin.

Interface	Typical use	Points to watch
BACnet	Cooling units, chillers, pumps, building controllers	Object naming varies by installer. Confirm units and update behavior.
Modbus (serial or TCP)	Power meters, PDUs, drives, some cooling units	Register maps differ by device. No built-in security, so isolate it.
SNMP	UPS, PDUs, network equipment	Prefer the version with authentication and encryption. Poll rate affects device load.
Redfish	Server power, temperature, fans, inventory	Coverage varies by vendor and firmware. Test each hardware generation.
IPMI	Legacy server management	Older, with known security concerns. Use Redfish where available.
OPC UA	Structured industrial data exchange	Supports security and information models. Useful as a gateway layer.
MQTT	Lightweight publish and subscribe between edge and platform	Agree topic structure, payload format and quality of service up front.
Streaming telemetry (for example gNMI)	High-rate network device data	Volume can be large. Filter at the source.
REST APIs	DCIM, CMDB, ticketing, schedulers, cloud platforms	Rate limits, pagination and authentication expiry.

Appendix C. A sample data model

An illustrative record for one cooling unit and two of its points. It is not a standard, but shows the information a semantic model needs to carry.

```
{
  "asset": {
    "id": "crah-h3-07",
    "type": "CRAH",
    "location": { "site": "dc1", "hall": "h3", "zone": "row-c" },
    "serves": ["cold-aisle-c2"],
    "fed_by": { "power": "rpp-h3-b-04", "water": "cdu-loop-2" },
    "rating": { "cooling_kw": 120, "airflow_m3_per_h": 24000 }
  },
  "points": [
    {
      "id": "crah-h3-07.supply_air_temp",
      "quantity": "supply_air_temperature",
      "unit": "degC",
      "source": "bms/bacnet:AI-0342",
      "sample_seconds": 30,
      "writable": false
    },
    {
      "id": "crah-h3-07.fan_speed",
      "quantity": "fan_speed",
      "unit": "percent",
      "source": "bms/bacnet:AO-0110",
      "sample_seconds": 30,
      "writable": false
    }
  ]
}
```

Note three habits this example encodes. Identifiers are stable and human-readable. Units are explicit and normalized. And every point is marked read-only by default, so that a write path has to be a deliberate, reviewed exception.

Appendix D. Pilot checklist

Use this list to test whether a pilot is ready to start, and again before it is called a success.

Before you start

☐One or two use cases are chosen, each with a named decision-maker.

☐A baseline is recorded for every metric you intend to improve.

☐Success and stop criteria are written down and agreed.

☐One person owns keeping the model current.

☐Security and OT staff have reviewed the design and the data paths.

☐The data audit is complete, with gaps ranked by impact.

During the pilot

☐Every point has an asset, quantity, unit, location and source recorded.

☐Data quality checks run automatically and their results are visible.

☐The topology model has been reconciled with a physical audit of a sample of racks.

☐Thermal and electrical models have been calibrated on one dataset and tested on another.

☐Prediction accuracy is measured and published, with uncertainty.

☐The twin runs in shadow mode before it influences any decision.

Before expanding

☐The measured benefit is attributed to specific decisions.

☐Operators have been asked what they would change, and the answers are recorded.

☐The cost of keeping the twin current has been measured, not guessed.

☐Authority levels, fallbacks and demotion conditions are documented.

☐Data and semantic models can be exported in an open format.

Appendix E. Standards and further reading

Names of standards, bodies and open projects mentioned in this book, so you can find current editions and documentation. Editions change; always check the latest version.

Name	Relevance
ASHRAE Technical Committee 9.9	Thermal guidelines for data processing environments, including equipment classes and inlet temperature envelopes.
ISO/IEC 30134 series	Key performance indicators for data centers, including PUE, WUE and CUE.
The Green Grid	Origin and further guidance on PUE, WUE and CUE and related metrics.
Uptime Institute Tier Standard	Classification of data center topology and operational sustainability.
EN 50600 series	European standards for data center facilities and infrastructures.
TIA-942	Telecommunications infrastructure standard for data centers.
Open Compute Project	Open hardware and facility specifications, including work on advanced cooling.
Digital Twin Consortium	Definitions, reference architectures and use-case work for digital twins in general.
ISO 23247	Digital twin framework for manufacturing, often borrowed as a reference architecture.
IEC 62443 series	Security for industrial automation and control systems.
Brick Schema and Project Haystack	Open metadata models for describing building and equipment data.
DMTF Redfish	Standard management interface for servers and infrastructure.

Where to look next

For the physics, search for the published literature on data center CFD, reduced-order airflow models and chiller plant optimization. For the data side, look at documentation for Brick, Haystack and OPC UA information models. For operations, use your equipment vendors' documentation and the incident reviews your own organization has written. The last is usually the most valuable source of all.

About Anomny

Anomny designs, builds and manages data centers.

> **Designing, building or running a data center?**
>
> Write to the presales team about your facility, your plans and what you would like a digital twin to help with.

Website: anomny.com

Presales: presales@anomny.com

Worked examples use illustrative numbers. Verify every figure, limit and standard against your own site, your equipment documentation and the current edition of the relevant standard before acting on it.