



---

## Wireless & Occupancy Intelligence

A wireless engineer's walk through every occupancy feature CrossConnect ships: how the Wi-Fi you already run becomes a trustworthy, privacy-respecting picture of how your spaces are used, the exact math behind every number, and where the technology is heading next, all the way to 802.11bf Wi-Fi sensing.

**Audience:** wireless / network engineers, facilities and space planning, security and privacy reviewers

**Scope:** sources, the occupancy engine, zones and calibration, every metric, presence and dwell, scheduling, planning, coverage, energy, privacy, delivery, and 802.11bf Wi-Fi sensing

**Posture:** aggregate counts only, no personal data, a confidence band on every figure, advisory

**Document:** engineering reference, 24 June 2026

**Contact:** [contact\\_us@cybriq.io](mailto:contact_us@cybriq.io)

## 0 How to read this document

---

I am going to take you from the radio to the rooftop dashboard, one honest step at a time. CrossConnect measures occupancy from the wireless network you already own. No cameras, no badge readers, no motion PIRs, and it never learns who anyone is. It counts the devices associated to your access points, turns those counts into a careful estimate of people, and stamps every figure with how much you should trust it. Each section below explains one piece in plain language, shows you the actual formula we use, and draws the picture. Read it top to bottom and you will understand the whole system; jump to a section and it will stand on its own.

**PRODUCTION** shipped and working today

**CONFIGURABLE** shipped, operator-enabled

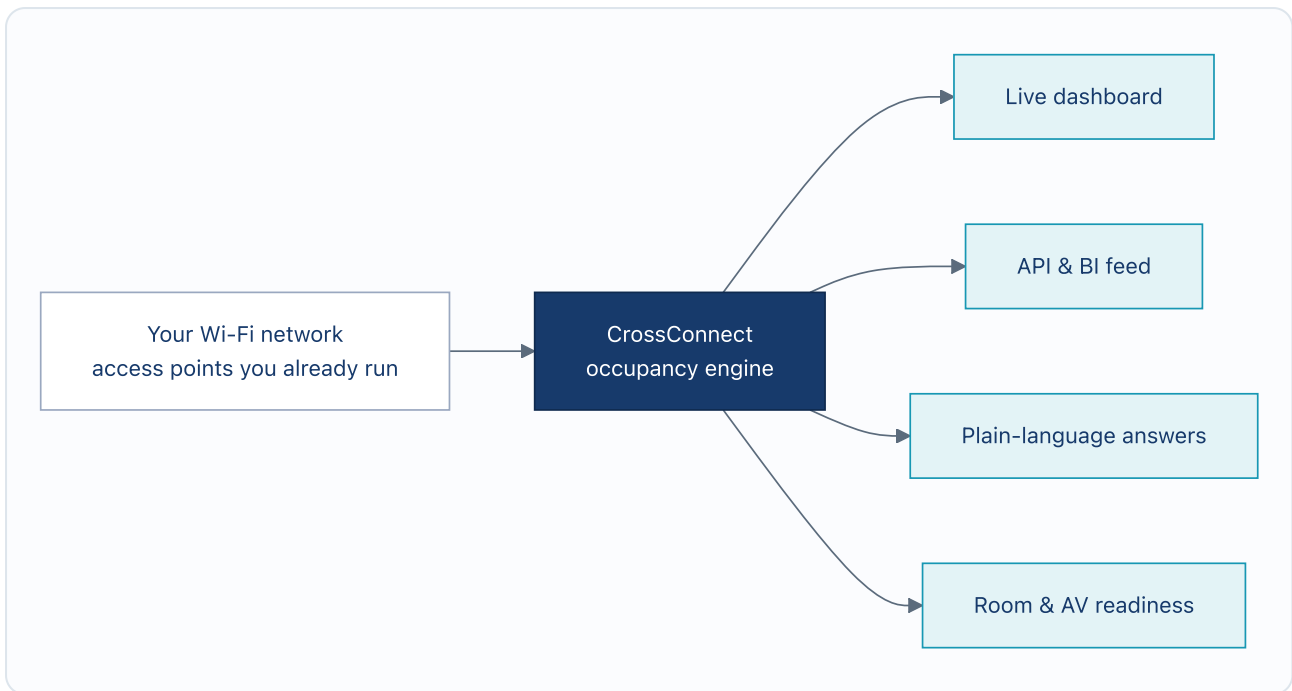
**EXPERIMENTAL** a preview, advisory only

- |   |   |
|---|---|
| <b>1</b> What it does, and how it fits            | <b>12</b> Energy & carbon opportunity           |
| <b>2</b> The wireless sources                     | <b>13</b> Coverage & RF                         |
| <b>3</b> From devices to people                   | <b>14</b> Confidence & honest numbers           |
| <b>4</b> Calibration: teaching it your building   | <b>15</b> Privacy & data handling               |
| <b>5</b> Zones, geometry & AP placement           | <b>16</b> Delivery & integration                |
| <b>6</b> The metrics, and what each one tells you | <b>17</b> The switch-derived complement (Sense) |
| <b>7</b> Presence & dwell                         | <b>18</b> 802.11bf Wi-Fi sensing                |
| <b>8</b> Patterns & the week-ahead forecast       | <b>19</b> What ships today vs roadmap           |
| <b>9</b> Scheduled vs actual                      | <b>A</b> Formula reference                      |
| <b>10</b> Space planning & recommendations        | <b>B</b> Data model & retention                 |
| <b>11</b> Find a seat                             |   |

### 1 What it does, and how it fits **PRODUCTION**

---

Here is the whole idea in one breath. Every access point already knows how many client devices are talking to it right now. CrossConnect reads that number, figures out which room each access point lives in, adds the rooms up into floors and buildings, divides devices by a calibrated ratio to get an estimate of people, and writes a clean reading every fifteen minutes. That one engine then feeds a live dashboard, a read-only API, a plain-language assistant, and the room-readiness picture the rest of the platform uses. You did not install anything new in the ceiling. You are reading a sensor you already paid for.



**Figure 1. One engine, many windows.** The Wi-Fi you already run feeds a single occupancy engine; every screen, feed, and answer reads from the same record.

Four promises run through everything that follows. Keep them in mind and the rest of the document makes sense.

**Counts, never identities**

The unit is an associated device, aggregated to a zone. No MAC address, no client id, no personal data is ever stored. Identity is dropped at the door, before anything is written.

**People, with a stated confidence**

A calibrated devices-per-person ratio turns device counts into a headcount, and every figure carries a HIGH, MEDIUM, or LOW band so a rough number is never mistaken for a precise one.

**Real first, never a mock**

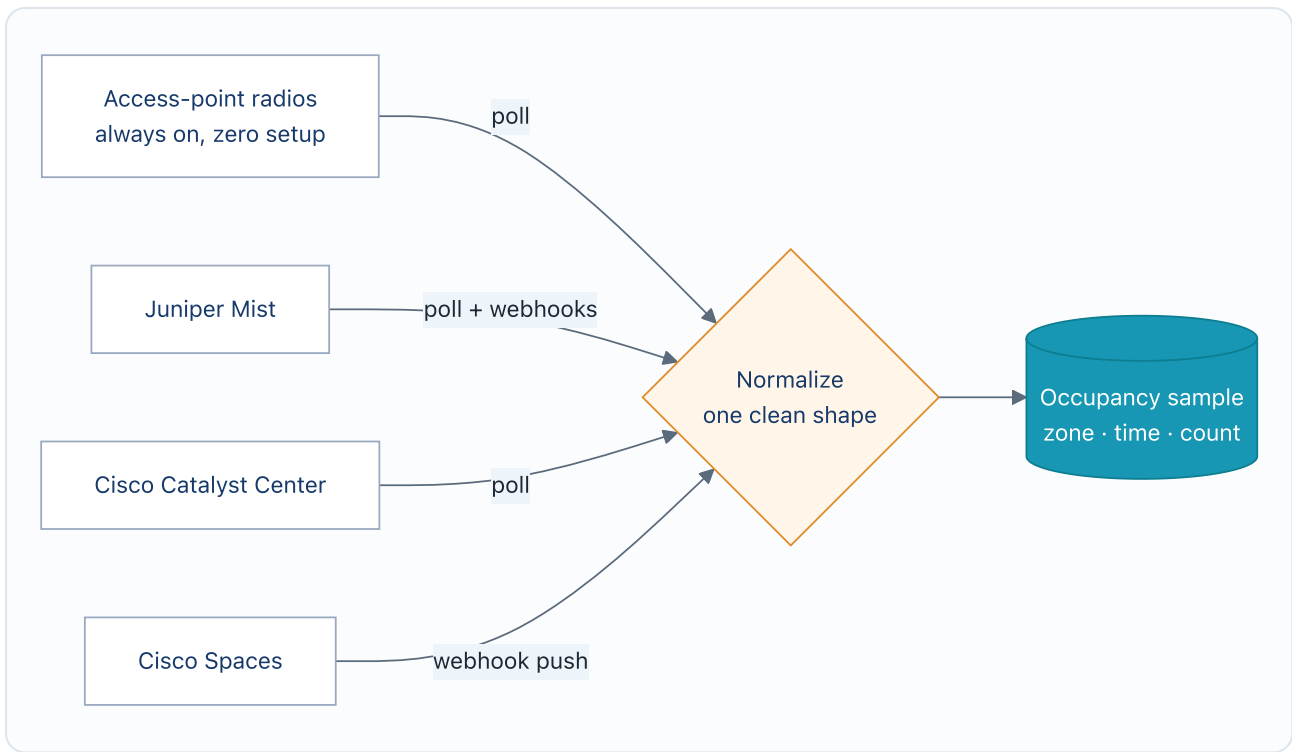
If a richer vendor feed is connected it is used; if not, the engine still produces occupancy from the access-point poll the platform already runs. It never waits on an integration to start working.

**Part of one model**

Zones, floors, and buildings are the same source-of-truth locations the rest of CrossConnect uses. Occupancy is not a bolt-on silo; it sits next to coverage, RF, and the bill of materials, so a weak reading can hand you the fix.

**2 The wireless sources** PRODUCTION

Four kinds of source feed the engine. They look different on the wire, but every one of them is boiled down to the same simple shape before it is stored: a zone, a timestamp, and a device count. The vendor-specific code stops at the parser. Everything downstream is identical no matter where the number came from.



**Figure 2. Many radios in, one record out.** Polls and pushes from any supported source are normalized into one append-only series. The baseline access-point poll means there is always a signal, even with nothing else connected.

Source	How it arrives	What it gives us	Access
<b>Access-point radios the baseline</b>	The controller poll the platform already runs, every cycle.	Associated client count per radio, plus band, channel, and transmit power. On Cisco AireOS this is the SNMP OID <code>bsnAPIfLoadNumOfClients</code> ; on Catalyst 9800 it is the RESTCONF field <code>num-assoc-clients</code> (the 9800 dropped the old client-count OID).	Read-only, no extra config.
<b>Juniper Mist</b>	Poll the site zones API every 15 minutes; optional WebSocket for live coordinates; webhooks for zone enter/exit.	<code>GET /sites/:id/zones</code> returns per-zone client counts and average dwell. Enter/exit webhooks feed the dwell engine.	Org API token, Observer (read-only) role, encrypted at rest.
<b>Cisco Catalyst Center</b>	Poll the client trend API every 15 minutes.	True 15-minute and hourly client-count buckets from <code>clients/trendAnalytics</code> . We keep our own copy because Catalyst only retains a couple of weeks.	<code>X-Auth-Token</code> , encrypted at rest.

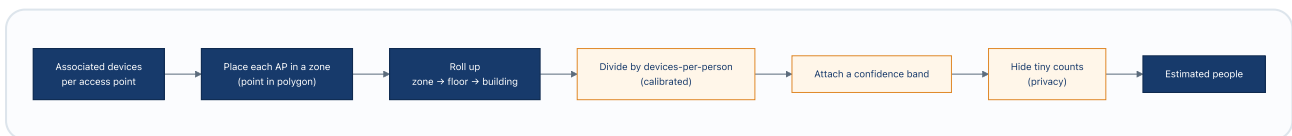
Source	How it arrives	What it gives us	Access
Cisco Spaces	Spaces pushes events to our webhook (Firehose).	DEVICE_COUNT and SPACE_OCCUPANCY events, matched to a space by name.	Issued key in X-API-Key, validated strictly. A tenant id is not a secret.
Calendar / scheduling	CSV import today (location, expected people, start, end); a direct 25Live / Series25 pull is on the roadmap.	The room booking schedule, used in Section 9 to compare what was booked against what actually happened.	Operator import, advisory only.
Floor plans & room inventory	GeoJSON / IMDF import; CAD/DWG conversion is done upstream on purpose.	Zone polygons, room numbers, and (joined from an IWMS like Archibus or TRIRIGA) the authoritative room name, type, and rated capacity.	Operator import.

**A small engineering honesty.** Mist's polished 15-minute heatmap lives in Premium Analytics, which has no API. So we do not screen-scrape it. We rebuild our own 15-minute buckets from the /zones poll and the zone webhooks. Every source, no matter how fancy its own dashboard, lands in the same humble row: zone, time, count, and where it came from.

Every collector is read-only, dormant until you give it a credential, and built to fail quietly: a five-second connect timeout, a ten-second read timeout, and on any error it logs and returns nothing rather than throwing. If a vendor feed goes dark, the access-point baseline keeps the lights on.

### 3 From devices to people PRODUCTION

This is the heart of it, so let me slow down. A radio hands us a device count. We need a people count. Five steps get us there, and I will show you the math at each one.



**Figure 3. The estimate pipeline.** Device counts become a people estimate through placement, roll-up, calibration, a confidence band, and a privacy guard, in that order.

**Step one, place the radio.** Each access point sits on a floor plan at an (x, y) coordinate. A tested point-in-polygon routine asks "which room polygon contains this point?" and assigns the AP to exactly one zone. If we do not have geometry for an AP yet, it falls back to its nominal location, so a partial setup still works.

**Step two, roll it up.** Each zone's devices are added up the location tree, zone into floor into building into campus, so you can ask the question at any altitude. We only sum the leaf zones for the estate total, so nobody gets counted twice.

**Step three, divide by the ratio.** People do not carry exactly one device. They carry a phone, maybe a laptop, maybe a watch. So we divide the device count by a *devices-per-person* ratio that we measured for your building (Section 4 is all about measuring it).

```
estimatedPeople = round( associatedDevices ÷ devicesPerPerson )
// 0 if there are no devices. If the ratio is somehow ≤ 0, fall back to 1.0
// (one device ≈ one person) instead of dividing by zero. Never negative.
```

Until a building is calibrated, the ratio defaults to `1.0` (one device equals one person) and the reading is automatically marked LOW confidence. That is deliberate: an honest rough number, clearly labelled, beats a confident wrong one.

**Step four, attach a confidence band,** covered in Section 14. **Step five, guard small numbers,** covered in Section 15. By the time a people estimate reaches a screen it has already passed through privacy.

## 4 Calibration: teaching it your building PRODUCTION

The devices-per-person ratio is the one number that turns a generic counter into *your* counter. We learn it the old-fashioned way: someone walks a space during a busy hour and writes down the real headcount a few times. We pair each headcount with the device count the radios saw at that moment, and we pool them.

```
devicesPerPerson = Σ(associatedDevices) ÷ Σ(groundTruthHeadcount)  across all
observations
= max(0.1, ratio) // never below
0.1
```

We sum the totals rather than averaging per-sample ratios, because pooling is far more robust to the noise in any single count. If we have no usable ground truth yet, the ratio stays at `1.0` and confidence stays LOW.

We also report how good the fit is, so you can trust the calibration itself. The model is the simplest possible straight line through the origin, `devices ≈ factor × people`, and we score it with R-squared: the fraction of the variation the ratio explains, on a 0 to 1 scale.

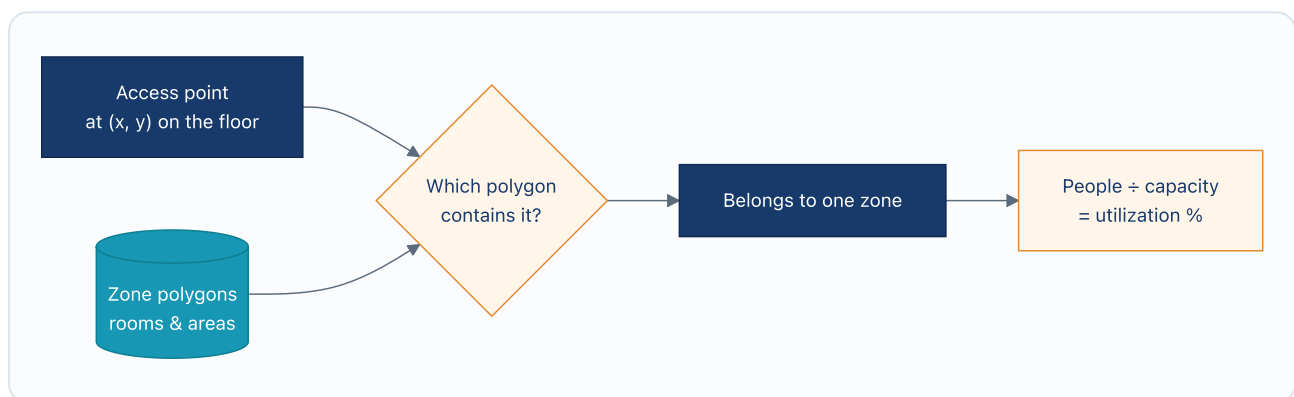
```
R² = clamp01( 1 - ssRes ÷ ssTot )
ssRes = Σ ( associatedDevices - devicesPerPerson × groundTruthHeadcount )² // what
the ratio misses
ssTot = Σ ( associatedDevices - mean(associatedDevices) )² //
total spread
```

An R-squared near 1.0 means the ratio predicts your device counts almost perfectly; near 0 means it is no better than guessing the average. We require at least two observations and some real spread before we claim any fit. Each calibration is stamped with the ratio, the R-squared, and a 90-day validity window; after that, confidence quietly drops until you re-measure. In the field this is usually a one-to-two-week ground-truth pass, then it just runs.

**Where it is strong, where it is humble.** Aggregated to a building or a floor, this approach is genuinely strong, with published work showing R-squared around 0.95. At the single small-room level it is weaker, on the order of a few people of error, which is exactly why per-site calibration matters and why every figure carries a confidence band. We count *associated* clients on your secured network, not probe requests, which sidesteps the MAC-randomization problem that limits probe-sniffing footfall products to capturing only six to twelve percent of real visitors.

## 5 Zones, geometry & AP placement PRODUCTION

Occupancy is a spatial idea, so the system is spatial first, not a spreadsheet with room names. A zone is a real polygon on a floor plan, with an optional area and a rated capacity. An access point is a point on that same plan. The point-in-polygon test is what marries the two.



**Figure 4. Geometry does the bookkeeping.** Each access point resolves to exactly one room by polygon containment; capacity turns a headcount into a utilization percentage.

Capacity is what makes a raw count meaningful. Twelve people in a fourteen-seat control room is a packed room; twelve people in a ninety-seat atrium is nearly empty. The conversion is the most basic one in the system.

```
utilization% = max( 0, 100 × people ÷ capacity ) // blank if no capacity is set
```

Zone polygons come in through a GeoJSON / IMDF importer. We deliberately keep CAD/DWG conversion upstream, where mature tools (FME, ArcGIS Indoors) already turn architectural drawings into clean room polygons, and we join your space-management system for the authoritative room name, type, and seat count. When a floor plan is revised, the re-import is an operator-confirmed proposal, never a silent overwrite. A wiring-closet marker layer (IDF and MDF) rides on the same floor coordinates and feeds the cable-run math in the coverage tools.

## 6 The metrics, and what each one tells you PRODUCTION

Now the payoff. Once a clean series exists, the engine computes a small, deliberate vocabulary of metrics over any window you ask for. We chose these names on purpose to match how facilities and space-planning teams already think, with one column the competition does not have: a confidence band on every single one. Here is every metric, what it answers, and exactly how it is computed.

Metric	What it tells you	How it is computed
<b>Estimated people</b>	How many people are in the space right now.	Devices ÷ devices-per-person, per Section 3.
<b>Peak</b>	The busiest it got, and when.	The maximum estimated people in the window; ties go to the earliest time.
<b>Average</b>	The typical level of use.	$\Sigma \text{ people} \div \text{number of intervals}$ .
<b>Fill Rate</b>	How full versus the seats you have. Peak Fill and Average Fill.	$100 \times \text{people} \div \text{capacity}$ , at the peak and on average. Can exceed 100% (over capacity).
<b>Occupied Rate</b>	What share of the time the space was actually in use.	$100 \times \text{intervals with } \geq 3 \text{ people} \div \text{total intervals}$ . The threshold (default 3) is configurable.
<b>Intensity</b> People-Minutes / m <sup>2</sup>	How hard the floor space works, independent of room size. Good for comparing a big hall to a small office.	$\Sigma(\text{ people} \times \text{interval minutes} ) \div \text{floor area (m}^2\text{)}$ .
<b>Dwell</b>	How long people stay (average and median).	Mean and median of merged visit lengths, per Section 7.
<b>Pass-through ratio</b>	How much of the traffic is just walking through versus staying.	$100 \times \text{visits under 5 minutes} \div \text{total visits}$ .
<b>Confidence</b>	How much to trust the figures above.	The worst band of any sample in the window: one LOW makes the window LOW (Section 14).

The two intensity formulas deserve a second look because they are the ones planners actually fight over.

```

avgPeople           =  $\Sigma \text{ estimatedPeople} \div n$ 
occupiedRate%      =  $100 \times (\text{intervals with people } \geq 3) \div n$ 
peakFill%          =  $100 \times \text{peak} \div \text{capacity}$ 
avgFill%           =  $100 \times \text{avg} \div \text{capacity}$ 
peopleMinutesPerM2 =  $\Sigma(\text{ people} \times (\text{intervalSeconds} \div 60) ) \div \text{areaM}^2$ 

```

**A unit note, because we would rather tell you.** The area column is historically named "area\_sqft," but the importer actually stores square *metres* (the polygons are measured in metre vertices). So intensity is honestly labelled per square metre everywhere. If you ever need square feet, the factor is ×10.764.

To make this concrete, here is the metric set computed on a live 72-hour window for a working broadcast facility. Read it like a wireless engineer would: Studio A is genuinely over capacity (122% peak fill), and the Atrium is a ninety-seat room that never breaks twelve percent, which is a right-sizing conversation waiting to happen.

Zone	Capacity	Peak	Avg	Occupied Rate	Peak Fill	Avg Fill	Intensity (per m <sup>2</sup> )
Studio A	60	73	38.2	100%	122%	64%	1528

Zone	Capacity	Peak	Avg	Occupied Rate	Peak Fill	Avg Fill	Intensity (per m <sup>2</sup> )
Studio B	45	40	20.8	100%	89%	46%	833
Newsroom	50	32	16.5	100%	64%	33%	510
Control Room	14	14	7.1	79%	100%	51%	263
Atrium	90	11	6.0	76%	12%	7%	217
Edit Bays	18	5	2.8	54%	28%	16%	111

## 7 Presence & dwell PRODUCTION

Counting people is a snapshot. Dwell is the story over time: how long do people actually stay? When a source emits enter and exit events (Mist does today), we pair them into a visit and store only the zone, the start, the end, and the duration. Never an identity. The one subtlety is human behaviour: people step out for coffee and come back. We do not want that to read as two short visits, so we merge any sessions whose gap is small.

```
For sessions sorted by start time:
  if next.enter ≤ runningEnd + mergeGap → extend the current visit
  else → close it, start a new one
dwell = sum of the merged visit lengths // mergeGap = 300 seconds (5 minutes)
```

From those merged visits we report average dwell, median dwell, and the pass-through ratio (the share of visits shorter than five minutes, which is how you tell a corridor from a meeting room). On the same live facility, the dwell engine already shows an average stay of about 29.5 minutes across the populated zones.

## 8 Patterns & the week-ahead forecast PRODUCTION

People are creatures of habit, and buildings breathe on a weekly rhythm. The patterns view collapses your history into a grid of 168 cells, one for every hour of every weekday, and colours each by how full the space typically is at that hour. Tuesday at 10am lights up; Friday at 4pm fades. It is the single most useful picture for scheduling.

```
cell[weekday][hour] = average estimated people for that slot
cell colour          = 100 × avgPeople ÷ capacity // the Fill % for that
slot
peak slot           = the highest non-suppressed cell
```

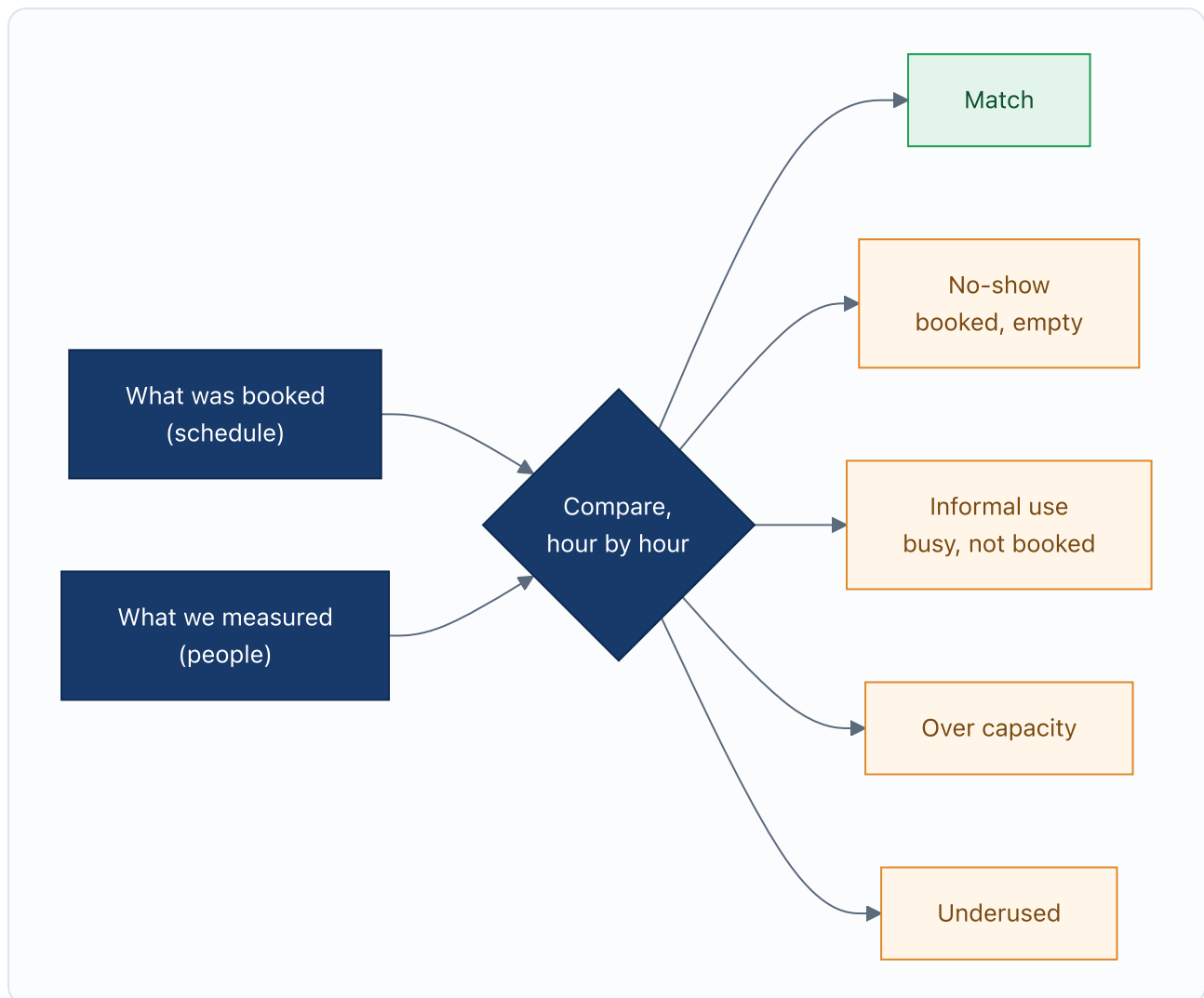
From that same grid we offer a deliberately simple, honest forecast. We do not pretend to do deep learning on a building's calendar. We take the typical level for each weekday-hour and nudge it by the recent trend.

```
trend%           = 100 × (last 7 days - prior 7 days) ÷ prior 7 days
predictedPeak    = round( slotAverage × (1 + trend% ÷ 100) )
confidence       = HIGH if ≥ 14 samples behind the slot, MEDIUM if ≥ 4, else LOW
```

That is the whole forecast: last week's shape, tilted by this fortnight's direction, with a confidence that honestly reflects how much history stands behind it.

## 9 Scheduled vs actual PRODUCTION

This is where occupancy earns its keep with facilities and registrars. Import the booking schedule, overlay it on what we actually measured, and label each hour. It writes nothing back to your calendar; it is pure, advisory comparison.



**Figure 5. Booked versus busy.** Each hour is labelled by comparing the reservation to the measured headcount. No-shows and informal use are the two findings rooms-bookers care about most.

```
scheduled = sum of overlapping bookings for the hour
actual     = measured people nearest the hour

if scheduled = 0 → UNSCHEDULED if actual > 3, else MATCH
if actual ≤ 3 → NO-SHOW // booked, but empty
if actual < 0.5 × scheduled → UNDERUTILIZED // under half the booking showed
if actual > 1.5 × scheduled → OVERFULL // half again more than booked
otherwise → MATCH
```

Roll those labels up to a term or a semester and you get an attendance-validation report: registered versus occupied, building by building, with the no-shows that free up rooms and the informal use that reveals demand the schedule never captured.

## 10 Space planning & recommendations PRODUCTION

A pile of metrics is not a plan. The planning engine reads the estate and hands you a short, ranked list of things worth doing, each with the evidence behind it and a one-click way to act. The rules are simple on purpose, so you can always see why a recommendation appeared.

```
over-capacity           : utilization > 100%           // priority 0, fix first
low-confidence-coverage: confidence LOW and fewer than 2 APs // priority 1, the
closed loop
right-size              : 0% ≤ utilization < 15%      // priority 2, under-
used
low-confidence         : confidence LOW but enough APs // priority 3,
calibrate or refresh
```

The second rule is the one no occupancy-only tool can offer, and it is the reason occupancy lives next to coverage in this product. If a room reads LOW only because it is thin on access points, we do not just shrug; we send you straight to the coverage map and the bill of materials to add an AP. The occupancy problem and the wireless fix are the same screen. That is the whole argument for building these two things in one platform.

## 11 Find a seat PRODUCTION

The same numbers, pointed at a person standing in a hallway with a laptop. Find-a-seat turns occupancy into a live "where can I sit right now" board across study spaces, lounges, and open areas.

```
freeSeats = max( 0, capacity - estimatedPeople )
status:  Open   if utilization < 70%
        Filling if 70% ≤ utilization < 90%
        Full   if utilization ≥ 90%
```

It sorts the most-open spaces first. It is a small feature that quietly demonstrates the whole system: a real measurement, a capacity, a threshold, and a useful answer, with no camera in sight.

## 12 Energy & carbon opportunity CONFIGURABLE

If you know how often a space is empty, you know how much conditioning is being wasted on nobody. This view turns the Occupied Rate into an estimate of avoidable energy, cost, and carbon. The occupancy is ours and real; the energy factors are industry benchmarks you can override, so the result is labelled an inference until you connect a meter or building-management feed, at which point it becomes measured.

```

emptyHoursPerYear = conditioningHoursPerWeek × 52 × (1 - occupiedRate% ÷ 100)
avoidableKWh      = emptyHours × area(m2) × kWhPerM2PerHour
cost              = avoidableKWh × costPerKWh
co2               = avoidableKWh × co2PerKWh

```

The default benchmarks, all adjustable, are 60 conditioning hours per week, 0.04 kWh per square metre per hour, \$0.13 per kWh, and 0.40 kg of CO<sub>2</sub> per kWh. Change the tariff and the grid factor to your region and the estimate follows.

## 13 Coverage & RF PRODUCTION

Because occupancy depends on access points, the platform also models the radio side, on the same floor geometry. The coverage view predicts signal strength across a grid using the standard indoor log-distance path-loss model. This is a prediction, clearly labelled, not a survey; the client counts on it, however, are the real ones from the radios.

```

RSSI(dBm) = txPower - ( 40 + 30 × log10( distanceMetres ) )
// 40 dB is the loss at 1 metre; 30 = 10 × path-loss exponent (n = 3.0, typical
indoor).
// Each grid cell takes the strongest signal from any access point.

```

Cells are then banded the way any wireless engineer reads a heat map, and coverage is the share of the floor that is usable.

```

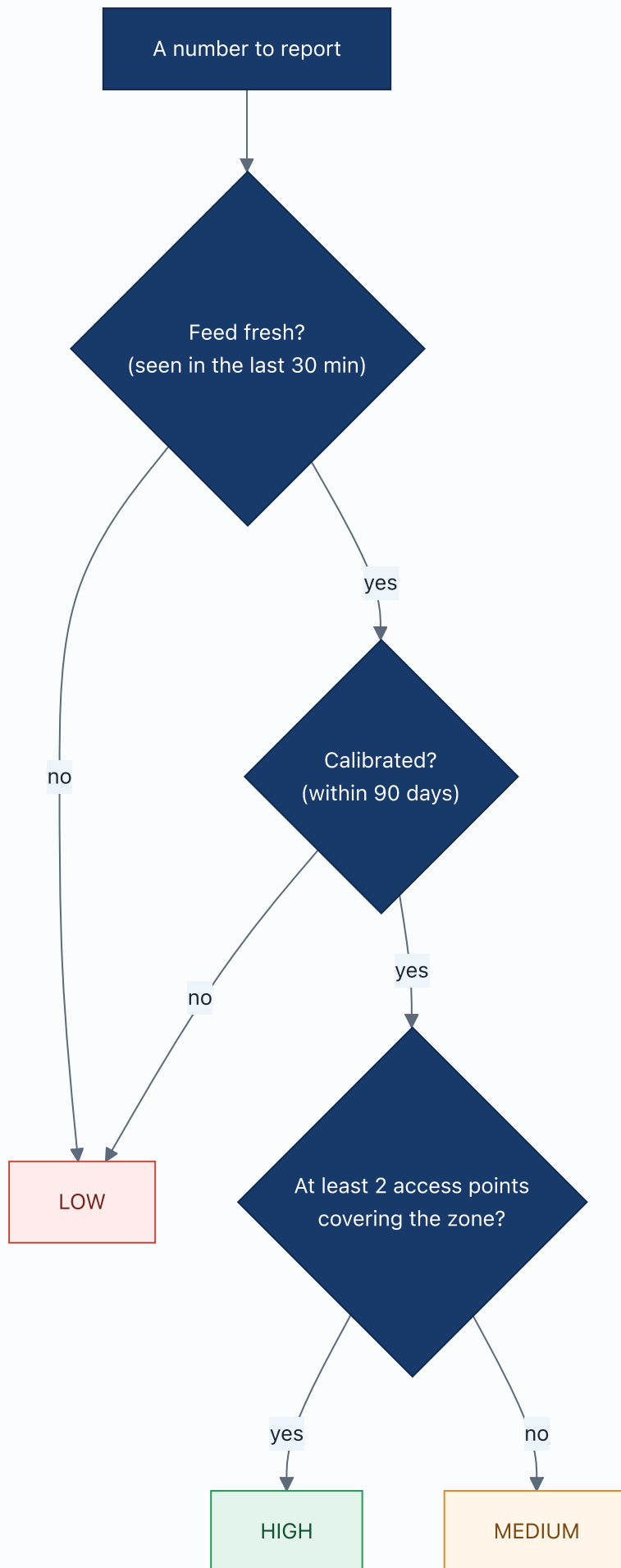
Strong ≥ -58   Good ≥ -67   Fair ≥ -75   Weak ≥ -82   Dead otherwise // dBm
coverage% = 100 × cells at or above -72 dBm ÷ total cells

```

On top of this sit the practical tools: a draggable AP and IDF/MDF placement, a co-channel overlap check (same 2.4 GHz channel and too close together), a per-site health roll-up (Good when coverage is at least 90% with no overlaps), and a Bill of Materials with a model-lifecycle check so you do not spec an access point that is going end-of-sale. This is the screen the planning engine sends you to when a room reads LOW for lack of coverage.

## 14 Confidence & honest numbers PRODUCTION

Here is the philosophy that separates this from a vanity dashboard: we never print a bare number. Every figure carries HIGH, MEDIUM, or LOW, and the rule is mechanical, not a vibe. Three things have to be true for HIGH.



**Figure 6. How a number earns its band.** A stale feed or a missing calibration is always LOW. Thin access-point density caps a good reading at MEDIUM. HIGH requires all three.

```
calibValid = calibrated AND calibration age ≤ 90 days
feedFresh  = last poll ≤ 30 minutes ago
adequateAps = at least 2 access points cover the zone

if NOT feedFresh OR NOT calibValid → LOW
else if adequateAps → HIGH
else → MEDIUM
```

Over a window, the confidence is the *worst* band of any sample inside it: one stale reading drags the whole window down, which is the conservative, defensible choice. The two things that cause LOW, a stale feed and a missing calibration, are surfaced as fixable operational issues with named severities, not buried. A weak number tells you the truth about itself and points at the remedy.

## 15 Privacy & data handling PRODUCTION

The strongest privacy guarantee is the one the database enforces, not the one in a policy PDF. There is no column to store a person in. The occupancy tables hold a zone, a time, and a count. That is all there is room for.



**Figure 7. Identity never lands.** Any client identifier in an incoming event is hashed with a per-run salt and dropped at ingest; only counts survive, and they age down a retention ladder.

- **Identity dropped at the edge.** If an event arrives with a MAC (a Mist enter/exit, say), it is hashed with a salt that rotates every restart, used only to pair an enter with its exit, and never written. The dwell record keeps zone, time, and duration.
- **Small numbers are withheld.** Any zone with one to three people is suppressed in the UI and exports and shown as LOW, so a near-empty room can never finger an individual. This applies to the heatmap cells too.
- **Bounded retention.** 15-minute detail rolls up to hourly after about two days, to daily after about sixty, and is pruned after roughly two years.
- **You own it, completely.** One export-all job emits everything, raw to derived to config, as CSV or JSON, and a verifiable purge deletes every sample, placement, calibration, polygon, and visit for a tenant and returns the counts it removed, with an audit record. Both require an issued key, because a tenant id is not a secret.
- **The method is the privacy.** Counting authenticated associations on your secured SSID, rather than sniffing probe requests, both defeats MAC randomization and avoids ever noticing the stranger walking past on the sidewalk.

## 16 Delivery & integration PRODUCTION

One engine, many ways to consume it. The dashboard gives a utilization-banded floor-plan heatmap, per-zone trends, a KPI rail, and role-oriented framing for executives, facilities, and planners. Beyond the screen there is a read-only REST API (current occupancy, trend series at 15-minute, hourly, and daily resolution, and average dwell), an OData v4 feed that drops natively into Power BI and Domo, a one-page executive PDF briefing, and the conversational assistant, which answers occupancy questions, cites the records it used, honours role-based access, and says "I don't know" rather than guessing. Occupancy also feeds the room and AV readiness picture the rest of the platform shows: empty, in use, headroom, and booking variance.

## 17 The switch-derived complement (Sense) EXPERIMENTAL

There is a second, independent way to tell whether a room is in use that needs no wireless feed at all. Sense reads the wired switch: how much Power-over-Ethernet a room is drawing, what services are announcing themselves over mDNS, and which multicast groups have joined over IGMP. From those three wired signals it infers whether a room is occupied or idle, whether AV is live, and whether something was physically unplugged. It is a complement, not a replacement, and it is off by default.

Because it is an inference rather than a count, Sense uses a different, additive confidence model. Each present signal adds points, and the remainder is shown as the model's honest uncertainty.

```
confidence = PoE draw (30) + IGMP AV group (28) + mDNS AV announcement (22) + recent
observation (14)
           capped at 100; the leftover is shown as "honest uncertainty"

presence: OCCUPIED if AV in use, or PoE draw ≥ 25 W
           IDLE     if powered but quiet
           UNKNOWN  if not seen in the last 30 minutes
```

Keep the two engines straight: the Wi-Fi engine *counts people* and bands them by a rule; Sense *infers room state* from wired signals and scores it by points. They answer different questions and never share a number.

## 18 802.11bf Wi-Fi sensing EXPERIMENTAL

Every method so far needs a *device*. We count phones and laptops and divide by a ratio, which works beautifully for a working population but has one real blind spot: the person who left their phone at the desk, the visitor with nothing connected, the quiet room of people simply listening. 802.11bf closes that gap, and CrossConnect reads it as one more source. Ratified in this generation of the IEEE spec as WLAN Sensing, it turns the radio itself into a presence sensor.

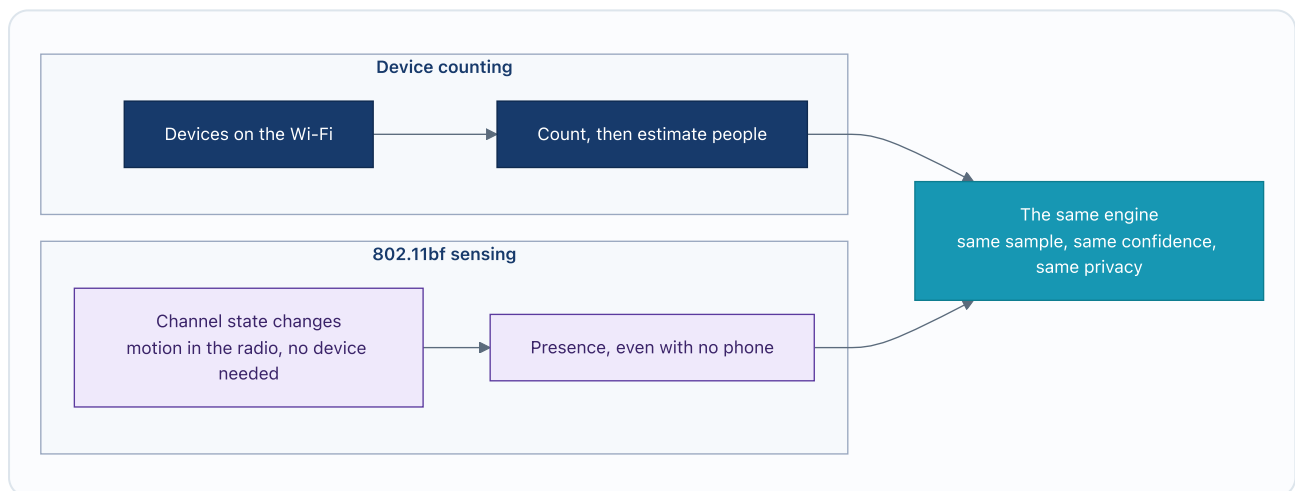
The idea is elegant. Your access points and clients already measure, hundreds of times a second, exactly how the radio waves bounce around a room, the channel state information, or CSI. When a person moves, breathes, or simply sits in the path, that pattern shifts. 802.11bf standardizes a way to read those shifts on purpose, so the Wi-Fi becomes a motion-and-presence sensor. No camera, no device on the person, no badge.

Inside the platform, sensing arrives exactly the way every other source does. A sensing collector reads per-zone CSI summaries from the access points, reduces them to a simple motion-and-presence signal, writes it into the same occupancy sample, earns the same confidence band, and lives under the same no-identity, small-number-suppressed privacy rules. We do not try to read gestures or count exact bodies from the waveform. We keep it to the honest question the physics answers well, "is this space active, and roughly how much," and then we fuse that with the device count.

```
motionIndex = how much the channel pattern moved versus the room's quiet baseline
// CSI variance
presence:  ACTIVE if motionIndex is above the room's learned quiet threshold
           STILL  if a steady occupied pattern holds without motion
           EMPTY  if the pattern matches the learned empty baseline

fused reading = device count where devices are present;
                sensing fills in when a zone reads empty of devices but active in
the RF,
                and raises confidence when the two signals agree
```

So the device count and the sensing signal reinforce each other: associations tell you how many, sensing tells you that someone is there at all, and together they are stronger than either alone, including the case of a busy room with almost nothing connected.



**Figure 8. Two ways to know a room is busy.** Device counting and 802.11bf sensing feed the same engine; sensing adds device-free presence and fills the gaps counting cannot see, without touching the privacy and confidence model.

There are real things to respect, and we treat them the same honest way as everything else. Sensing is exquisitely sensitive, which is exactly why the privacy posture matters more, not less: we keep it to coarse presence and motion at the zone level, never gesture or biometric inference, and the no-identity guarantee holds by construction because there is still no person to name. Support also matures vendor by vendor and radio generation by radio generation, so it ships Experimental: a labelled, advisory signal that strengthens a zone's reading and provides a device-free fallback, not a precise new truth. It is the same discipline you have read in every section above, now pointed at the newest capability in the standard.

## 19 What ships today vs roadmap

Capability	Status	Note
Occupancy engine, counts & people	PRODUCTION	AP-radio baseline plus Mist, Catalyst, and Cisco Spaces.
Calibration & confidence bands	PRODUCTION	Per-building ratio, R-squared, 90-day validity.
All metrics & the day-by-hour patterns	PRODUCTION	Fill, Occupied Rate, Intensity, dwell, pass-through.
Forecast (next-week peak)	PRODUCTION	Deliberately simple: typical shape, recent trend.
Scheduled vs actual, attendance	PRODUCTION	CSV import today; direct 25Live/Series25 on the roadmap.
Space planning, find-a-seat, coverage, BoM	PRODUCTION	The occupancy-to-coverage closed loop.
Energy & carbon opportunity	CONFIGURABLE	Inferred from benchmarks until a meter or BMS connects.
Sense (switch-derived presence)	EXPERIMENTAL	PoE, mDNS, IGMP. Off by default.
Automated CAD/DWG zone import; per-room PoE	ROADMAP	GeoJSON in place; CAD conversion stays upstream.
802.11bf Wi-Fi sensing	EXPERIMENTAL	Device-free presence and motion from CSI, fused with device counts. Coarse, advisory.

## A Formula reference

Name	Formula	Key constants
Estimated people	$\text{round}(\text{devices} \div \text{devicesPerPerson})$	default ratio 1.0 (uncalibrated → LOW)
Calibration ratio	$\Sigma \text{devices} \div \Sigma \text{headcount}$ , floored	floor 0.1; validity 90 days
Fit quality	$R^2 = \text{clamp}_{01}(1 - \text{ssRes}/\text{ssTot})$	needs $\geq 2$ observations
Utilization / Fill	$100 \times \text{people} \div \text{capacity}$	may exceed 100%
Occupied Rate	$100 \times \text{intervals}(\text{people} \geq 3) \div \text{intervals}$	threshold 3
Intensity	$\Sigma(\text{people} \times \text{minutes}) \div \text{area}(\text{m}^2)$	per square metre

Name	Formula	Key constants
Dwell	sum of merged visit lengths	merge gap 300 s; pass-through < 5 min
Confidence	fresh & calibrated & $\geq 2$ APs $\rightarrow$ HIGH	feed 30 min; calib 90 days
Small-N suppression	hide if $1 \leq \text{people} \leq 3$	floor 3
Forecast peak	$\text{slotAvg} \times (1 + \text{trend}\%)$	$\geq 14$ HIGH, $\geq 4$ MEDIUM
Schedule vs actual	empty 3, under 0.5x, over 1.5x	hourly buckets
Find a seat	$\text{capacity} - \text{people}$ ; Open/Filling/Full	70% / 90%
Energy	$\text{condHrs} \times 52 \times (1 - \text{occRate}) \times \text{area} \times \text{kWh/m}^2/\text{h}$	60 hrs/wk, 0.04 kWh, \$0.13, 0.40 kg
Coverage RSSI	$\text{txPower} - (40 + 30 \cdot \log_{10}(d))$	$n = 3.0$ ; usable $-72$ dBm
Sense confidence	PoE 30 + IGMP 28 + mDNS 22 + recency 14	occupied $\geq 25$ W; stale 30 min

## B Data model & retention

Table	Holds	Privacy / retention
occupancy_sample	Append-only series: zone, bucket time, interval (900/3600/86400 s), raw devices, estimated people, confidence, source, method note.	No identity column exists. 15-min $\rightarrow$ hourly (~2 days) $\rightarrow$ daily (~60 days) $\rightarrow$ pruned (~2 years).
zone_geometry	Zone polygon (GeoJSON/IMDF), floor, area (m <sup>2</sup> ), rated capacity.	Spatial only.
ap_placement	Access point (x, y) on a floor; resolved zone.	Point-in-polygon resolution.
occupancy_calibration	Per-building devices-per-person ratio, the raw (devices, headcount) pairs, R-squared, calibrated-at, valid-until.	One active row per building.
occupancy_visit	Anonymized dwell: zone, start, end, dwell seconds, source.	No device or client identity.
mist_source_config / catalyst_source_config	Site id, host/base URL, encrypted token, enabled flag.	Credentials encrypted at rest.
scheduled_occupancy	Imported bookings: location, title, expected people, start, end, source (CSV/25Live).	Advisory only.

Table	Holds	Privacy / retention
floor_marker	IDF/MDF wiring-closet markers on the floor, for cable-run math.	Same floor coordinates as AP placement.

---

CrossConnect by CybrIQ · Wireless & Occupancy Intelligence · Engineering reference · 24 June 2026 · Aggregate counts only, no personal data, confidence-banded, advisory.