

Data prerequisites — what has to exist before a capability is deployable

Draft v0.2, contributed for review. Licence CC BY 4.0.

Why this column

The capability checklist answers what a platform should be able to do. This answers what an operator asks next: will it work at my property, on my stack, this quarter. The answer usually turns not on the platform but on whether a specific field, endpoint or event exists in the systems already installed.

Each row states the prerequisite, who can answer it and how long that takes, and — the part usually missing — what the capability becomes when the prerequisite is absent. A checklist that only marks present/absent turns every gap into a blocker; most are not.

△ v0.1 called the middle column a "ten-minute test". A reviewing operator went through three of them and found that ten of the twelve begin with querying an API they have no credentials for: the real instruction is "send this row to your vendor and wait", and calling it ten minutes gets the column filled in badly by exactly the people it was written for. Renamed.

Provenance, and its limits

These are gaps encountered running booking and guest-service agents in production against six property-management systems and one booking-engine layer, over roughly two years.

△ The sampling frame is biased in two ways and both matter. The six systems are the ones our customers happened to run, not a sample of the market; and observed means observed in incident review, which only ever sees what broke. A prerequisite absent everywhere but quietly worked around never appears here. A prerequisite present almost everywhere but loud when absent looks universal. Read the column accordingly.

△ No row carries an absolute grade. v0.1 marked four rows "Blocker for booking". A reviewer from a large PMS pointed out that those four words are a procurement weapon, that they contradict the same row's own degradation text — a row cannot both block and degrade to a working manual flow — and that a grade set by one vendor about other vendors' systems does not belong in an alliance document. He offered to counter-sign six rows and report honestly where his product fails. That is the right mechanism, and until rows are counter-signed the impact column stays descriptive.

△ And v0.2 did not carry that out. It removed the words "Blocker for booking" and put "Blocks transacting" in the same four cells, "Blocks serving" in a fifth and "Blocks rentals" in a sixth — the same grade in a synonym, with the same contradiction against the same rows' own degradation text ("a person finishes every booking"). The column below now says what is lost, in the row's own terms, and nothing in it is a grade. No row is counter-signed yet, so no row carries one.

Written in capability terms rather than product terms on purpose: naming products goes out of date within two releases and says more about which systems we happen to have integrated than about the industry. Several vendors in this room could fill their own row, and that would be worth more than anything one contributor can write.

Rows

#	Prerequisite	Who can answer, and how long	Capability without it	Impact	Basis and its evidence
D1	An availability read that returns counts, not a yes/no — and it should say which count: physical, sellable, yieldable net of sell limit, net of allotment, or net of overbooking allowance	Vendor or integration engineer, one email. An operator cannot: it needs API credentials	Single-room quoting only. Multi-room, group and family requests are quoted wrongly or refused	Quoting degrades to single room	reasoned · Δ v0.2 said "observed · 2 of 7 ... both legacy deployments". Checked against our own incident record: the availability incidents there are four, none of them a boolean return, and the case that actually motivated this row is a booking engine rather than a PMS. We have no legacy or on-premise deployment among our clients at all, so "legacy" had nothing behind it
D2	The quoting layer sees the same number the property sells from	Vendor plus revenue manager, days	The agent states a price but cannot commit occupancy	Agent quotes what it cannot honour	observed · 3 of 7. Δ v0.1 said "compare with the front desk". There is no one front-desk number — the disagreement between physical, sellable and yieldable inventory is by design, and the old test produced a mismatch every time
D3	A deterministic write path the model does not authorise — a parameterised booking surface, or a scoped server-validated write API with a per-agent principal	Vendor, one email	A person finishes every booking	Every booking finished by a person	reasoned · Δ v0.1 offered only two options and called an API "ruled out by this blueprint", citing a section the same contributor drafted. A reviewer was right that this is circular: a scoped, policy-gated, server-validated write executor achieves the same property as a link. The requirement is deterministic authorisation, not a URL
D4	Card capture and charge initiation with outcome events	Vendor plus payments lead, days	The guarantee step fails last: the agent composes, a person finishes	Guarantee step finished by a person	observed · 4 of 7. Δ On-premise deployments that keep the PMS out of PCI scope by routing to a certified terminal are not deficient here; record that as an architecture, not a gap
D5	Reservation lookup by keys a guest actually gives	Vendor, one email. Δ Not the PMS search screen — the integration often searches different fields, and checking the screen produces a confident wrong answer	Every request needing the guest's own booking goes to a person	Every lookup goes to a person	observed · 5 of 7. Δ In tension with GD-6 in the companion checklist, which fails a lookup that can be enumerated. Both are right; the work is rate limiting plus uniform responses

D6	Agent-visible internal fields, at property and unit level	Vendor, one email	Operator procedures live outside the system, updated by hand. Content going stale becomes the normal state	High	observed · 4 of 7
D7	Content read API, multilingual	Vendor, one email	The agent keeps a copy alongside live availability. △ A governed copy with lineage is an architecture, not a degradation — the source system has no history, and history is what you need when a guest disputes a price. An ungoverned copy is the problem	High	observed · 6 of 7 for multilingual. △ Content is often brand- or licence-controlled; an agent restating it is republication, which is a different risk from staleness
D8	Single-unit addressability	Vendor, one email	Rental and mixed portfolios cannot be served	Rental portfolios not addressable	reasoned · △ v0.2 marked this observed and gave no count, against its own convention two paragraphs below. It also states our constraint as theirs: the one-agent-per-account limit is ours, documented on our side, and a PMS that exposes units is not obliged to solve it. This is principle 8 in operational clothing. If the group keeps both, say so, or the deliverable states one rule in two vocabularies
D9	Service and rate detail matching what the property sells — meal choice per day, additional discounts, more than one service	Vendor, days	The agent sells less than the property does, and the operator finds out when a guest asks	High	observed · 5 of 7. △ Raised from Medium: a reviewer called this the real gap, and he is right. And the companion payload profile could not express it either until v0.2 — the gap list was billing systems of record for a capability our own wire format lacked
D10	Events on reservation and verification state change	Vendor, one email	The agent polls. △ Polling loses history, not latency: intermediate transitions are unrecoverable, so no one can answer what a reservation looked like at 14:20	High	observed · 5 of 7 for reservation events. △ Verification state is an agent-layer concept the PMS has no model of; asking for it is scope transfer, not a gap
D11	An agent identity recorded on writes, in a structured field	Vendor, one email	No per-agent audit, and no basis for precedence when two agents act	High where more than one agent	reasoned · we have no direct evidence about how other systems attribute writes. We have not published a number on our own side either: the one we had counts a register built from one mailbox's tagged notifications against a product database that grows on its own, so it measures our tag coverage and not the difficulty of attribution. It answers a different question than this row asks

D12	A messaging identity the property controls, and the registration behind it	Programme manager plus telco, weeks	The platform is not the blocker; the calendar is	Schedule risk	reasoned · △ v0.2 marked this measured and cited "1–3 weeks in the US" from client-facing material. Checked: the phrase appears once, in a sales estimate written two days earlier about a registration not yet submitted, and our confirmed onboarding letter says "a few days". We have never measured a US carrier review. The one end-to-end duration we do hold is a UK number at 4 h 18 min, which points the other way, and a UAE toll-free at 7–9 weeks, which points further. The country count was also wrong: live clients sit in six countries, not three
-----	--	-------------------------------------	--	---------------	---

Rows for the agent vendor

△ A reviewer counted correctly: v0.1 was twelve rows of homework for systems of record and zero for the party selling agents. These four are the beginning of the other side. They are ours to be held to, and they belong in the same table.

#	Prerequisite on the agent vendor	Who can answer	Absent means	Basis
A1	Declared model and prompt version per deployment, with change notice to the operator	Vendor	The operator's agent changes behaviour without anyone telling them	reasoned
A2	A rollback path to a previous agent configuration, tested	Vendor	A bad change is fixed forwards, live, during service	reasoned
A3	Availability and incident reporting for the agent itself, on the operator's terms	Vendor	The guest-facing channel has no stated uptime, and the operator learns of an outage from guests	reasoned
A4	Stated liability for a wrong quote — who bears an incorrect rate the agent stated and the guest accepted	Vendor and operator, in the contract	The question is answered for the first time during the dispute	reasoned

The basis column

Following Gergana Zaharieva's convention, △ with a change the group should know about: hers has two values — portfolio evidence or reasoning — and this is a three-value extension. That is her call to accept or reject, not ours.

```

None
measured  a number, printed in the row. No number, no label.
observed  encountered in production incident review, with the
count:
```

n of the 7 systems, or n incidents over what period.
reasoned follows from the rest; not measured. Challenge these
first.

△ v0.1 marked ten of twelve rows observed with no counts, one row reasoned + observed — a fourth value against three definitions — and put measured on the only row containing no number. A reviewer's verdict was that a classification where 83 % of rows share one value decorates rather than discriminates. Counts added, the fourth value removed, and the one measured row now carries its numbers.

Which axis the two implementation profiles split on

WG3 is to produce two worked implementation profiles, and it is worth stating in the deliverable which axis they split on and why, whichever axis wins.

Our proposal is stack class rather than declared property type, and it follows from the rows above without needing a study: D1, D3 and D8 all say the installed stack determines what an agent can service, and the stack is knowable before an engagement begins. That is internal consistency, not an empirical finding, and it should be argued as such.

△ We had an observation attached to this and we are withdrawing it from the deliverable. Across seven operator portfolios, grouping by stack class separated them on the share of arrival-and-access requests and grouping by declared business type did not. Reviewers took it apart correctly: the two groupings agree on six of the seven, so the entire result is the seventh — $n=1$ written as $n=7$. At that size perfect separation reaches $p \approx 0.03$ – 0.14 depending on the split, and the two-sided minimum at a $4/3$ split is 0.057 , so significance is unreachable in principle. Two predictors were compared and the winner reported. And most likely the arrow points the other way: if the stack determines which requests are servable, the request mix partly measures the integration surface rather than guest behaviour.

It stays as a signed research note for anyone who wants it, and out of anything published under the alliance's name. Published with a caveat becomes cited without one.

△ One thing the proposal does not survive, raised by an operator: he runs three systems across two classes. Mixed portfolios are the first audience WG3 names, and a two-class split does not place him. Whatever axis the group picks, the profiles need a rule for portfolios that straddle it.

A note the committee may want to act on

Two reviewers independently pointed out that the "to finish" list still has no owner for a technical review against a real hotel stack, days from a release candidate. We offered ours and were told, correctly, that one vendor grading other vendors' systems under the alliance's name is the wrong shape. A joint exercise with a second member's stack and a report template agreed before either run would fix that, and one reviewer volunteered his. We support that version and withdraw the unilateral one.

