

FOUNDATIONS · CATEGORY ARCHITECTURE

The Amalgamation Fallacy

Why discovery, CMDBs, and telemetry can never produce intent — no matter how well you integrate them.

ABSTRACT

There is a roadmap most enterprises are following, and most analyst framing quietly assumes. Buy discovery. Add a CMDB. Layer in telemetry, observability, and correlation. Connect them properly, and somewhere in that assembly you converge on a model of the environment rich enough for automation — and now for AI — to act against. It is a persuasive story, because every part of it is true except the conclusion.

1. The descriptive stack

Consider what each layer of the modern stack actually records.

Discovery records what exists. A **CMDB** records what exists, with ownership attached. **Telemetry** records what is happening. **Observability** records what is happening, in context. **Correlation** records how the things happening relate to each other. **Configuration** records how things are currently set.

Six layers, one question. Every one of them answers *what is*. Integrating them produces a larger and more elegant *description* — never a statement of what is *allowed*. The stack can grow indefinitely along one axis and never advance an inch along the other.

Description accumulates. Authority must be declared.

2. Why description never becomes intent

This is an old distinction in new clothes. You cannot derive an *ought* from an *is*: no quantity of facts about how the world is settles the question of how it should be. Infrastructure is not exempt. A perfect record of your estate tells you everything except the one thing governing change requires — which of those states you meant, and which you would refuse.

Observation is not authority. Discovery is not intent. A fact is not a permission.

Three concrete cases, and no amount of integration resolves any of them:

- A CMDB can tell you a server exists and who owns it. It cannot tell you whether that server is *allowed* to accept traffic from the internet.
- Telemetry can tell you a configuration changed at two in the morning. It cannot tell you whether that change *should have been permitted*.
- Discovery can map every dependency in your estate. It cannot tell you which of those dependencies are *intentional* and which are accidents nobody has cleaned up.

In each case, what is missing is not more data. It is a decision somebody has to have made — recorded somewhere a change can be checked against it before it runs. Intent is a different kind of artifact from a fact, and no pipeline converts one into the other.

3. Why AI makes the fallacy impossible to ignore

For two decades the gap was survivable, because the consumers of the descriptive stack were people. A human operator reading a dashboard supplies the missing half silently: they know, or believe they know, what the environment is supposed to look like. The description on the screen is interpreted through intent carried in someone's head.

Automation weakened that arrangement. AI ends it.

An AI system acting on infrastructure consumes the descriptive stack directly, at machine speed, with no head to carry the missing half. Give it a perfect description and it will act on the description *as if it were permission* — because nothing in the data distinguishes the state you intended from the state that merely exists. The more complete the description, the more confident the action, and confidence is precisely the wrong thing to scale when authority was never in the data.

This is why the amalgamation roadmap fails at the exact moment it is supposed to pay off. The integration project was justified as preparation for autonomous operations. But the thing autonomous operations require — a declared, evaluable statement of what is allowed — was never in any of the systems being integrated, and connecting them cannot create it.

4. What crossing the gap actually requires

The gap between *what is* and *what is allowed* is crossed the only way it can be: by declaration. Somewhere in the operating environment there must be a layer that records intent as a first-class artifact — permitted states, ownership, boundaries, exceptions — and evaluates connected change against it *before execution*, for every actor: people, automation, and AI.

That layer is not a better CMDB, because a record of what exists, however accurate, is still a record of what exists. It is not richer observability, because seeing more is not deciding anything. It is a normative layer above the descriptive stack — the layer the Infrastructure Operating Model standard exists to define.

The descriptive stack is not wasted investment. It is half of the answer: the facts the normative layer reconciles against. The fallacy was never in building it. The fallacy is believing that if you build enough of it, the other half appears.

It never does. It has to be declared.

The Infrastructure Operating Model standard, including the definition of the authority layer and its relationship to the descriptive stack, is maintained at theiom.org/standard/. The whitepaper "Representation Is Not Authority" develops the argument of this article in depth: theiom.org/whitepapers/.

The IOM Standard is an open, vendor-neutral specification for modeling infrastructure intent, ownership, constraints, and meaning. Learn more at theIOM.org.