



DOME SYSTEMS

The Scarcity Isn't Code Anymore

When code is cheap, operating it safely is the scarce resource — where enterprise value concentrates next

What enterprises actually buy when they buy SaaS	3
The cost collapse	4
Disruption from two directions	5
The management plane becomes your problem	6
Where value concentrates	7
A new budget category	8

What enterprises actually buy when they buy SaaS

For most of the SaaS era, the transaction has been straightforward: enterprises pay a vendor to build, host, and operate software on their behalf. The per-seat license covers not just the application, but the entire management plane beneath it: identity, access control, audit logging, compliance, uptime, security patching, data residency.

Enterprises understood this, even if they rarely articulated it in these terms. When a CISO approved a SaaS vendor, they were not approving the features alone. They were approving the operational envelope: the SOC 2 report, the incident response process, the data handling practices, the authentication model. The management plane came bundled with the product. The price of the license covered both.

This model works when building software is expensive. If it costs millions to build a CRM, it makes sense to buy one and let the vendor absorb the operational overhead. The economics were clear for decades.

The economics are changing.

The cost collapse

The marginal cost of producing working software is approaching zero.

Large language models have compressed the cycle from specification to functioning code from months to hours. A single engineer with an AI coding assistant can produce in a day what previously required a team and a quarter. A product manager can prototype a working system over a weekend. An operations team can build an internal tool between meetings.

This is not a marginal improvement in developer productivity. It is a structural change in who can build software, how fast, and at what cost. The capability is real, it is accelerating, and it is available to every enterprise with a browser and a subscription.

This changes the calculus of make versus buy. When the application layer was expensive to produce, buying made sense. When it is cheap to produce, the question shifts: what are you actually paying for when you pay a SaaS vendor \$150 per seat per month?

Some enterprises are already asking. Not about all SaaS as some complex systems of record are not weekend projects but about the long tail: the workflows, the integrations, the report generators, the internal tools. The applications that an agent can now build and operate for a fraction of the previous cost.

Disruption from two directions

The pressure on the existing model is not coming from one direction. It is coming from two, simultaneously.

From below, the cost of building replaces the need to buy. Internal teams can now produce functional alternatives to SaaS tools that previously required a vendor, a contract, and a six-month implementation. The long tail of enterprise SaaS — the tools that are useful but not critical, the applications that exist because building was too expensive — is being rebuilt in-house, by teams that never previously wrote software.

From above, hundreds of agentic SaaS tools are spreading through enterprises with minimal oversight. Every AI-native application that connects to enterprise data from writing assistants, research tools, code generators, to workflow automators, brings a new integration point, a new set of permissions, a new data flow that someone needs to govern. The vendor ecosystem is not shrinking. It is fragmenting into smaller, more numerous, less manageable pieces.

Both directions converge on the same problem. The management plane that SaaS vendors bundled into their products: identity, policy, audit, compliance, is becoming the enterprise's direct responsibility. In the first case, because the enterprise is building what it used to buy. In the second, because no single vendor provides the operational envelope across a fragmented landscape of agentic tools.

The management plane is disaggregating from the application layer. And most enterprises have no platform to provide it themselves.

The management plane becomes your problem

Here is what the make-versus-buy reframing misses: when you build rather than buy, the management plane is no longer someone else's problem. It is yours.

A SaaS vendor bundles identity management into the product. When you replace that product with an agent, you need identity management for the agent. A SaaS vendor provides audit logging. When you replace the product, you need audit logging for the agent's actions. Compliance evidence. Access control. Data residency. Incident response. All of it, for each system, from day one.

The application layer got cheaper. The management plane did not.

Every enterprise replacing a SaaS tool with an agent-built alternative is making an implicit decision to take on the management plane for that system. Most have not yet reckoned with the cost. Not because the cost is hidden, but because the management plane was so thoroughly bundled into SaaS that enterprises forgot it was a separate concern.

As the number of agent-built and agent-operated systems grows — and it will grow, because the economics are irresistible — the management plane cost does not grow linearly. It grows combinatorially. Each new agent needs identity, policy, audit, compliance. Each new agent-to-tool connection needs governance. Each new data flow needs oversight.

An enterprise running fifty agents across a dozen tools does not have a governance problem fifty times larger than one running a single agent. It has a governance problem that is qualitatively different: dozens of identity schemes, dozens of access control models, dozens of audit formats, none of them consistent with each other. No single vendor is providing the operational envelope. The enterprise must provide it or accept the risk of operating without one.

Where value concentrates

The structural inversion is now visible: as the cost of producing working software falls, value migrates to the constraint that does not fall.

That constraint is operating software safely: identity, policy, audit, compliance. The proof that software behaves correctly, at scale, to regulators, boards, and customers.

This is not a temporary gap waiting for "AI governance" features to be patched onto existing tools. It is a permanent feature of the agentic economy. Code will continue to get cheaper. The management plane will not, because the management plane is where trust is produced, and trust does not scale through automation alone. It scales through a platform: consistent identity across every agent, consistent policy enforcement across every tool call, a single audit trail across every action.

Every prior era of enterprise computing produced this pattern. When networks grew complex, the bottleneck was not connectivity then the answer was authentication and directory services. When applications became distributed, the bottleneck was not building APIs, it was governing API access. When infrastructure moved to the cloud, the bottleneck was not provisioning servers, it was managing secrets and orchestrating containers. In each case, the application layer commoditized. Value concentrated at the operations layer.

The agent era follows the same arc in that the scarcity is not code, it is control.

A new budget category

Enterprise technology spend is reorganizing around this shift. The traditional model of SaaS purchasing is being arranged around efficiency and effectiveness of continuing spend, and supplemented by a new category: agentic operations spend.

The enterprises that recognize this early will build an agent operations platform — DevOps for agents — that applies consistently across every agent, every tool, every runtime. One identity model. One Rules engine. One audit trail. Deployed once, extended across the estate.

The enterprises that do not will accumulate governance debt: per-agent, per-tool, per-team solutions that become more expensive and more fragile as the number of agents grows. Each new deployment requires its own identity scheme, its own access controls, its own audit mechanism. The cost compounds. The inconsistency compounds faster.

When code is cheap, operating it safely is the scarce resource. The enterprise that treats agent operations as a durable platform captures the value that the commoditizing application layer leaves behind.