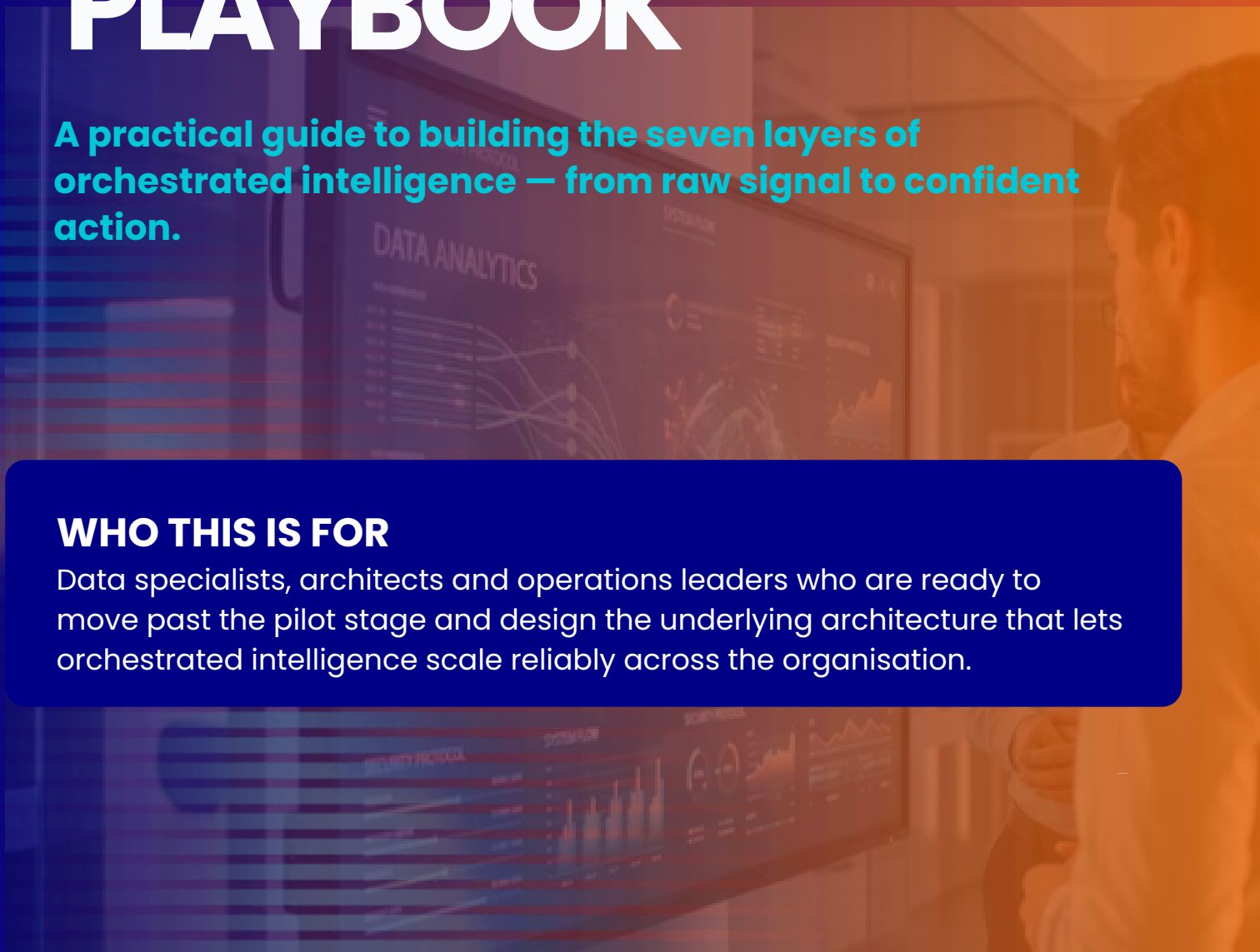
emite
A PROPHECY SOLUTION

THE ORCHESTRATION ARCHITECTURE PLAYBOOK

A practical guide to building the seven layers of orchestrated intelligence — from raw signal to confident action.



WHO THIS IS FOR

Data specialists, architects and operations leaders who are ready to move past the pilot stage and design the underlying architecture that lets orchestrated intelligence scale reliably across the organisation.

CHAPTER 1 THE CASE FOR ARCHITECTURE

Most organisations don't fail at orchestration because the idea is wrong. They fail because the architecture underneath it was never built to carry the weight of what they're now asking it to do.

The pattern shows up consistently in the research. Gartner predicts that over 40% of agentic AI projects will be cancelled by the end of 2027, driven by escalating costs, unclear business value and inadequate risk controls, not by model capability.

McKinsey finds that nearly two-thirds of enterprises have experimented with AI agents, but fewer than 10% have scaled them to deliver tangible value, with eight in ten pointing to data limitations as the roadblock.

And Gartner's data and analytics predictions go further still, forecasting that by 2030, 50% of AI agent deployment failures will trace back to insufficient AI governance platform runtime enforcement, architecture gaps, not intelligence gaps.

40%+	<10%	50%
Of agentic AI projects predicted to be cancelled by end of 2027 (Gartner)	Of enterprises have scaled agentic AI to deliver value (McKinsey)	Of AI agent failures by 2030 traced to governance gaps, not models (Gartner)

> ***The business case for orchestration is relatively easy to make. The architecture underneath it is where most of the real work, and most of the real risk, actually lives.***

This playbook is a practical answer to that gap. It takes the seven-layer reference architecture behind orchestrated intelligence and turns it into something you can actually build against, one layer at a time.

THE SEVEN-LAYER MODEL, AT A GLANCE

Each layer has a distinct job. Getting orchestration right doesn't mean building all seven simultaneously, it means understanding how they depend on each other, and sequencing the build accordingly.

1	**Experience Data**	Voice, digital, queues, agents, WFM, quality, interaction metadata.
2	**Enterprise Data**	CRM, ERP, billing, finance, orders, service, HR, operational systems.
3	**Integration & Processing**	Ingestion, transformation, normalisation, correlation, freshness, quality.
4	**Semantic Context**	The relationships that turn connected data into understood data.
5	**Governance & Persona**	What context is permitted, for whom, and under what controls.
6	**Intelligence**	Dashboards, KPI engines, forecasting, anomaly detection, conversational AI.
7	**Decision & Action**	Where the architecture produces business value, the outcome layer.

The chapters that follow group these seven layers into three practical build phases:

- ***the data foundation (Layers 1–3),***
- ***the context and governance layer (Layers 4–6),***
- ***and the outcome layer (Layer 7),***
- ***followed by a five-phase roadmap for sequencing the whole build.***

CHAPTER TAKEAWAYS

- Orchestration failures are overwhelmingly architecture failures, not model failures.
- The seven-layer model separates concerns so each layer can be built, tested and matured independently.
- This playbook sequences the build into three practical groupings, then a five-phase roadmap.

CHAPTER 2
THE DATA FOUNDATION, LAYERS 1–3

Everything above this foundation depends on it. If the data feeding your architecture is inconsistent, poorly integrated, or stuck in silos, no amount of sophistication in the layers above will fully compensate.

LAYER 1, EXPERIENCE DATA

This is the interaction and workforce layer, for a Genesys environment, voice, digital interactions, queues, agents, WFM, quality, journey information, interaction metadata and conversation outcomes.

It's usually the richest, best-instrumented part of the stack, because it's been the primary focus of contact-centre analytics for years. That's also its limitation: on its own, it describes what happened inside the interaction, not why, and not what it means for the wider business. Layer 1 is necessary but never sufficient on its own.

LAYER 2, ENTERPRISE DATA

The second layer extends context into other business systems: CRM, ERP, billing, finance, orders, service management, HR, web and digital platforms, product platforms, operational systems, and industry-specific applications.

Data at this layer is heterogeneous by nature, transactional, event-based, historical, real-time, structured, semi-structured. A common early mistake is trying to centralise all of it before anything becomes useful.

BUILD GUIDANCE

Don't start with a comprehensive enterprise data warehouse. Start with the two or three enterprise sources your highest-value use case actually needs, connected properly.

You'll learn far more about what “consistently available” needs to mean from one well-built connection than from a big-bang integration project.

LAYER 3, INTEGRATION AND DATA PROCESSING

This is where source-system data becomes usable across applications, rather than trapped inside the system that generated it. It's also the layer where McKinsey's research says most scaling problems actually surface: **eight in ten companies scaling agentic AI cite data limitations as the roadblock, and a shaky integration layer is very often where that shows up first**.

The integration layer needs to manage six distinct functions. Each deserves its own attention, treating this as a single “data pipeline” step tends to be where projects get into trouble.

Ingestion

Receiving information from source systems, via API, event stream, database, or file. Design consideration: what happens when a source system changes its API or schema without warning? Integrations built to fail loudly, rather than silently, save far more time downstream.

Transformation

Converting data into the required format for downstream use. Design consideration: transformation logic tends to accumulate special cases over time. Document the business reason for each transformation rule, not just the rule itself, or it becomes unmaintainable within a year.

Normalisation

Creating consistency across sources that describe the same thing differently, one system's “closed” case status might be another's “resolved.” Design consideration: normalisation decisions are effectively business decisions about what a shared term means. They belong with the people who own the metric definitions, not solely with the integration team.

Correlation

Identifying relationships between entities and events across systems, this is the bridge into Layer 4's semantic model. Design consideration: correlation needs a reliable shared key (a customer ID, an account number) across systems. Where that key doesn't exist cleanly, resolving it is foundational work, not a nice-to-have.

Freshness

Ensuring information is current enough for the use case it's serving. Design consideration: not every use case needs real-time data, and treating everything as real-time is expensive and often unnecessary. Match freshness requirements to the decision the data supports.

Quality

Identifying incomplete, invalid or inconsistent information before it propagates downstream. Design consideration: quality checks that only run at ingestion miss problems introduced by transformation or correlation. Quality needs checkpoints throughout the pipeline, not just at the start.

Within the emite architecture, this is where the data-integration capabilities in emite and ProDataIQ sit, creating a reusable foundation rather than rebuilding the path to each source for every new dashboard or AI use case.

CHAPTER TAKEAWAYS

- Layers 1 and 2 are about breadth of data; Layer 3 is about making that data trustworthy and usable.
- Most scaling failures trace back to Layer 3, treat its six functions as distinct disciplines, not one generic “pipeline” step.
- A reusable integration layer means the first use case contributes to the next, instead of every team starting from zero.

CHAPTER 3
CONTEXT, GOVERNANCE & INTELLIGENCE, LAYERS 4–6

With a solid data foundation in place, the next three layers turn connected data into something an organisation, and increasingly, an AI system, can actually reason with.

LAYER 4, SEMANTIC CONTEXT

Connected data does not automatically mean understood data. This is the layer that establishes meaning, and it's the layer most architectures underinvest in, because it's less visible than a dashboard and less immediately gratifying than a model.

Concretely, this layer answers questions like: does “customer” mean an individual, an account, a household, or an organisation? Does “abandonment” use the same calculation across every business unit? Which employee handled a particular customer interaction? Which operational event corresponds with a spike in contact volume?

Answering those requires more than schema mapping. It requires modelling relationships between the entities the data represents:

Customer ↔ Interaction · Interaction ↔ Agent ·Agent ↔ Workforce Plan

Customer ↔ Account · Account ↔ Order ·Order ↔ Operational Event · Event ↔ Contact Demand

Once those relationships exist, AI can retrieve context rather than disconnected records, a distinction that becomes critical, not cosmetic, once agentic systems are expected to act rather than just report.

BUILD GUIDANCE

Start your semantic model with the two or three relationships your highest-value decisions depend on most (often Customer ↔ Interaction and Interaction ↔ Agent in a contact-centre context). Expand outward from there rather than trying to model the whole enterprise at once.

LAYER 5, GOVERNANCE AND PERSONA

Before information reaches analytics or AI, the system needs to know what context is permitted. In practice, this layer operates across six distinct controls:

- Data access, which sources and fields is the user authorised to access?
- Persona, what role is the person performing?
- Context, which information is relevant to the question being asked?
- AI response, what can the AI include in its answer?
- Action, what can be recommended, initiated automatically, or requires approval?
- Oversight, how can the organisation understand and monitor the intelligence being produced?

The persona becomes part of the context request itself, not a filter applied afterward. A workforce manager and an executive asking a related question about service performance may legitimately receive different levels of detail, even though the underlying data is shared

> **AI should have the context it needs, not simply access to every context available.**

This is the layer Gartner's 50%-by-2030 governance-failure forecast points directly at. It's also the layer most often bolted on late, as a compliance step, rather than designed in from the start, which is precisely why it fails so often at scale.

LAYER 6, INTELLIGENCE

Once data has been connected, contextualised and governed, multiple intelligence capabilities can operate over it: dashboards, KPI engines, trend analysis, anomaly detection, forecasting, machine learning, natural-language querying, generative AI, and conversational analytics.

This is also where AskEmite provides a conversational interface into the governed business context. Instead of navigating through multiple reports, a user can begin with the business question itself:

"***Why did abandonment increase this afternoon?***"

The intelligence process resolves that question against the relevant data, persona and time period, identifying, for example, that interaction demand increased, two queues accounted for most of the variance, staffing was below forecast, average handling time increased, and the demand spike correlated with a specific operational event. The answer is constructed from connected context across Layers 1 through 5, not simply generated from a language model working in isolation. That distinction is what keeps the answer auditable.

CHAPTER TAKEAWAYS

- Layer 4's semantic model is what allows Layer 6's intelligence to retrieve context instead of disconnected records.
- Layer 5's governance needs to be designed in from the start, it's consistently the layer that causes failures when bolted on late.
- Layer 6 is where the architecture becomes visible to end users, but its quality depends entirely on the four layers beneath it.

CHAPTER 4 FROM DECISION TO ACTION, LAYER 7

This is where the architecture earns its investment. Layer 7 is where connected, governed, contextualised intelligence turns into something that actually changes what the organisation does next, through a dashboard, an alert, an AskEmite response, a recommendation, a workflow, or an operational intervention.

Walkthrough: an unexpected spike in repeat contacts

To make Layer 7 concrete, it helps to walk a second scenario end-to-end, distinct from the abandonment example used elsewhere in this series, through all seven layers.

A CX leader notices repeat contact rates have crept up over the past week. On its own, that's a Layer 1 signal: more people are calling back about something. Here's how the architecture turns that signal into action:

- **Detect (Layer 1 → 6)**, Repeat contact rate moves outside its expected range for three consecutive days, flagged by an anomaly-detection model running over experience data.
- **Correlate (Layer 3),** The integration layer pulls in related signals: which queues and topics are driving the repeat contacts, whether case-resolution times have changed, and whether there's a pattern in the interaction metadata (channel, time of day, agent).
- **Apply context (Layer 4),** The semantic layer connects repeat interactions back to the original case, the account, and any related order, surfacing that a disproportionate share of repeat contacts trace back to accounts with a specific billing plan changed three weeks earlier.
- **Apply persona + governance (Layer 5),** The CX leader receives a business-impact view (which journeys are affected, estimated customer impact); a data analyst investigating the same anomaly can drill into the underlying case-level detail their role is authorised to see.
- **Analyse (Layer 6),** The intelligence layer identifies the billing plan change as the strongest correlated factor, ahead of channel mix or agent variance.
- **Explain (Layer 6),** AskEmite surfaces the finding in plain language, with the supporting data available on request: “Repeat contacts are up 18% this week, concentrated in accounts moved to the new billing plan on [date]. This accounts for approximately two-thirds of the increase.”
- **Act (Layer 7),** The CX leader routes the finding to the billing team with the affected account segment attached, and flags the pattern to the contact-centre team so agents handling these calls have the context already surfaced rather than starting from zero.

Notice what didn't happen: nobody had to manually pull a case report, a billing report and an interaction report and line them up by hand. The architecture did that correlation work, and a person made the call about what to do with the answer.

The pathway, end to end

Source → Context → Intelligence → Persona → Decision → Action

It's worth being explicit about what this architecture doesn't require:

it doesn't require one system to perform every role. Genesys continues to provide specialised CX, interaction and workforce capabilities. Enterprise applications continue to run their own business processes. emite and ProDataIQ create the connected intelligence layer across those sources. AskEmite provides the natural-language pathway into that intelligence.

The architecture's job is to bring those capabilities together around the decision, not to replace any of them.

> **The value doesn't come from connecting systems. It comes from connecting the right context to the moment a decision needs to be made.**

CHAPTER TAKEAWAYS

- Layer 7 is where the previous six layers convert into a business outcome, it's the layer stakeholders actually see the value of.
- A good Layer 7 outcome routes the right information to the right owner, not just a notification that something changed.
- No single platform needs to own every layer, the architecture's job is coordination, not consolidation.

THE ROADMAP:
FIVE PHASES TO BUILD THIS ARCHITECTURE

Building all seven layers at once isn't realistic, and it isn't necessary. The sequence below sets out a practical order, each phase builds on the one before it, and each has a clear signal for when you're ready to move on.

PHASE 1 Establish the Data Foundation

Build a stable base across Layers 1 and 2 for one high-value use case.

- Pick one recurring business question your team already tries to answer manually.
- Identify the minimum experience and enterprise data sources needed to answer it.
- Agree and document consistent definitions for the core metrics involved.
- Assign clear ownership for each data source.

You're ready for the next phase when: two teams pulling the same metric get the same number, and can explain why.

PHASE 2 Create the Integration Backbone

Turn one-off connections into a reusable Layer 3 foundation.

- Build the ingestion, transformation and normalisation logic once, designed for reuse rather than a single use case.
- Add quality checkpoints at ingestion and after correlation, not just at the start.
- Document freshness requirements per use case rather than defaulting everything to real-time.
- Confirm the pipeline survives a source-system change without manual rework.

You're ready for the next phase when: a second use case can plug into the same integration layer without rebuilding it.

PHASE 3 Build Semantic Relationships

Model the Layer 4 relationships your highest-value decisions depend on.

- Map the two or three entity relationships most relevant to your first use case.
- Establish a reliable shared key across the systems involved.
- Validate that a query can retrieve context (not just records) using the model.
- Expand the model outward to adjacent relationships as new use cases require them.

You're ready for the next phase when: AI or analytics can answer a "why" question, not just a "what" question.

PHASE 4 Embed Governance and Persona

Design Layer 5's access, persona and oversight controls in from the start.

- Define role-based access for the data involved, rather than relying on informal norms.
- Map your key personas and what level of detail each is authorised to see.
- Decide explicitly which actions require human approval versus automatic execution.
- Set up logging so AI-generated recommendations and actions are auditable.

You're ready for the next phase when: access controls would still hold if usage tripled overnight.

PHASE 5 Activate Intelligence and Close the Loop

Bring Layers 6 and 7 online, and make sure insight turns into action.

- Connect the governed, contextualised data to dashboards, anomaly detection, or conversational AI such as AskEmite.
- Assign a clear owner responsible for acting on the intelligence produced.
- Define how you'll measure the business outcome the capability is meant to influence, not just usage.
- Feed learnings back into Phases 1–3 as new use cases surface new data and context needs.

You're ready for the next phase when: you can point to a specific decision that changed as a result, not just a dashboard that gets viewed.

WHERE TO START

You don't need all seven layers built to get value, you need the right layers built for your highest-priority use case, in the right order. Most organisations get the most immediate return from getting Phases 1 and 2 right, because nearly everything above depends on them.

If you've already run the AI Pilot Scaling Checklist, your lowest-scoring foundation is a good indicator of which phase to prioritise first.

BUILD THIS ARCHITECTURE WITH US

Let's talk about what this seven-layer architecture looks like in your specific environment, which layers you already have, and where the highest-value gaps are.

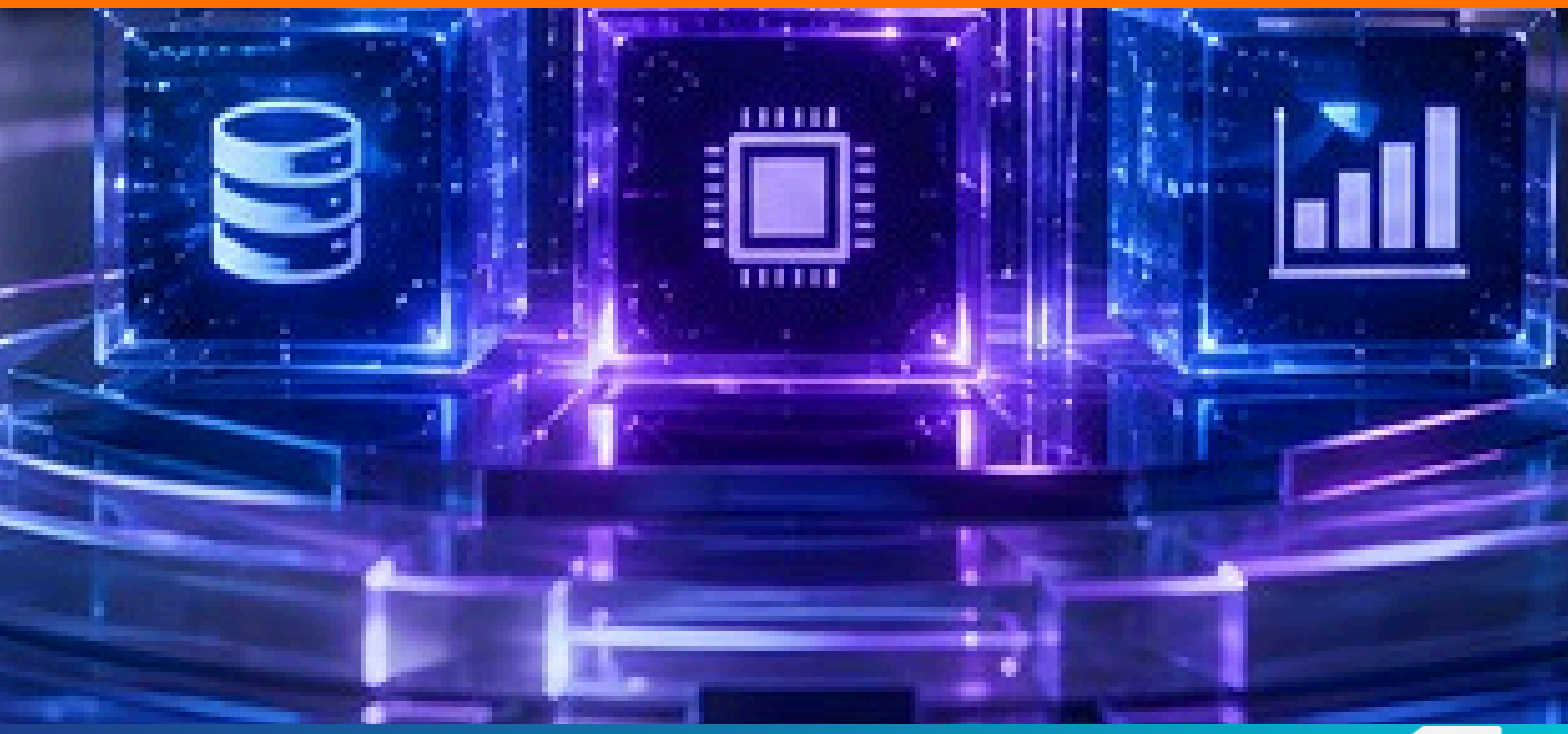
Request a demo → emite.com/request-a-demo ·

Read Issue 11 → emite.com/data-advantage-issue-11 ·

Get the checklist → emite.com/ai-pilot-scaling-checklist

READY TO SEE THIS ARCHITECTURE IN PRACTICE?

Book a call today www.emite.com



emite
A PROPHECY SOLUTION

www.emite.com