The 2-Step Agent Architecture That Keeps Token Costs Down

A structured approach to agentic AI: isolate, generate, and apply — without burning your budget.

THE CHALLENGE

Traditional LLM agents pass entire data payloads as context — exploding token usage and cost at scale. The answer is architecture, not more c

STEP ONE

Build the Standalone Database

01

- **Ingest & structure**
  Raw data lands in an isolated, purpose-built database — separate from your production systems.
- **Schema optimized for queries**
  Tables and indexes designed around the questions your agent will ask, not general OLTP use.
- **No raw data in the prompt**

Result: A clean, queryable layer the agent can interrogate precisely.

STEP TWO

Prompt □ Code □ Execute

02

- **Natural language in**
  The agent receives a user intent or task in plain language — not a table of records.
- **Translate to query code**
  The LLM's job is narrow: convert the prompt into SQL, Python, or API calls against the database.
- **Execute & return results**

Result: Token usage is bounded by output size, not data volume.

WHY THIS ARCHITECTURE WINS

60–80%	Faster agent response loops	Data stays in your infrastructure	Scales with data volume — cost doesn't
Reduction in token spend			

BUILT FOR:

- Non-profit platforms with high query volume
- Marketing dashboards & automated reporting
- Regulated industries (Pharma, Finance, DoD)
- Service businesses scaling to 1,000+ users

Ready to build agents that scale?

Let's scope your architecture in a 30-minute working session.

www.2516technologies.com