Autonomous Postgres

From self-managed operations to an AI-ready data platform

PGConf.EU 2026 · Community Events Day: Autonomous Postgres · Valencia · Friday 23 October, 09:15

Rajesh Nagpal · Anil Kumar · EDB

©EDB 2026 — ALL RIGHTS RESERVED.

YOUR SPEAKERS

PART 1 · FOUNDATION FOR AI



RAJESH NAGPAL

VP of Product - AI, EDB

Experiments with AI

Figuring out bringing AI into the database. And scale AI to multiple databases and distributed data systems without data egress

25+ Years in Data and AI

Led Engineering and Product at Microsoft, Oracle, Meta, Compass and a GPU Cloud

PART 2 · WHAT WE BUILD ON IT



ANIL KUMAR

Head of Product Management, Database Portfolio · EDB

Owns the Postgres engine roadmap at EDB

EPAS and PGE, Postgres Distributed, HA and operations tooling, and in-database AI.

Two decades building systems

Mission-critical platforms across Postgres, NoSQL and cloud-native infrastructure: EDB, Axon, Couchbase, Microsoft, Intel.

©EDB 2026 — ALL RIGHTS RESERVED.

AI RESTS ON THREE PILLARS: MODEL, AGENT AND CONTEXT

1 MODEL

Intelligence

Reasons over language, code and data, and generates answers.

2 AGENT

Action

Asks the model what to do, calls tools to do it, and loops until the task is done.

3 CONTEXT

Knowledge

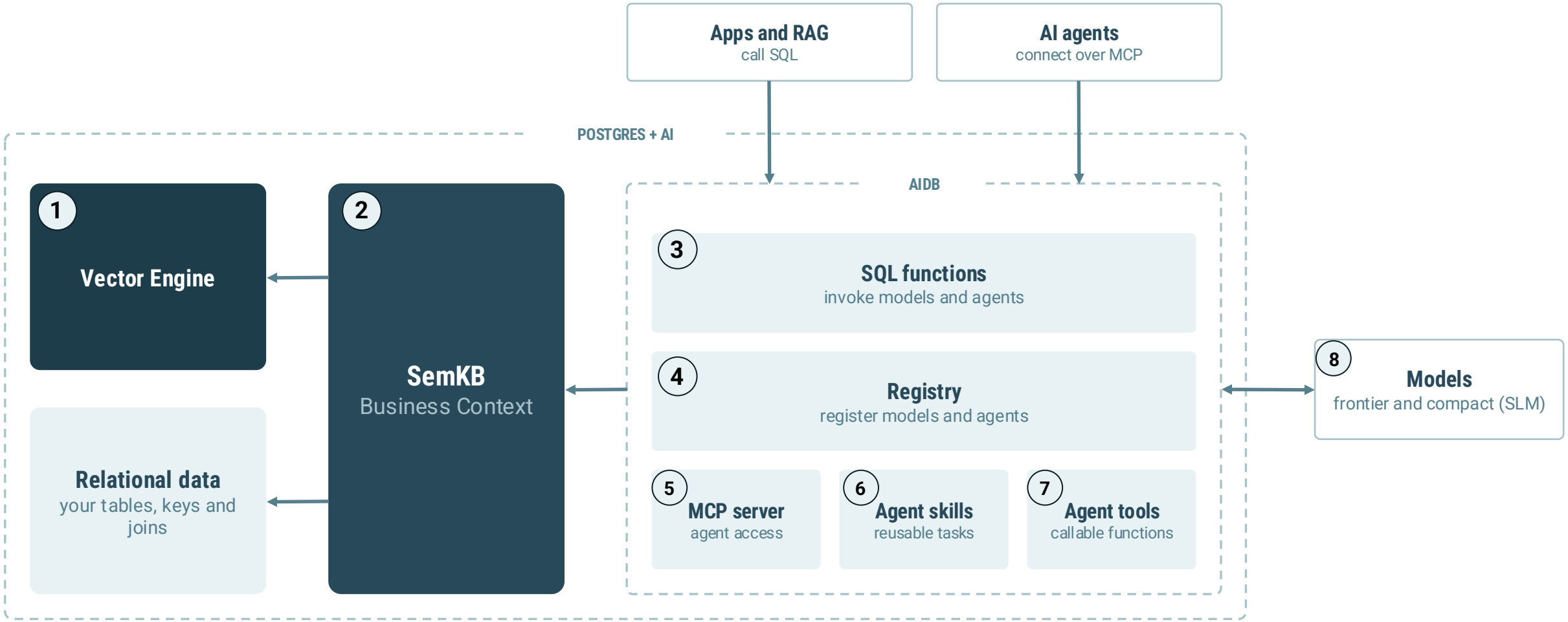
Adds the relevant data plus its meaning to the prompt, for a more accurate response.

MISSION : Bring AI to Postgres to benefit every user.

EDB POSTGRES AI

©EDB 2026 — ALL RIGHTS RESERVED.

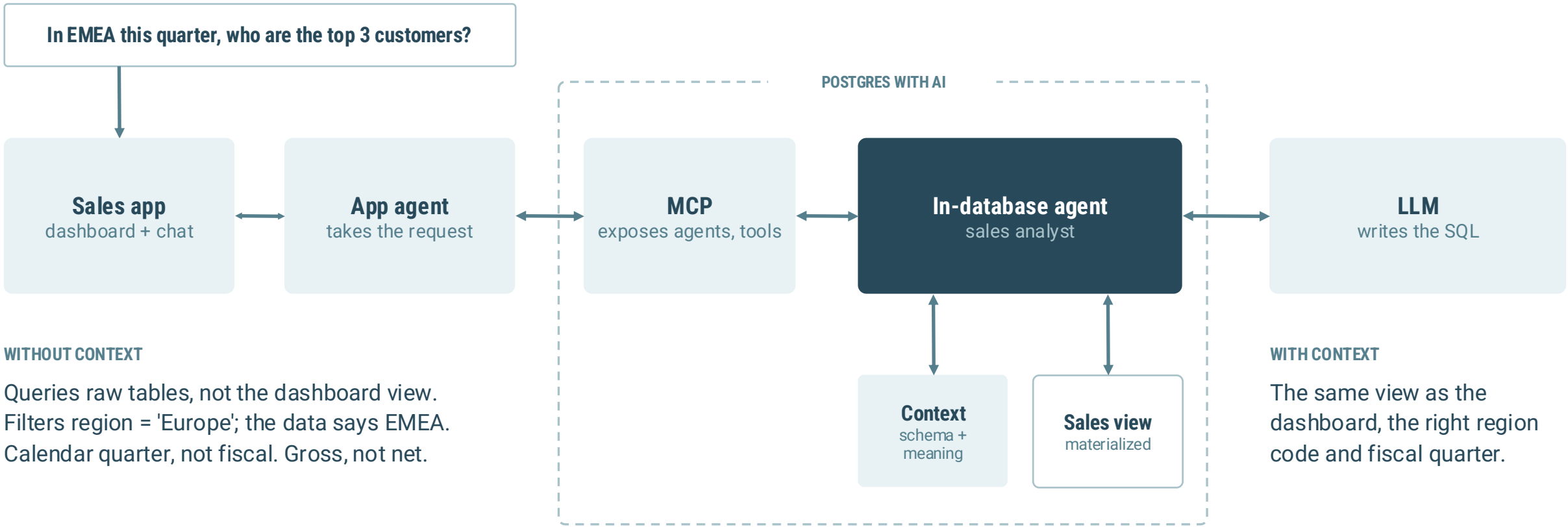
WHAT IT TAKES TO ENABLE AI IN POSTGRES



©EDB 2026 — ALL RIGHTS RESERVED.

USE CASE 1: SALES ANALYTICS

Intelligent Sales Dashboard



WITHOUT CONTEXT

Queries raw tables, not the dashboard view.
Filters region = 'Europe'; the data says EMEA.
Calendar quarter, not fiscal. Gross, not net.

WITH CONTEXT

The same view as the dashboard, the right region code and fiscal quarter.

Ask -> delegate -> call over MCP -> bundle context -> SQL back -> run -> answer

©EDB 2026 — ALL RIGHTS RESERVED.

USE CASE 2: DBA COPILOT

AT 3 AM, DBA On-Call

TODAY

Alert fires
pager at 3 am

→

Find the runbook
which one applies?

→

Read and interpret
which step fits this alert?

→

Adapt by hand
swap in real names

WITH A DBA COPILOT

Alert fires
same pager

→

Match the runbook
vector search

→

Add local context
live pg_stat data

→

Tailored steps
real PIDs, ready SQL

→

DBA decides
human in the loop

THE RUNBOOK SAYS

Find the blocker with pg_blocking_pids(), inspect its query and age, then decide whether to wait or terminate.

THE COPILOT SAYS (ILLUSTRATIVE)

```
PID 4821 blocks 3 sessions: idle in transaction for 14 min
on shop.customers.SELECT pg_terminate_backend(4821);
-- after you confirm
```

©EDB 2026 — ALL RIGHTS RESERVED.

INVESTMENT AREAS TO MAKE AI WORK IN THE DATABASE

MODEL	AGENT	CONTEXT
Small models that run well inside Postgres	Scoped agents, one job each	A semantic layer over schema and values
A governed registry of models	Tool and skill registries	Fast vector search at scale
Cost and latency you can predict	Human approval for risky actions	Metadata that stays fresh

ACROSS ALL THREE

Security and audit
who can ask what, and what ran

Accuracy evaluation
measure it, do not assume it

Open standards and hooks
so it is not a bespoke build

WHAT THIS UNLOCKS

- **Text-to-SQL-to-text**
- **DBA copilot**
- **Fraud detection**
- **Inline classification**
- **PII redaction**
- **Forecasting**
- **and many more**

EDB POSTGRES AI

©EDB 2026 — ALL RIGHTS RESERVED.

A DATABASE THAT RUNS ITSELF AND UNDERSTANDS ITSELF

In-database intelligence: operational and data

©EDB 2026 — ALL RIGHTS RESERVED.

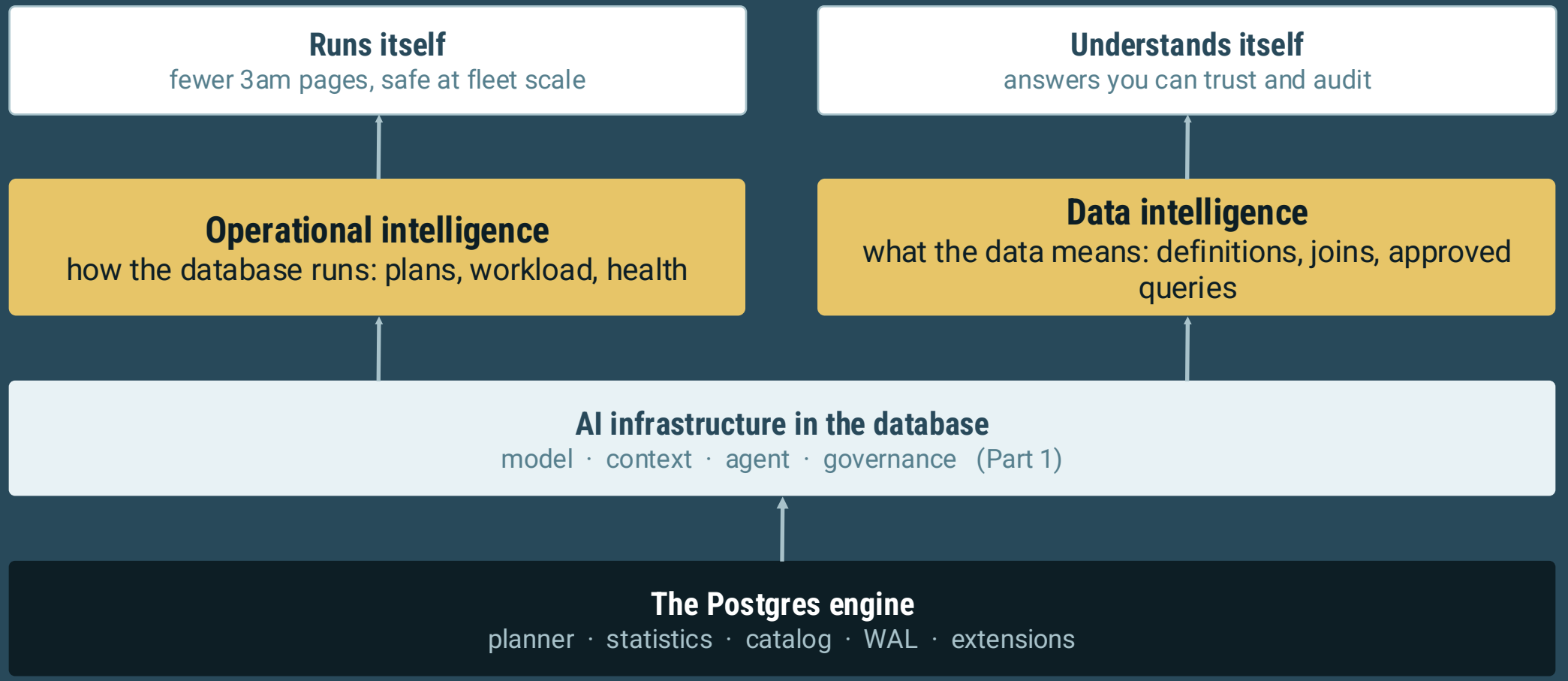
THE BOTTLENECK MOVED FROM THE ENGINE TO THE EXPERT

RUNNING IT	UNDERSTANDING IT
More Postgres instances than people who can tune them	Meaning lives in dashboards, dbt models and people's heads
Agents create databases, branches and load nobody planned for	Agents query a schema they cannot interpret
Tuning knowledge lives in a few heads and many runbooks	A wrong answer looks exactly like a right one

Postgres won the developer. Now it has to run itself and explain itself.

©EDB 2026 — ALL RIGHTS RESERVED.

TWO INTELLIGENCES, ONE CHASSIS, ALL INSIDE POSTGRES



©EDB 2026 — ALL RIGHTS RESERVED.

THE BEST OBSERVER OF POSTGRES IS POSTGRES

SEEN FROM OUTSIDE	KNOWN ONLY TO THE PLANNER
The query text	**Why it chose this plan,** and which ones it rejected
How long it took	**Where estimates and reality diverged,** for every plan shape
The plan it finally ran	**What a query would cost** with an index, a setting or a plan that doesn't exist yet

Tools outside see what happened. Only Postgres knows why.

The pieces already exist in Postgres: pg_stat_statements, auto_explain, HypoPG, and pg_plan_advice in PostgreSQL 19. It's time to connect them.

©EDB 2026 — ALL RIGHTS RESERVED.

AUTONOMY IS GATED BY REVERSIBILITY, NOT BY INTELLIGENCE

Level 0	Level 1	Level 2	Level 3	Level 4
Observe	Advise	Act with approval	Act within policy	Self-driving

MOST OF US ARE HERE, MOVING TO 2

Levels after the SAE framing Marc Linster and Luigi Nardi applied to Postgres at pgDay Paris 2026.

ORDER THE ACTIONS BY THE COST OF UNDOING THEM

Create an index	Pin a plan	Change a setting	Rewrite a table	Change schema or data
undo: drop it	undo: remove the advice	undo: reload, some need a restart	undo: costly, even online	undo: a human decision

cheap to undo → expensive to undo

Let agents climb the levels in the order their actions can be undone.

©EDB 2026 — ALL RIGHTS RESERVED.

DON'T OUT-REASON THE FRONTIER MODEL. OUT-CONTEXT IT.

When the answer is wrong, the model didn't fail. Nobody gave it the context.

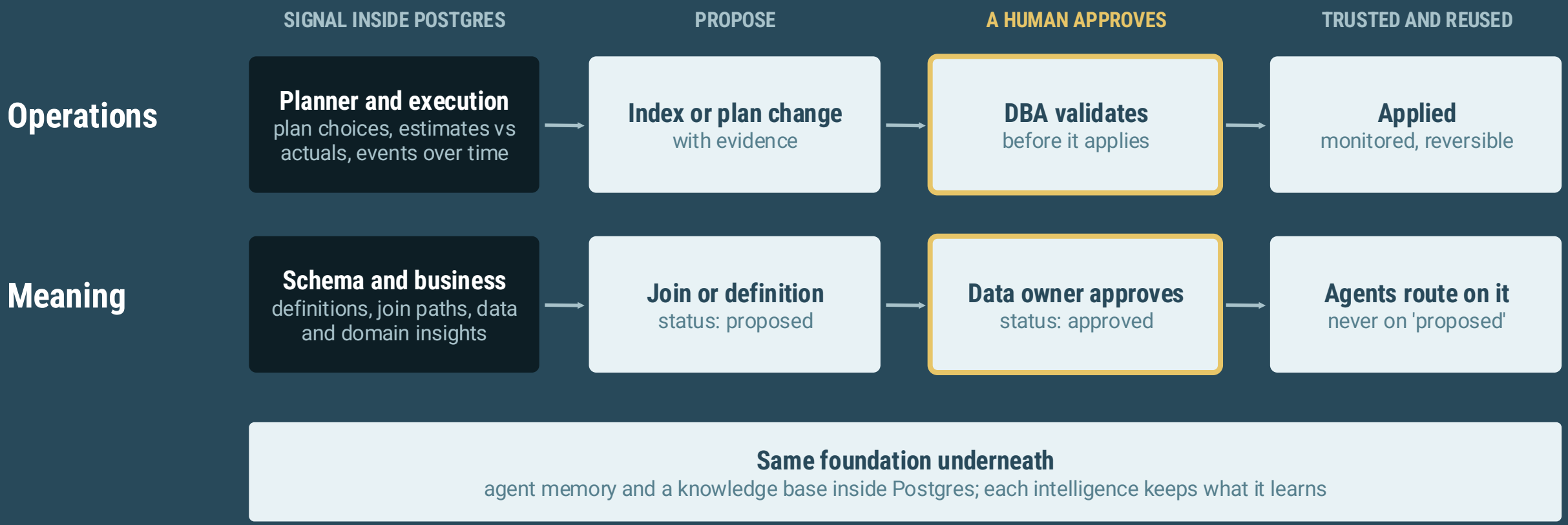
	SCHEMA ONLY, ANY MODEL	MEANING IN POSTGRES
Knows	table and column names	definitions, reviewed comments, approved join paths
Joins	inferred from names	routed over approved relationships, ranked by confidence
Recurring questions	generated fresh every time	resolved to a reviewed, parameterized query
Data path	schema and samples travel in prompts	the model can run local; meaning stays in Postgres

A wrong query doesn't throw an error. It returns a number.

©EDB 2026 — ALL RIGHTS RESERVED.

DIFFERENT SIGNALS, ONE RULE OF TRUST

Nothing is trusted until a human approves it. Then it is reused without asking again.



An index waits for the DBA. A definition waits for the data owner.

©EDB 2026 — ALL RIGHTS RESERVED.

A CHALLENGE TO THE COMMUNITY

THREE QUESTIONS BEFORE POSTGRES DRIVES ITSELF

01

Should intelligence live inside Postgres?

Or keep living in tools that watch it from the outside?

02

How does an agent earn the right to act?

Crawl, walk, run: advice first, act only where the action can be undone. No self-driving on mission-critical systems without proof.

03

How do we trust what an agent reads?

Accurate answers need meaning: definitions, join paths and data insights, owned and approved.

Autonomy isn't a feature you ship. It's trust you earn.

EDB POSTGRES AI

©EDB 2026 — ALL RIGHTS RESERVED.

THANK YOU

Bringing AI to Postgres.

Questions now · open floor at 12:00

EDB POSTGRES AI

©EDB 2026 — ALL RIGHTS RESERVED.