THE 6 PILLARS OF AI

AI adoption in simple terms.

Most AI failures aren't the model breaking — they're predictable, and avoidable. Six things every team should understand before they build, and the one lens we use to assess any system.

PILLAR 1

AI is broader than generative AI.

Generative AI — the kind most people use — writes by **predicting the next word; it predicts rather than knows**. Knowing that keeps your expectations realistic.

→ twicedata.com/labs/ai-is-broader-than-genai

PILLAR 2

It will confidently make things up.

Generative AI sometimes states false things in a convincing voice — it's built to sound **plausible, not to be correct**. You can't switch this off; you manage it by feeding it trusted data and checking its answers.

→ twicedata.com/labs/why-ai-hallucinates

PILLAR 3

"Is my data safe?" depends on where it runs.

Whatever you put into a generative AI tool can leave your company. **Free consumer chatbots may use it to train their models; paid business and developer plans can keep it private.** Where it runs decides who sees it.

→ twicedata.com/labs/is-my-data-safe

PILLAR 4

Use AI as a tool, not a master.

The costly failures happen when people **act on AI output without checking it**. Keep a person reviewing anything that really matters before it goes out.

→ twicedata.com/labs/use-ai-as-a-tool-not-a-master

PILLAR 5

The real cost is bigger than the usage fee.

The per-use price is the smallest part. **Preparing the data, testing quality, and human review** usually cost far more. Budget for the whole system, not the sticker.

→ twicedata.com/labs/the-real-cost-of-ai

PILLAR 6

Build AI you can account for.

When AI makes a decision, you should be able to **explain why and trace what happened**. Customers and new regulations increasingly require it.

→ twicedata.com/labs/accountable-ai

The CIDA lens — assess any AI system in four buckets

EXPANDED OVERLEAF →

C — **Context**
Where and how it's used — which sets the stakes.

I — **Inputs**
What data goes in, and where it's allowed to go.

D — **Decision procedure**
The model itself, and the rules around it.

A — **Action**
What's done with the output — and who checks it.

The six as one argument: AI is a prediction engine (1) that will confidently invent (2), running on data you have to protect (3), producing output you can't blindly trust (4), at a cost that hides below the sticker (5) — so build it to be accounted for (6).

Six-part series
twicedata.com/labs/pillars-of-ai

THE THROUGHLINE

CIDA — run it against any AI system, in four steps.

Because the interface hides the mechanism, you need a habit that makes it visible again before you trust an output. It is deliberately small enough to use in a meeting, and it works on any AI system — generative or not. **Three of the four you mostly assess. The Decision procedure is the one you build.**

C
Context
Where and how it's used. A throwaway draft and a clinical decision sit at opposite ends of the risk scale, even with the same model.

→

I
Inputs
What data goes in, and how it's filtered. Inputs set accuracy, leak private information, and decide what the system could possibly know.

→

D
Decision procedure
The model, its system prompt, the retrieval and grounding, the guardrails — the whole apparatus that produces the output. The biggest lever.

THE OUTPUT →

A
Action
What happens to that output. The hand-off — to a person, or to the next automated step — is where most real-world failures occur.

THE BUCKET	THE QUESTION YOU ASK	WHAT GOES WRONG IF YOU SKIP IT
C Context — YOU ASSESS IT	Where is this used, and what does a wrong answer actually cost *here*?	Scrutiny gets spread evenly instead of aimed — smothering the cheap decisions in needless review while the expensive ones go out unchecked. It is also where you find out whether a use case earns its total cost, which is **the cheapest money you will ever spend**.
I Inputs — YOU ASSESS IT	What data goes in — and where is it allowed to go?	Sensitive data walks out through a text box. Whatever protections you had inside your walls stop at the edge of them, and the model can only ever know what you actually gave it. **Samsung: three leaks in twenty days.**
D Decision procedure — YOU BUILD IT	What actually produces this output — which model, prompt, grounding and guardrails?	An ungrounded predictor optimizes for plausible, not true, with no alarm that fires when the two come apart. This is **the step you engineer and the one that moves the result most** — and it is where a data team earns its keep.
A Action — YOU ASSESS IT	Who or what checks this before it acts — and can that check say no?	Most real-world failures happen right here, in the hand-off. A reviewer rubber-stamping two hundred decisions an hour isn't checking them. Watch the rejection rate: **when the no-rate drifts to zero, the oversight is already gone.**

Run those four questions and the fog lifts. You stop asking "is the AI good" — an unanswerable question — and start asking **"is this the right kind of system for this context, fed the right inputs, with a sane check on the action it triggers."** That is a question a team can actually answer.

Which pillar lives in which bucket

1 Prediction
D · Decision procedure
The pillar that introduces the lens, and marks D as the biggest lever — the part you engineer.

2 Hallucination
A · Action
All the containment converges on the handoff, where something has to check the output before it acts.

3 Data safety
I · Inputs
What you feed the system and where you let it travel is the whole game — "almost entirely this bucket."

4 Oversight
A · Action
The same bucket as Pillar 2. Everything upstream can be perfect; the failure is in the hand-off from output to act.

5 Cost
C · I · D · A — all four
Doesn't live in a single square. Read the lens as cost centers: every bucket bills you.

6 Accountability
C · I · D · A — all four
Read the lens as an audit trail: each square is something you must be able to reconstruct and show.

Five questions worth asking before you ship

- → Which of our decisions does AI touch — and which run unsupervised?
- → Can we reconstruct *why* the AI did what it did last week?
- → When the model is wrong, what catches it before the customer does?
- → Where does our data go when we use it, and are we on the right tier?
- → What's the total cost — not the API fee — and is the use case worth it?