

The data capability layer

Four progressive capabilities sit in the foundation layer that every analytics, ML and AI capability stands on. Each answers a different question about the enterprise's relationship to its own data: can I get it in, can I trust it, can I find it, can I build on it. Each depends on the one before it being solid.

This is a working reference, not a summary. The targets below are illustrative starting points; the questions are the useful part. Take the one capability you are actively investing in, answer its questions honestly, and fill in its current-state column. That filled-in gap, not the illustrative target, is what should drive your next prioritization conversation.

THE FOUR QUESTIONS BEHIND EVERY ROW

What is it? The outcome the enterprise can produce or protect, not the tools used to produce it.

How well must we do it? The performance expectation, and the gap between current and desired state.

Why does it matter? Which lever it pulls: revenue, cost, service, compliance, or future positioning.

What could go wrong? In performing it, or in failing to perform it at all.

The four capabilities at a glance

CAPABILITY	THE OUTCOME IT PRODUCES	WHERE ITS VALUE CONCENTRATES	THE QUESTION THAT EXPOSES YOUR GAP
1 · Ingestion & Integration	Data of any shape, speed and source arriving intact and on time, without destabilizing the systems it came from.	Cost and reliability. The ceiling on everything above.	Of last month's pipeline failures, how many were caught before a report or model consumed the bad data?
2 · Quality, Modeling & Governance	Data that reaches a person or a model has already been validated, defined and assigned an owner.	Risk and compliance, and the trust level everything inherits.	If this dataset produced a wrong number in a board report tomorrow, whose job is it to explain why?
3 · Access & Discoverability	The people allowed to use the data can find it, understand it and use it themselves, without an engineering ticket.	Adoption. Whether trustworthy data actually gets used.	How many of last month's data requests could the requester have resolved alone?
4 · Strategic Data & Document Assets	Durable, versioned data products and knowledge assets that other teams and systems can safely depend on.	Future positioning, and the substrate for AI work.	If this table's structure changed tomorrow, how many teams would find out when their system broke?

CAPABILITY 1

Ingestion & Integration

The ability to get data of any shape, speed and source into the platform reliably, and shape it into a usable form, without destabilizing the operational systems it comes from. Streaming, batch, change data capture or API polling: the mechanism is an implementation choice below the capability line.

Nothing downstream can be governed, found or productized before it exists in the platform in the first place.

Reduce cost	Resilient, replayable pipelines cut the recurring cost of manual reruns, firefighting and one-off fixes whenever a source changes shape.
Improve service	Reliable, low-latency ingestion means reports, models and applications reflect current reality rather than a partial copy of it.
Achieve compliance	Controlled, auditable ingestion creates a defensible record of how data entered the enterprise, and from where.
Position for future	Every capability above this one is only as reliable as the ingestion feeding it.

PERFORMANCE DIMENSION	ILLUSTRATIVE TARGET	OUR CURRENT STATE
Pipeline reliability	≥ 99% of ingestion jobs complete without manual intervention	
Freshness and latency	Data lands within an agreed SLA of the source-system change	
Replay and recovery time	A failed window can be replayed without side effects inside a defined recovery time	
Source onboarding time	A new source can be connected and flowing within an agreed cycle time	

ASK BEFORE YOU SET A TARGET

- Of last month's pipeline failures, how many were caught by monitoring rather than discovered downstream by an analyst? A low number means the gap is detection and replay discipline, not pipeline volume.
- Can a failed window be reprocessed safely, or does replay create duplicates?
- When a source schema changes, who is accountable for noticing before it breaks a consumer?

CAPABILITY 2

Quality, Modeling & Governance

The ability to make ingested data trustworthy, well-structured, secure and compliant before anyone downstream relies on it: quality checks inline with processing, deliberate schema design, lineage tracking, and access controls enforced by default rather than bolted on after an incident.

This capability creates no new data and makes nothing easier to find. It is the gate that decides whether what flows through the platform can be trusted, and by whom.

Reduce cost	Catching defects inline is dramatically cheaper than a bad decision, a model retrain or a public correction afterward.
Achieve compliance	Enforced lineage, ownership and access control are frequently the explicit regulatory requirement, not an engineering nicety.
Improve service	An automatic governance gate releases new data faster, not slower. Governance debt is what actually slows delivery over time.
Position for future	Work built on ungoverned data doesn't just risk being wrong; it risks being wrong in a way nobody can trace.

PERFORMANCE DIMENSION	ILLUSTRATIVE TARGET	OUR CURRENT STATE
Automated quality coverage	Share of critical datasets with inline null, type and referential checks	
Ownership coverage	100% of production datasets have a named, accountable owner	
Lineage traceability	Share of datasets whose source-to-consumption lineage can be produced on demand	
Access control model	Share of access granted under least-privilege policy rather than one-off manual grants	

ASK BEFORE YOU SET A TARGET

- How many production datasets have no named owner? If this one produced a wrong number in a board report tomorrow, whose job is it to explain why?
- Are quality checks and access controls enforced centrally, or dataset by dataset, so two “governed” datasets meet two different standards?
- Is ownership assigned, or assumed? Assumed ownership is what turns a broken dataset into a standoff.

CAPABILITY 3

Access & Discoverability

The ability to make governed data easy for the right people to find, query and consume: a catalog, defined data contracts, and self-service query and preparation tools, without an engineering ticket for every request.

This capability doesn't decide whether data can be trusted. It decides whether trustworthy data gets used at all, or sits idle because nobody outside the team that built it knows it exists.

Reduce cost	Self-service discovery deflects the steady stream of “where do I find X” and “can you pull this” requests.
Improve service	Turns the gap between needing data and using it from a ticket-and-wait cycle into a search-and-query action.
Achieve compliance	A catalog with visible access rules lets you show who can see what, and why, on demand rather than after the fact.
Position for future	Decides whether the analytics, ML and AI work above it is adopted. A governed dataset nobody can find generates nothing.

PERFORMANCE DIMENSION	ILLUSTRATIVE TARGET	OUR CURRENT STATE
Catalog coverage	Share of production datasets registered with an owner, description and access path	
Self-service resolution rate	Share of discovery and access requests resolved without an engineering ticket	
Time to access	Median time from a request to a granted, usable connection	
Data contract adherence	Share of consumed datasets under a versioned contract rather than a freely changing schema	

ASK BEFORE YOU SET A TARGET

- Of last month's “can you get me this data” requests, how many could the requester have resolved if the dataset were cataloged and access pre-approved?
- Is discoverability a metadata layer, or asking around in chat and getting a different answer depending on who replies?
- Are data contracts enforced, or can a producing team change a schema and break every consumer at once?

CAPABILITY 4

Strategic Data & Document Assets

The ability to turn governed, discoverable data into durable, versioned assets other teams and systems depend on: data products and APIs under a defined contract, disciplined lifecycle management, living documentation and lineage, and vectorized representations of enterprise content for LLM-powered applications.

A dataset promoted to “strategic asset” before ingestion, governance and discoverability are solid is really just a table other teams have started depending on without a contract.

Position for future	Versioned data products and knowledge bases are the substrate every later analytics, ML and GenAI initiative builds on.
Reduce cost	Retention, tiering and versioning discipline prevents the unbounded storage cost of data kept forever because nobody decided otherwise.
Improve service	A versioned contract lets consumers build with confidence instead of treating every dependency as a risk.
Achieve compliance	Living documentation, lineage and retention are often the exact artifact an auditor asks for, built once as normal operation.

PERFORMANCE DIMENSION	ILLUSTRATIVE TARGET	OUR CURRENT STATE
Data product coverage	Share of high-reuse datasets promoted to versioned, contract-backed products	
Dependency count	Teams or systems depending on a product under a defined contract, as an adoption signal	
Documentation currency	Share of products whose lineage and dictionaries are kept current by the pipeline itself	
Lifecycle discipline	Share of datasets under an active retention or tiering policy	

ASK BEFORE YOU SET A TARGET

- If this table's structure changed tomorrow, how many teams would find out only when their system broke? More than zero means the gap is the contract, not the data.
- Is documentation kept current by the pipeline, or written once at launch and quietly stale within months?
- Who decides which datasets get promoted to a governed product, and which stay internal working tables?

Four things that hold across all four

Trust compounds upward, and so does its absence.

A strategic data product built on undiscoverable, ungoverned, unreliable ingestion inherits every one of those weaknesses at once, and so does every analytics, ML and AI capability above it. Fixing a distrusted asset almost always means fixing an ingestion or governance gap first, not adding more documentation.

Value shifts from keeping the lights on to building the future.

Ingestion and governance mostly reduce cost and risk by making sure data arrives and can be trusted. Access and strategic assets start enabling new value directly, faster time-to-insight and new AI use cases, because they decide whether trustworthy data gets used and built upon rather than merely existing.

Risk changes in kind, not just in degree.

Ingestion risk is mostly reliability and replayability. By strategic assets it is an external team depending on something never meant to be depended on, with no contract to protect either side when it changes. One review process across all four slows routine pipeline work and still under-governs the highest-stakes dependencies.

A gap is only as good as the current-state number behind it.

Every illustrative target here needs a real current-state figure before it becomes actionable. The fastest route to usefulness is to fill in the current-state column for the one or two capabilities under active investment this planning cycle, not to complete all four at once.

DO THIS NEXT

Pick the single capability under active investment or debate right now. Complete its performance table with real current-state numbers from whatever monitoring exists today, even informal monitoring. That filled-in gap, not the illustrative targets above, is what should drive the next round of prioritization and investment conversations.