

JOSHUA BURKHOW

Chief Evangelist

Joshua.burkhow@alteryx.com

BYOAI

Building the AI-Agnostic Analyst Stack

alteryx

alteryx.com

ABOUT THE SPEAKER

Joshua Burkhow



WORK HISTORY

Analytics + Data Science + AI

Chief Evangelist @ Alteryx
20+ years building analytics practices across
Fortune 500 enterprises. Nike, PwC, Fujitsu

PUBLISHED

O'Reilly Author + Podcaster

Book: Alteryx Designer: The Definitive Guide
Writes on data, analytics, and the operating
model behind AI-driven work.
Podcast: Alter Everything – AI, Analytics, and
the Future of Work

BACKGROUND

U.S. Marine Veteran

Discipline, Details, bias for action. Carries
into how I think about systems.

Pick your favorite AI.

None of them does absolutely everything.

Frontier models are specialized — and a new one ships every week.

Claude

Reasoning, long context, tool use

Gemini

Multimodal, search-grounded retrieval

ChatGPT

Conversational breadth, ecosystem

Codex / Claude Code

Code generation and refactoring

No one runs a business on a single data source.

Your data estate is already heterogeneous. You designed it that way on purpose.

ERP

SAP, Oracle, NetSuite

CRM

Salesforce, HubSpot, Dynamics

Warehouse

Snowflake, BigQuery, Redshift

Lakehouse

Databricks, Iceberg, Delta

Operational

Postgres, MongoDB, Kafka

SaaS APIs

Stripe, Workday, ServiceNow

Files

S3, SharePoint, Drive

Streaming

Kinesis, Pub/Sub, Confluent

You picked the right tool for each job — then built integration, governance, and lineage on top.

THE SAME PATTERN, NOW WITH AI

The AI ecosystem looks exactly like your data estate.

Specialized providers, overlapping capabilities, weekly change. Lock-in is the same trap, one layer up.

DATA LAYER		AI LAYER	
Transactions	ERP / OLTP	Long reasoning	Claude, GPT-5 reasoning
Customer	CRM / CDP	Multimodal grounding	Gemini, GPT-4o
Analytics	Warehouse / Lakehouse	Cheap classification	Haiku, Gemini Flash, OSS
Real-time	Streams / Queues	Code generation	Codex, Copilot, Sonnet
Unstructured	Files / Object Store	Private / sovereign	Self-hosted Llama, Mistral

We abstracted the data layer. We have not yet abstracted the AI layer.

Three risks that compound silently.

01

Capability Drift

Model providers retrain, deprecate, and re-tune. The same prompt against the same model returns different answers month over month — and you find out from a stakeholder, not a release note.

02

Vendor Risk

Pricing changes. Rate limits change. Regions change. SOC2/ISO posture changes. Every one of these is a single-vendor decision you can't influence.

Tokens anyone?

03

Auditability Collapse

When the rule lives in the prompt, you can't diff it, version it, or defend it. Regulators and auditors don't accept 'the model decided' as a control.

Three layers. Isolate the volatile one.

The AI layer changes weekly. The logic and data underneath should not.

AI Layer

INTERCHANGEABLE

Claude · Gemini · GPT · Codex · OSS

Routing, prompting, and tool-use across many providers. Treat as commodity. Replace any model in 10 minutes.

Business Logic Layer

GOVERNED · AUDITABLE · PORTABLE

Semantic · workflows · rules · contracts · lineage

Metrics, transformations, rules, quality, and lineage. Owned by the business. Versioned, tested, and visible.

Data Foundation

TRUSTED · ACCESSIBLE

Warehouse · Lakehouse · ERP · streams

The sources of record and the platform that serves them. Quality, access, and freshness are foundational.

The Business Logic Layer

What lives here, why it matters, what it looks like in code.

alteryx

alteryx.com

Five components. All of them must be code or friendly, not prompts.

If it can't be diffed, tested, and versioned, it doesn't belong in the BLL.

01

Semantic Layer

Metrics, dimensions, and entity definitions
— declared once, served everywhere.

revenue · churn · ARR · retention · NPS

02

Transformation Pipelines

Deterministic, idempotent ELT/ETL. Inputs
and outputs are both contracts.

stage → conform → enrich → publish

03

Business Rules Engine

Externalized decision logic. Eligibility,
pricing, routing, segmentation.

if/else as data, not as prompt

04

Data Quality & Contracts

Validation, expectations, and freshness
checks at every boundary.

unique · not_null · domain · freshness

05

Lineage & Observability

Column-level lineage, run history, SLA, and
incident replay.

where did this number come from?

The AI calls into the BLL

Tool use, function calling, and MCP turn the BLL into a stable interface any model can consume.

01

Function Calling

BLL exposes typed functions (`get_metric`, `run_rule`, `query_contract`). Every major model speaks this protocol.

```
metric.get('net_revenue', region='NA')
```

02

Model Context Protocol

MCP servers wrap the BLL. Any MCP-aware model — Claude, IDE agents, custom — gets the same surface area.

```
mcp://bll/metrics • mcp://bll/rules
```

03

Retrieval over Catalog

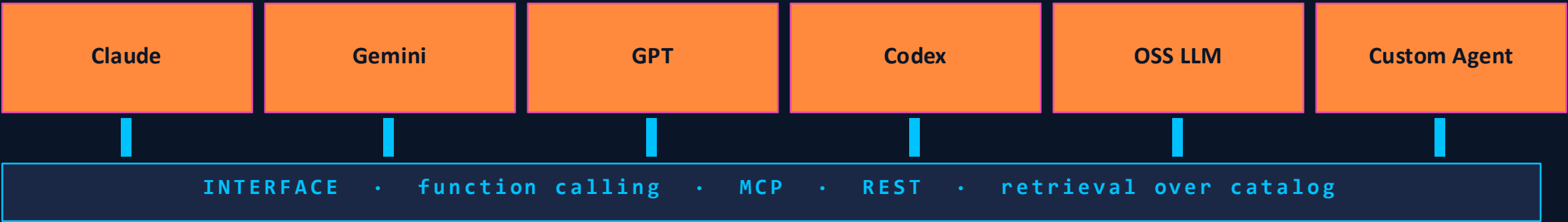
Catalog entries (definitions, contracts, lineage) are the AI's reference docs. RAG into them, never against raw rows.

```
search('definition of churn')
```

*Rule of thumb: if the AI can do its job by talking only to the BLL, **you can swap models tomorrow.***

The AI-Agnostic Analyst Stack

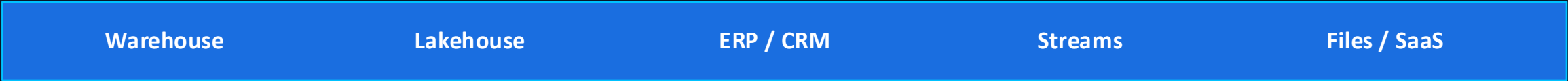
AI LAYER · swappable



BUSINESS LOGIC LAYER · governed, auditable, portable



DATA FOUNDATION · trusted, accessible



Same question. Same logic. Different model. Same answer.

If the answers diverge, the BLL is incomplete — not the model.

01

Same Question

What was Q1 net revenue in
North America for Product Line A?

02

Same Logic

Both calls resolve to
metric.net_revenue with the same
dimensions and the same date
range.

03

Different Model

Run once against Claude. Run the
identical request against Gemini.

04

Same Answer

Numbers match. Provenance
matches. Confidence levels match.
The AI was a thin client.

ARE YOU READY?

Five signals your stack is model-agnostic.

If you can answer yes to all five, you can swap the AI layer in a sprint. If you can't, that's your roadmap.

01	Quality measured	Data quality has a score, is tracked, and is acted on. Not assumed — verified.
02	Definitions documented	Business terms live in a catalog anyone (and any AI) can query.
03	Lineage tracked	Every metric traces back to its source. Audit-ready, model-retraining-ready.
04	Logic externalized	Rules and metrics live in code or config — not in prompts.
05	Models tested	Same business question runs through 2-3 models on every change. Divergence is a bug.

A 4-step path you can begin Monday.

Sequenced. Each step makes the next one cheaper.

01

Audit your data foundation

Quality, access, lineage. Score your top 20 datasets honestly. Know what you have before layering AI on top.

WEEKS 1-2

02

Externalize one metric, end to end

Pick one painful metric. Move its definition out of dashboards into a semantic layer. Wire one AI call to it. Prove the pattern.

WEEKS 3-6

03

Standardize the interface

Wrap the BLL behind function-calling / MCP. Make it the only way the AI gets data. Document the contract.

WEEKS 7-10

04

Test across models

Run the same question through 2-3 models on every change. Make divergence a CI failure, not a surprise.

ONGOING

Four questions that will likely make your team uncomfortable

01

If your preferred AI model became unavailable tomorrow, what specifically would break?

02

Can you explain where a key metric came from — without calling the person who built it?

03

Do your business rules exist outside of prompts, or only inside the AI?

04

Have you tested the same business question across more than one model? When?

JOSHUA BURKHOW

Chief Evangelist

Joshua.burkhow@alteryx.com

THANK YOU!
(Connect with Me ;)

