The Open Agentic Platform

Building an open data and AI ecosystem



Alex Merced

Head of Developer Relations, Dremio

SPEAKER PROFILE

About the Speaker



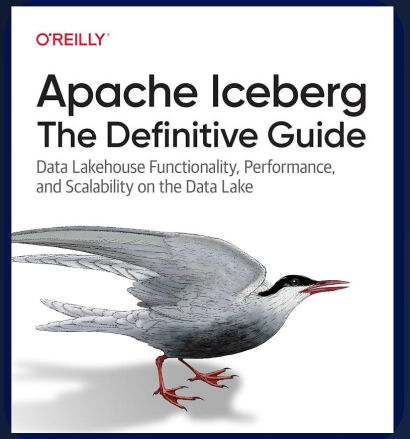
dremio
An SAP Company

Alex Merced

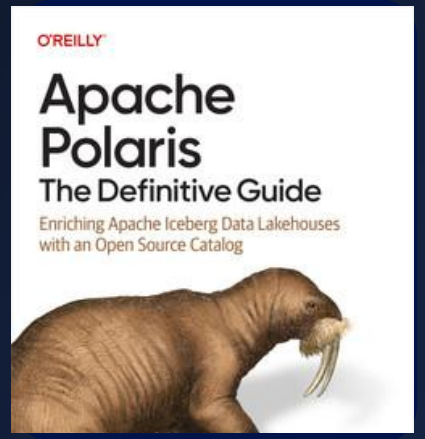
Head of Developer Relations (DevRel), Dremio

Alex is a prominent advocate and technology leader in the data lakehouse space, dedicated to helping organizations transition to modern, open data architectures. As a developer, educator, and author, he specializes in distributed query engines, open table formats, and cloud analytics platforms that empower the next generation of AI and data intelligence.

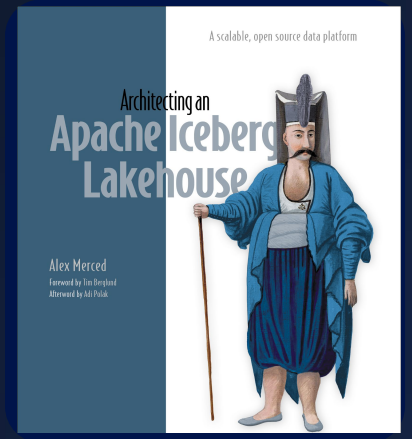
FEATURED PUBLICATIONS



Apache Iceberg
The Definitive Guide



Apache Polaris
The Definitive Guide



Architecting an Iceberg
Lakehouse

RESOURCE DOWNLOAD

Get Your Free Copy



Featured Guide

The Open Lakehouse Architecture

Dive deep into the next generation of data engineering. Learn how to leverage open table formats, governed catalogs, and agentic workflows to build scalable data platforms.

WHAT YOU'LL LEARN:

- Building scalable agentic infrastructure
- Optimizing query performance with Apache Iceberg
- Implementing robust governance via Apache Polaris

Find More Useful Free Guides at

Developer.Dremio.com

Scan to Download

Point your camera at the QR code to access the instant PDF download.

Instant Access

THE SHIFT

Agents are becoming a new interface to enterprise data



Discover

Find the right table, metric or document without knowing where it lives.



Reason

Interpret results using what the business means, not just what the schema says.



Act

Take multi-step actions on that data, inside limits someone can audit.

THE GAP

An LLM wired to a database is not yet an agentic system

01

Open data formats

One copy of the data any engine or tool can read.

02

Governed catalogs

Permissions enforced below the agent, not in its prompt.

03

Shared semantics

One definition of revenue for people, BI and agents.

04

Portable knowledge

Context that outlives a session and a vendor.

05

Transparent agents and work

Plans and agent definitions you can read, review and move.

THE ARCHITECTURE

One stack, built from open parts

The Open Agentic Lakehouse joins open lakehouse technology with open agent standards. Every layer is a spec, so every layer is replaceable.

AGENTIC LAYER

Agent profiles	Open Agent Profile (OAP)
Execution graphs	Agentic Graph Spec (AGS)
Knowledge graphs	Persistent context

DATA LAYER

Semantics	Apache Ossie
Catalog and governance	Apache Polaris
Data movement	Apache Arrow
Tables	Apache Iceberg
Storage	Apache Parquet

DATA LAYER · STORAGE AND TABLES

Parquet stores the data. Iceberg makes it a table.

Apache Parquet

COLUMNAR FILE FORMAT

- Reads only the columns a query needs
- Row-group stats let engines skip data
- Strong compression and encoding
- Readable by nearly every engine

Apache Iceberg

OPEN TABLE FORMAT

- ACID commits through snapshots
- Schema and partition evolution
- Time travel to any prior snapshot
- Many engines read and write one table

For agents: one governed copy of the data, reachable from whatever engine an agent’s tools call.

DATA LAYER · CATALOG AND GOVERNANCE

Apache Polaris is the open catalog

Iceberg REST catalog
Any engine that speaks the REST spec can find and commit to the same tables.

Role-based access control
Privileges on catalogs, namespaces and tables, held centrally.

Credential vending
Engines get short-lived, scoped storage credentials, never standing keys.

Semantic models in the catalog
Ossie documents stored next to the tables they describe. [Confirm status before the talk]

WHY IT MATTERS TO AGENTS

An agent’s access is enforced at the catalog, not trusted to its prompt.

Top-Level Apache project since February 2026.

DATA LAYER · DATA MOVEMENT

Apache Arrow moves results without the translation tax

Columnar memory

One in-memory layout shared across languages and processes, often with zero copies.

Arrow Flight

High-throughput transport for result sets, with Flight SQL for query access.

ADBC

A database connectivity API that returns Arrow data instead of rows.

For agents: tools pass results between steps without re-serializing every row.

DATA LAYER · SEMANTICS

Apache Ossie gives people and agents one meaning

- Vendor-neutral YAML or JSON for datasets, fields, relationships and metrics
- **ai_context** carries instructions, examples and synonyms for agents
- A spec, not a runtime: converters emit each tool's semantic layer
- Formerly Open Semantic Interchange; incubating at the ASF since June 2026

```yaml
semantic_model:
  - name: sales_analytics
    datasets:
      - name: orders
        source: lake.sales.orders
    metrics:
      - name: total_revenue
        expression:
          dialects:
            - dialect: ANSI_SQL
              expression: SUM(amount)
        ai_context:
          synonyms: ["revenue", "sales"]
```

ABOVE THE DATA LAYER

The data is open. Most agent stacks still keep three things private.

Knowledge

What the agent knows about your business

The plan

What the agent intends to do, step by step

The agent

Who it is, what it may touch, what it learned

AGENTIC LAYER · KNOWLEDGE

Knowledge graphs hold context that persists

Entities

People, products and systems, linked to catalog tables and Ossie metrics.

Relationships

Who owns what, what depends on what, and which source is authoritative.

Provenance

Decisions and findings with their source, so the next session starts from evidence.

Rule: reference governed data by catalog ID. Never copy it.

AGENTIC LAYER · KNOWLEDGE IN PRACTICE

Two open paths to a knowledge graph

PATH 1 · METADATA PLATFORM

OpenMetadata

- Harvests metadata from many systems
- RDF and OWL graph, SPARQL queries
- Glossary, lineage and ownership
- An MCP server for agents

Best when: you need one graph across the whole data estate, not just the lakehouse.

PATH 2 · CATALOG NATIVE

Apache Polaris + Apache Ossie

- Ossie models stored in the catalog
- Metrics and joins next to their tables
- Polaris access rules govern meaning
- Engines and agents read one API

Best when: you want meaning governed and versioned with the data itself.

> Either way, the graph references governed data by ID. Neither path copies it.

AGENTIC LAYER · EXECUTION

Today the plan lives inside the harness

1 **You can’t review it before paying for it**
By the time you see the steps, the tokens are spent.

2 **You can’t move it**
A plan from one harness means nothing to another.

3 **“Done” is whatever the model says**
Without acceptance criteria, completion is a claim.

4 **Every step gets the same model**
Overspend on trivia, underspend on the decision that mattered.

AGENTIC GRAPH SPECIFICATION (AGS)

The plan becomes a reviewable graph

A directed acyclic graph. Each node is one bounded agentic loop, and it carries:

- **A precise brief**
- **Typed inputs and outputs**
- **Success criteria**
- **An intelligence tier**
- **Permissions and budgets**
- **Failure handling, gates**

```
ags_version: "1.0"
kind: AgenticGraph
nodes:
  revenue_brief:
    title: Draft Q3 revenue brief
    intelligence:
      tier: advanced
    requirements:
      tools: [sql_query, file_write]
    success:
      criteria:
        - id: finance_signoff
          kind: human
          description: Finance confirms
            the figures match the
            certified metric.
```

AGENTIC GRAPH SPECIFICATION (AGS)

Done is a check, and capability is a tier

Nine criterion kinds, evaluated by the harness

`command` `file_exists` `artifact_present` `json_schema` `regex` `expression` `llm_judge` `human` `external`

A failed check’s description and evidence go into the next attempt, so a retry tries something new.

Tiers, never model names

Tier	Use when the step is
minimal	Mechanical, mistakes are obvious
standard	Routine, with a known pattern
advanced	Multi-step reasoning, some ambiguity
frontier	Novel, high stakes, errors hard to see

A harness that can’t meet a tier fails the node before spending tokens. No silent downgrades.

OPEN AGENT PROFILE (OAP)

The agent becomes a file

A harness reads the profile to start a fresh session, then writes a new revision back when the session ends. No process stays resident.

- **Identity and role**
- **Instructions**
- **Model, with tier fallback**
- **Tools and permissions**
- **Attached context**
- **Learned state**

```yaml
oap: "1.0"
kind: AgentProfile
metadata:
  name: revenue-analyst
  revision: 7
spec:
  model:
    tier: advanced
  tools:
    policy: allowlist
    allow: [sql_query, read]
  lifecycle:
    writeback: propose
state:
  facts:
    - text: Fiscal year starts Feb 1.
      confidence: 0.9
```

OPEN AGENT PROFILE (OAP)

Three rules make a portable agent safe



A profile narrows, never widens

The harness grants the intersection of what the profile asks for and what local policy allows.



An agent can’t rewrite its contract

Sessions may change only state. Tool or permission changes become proposals a human approves.



Learned state is untrusted

What an agent wrote about itself is information, never authority.

PUTTING IT TOGETHER

One question, every layer

1 · OAP

The harness loads the analyst's profile, narrowed by local policy.

2 · AGS

The agent drafts a graph for the work. A person approves it.

3 · Ossie + knowledge graph

Each node resolves "revenue" to the certified metric.

4 · Polaris

The catalog checks privileges and vends scoped credentials.

5 · Iceberg, Parquet, Arrow

Any engine reads the table; results flow back as Arrow.

6 · Back to the top

Criteria are checked; new facts are proposed to the profile.

THE PAYOFF

Neither your data nor your agents are trapped

What you own	Open format	Where it can live
Data	Parquet and Iceberg	Your object storage
Access rules	Polaris, Iceberg REST	Your catalog
Meaning	Apache Ossie	The catalog, or git
Knowledge	RDF and OWL	OpenMetadata, any triplestore
Plans	AGS	Git, reviewed like code
Agents	OAP	Git or ~/.agentprofiles

NEXT STEPS

Four moves for this quarter

Open the tables

Land key data in Iceberg behind an Iceberg REST catalog such as Polaris.

Write down meaning

Define your ten most-asked metrics in Ossie, with ai_context.

Review one plan

Express one recurring agent workflow as an AGS graph before it runs.

Save one agent

Turn your most useful agent into an OAP profile and commit it.

Thank you

Feedback on AGS and OAP is welcome, especially from harness and catalog builders.

github.com/AlexMercedCoder/agentic-graph-spec

github.com/alexmerced-oss/open-agent-profile

books.alexmerced.com

ALEX MERCED

Head of Developer Relations, Dremio

Apache Iceberg: The Definitive Guide

Apache Polaris: The Definitive Guide

Architecting an Apache Iceberg Lakehouse



alexmerced.com