AIRFLOW SUMMIT

Agentic AI Automating Semantic Layer Updates with Airflow 3

3.0

Andres Astorga Espriella
Soren Archibald

Agenda

- Problem Space
- Solution
- Demonstration
- What did we learn?
- Additional applications
- Questions?

Problem Space

Even with powerful tools like Airflow, keeping pipelines healthy still consumes a lot of engineering time.

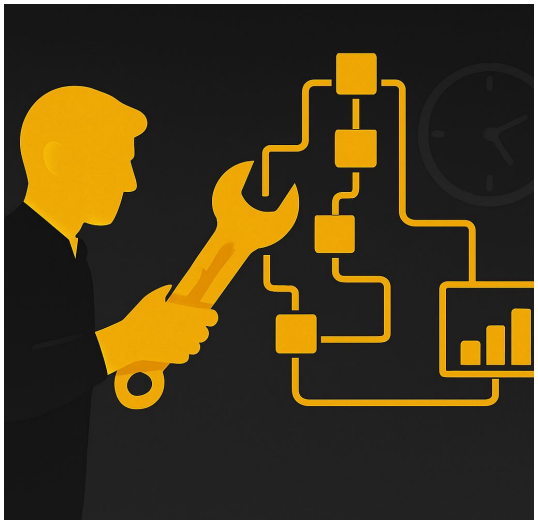
- **Find:** Logs, configs, and DAG code are scattered.
  - It takes minutes to hours to notice patterns like flaky tasks, dependency bottlenecks, or upstream changes that break downstream jobs.
- **Fix:** Root causes hide in long dependency chains.
  - The more time it takes to find the problem — the longer business reports are delayed.
- **Prevent:** — Fixes are often one-off.
  - Without tests, monitoring, or shared knowledge, the same failures come back and keep disrupting workflows.

Airflow has all the data about these problems — but right now, only humans can interpret it.

Problem Space

Find

Fix



Prevent

Solution Space

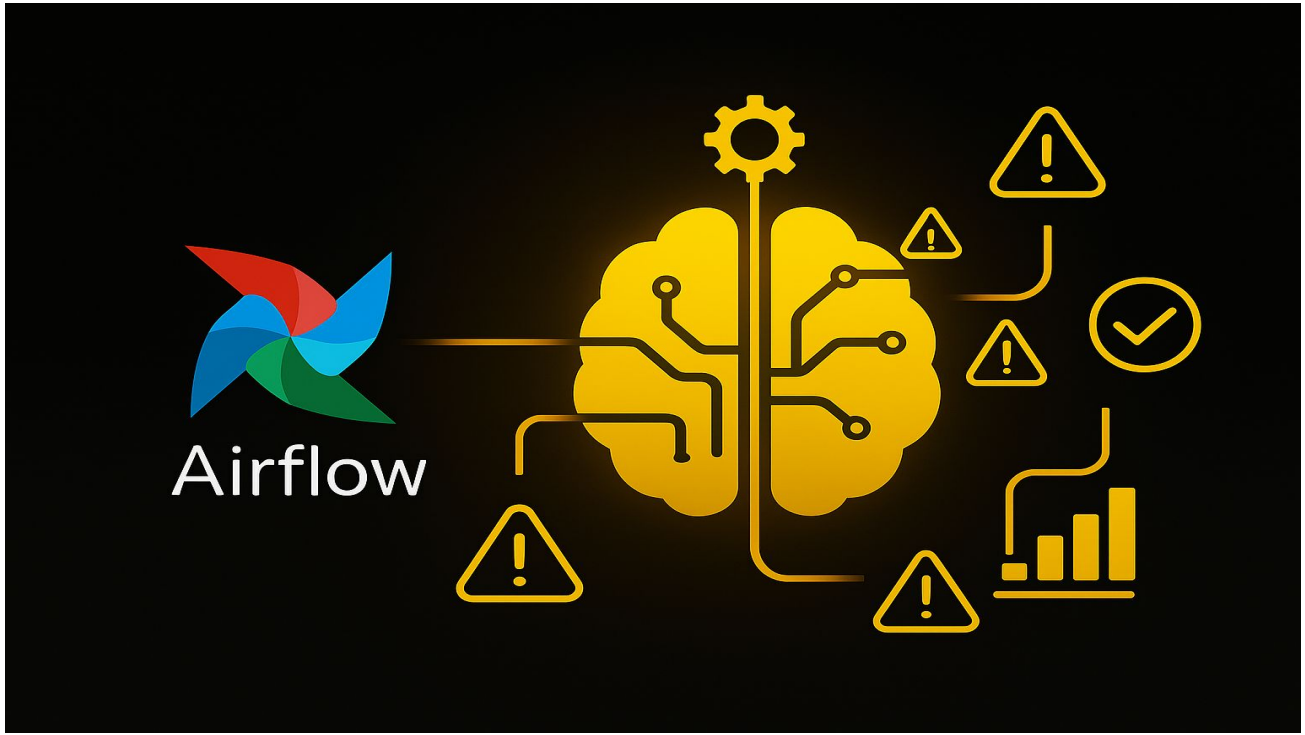
What if an AI agent could:

- **Find opportunities** by analyzing Airflow's logs, DAG definitions, configs, and historical runs to spot failures, bottlenecks, and risky patterns

- **Fix issues faster** by identifying the root cause, suggesting solutions, and even applying safe automated fixes upstream before they impact business teams

Instead of just reacting to failures, we could prevent them — and free engineers to focus on building, not debugging.

Solution Space



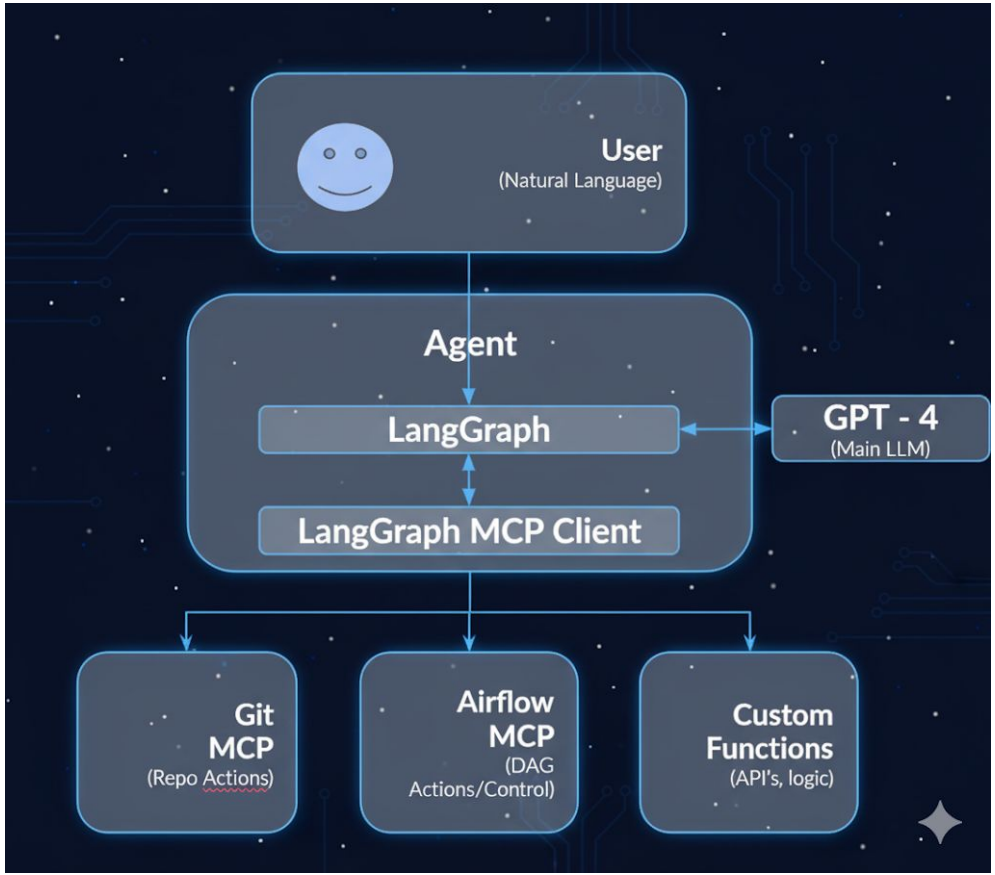
From Agents to Actions

Rise of AI Agents

- Work directly with your applications on your behalf
- Act as a bridge between LLMs and MCPs
- Use MCP metadata to decide which API or function to call

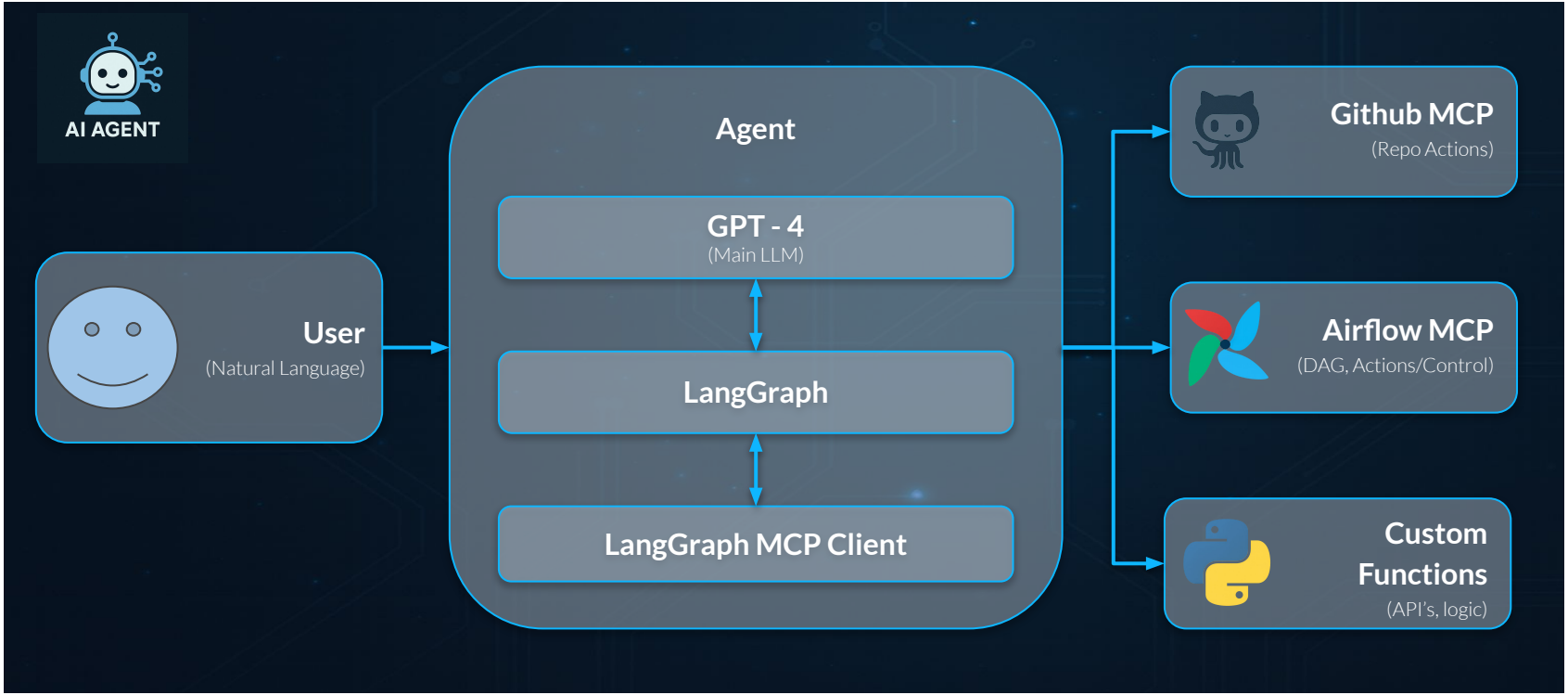
MCP Servers

- Provide a standard interface for connecting AI agents to applications
- Expose function names, descriptions, and metadata so agents have the right context to act



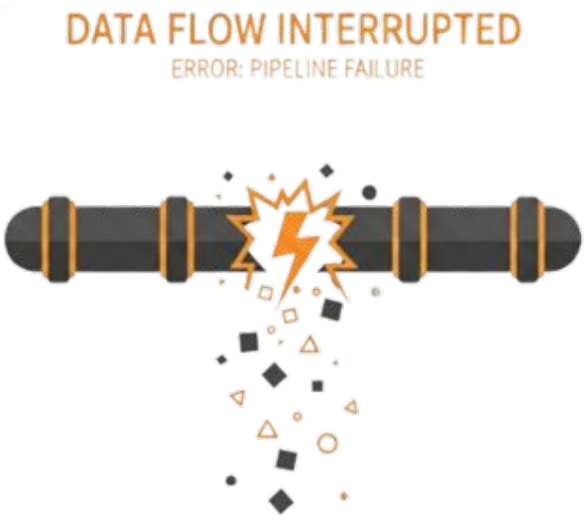
Qbiz

From Agents to Actions



Qbiz

Example Scenario



A critical pipeline breaks due to a key column being renamed. That small tweak sets off a chain reaction:

- **Log errors:** Downstream DAGs start failing with “column not found” errors
- **Pipelines stuck:** Dependent jobs stop running, critical reports are running late!
- **Manual cleanup:** Logs to parse, jobs to update, stakeholders to update

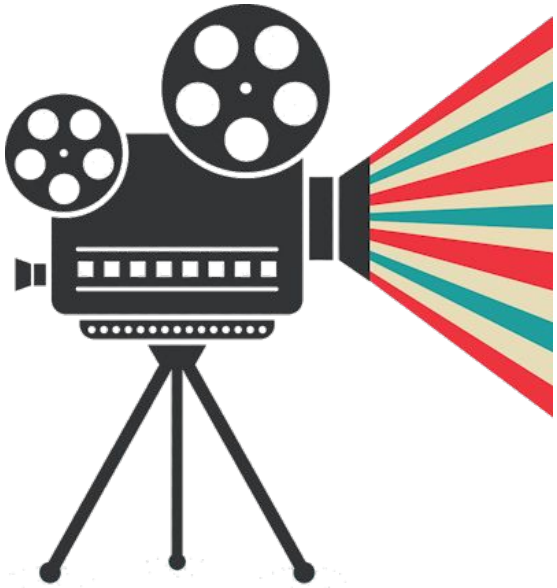
Agentic Solution

We built an AI-powered agent to make Airflow pipeline maintenance faster and easier. Think of it as a self-correcting system with human oversight. Here's what it does:

- **Smart error detection:** Scans Airflow logs for failures and finds root causes
- **Fix suggestions:** Proposes specific fixes (code snippets, config tweaks, etc.)
- **Human-in-the-loop:** Nothing changes automatically — a person always reviews and approves suggestions.
- **Auto pull requests:** If approved, creates a PR and implements all fixes

Demo

We present a short demonstration of a proof-of-concept we built.



Demo Recap

1. We saw the agent load tools that interface with the APIs of multiple applications – GitHub and Airflow.
2. The user asks the agent to scan and analyze error logs.
3. The agent analyzes the logs and proposes some fixes based on the errors to the user.
4. The user authorizes the agent to create a branch, fix the code and create a pull request.
5. The user reviews the PR.

Prompt Engineering

Best Practices

- Ask the agent what tools are available.
- Tell the Agent which MCP server or tools you want to use.
- Be explicit with input for that tool, e.g.:

  action: get_dag_run

  input : dag_id=nightly_process, dag_run=yesterday

Reliability

- Improves with with more context – “Apache Airflow” not just “Airflow”.
- Various levels of prompt engineering depending on LLM.

Various LLMs performance

LLM	Loaded Tools	Thinking	Complexity
Llama 3.2:3b	✅	❌	❌
Qwen3:8b	⚠️	✅	❌
GPT4o	✅	✅	✅
Claude Opus	✅	✅	✅

Challenges

Finding the best LLM-MCP-Agent combination – we tested 4, 5 and 3, respectively.

Agent:

- Enriching descriptions of tools to avoid agent confusion
- When you provide more than 128 tools -> Error. Prefilter of tools before send request to LLM

MCPs:

- No “official” MCP server for Airflow or Slack.
- Available Airflow MCP Servers are pre v1.0 and have quality and documentation issues.
- Security: how much power should they have?

LLMs:

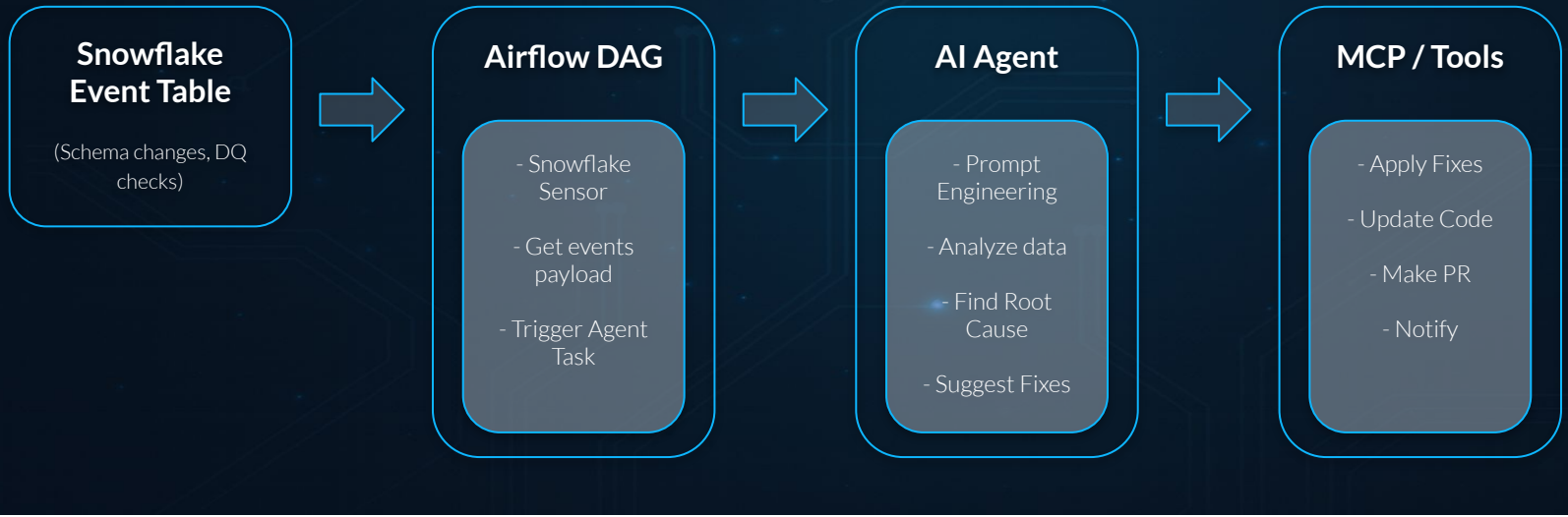
- LLM subscription rate limits.
- System memory for local LLMs.
- Smaller, local models don’t perform well; larger, more expensive, remote models perform much better.

Additional Applications

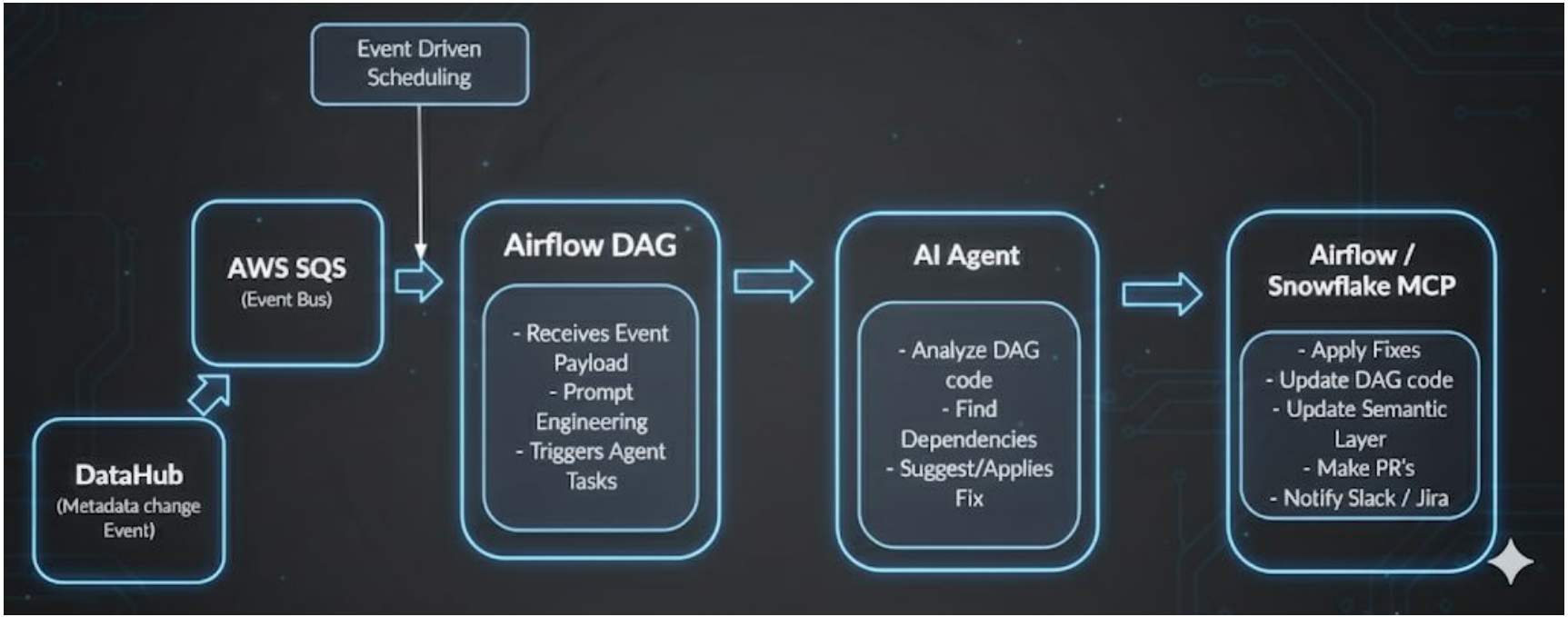
Airflow sits at the center of your tech stack — integrate it with real signals so it reacts instantly, not just on the clock.

From Agents to Actions

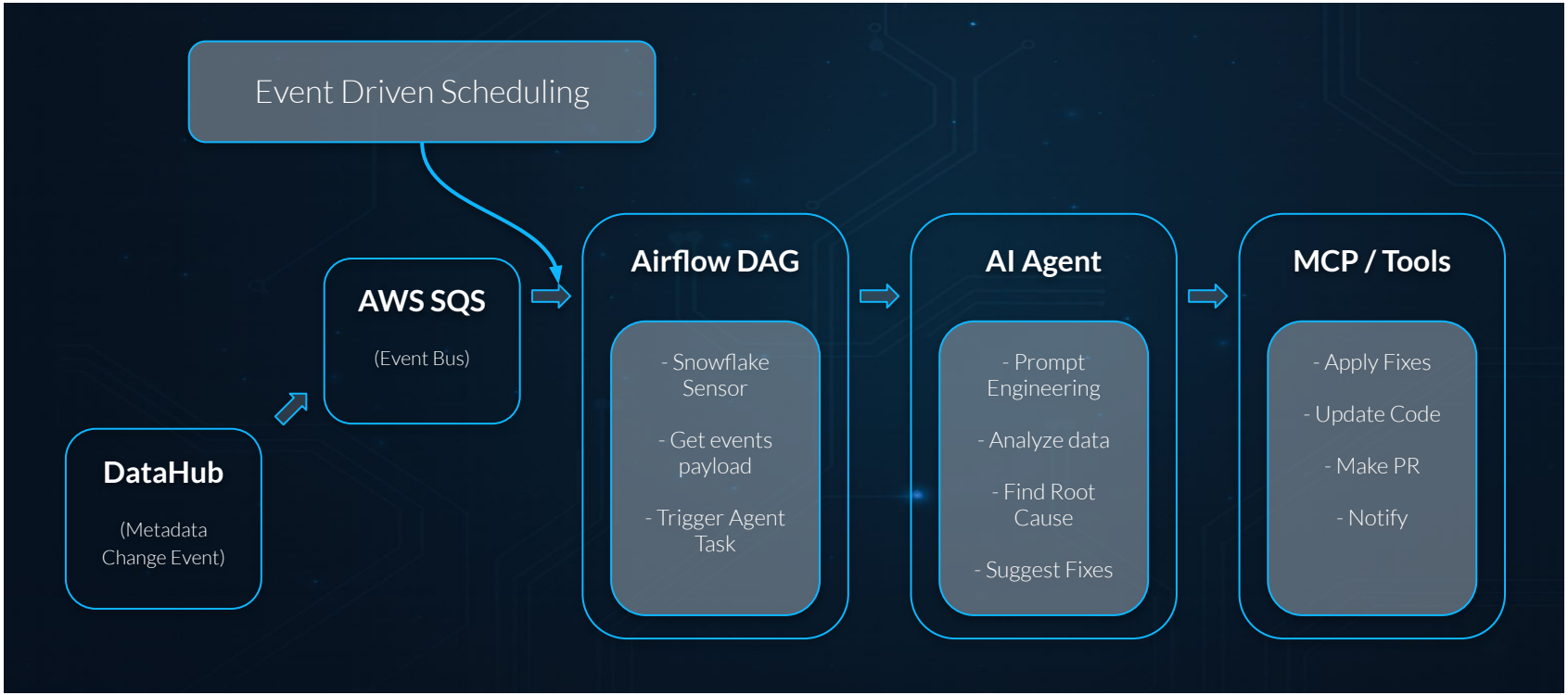
Airflow sits at the center of your tech stack — integrate it with real signals so it reacts instantly, not just on the clock.



Additional Applications



From Agents to Actions



Recap

- Even with Airflow, keeping pipelines healthy takes too much time.
- Use an AI agent with Airflow and MCP servers to spot failures early.
- We saw the agent analyze Airflow errors and automatically propose, implement, and deliver fixes through GitHub.
- Build with Airflow and Agents a solution tailored to your needs.

Qbiz

Questions?