



Ref: ODGS-CORE-V1.2.1

ODGS Protocol: Architecture Specifications

1. Integration Pattern

ODGS (v1.2.1) is a deterministic binding layer for high-risk industrial AI, available as a native package for Node.js and Python ecosystems.

- Package Access:** `npm install odgs` or `pip install odgs`
- Data Agnostic:** Native support for Parquet, Iceberg, and Delta Lake formats.
- Runtime Binding:** Injects as validation middleware between the Data Registry and Inference Engine.

2. Deterministic Semantic Hashing

Hashing Logic:

Utilizes **SHA-256** hashing of JSON business definitions to ensure absolute data-to-concept integrity.

- Unit dimensions (ISO alignment).
- Temporal and Drift constraints.
- EU AI Act Article 10 Audit Flags.

The "Hard Stop":

A 1-bit mismatch between the runtime hash and the Governance Registry triggers an **Immediate Execution Freeze**. This prevents "Silent Drift" where AI makes decisions on data no longer matching the compliance definition.

3. Governance as Code: Core CLI

| Command | Action | Outcome |
|---------------|------------|--|
| odgs init | Initialize | Creates standard protocol directory and 7 core JSON schemas. |
| odgs validate | Verify | Ensures definitions adhere to strict protocol constraints. |
| odgs build | Compile | Generates artifacts for dbt (MetricFlow), Power BI, and Tableau. |

4. Interoperability & Compliance

- **TNO Semantic Treehouse:** Engineered for native alignment with Federated Object Libraries (TNO Standards).
- **CIRPASS 2:** Validated against Digital Product Passport (DPP) data requirements.
- **EU AI Act:** Solves Article 10 (Data Quality) & Article 12 (Record Keeping) via cryptographic enforcement.

System Architecture

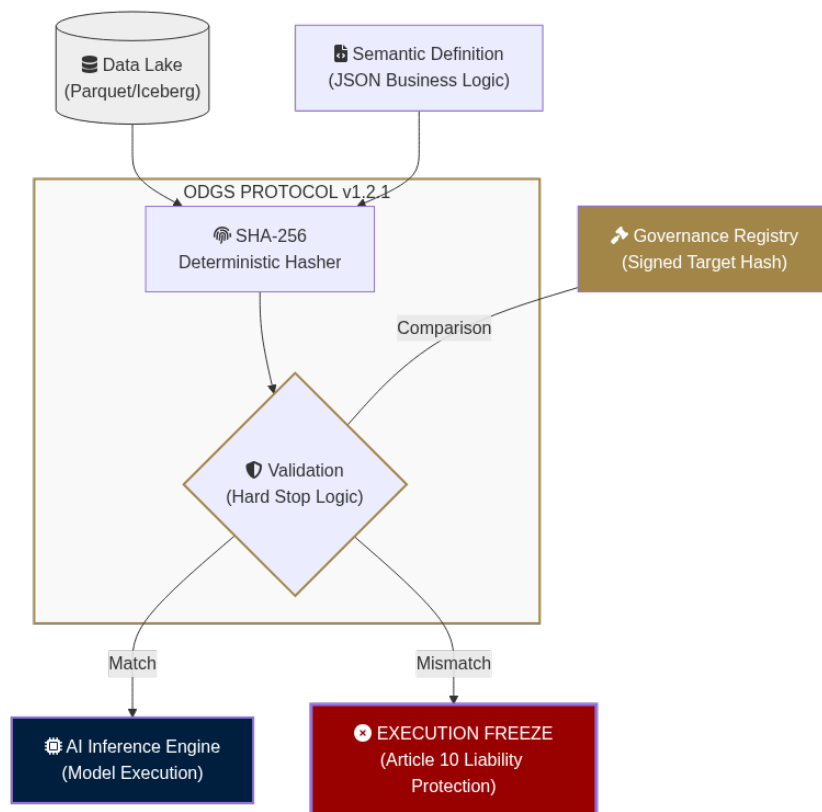


Figure 1: *

Figure 1: ODGS Binding Logic - Enforcing Deterministic Compliance

GitHub Repository
github.com/MetricProvenance/odgs-core

Reference Implementation
www.metricprovenance.com

Technical Inquiries: partner@metricprovenance.com