* 1. **Types of Metadata Architectures**

Metadata architecture defines **how metadata is collected, stored, accessed, and managed** across systems. The choice of architecture affects scalability, governance, performance, and user access.

**1. Centralized Metadata Architecture**

**Definition**

Metadata from various systems is **copied and stored in a single, centralized repository**.

**Characteristics**

* A central metadata catalog or repository
* Metadata is **extracted from source systems** and **loaded (ETL-style)** into the repository
* Commonly used in data governance and analytics programs

**Benefits**

* Single source of truth for metadata
* Simplifies governance and lineage tracking
* Easier to manage and search metadata

**Limitations**

* Requires continuous metadata synchronization
* Can become outdated if not refreshed regularly

**Use Case**

A bank pulls metadata from its CRM, ERP, and BI tools into a central repository (e.g., Collibra, Informatica EDC) for data stewards to manage definitions, owners, and lineage.

**2. Federated Metadata Architecture**

**Definition**

Metadata remains in its **original source systems**, and a **virtual metadata layer** provides unified access across sources.

**Characteristics**

* No physical central copy of metadata
* Uses **real-time connections** or APIs to access metadata on demand
* Often used in **large or decentralized enterprises**

**Benefits**

* Reduces data duplication
* Always accesses the most up-to-date metadata
* Easier to scale across diverse systems

**Limitations**

* Can be **slower** due to real-time metadata retrieval
* Complex to implement consistent governance

**Use Case**

A multinational corporation connects various business units' systems to a metadata portal that reads and displays metadata in real time without moving or storing it centrally.

**3. Distributed Metadata Architecture**

**Definition**

Metadata is stored in **multiple, physically separate repositories**, each typically managed independently.

**Characteristics**

* Repositories may be organized by function, geography, or department
* No single master catalog

**Benefits**

* Suitable for global or federated organizations with legal or regulatory boundaries
* Local control over metadata governance

**Limitations**

* Difficult to achieve enterprise-wide consistency
* Harder to search or perform global impact analysis

**Use Case**

Each country division of an insurance company maintains its own metadata repository due to data sovereignty rules. There’s no enterprise-wide metadata view.

**4. Hybrid Metadata Architecture**

**Definition**

A **blend of centralized and distributed approaches**. Some metadata is stored centrally, while other elements remain in local or external systems.

**Characteristics**

* Combines **flexibility of distributed** with **consistency of centralized**
* Often includes **integration with federated tools**

**Benefits**

* Balances scalability and governance
* Supports both local autonomy and central oversight

**Limitations**

* Complexity in integration and metadata consistency
* Requires robust metadata governance processes

**Use Case**

An energy company centralizes critical business terms and definitions in one catalog but allows individual business units to manage technical metadata locally.

**5. Network Metadata Architecture (Less Common Term)**

**Definition**

This is **not a formal architecture type** recognized in DAMA or ISO, but the term may refer to a **web of interconnected metadata systems** that share metadata through APIs, ontologies, or linked data standards.

**Use Case**

A group of public agencies uses a shared metadata exchange format (e.g., W3C DCAT) to interoperate across open data portals.

**Note: This is closer to a semantic web or knowledge graph architecture than a traditional metadata system.**

**Conclusion**

Choosing the right metadata architecture depends on:

* Your **organization’s size and structure**
* The **systems and tools** in use
* **Governance maturity**
* **Regulatory and operational needs**

Each architecture has trade-offs between **central control**, **scalability**, and **real-time access**.