Evaluating Metadata Architectures

Introduction

The IT world has practiced metadata management in one form or another for more than twenty years. IT has developed three models for metadata management:

- Centralized metadata management
- Distributed metadata management
- Hybrid centralized/distributed metadata management

If you understand the strengths and weaknesses of these models, you can better judge how you can use them to implement an effective metadata management strategy. This paper will provide an outline of the advantages and disadvantages inherent in each architectural approach.
Centralized Architecture

The centralized architecture, also known as a passive repository, is the oldest approach to metadata management. Legacy metadata repository solutions often use this approach, which uses a centralized repository to which you copy metadata from other data sources (Figure 1). The earliest centralized metadata repository solutions were homegrown solutions in flat files that captured information within text documents and spreadsheets. These repositories are considered to be the ‘first-generation’ metadata repositories. Over the past two decades, the amount of information these solutions stored has grown exponentially and these solutions have evolved to be more sophisticated. The metadata has moved out of flat-file based systems and into relational databases, the ‘second-generation’ repository which in-turn became the first commercial metadata management products.

Figure 1 - Centralized Architecture: The repository copies data from other sources.

Strengths of Centralized Architecture

A centralized architecture approach has the following strengths:

**Efficient access to information**

Users only need to access one area to find information by navigating the repository, performing searches, or running reports.

**Independence from integrated systems**

When a centralized repository integrates with third-party products, it copies data from those products into the metadata repository. Accessing the metadata is then independent of access to the original system, as the duplication of the data in the repository frees the system from any required access to the original metadata.

**Ability to capture additional metadata**
Since a separate database stores the metadata, you can capture additional metadata. This additional metadata may be customized extensions to the source metadata, or it may be entirely separate information not previously created by your organization. For example, you may have a system that tracks workflows extensively and provides metadata about the flows. You may want to add information such as process definitions and metrics that the source system doesn't provide. Another possibility is to associate contact information with specific objects in the target system, where the source system doesn't do contact management.

**Enhanced performance**

A single database stores all metadata, even if it comes from a third-party product, so when a user accesses information, the system retrieves it from only one location, reducing query response time.

### Disadvantages of Centralized Architecture

A centralized architecture approach has the following weaknesses:

**Cumbersome implementation and maintenance**

If a third-party system stores the only copy of the metadata, the centralized approach must copy it into the centralized database. This can be a time-consuming procedure that requires developers to create and maintain the data transfer routines, and to execute them regularly. Some vendors may offer pre-built software solutions for data transfer, but often you'll need to customize these solutions to work with the specifics of today's highly complex and diverse systems. Keeping the metadata synchronized requires a massive effort on the part of IT. One result is to limit the amount and detail of the captured metadata. Another result is the highly manual intervention in the data loading process. The centralized copy will always be out-of-date compared to the original metadata, especially in today's fast-changing enterprise systems. The need to create and maintain the complex copying routines dramatically reduces IT productivity, while the need to wait for data to become current dramatically reduces end user productivity. Both such reductions result in low-ROI systems that often become shelfware.

**Reduced data quality**

Data duplication in the centralized repository also compromises the quality of the data. End users often do not know the currency or validity of the data they are accessing. Additionally, the redundancy in data may result in significant divergence between the original system and its representation in the repository, despite the best efforts of IT programmers. User confidence in the system then diminishes significantly, often resulting in lower adoption by end users.
Distributed Architecture

A distributed architecture, also referred to as an active repository, was developed in response to the weaknesses of centralized architecture. This approach avoids the tedious process of maintaining copies of the source metadata within the metadata repository. Rather, the distributed system accesses metadata from third-party repositories in real time. As with the centralized architecture, this approach facilitates a single access point to the information. The distributed metadata solutions are considered to be ‘third generation’ repositories (Figure 2).

*Figure 2: Distributed Architecture – The metadata engine accesses metadata in real time.*

Strengths of Distributed Architecture

A distributed architecture has the following benefits:

**Efficient access to information**
As with centralized architecture, distributed architectures access information through a single system, making access to metadata easy. The focus on distribution can make it much easier to integrate disparate systems throughout the enterprise.

**Increased ROI through productivity**
The distributed architecture doesn’t copy metadata into a central repository and does not require any customized transfer routines and periodic synchronization. The elimination of the need to synchronize data eliminates any need for maintenance or manual intervention. This makes for much higher productivity for both IT and the end user.

**Superior data quality**
Since metadata is accessed in real time from source systems, end users always see the most current information available. This approach guarantees that the end user is looking at the correct data, as it currently exists in the source systems.
The analysis tools of the system, such as searching and impact analysis tools, always work with current data. This greatly improves confidence of end users and increases system adoption.

**Disadvantages of Distributed Architecture**

A distributed architecture approach has the following disadvantages:

**Dependent on integrated systems**

Distributed systems access metadata in real time. The availability of the source system thus becomes a limit on the availability of metadata through the distributed system. Since a production metadata solution typically accesses production third party repositories, the issue of availability is important but is seldom an issue, as production systems typically have availability goals of greater than 99.9%.

**Inability to capture additional metadata**

A distributed approach can only access metadata from third-party repositories in real time so it cannot capture additional metadata, as it does not maintain a centralized database. This inability to customize metadata often means that IT must implement custom requirements through other tools, leading to lower IT productivity and decreasing user acceptance of the system.

**Hybrid Architecture**

A hybrid approach leverages the strengths and mitigates the weaknesses of both distributed and centralized architectures. The hybrid approach facilitates metadata access in real time from third-party sources and provides the ability to capture additional metadata attributes not existing in the source repositories. Additionally, it lets users create original data within the repository.
Figure 3: Hybrid Architecture – The repository accesses metadata in real time and facilitates the capture and storage of additional metadata and original information.

**Strengths of Hybrid Architecture**

A hybrid architecture has the following benefits:

**Efficient access to information**

As with the other approaches, hybrid systems offer a single point of access to metadata.

**Increased ROI through productivity**

The hybrid approach combines the ability to get metadata in real time with the ability to provide customized solutions using additional metadata linked to the original metadata. This reduces the need for IT customization work and gives end users the ability to maintain their own metadata.

**Ability to capture additional metadata**

The metadata repository can store additional information, so you can capture new metadata. This additional metadata can be customized extensions to the source metadata, or it can be entirely separate information. The hybrid system can also implement metadata versioning, enabling IT and end users to track changes to their systems and metadata.

**Superior data quality**

Since end users see their metadata in real time, they always see the most current information. The ability to add extended metadata and completely new metadata can add tremendous value to the end user’s productivity. This greatly improves end-user confidence and increases system acceptance.
Enhanced performance

By providing both real-time metadata and custom metadata together, the system improves the overall system response time by integrating disparate applications into a single system. By distributing data access across multiple back-end systems, the hybrid system can perform as well as a centralized database while providing much of the advantages of the centralized approach.

Disadvantages of Hybrid Architecture

Accessing source metadata in real time presents one problem: if the source system is not available, neither is the metadata. Since a production metadata solution typically accesses production third-party repositories, the issue of availability is important but is seldom an issue, as production systems typically have availability goals greater than 99.9%. Hybrid systems can mitigate this weakness through storage of baseline metadata; the end user can see the baseline as of a certain time along with the extended and new metadata even while the source system is down.

Conclusion

Most of the time, a hybrid approach is the optimal solution due to its flexibility in leveraging the strengths of both the distributed and centralized architectures. Even if your organization’s requirements fall within the scope of a distributed or centralized architecture, in time the necessity to accommodate flexibility may change. Therefore, planning ahead by adopting the hybrid approach in lieu of limiting your system to a centralized or distributed approach may be the most scalable deployment strategy to meet your organization’s short and long term growth expectations.

However, if your organization only anticipates the need to access metadata within existing systems and does not expect to have the need to capture additional information, then a distributed architecture may be the most appropriate solution.

A centralized approach is only appropriate when there is a technical or political roadblock preventing the adoption of a hybrid or distributed system. For instance, a technical roadblock exists if it is not possible to access metadata from a source system through programming interfaces or database queries. This is often the case for legacy systems from the mainframe days as well as for in-house developed systems. A political roadblock may often arise when administrators of mission critical systems worry that integrating their system with the metadata solution may interfere with system uptime and integrity. This is typical in systems with large groups of end-users accessing complex data sets (ERP, CRM, payroll, etc.).
Organizations should be mindful of long-term IT strategies when examining distributed or centralized solutions. Business requirements are always evolving, often resulting in changes to their supporting technical infrastructures. Keeping your metadata strategy flexible can help in managing these changes. Figure 4 outlines the requirements of metadata solutions and highlights how each of the three architectures fulfills them.

**Figure 4: Metadata Architectures & Requirements Matrix**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Centralized Architecture</th>
<th>Distributed Architecture</th>
<th>Hybrid Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expedited Implementation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>One Stop Shopping for Metadata</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Independent from Integrated Systems</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Independent from Integrated Systems</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Performance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>No Data Redundancy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>High Data Quality</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>No Synchronization Issues</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Low Maintenance Cost</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cost of Ownership</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Return on Investment ROI</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

*Dependent on implementation. Metadata may be copied into the repository to make it independent from integrated systems.

**For Additional Information Contact:**
Data Advantage Group, Inc.
604 Mission Street
San Francisco, California 94105

Tel: +1 (415) 947-0400
Fax: +1 (415) 947-0401
Email: info@dag.com
Web: [www.dag.com](http://www.dag.com)