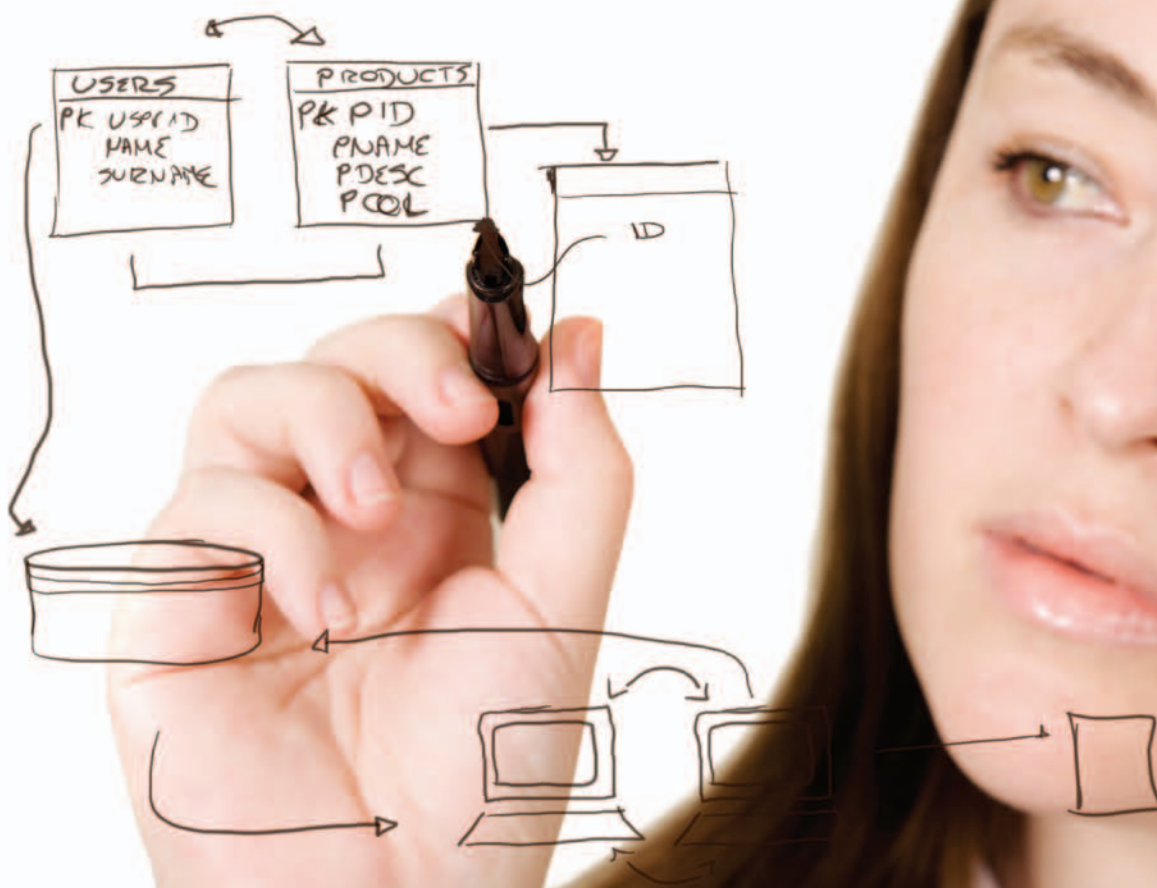


# The Importance of Architecture in ERM Software Selection



The underlying differences in records software have implications for an enterprise's strategy, implementation, and risk challenges.

**Julie Gable, CRM, CDIA, LIT, and Baron Gemmer, MIT**

When considering electronic records management (ERM) software products, records managers tend to focus on their functionality. However, all ERM products perform approximately the same functions. Their true differences lie deep within their design – in their architecture. Understanding those differences can inform ERM software selection decisions, but – more importantly – it can affect an organization’s overall strategy for managing electronic records because a product’s architecture will help determine what is possible, at what cost, and in what timeframe.

### ERM Functionality

All ERM software products provide basic functionality that allows organizations to

- Declare records
- Classify records and attach retention periods
- Store records
- Search and retrieve
- Enforce retention and disposition
- Place and process tax, legal, or other holds
- Maintain retention rules
- Produce reports

To perform these functions, the ERM software is concerned with four important components:

1. *The electronic records themselves* – which must be kept secure and protected against unauthorized access so their integrity and authenticity are preserved
2. *Metadata* – which manages information about the records that is crucial for their control. For example, metadata would include attributes that identify a record, as well as a date from which to calculate retention.
3. *A classification scheme* – which provides categories for keeping similar records together. Often called a file

## At the Core This article

- ▶ Defines the concept of software architecture
- ▶ Discusses prevalent electronic records management (ERM) architecture models
- ▶ Provides a framework for analyzing ERM software architecture models

plan, the classification scheme may also assist in limiting access to certain types of records, and it may provide a mechanism for placing legal or other holds on certain classes of records.

4. *Records retention rules* – which govern how records are managed. Retention rules are the expression of the organization’s policy regarding its records.

How each ERM software product handles these components, as well as how it interfaces with other software applications, form the basis for its architecture.

### Architecture Defined

Simply put, an *architecture* is the conceptual description of how a system works. At the basic level, architecture describes the components the system comprises. Other levels describe what these components do and how they interact.

In the ERM world, architecture models must consider how the ERM software product interacts with various electronic source applications that contain the records to be managed. Source applications can include office applications that produce electronic documents, document or content management systems that may have their own metadata and document stores, and enterprise databases that manage structured data in rows and columns. It is the combination of ERM software with source applications that delivers solutions for managing electronic records.

An ERM software product’s architecture has implications in many organization areas. From an information technology perspective, architecture can affect overall system cost, system scalability, and the resources and time required for its installation, integration, and maintenance. Records managers and business users can be affected by architecture in areas such as system speed and ease of use. When considering ERM software products, it is important to be able to differentiate among architecture models and to recognize each model’s implications for what will be required to manage electronic records effectively.

### ERM Architecture Models

ERM architecture models are defined by whether the components – that is, the records, metadata, classification scheme, or retention rules – reside in the ERM product, in the source applications, or in both.

There are basically four architecture models that describe current ERM software products. Each model has been named to reflect the way it approaches electronic records management elements. Figures 1 through 4 provide graphic representations as an aid to understanding differences among models, and figure 5 is a summary of the differences among models.

Each architecture model has trade-offs for records managers, and these are discussed within the model’s description. More importantly, however, each model has implications for an enterprise’s business strategy, information technology, and risk response abilities, and these are discussed in succeeding sections.

1. *Single Repository Model* – In this model (see figure 1), all components of records management are kept within one system – the ERM software. Records and metadata move out of source applications to the ERM software’s repository and database, respectively. This is an approach favored by many early ERM software products, most likely in response to the DoD 5015.2 requirement (based on

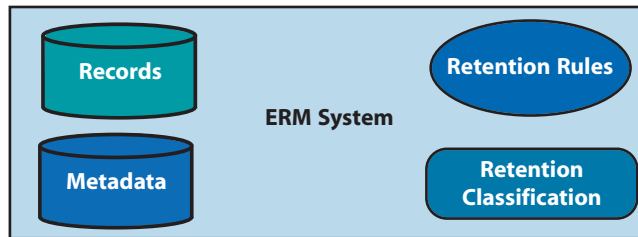
archives theory) that records are more secure if they are retained in a protected repository.

The trade-off is that moving the records and metadata to the ERM software may make them inaccessible and unsearchable from within the source application, a situation that is not popular with end-users. Document management systems, for example, generally have better search capabilities than ERM products. Also, moving documents to the ERM system may actually compromise security because many document management systems have better security capabilities than do ERM products.

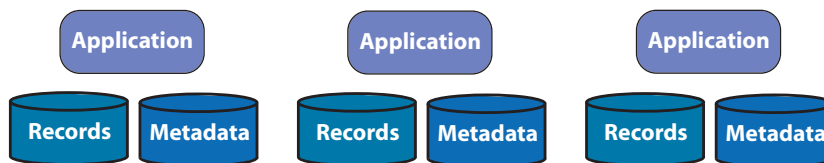
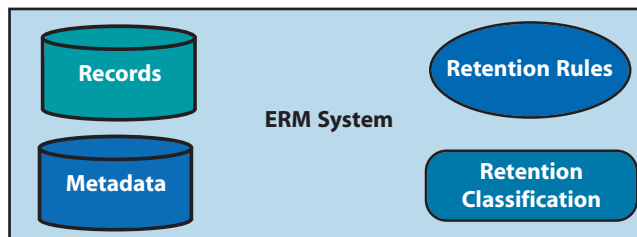
**2. Replicated Model** – In a replicated approach (see figure 2), classification scheme and retention rules are maintained within the ERM software, but records and metadata are essentially copied and kept in both the ERM software and the source application. This creates a burden on the organization to keep records and metadata synchronized in both places to ensure that both systems contain exactly the same information at all times. This can be difficult, particularly where versioning of electronic documents occurs in document management systems. The implication is that the ERM system’s records repository could contain a different version of the record than resides in the document management system’s repository, leading to questions about which repository contains the official record.

**3. Catalog Model** – In a catalog model (see figure 3), records remain in the source application repository, but metadata about the records is kept in both the source application and the ERM software system. While this is something of an improvement over the replicated model, the difficulty for the organization is in ensuring that metadata is always synchronized between the two applications. Changes, additions, or corrections made to metadata in the source application would not necessarily be reflected in the ERM system unless, and until, some synchronization routine is run on a regular basis. Unsynchronized metadata can

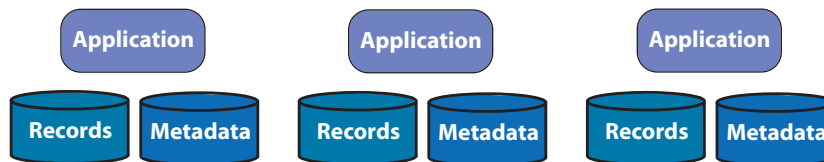
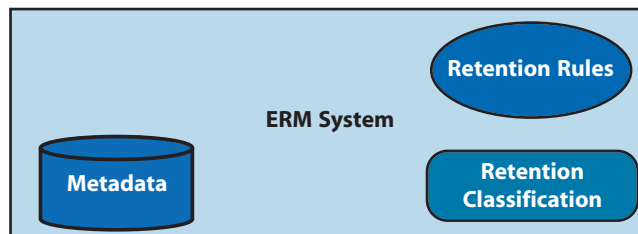
**Figure 1. Single Repository Model**



**Figure 2. Replicated Model**



**Figure 3. Catalog Model**



mean that searches conducted in both the ERM system and in the source applications would not produce the same

results.

**4. Delegated Model** – In a delegated model, (see figure 4), sometimes referred

to as a federated approach, records and metadata remain in their source application, but the ERM software acts as an engine to associate a classification scheme and retention rules with the records. While this approach prevents the burdens associated with synchronizing records or metadata in two different systems, it has one significant implementation requirement: in order to work, the native or source application must be able to accept an application programming interface so the ERM system can overlay its classification scheme and retention rules on the source application. Older, mainframe-based systems may not be able to do this.

A summary of architecture models appears as figure 5.

### ERM Architecture Models and the Enterprise

Most software selection processes involve considering a particular department's functional requirements along with some standard technical requirements and considerations. ERM systems, on the other hand, require organizations to take an enterprise approach to be successful. Because electronic records governance requires an understanding of business strategy, information technology, legal, and compliance needs, ERM product architectures affect how – or if – solutions will be possible for a wide range of issues including:

- Business Strategy Issues
  - Outsourced systems
  - Multiple model coexistence
- IT Issues
  - Size/scalability
  - Application integration
  - ERM system programming
  - Record/metadata synchronization
  - Application programming
- Risk Response Issues
  - Search ease
  - Search speed
  - Record retrieval speed
  - Hold processing ease

Figure 4. Delegated Model

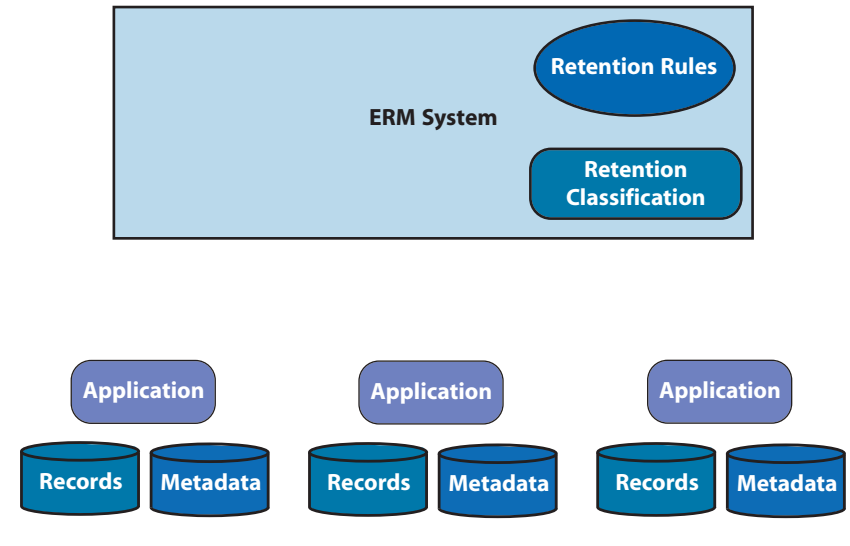


Figure 5. ERM Product Comparison

	ERM System	Source Application
<b>Single Repository</b>	Records, Metadata, Retention Rules, Retention Classification	
<b>Replicated</b>	Records, Metadata, Retention Rules, Retention Classification	Records, Metadata
<b>Catalog</b>	Metadata, Retention Rules, Retention Classification	Records, Metadata
<b>Delegated</b>	Retention Rules, Retention Classification	Records, Metadata

*Business Strategy Issues* – Although it may not be obvious at first, there are global business strategies that present issues for ERM product installations. One is system outsourcing. While many companies outsource systems based upon return on investment expectations, they are not absolved from responsibility for the outsourced records. Can the records or metadata in outsourced systems be replicated if an ERM product architecture requires it? Beyond that, can the systems communicate to exchange classifications, retention rules, holds, and retention trigger events?

Other business realities that may need to be considered are currently installed ERM systems and preferred strategies in moving toward a global system. If a company needs to integrate multiple existing systems, or if it wants to install departmental systems that will

eventually coalesce into a more global system, it is important to determine which ERM architectures can most easily coexist with each other.

*IT Issues* – There are many IT issues that need to be considered for any software installation. But the scope and pervasiveness of an ERM software installation increases the effects of these issues by orders of magnitude. Any replication, whether for both records and metadata or just metadata, can present scalability issues for even a modest-sized company. In essence, replicating records and metadata requires storage capacity requirements to double. Replication also requires a carefully analyzed and executed synchronization strategy. Besides the time and resources required to program both sides of a synchronization interface, additional considerations need to be made for how often the

**Figure 6. ERM Architecture and Enterprise Issues**

		Single Repository	Replication	Catalog	Delegated
<b>Business Strategy Issues</b>	Outsourced System	Very Difficult	Very Difficult	Challenging	Moderate
	Multiple Model Co-existence	Very Difficult	Very Difficult	Challenging	Easy
<b>IT Issues</b>	Size/ Scalability	Very Difficult	Very Difficult	Challenging	Challenging
	Application Integration	Very Difficult	Challenging	Challenging	Moderate
	ERM System Programming	Easy	Challenging	Challenging	Challenging
	Synchronization	Easy	Very Difficult	Very Difficult	Easy
	Application Programming	Easy	Very Difficult	Very Difficult	Challenging
<b>Risk Response Issues</b>	Search Ease	Easy	Easy	Moderate	Challenging
	Search Speed	Easy	Easy	Challenging	Challenging
	Record Retrieval Speed	Easy	Easy	Challenging	Challenging
	Hold Processing Ease	Easy	Easy	Easy	Challenging

Legend: Very Difficult Challenging Moderate Easy

synchronizations will be performed. Even if no synchronizations are required, there may be interfaces required to identify classifications, retention rules, holds, and event triggers.

**Risk Response Issues** – The replication of records and metadata also has implications in handling risk responses, which include events like lawsuits and audits. In these cases, the replication is important because consolidated metadata from the source applications typically makes for faster and easier searches. Similarly, consolidated records usually increase the speed of record retrieval. Hold processing, like searches, is dependent primarily upon metadata and is improved by consolidated metadata.

**Selection Considerations**

As figure 6 shows, each architecture model has trade-offs in regard to business strategy, IT, and risk response issues. Looking at the chart from the perspective of the three issue sets, the single repository architecture presents significant – if not insurmountable – challenges for business strategy issues, is mixed on IT issues, and is overwhelmingly positive regarding risk response issues.

As the architectures move from left to right in the chart, most issues transition from one end of the “Very Difficult – Easy” spectrum to the other. A casual observer may infer that maximizing the easy issues

is the best selection method for choosing an ERM product. But this is not valid because it ignores important issues. Although companies have different needs and goals for ERM software implementation, most organizations will face most or all of the issues listed in figure 6. Therefore, the most effective selection approach is usually to minimize the very difficult issues, not to maximize the easy ones. This would lead most organizations to choose the delegated model.

The best selection methodology would include an analysis and prioritization of all ERM software-related issues. With that completed, the selection process could determine where the most important issues are more easily addressed relative to each product’s architecture. This could also be accom-

plished through a weighted scoring system, especially if some of the areas are of much lower importance to the organization.

**Final Selection**

Every product architecture model presents challenges, so it is critical for organizations to be aware of what the challenges are and the degree to which they are acceptable before the final selection is made. Factors that influence the acceptability of such challenges are an organization’s size, the industry it operates in, its IT resources and their expertise, and the enterprise’s risk response needs. Identifying the issues up front acknowledges the needs of all enterprise records stakeholders and sets realistic expectations going forward. ■

*Baron Gemmer, MIT, is president of Innovative Corporate Solutions. He has more than 20 years experience in information systems development, specializing in architecture, systems integration, and project leadership. He has extensive experience in developing enterprise-wide document management and insurance systems and has led project teams in successfully implementing them. Gemmer serves on AIIM’s International Board and on its Executive Committee. He holds BS degrees in Electrical Engineering/Computer Science and Mathematics from Rose-Hulman Institute of Technology. He may be contacted at baron@4ics.com.*

*Julie Gable, CRM, CDIA, LIT, FAI, is the president of Gable Consulting LLC, an independent firm that specializes in electronic records management. She is the Associate Executive Editor of The Information Management Journal and the author of ARMA’s monthly online NewsWire. Gable holds an MBA in Finance from St. Joseph’s University and a BS in Management from Drexel University. She may be contacted at juliegable@verizon.net*