Architecture Considerations in ERM Software Selection

Session REC08

Julie Gable

Baron Gemmer

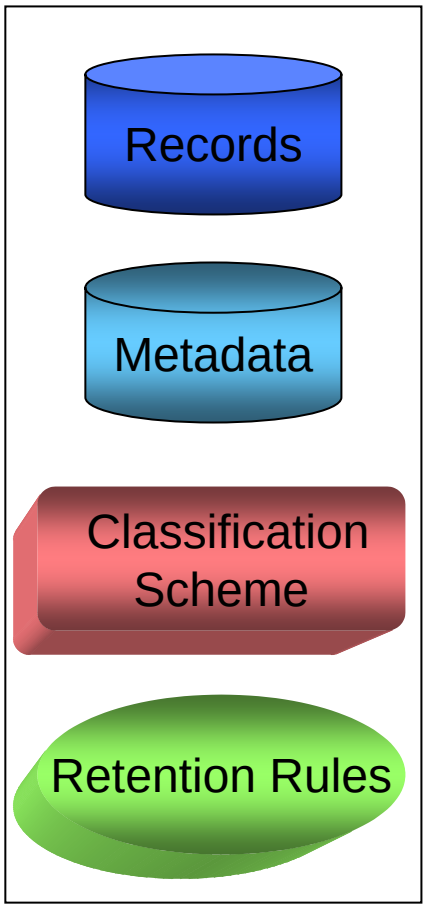
ERM Software Functionality

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

Why Architecture Matters

- Differences in ERM product architecture
  - Affect overall electronic records management strategy
  - Help determine what is possible, at what cost, in what timeframe

E-records Management Components



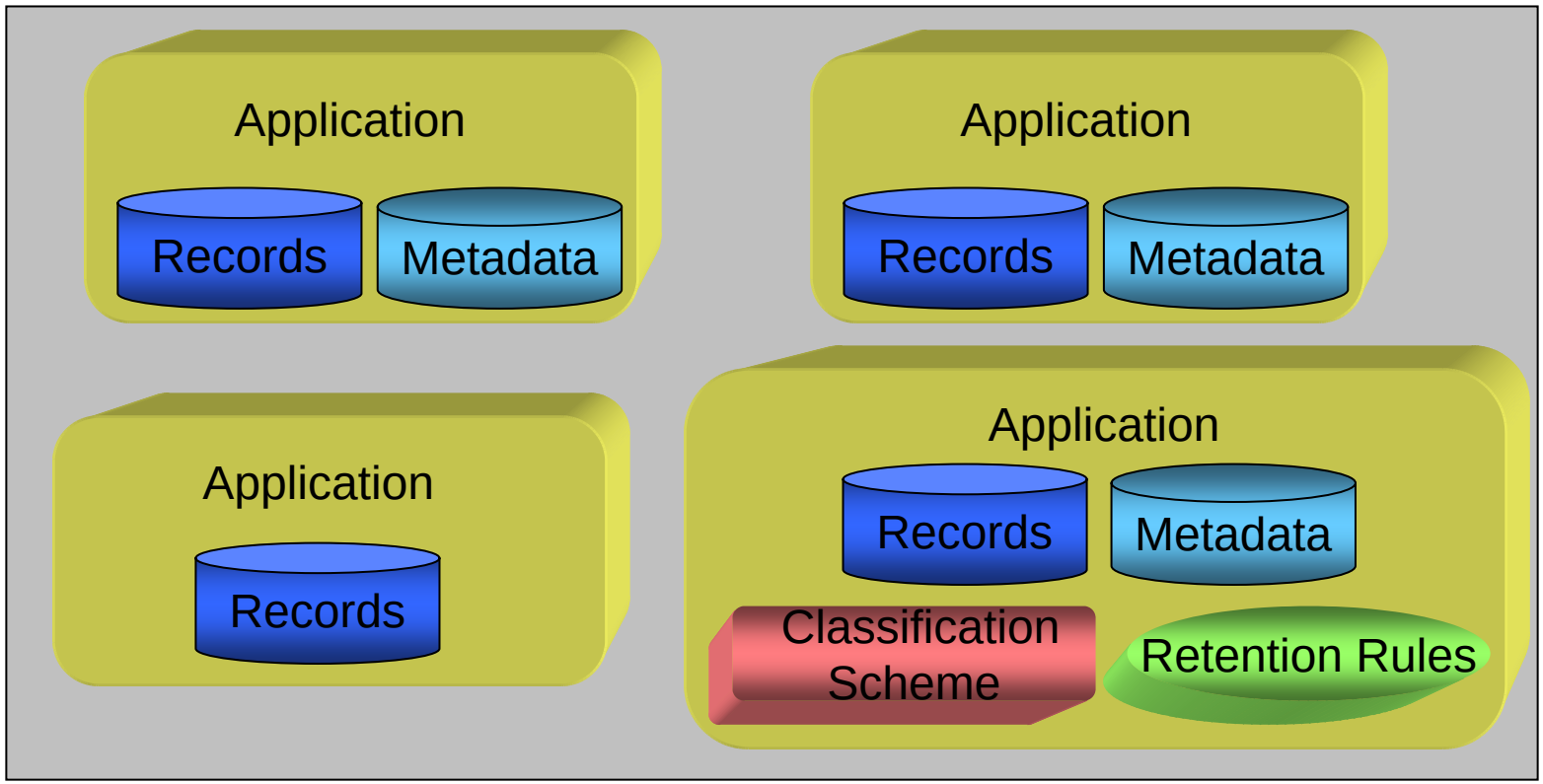
- Info created, received & maintained to conduct business
- Info about records needed for their control; attributes
- Categories for grouping similar records; a.k.a. file plan
- Organization's policy for how records are managed

ERM Architecture Models

- Defined by where the components reside:
  - ERM product
  - Source application(s)
  - Both

In Most Companies…

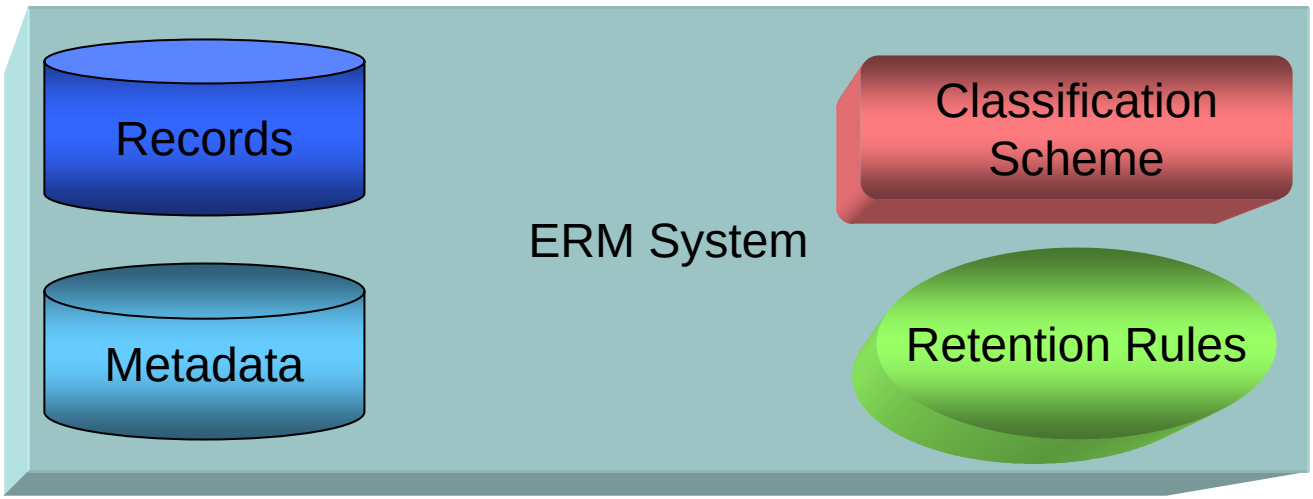
Exclusive of paper records management



ERM Architecture Models

- Single Repository
- Replicated
- Catalog
- Delegated

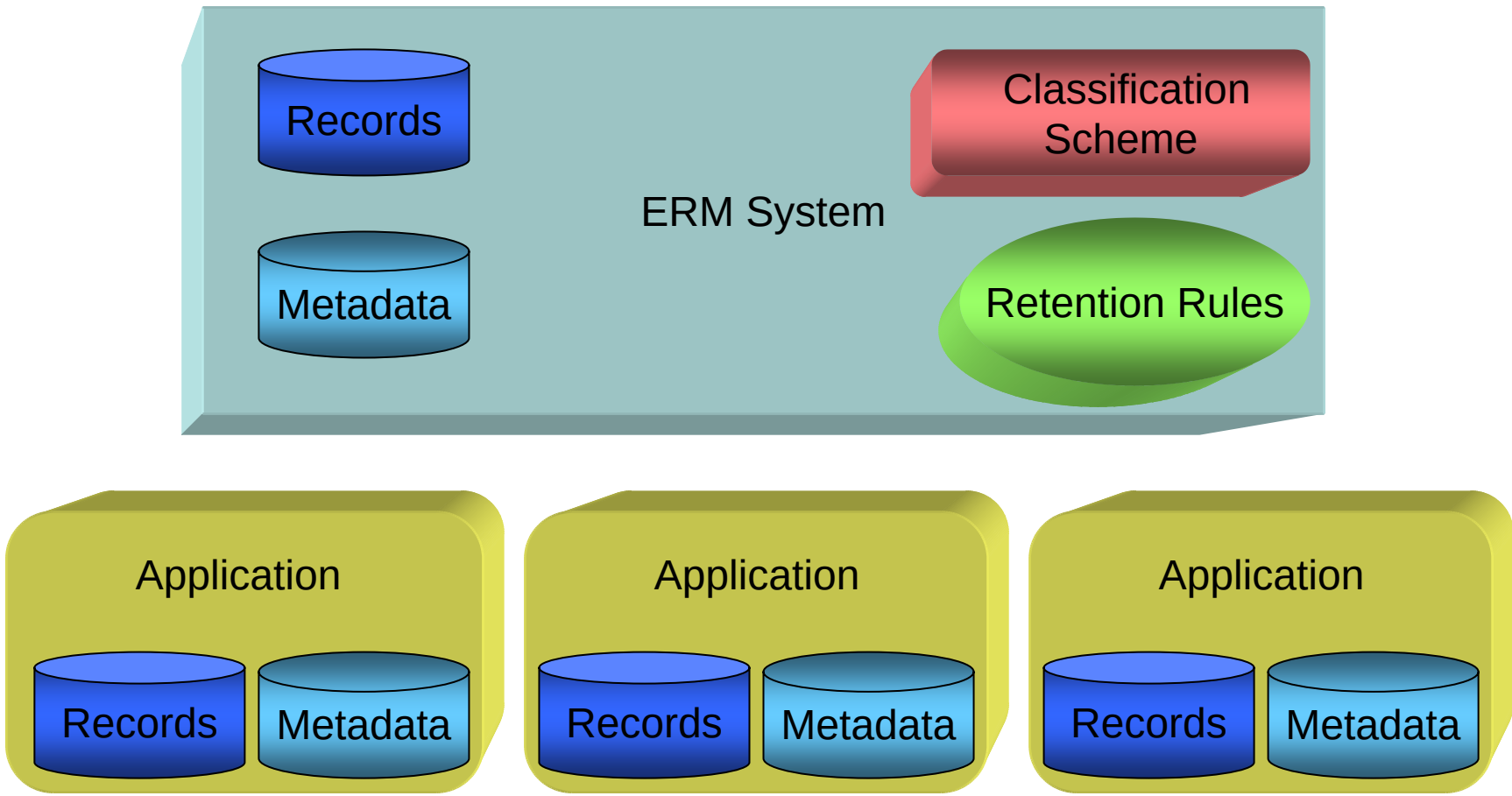
Single Repository Model



Trade-offs: Single Repository

- Easy to manage
  - All e-records components in one system
- Records and metadata move out of source application
  - May be inaccessible and unsearchable from within source app
  - May be less secure in ERM repository than in source app or document management system

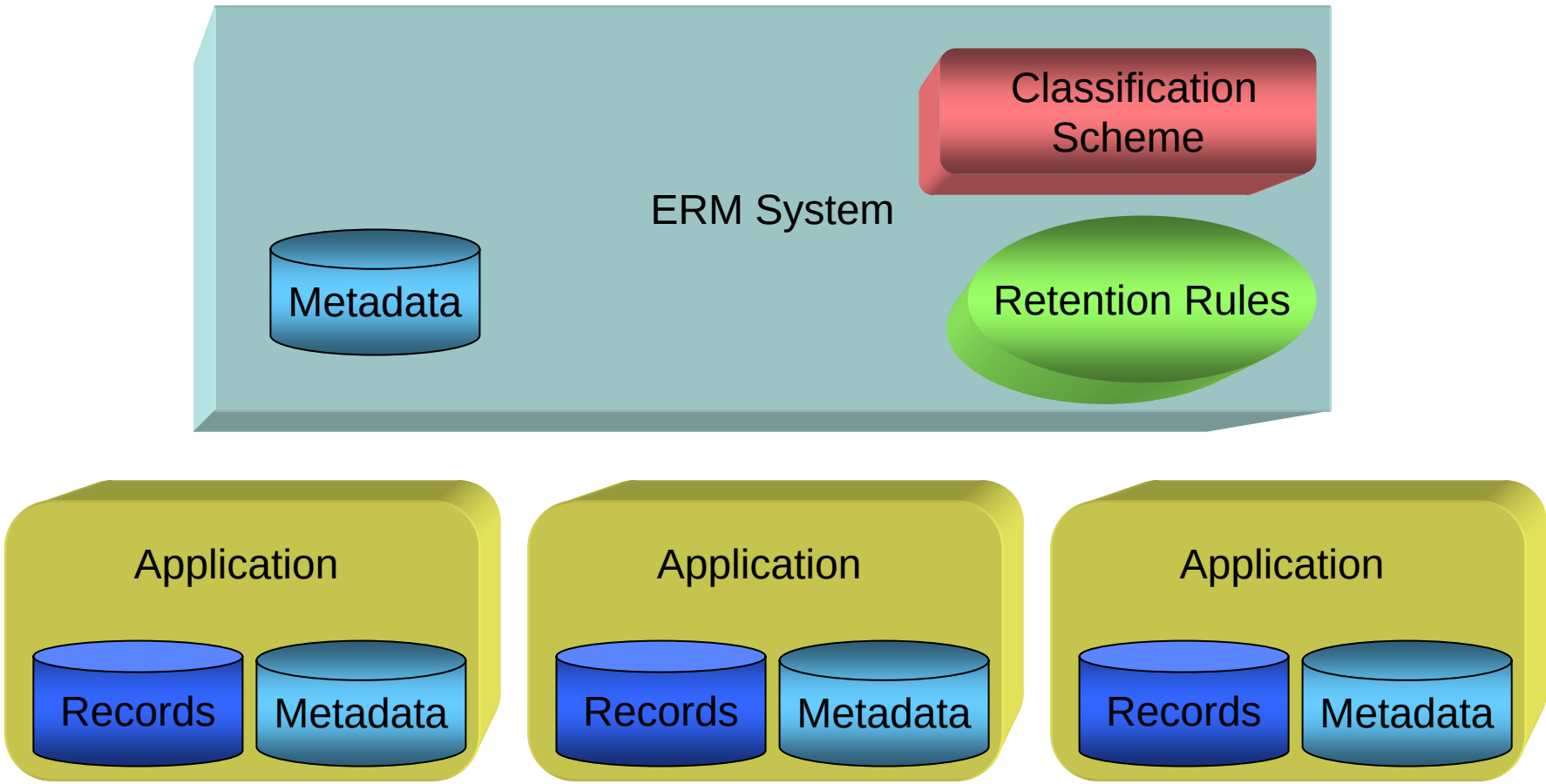
Replicated Model



Trade-offs: Replicated Model

- Records and metadata remain searchable and accessible in source application
- Must keep records and metadata synchronized in ERM system & source app
  - Versioning of records is problematic
    - Does record in document management system match record in ERM system?
    - Which one is official record?

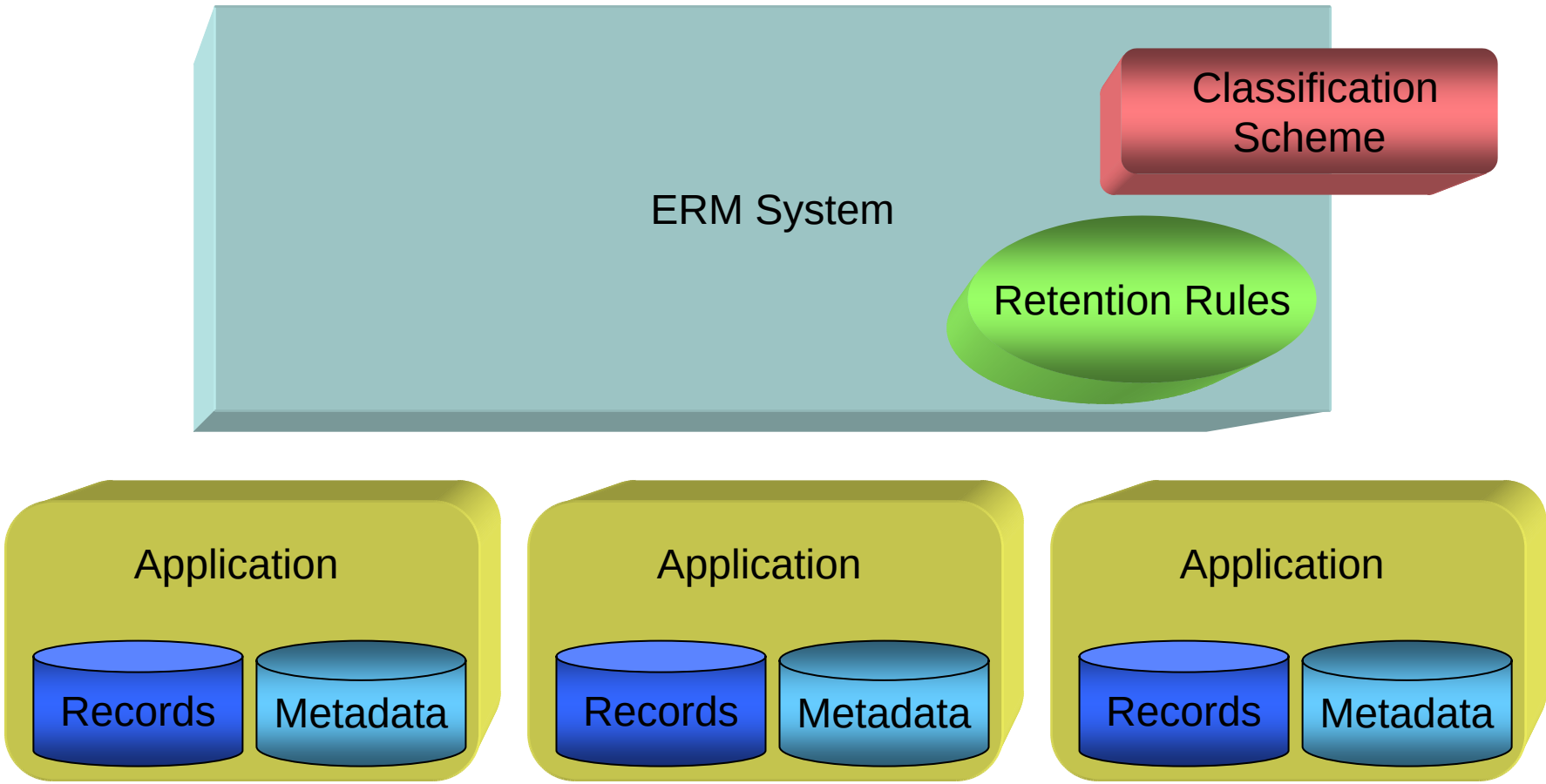
Catalog Model



Tradeoffs: Catalog Model

- Overcomes versioning issue for records
- Must synchronize metadata in ERM system and source application
  - Metadata changes in source application not reflected in ERM system until synchronization routine runs
- Metadata searches in ERM system and in source application might not produce same result

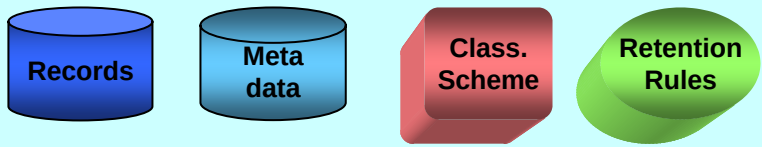
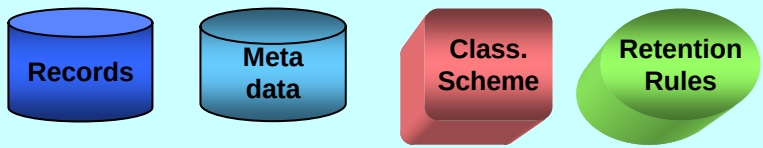
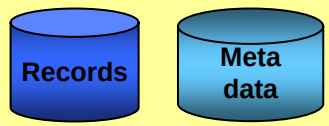
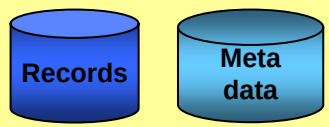
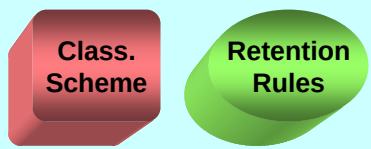
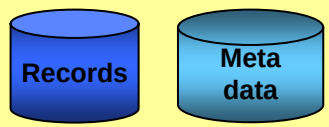
Delegated Model



Tradeoffs: Delegated Model

- Records & metadata remain in source application
- No synchronization required
- Implementation issue:
  - Source app must be able to accept API interface to ERM system

ERM Software - Architecture Summary

Model	ERM System	Source Application
Single Repository		
Replicated		
Catalogue		
Delegated		

Architecture and the Enterprise

- Business Strategy Issues
- IT Issues
- Risk Response Issues

Business Strategy Issues

- Outsourcing
- Multiple model co-existence

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

ERM Architecture & Enterprise Issues

		Single Repository	Replicated	Catalog	Delegated
Business Strategy Issues	Outsourced Systems				
	Multiple Model Co-existence				
IT Issues	Size / Scalability				
	Application Integration				
	ERM System Programming				
	Synchronization				
	Application Programming				
Risk Response Issues	Search Ease				
	Search Speed				
	Record Retrieval Speed				
	Hold Processing Ease				

Legend:

Very Difficult | Challenging | Moderate | Easy

Architecture Considerations in ERM Selection

- Does product have all needed functionality?
  - Paper records capabilities?
  - Security capabilities?
- Analyze and prioritize enterprise issues
- Determine which architecture best fits enterprise issues
- Can organization accept the challenges associated with that architecture?

Thank You

Gable Consulting
Document-Based Information Management

juliegable@verizon.net
www.gableconsulting.com

baron@4ics.com
www.4ics.com

ICS