ORACLE®

This document is for informational purposes. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Integrating DICOM Objects with a Database Management System

Dongbai Guo
db.guo@oracle.com
Oracle

Agenda

- Background
- Motivation, challenges and design goal
- Our solution: architecture and key concepts
- Example: metadata management
- Summary

Background

- Part of Oracle's *inter*Media feature

  A database application development framework for storage, manipulation, management, and indexing of multimedia data in Oracle database

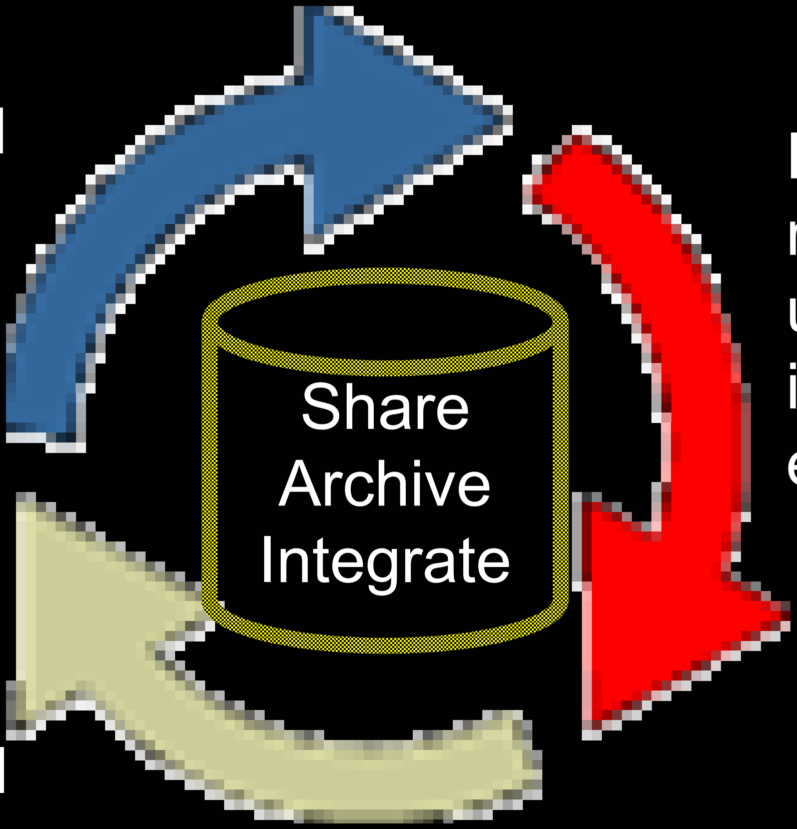
- Provides relational and object interfaces to support:
  - Web publishing
  - Streaming media
  - JDeveloper and Oracle Portal integration

Business Motivation

- Growth of healthcare IT market
- Large increase in digital imaging
- National archives for research and public health
- Healthcare information integration
  - image integrated EHR (electronic healthcare record)
  - DICOM/HL7 integration

The Imaging "Perfect Storm"

Innovation and adoption: new modalities and disciplines



Share
Archive
Integrate

Effective and more frequent utilization of imaging equipment

Technology advancement, more powerful hardware

Motivation for Integrating DICOM with Databases

- Scalability, extremely large data sets
  - Single BLOB of 128 TB, database up to 8 XB
- Ease of development and management
  - Uniform interface for all business data
  - Backup and restore
  - Data migration, integration and remote sharing
  - Information Lifecycle Management
- Search, retrieval and transaction control
  - Reporting, data mining and business activity monitoring
- Security and auditing

Requirements

- Reliable storage and delivery
- Secure data sharing within and across
  - Departments/organizations/regions/nations
- Ease of system integration
- Growth and scalability
- Long term accessibility

Design Objectives

- Open platform capable of managing DICOM content along with all other medical and business information.
- Flexible design to permit the development and customization of applications of different scale, geographical span and business needs
- Lower cost and complexity of developing, deploying, managing and integrating medical applications

Challenges of Integrating DICOM: Managing Attributes

- Evolution of the DICOM standard
  - Extension to a particular branch of medicine
  - New imaging modalities
  - New/updated object/attribute definitions
- Non-conformant DICOM objects
- Large number of attributes
  - 2,000 distinct attributes defined so far
- Private attributes

Challenges of Integrating DICOM: Managing Imagery

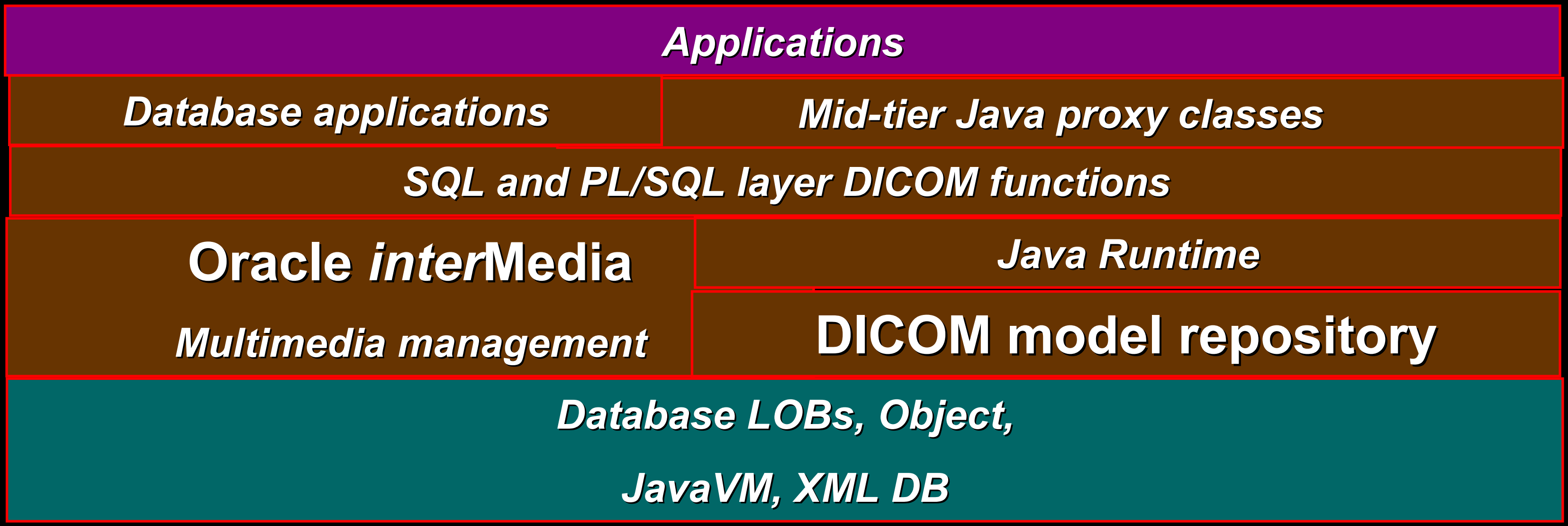
- Ever growing file size
  - Memory management
  - Processing, encryption, …
- Index and search
- Storage scalability
- Backup and restore
- Database tools

Solution

A model-driven architecture that employs a user-configurable repository to control the behavior of DICOM support in a database

- Tight integration
- Secure
- Scalable

Architecture



Applications

Database applications

Mid-tier Java proxy classes

SQL and PL/SQL layer DICOM functions

Oracle *inter*Media

Multimedia management

Java Runtime

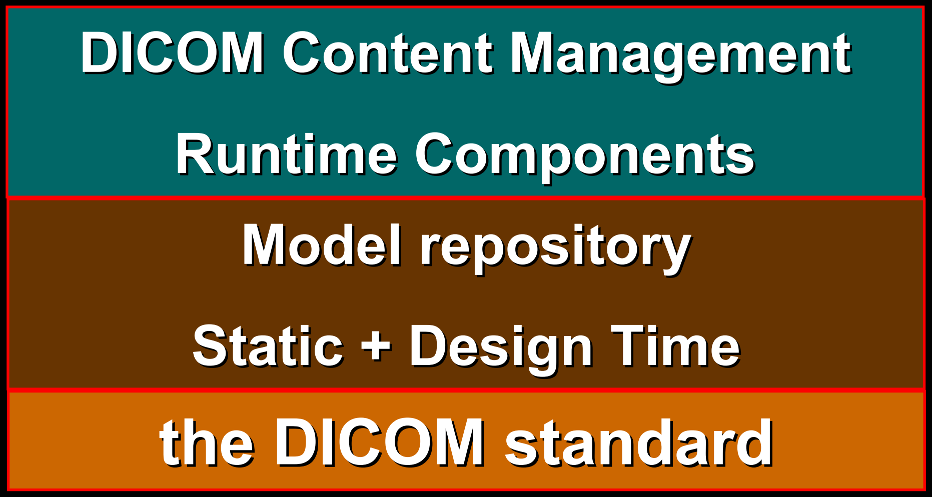
DICOM model repository

Database LOBs, Object,

JavaVM, XML DB

Key Concept: Repository

- Use pluggable model repository to encapsulate the DICOM standard definitions



DICOM Content Management

Runtime Components

Model repository

Static + Design Time

the DICOM standard

Repository, Cont.

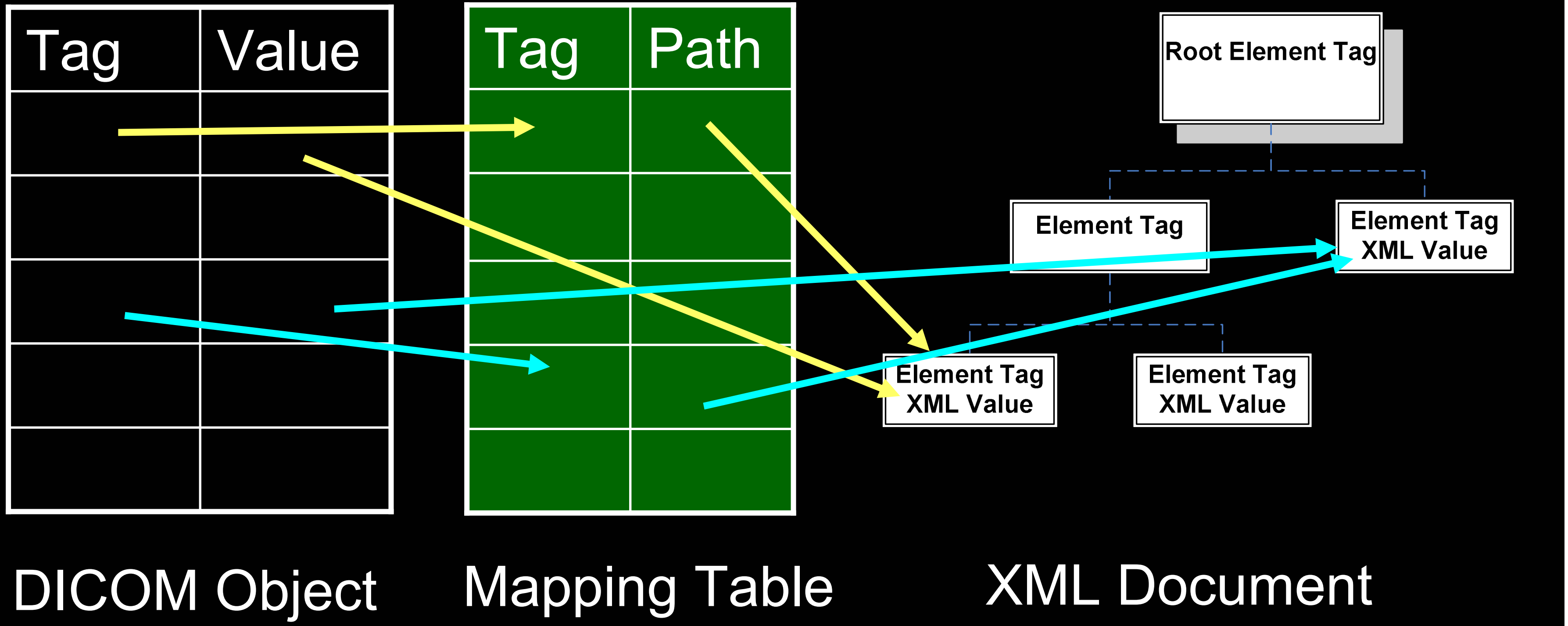
- Repository stores the data model that defines the behavior of a database instance
- Data model can be customized by a DBA
- Data model defines
  - DICOM data dictionary (part 6)
  - DICOM private attributes
  - DICOM to XML mapping
  - DICOM validation rules (part 3)
  - …

Key Concept: Data Type Definition

- Built-in support and validation for all VRs
- Canonical XML representation for each type of DICOM value representation (VR)
- Strongly typed and coupled to XML
  - Datetime DT → xsd:datatime
  - PersonName PN →
    - encoding="unibyte |phonetic | ideographic"
    - Family, Given, Middle, Prefix, Suffix
- Valid XML document ⇔ parseable DICOM object

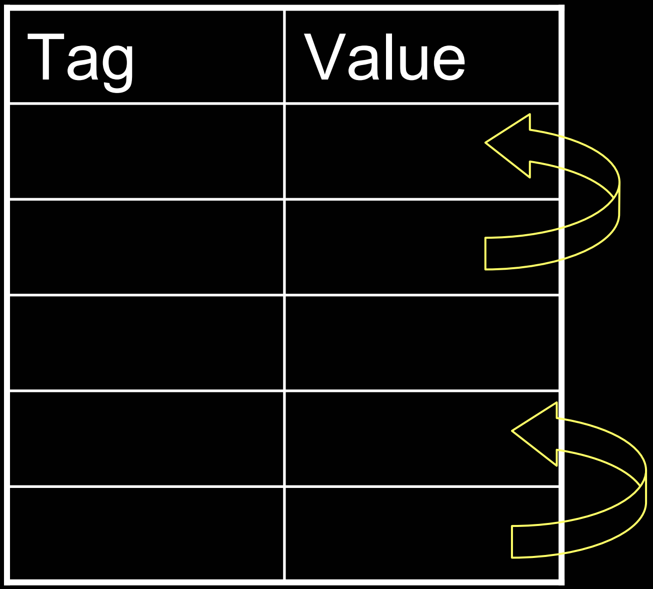
Key Concept: XML Mapping

- Use a mapping table to determine the XML representation of a DICOM attribute



Key Concept: Constraint

- Separate DICOM metadata content encoding and its constraint definitions



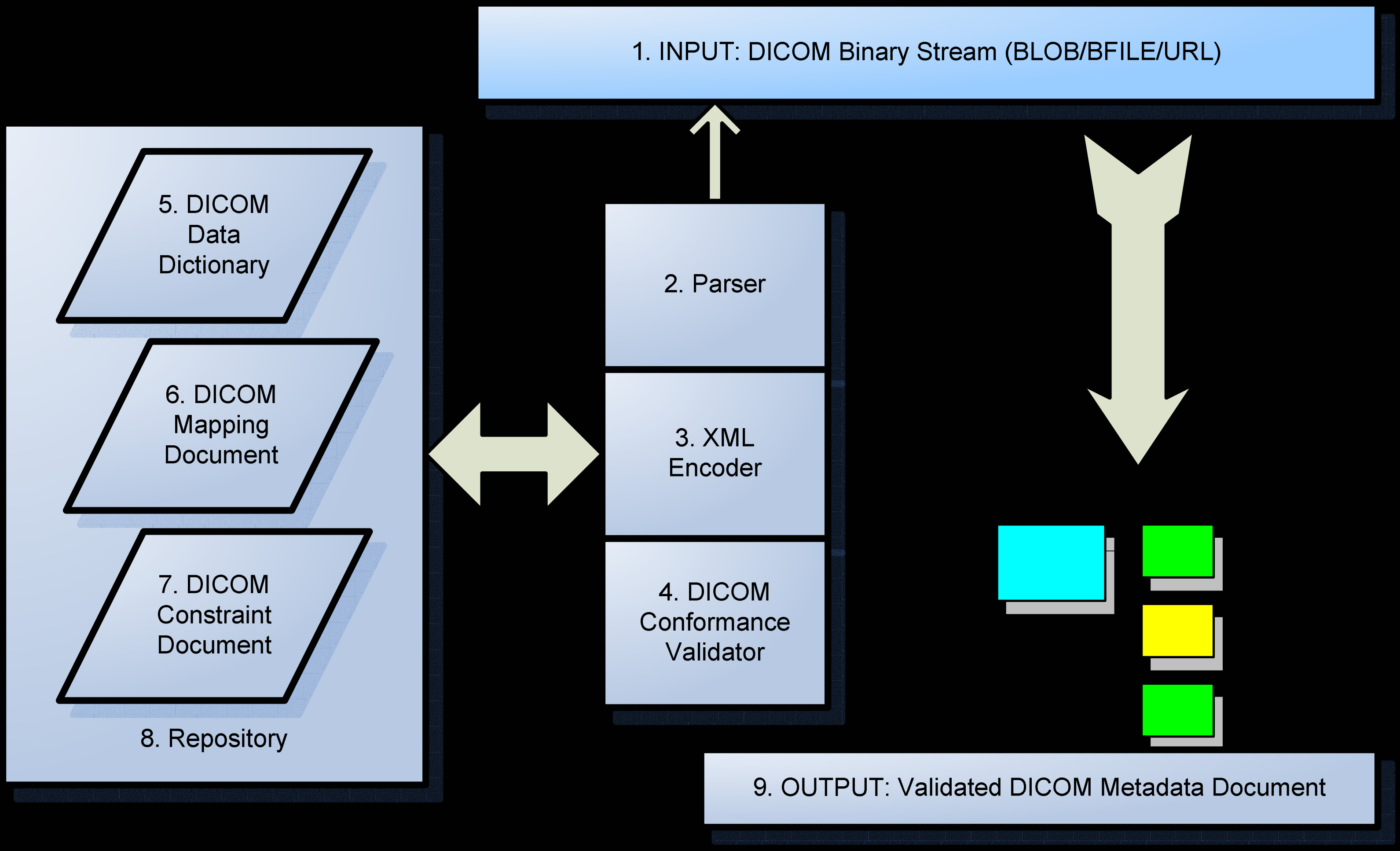
Tag	Value

DICOM Object Constraints

Constraint, Cont.

- Constraint can be grouped into validation rules
  - To verify the conformance of a DICOM object
  - Can be SOP class specific
  - Managed by the model repository
  - Can be disabled or updated at any time

Putting It All Together


1. INPUT: DICOM Binary Stream (BLOB/BFILE/URL)
5. DICOM Data Dictionary
6. DICOM Mapping Document
7. DICOM Constraint Document
8. Repository
2. Parser
3. XML Encoder
4. DICOM Conformance Validator
9. OUTPUT: Validated DICOM Metadata Document

DICOM Support in Databases

- Metadata management
- Indexing
- Image transcoding, processing and compression
- Transfer syntax conversion
- Anonymization
- Extensibility
- …

Example: Metadata Management

- Interpret and validate DICOM attributes (metadata)
- DICOM Object → XML metadata
- XML metadata + a non-DICOM image → a DICOM image
- Index and search DICOM objects by their metadata

User Defined XML Metadata Schema

Instead of extracting DICOM attributes and encoding them into a predefined XML schema, we extract attributes to fit a user defined XML metadata schema

- Tightly coupled with XML mapping table
- Constraints are defined external to XML
- Follows the XML metadata schema template

User Defined XML Metadata Schema, cont.

- Tuned to a specific application to achieve the best performance, interoperability and readability
- Avoid storing and processing metadata attributes that are irrelevant to an application
- Enforce localized conformance
  - Patient age to be type 1

XML Metadata Schema Template

- Strongly typed metadata attributes
- Two parts:
  - Attributes defined in the mapping document
    - <UserTag>value</UserTag>
    - Tree structure
  - Unmapped attributes
    - <type tag=“HHHHHHH”>value</type>
    - Flat list (optional)

Summary

A model driven DICOM solution:

- Invariant to modification of the DICOM standard
- Online upgrade of DICOM repository model
- Supports private attributes
- Lossless XML mapping
- Customizable XML schema
- Customizable conformance validation

QUESTIONS
ANSWERS