

Overview:

# Encapsulation of OBJ Models for 3D Manufacturing and Virtual Reality

Allan Noordvyk & Justin Ryan

Co-Chairs of WG17: 3D Manufacturing

- **Background**
- **Direction & Current Challenges**
- **Main Components**
- **Expected Use**
- **Specific Changes**

# Background

- Allow store/query/retrieve 3D models, intended for 3D manufacturing (and virtual reality), as DICOM objects
- Addressed by Work Item 1
- Leverage
  - a) Existing and growing ecosystem of DICOM-capable systems in use in healthcare institutions and
  - b) Standards and conventions already in use in the 3D printing industry

- In 2018 WG17 focused on getting **DICOM Encapsulated STL** was added to the standard
  - This provides a lowest common denominator for use cases
  - It was recognized that while everyone can utilize STL, there are more advanced options

- Approached by members of medical **Virtual Reality (VR)**, **Augmented Reality (AR)**, and **Mixed Reality (MR)**
- This community also uses non-medical 3D models and have **overlapping use cases** with 3D manufacturing
- WG17 is now including **their input** into selection of formats for encapsulation and other needs
  - Primary format in current VR/AR/MR use is **OBJ**
  - Also concerned with multi-part assemblies and color



# Direction & Current Challenges

- Both the **3D printing** and **AR/VR/MR communities** (together known as XR) have provided the following direction to WG17...
  - Address limitations of STL by allowing option for encapsulation of a more advanced format
    - Select based on **current** ubiquity of use in both communities
  - Address model management challenges related to...
    - Multi-part assemblies
    - Persistent component color

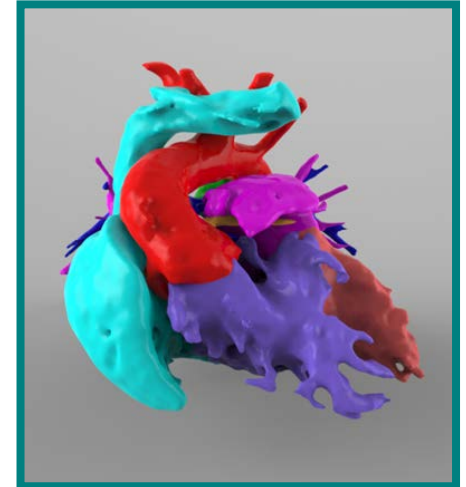


# Challenge 1: Beyond STL

- Limitations of STL format
  - No ability to indicate color/texture individual polygons in model
    - Important for replicating real-world appearance of modeled anatomy/pathology or delineate between structures
  - Poor adoption in virtual/augmented/mixed reality applications
- Many other 3D model file formats address these deficits (OBJ, X3D, AMF, 3MF)
- **OBJ format** has high **current** adoption among both 3D printing and VR/AR/MR applications & users

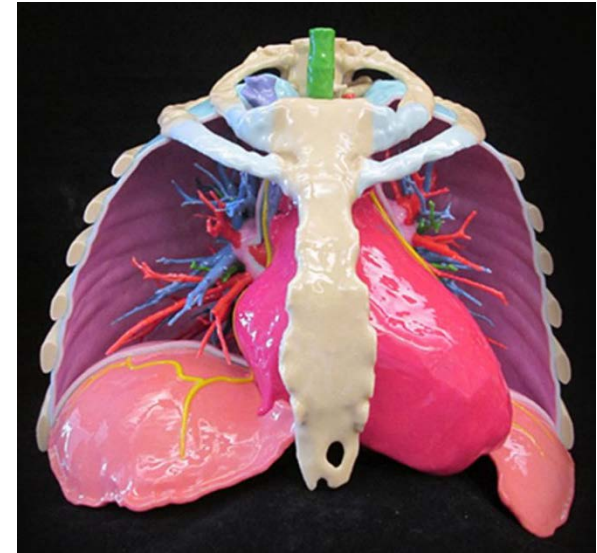
# Challenge 2: Assemblies

- Many 3D models are meant to be assembled together, example:
  - Multi-part implants
  - Training simulators requiring different materials
  - Explorable anatomic models
- May be multiple assemblies in the same DICOM study, example:
  - Left and right versions of multi-part implants
- Any convention using study and series can be ambiguous and inconsistent
- Desire to **explicitly** leverage DICOM identify which subset of models belong to the **same assembly**



# Challenge 3: Persistent Color

- Many situations where **specific preferred color** should be used for a specific model
  - Example: Color-coded assemblies of multiple models (bone, venous, arterial, ...)
- No good solution inside STL or OBJ models...
  - STLs **completely lacks** standard ability to indicate the **color** of the model
  - OBJ can indicate color, but it must be done on the **polygon-by-polygon or vertex-by-vertex** level (overkill)
- Desire to leverage DICOM to persistently indicate **desired color** for a specific model



# Main Components of Supplement

- The second output of work item 1 is *Supplement 208: DICOM Encapsulation of OBJ Models for 3D Manufacturing and Virtual Reality*
- Enable encapsulation of OBJ in a pathway similar to STL encapsulation
- Augment current encapsulation approach for assemblies and color

## New Information Object Definition (IOD)s:

- **Encapsulated OBJ** (and supporting files) for Creation, Review, Update, and Printing (manufacturing)
- Encapsulated MTL

## New Attributes:

- Reference Name
- Model Group
- Recommended Display CIELab Value
- Recommended Presentation Opacity
- Referenced Instance Sequence
- Referenced Image Sequence

## New IOD Devices:

- Texture Map

# Expected Use

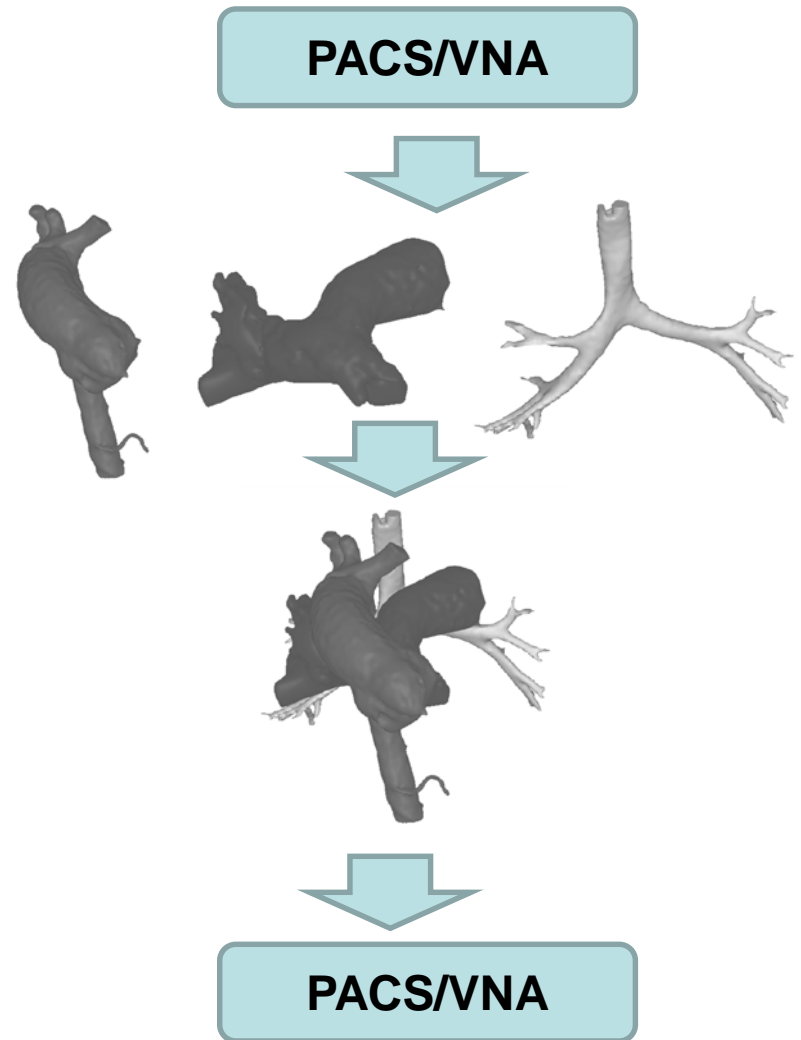
The new 3D Model encapsulation attributes is expected to address these real world use cases

- **Model Group**
- **Component Color/Opacity**



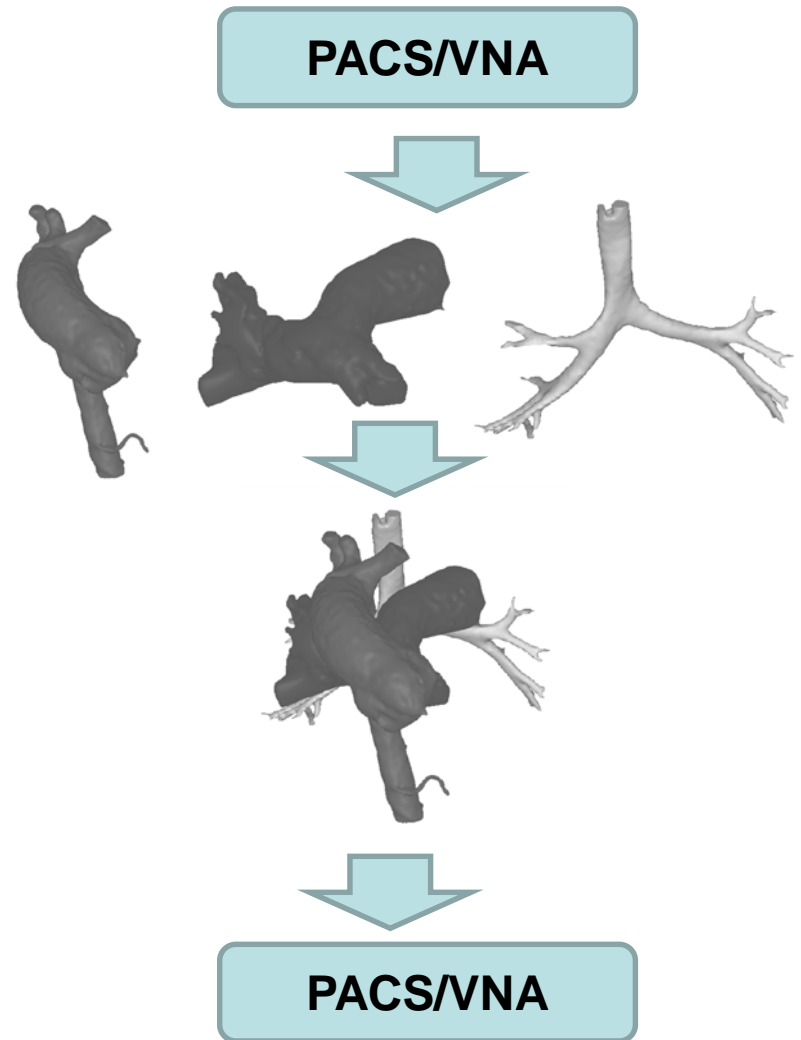
# Expected Use: Model Group

- Medical reconstruction software queries Image manager system
- **User creates patient-specific 3D model** (reconstruction and modeling)
- User segments different regions into discrete manifolds (e.g., aorta, pulmonary artery, and airway)
- Modeler system creates 3 DICOM objects containing the 3D models
- Specifying **same Model Group UID in each object** enables modeler or subsequent DICOM-enabled software to identify group for joint printing / presentation
- [To 3D printer or XR system]



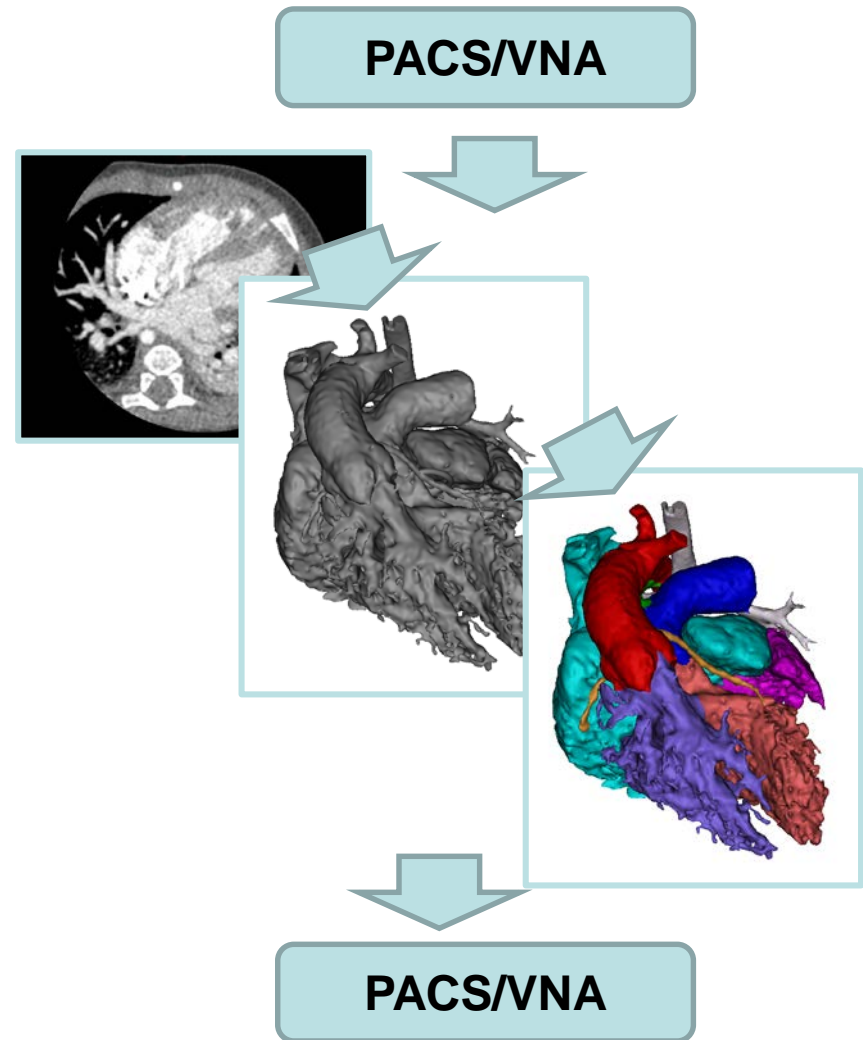
# Expected Use: Model Group (Alternate)

- Medical reconstruction software queries Image manager system
- User queries **DICOM Segmentation** objects and imports into 3D modeler software
- Modeler system creates 3 DICOM objects containing the 3D models
- Specifying **same Model Group UID in each object** enables modeler or subsequent DICOM-enabled software to identify group for joint printing / presentation
- [To 3D printer or XR system]



# Expected Use: Component Color

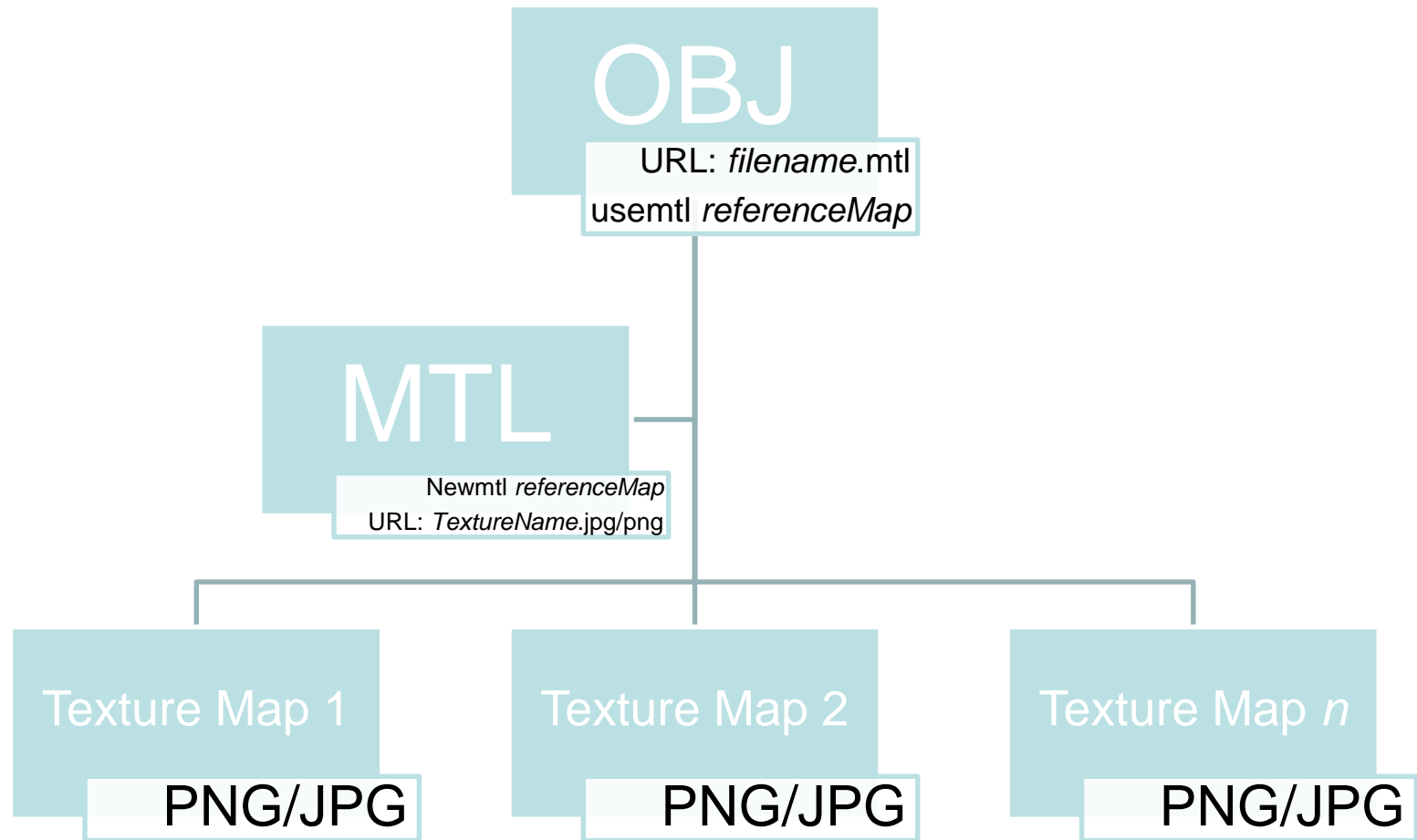
- Medical reconstruction software queries Image manager system
- **User creates patient-specific 3D model** (reconstruction and modeling)
- User segments different regions into discrete manifolds (e.g., aorta, left ventricle, left atrium)
- Modeler system creates 3 DICOM objects containing the 3D models
- **Color each component** (Recommended Display CIE Lab Value)
- **Assign alpha/transparency** (Recommended Presentation Opacity Value)
- [To 3D printer or XR system]



## The proposed OBJ encapsulation necessitates multiple file encapsulation

1. An OBJ object may actually be comprised of 2 or more files:
  - 1 **OBJ** main file
  - 0-1 **MTL** supporting file
  - 0-*n* **Texture Map Image** supporting files
2. These files currently refer to each other by filename

# Background - OBJ Schema



# Background: OBJ Referencing MTL

```
1368658 vn -0.3449577391148 -0.6821315884590 -0.6447485089302
1368659 vn -0.2826717197895 0.0076980688609 -0.9591858386993
1368660 vn 0.6301043033600 0.7272020578384 0.2722969949245
1368661 vn 0.5974761843681 0.7891337871552 -0.1424432843924
1368662 vn -0.4890609681606 -0.7319647669792 0.4743911325932
1368663 vn -0.6451370120049 -0.6867179870605 -0.3349875509739
1368664 vn 0.9334340691566 -0.1631953120232 0.3194811046124
1368665 vn 0.0918825566769 -0.7192199826241 0.6886800527573
1368666 vn 0.0357546992600 0.8034136295319 0.5943468213081
1368667 vn 0.9314748644829 0.3603391945362 -0.0501018352807
1368668 vn -0.6770498156548 0.6993246078491 0.2292350381613
1368669 vn 0.6219043731689 0.7298039793968 0.2839385569096
1368670 mllib reference.mtl
1368671 o Polygonal_Model_1 Ao
1368672 usemtl _image
1368673 # Number of triangles: 58932
1368674 f 1161/1/25034 1172/2/25035 1149/3/25036
1368675 f 1174/4/25042 1175/5/25043 1176/6/25044
1368676 f 1175/5/25045 1174/4/25046 1177/7/25047
1368677 f 1175/5/25144 1177/7/25145 1194/8/25146
1368678 f 1176/6/25147 1184/9/25148 1174/4/25149
1368679 f 1229/10/25300 1194/8/25298 1230/11/25299
1368680 f 1229/10/25300 1230/11/25299 1231/12/25301
1368681 f 1231/12/25351 1230/11/25349 1243/13/25350
1368682 f 1231/12/25351 1243/13/25350 1244/14/25352
1368683 f 1393/15/25702 1175/5/27097 1394/16/299
1368684 f 1150/17/25880 1406/18/25881 1407/19/25882
1368685 f 1149/3/25036 1172/2/25035 1408/20/301
1368686 f 1149/3/25036 1408/20/301 1406/18/25037
1368687 f 1415/21/25283 1408/20/301 1172/2/25035
1368688 f 1419/22/25892 1415/21/25890 1172/2/25891
1368689 f 1229/10/25300 1423/23/303 1424/24/25906
```

# Background: MTL Referencing Texture Maps

```
8 newmtl image
9 Kd 1.0 1.0 1.0
10 illum 1
11 d 1
12 Ns 0.125
13 sharpness 60
14 Ni 1
15 map_Kd_wrapU Clamp
16 map_Kd_wrapV Clamp
17 map_Kd_magfilter Linear
18 map_Kd reference_image.png
19
20 newmtl image 2
```

The encapsulation strategy for OBJ will **introduce 2 new DICOM IODs**:

- **Encapsulated OBJ**
  - Stores the main OBJ byte stream
- **Encapsulated MTL**
  - Stores the MTL byte stream

And make minor extensions to 2 existing DICOM modules to support texture maps:

- **Multi-frame True Color SC Image**
- **General Image**



- The URI used in the encapsulated document to reference the SOP Instance in this Item:
  - **Relative URI Within Encapsulated Document**
- Stores the file name under which the object may be referenced in encapsulated objects
  - From earlier examples
    - "matlist.mtl"
    - "ntissue.png"
- It is important to note that when de-encapsulating MTL file, the texture map images must be restored to both their original file name and file format.

- A new attribute is added to the Encapsulated OBJ object
  - **Referenced Instance Sequence**
- A new attribute is added to the Encapsulated MTL object
  - **Referenced Image Sequence**
- This is a sequence of UIDs for the:
  - **Encapsulated MTL**
  - **Texture Map Images**
- This allows a simple DICOM query to easily retrieve all of the supporting objects for a given Encapsulated OBJ

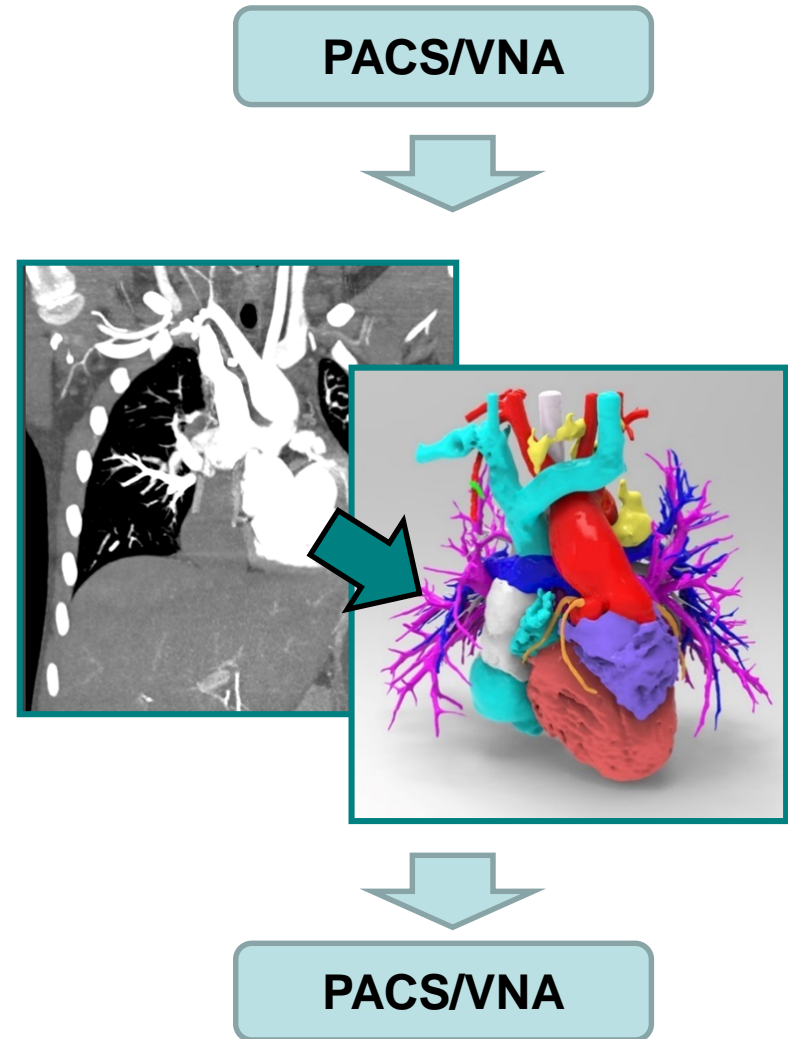
# Use Cases for OBJ Encapsulation

**The new IOD/SOP is expected to address these real world use cases:**

- **Creation**
- **Review / XR Display**
- **Update**
- **Print**

## Use Case 1: Creation

- Medical reconstruction software queries Image manager system
- **User creates patient-specific 3D model** (reconstruction and modeling)
- Alternatively, user queries a **DICOM Segmented Object**
- Modeler system creates the new type DICOM object containing the 3D model along with color information, populating all required metadata
- **User saves 3D model back to the patient's record in DICOM** format as either (a) an addition to an existing study or (b) a new study
- The **Modeler system stores the new DICOM** object in the Image Manager system

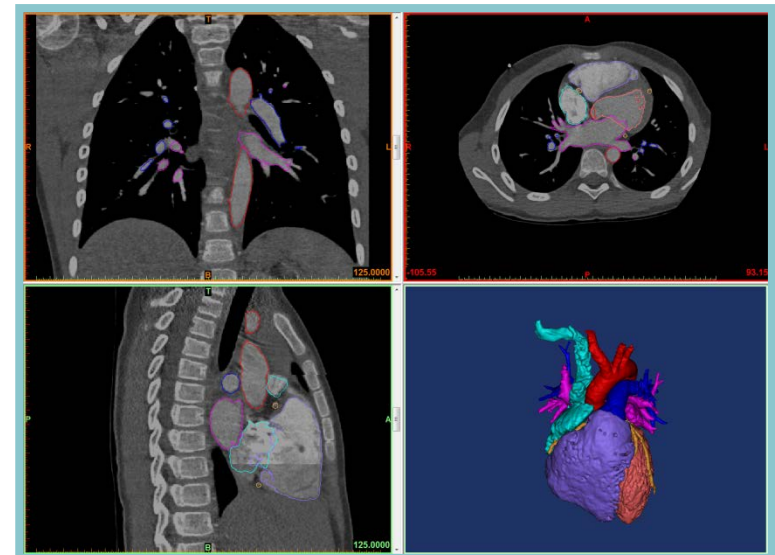


## Use Case 2: Review

- At a later time to Use Case 1, a user indicates desire to **visually review a 3D model , prior to 3D printing**
- The Display system queries the Image Manager for the DICOM objects of new type
- The Display system retrieves the indicated object
- The 2D or XR **Display system extracts the 3D model** from the object and displays it to the user, potentially registered for simultaneous display with source images

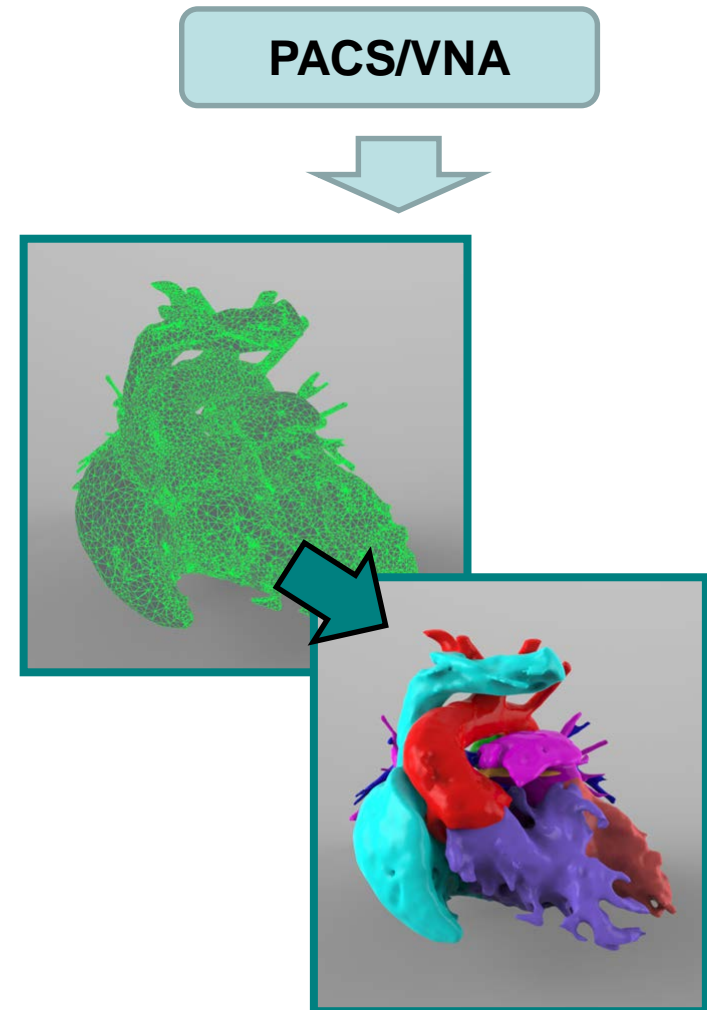
PACS/VNA

Display System



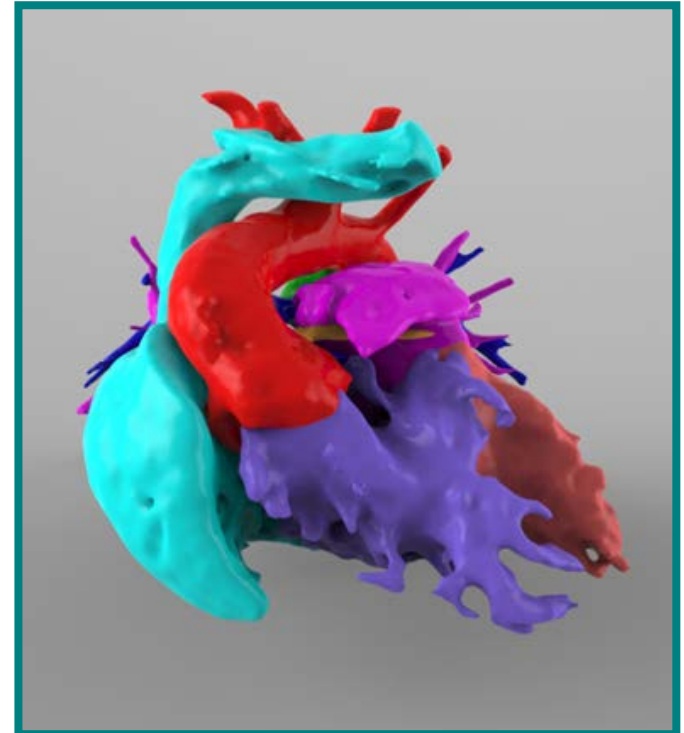
## Use Case 3: Update

- At a later time to Use Case 1, a **user indicates desire to modify a 3D model** for a particular patient
- The Modeler system queries the Image Manager for the DICOM objects of new type
- If necessary, **the Modeler system retrieves any source images (s1 to sN)** required for this modification to occur
- User interacts with the Modeler system to adjust the 3D printable model as desired



## Use Case 3: Update (cont'd)

- **User saves back to the patient's record in DICOM format** as either (a) an addition to an existing study, or (b) a new study
- The Modeler system creates the new type DICOM object containing the new version 3D model, populating all required metadata and including a unique identifier reference to the supplanted earlier 3D print model object
- The Modeler system stores the new DICOM object in the Image Manager system

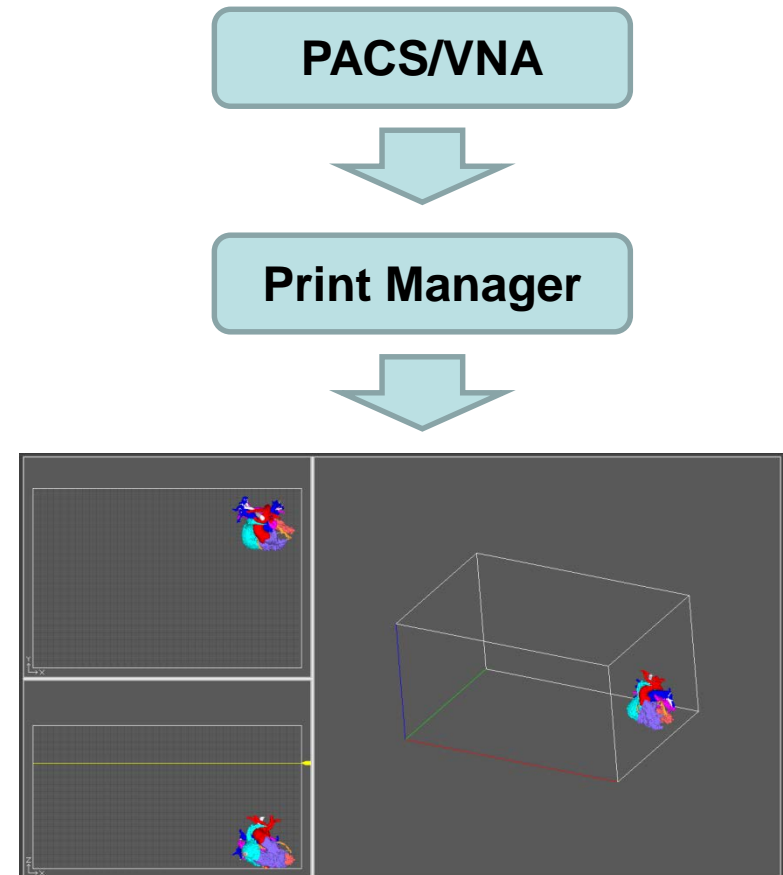


**PACS/VNA**



## Use Case 4: Print

- At a later time to Use Case 1, a user indicates desire to print a 3D model for a particular patient
- The Print Manager system queries the Image Manager for the DICOM objects of new type belonging to the patient
- The Print Manager system retrieves the indicated 3D print model object
- The Print Manager access the 3D model information within the object, using this to create non-DICOM print instructions for a specific 3d printer (e.g. \*.obj)



## Use Case 4: Print (cont'd)

- The Print Manager prompts the user for any necessary additional print parameters (e.g. support, bed placement, material parameters, etc.)
- The Print Manager submits the print job to the printer
- Optionally, the Print Manager may save an updated 3d print object back to the Image Manager in order to preserve exact print parameters used (per Use Case 3, steps 7+).

**Print Manager**



**PACS/VNA**

# Specific Changes to Standard

## ISSUE TO RESOLVE

- Encapsulated models lack uniform inherent component color

## ADDRESS VIA

- **New Attribute: Recommended Display CIELab Value**
  - Specifies color for DICOM component
  - Modeled on similar attributes in the standard

Attribute Name	Tag	Type	Attribute Description
<b>Recommended Display CIELab Value</b>	(0062,000D)	3	<p>Specifies the color recommended to be used for the model when visually representing and selecting material for manufacturing. This would typically be used to visually distinguish between models that are part of the same assembly and/or provide best analog to real world appearance.</p> <p>The CIELab triplet value is specified in PCS-Value units. See section 10.7.1.1.</p> <p>This value should be ignored if individual colors have been specified inside the encapsulated model (when encapsulated format allows this).</p>

Addition to C.35.1 Manufacturing 3D Model Module

## ISSUE TO RESOLVE

- Encapsulated models lack uniform inherent component opacity

## ADDRESS VIA

- **Extended Attribute: Recommended Presentation Opacity**
  - Specifies opacity for DICOM component
  - a fraction between 0.0 and 1.0

Attribute Name	Tag	Type	Attribute Description
<b>Recommended Presentation Opacity</b>	(0066,000C)	3	<p>Specifies the opacity in which it is recommended to be used for the model when visually representing and selecting material for manufacturing.</p> <p>A value of zero is interpreted as complete transparency, while a value of 1 is interpreted as fully opaque.</p> <p>A non-opaque value would typically be specified when either (a) another model grouped in the same assembly needs to be visible behind or inside this model, or (b) the model represents anatomy that is not fully opaque.</p> <p>If not present, then it is assumed the model should be presented and manufactured as opaque.</p> <p>See Section C.27.1.1.3.</p>

## Extended Use of C.27.1.1.3 Recommended Presentation

## ISSUE TO RESOLVE

- Encapsulated models that are part of same assembly have no inherent grouping
  - Relying on humans to guess grouping of numerous DICOM encapsulated models is problematic

## ADDRESSED VIA

- Optional **Model Group** UID
- Explicitly allows model grouping if part of same assembly

Attribute Name	Tag	Type	Attribute Description
<b>Model Group</b>	(aaa1,bbb1)	3	UID shared by manufacturing models that are considered distinct parts within the same assembly.

Addition to C.35.1 Manufacturing 3D Model Module

- **Builds on approach used for encapsulation of STL**
  - Encode OBJ and MTL files via *Encapsulated Document* (0042,0011) attribute
  - Store texture map Images as Multi-frame True Color Secondary Capture
  - Registration of 2 new MIME types *model/obj* and *model/mtl* to be completed with IANA

## ISSUE TO RESOLVE

- How does someone locate & retrieve all of the DICOM objects that an OBJ model directly and indirection references?

## ADDRESS VIA

- **New Attribute: *Referenced Instanced Sequence***
  - Designation of UIDs of
    - OBJ -> MTL

Attribute Name	Tag	Type	Attribute Description
Referenced Instance Sequence	(0008,114A)	3	<p>Sequence of UIDs corresponding to supporting instances directly referenced within the encapsulated model.</p> <p>In an Encapsulated OBJ, only a single item shall be permitted in this sequence and that item shall be the UID of a Encapsulated MTL instance.</p> <p>In an Encapsulated MTL, all items shall be images and have the Reference Name attribute defined.</p>

Addition to C.35.1 Manufacturing 3D Model Module



## ISSUE TO RESOLVE

- How does someone locate & retrieve all of the DICOM objects that an OBJ model directly and indirection references?

## ADDRESS VIA

- **New Attribute: *Referenced Image Sequence***
  - Designation of UIDs of
    - MTL -> Texture Maps

Attribute Name	Tag	Type	Attribute Description
Referenced Image Sequence	(0008,1140)	3	The set of image instances referenced in the encapsulated document. One or more Items are permitted in this Sequence.

Addition to C.35.1 Manufacturing 3D Model Module

## ISSUE TO RESOLVE

- OBJ files refer to MTL files by file name
- MTL files refer to texture map images by file name
- This naming must be preserved when recreating the files or linkage broken

## ADDRESS VIA

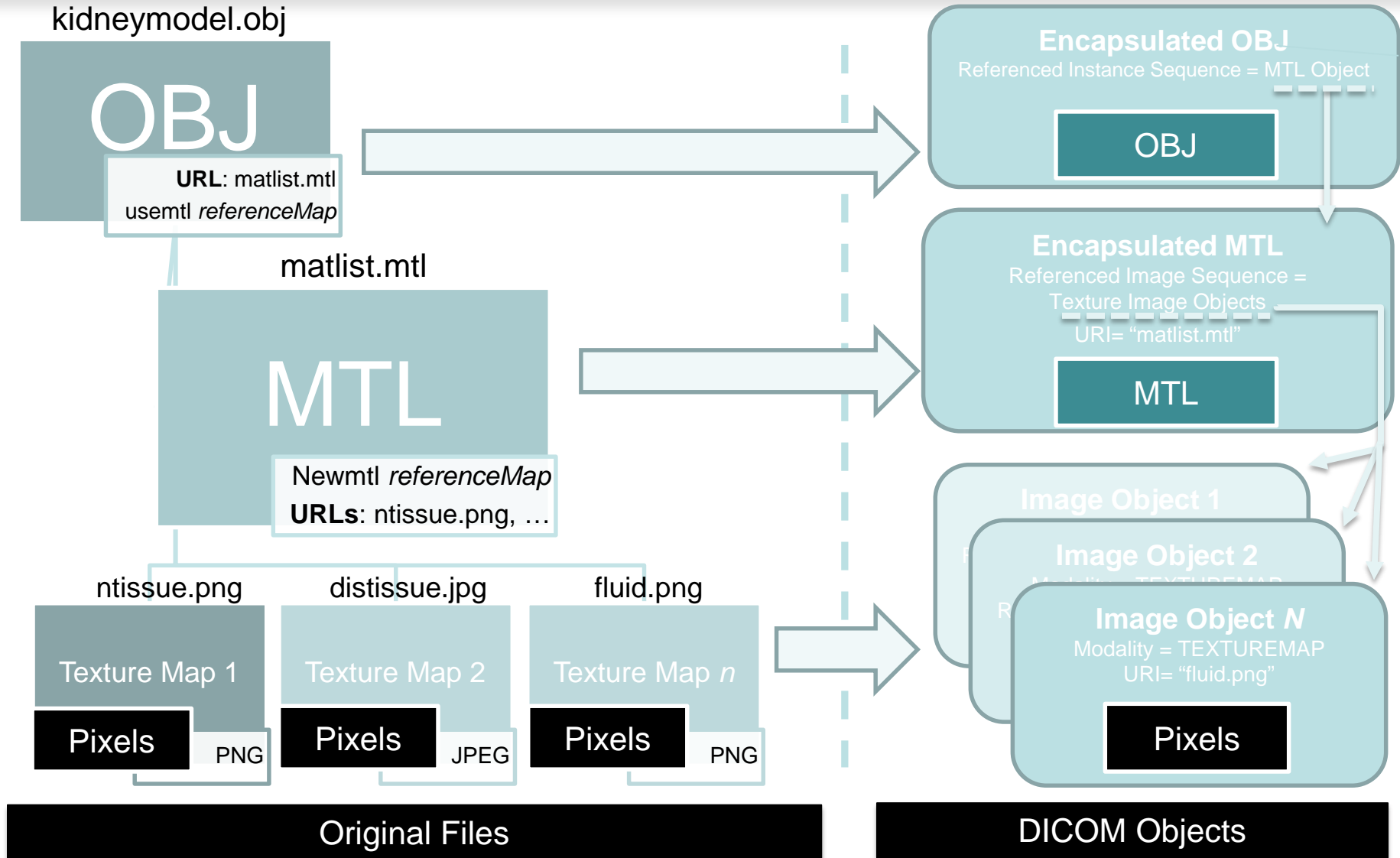
- **New Attribute: *Reference Name***
  - Allows disambiguation and recreation of files when de-encapsulating
  - Stored in the Encapsulated MTL and the Texture Map Image objects

Attribute Name	Tag	Type	Attribute Description
<b>Reference Name</b>	(aaa1,bbb3)	1C	The name that will be used when the data is re-constituted as an unencapsulated file. This preserves referential integrity. This attribute is required if the object is referenced from an encapsulated object.

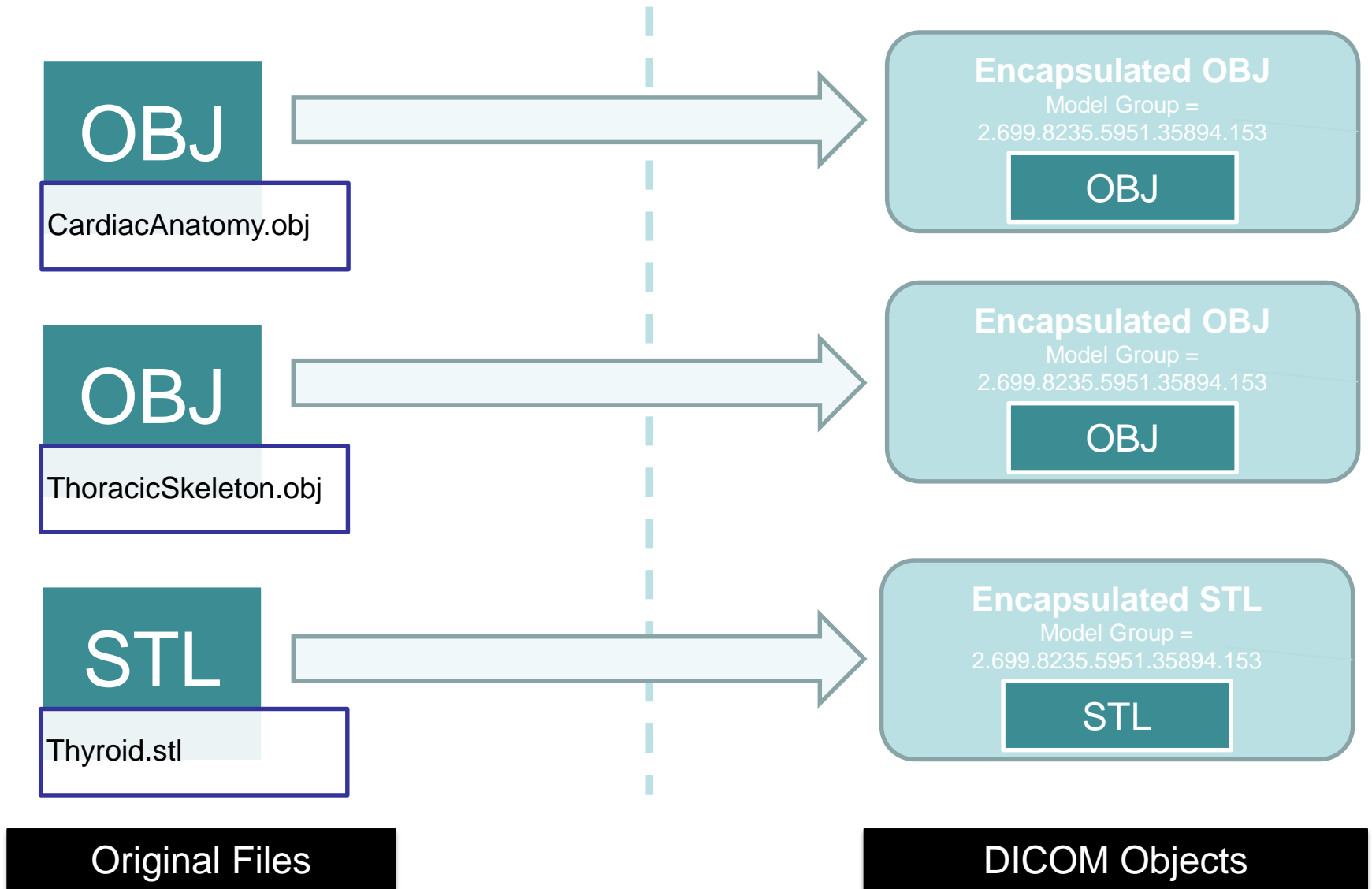
Addition to C.35.1 Manufacturing 3D Model Module

Addition to General Image Module (for Texture Map Images)

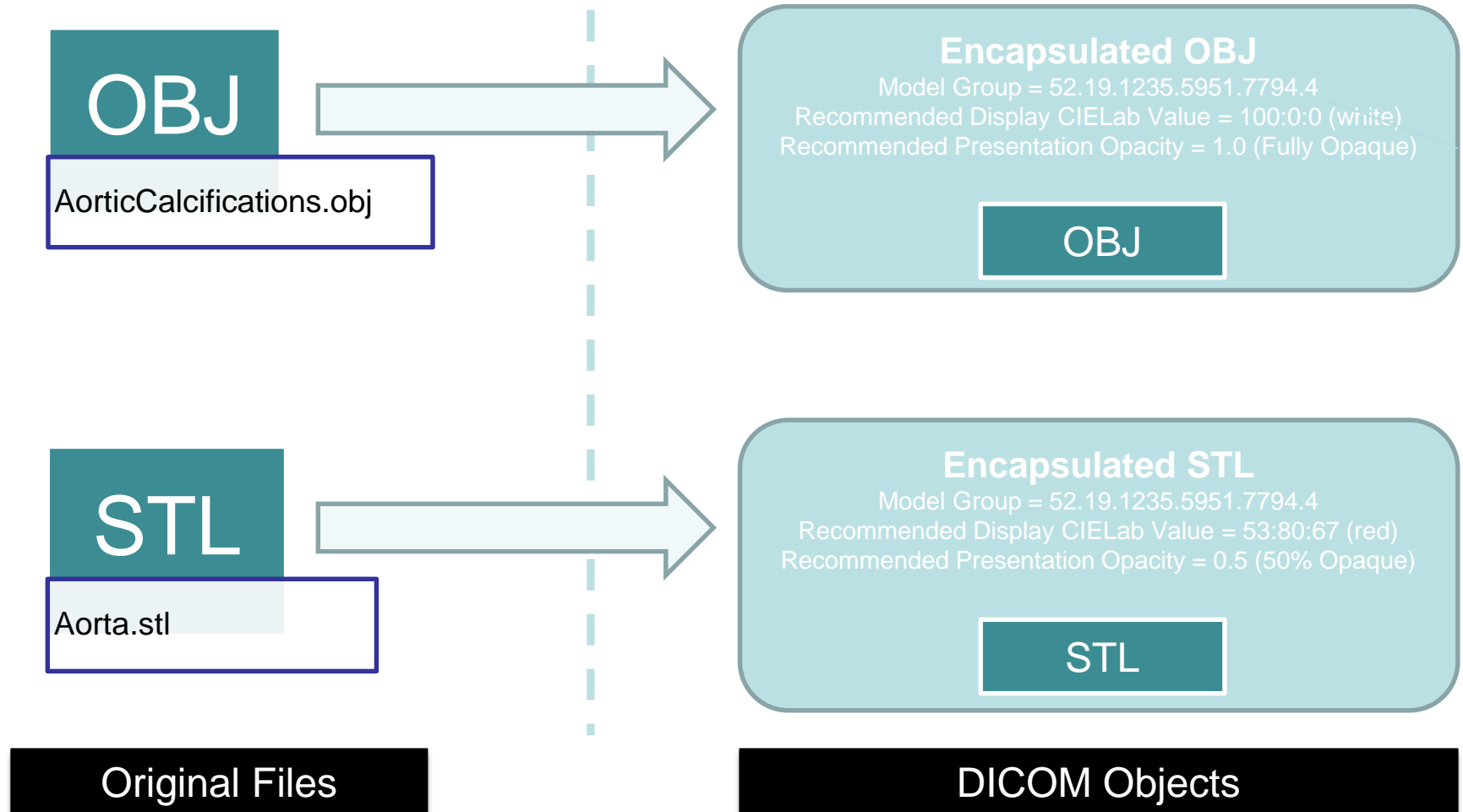
# Ex. OBJ, MTL, and TM Encapsulation



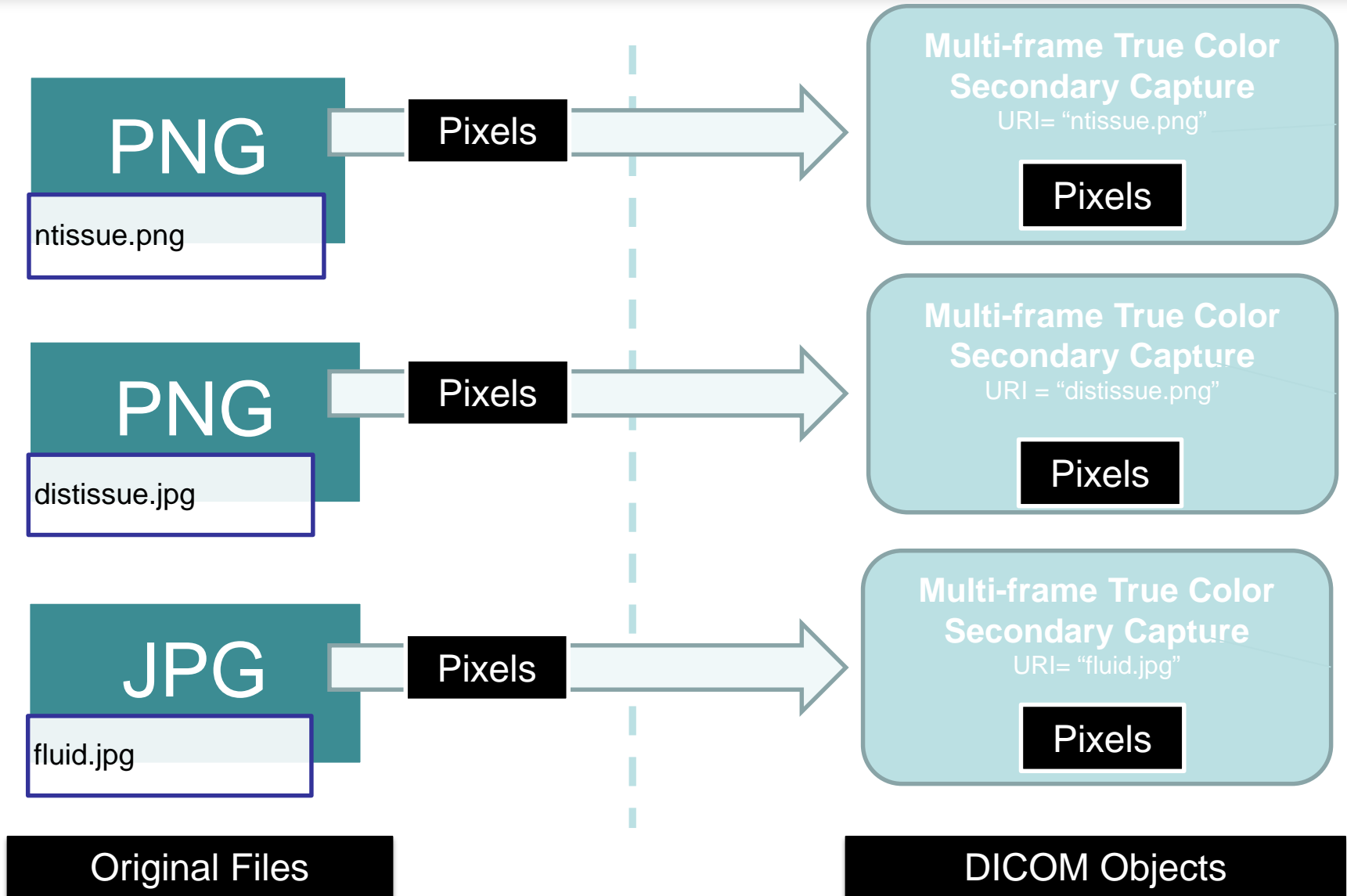
# Ex. Model Group



# Ex. Model Group, Color, and Opacity



# Ex. Texture Mapping



# Thank You for your Attention



## END