Advanced Application of the DRM – Terrain

SEDRIS™ Technology Conference
Lake Buena Vista, FL
7 January 2004

Kevin C. Trott
Northrop Grumman Information Technology
kevin.trott@ngc.com

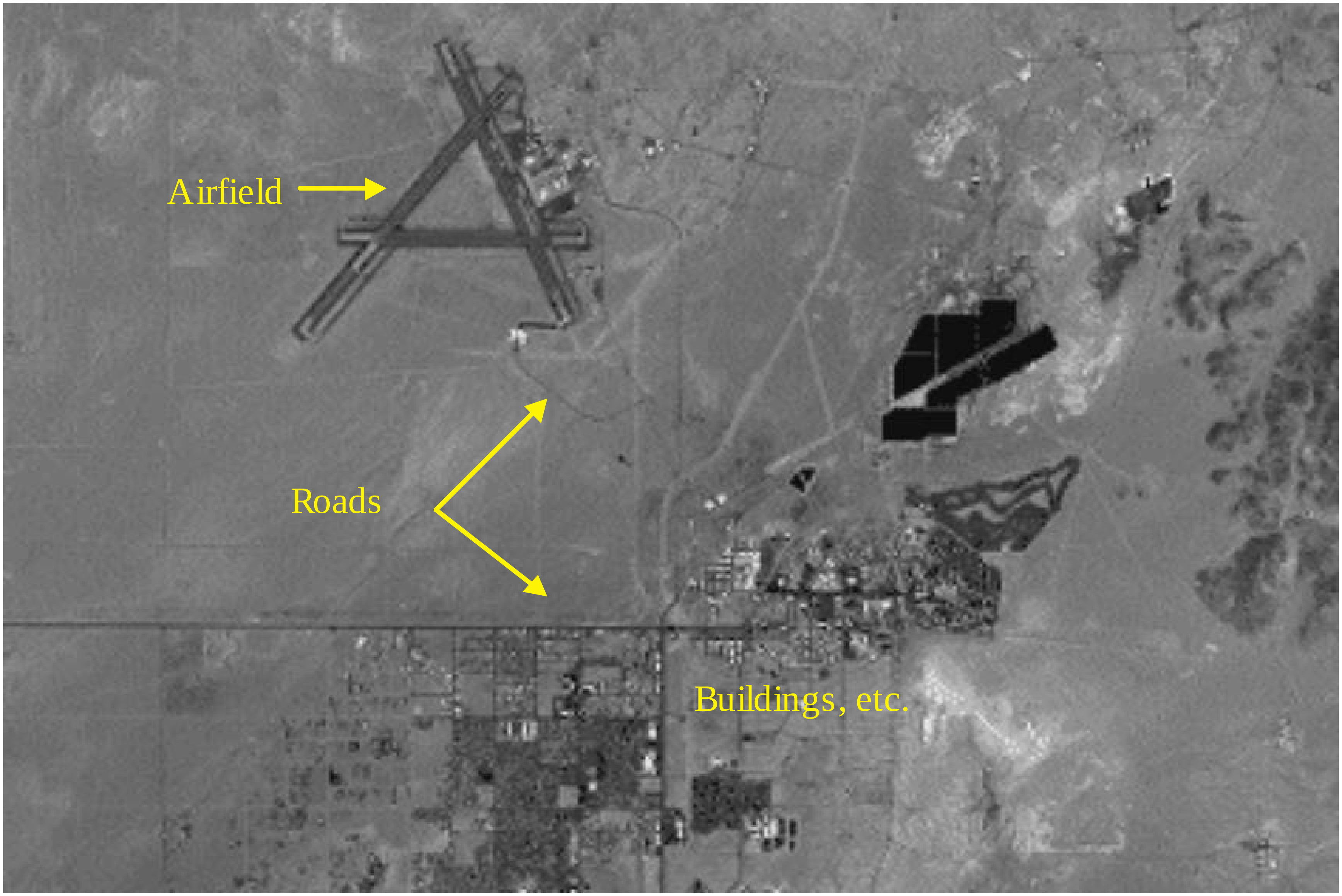
Outline

- Raster Data
  - Definition, Sources, & Uses
  - Raster Data in the DRM – Images
- Gridded Data
  - Definition, Sources, & Uses
  - Gridded Data in the DRM – Property Grids
  - DTED-based SEDRIS Transmittal Structure
- Vector Data
  - Definition, Sources, & Uses
  - Vector Data in the DRM – Features
  - Feature Topology
  - VPF-based SEDRIS Transmittal Structure
- Polygonal Data
  - Definition, Sources, & Uses
  - Polygonal Data in the DRM – Geometry
  - Geometry Topology

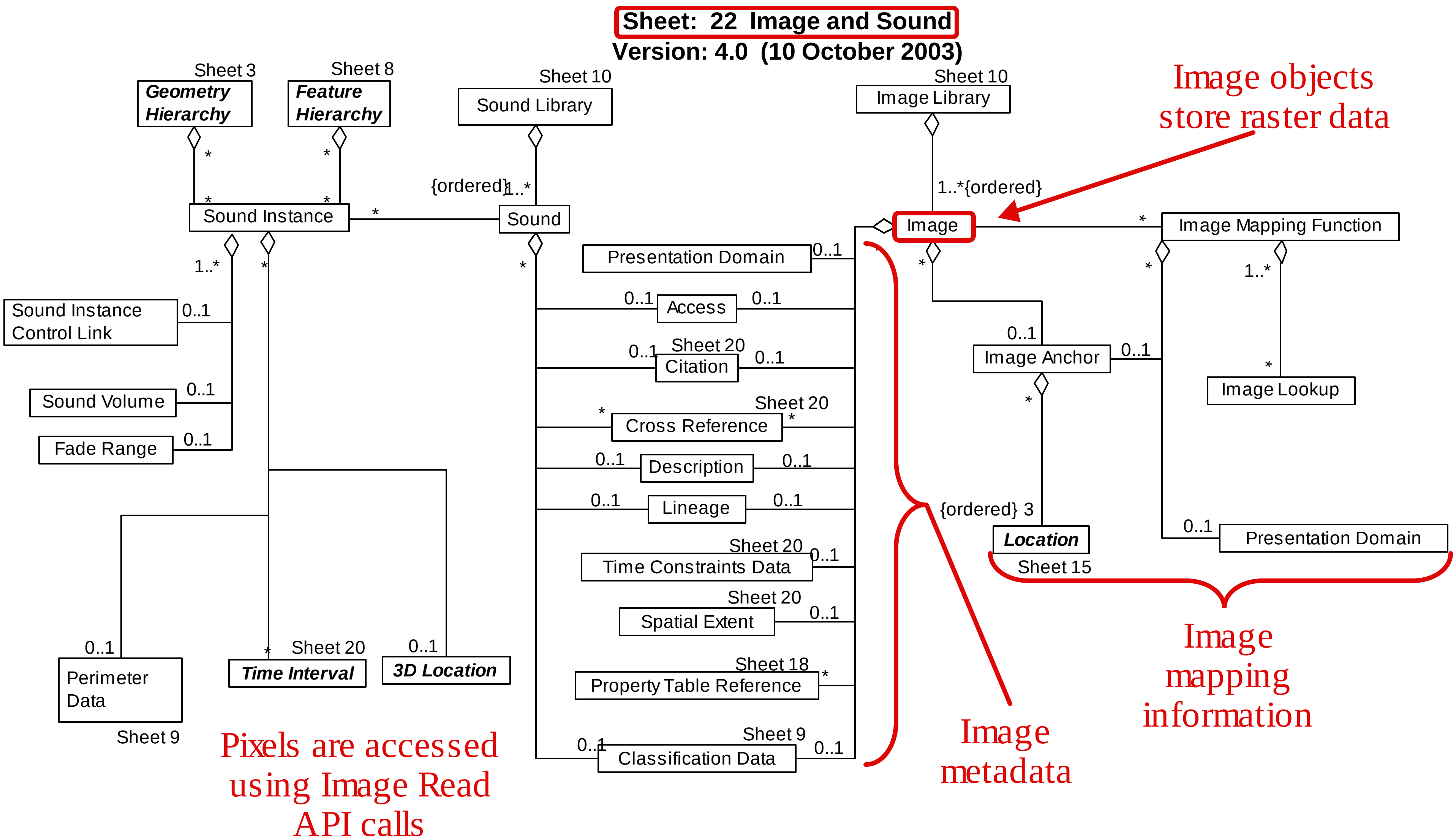
Raster Data

- Raster Data – a regular multi-dimensional array of intensity values (i.e., pixels)
  - Geospatial Imagery – used for ground texture and/or feature extraction
  - Scanned Maps – used for map background displays (not very accurate)
  - Other photos/images – used for generic ground texture, model texture, etc.
- Raster Data Sources
  - NGA's Raster Product Format (RPF)
    - Controlled Image Base (CIB)
      - High-resolution (5-10m) digital imagery
    - Compressed ARC Digital Raster Graphics (CADRG)
      - Scanned hardcopy maps at a variety of scales
  - USGS Digital Raster Graphics (DRGs)
  - Landsat, Spot, & other commercial geospatial imagery
  - Other miscellaneous satellite, aerial, or ground-based imagery

Raster Data Example – CIB



Raster Data in the DRM – Images



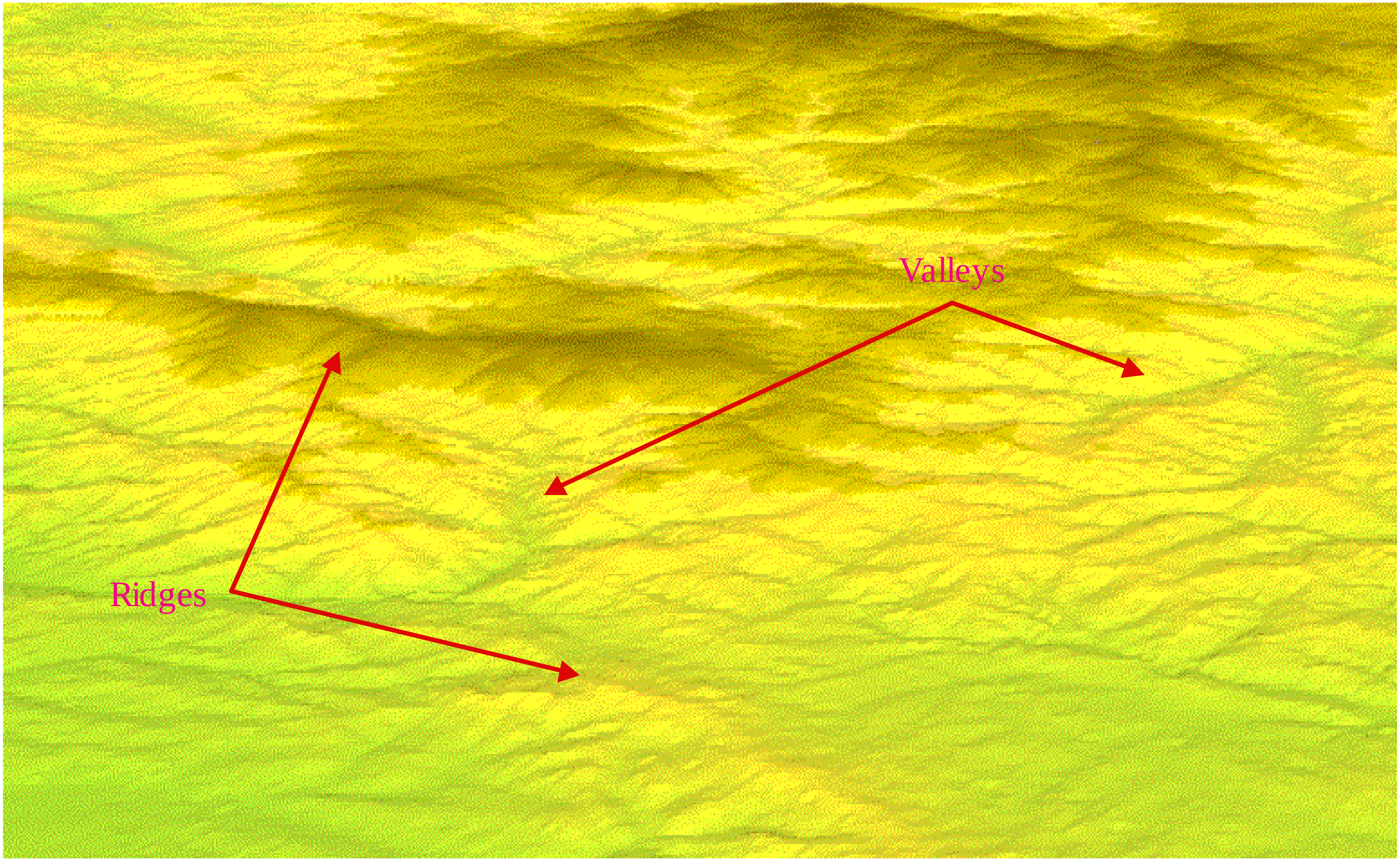
Raster Data Summary

- Raster Data is stored in Image objects
- Raster Data content (i.e., pixel values) is accessed using specific Image-related functions in the Read and Write APIs
- All Image objects in a transmittal are contained in the Image Library
- Images may be geospecific or generic
  - An Image can be mapped to a specific location to provide geospecific texture
  - A Image can be reused many times to provide generic texture information
- Image Mapping Functions are used to register Images to specific locations
- Images can be associated with Features, Geometry, and Data Tables through Image Mapping Functions
- Each Image object can have a complete set of supporting metadata

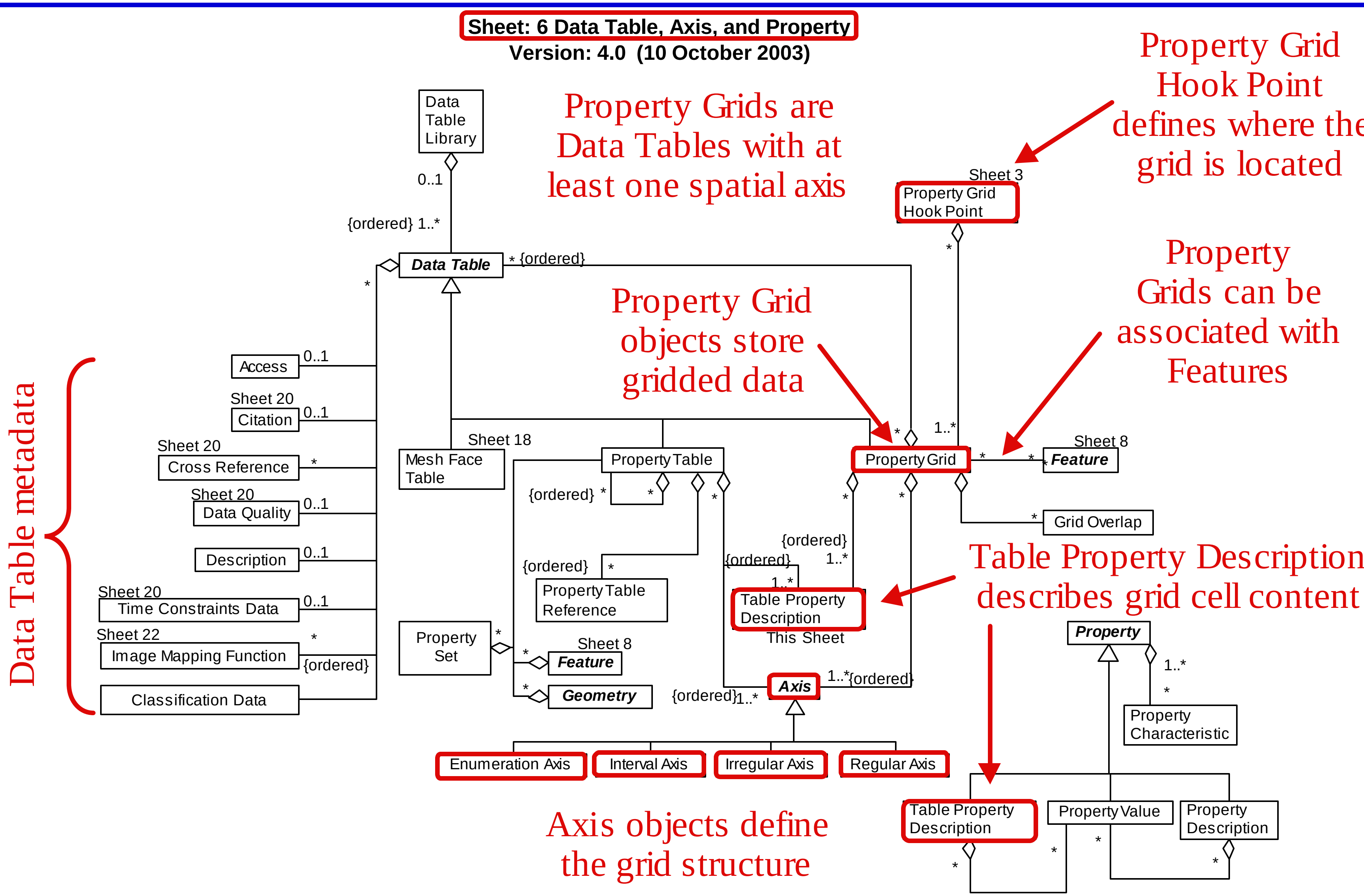
Gridded Data

- Gridded Data – a multi-dimensional array of measured or estimated values associated with specific locations
  - Digital Terrain Elevation Models - height values
  - Bathymetry Models - depth values
- Gridded Data Sources
  - NGA's Digital Terrain Elevation Data (DTED)
  - NGA's Digital Bathymetric Data Base (DBDB)
  - USGS Digital Elevation Models (DEMs)
- Most atmospheric and ocean data is also in either 2D or 3D gridded form

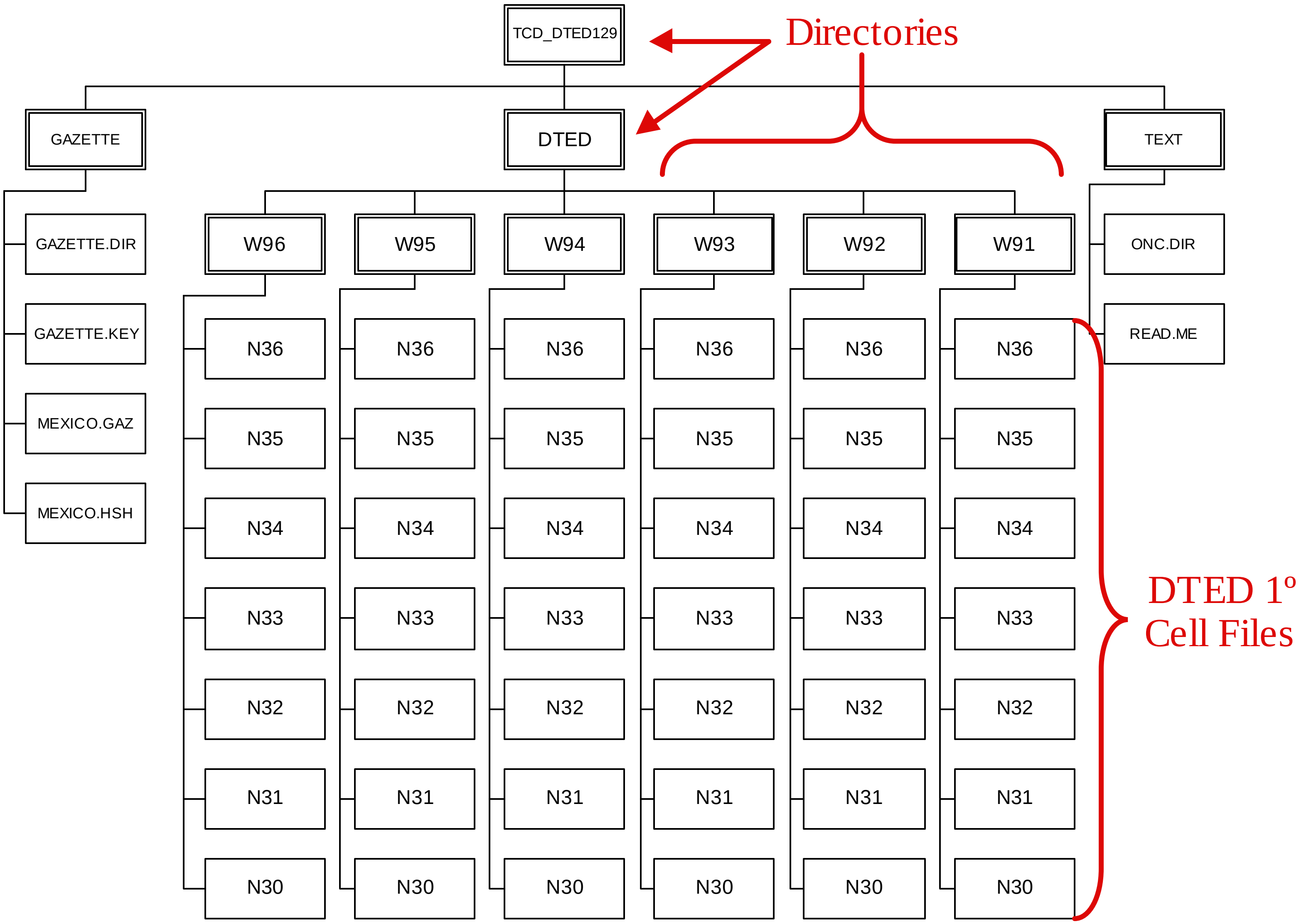
Gridded Data Example – DTED



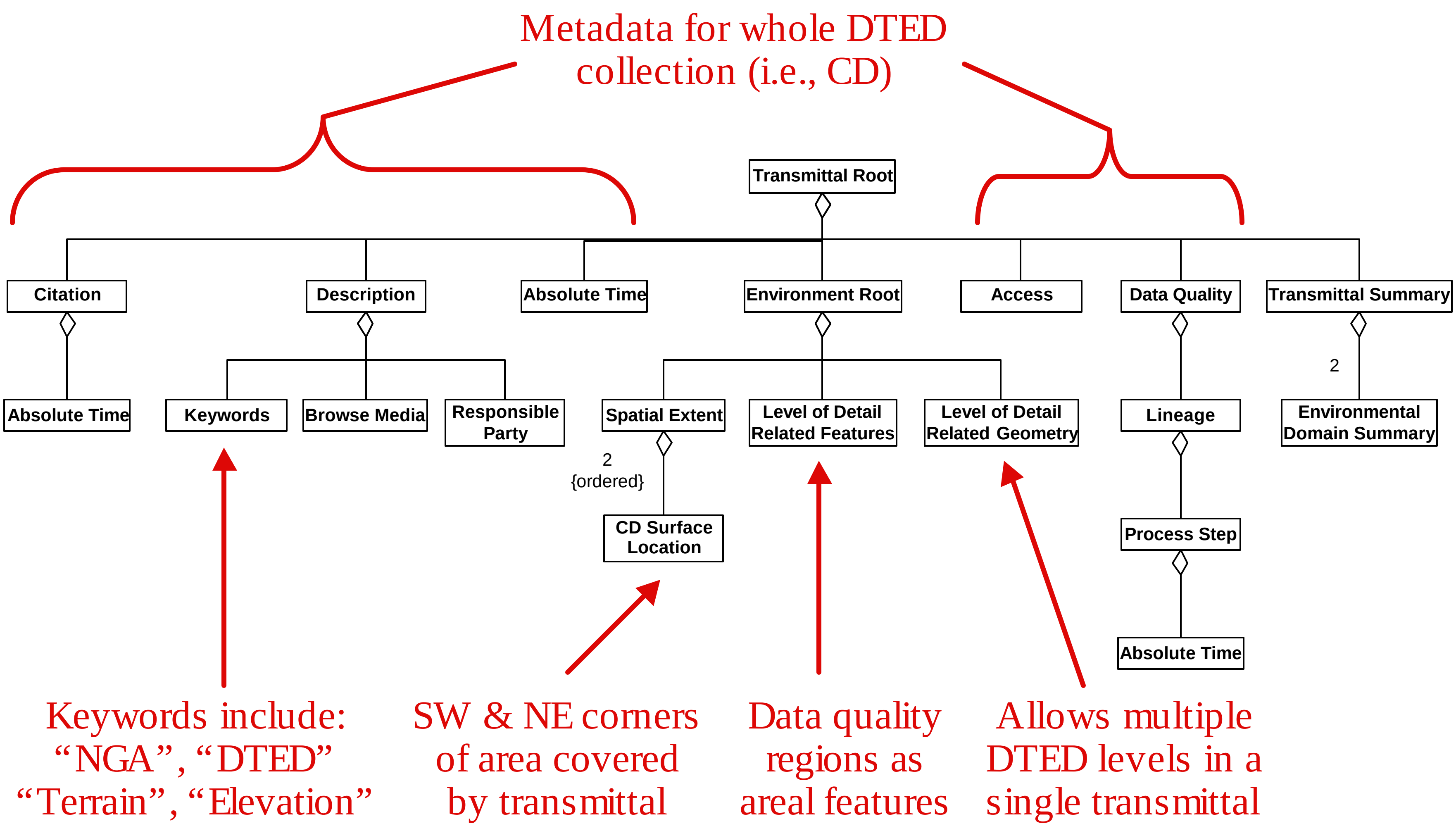
Gridded Data in the DRM – Property Grids



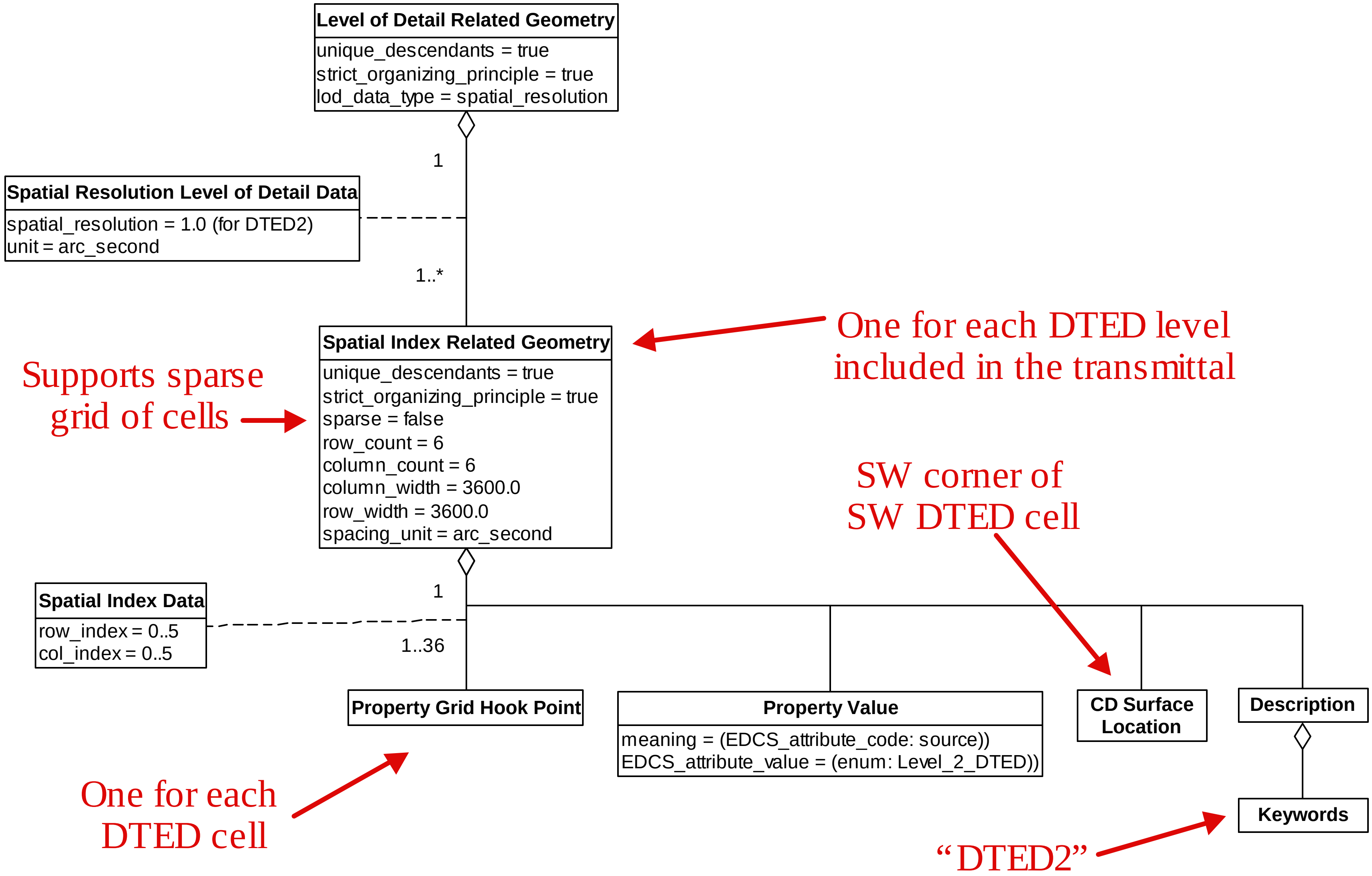
DTED CD-ROM Directory Tree



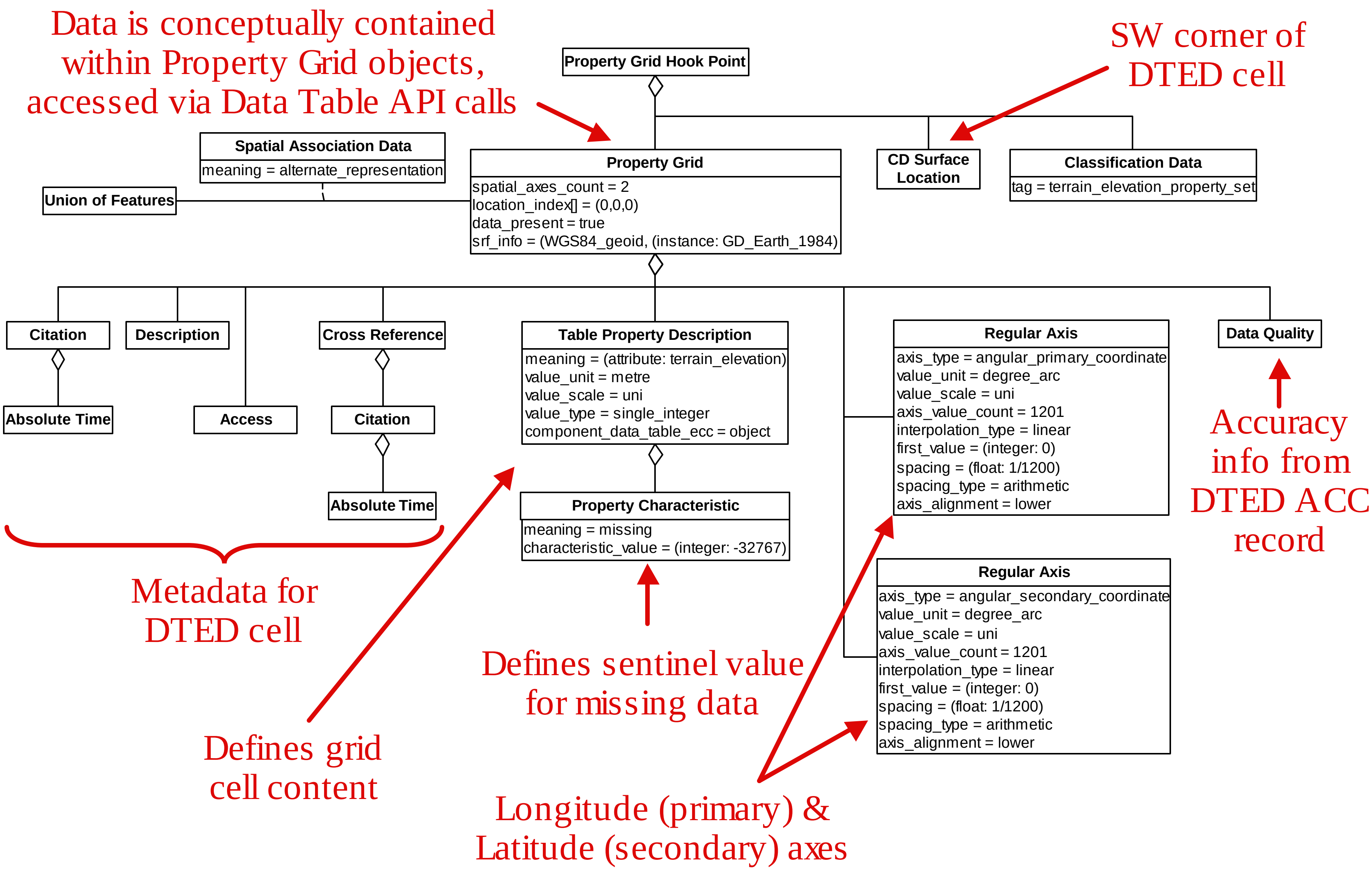
DTED-based Transmittal – Transmittal Root



DTED-based Transmittal – Geometry Hierarchy



DTED-based Transmittal – Property Grid



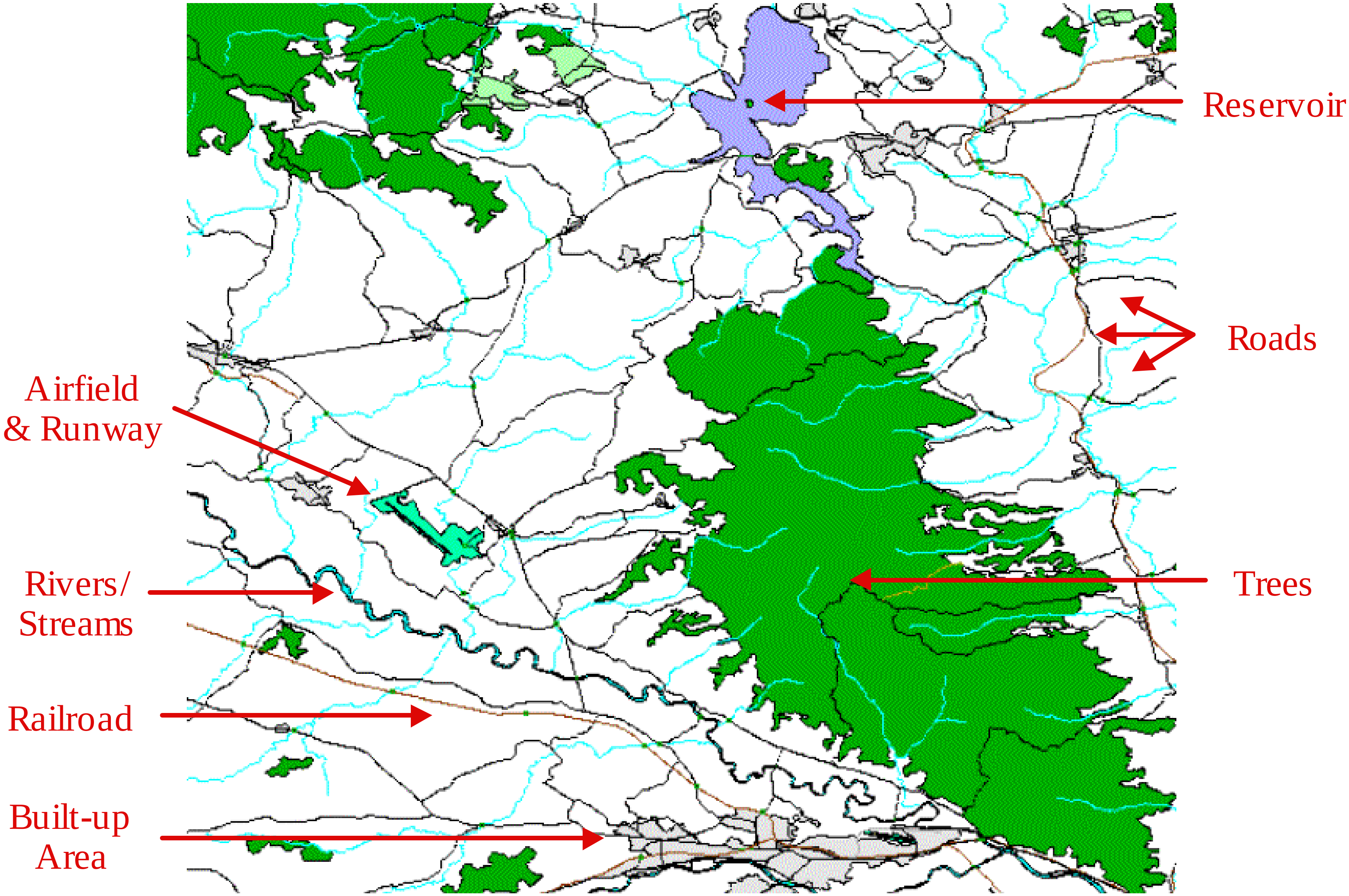
Gridded Data Summary

- Gridded Data is stored in Property Grid objects
- Gridded Data content (e.g., elevation values) is accessed using specific Data Table-related functions in the Read and Write APIs
- Property Grids are Data Tables with at least one spatial Axis
  - The Axis objects define the Property Grid structure
  - The Table Property Description object defines the Property Grid content
  - The Property Grid Hook Point object defines where the Property Grid is located
- All Property Grid objects in a transmittal are contained in the Data Table Library
  - Property Grid objects can also be attached to the geometry hierarchy under the Environment Root
- Property Grid objects can be associated with Features
- Each Property Grid object can have a complete set of supporting metadata
- DTED-to-STF 4.0 software will be available shortly at http://tools.sedris.org/

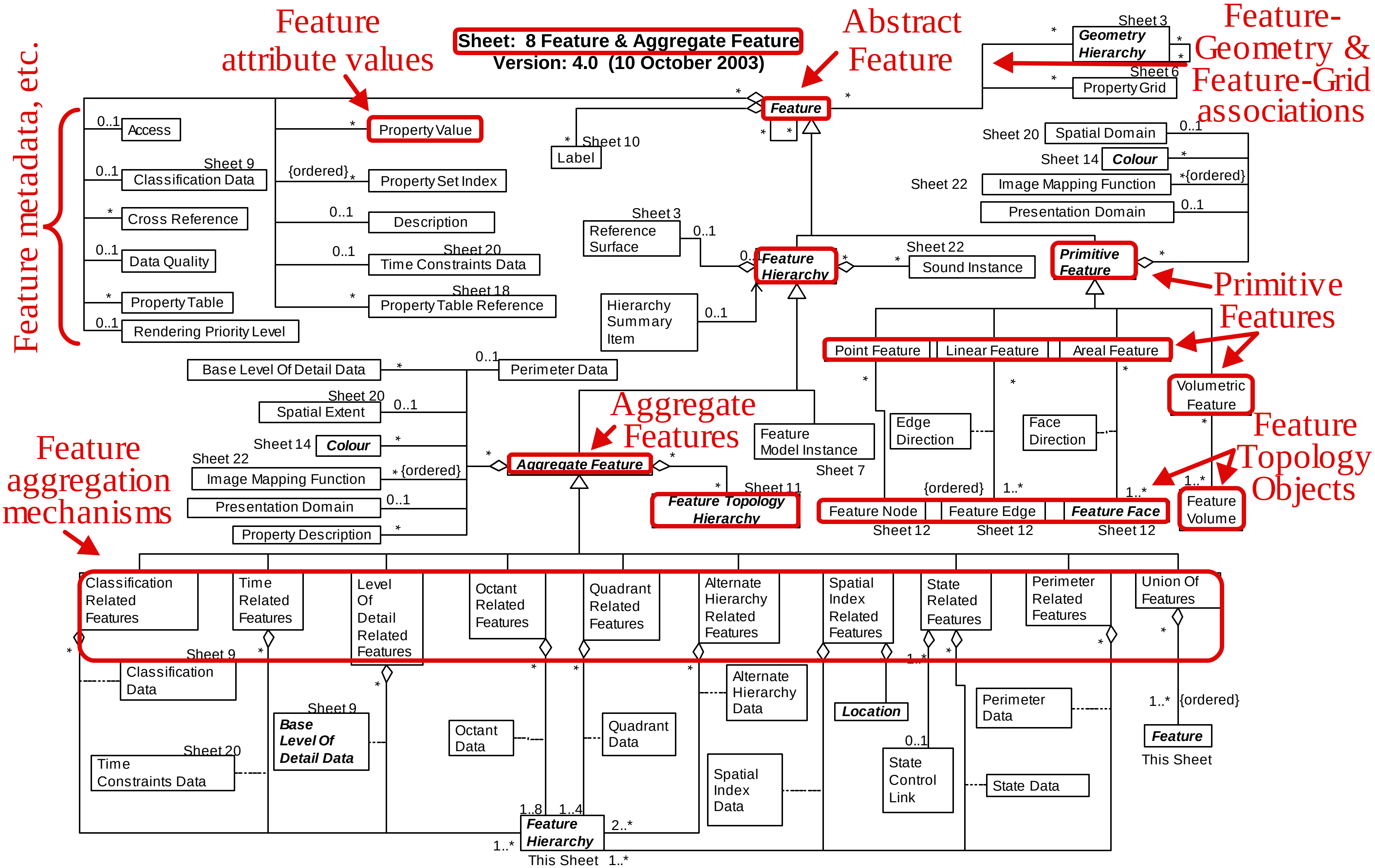
Vector Data

- Vector Data – a collection of abstract features (point, linear, areal)
  - Spot elevations & contour lines
  - Road, rail, drainage, & utility networks
  - Vegetation and soil classification
  - Cultural features – buildings, industrial facilities, transportation facilities, etc.
- Vector Data Sources
  - NGA's Vector Product Format (VPF)
    - VMAP/UVMAP – features extracted from scanned maps
    - VITD – features extracted from 1:50,000 TLMs & overlays
    - FFD/MSDS/DTOP – features extracted from stereo imagery
  - USGS Digital Line Graphs (DLGs)
- Topology is a mandatory component of vector data in SEDRIS
  - Location information is stored within the feature topology objects
  - Topological relationships support efficient terrain reasoning computations (e.g., route finding), and helps to maintain the consistency of vector data

Vector Data Example – FFD



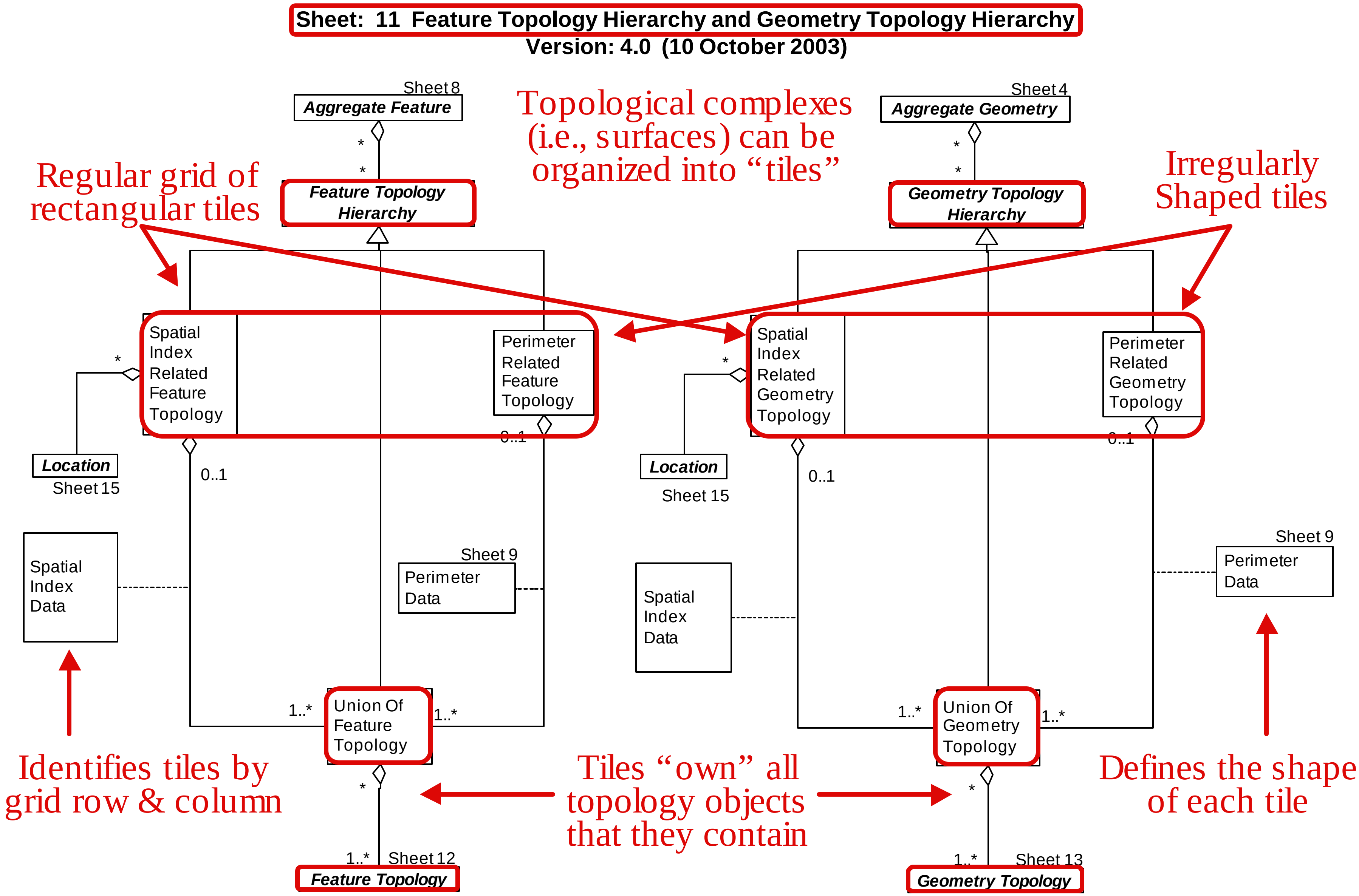
Vector Data in the DRM – Features



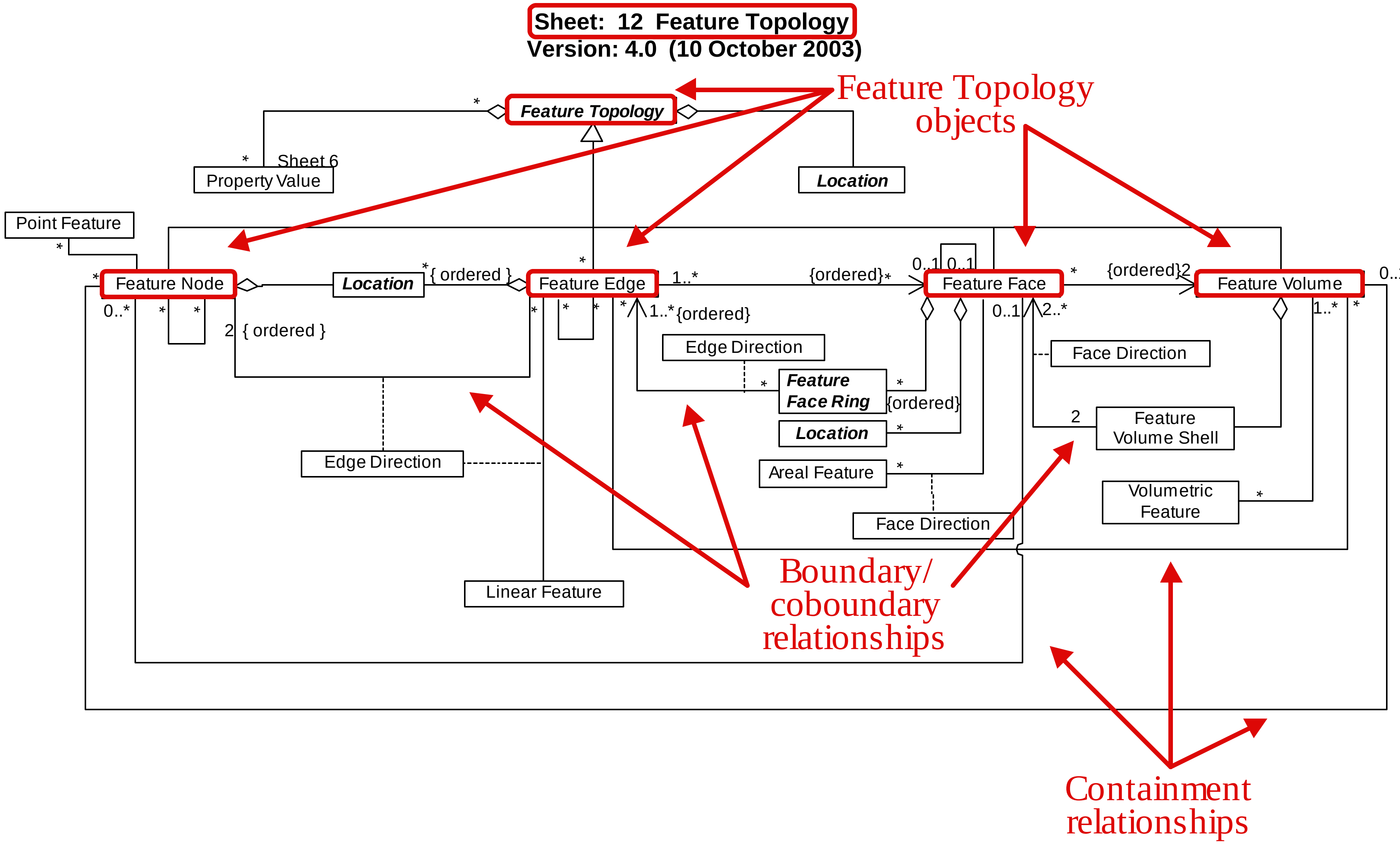
Topology

- Explicitly stores boundary, co-boundary, and containment relationships
  - Allows higher-level adjacency and connectedness relationships to be efficiently derived
- Advantages:
  - Eliminates redundant storage of coordinates
  - Accelerates geometric calculations by eliminating extensive searching and coordinate comparisons
  - Provides "glue" that ties primitives together and helps ensure consistency
- Disadvantages:
  - Requires a modest amount of additional storage space
  - Topological errors can disrupt algorithms that rely on topological relationships
- The SEDRIS DRM currently supports 2D topology, with two extensions that provide limited 3D support:
  - The Edges connected to a node can be organized into an ordered collection of ordered groups
  - An Edge can bound zero or more faces

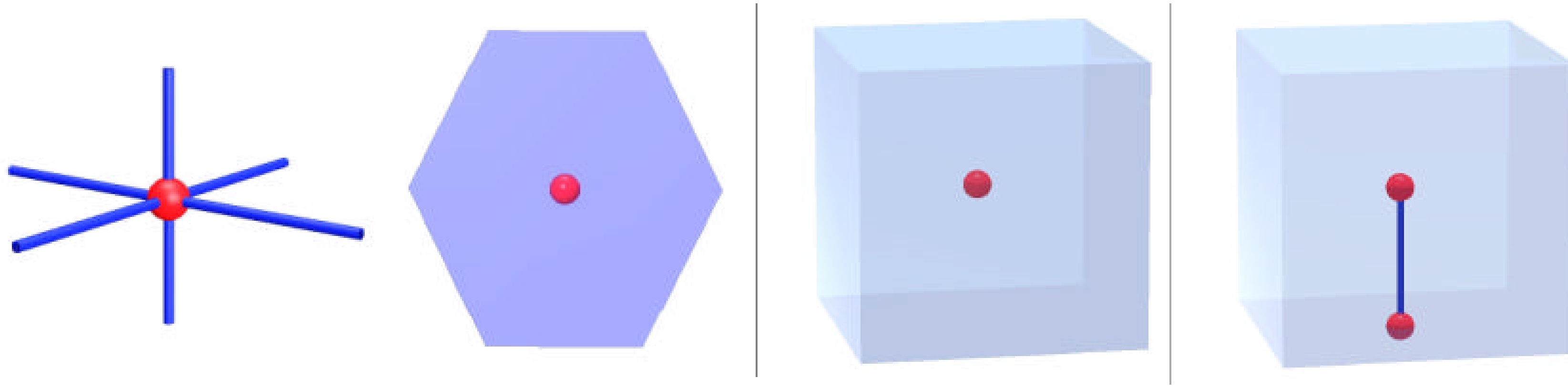
Topology Hierarchies



Feature Topology



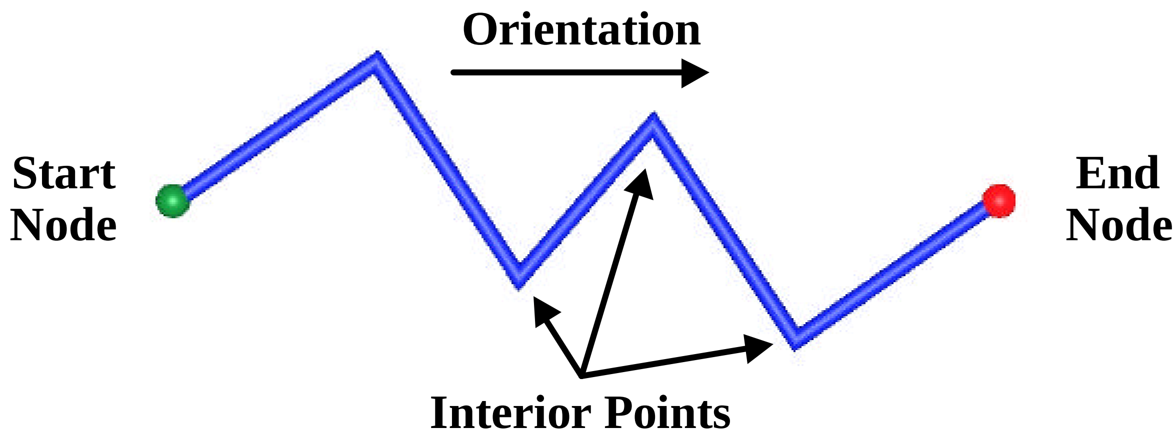
Feature Nodes



Connected to Edges | **Contained within a Face** | **Contained within a Volume** | **Multiple Relationships**

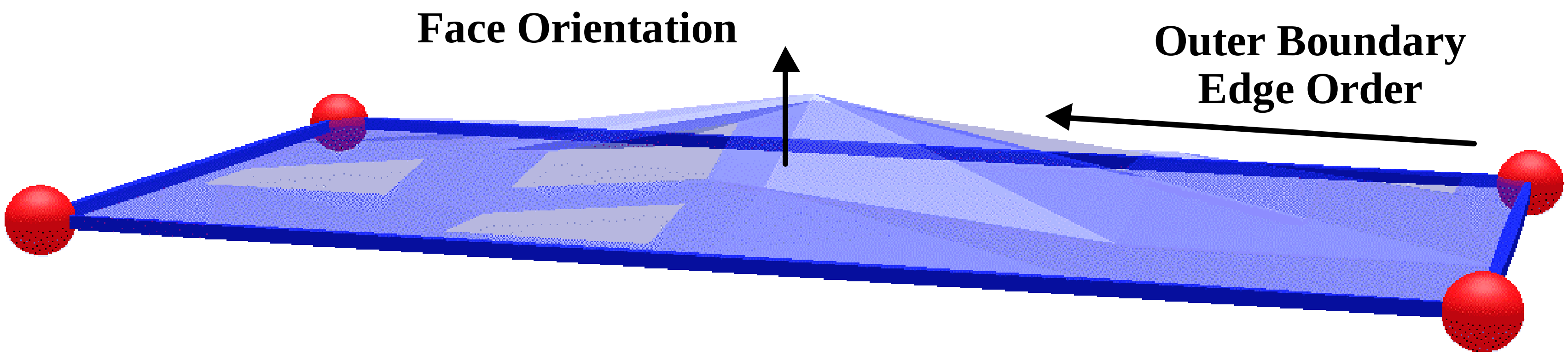
- Feature Node – a zero-dimensional spatial entity that defines a single location in 2D or 3D space
  - Location of each node must be unique – multiple nodes cannot be co-located (Exception: Feature Topology Level 0)
- A node cannot be located in the interior of an edge (Exception: Feature Topology Level 0)
- A node may be located within the interior of a face or a volume

Feature Edges



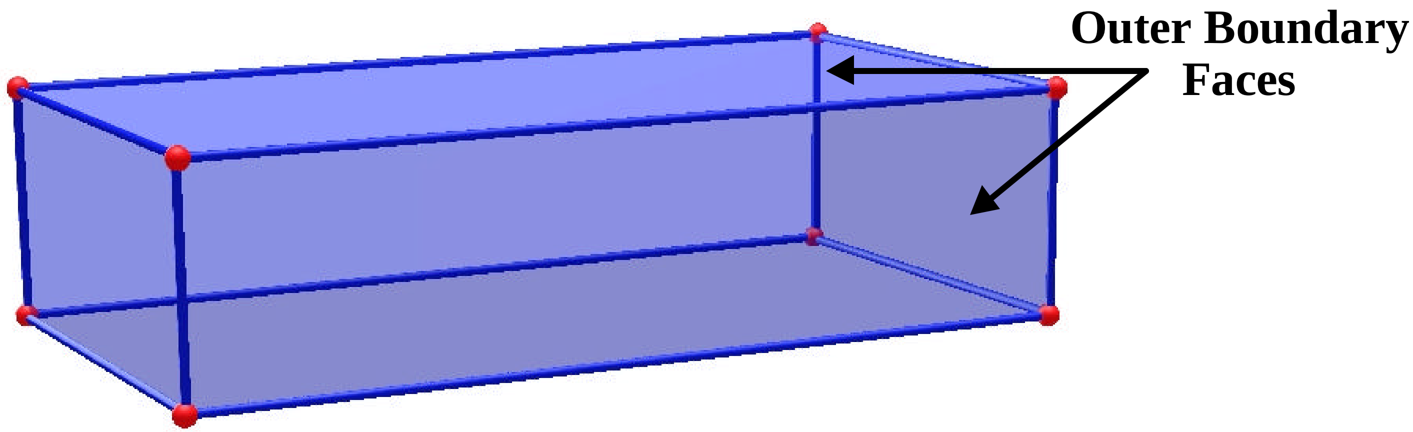
- Feature Edge – a one-dimensional spatial entity that defines a path through 2D or 3D space
  - The shape of an edge is defined by a sequence of two or more distinct 2D or 3D coordinate tuples
  - The orientation of an edge is defined by the order of its coordinate tuples
- An edge is bounded by a node at each of its two endpoints (conceptually, the endpoints are included in the edge, but they are not stored with it)
- Edges may meet only at common nodes
  - Edge interiors may not intersect one another (Exception: Feature Topology Levels 0 & 1)
- An edge may be completely contained within a volume

Feature Faces



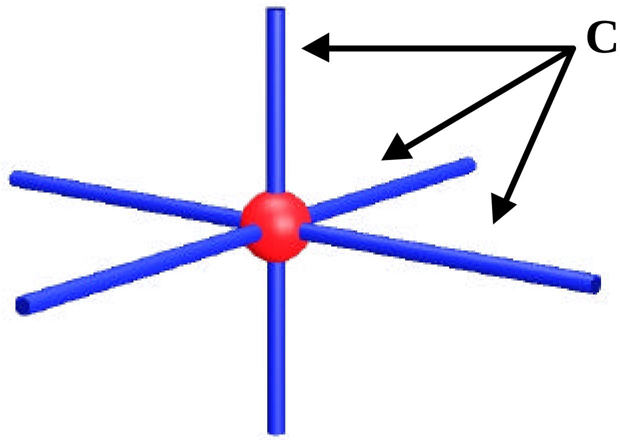
- Feature Face – a two-dimensional spatial entity that defines a closed area in 2D or 3D space
  - The shape of a face is defined by the edges that bound the face, as well as any nodes contained within the interior of the face – in 3D, a face need not be planar
  - The orientation of a face is defined the order of the edges in its outer boundary (counterclockwise as seen from above)
- A face is bounded by one or more ordered collections of edges, defining its outer boundary, and zero or more inner boundaries (i.e., "holes")
- Faces may meet only along common edges
  - Face interiors may not intersect one another (Exception: Feature Topology Levels 0 through 2)

Feature Volumes



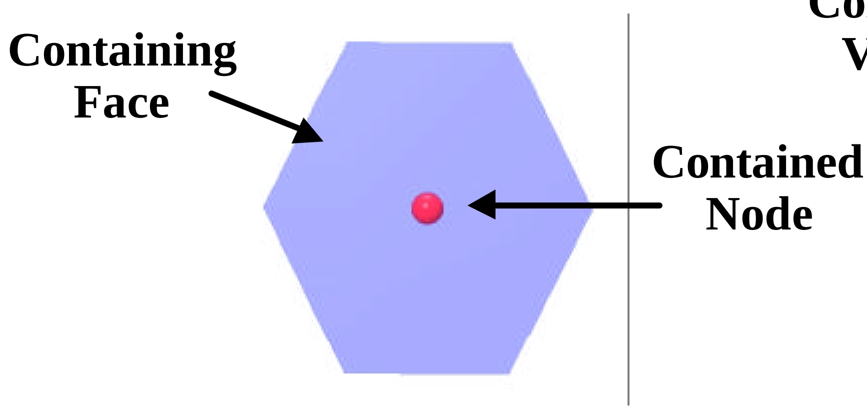
- Feature Volume – a three-dimensional spatial entity that defines a closed region of 3D space
  - The shape of a volume is defined by the faces that bound the volume
- A volume is bounded by one or more (unordered) collections of faces, defining an outer boundary and zero or more inner boundaries
- Volumes may meet only along common faces
  - Volume interiors may not intersect one another (Exception: Feature Topology Levels 0 through 4)

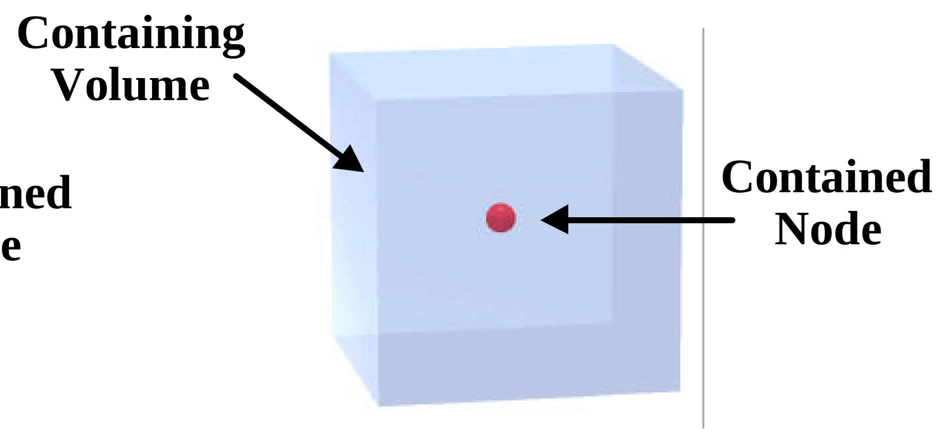
Node-Edge Relationships



- Each edge is associated with an ordered collection of two nodes: a <u>starting node</u> and an <u>ending node</u>
  - When an edge forms a loop, its starting and ending nodes are the same node
- Each node is associated with an unordered collection of zero or more <u>connected edges</u>
  - In 2D, the connected edges are ordered counterclockwise around the node (starting with an arbitrary edge)
  - In 3D, the connected edges cannot, in general, be ordered, since they can connect to the node from any direction in 3D space

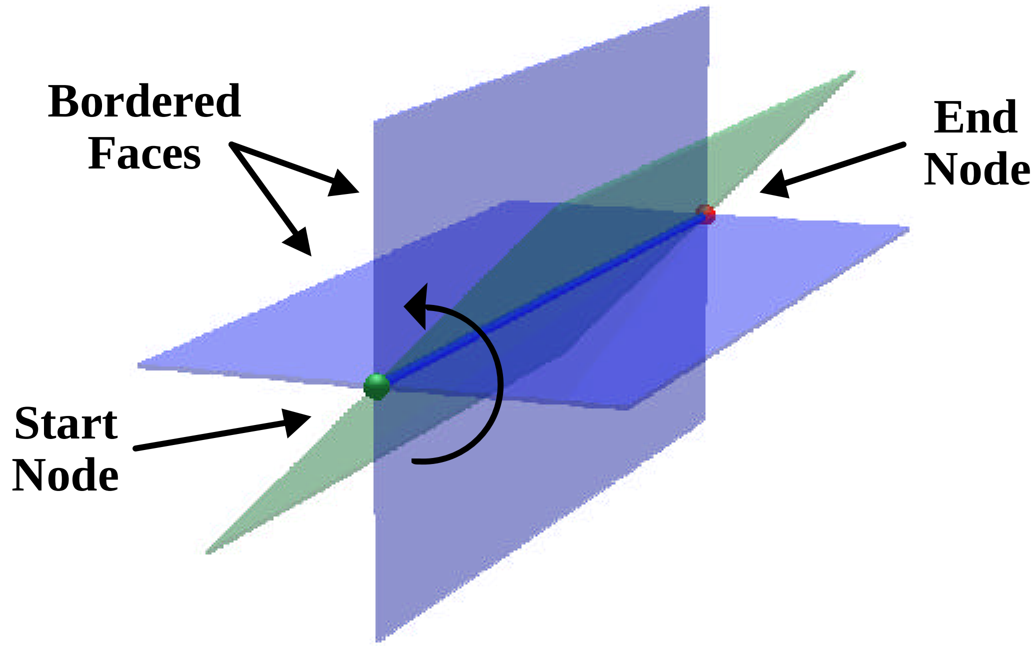
Node-Face Relationships





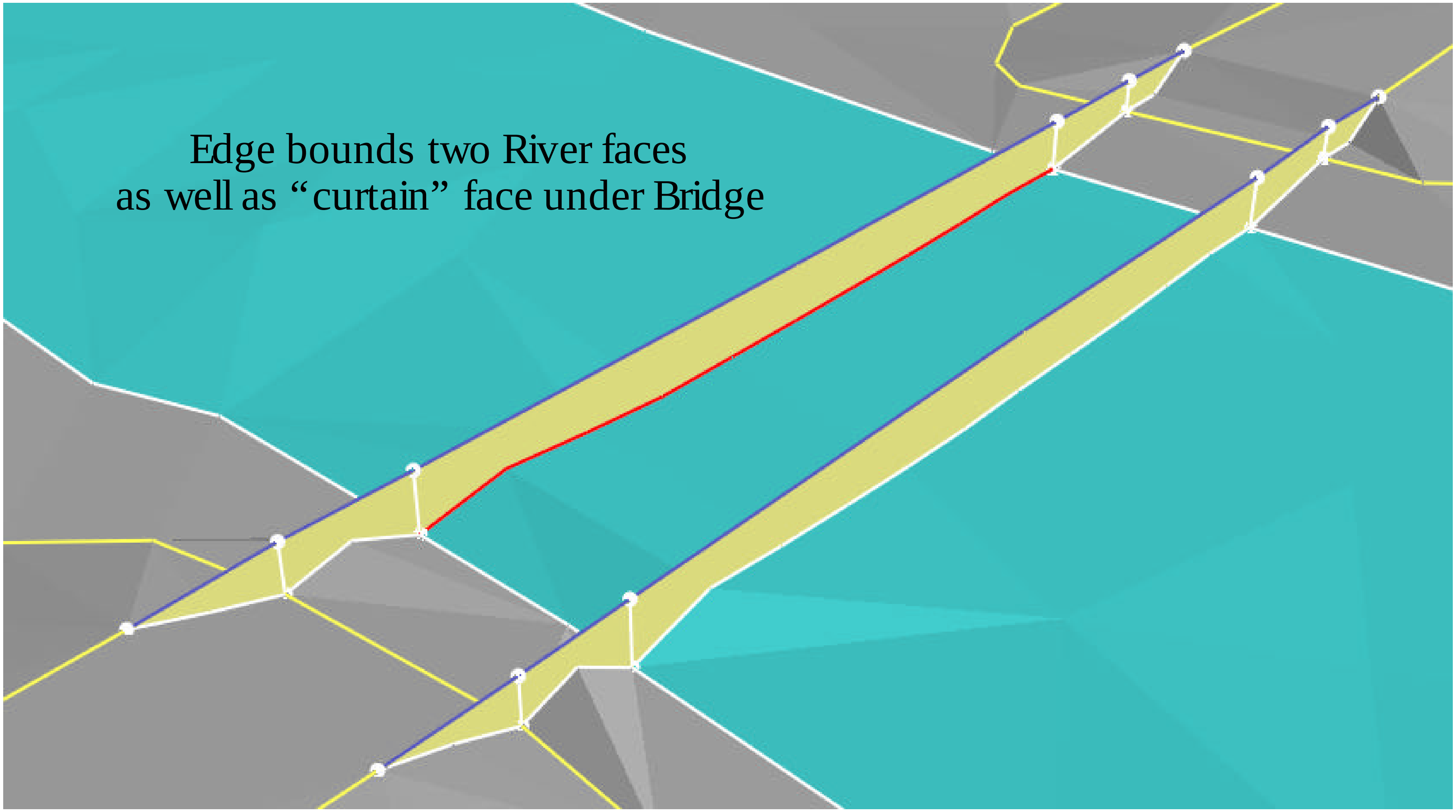
- Node-Face Relationships
  - Each face is associated with a collection of zero or more contained nodes
  - Each node may be associated with zero or more containing faces (Note that multiple faces can meet at a common node that is not part of the boundary of any of the faces)
- Node-Volume Relationships
  - Each volume is associated with a collection of zero or more contained nodes
  - Each node may be associated with zero or one containing volume

Edge-Face Relationships

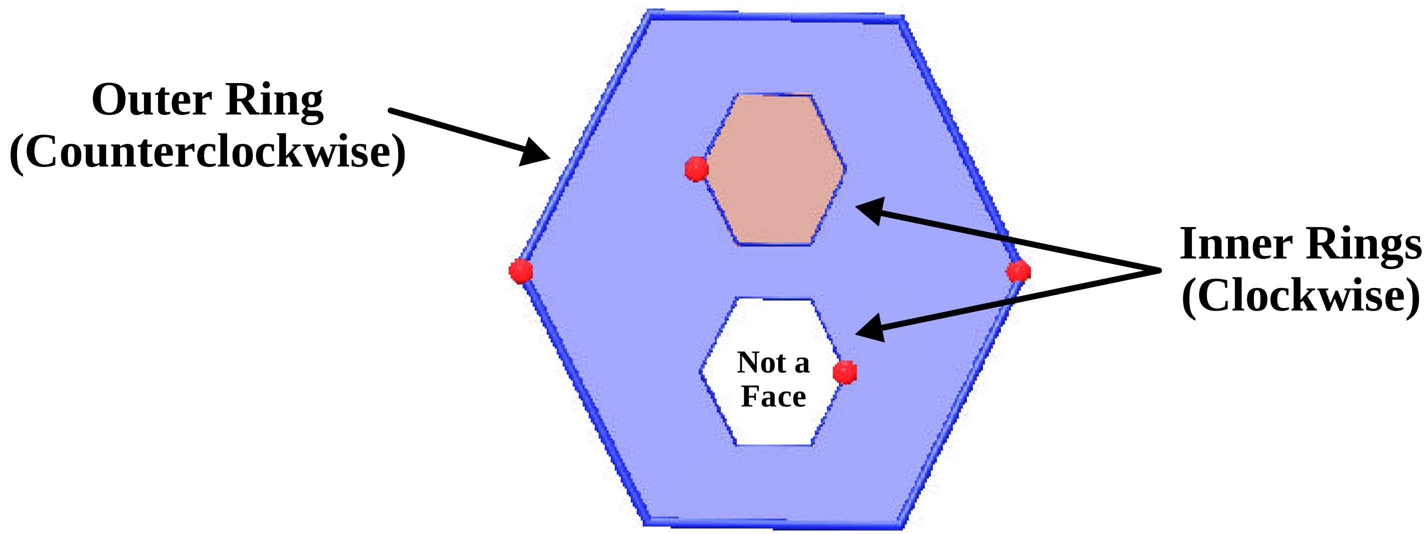


- In 2D, each edge is associated with with exactly two faces: its left face and its right face
- In 3D, each edge is associated with a collection of zero or more bordered faces, ordered counterclockwise relative to the edge, starting with an arbitrary face

Example: "Curtain" Face under a Bridge

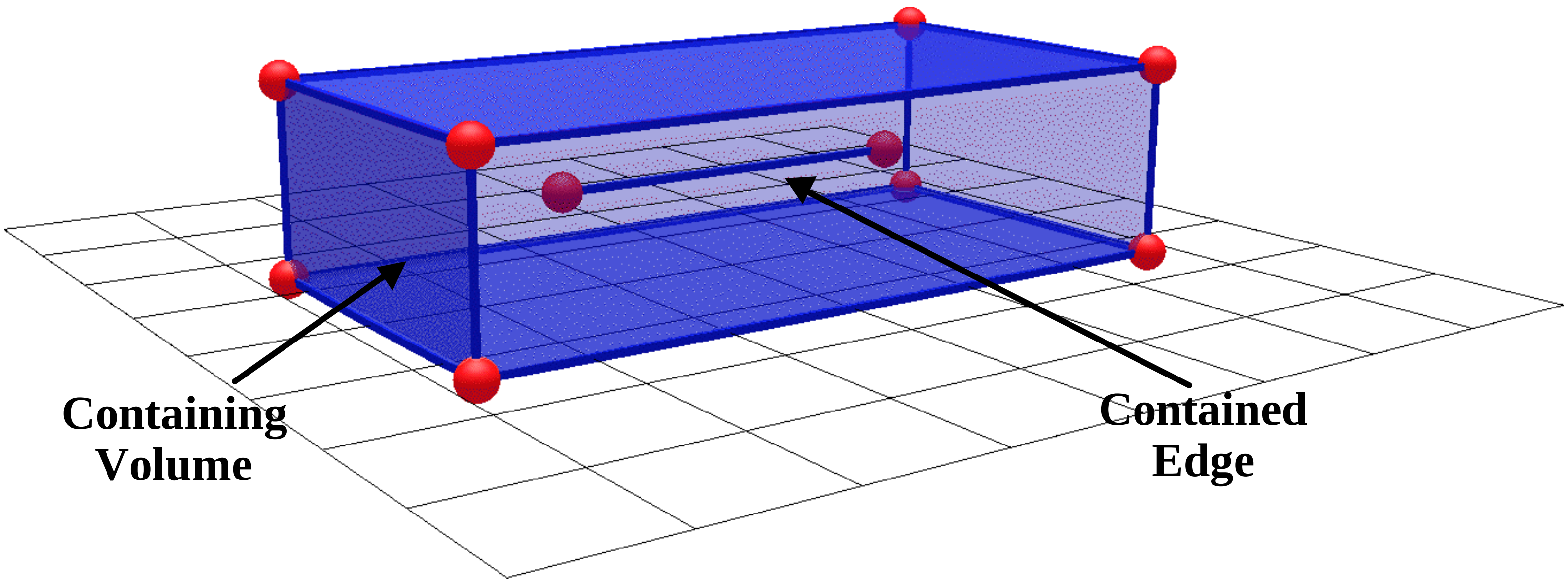


Face-Edge Relationships – Rings



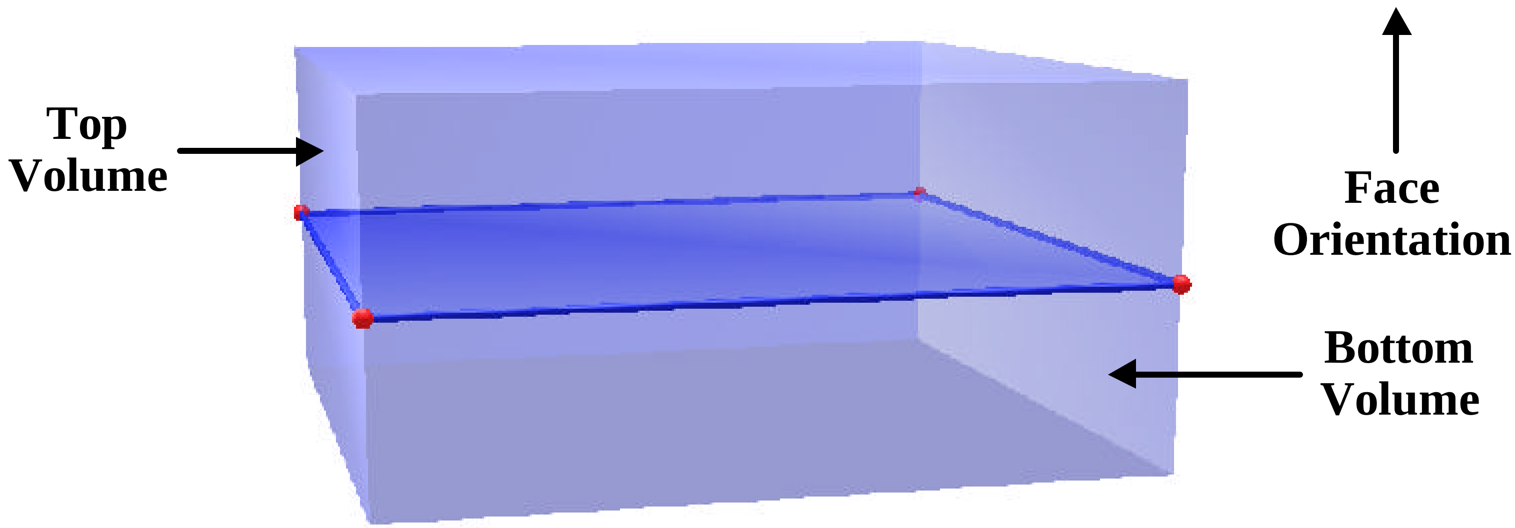
- Ring – a sequentially connected set of edges that bound a face
  - Outer Ring (counterclockwise) – defines the outer boundary of a face
    - Includes any edges within the face, but connected to the outer boundary
  - Inner Ring (clockwise) – defines an inner boundary of a face (i.e., a "hole")
    - A collection of edges that are connected to one another, but not connected to the outer boundary, form an inner ring, even if they do not enclose an area
    - An inner ring need not contain any faces - it may represent a real hole
- An edge can appear twice in the same ring, once in each orientation

Edge-Volume Relationships



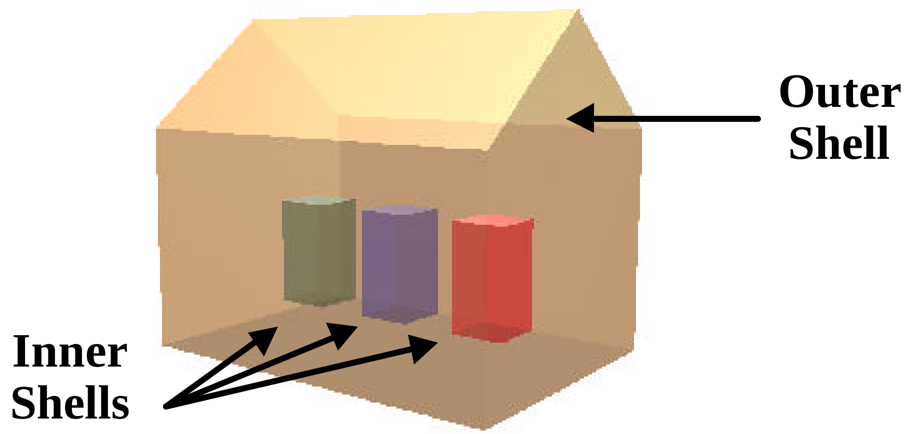
- Edge-Volume Relationships
  - Each volume is associated with a collection of zero or more contained edges
  - Each edge may be associated with one containing volume

Face-Volume Relationships



- In 3D, each face is associated with exactly two volumes: its upper volume and its lower volume
  - The upper and lower volumes of a face may be the same volume

Volume-Face Relationships – Shells

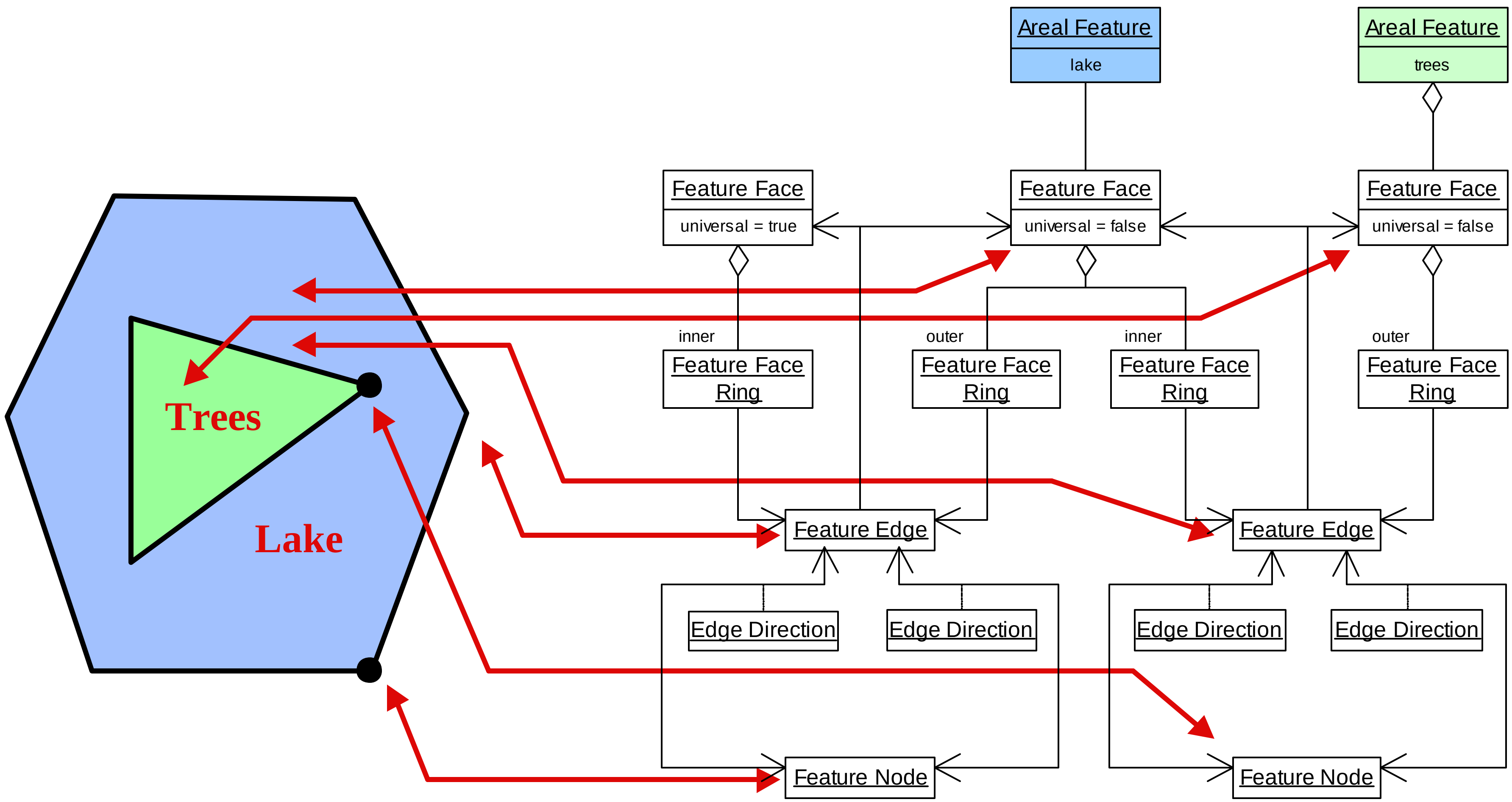


- Shell – an unordered collection of two or more faces that bound a volume
  – Outer Shell – defines the outer boundary of a volume
  – Inner Shells – define any inner boundaries of a volume (i.e., "bubbles" within the volume)
    – A collection of one or more faces that are connected to one another, but that are not connected to the outer boundary of the volume, form an inner shell even if they do not enclose a space
- A face can appear twice in the same shell, once in each orientation

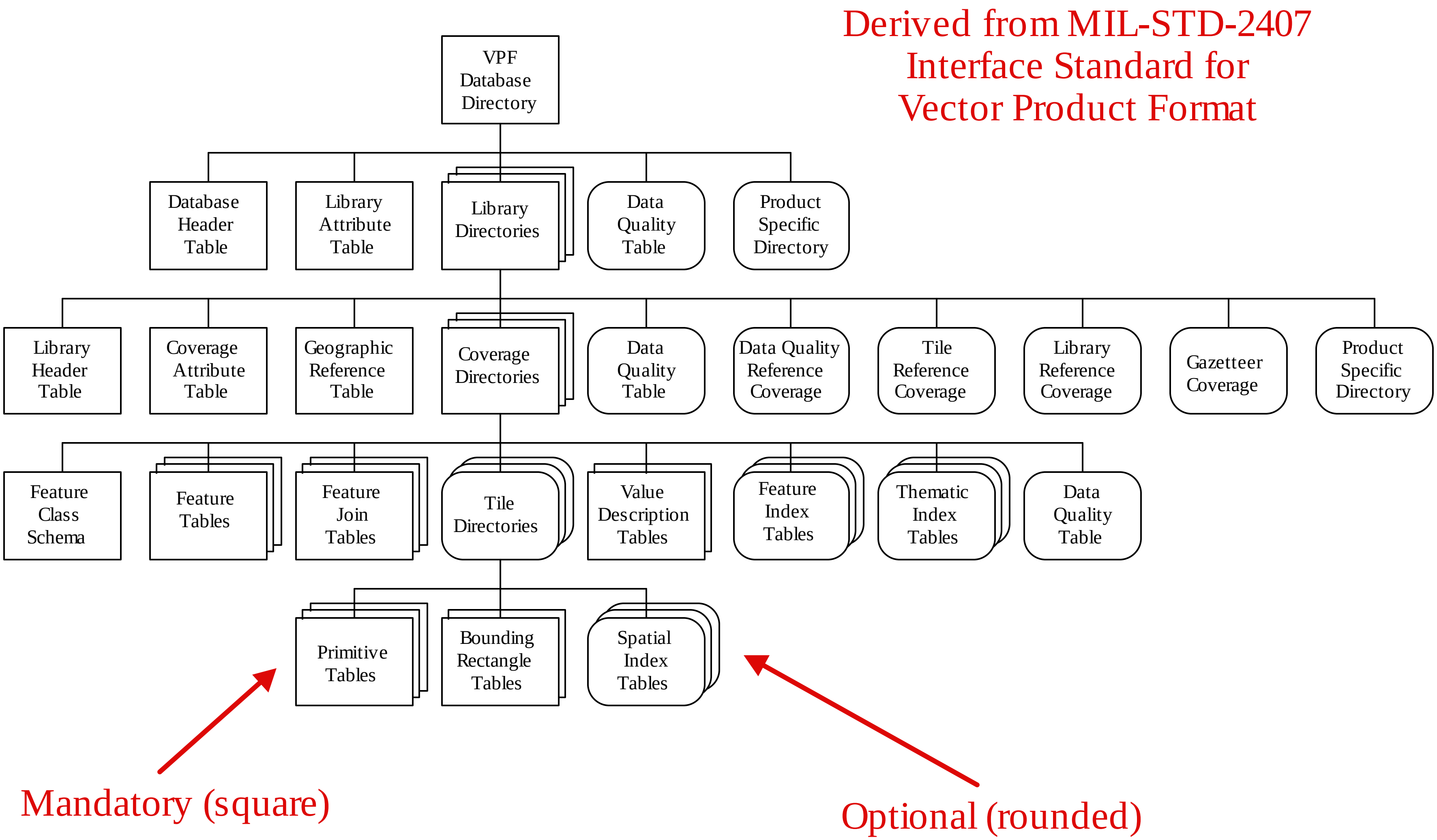
Feature Topology Levels

- Level 0 (2D or 3D coordinates)
  - One or more pairs of Feature Nodes are collocated
- Level 1 (2D or 3D coordinates)
  - No Feature Nodes are collocated
  - One or more pairs of Feature Edges have intersecting interiors
- Level 2 (2D or 3D coordinates)
  - No Feature Edges have intersecting interiors
- Level 3 (2D or 3D coordinates)
  - 2D topological surface completely partitioned into non-overlapping faces
- Level 4 (3D coordinates)
  - Edges and perhaps faces "sticking out" of 2D topological surface
- Level 5 (3D coordinates)
  - 3D topological space completely partitioned into non-overlapping volumes

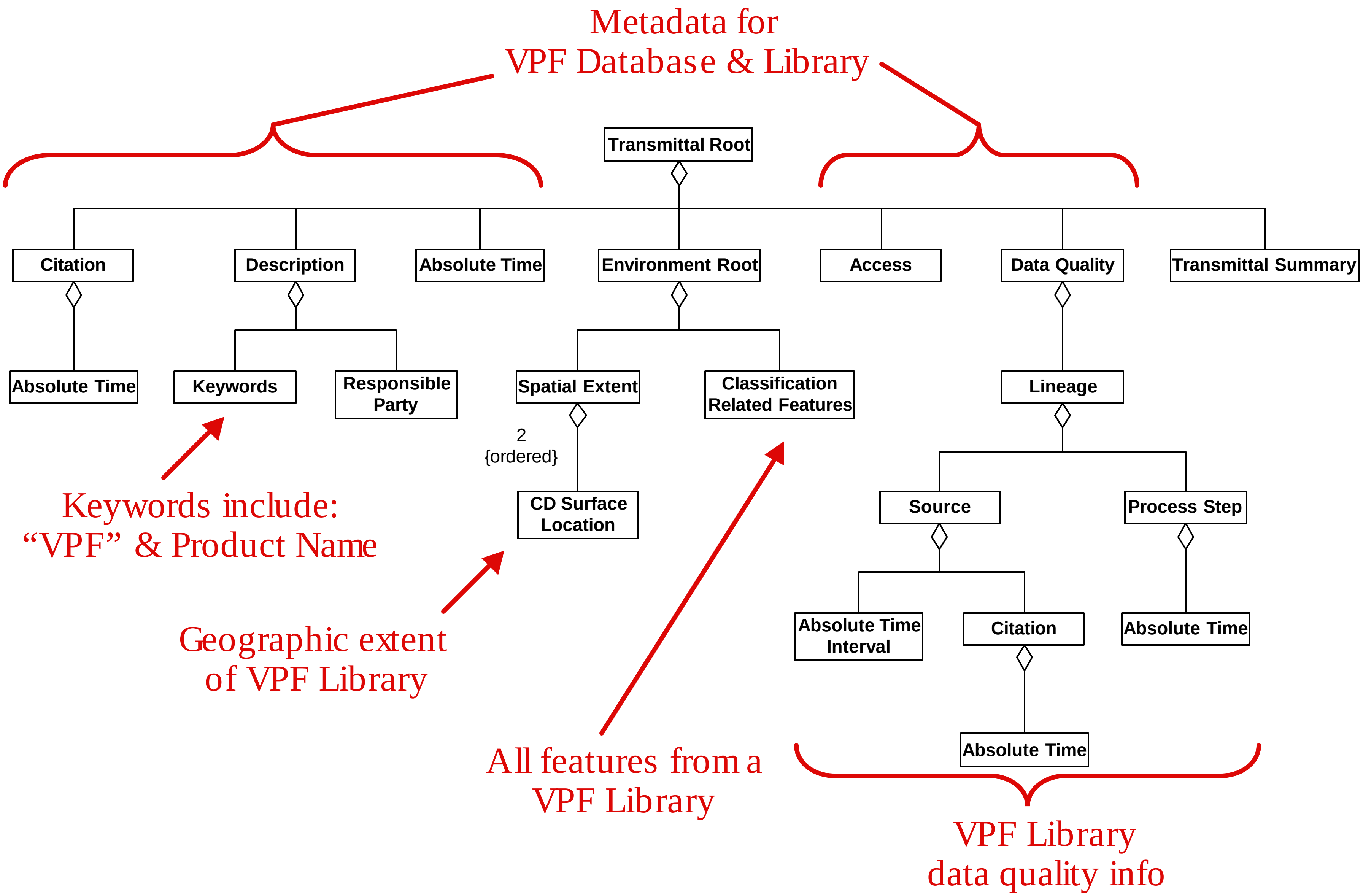
Feature Topology Example



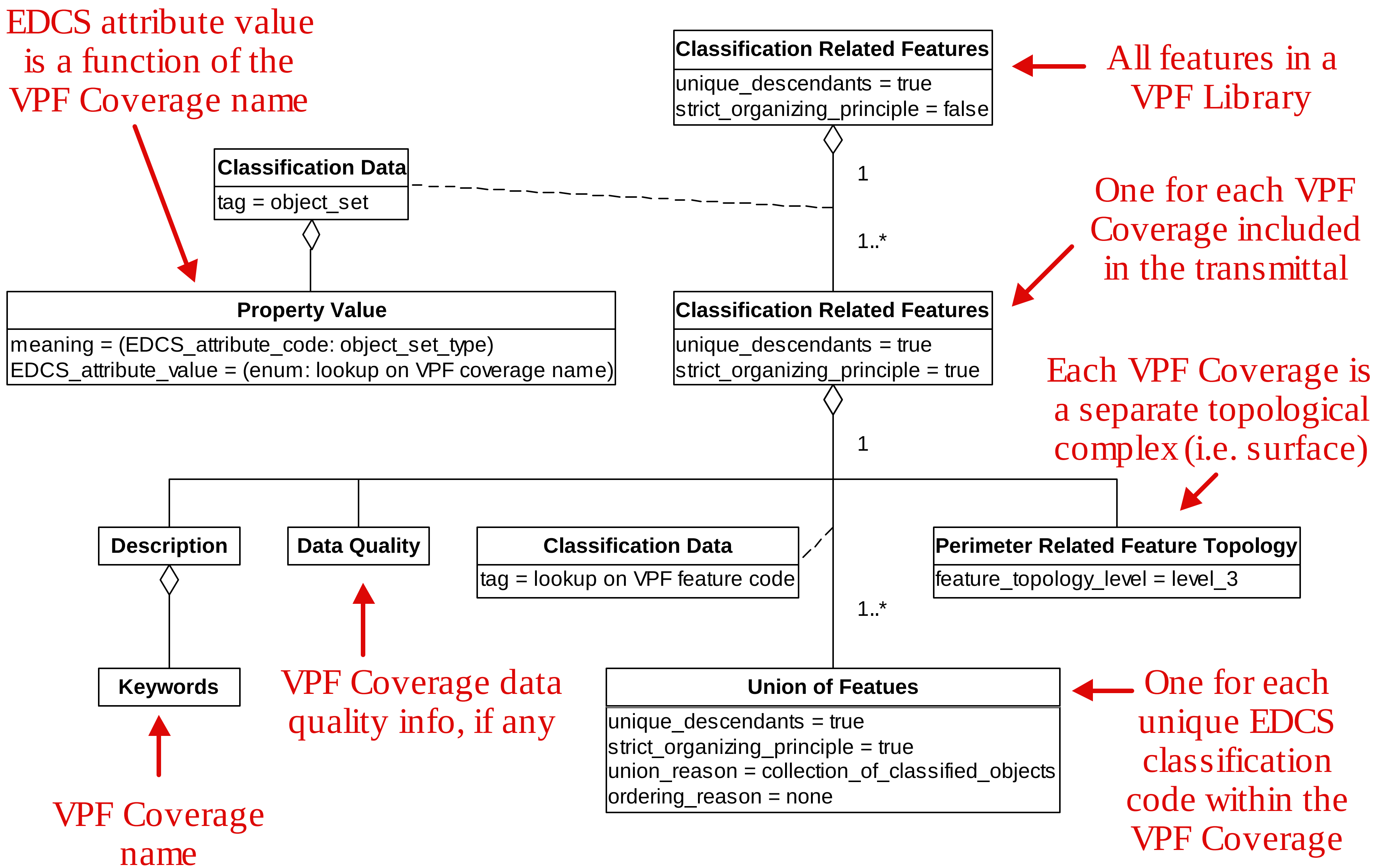
VPF Database Directory Tree



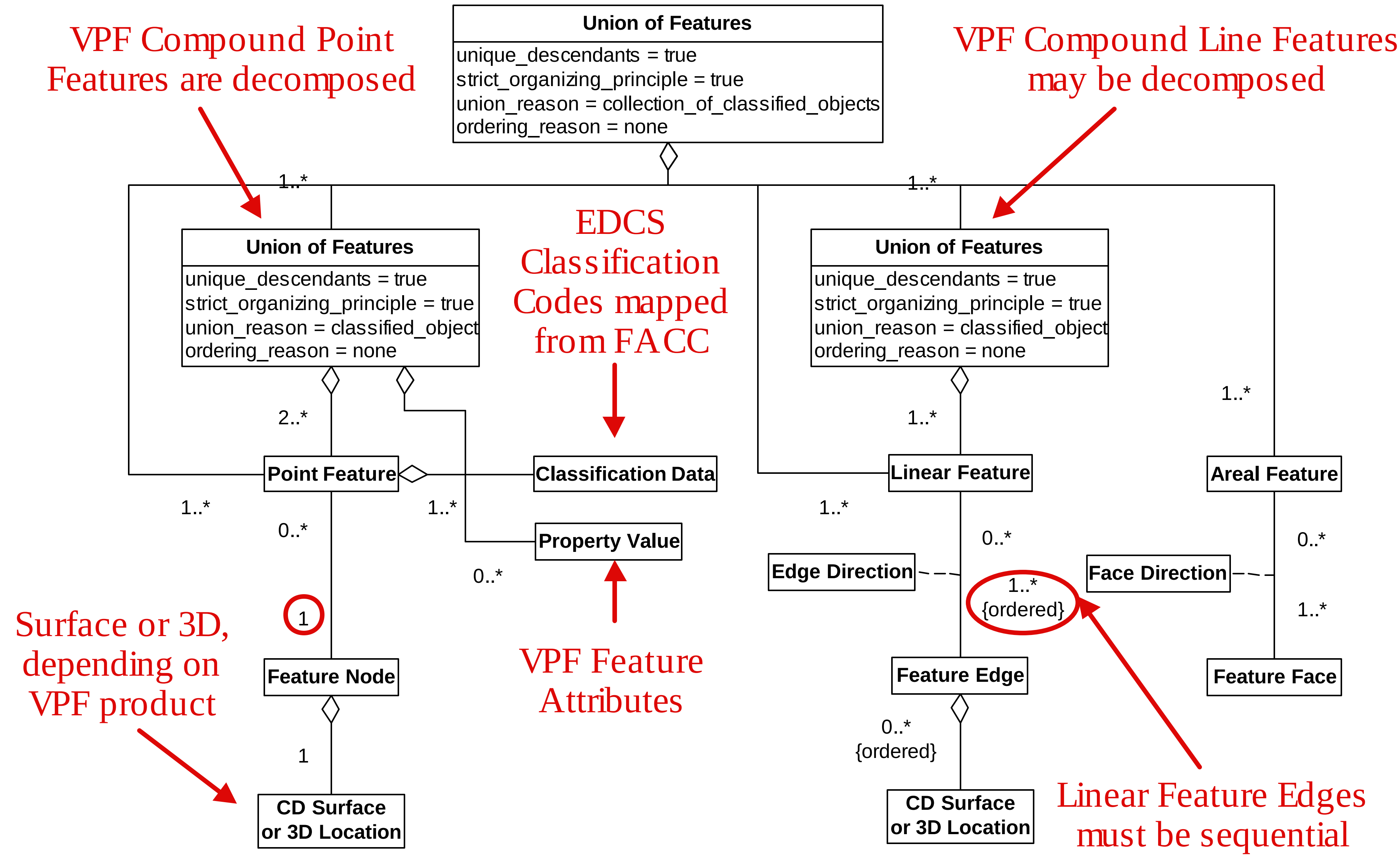
VPF-based Transmittal – Transmittal Root



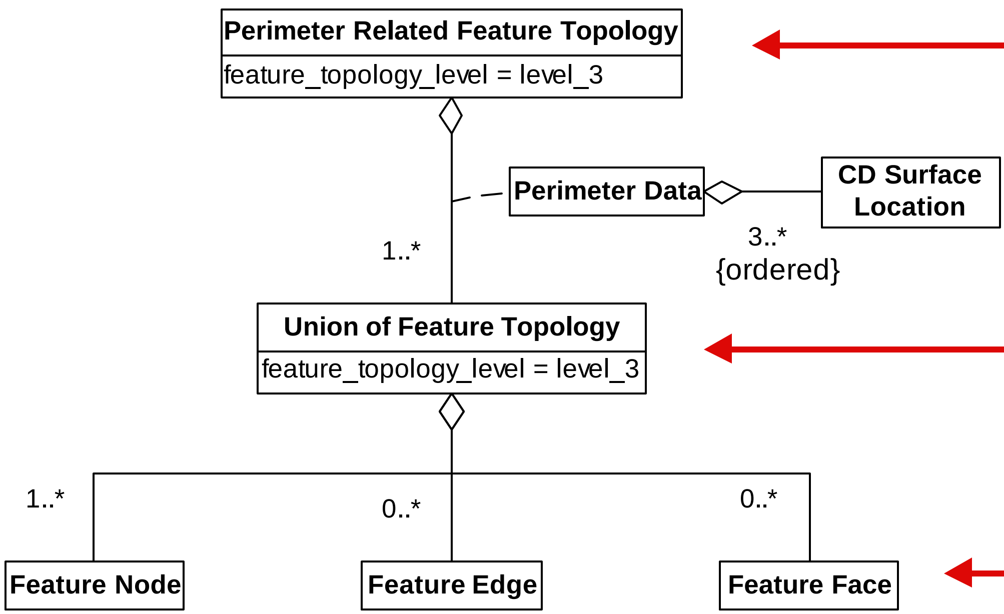
VPF-based Transmittal – Feature Hierarchy



VPF-based Transmittal – Primitive Features



VPF-Based Transmittal –Topology Hierarchy



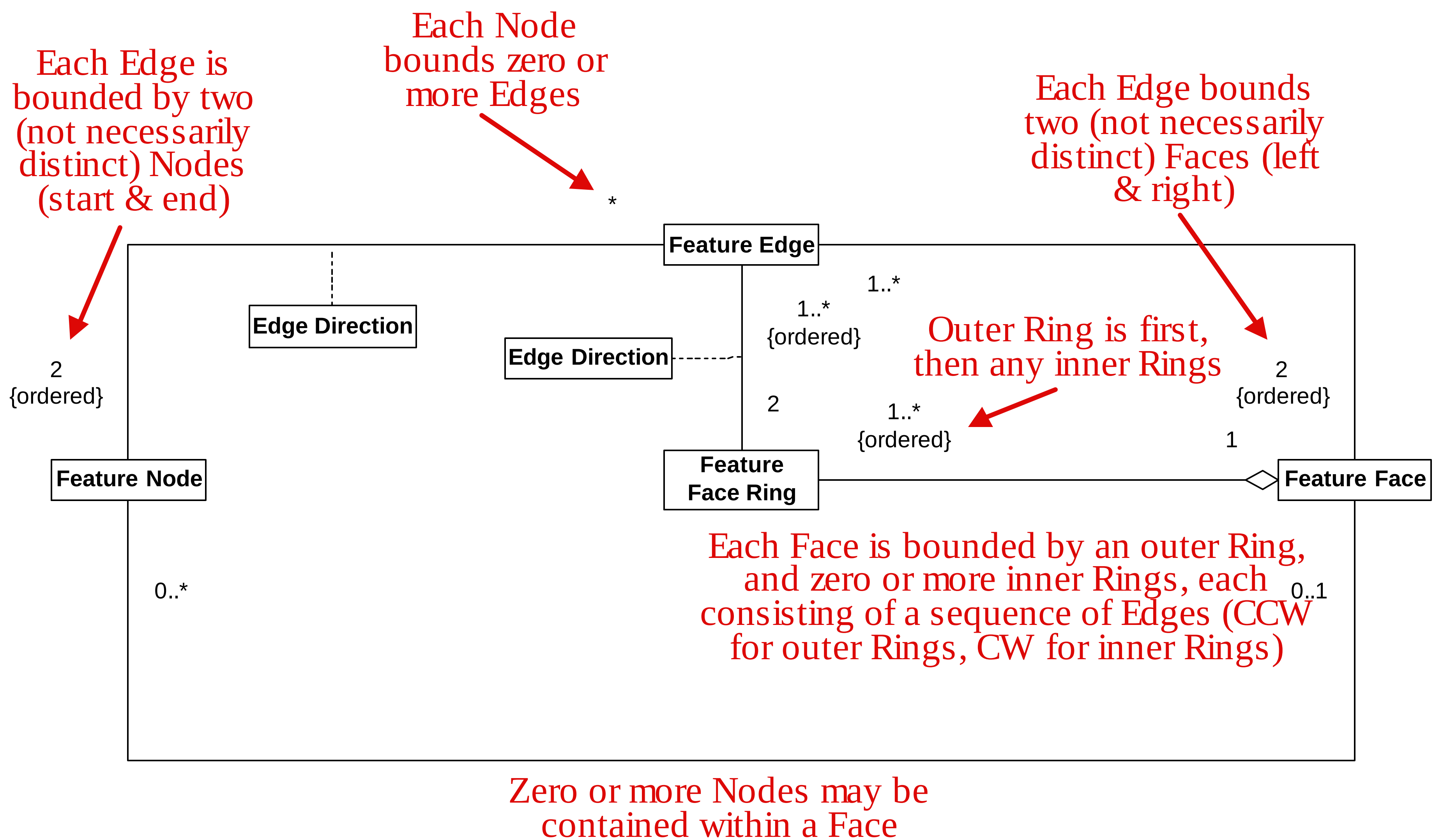
One for each VPF Coverage, each represents a separate topological complex

Sequence of locations describes the tile shape

One for each tile within a VPF Coverage

All Feature Topology objects contained within the tile

VPF-Based Transmittal – Feature Topology



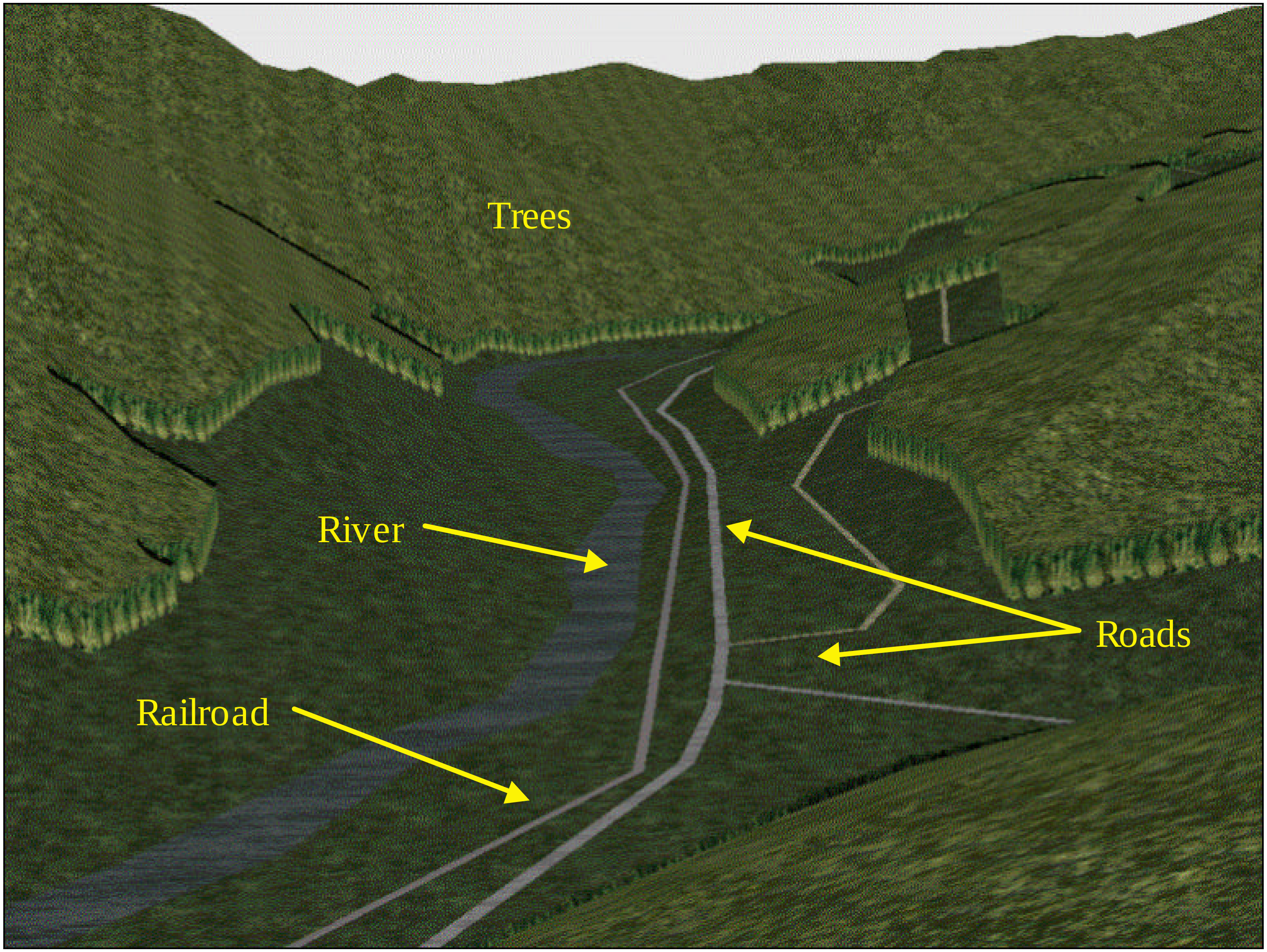
Vector Data Summary

- Vector Data is stored in Feature objects
- Primitive Features include Point, Linear, Areal, & Volumetric Features
- Features can be hierarchically organized in many different ways, including spatially, temporally, by classification, etc.
- Feature attributes can take many forms, including Property Values, Property Tables, etc., and can be associated with both primitive and aggregate Features
- Feature Topology objects contain both the geometric properties (based on location) and topological relationships of Features
  - Feature Topology objects are organized into topological complexes (i.e., tiles), represented by Feature Topology Hierarchy objects
- Feature objects can be associated with Images, Property Grids, Geometry, and/or other Features
- Each Feature object can have a complete set of supporting metadata
- VPF-to-STF 4.0 software will be available soon at http://tools.sedris.org/

Polygonal Data

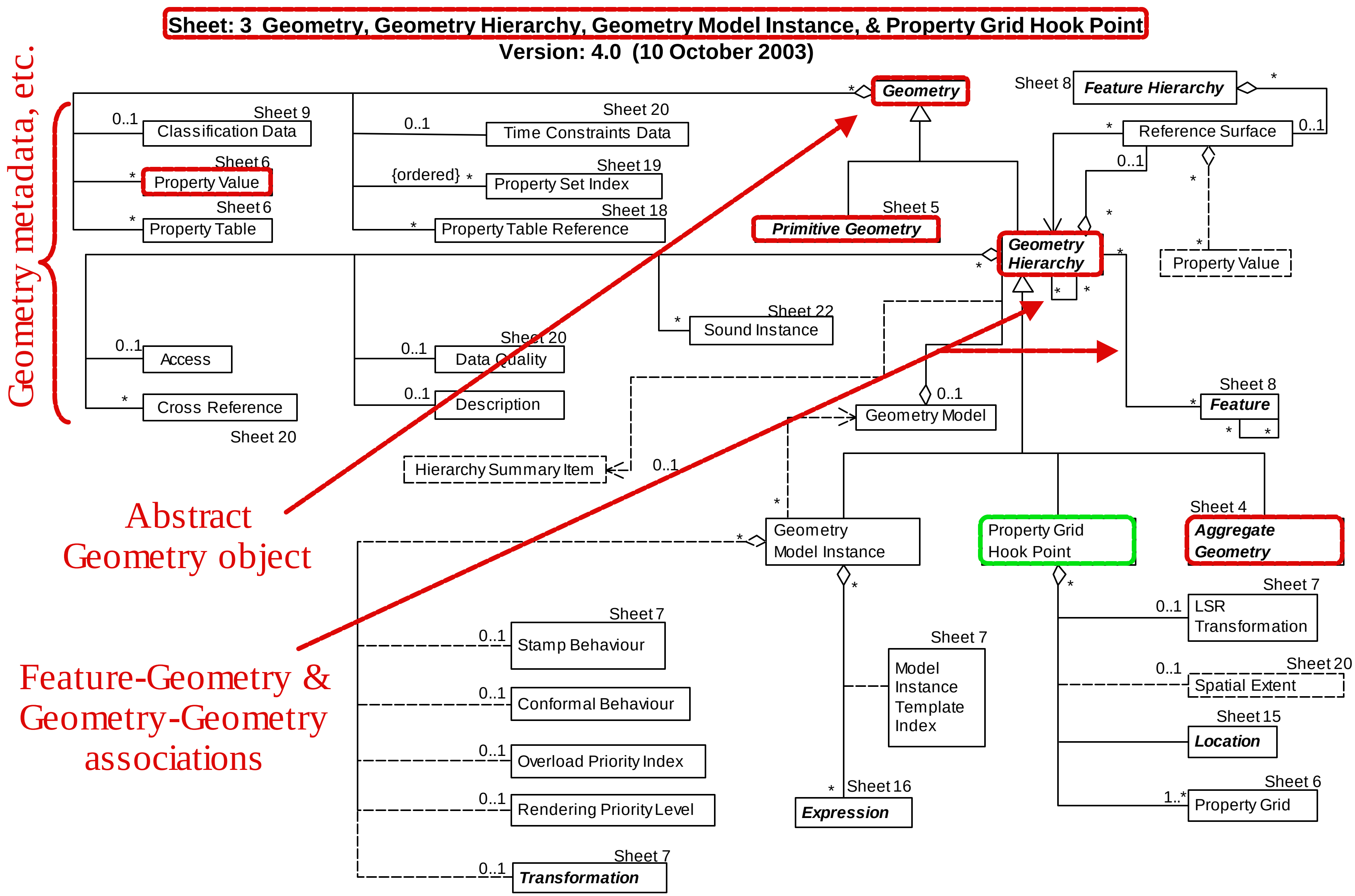
- Polygonal Data – a collection of primitive geometric elements (points, lines, polygons)
  - Terrain surface (e.g., a TIN)
  - Features integrated into the terrain, including roads, streams, etc.
  - Models of buildings & other cultural features
- Polygonal Data Sources
  - Existing Visual, Sensor & SAF Environment Databases (from projects such as STOW, CCTT, WARSIM, OneSAF, etc.)
- Polygonal data is analogous to vector data, but with simpler primitive elements and optional topology
  - Edges are limited to single line segments
  - Faces are usually limited to triangular or quadrilateral planar polygons

Polygonal Data Example – From FFD & DTED

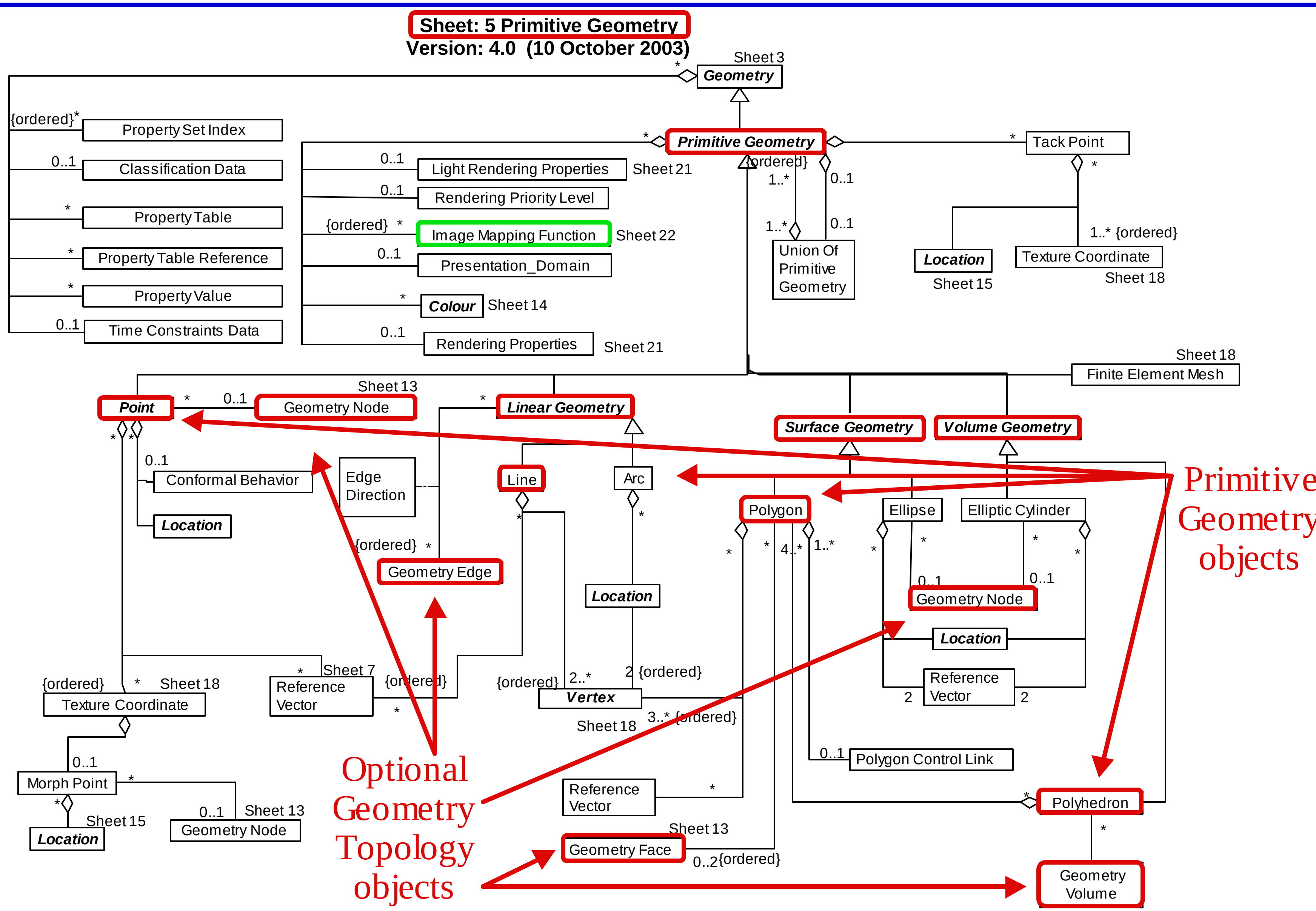


Database and image by TEC

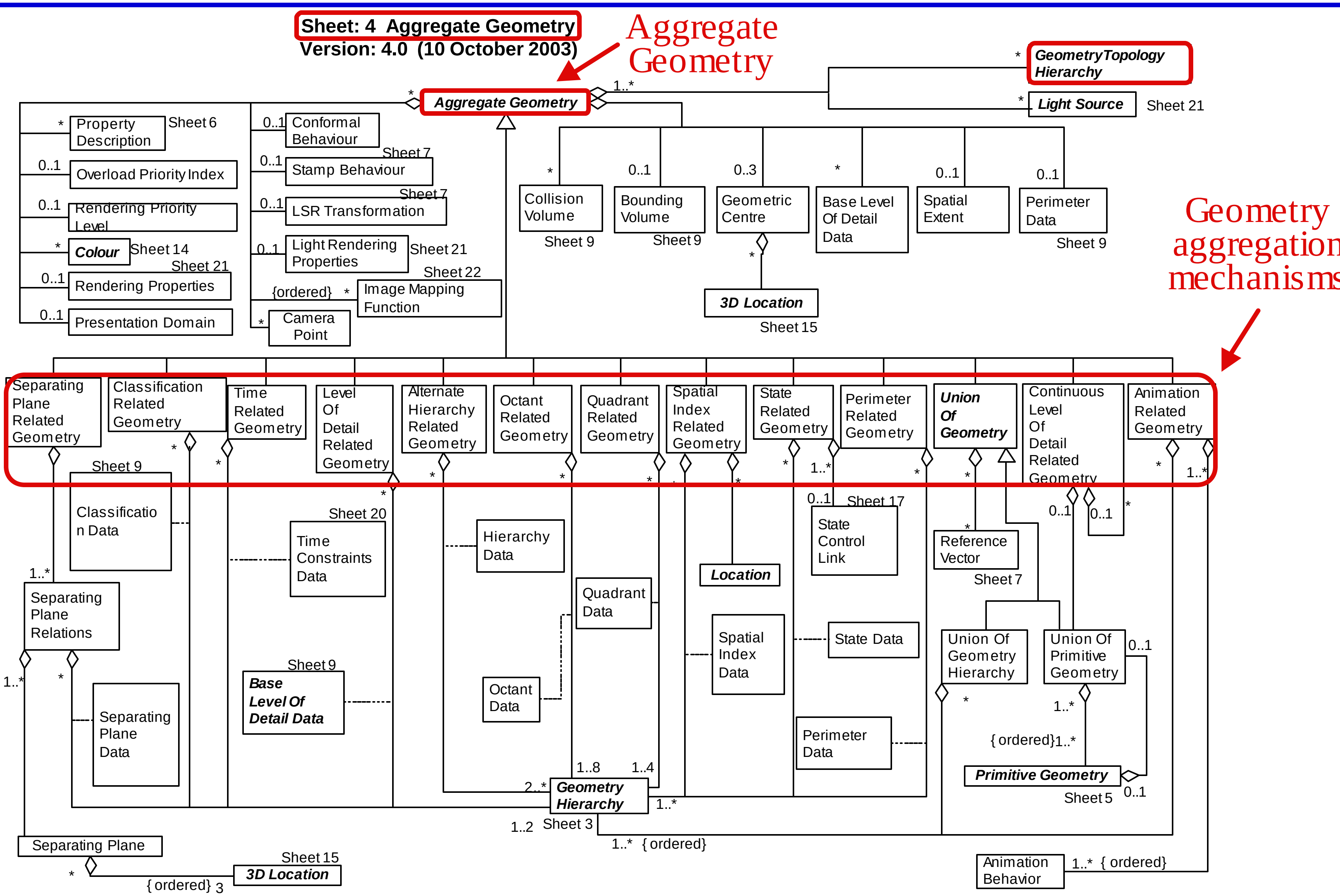
Polygonal Data in the DRM - Geometry



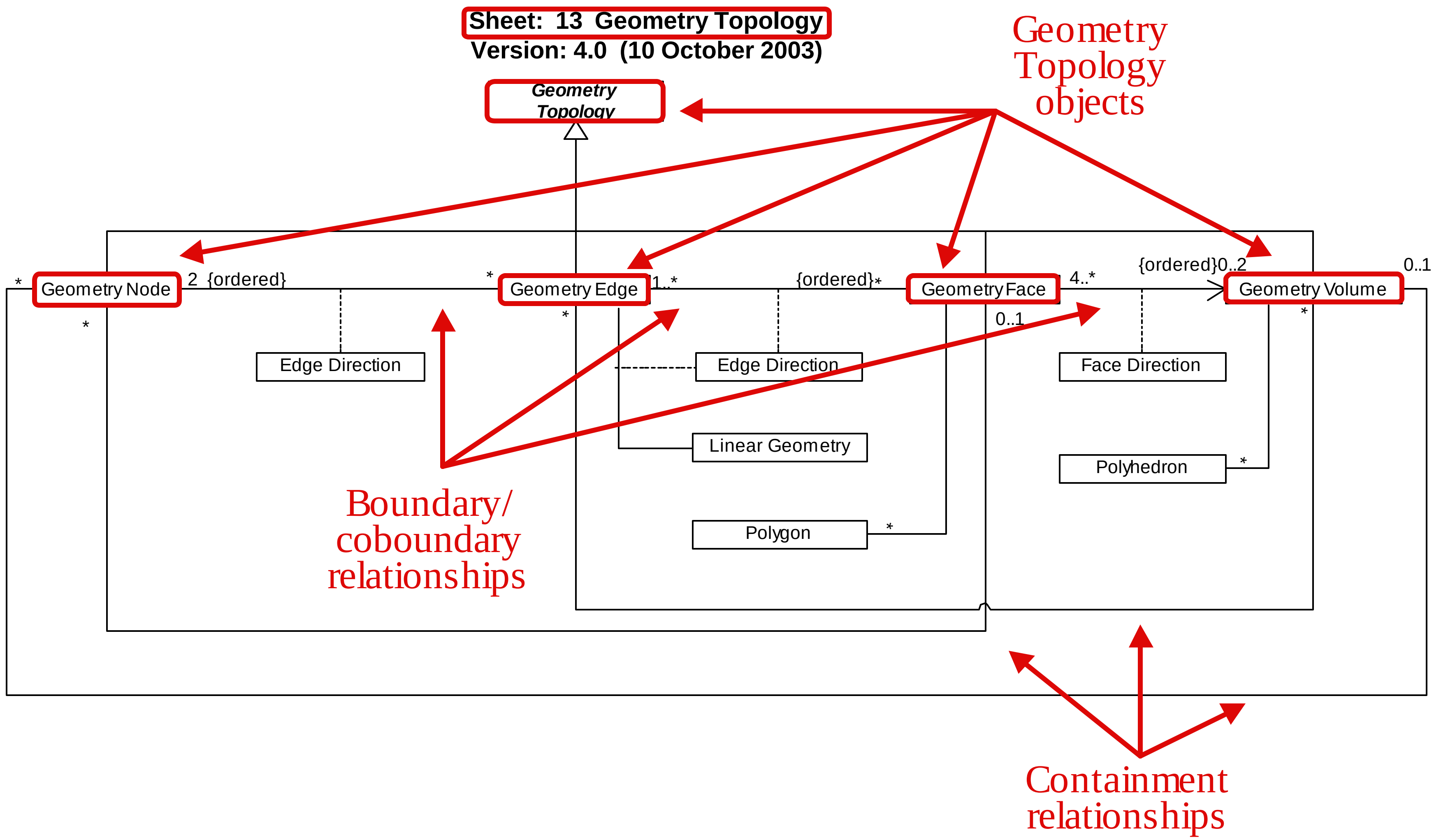
Primitive Geometry



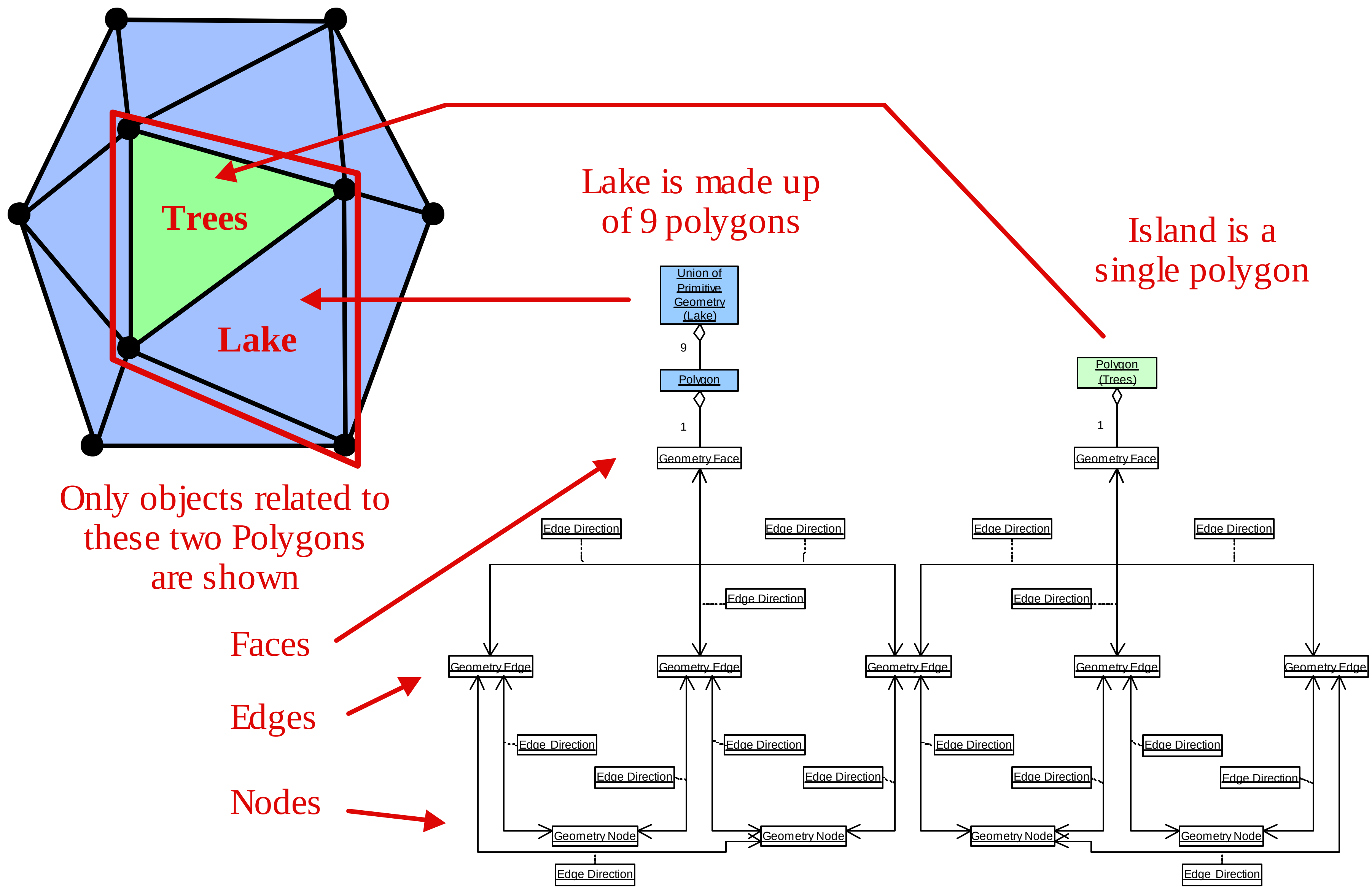
Aggregate Geometry



Geometry Topology



Geometry Topology Example



Polygonal Data Summary

- Polygonal Data is stored in Geometry objects
- Primitive Geometry objects include Point, Line, Polygon, & Polyhedron objects
- Geometry objects can be hierarchically organized in many different ways, including spatially, temporally, by classification, etc.
- Attributes can take many forms, including Property Values, Property Tables, Colors, etc., and can be associated with either Primitive Geometry or Geometry Hierarchy objects
- Primitive Geometry objects contain their own geometric information
- Optional Geometry Topology objects contain only the topological relationships between Primitive Geometry objects
  – Geometry Topology objects are organized into topological complexes (i.e., tiles), represented by Geometry Topology Hierarchy objects
- Geometry Hierarchy objects can be associated with Images, Property Grids, Features, and/or other Geometry Hierarchy objects
- Each Geometry Hierarchy object can have a complete set of supporting metadata

Associations Between Terrain Representations

- Multiple terrain representations can be associated with one another:
  - Feature-to-Feature
  - Feature-to-Geometry Hierarchy
  - Feature-to-Property Grid
  - Geometry Hierarchy-to-Geometry Hierarchy
- These associations may have an attached semantic meaning:
  - Spatial:
    - Alternate Representation – different representations of the same thing
    - Spatial Relationships – describes relative positions
      - Disjoint/Intersect, Touch, Contains/Contained By, Overlap, Cross
    - Above/Below – complete or partial horizontal overlap
    - Proximity – within a specified distance of one another
  - Functional:
    - Controls/Controlled By – acts to influence state or position
    - Supports/Supported By – acts to maintains position
    - Attached – resist movement relative to one another

Summary

- The DRM allows terrain data to be represented in several different ways:
  - Raster, as Image objects in an Image Library
  - Gridded, as Property Grid objects in a Data Table Library and/or in the geometry hierarchy within the Environment Root or a Model
  - Vector, as Feature and Feature Topology objects hierarchically organized within the Environment Root or a Model
  - Polygonal, as Geometry and optional Geometry Topology objects hierarchically organized within the Environment Root or a Model
- The DRM provides many mechanisms for hierarchically organizing Feature and Geometry objects that represent terrain:
  - Spatial, Level of Detail, Classification, State, Time, etc., etc., etc.
- The DRM allows multiple terrain representations of the same or different types to be associated, with attaching semantic meanings
  - Alternate Representation of the same terrain
  - Spatial Associations – Disjoint, Touch, Within, Overlap, Cross, Above/Below
  - Proximity