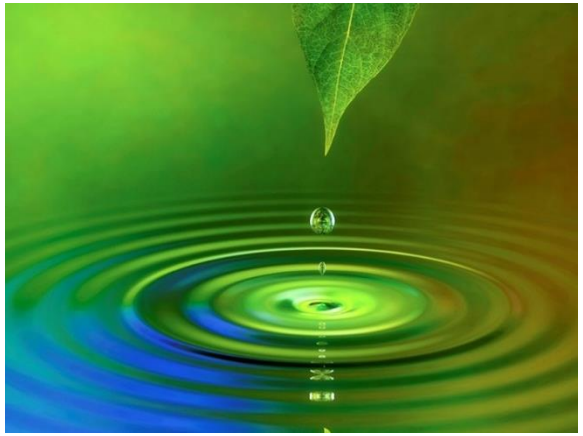



Data Integration – Industry Practices and Opportunities for Improvement

PHMSA Risk Model Working Group Data Team (RMWG)
Data & Risk Tolerance/Acceptability

Wednesday, March 8, 2017
Peter Veenstra, TRC

Outline

- Background\Context
- Current Trends in Data Management
- RISK Models and modeling GIS Data Uncertainty
- Making this accessible to Small Operators
- Observations

Background

- Principal GIS Technologist
- Involved in GIS since 1992
- GIS and Remote Sensing Background
  - Tobler's first law of Geography
- Analyst, Programmer, Architect
- Involved in designing/implementing ISAT, PODS, APDM, UPDM, PODS Next Gen
- Current PODS BOD Member, Chairperson of PODS Next Gen

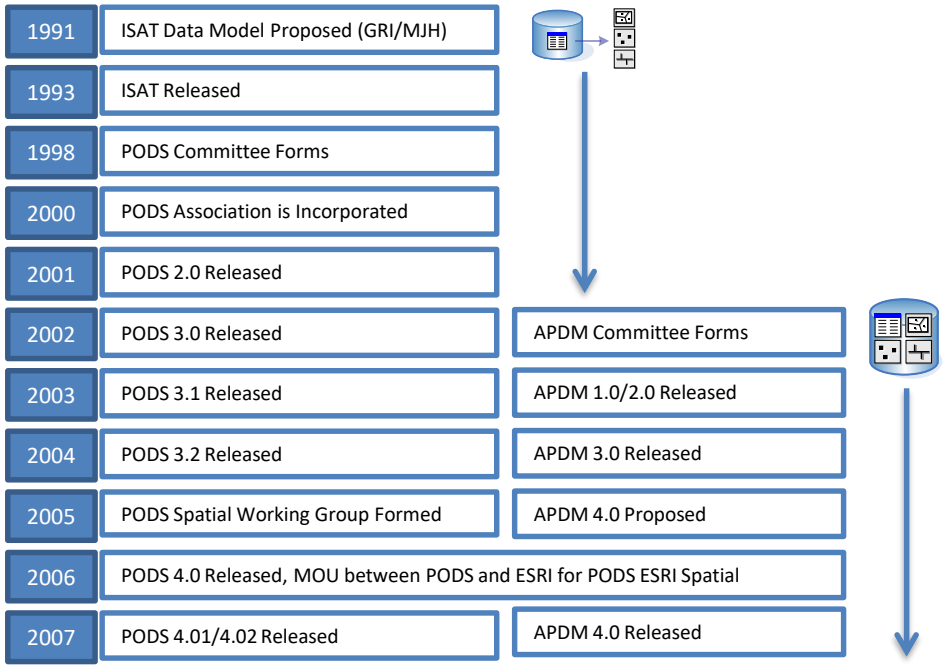
Context

- Are we looking at “Data Integration – Practices and Opportunity” within the industry …
  - For the industry as a whole …?
  - For the regulators …?
  - For the operators individually or as a whole …?
- Difficulty in sharing of information and knowledge
  - Release of perceived competitive advantage and/or proprietary knowledge
  - Risk of exposure and damage to image and perception
  - Cost of doing business (is it worth it to spend this kind of money?)
  - Balance between under or over-proscribing regulatory requirements and compliance

Part 1 - Current Trends in Data Management

- Relational Databases and Pipeline Data Models
- Geography rears its ugly head
- Scalability for small operators
- Systems built in silos
- Operational Excellence

Part 1 – Relational Databases and Pipeline Data Models



1991	ISAT Data Model Proposed (GRI/MJH)	
1993	ISAT Released	
1998	PODS Committee Forms	
2000	PODS Association is Incorporated	
2001	PODS 2.0 Released	
2002	PODS 3.0 Released	APDM Committee Forms
2003	PODS 3.1 Released	APDM 1.0/2.0 Released
2004	PODS 3.2 Released	APDM 3.0 Released
2005	PODS Spatial Working Group Formed	APDM 4.0 Proposed
2006	PODS 4.0 Released, MOU between PODS and ESRI for PODS ESRI Spatial	
2007	PODS 4.01/4.02 Released	APDM 4.0 Released

2008		
2009	PODS 5.0 Released	
2010	PODS ESRI Spatial 5.0 Released	APDM 5.0 Released
2011	PODS/ESRI Spatial 5.1 Released	
2012	PODS/ESRI Spatial/Open Spatial 5.1.1 Released	
2013	PODS 6.0 w/ Modularization Released	APDM 6.0 Released
2014		APDM No Longer Supported
2015	PODS APR and PODS Next Gen	ESRI UPDM / APR Released

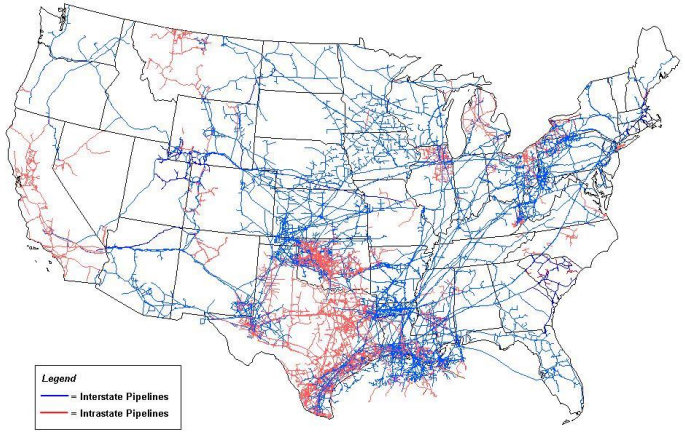
So here we are …

Part 1 – Relational Databases and Pipeline Data Models

- Relational Databases and Pipeline Data Models
  - This seems to be our only option
  - RDBMS\GeoSpatial Data Models are excellent for storing the location of the centerline and managing the stationing LRS associated with that
  - RDBMS are well suited to managing ASSETS, but not necessarily CONDITIONS
- Flexibility for storing data easily
  - Geographic Coordinates vs Linear Referencing
- Cannot support unstructured data, un-validated data or the massive amounts of data
- Excellent for Fit-for-purpose data and applications – Centerline and Assets
- Also excellent for tracking data edits via history and versioning (ESRI Geodatabase Versions/Long Transactions)

Part 1 – Geography rears its ugly head

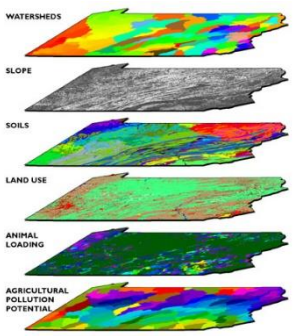
- Geography in this context represents different parts of the United States and the World
- Different geo-political agencies look at standards and methods differently (regulations – US, Canada, UK, Europe, Russia, Australia)
- Geography allows us to integrate things in a common coordinate system but forces us to deal with different things
- Each geographic area has a different set of RISK (some are similar) but there is no one-size fits all

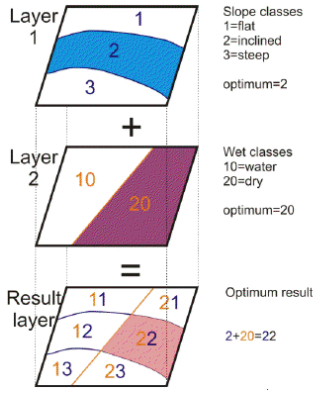


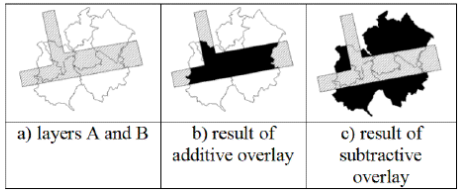
Source: Energy Information Administration, Office of Oil & Gas, Natural Gas Division, Gas Transportation Information System

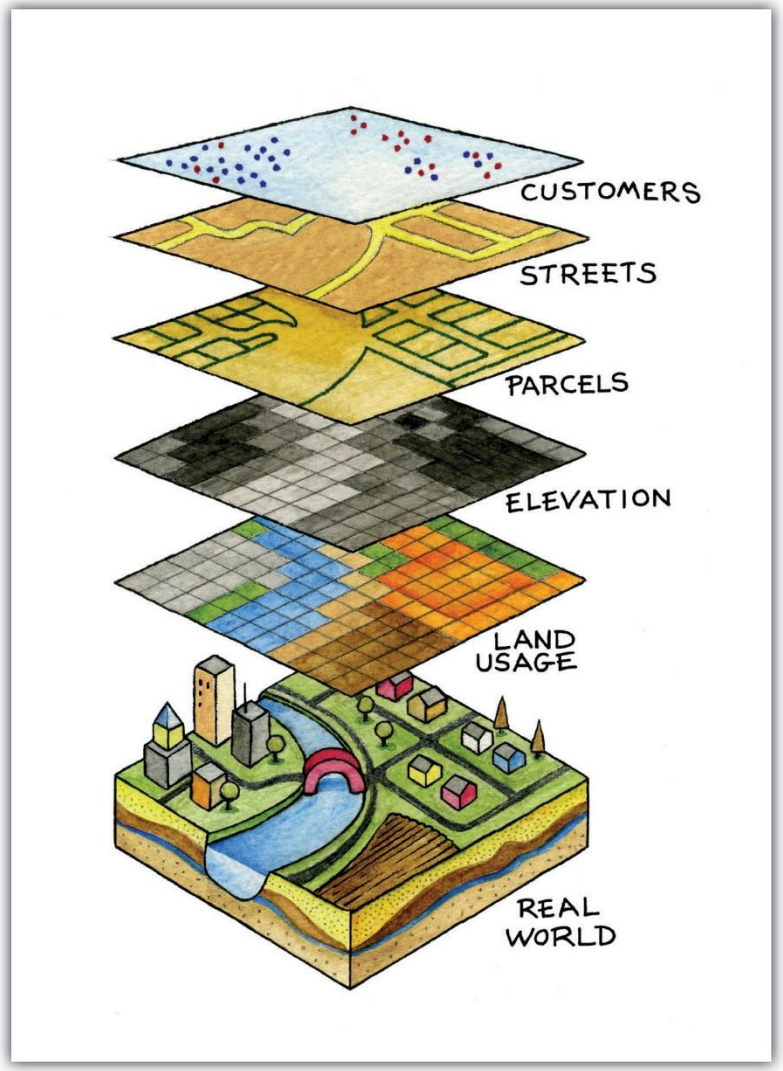
Part 1 – Conversely, Geography is beautiful

- Coordinates unify different geographic information into a common plane of reference
- Creates inherent relationships between objects in space
  - Tobler's Law Of Geography
- Data can be overlaid with each other to form new data in a different manner than the standard SQL table join
- This is the basis for most 'classic' risk implementations









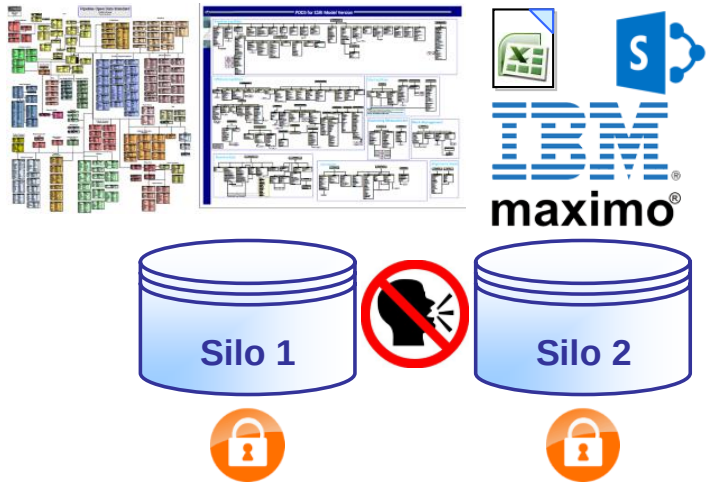
Part 1 – Scalable for small operators

- Access to GIS systems, PODS or other GIS functionality
  - Complicated systems, detailed knowledge, specific requirements
- They are still bound by the same regulations and standards
- How do we support smaller operators

- Other industries have set up smaller data models and collectively built open source technologies for data management
- Agencies within this sphere could provide source data as services (Federal Geospatial Data Committee)
- Still not granular or specific enough for the pipeline industry but the model works

Part 1 – Systems built in Silos

- Risk requires data from many different systems
  - Work order management, document management, SCADA, Cathodic Protection
  - Pipeline GIS may not be the place to store this
  - Systems must be integrated after the fact
- Systems built on PMP
  - “My job, done now, on time, on budget, in isolation of the down-stream effects”.
  - Systems are compartmentalized and not built with the end-in-mind
  - There is little data exchange, data flow, data hand-off (this is HARD to do)
  - Varying levels of system complexity, completeness, and data quality/structure
  - If you are lucky there are coordinates being saved



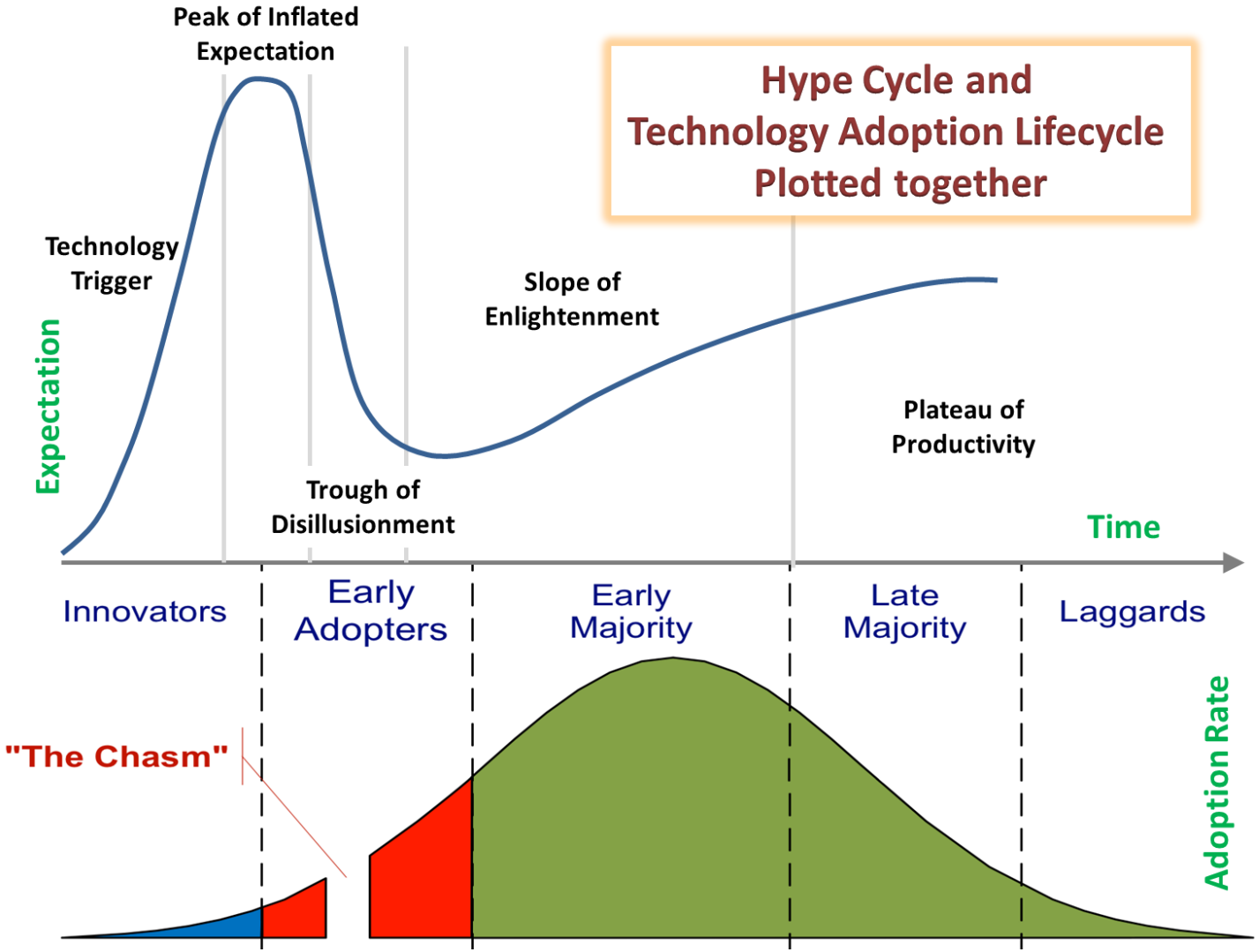
Part 1 – Operational Excellence

- Lack of Operational Excellence for the “System of Record”
  - Fit for purpose data with quality sufficient for the task – Data Excellence
  - Are end users challenged to get the right data into the system?
  - Is the system providing the end users with the right data?
  - Younger generation – quality on demand (ADD)
- Organizations are relying on ‘white’ expertise
  - Subject Matter Experts know the data better than the systems – how do we capture this?
- More is being done with less (continued separate of church and state)
  - Pipeline Integrity Engineers are being forced to learn GIS
  - GIS users need to understand how pipeline integrity data management works
- Digital Workflow is incomplete
  - Data is missing or is not collected or delivered
  - Field to enterprise work-flow is still kludge
  - Once data are gathered or collected, how are they integrated?
  - Many operators are gathering, massaging and formatting the same data (but separately)
- IT gets in the way
  - Need to enable solutions

Part 2 - Potential Trends for Pipeline Data Management

- The Cloud
- Unstructured Data (Big, IOT)
- Services Architecture (API)
- Data Integration\Lakes\Exposure
- Machine Learning

Adoption of Technology (Pipeline Industry)



https://setandbma.wordpress.com/2012/05/28/technology-adoption-shift/

Part 2 – The Cloud

- Scalability on demand
  - Development environments – experimentation, learning, performance
  - Built on Open Source Technologies (organic infrastructure and Development)
- Collaboration
  - Shared computers and data stores (workspaces and servers)
    - migration, training, testing, data quality checking
  - Remove bureaucracy of IT
    - Rapid deployment and sizing without CAPEX process
  - Crowd (Company/Organizations) sourcing for data sharing
    - Build data storages and entry procedures for capturing shared data
  - Provides services (SMS)
    - Provide notifications
    - Provides map services

Part 2 – The Cloud

- Data Services
  - High performance big table (tall and wide) performance (use SQL)
  - Built in RDBMS technology and services (not just Oracle and MS SQL Server) but RDS (POSTGIS)
  - Big Data (Hadoop, MongoDB, Google)
  - Deep Storage (AWS S3)
- Notification Platform
  - Built in services architecture – SMS (information on your phone)
  - Notifications – something has changed, please respond (devices, location)
  - Push/Pull – I have changed something, what is happening here?
  - Post to Default (ESRI Spatial Geodatabases)

Part 2 - Services Architecture

- Individual Cloud Based Components that talk to each other (with common technology) rather than monolithic software systems
- Constant and iterative data processing from local/disparate/siloed systems to shared data stores
- Processing on demand (as-needed) (FME Cloud)
- Once data is formulated serve it to systems that can easily consume it and provide access to it in a simple manner

Part 2 – Services Architecture

Compute

Storage

Database

Migration

Networking & Content Delivery

Developer Tools

Management Tools

Security, Identity & Compliance

Analytics

Artificial Intelligence

Mobile Services

Application Services

Messaging

Business Productivity

Desktop & App Streaming

Internet of Things

Ease of Use and Interaction (Value not complexity)

Connect the field to the office Grant your team the ability to access and update job specific information in real time on any mobile device.

Track all your progress to ensure your projects are traceable, verifiable, and complete.

Keep up to date in real time with changes and alerts that come down the pipeline. *Did my risk profile change?*

Support for defined and custom workflows. *The nature of pipeline data and integrity management is highly flexible.*

Scalability to support any size project. Leveraging Cloud infrastructure and a managed secure cloud model.

Affordable and quick to implement. No lengthy implementation cycle and offered in a SaaS infrastructure and with a simple pricing model.

Part 2 – Unstructured Data

- JSO
- Com
- Just
- Red
- Can store massive amounts c
  - Mega/Giga/Terra/Peta
  - 200 runs of ILI data per year (align, ma
- Rapid Query (pull it down) and Analysis (cho
- Big Data (Scale Out) and Big Table Data (Scal

(point clouds), SCADA (
Conditions), ILI Tally AN


```
{ "type": "FeatureCollection",
    "features": [
      { "type": "Feature",
        "geometry": {"type": "Point", "coordinates": [1425610.0, 1425610.0, 36.0]},
        "properties": {       "featureType": "Valve",
                                "searchIndex": "ML100 Valve 36.00\" Ball Valve ML100-001A-S",
                                "groupId": "8E6B075D-A8D9-44B0-AE45-F19851159A77",
                                "groupName": "ML100"
      }
    ]
}
```

```
{"valve": {
  "id": "ML100 Valve 36.00\"",
  "valu
  "insp
    "in
      {
      {
      {
    ]
  }
}}
```

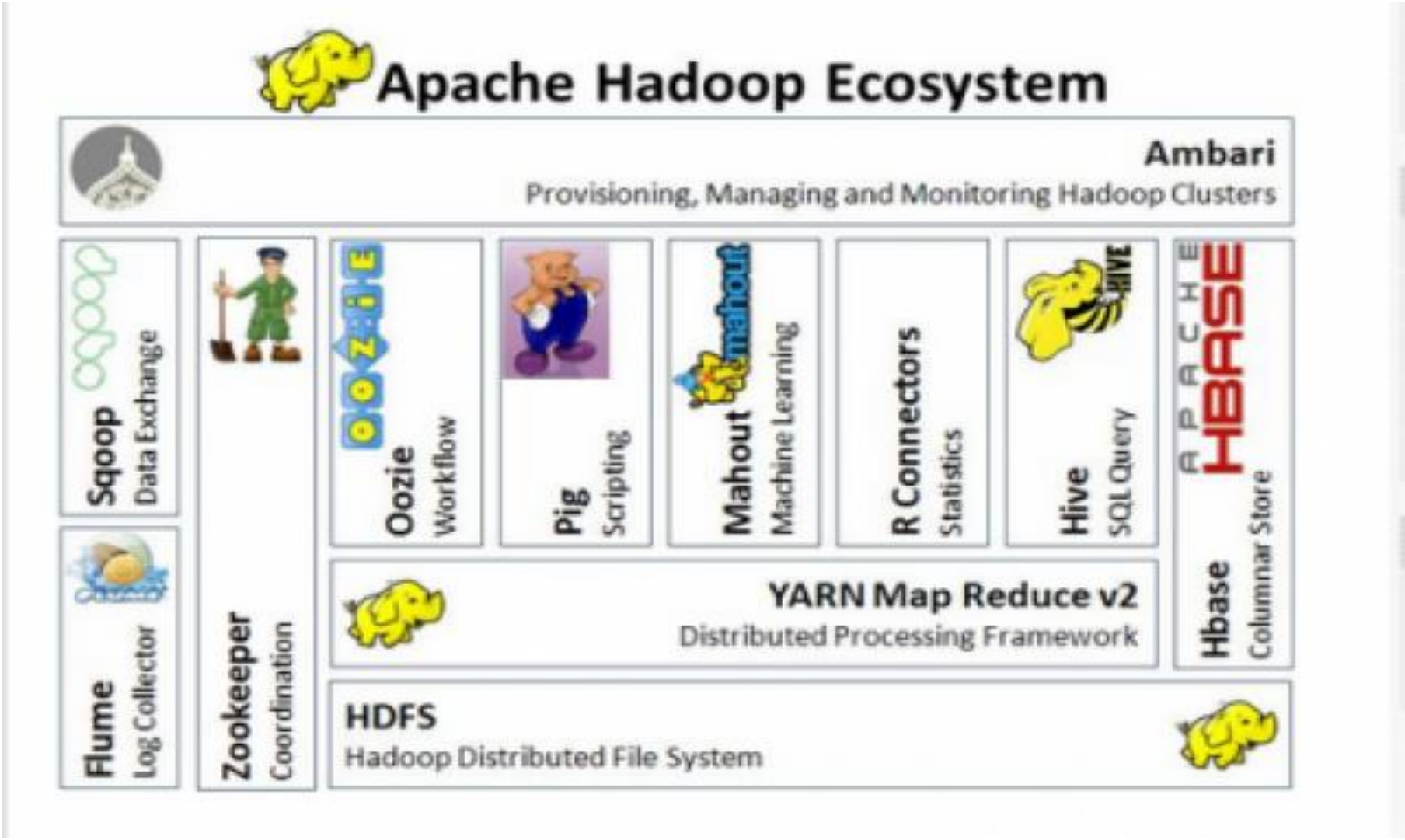
```
{"valve": {
  "id": "ML100 Valve 36.00\"",
  "value": "feature",
  "engin
      }
  }
}
```

```
{"valve": {
  "id": "ML100 Valve 36.00\"",
  "value": "feature",
  "throughput": {
          "throughput": [
    {"date": "01/01/2001", "volume": "standard"},
    {"date": "03/01/2001", "volume": "below capacity"},
    {"date": "06/06/2001", "volume": "over capacity"}
  ]
      }
  }
}
```

Part 2 – Big Data

- Large amount of small computers holding JSON format data
- Query layer for retrieve and organize
- Reminiscent of main-frame computers (fast, but required black magic interface)
- Cloud providers support OOTB toolkits (Native SQL Interface)
- Requires the inclusion of geography or tags

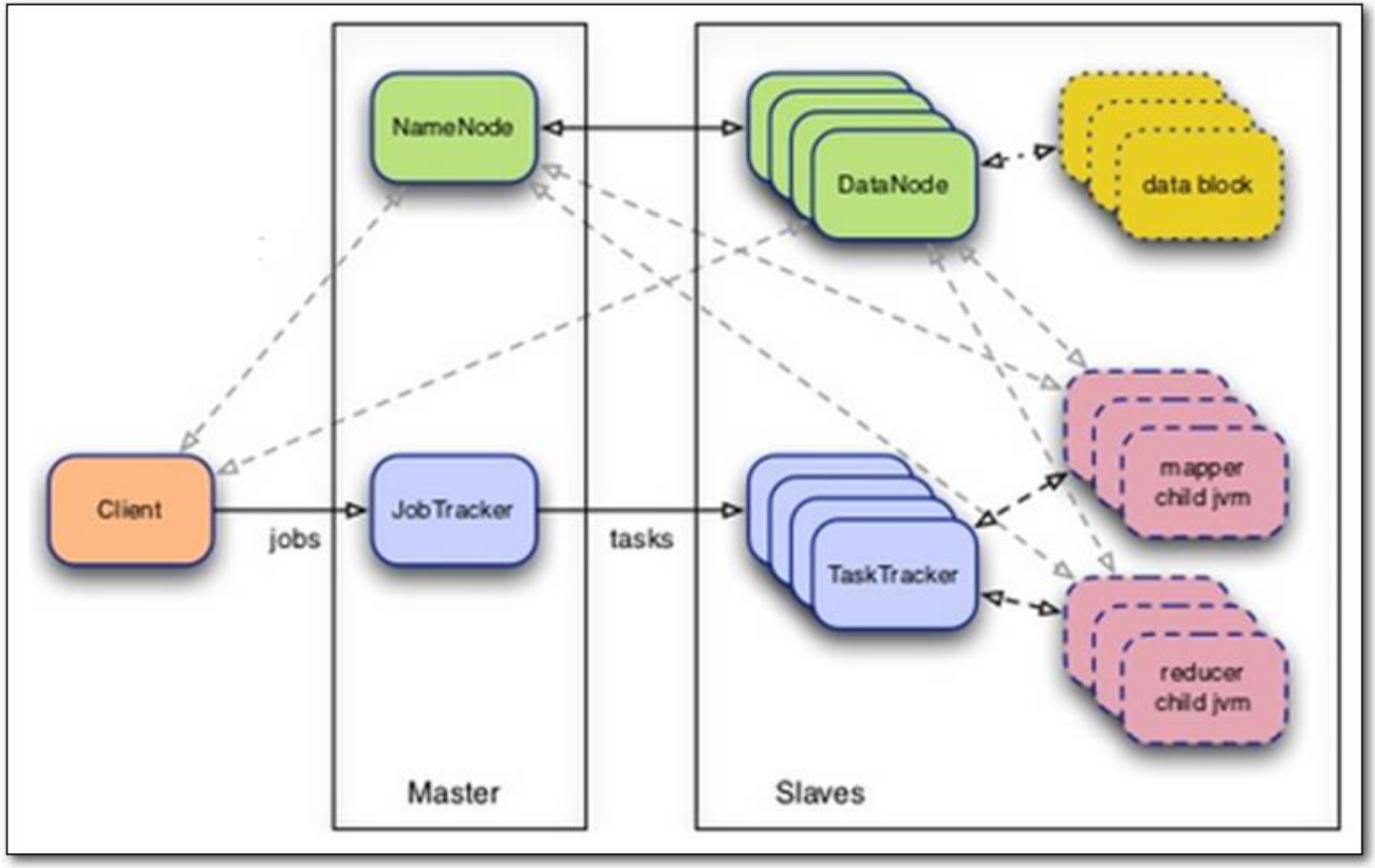
Hadoop


Apache Hadoop Ecosystem
Ambari
Provisioning, Managing and Monitoring Hadoop Clusters
Sqoop
Data Exchange
Flume
Log Collector
Zookeeper
Coordination
Oozie
Workflow
Pig
Scripting
Mahout
Machine Learning
R Connectors
Statistics
Hive
SQL Query
APACHE HBASE
Hbase
Columnar Store
YARN Map Reduce v2
Distributed Processing Framework
HDFS
Hadoop Distributed File System

Hadoop

- There are two parts to Hadoop:
  - HDFS - split the data, put it on different nodes, replicate and manage it
  - MapReduce - batch processing over an increasing amount of data
- Physically there are many parts
  - Scale out instead of scale up
  - Many computers/servers in a managed cluster
- Amazon, Mapr, Cloudera

Hadoop


NameNode
DataNode
data block
Client
jobs
JobTracker
tasks
TaskTracker
mapper
child jvm
reducer
child jvm
Master
Slaves

Hadoop

- Hadoop uses key/value pairs as its basic data unit and is flexible enough to work with the less-structured data types
- Data can originate from any form, but it eventually transforms into key/value pairs for processing functions to work on
- Text Files
  - XML, txt, JSON
- Binary Files
  - Video dissected frame-by-frame

Hadoop

- Functional programming (MapReduce) instead of declaritive queries (SQL)
  - You query data by stating the result you want and let the database engine figure out how to derive it
  - MapReduce you specify the actual steps in processing the data, which is more analogous to an execution plan for a SQL engine
  - SQL has query statements, MapReduce has scripts and code
- Better get some Java (or Python)

- Pipeline Use Cases
  - Tracking location on a massive scale – popcorn trail density, ILI density en masse (analytics vs other geography)

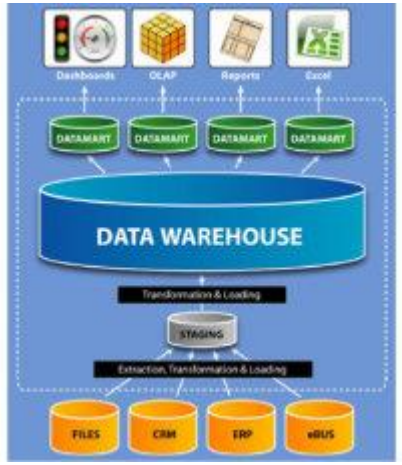
Part 2 –Big Table

- Massive number of rows and columns
- Dynamically segmented or merged tables
- Supported by providers or in private cloud
- Can be linked via standard SQL with other Big Tables
- Can store geometry via GeoJSON
- Allows for storage of complex data structures within an attribute
- Requires a mechanism for getting it out and displaying it
- LAYERLESS!!!

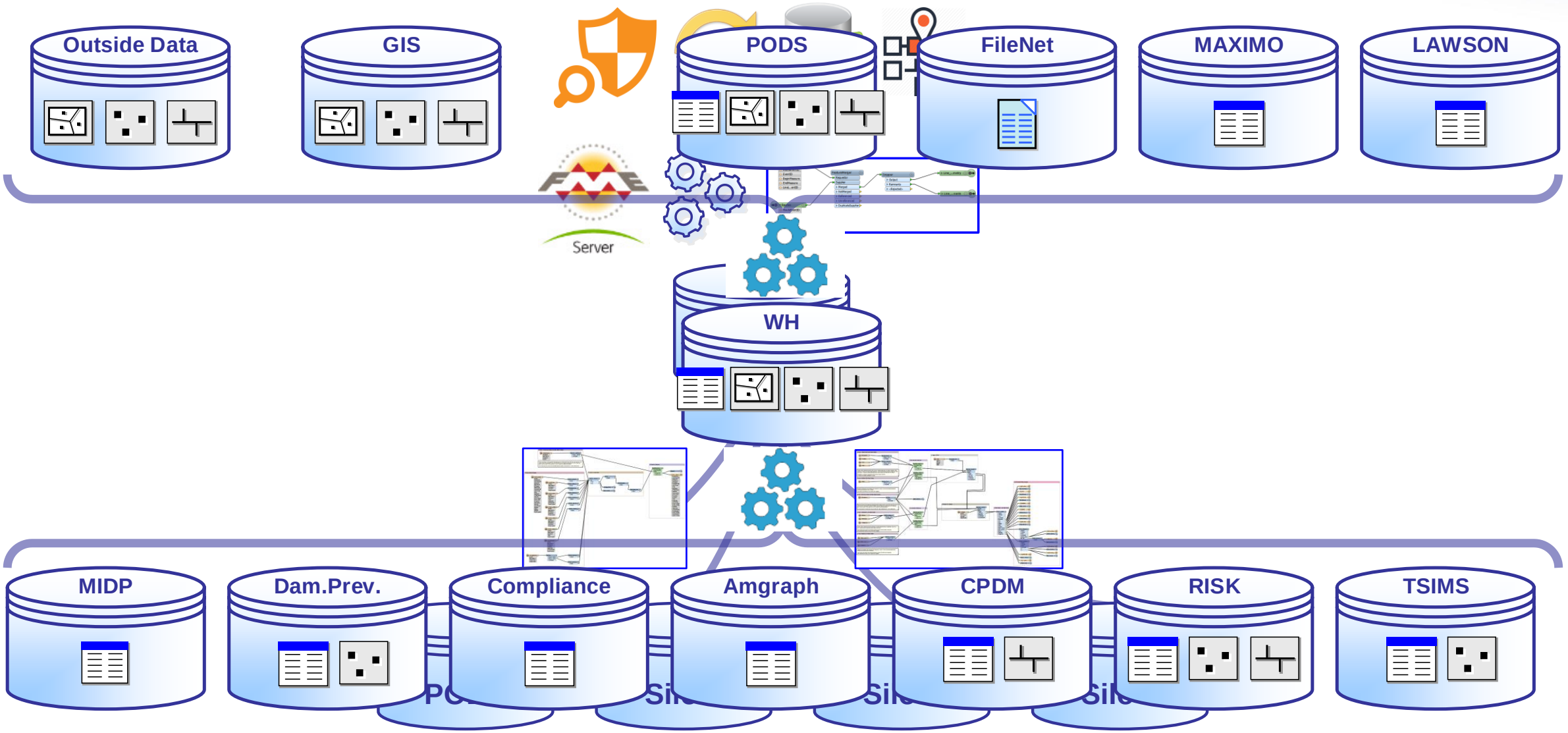
- Use cases – Previous demonstration

Part 2 – Data Integration

- Data Lake
  - A data lake is a collection of storage instances of various data assets additional to the originating data sources. These assets are stored in a near-exact, or even exact, copy of the source format. (www.gartner.com/it-glossary/data-lake/)
- Data Warehouse
  - A data warehouse is a storage architecture designed to hold data extracted from transaction systems, operational data stores and external sources. The warehouse then combines that data in an aggregate, summary form suitable for enterprise-wide data analysis and reporting for predefined business needs (http://www.gartner.com/it-glossary/data-warehouse/)
- Put the data into the hands of those who need to use it, in a format that they need to use it
- Publish it for consumption



Part 2 – Data Integration



Part 2 – Machine Learning and BI

- Machine Learning
  - Advanced machine learning algorithms are composed of many technologies (such as deep learning, neural networks and natural-language processing), used in unsupervised and supervised learning, that operate guided by lessons from existing information. (http://www.gartner.com/it-glossary/machine-learning/)
- Business Intelligence
  - Business intelligence (BI) is an umbrella term that includes the applications, infrastructure and tools, and best practices that enable access to and analysis of information to improve and optimize decisions and performance. (www.gartner.com/it-glossary/business-intelligence-bi/)

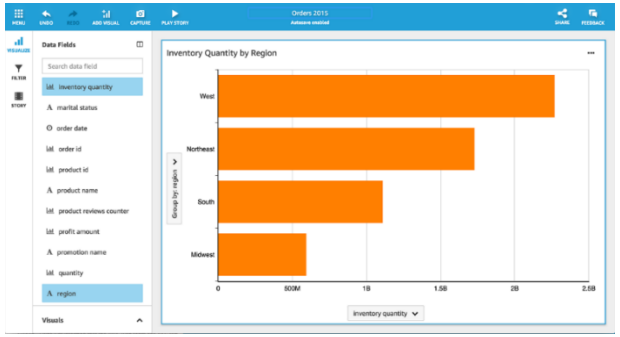
Part 2 – Machine Learning

- Machine Learning
  - Advanced machine learning algorithms are composed of many technologies (such as deep learning, neural networks and natural-language processing), used in unsupervised and supervised learning, that operate guided by lessons from existing information.
    (http://www.gartner.com/it-glossary/machine-learning/)
- Requires data, systems and business\problem statement
- Similar to classification of remotely sensed imagery
  - Supervised and Unsupervised classification

- Classification of ILI Data and Alignment of ILI Data to GIS
- Identification of trends in incidents and pipeline defects compared to geographic conditions

Part 2 – Business Intelligence

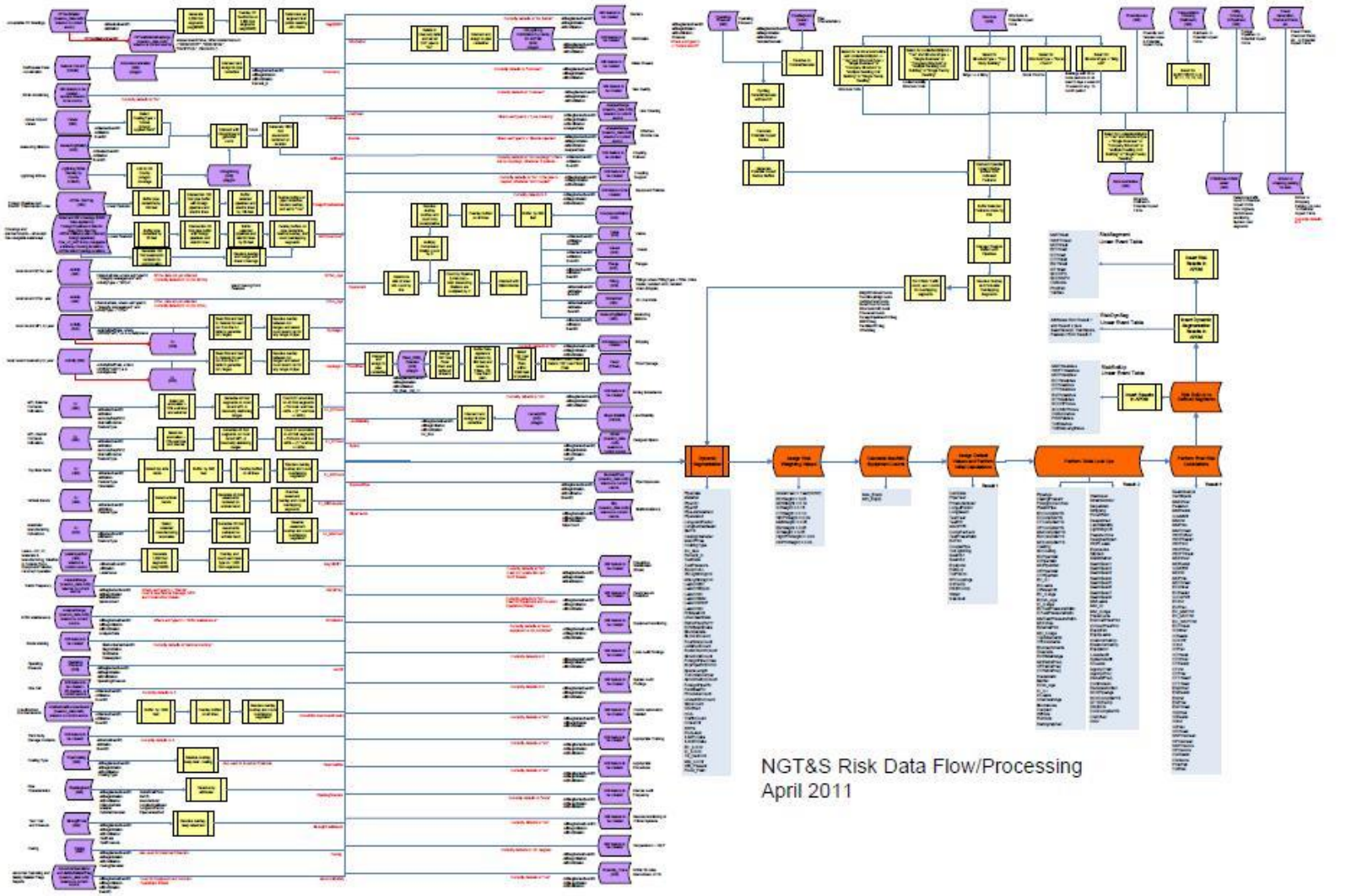
- Business Intelligence
  - Business intelligence (BI) is an umbrella term that includes the applications, infrastructure and tools, and best practices that enable access to and analysis of information to improve and optimize decisions and performance.
    (www.gartner.com/it-glossary/business-intelligence-bi/)
- Requires data, systems and business\problem statement
- Allows for the visualization and analysis of data in ad-hoc fashion
- Tableau, Tibco Spotfire, Amazon QuickSite, ESRI Insight
- SME review and visualization for potential risk



Part 3 - Risk Models and Data Uncertainty

- RISK Models
  - Qualitative and Quantitative
  - Machine Driven and SME Driven
  - Standard Methodology
  - Require data
- Data Uncertainty
  - Uncertainty
  - Methods for modeling uncertainty
  - Modeling GIS data quality using uncertainty\

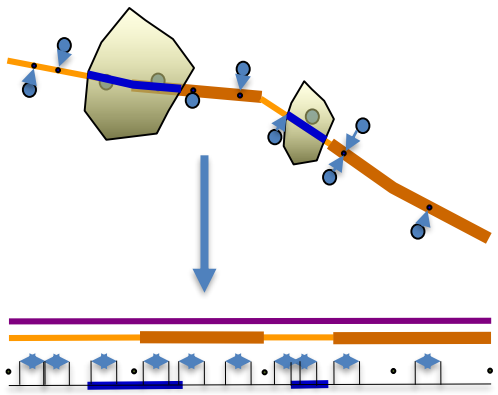
RISK PROCESS

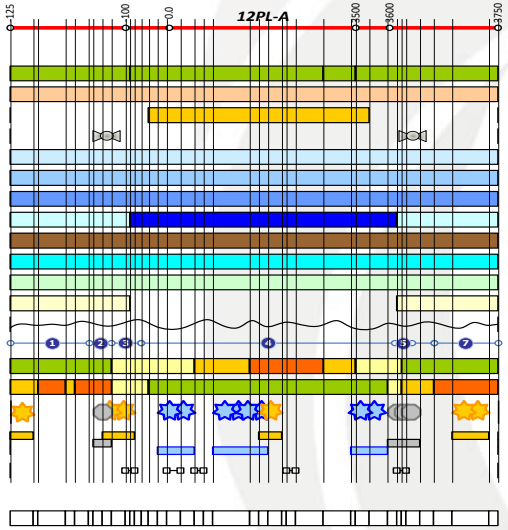


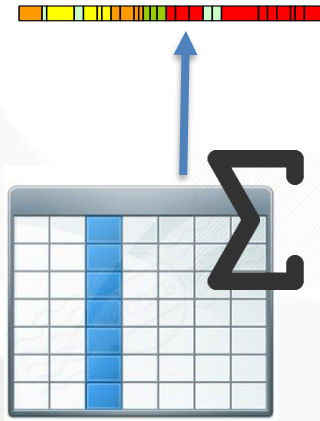
Network/Spatial-Based Risk Analysis

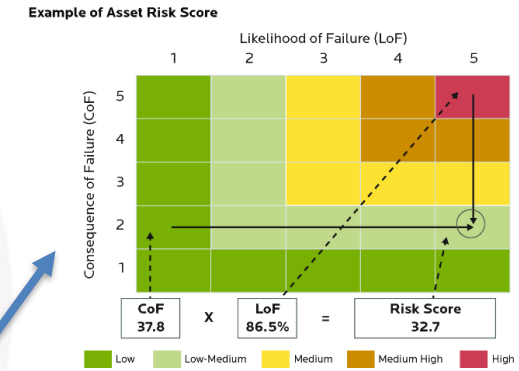
- GIS based Risk Application
- Geo-Aggregation
- Dynamic Segmentation
- Calculate Risk Score
- Risk Results Visualization

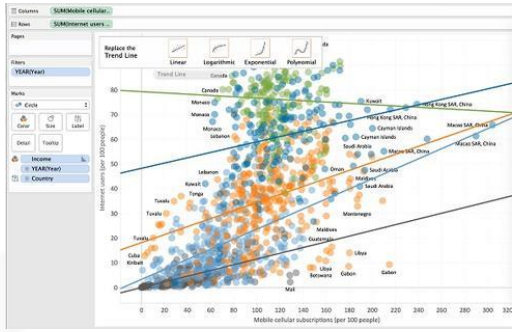
RISK











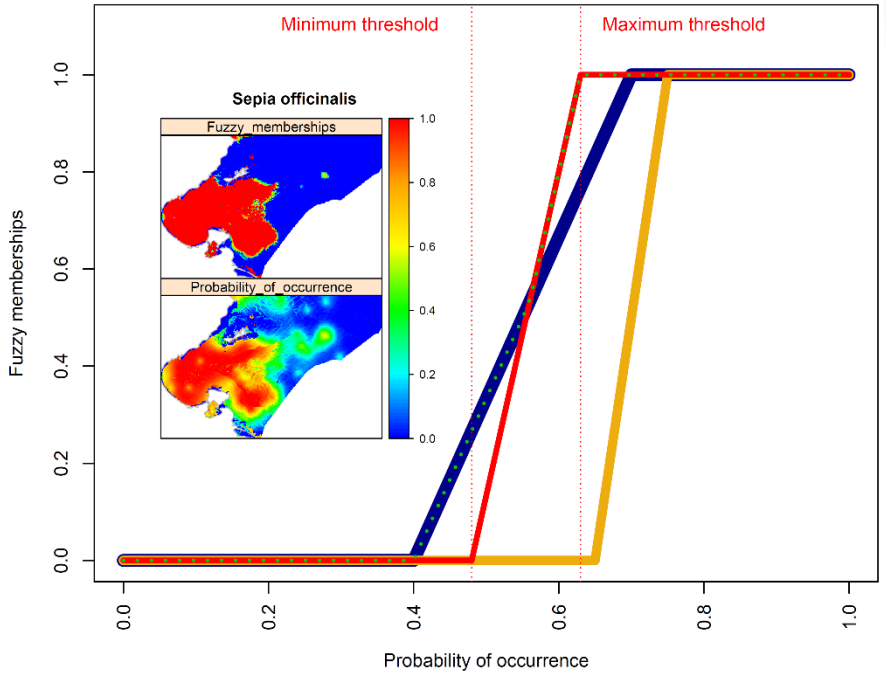
Part 3 – Risk Models

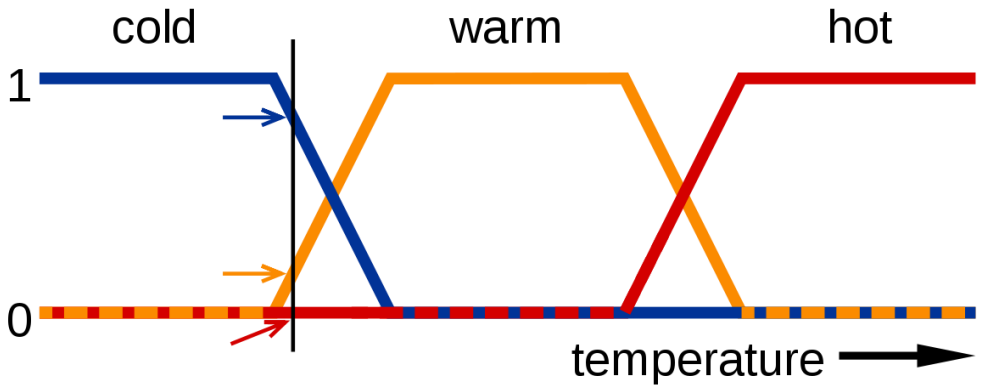
- Abstraction of reality (is a model)
  - Your tomato is my tomatoe
  - Geographic location changes the inputs
- Standard Methodology
  - Gather, convert, segment, calculate, organize, slice/dice
- Qualitative and Quantitative
  - Zero Incidents, Zero Cost, Zero loss of life
- Machine Driven and SME Driven
  - Somethings require a computer
  - All result require validation against experience and actual data (incidents, leaks)

- Require data, require complete data, require accurate data
  - Or need a mechanism to deal with unknown, missing or uncertainty

Part 3 - Data Uncertainty

- Is data quality a measure of uncertainty?
- How do we measure quality and uncertainty?
- In PODS we model uncertainty:
  - Via attributes: SourceGCL, ValidityTolerance, DocumentCrossRef
  - Via linear events: CenterlineAccuracy
- Methods for modeling uncertainty
  - Bayesian (Probability), Dempster-Shafer (Evidence), Fuzzy Theory (Incomplete/Imprecise)
- Modeling GIS data quality using uncertainty
  - Location Quality/Precision/Accuracy
  - Attribute validity



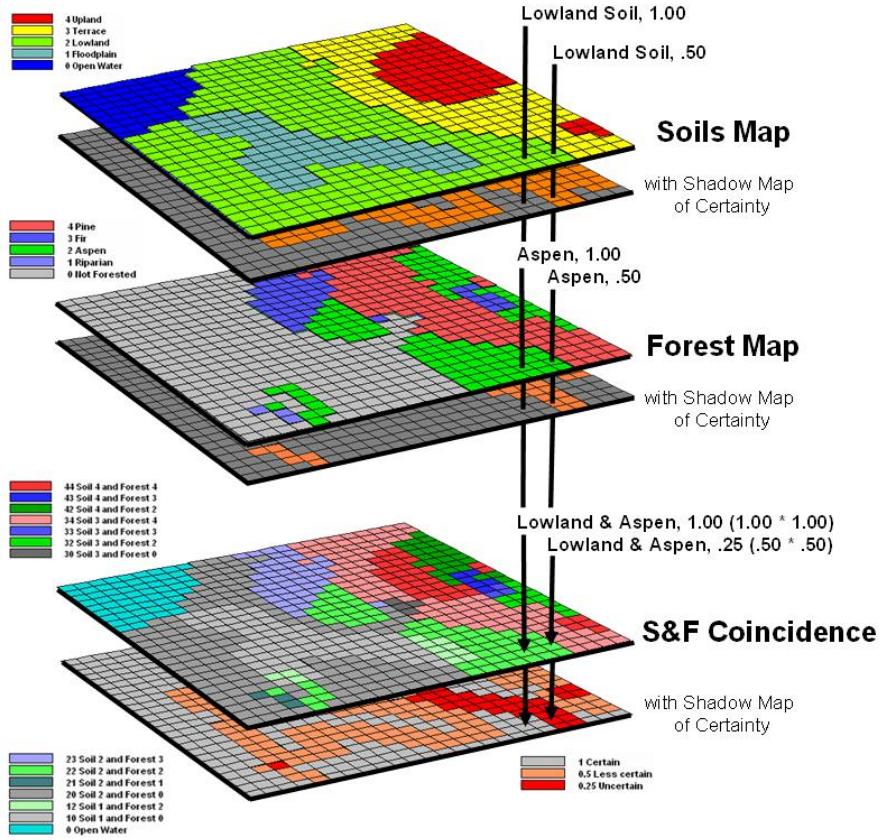


Part 3 - Data Uncertainty

- Risk = Consequence x Likelihood
- Consequence is critical part of risk.
- Location is a critical part of consequence.
- Causality is an important part of likelihood.

- Accuracy of spatial dataset are suspect esp. public or purchased data sets.
- Incorporate Fuzzy Surfaces within data analysis.

- How you analyze or deal with this uncertainty might determine the priority of your actions

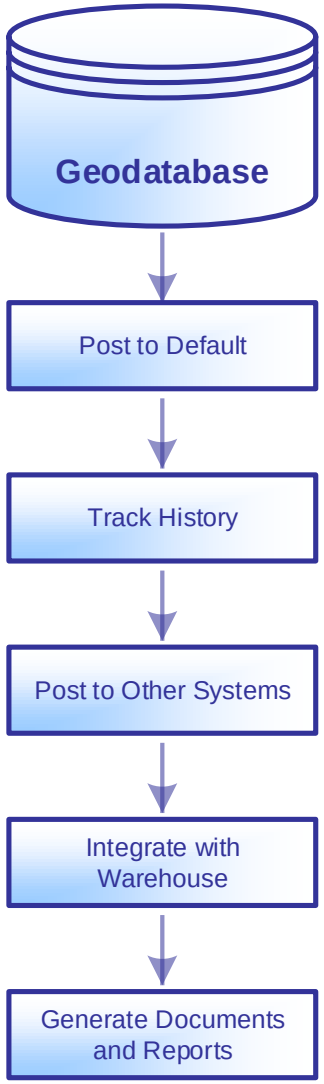


Part 3 - Data about the data

- How do we model what we don't have or do not know
  - Critical data is missing or unknown – can we overcome this in our models?
- Data is not available
  - How do we get it?
    - Legacy conversion
    - Field Data Collection
- Data is stored in other systems
  - How do we integrate it?
  - Does it share a common coordinate reference? Does it share a tag or name reference?
- Everyone has their own way of doing it (different data owners)
  - Is there a way to make standardization easy, accessible and well understood?
- Data is stored in different frame of reference
  - All data will require some transformation to make it useful
  - Data has different structures – how do we merge it?
- Data has different patterns that go beyond coordinates and PK-FK relationships (Semantics, Ontology)

Sidebar - Post to Default (Gateway to Information Management)

- ESRI Spatial Geodatabases have a technology called versioning
- Versions are long transactions that track the adds/deletes to a database
- At the root level is the DEFAULT version of the system/database
- When data is posted to DEFAULT it becomes accessible to standard queries against the base table
- Triggers and other technologies can be leverage to track ‘post to default’
- This starts the data and information management process
  - Post to other systems
  - Perform QA/QC
  - Start processing and integration workbenches
  - Build other data stores



Part 3 - Small Operators

- How do we engage small operators in the GIS and RISK process?
  - Allow for flexibility on data management
  - Provide GIS solutions and RISK models for different levels of complexity
- PODS Next-Gen – PODS LITE
  - Need to provide the minimum data set for a pipeline operator to understand the position of the pipeline centerline and to operate their pipeline safely
  - Provides the minimum required data set for submitting a complete NPMS submittal

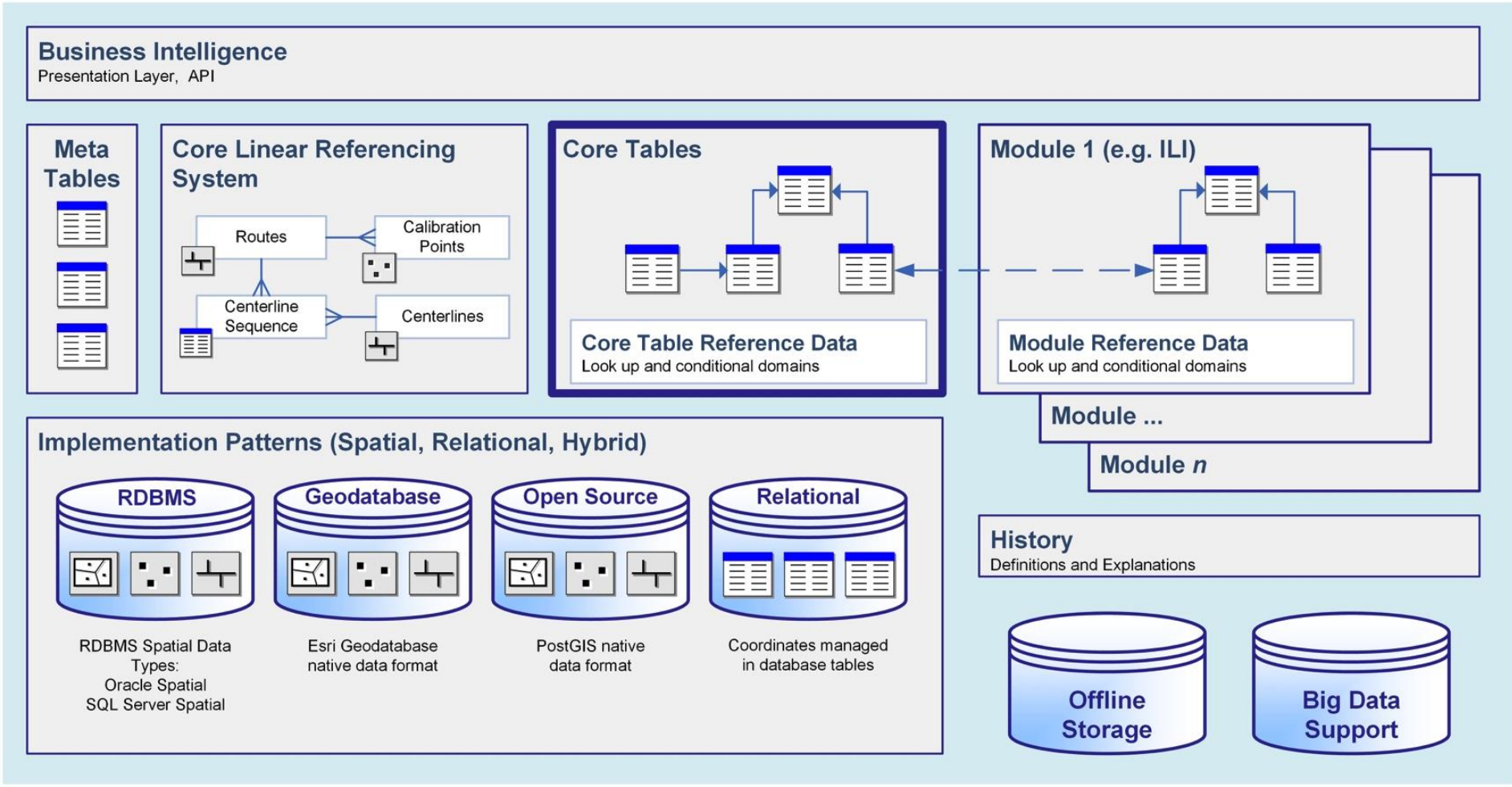
Part 3 – PODS LITE (Part of PODS Next Gen)

Documentation
- Logical Data Model
- Data Dictionary
- Governance (Model Usage, Editing Standards, Data Content Specification)

Reference Modes
- Continuous (2d/3d),
- Interrupted (2d/3d)
- Referent Point and Offset (Milepost)
- XYZ
- Odometer

Software
- Schema / Template Creation
- QA/QC
- Module Data Loader
- Module approval and documentation

Best Practices
- Schema/Module Specification, Definition, Creation and Validation
- App Specification, Creation and Validation
- Managing History, Work Orders, Documents, Re-Routes and Activities
- Feature/Condition Provenance
- Managing Metadata
- Module Management
- Data Exchange



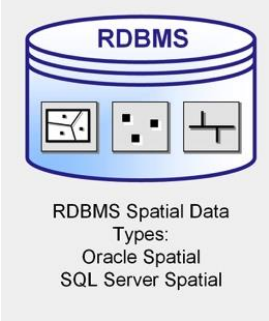
Part 3 – PODS LITE (Implementation Pattern)

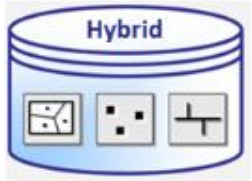
An implementation pattern is the sum of the following inputs for implementing PODS Next Gen database including the database type, the spatial storage type, and how edits are performed/managed.

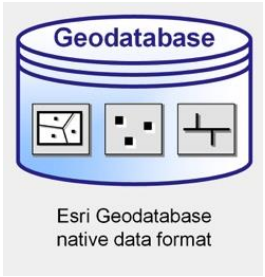
Implementation Pattern Focuses on:

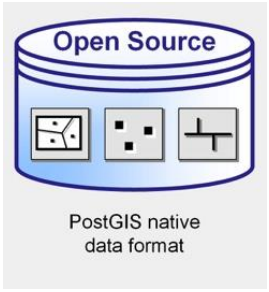
- Database Platform - support for multiple RDBMS (Oracle, SQL Server, PostGres)
- Spatial Data Storage Type - storing spatial data through native types (Geometry, SDO_Geometry, ST_Geometry) or the geodatabase (SDE_Geometry, ST_Geometry), or OGS WKT LineString
- Editing Paradigm – Versioned, ArcObjects, Native SQL

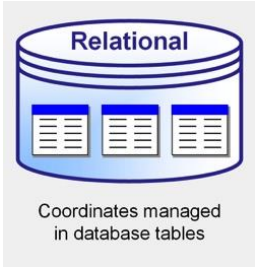
Part 3 – PODS LITE (Implementation Patterns)





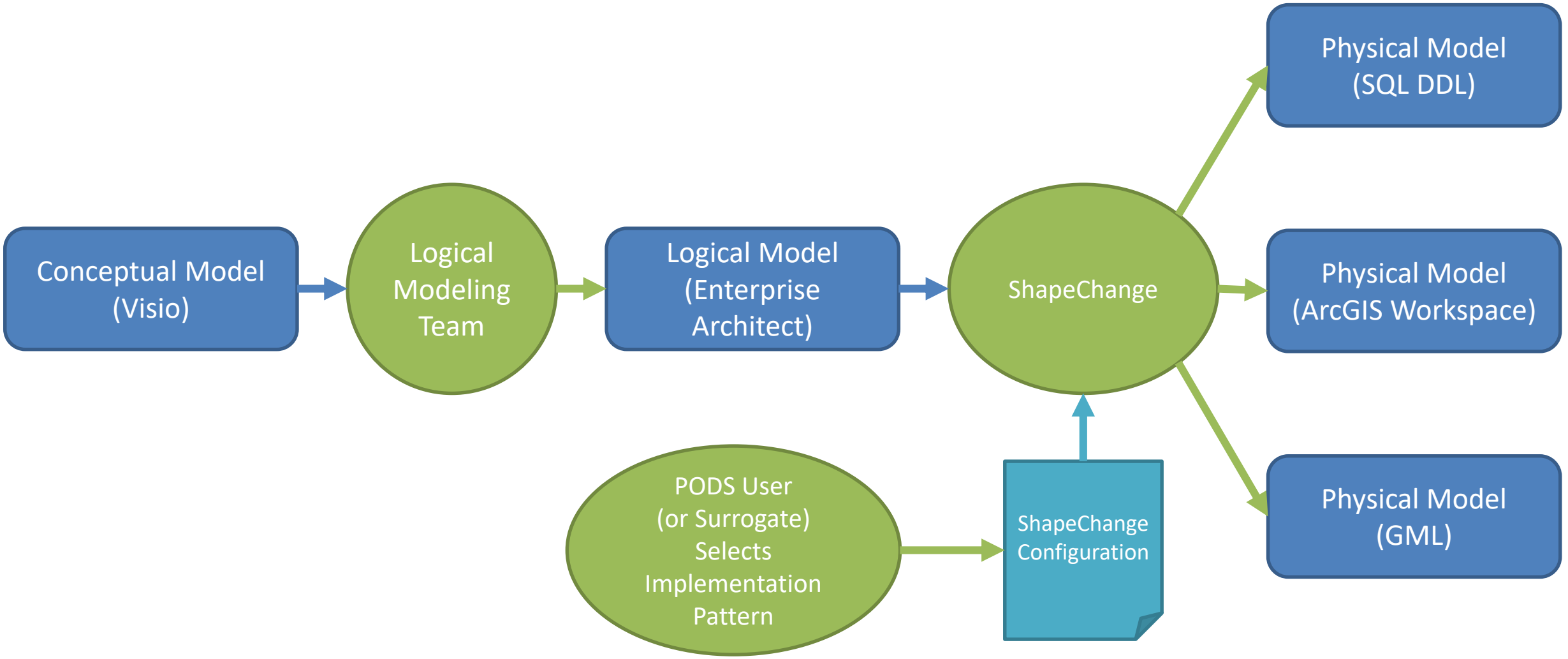






- **RDBMS** – Oracle/SQLServer/PostGIS, with APR, Shape column in ST_Geometry, Spatial SQL Library used to update data, Non-Versioned
- **Hybrid** - Oracle/SQLServer/PostGIS, ArcSDE to store APR (centerline as feature class), Event/Feature tables stored in Relational Tables, Shape columns in either SDO or ST_Geometry, Editing is both ArcObjects (Centerline), and SQL (Event/Feature Tables), Partially Versioned
- **Geodatabase** – Oracle/SQLServer/PostGIS, APR & Event/Feature data stored as feature classes, Shape column store as SDO or ST_Geometry, fully Versioned
- **Open Source** – PostGIS, APR & Event/Feature data stored as relational tables, ST_Geometry (native spatial type), not Versioned.
- **Relational** - Oracle/SQLServer/PostGIS, APR & Event/Feature data stored as relational tables, OGC WKT Linestring in Text/VarChar

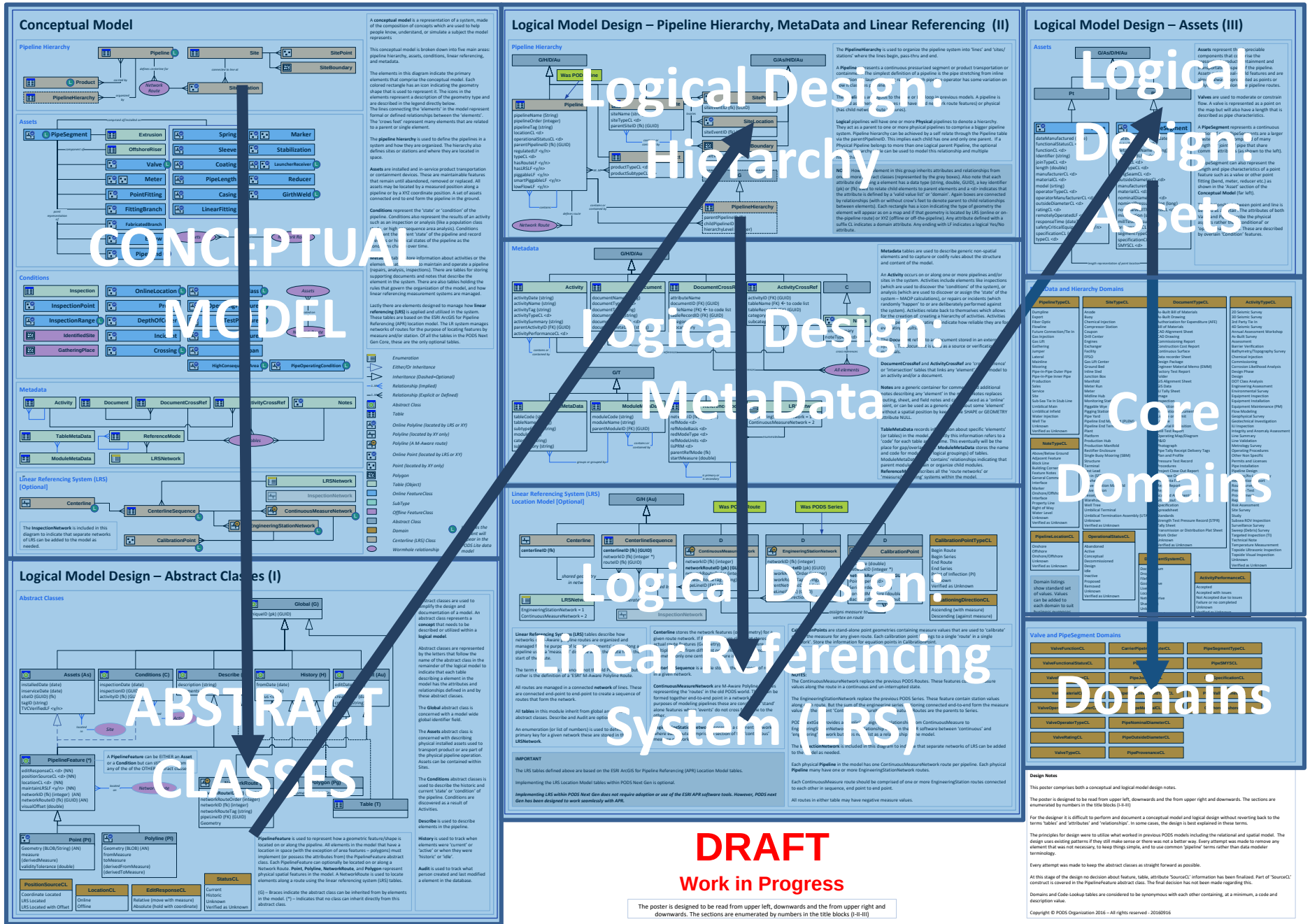
Part 3 – PODS LITE (Modeling Approach)



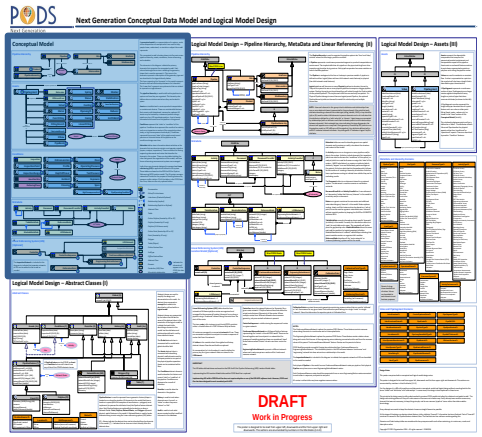
PODS Next Generation

Next Generation Conceptual Data Model and Logical Model Design

- Conceptual Model
- Model Sections
- Reading the Drawing

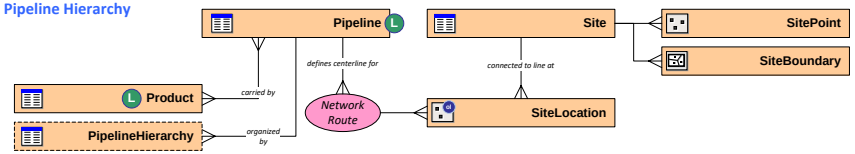


- Conceptual Model

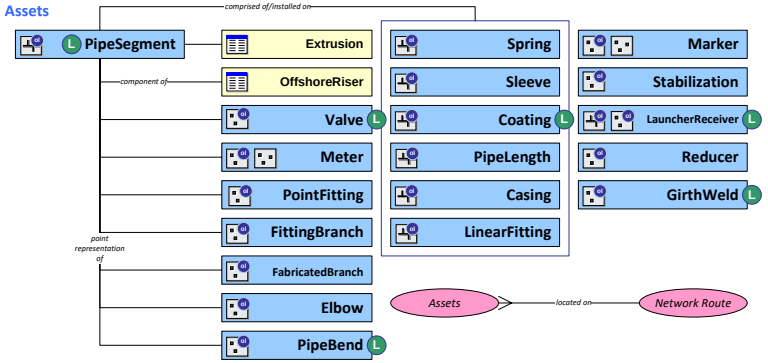


Conceptual Model

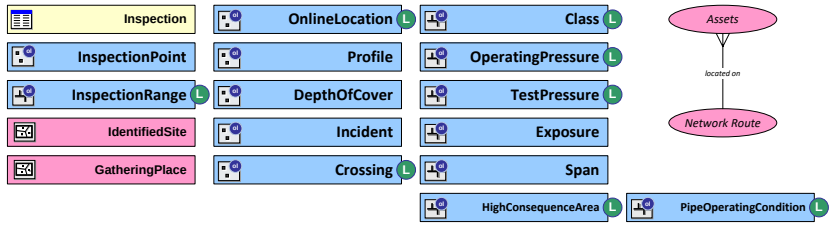
Pipeline Hierarchy



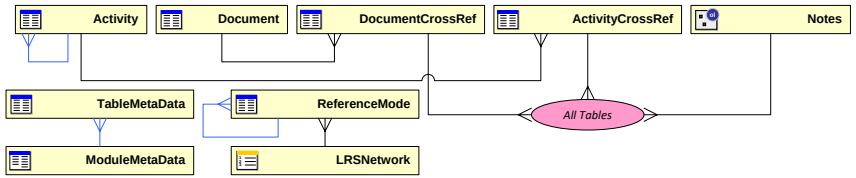
Assets



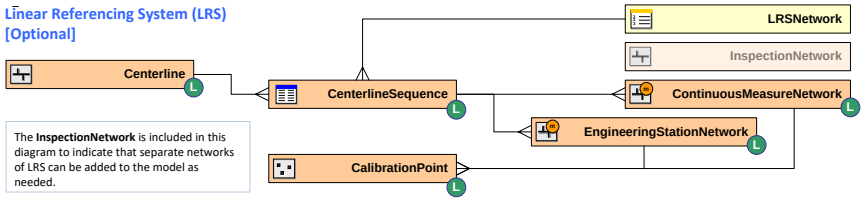
Conditions



Metadata



Linear Referencing System (LRS) [Optional]



The **InspectionNetwork** is included in this diagram to indicate that separate networks of LRS can be added to the model as needed.

A **conceptual model** is a representation of a system, made of the composition of concepts which are used to help people know, understand, or simulate a subject the model represents

This conceptual model is broken down into five main areas: pipeline hierarchy, assets, conditions, linear referencing, and metadata.

The elements in this diagram indicate the primary elements that comprise the conceptual model. Each colored rectangle has an icon indicating the geometry shape that is used to represent it. The icons in the elements represent a description of the geometry type and are described in the legend directly below.
The lines connecting the 'elements' in the model represent formal or defined relationships between the 'elements'. The 'crows feet' represent many elements that are related to a parent or single element.

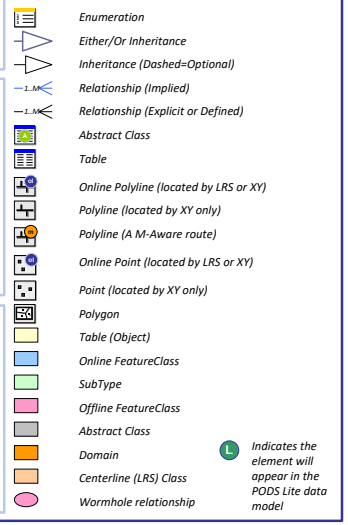
The **pipeline hierarchy** is used to define the pipelines in a system and how they are organized. The hierarchy also defines sites or stations and where they are located in space.

Assets are installed and in-service product transportation or containment devices. These are maintainable features that remain until abandoned, removed or replaced. All assets may be located by a measured position along a pipeline or by a XYZ coordinate position. A set of assets connected end to end form the pipeline in the ground.

Conditions represent the 'state' or 'condition' of the pipeline. Conditions also represent the results of an activity such as an inspection or analysis (like a population class study, or high consequence area analysis). Conditions represent the current 'state' of the pipeline and record previous or historical states of the pipeline as the conditions change over time.

Metadata tables store information about activities or the elements that are done to maintain and operate a pipeline (repairs, analysis, inspections). There are tables for storing supporting documents and notes that describe the element in the system. There are also tables holding the rules that govern the organization of the model, and how linear referencing measurement systems are managed.

Lastly there are elements designed to manage how **linear referencing (LRS)** is applied and utilized in the system. These tables are based on the ESRI ArcGIS for Pipeline Referencing (APR) location model. The LR system manages networks of routes for the purpose of locating features by measure and/or station. Of all the tables in the PODS Next Gen Core, these are the only optional tables.

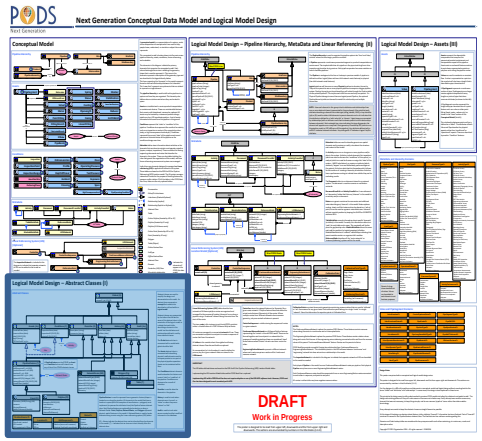


A conceptual model is a representation of a system, made of the composition of concepts which are used to help people know, understand, or simulate a subject the model represents.

Next Gen Design Principle:

The system of record for pipeline centerlines for all pressurized containment assets, for the transport of product, for the safe operation of the pipeline, and to mitigate the potential consequence of failure.

- Abstract Classes



Logical Model Design – Abstract Classes (I)

Abstract Classes

Global (G)
uniqueID (pk) (GUID)

Assets (As)
installedDate (date)
inserviceDate (date)
siteID (GUID) (fk)
tagID (string)
TVCVerifiedLF <y/n>

Conditions (C)
inspectionDate (date)
inspectionID (GUID) (fk)
activityID (fk) (GUID)

Describe (D)
description (string)
comments (string)

History (H)
fromDate (date)
toDate (date)
status <d>

Audit (Au)
editDate (date)
editor (string)
createDate (date)
creator (string)

discovered by — *Activity*

located in — *Site*

PipelineFeature (*)
editResponseCL <d> (NN)
positionSourceCL <d> (NN)
locationCL <d> (NN)
maintainLRSLF <y/n> (NN)
networkID (fk) (integer) (AN)
networkRouteID (fk) (GUID) (AN)
visualOffset (double)

A **PipelineFeature** can be EITHER an **Asset** or a **Condition** but can still inherit from any of the of the OTHER abstract classes

located on — *Network Route*

NetworkRoute (NR)
networkRouteID (pk) (GUID)
networkRouteOrder (integer)
networkID (fk) (integer)
networkRouteTag (string)
pipeLineID (FK) (GUID)
Geometry

Polygon (Pg)
Geometry (BLOB/String) (AN)

Table (T)

Point (Pt)
Geometry (BLOB/String) (AN)
measure
(derivedMeasure)
validityTolerance (double)

Polyline (Pl)
Geometry (BLOB) (AN)
fromMeasure
toMeasure
(derivedFromMeasure)
(derivedToMeasure)

PositionSourceCL
Coordinate Located
LRS Located
LRS Located with Offset

LocationCL
Online
Offline

EditResponseCL
Relative (move with measure)
Absolute (hold with coordinate)

StatusCL
Current
Historic
Unknown
Verified as Unknown

PipelineFeature is used to represent how a geometric feature/shape is located on or along the pipeline. All elements in the model that have a location in space (with the exception of area features – polygons) must implement (or possess the attributes from) the PipelineFeature abstract class. Each PipelineFeature can optionally be located on or along a Network Route. **Point**, **Polyline**, **NetworkRoute**, and **Polygon** represent physical spatial features in the model. A NetworkRoute is used to locate elements along a route using the linear referencing system (LRS) tables.

(G) – Braces indicate the abstract class can be inherited from by elements in the model. (*) – Indicates that no class can inherit directly from this abstract class.

Abstract classes are used to simplify the design and documentation of a model. An abstract class represents a **concept** that needs to be described or utilized within a **logical model**.

Abstract classes are represented by the letters that follow the name of the abstract class in the remainder of the logical model to indicate that each table describing a element in the model has the attributes and relationships defined in and by these abstract classes.

The **Global** abstract class is concerned with a model wide global identifier field.

The **Assets** abstract class is concerned with describing physical installed assets used to transport product or are part of the physical pipeline operation. Assets can be contained within Sites.

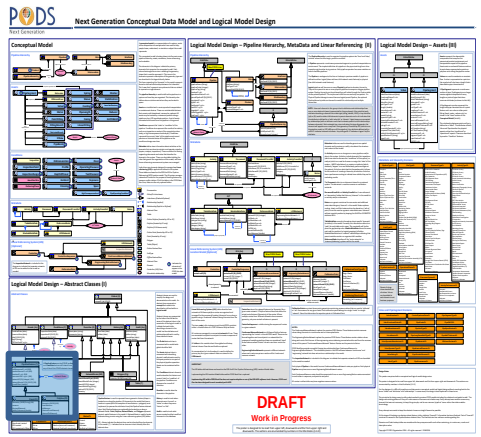
The **Conditions** abstract classes is used to describe the historic and current ‘state’ or ‘condition’ of the pipeline. Conditions are discovered as a result of Activities.

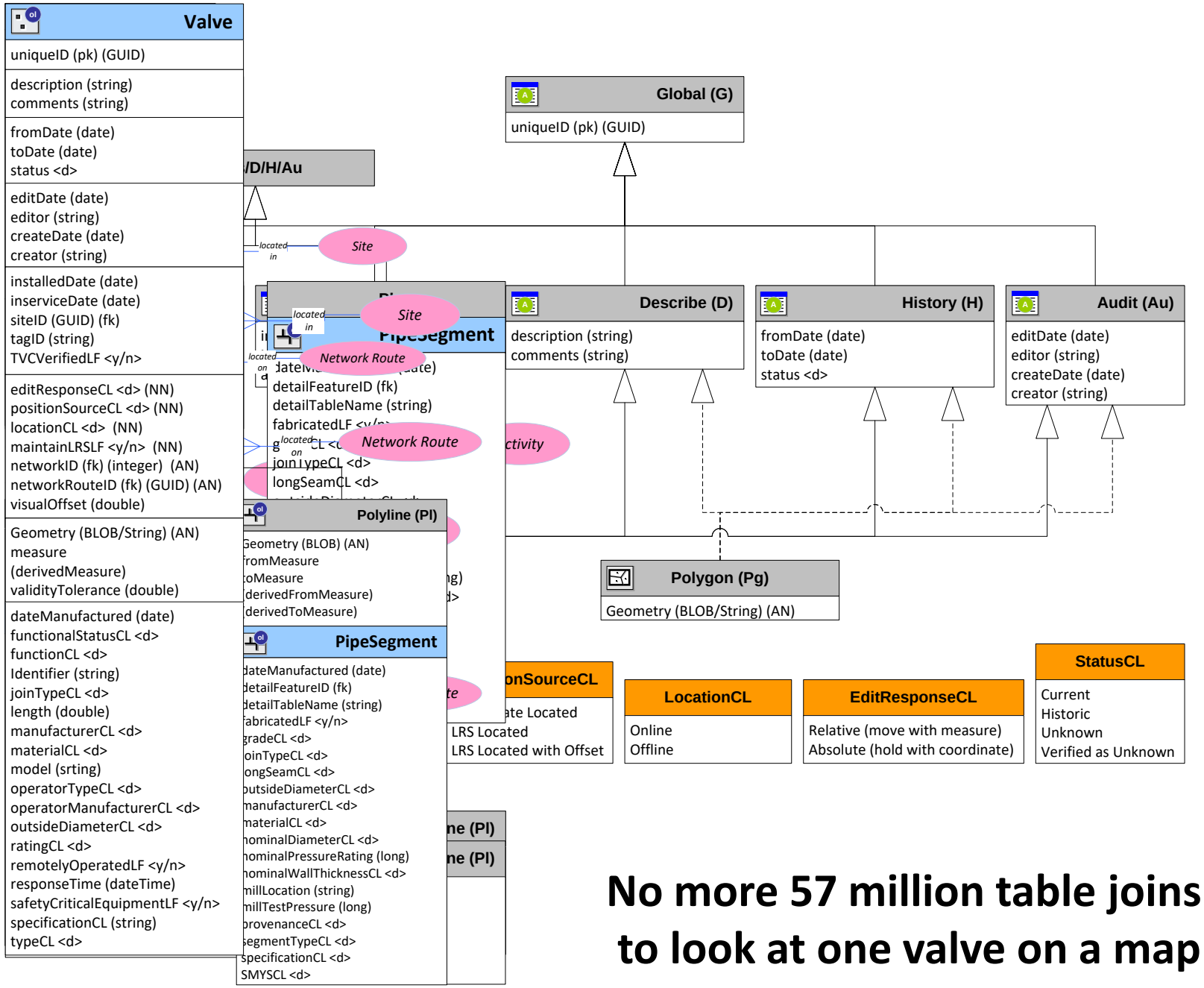
Describe is used to describe elements in the pipeline.

History is used to track when elements were ‘current’ or ‘active’ or when they were ‘historic’ or ‘idle’.

Audit is used to track what person created and last modified a element in the database.

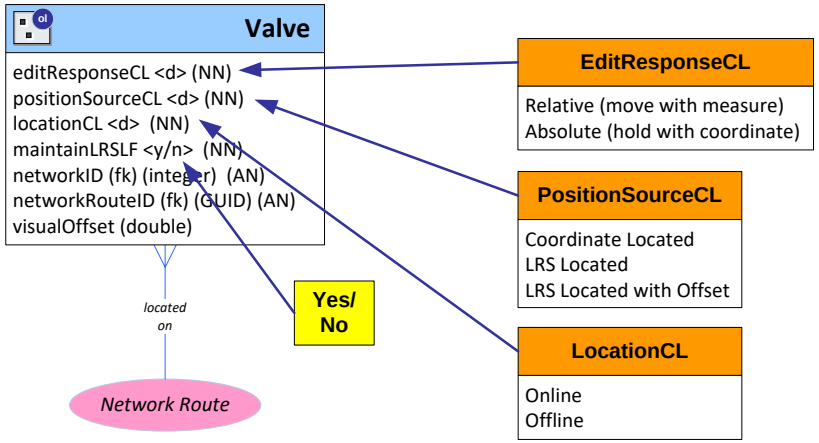
- Implementation of abstract classes to physical tables/ FeatureClasses

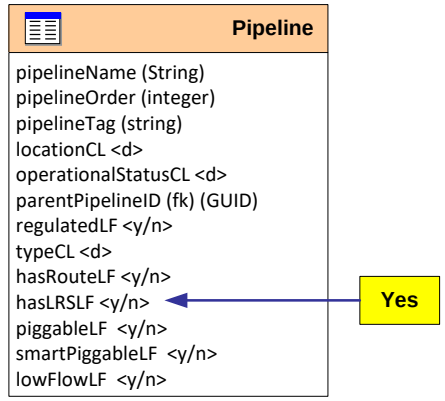


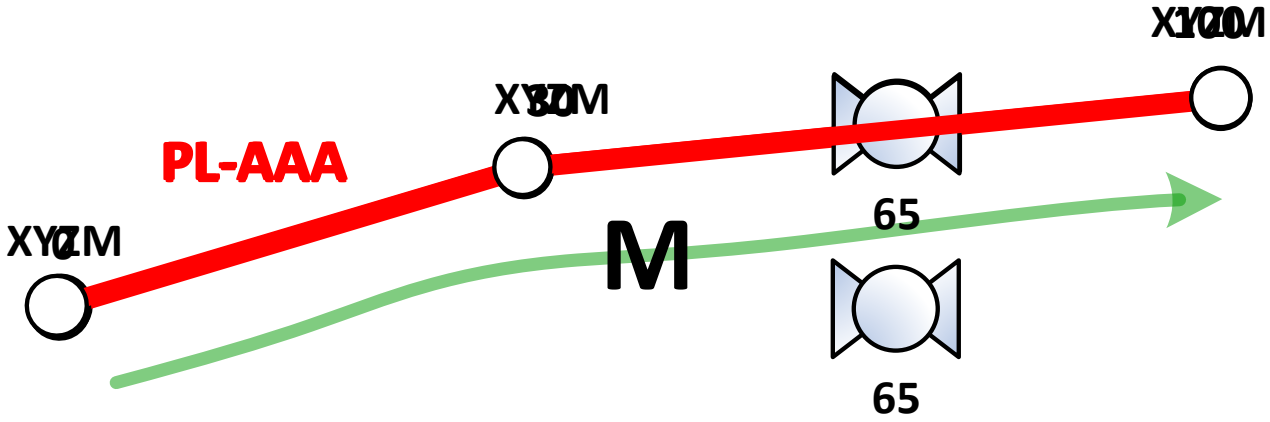


No more 57 million table joins to look at one valve on a map

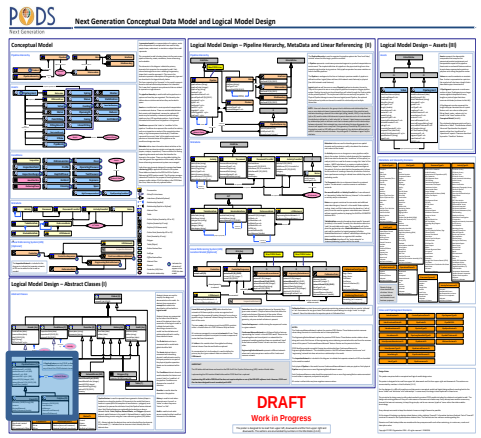
- HasLRSLF ?
- Coordinate based or LRS-based?



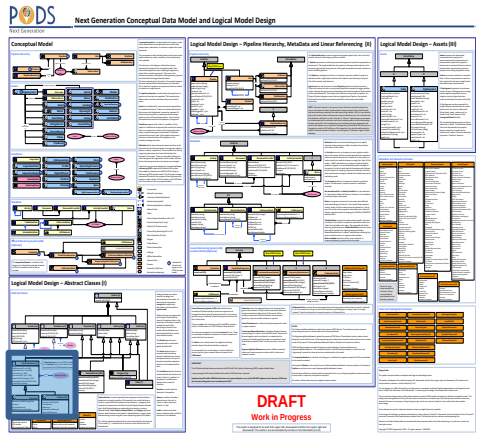


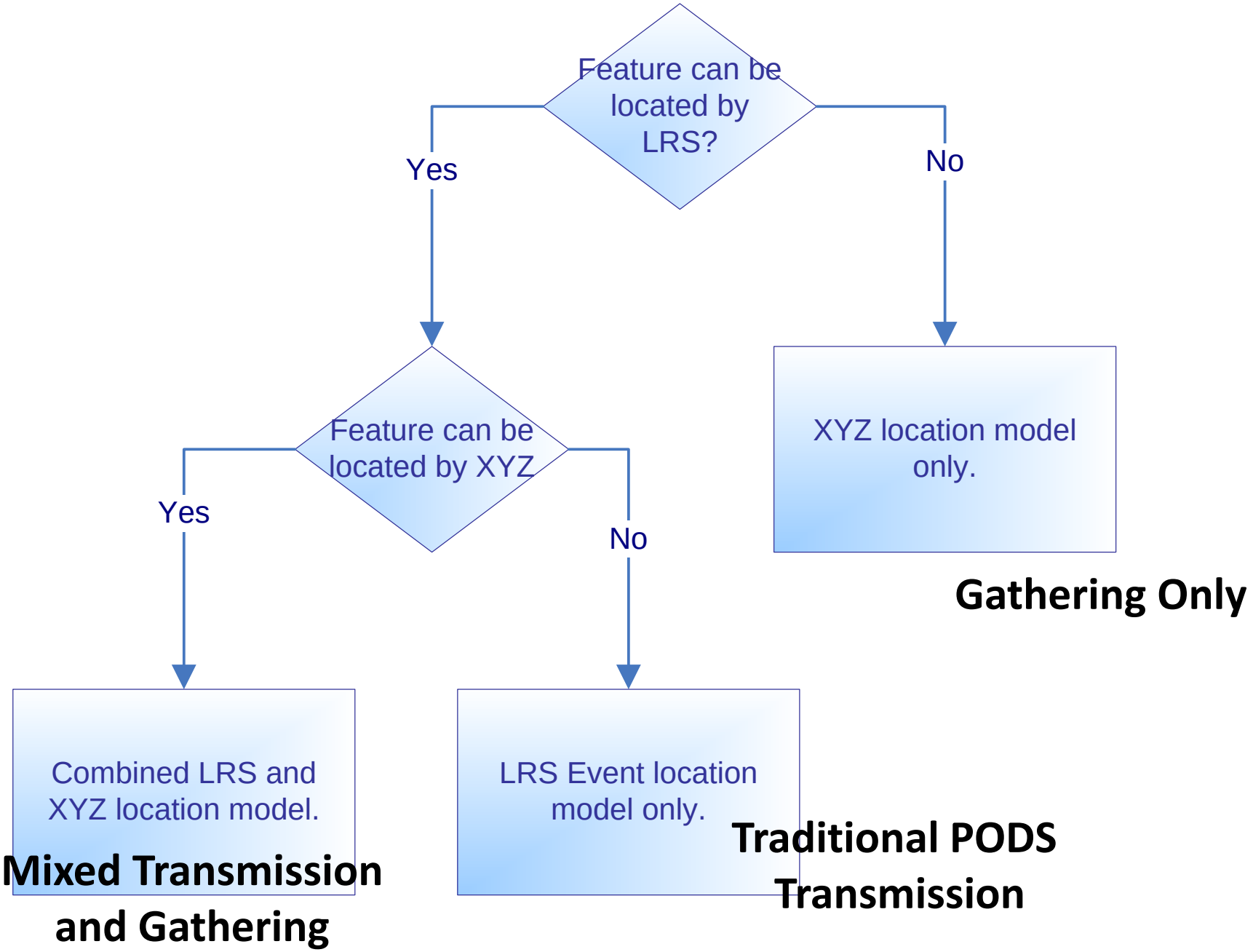


Valve			
editResponseCL <d> (NN)	positionSourceCL <d> (NN)	locationCL <d> (NN)	maintainLRSLF <y/n> (NN)
Absolute	Coordinate Located	Offline	Yes

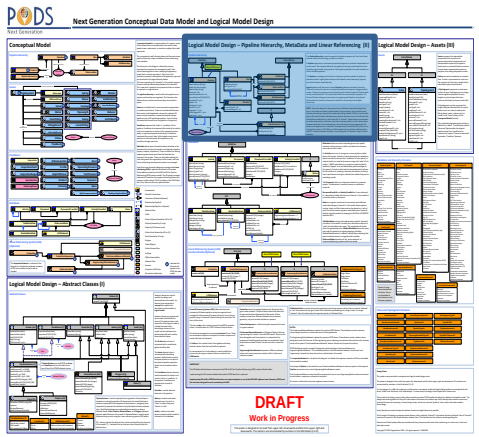


- LRS & Geometry Data Editing Patterns at the **FEATURE** level





- Logical Design:
  - Hierarchy



Pipeline Hierarchy

G/H/D/Au

Was PODS Line

G/As/H/D/Au

Pipeline

pipelineName (String)
pipelineOrder (integer)
pipelineTag (string)
locationCL <d>
operationalStatusCL <d>
parentPipelineID (fk) (GUID)
regulatedLF <y/n>
typeCL <d>
hasRouteLF <y/n>
hasLRSLF <y/n>
piggableLF <y/n>
smartPiggableLF <y/n>
lowFlowLF <y/n>

Site

siteName (string)
siteTypeCL <d>
parentSiteID (fk) (GUID)

contains or contained by

locates

SitePoint

siteEventID (fk) (GUID)

SiteLocation

siteEventID (fk) (GUID)

SiteBoundary

siteEventID (fk) (GUID)

Product

productTypeCL <d>
productSubtypeCL <d>

M..M

contains

defines route

Network Route

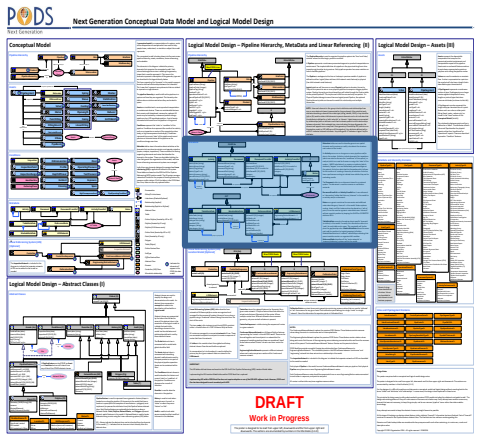
contains or contained by

G

PipelineHierarchy

parentPipelineID (fk)
childPipelineID (fk)
hierarchyLevel (integer)

- Logical Design:
  - MetaData



Metadata

G/H/D/Au

Activity
activityDate (string)
activityName (string)
activityTag (string)
activityTypeCL <d>
activitySummary (string)
parentActivityID (FK) (GUID)
activityPerformanceCL <d>

Document
documentName (string)
documentTypeCL <d>
documentTag (string)
documentPath (string)
documentSystemCL <d>
documentMetaData (string)

DocumentCrossRef
attributeName
documentID (FK) (GUID)
tableName (FK) ← to code list
tableRecordID (FK) (GUID)
category
subcategory

ActivityCrossRef
activityID (FK) (GUID)
tableName (FK) ← to code list
tableRecordID (FK) (GUID)
category
subcategory

C

Notes
noteSummary
noteTypeCL <d>

contains or contained by
cross references
cross references
cross references
cross references
All elements

G/T

TableMetaData
tableCode (string)
tableName (string)
subtypeFilter (string)
moduleID (fk)
category (string)
subcategory (string)

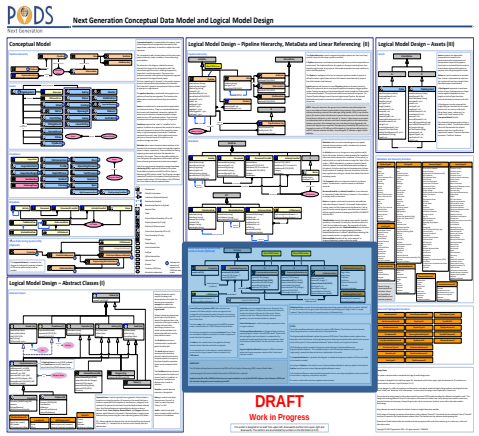
ModuleMetaData
moduleCode (string)
moduleName (string)
parentModuleID (FK) (GUID)

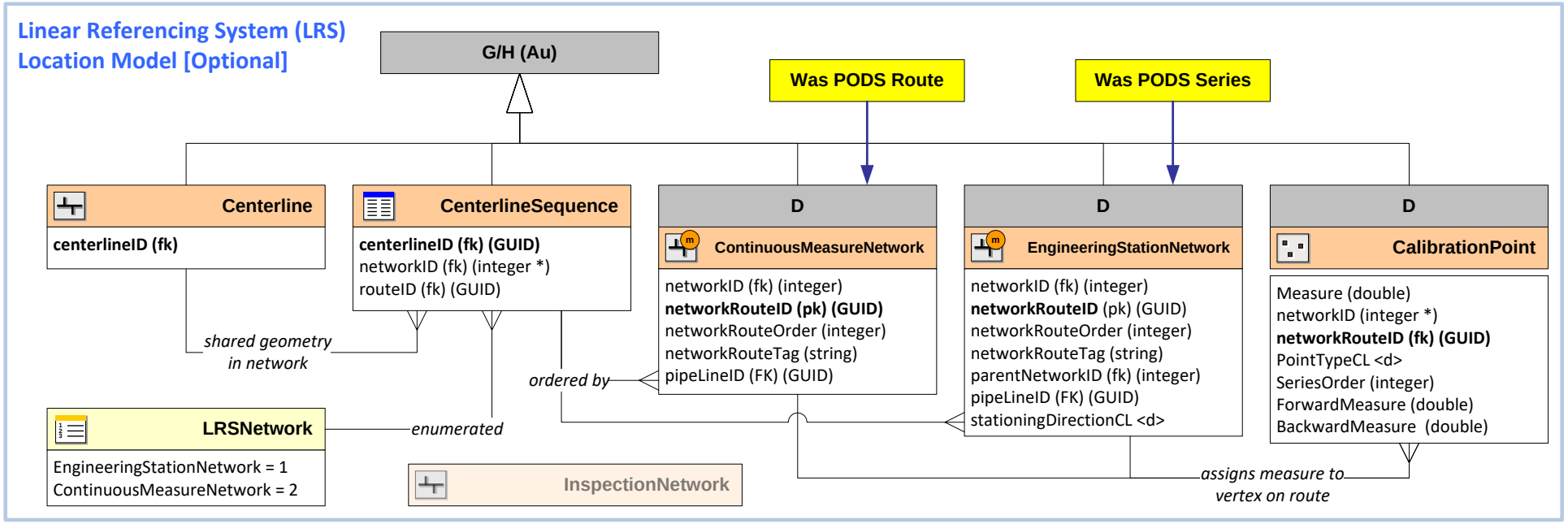
ReferenceMode
networkID (fk) (integer)
refMode <d>
refModeBasis <d>
redModeType <d>
refModeUnits <d>
IisPRM <d>
parentRefMode (fk)
startMeasure (double)

LRSNetwork
EngineeringStationNetwork = 1
ContinuousMeasureNetwork = 2

groups or grouped by
contains or contained by
enumerated
is primary or is secondary

- Logical Design:
  - Linear Referencing System (LRS)





CalibrationPointTypeCL

Begin Route
Begin Series
End Route
End Series
Point of Inflection (PI)
Unknown
Verified as Unknown

StationingDirectionCL

Ascending (with measure)
Descending (against measure)

IMPORTANT

The LRS tables defined above are based on the ESRI ArcGIS for Pipeline Referencing (APR) Location Model tables.

Implementing the LRS Location Model tables within PODS Next Gen is optional.

Implementing LRS within PODS Next Gen does not require adoption or use of the ESRI APR software tools. However, PODS next Gen has been designed to work seamlessly with APR.

Observations

- It is all about data, the systems that store and present data
  - I either don't have the data or I have too much data or I don't trust my data (for sure I don't trust your data)
  - Data is Siloed (lost, not digital, within organizations, between organizations)
- Data Accuracy is Missing
  - Focus on data accuracy for the most important part of the system - The Centerline and the Assets
- Geography becomes a beautiful thing (planar geometry and coordinates)
  - Allows for spatial interactions
  - But not always business interactions
- Location and Tags are the common reference systems
  - If I have a coordinate I can locate something in space
  - If I have a tag then I can tie two seemingly disparate pieces of information together

Observations

- True Operational and Data Excellence
  - Build and maintain the system of record
  - Turn everyone into data hunters, gatherers and consumers
  - Notify people when things change
  - Provide constant access to ‘what is here’ information
- Commoditize of Information Technology (Availability)
  - “Put your stuff in the cloud because we are more tired of your IT people than you are”
  - “Our IT department would never allow us to do that …”
  - “More people work on security (for any cloud provider) then you have working for your entire company”

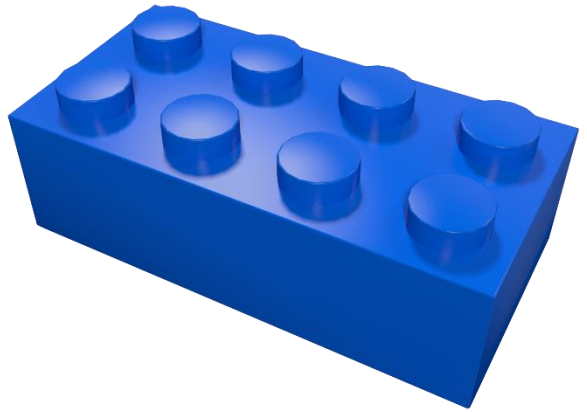
Observations

- Form a Data Sharing Alliance
  - Leverage the cloud to store and provide service based access to commonly shared and publically available data
  - Provide uncertainty level indexes to these data based on spatial position
  - Provide concurrency and provenance (pedigree) metadata for attribute data
  - Crowd-source data in shared corridors (line crossing, structures)
  - Work with industry groups (API, INGAA, GTI, NACE, CGA) to support and fund (host) these initiatives
  - This is where standards come into play
  - Charge modicum of cost for use of data services

Observations

- Doubt, there is a one-size-fits-all solution
- All, some or most of the parts, form the solution

BIG Data	Machine Learning	Business Intelligence
Cloud Technology	Services	Uncertainty Management
Data Warehouse	SME	Qualitative
PODS	Quantitive	GIS



Thank you

Questions?

Peter Veenstra
P: 816-820-7841
E: pveenstra@trcsolutions.com

TRC's Guiding Principles

Our Mission

We understand our clients' goals and embrace them as our own, applying creativity, experience, integrity and dedication to deliver superior solutions to the world's energy, environment and infrastructure challenges.

Our Vision

We will solve the challenges of making the Earth a better place to live – community by community and project by project.