The Pacific Northwest Smart Grid Demo

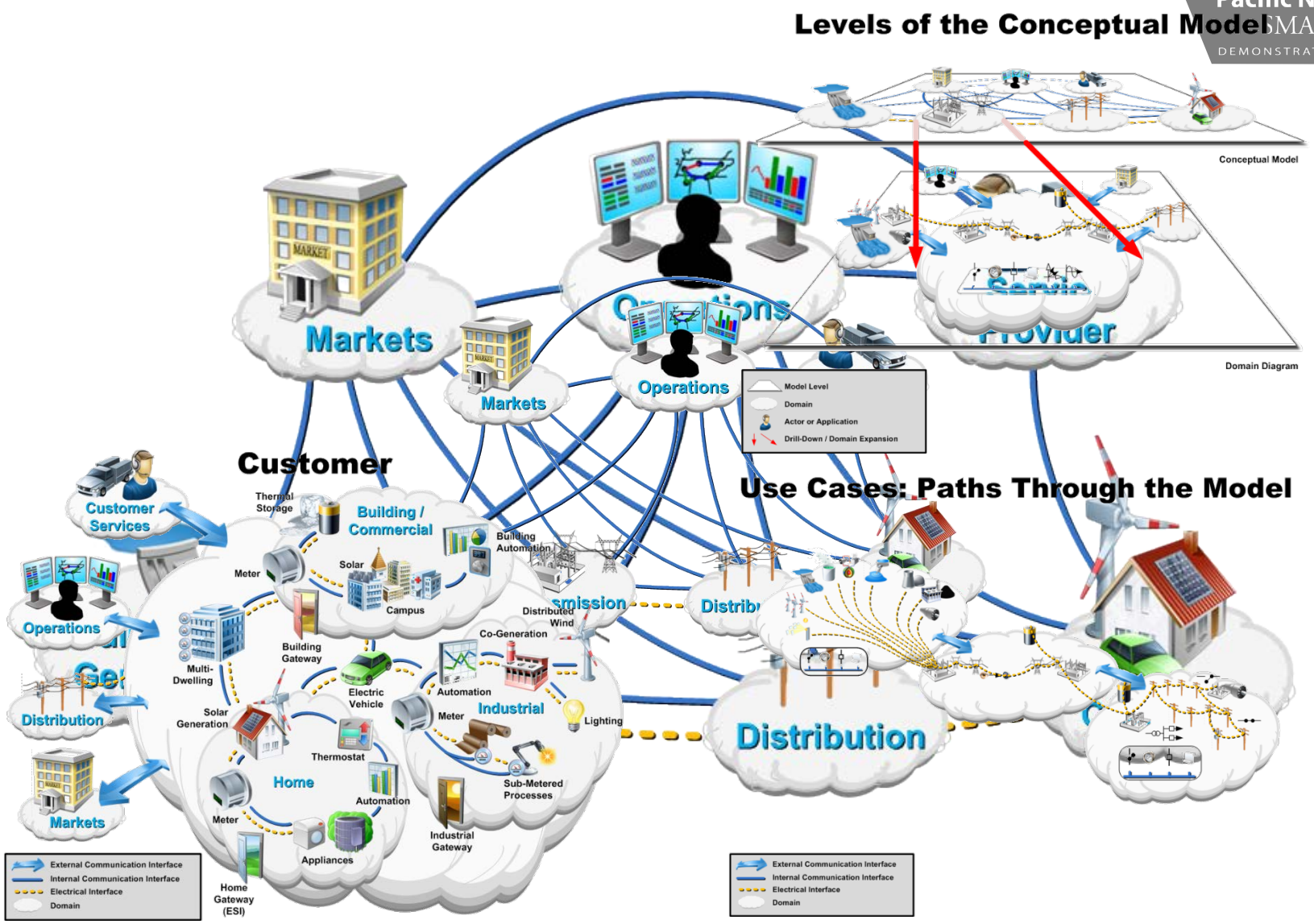
Transactive Control Implementation

Ron Ambrosio
Global Research Executive, Energy & Utilities Industry
Senior Technical Staff Member
IBM TJ Watson Research Center, Yorktown Heights, NY

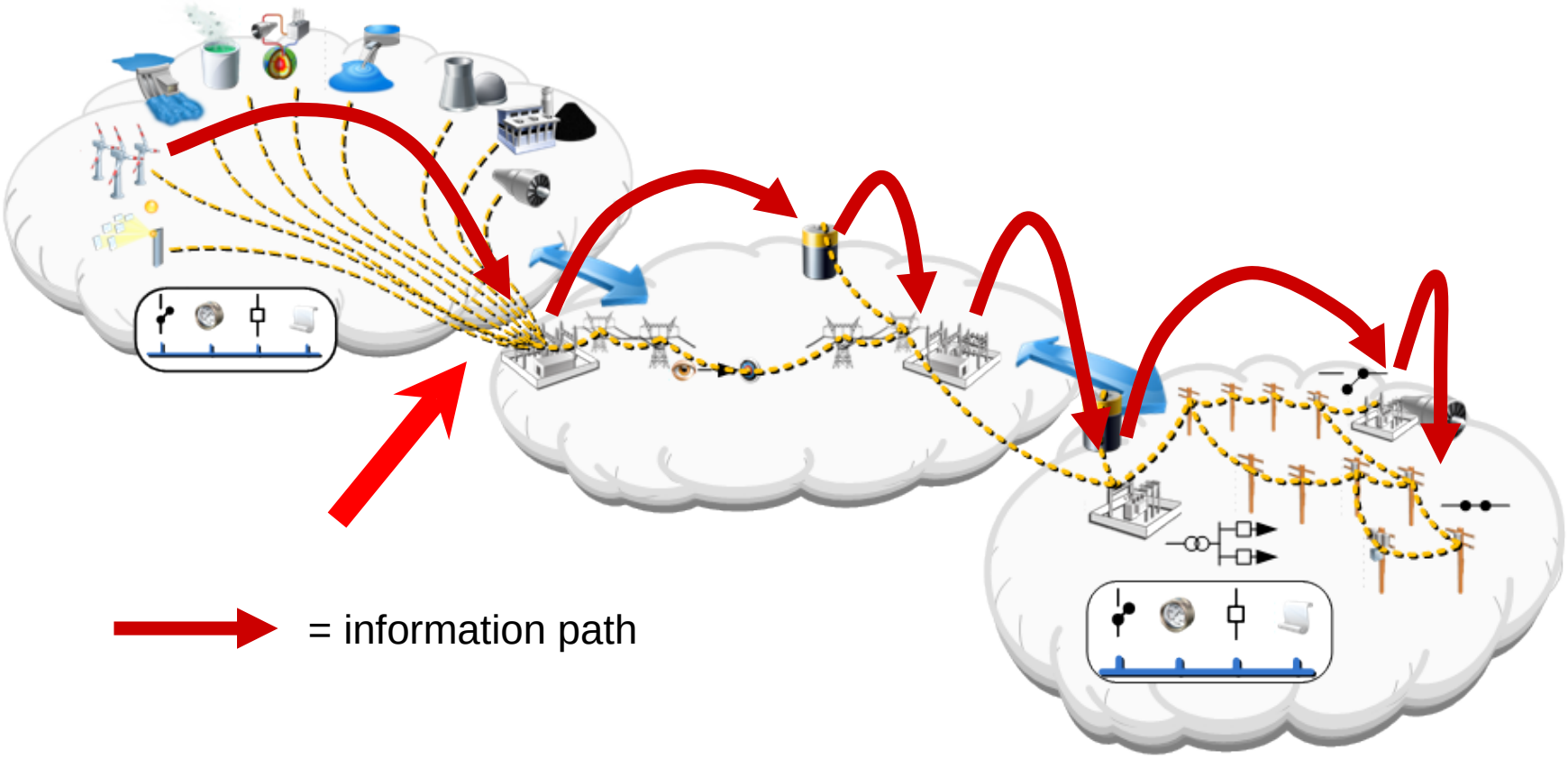
Chairman Emeritus & Member, U.S. Dept. of Energy GridWise Architecture Council
Chairman, U.S. National Inst. of Standards and Technology SGIP Architecture Committee
Convenor, ISO/IEC JTC 1 Special Working Group on Smart Grid

Let’s start at 100,000 feet

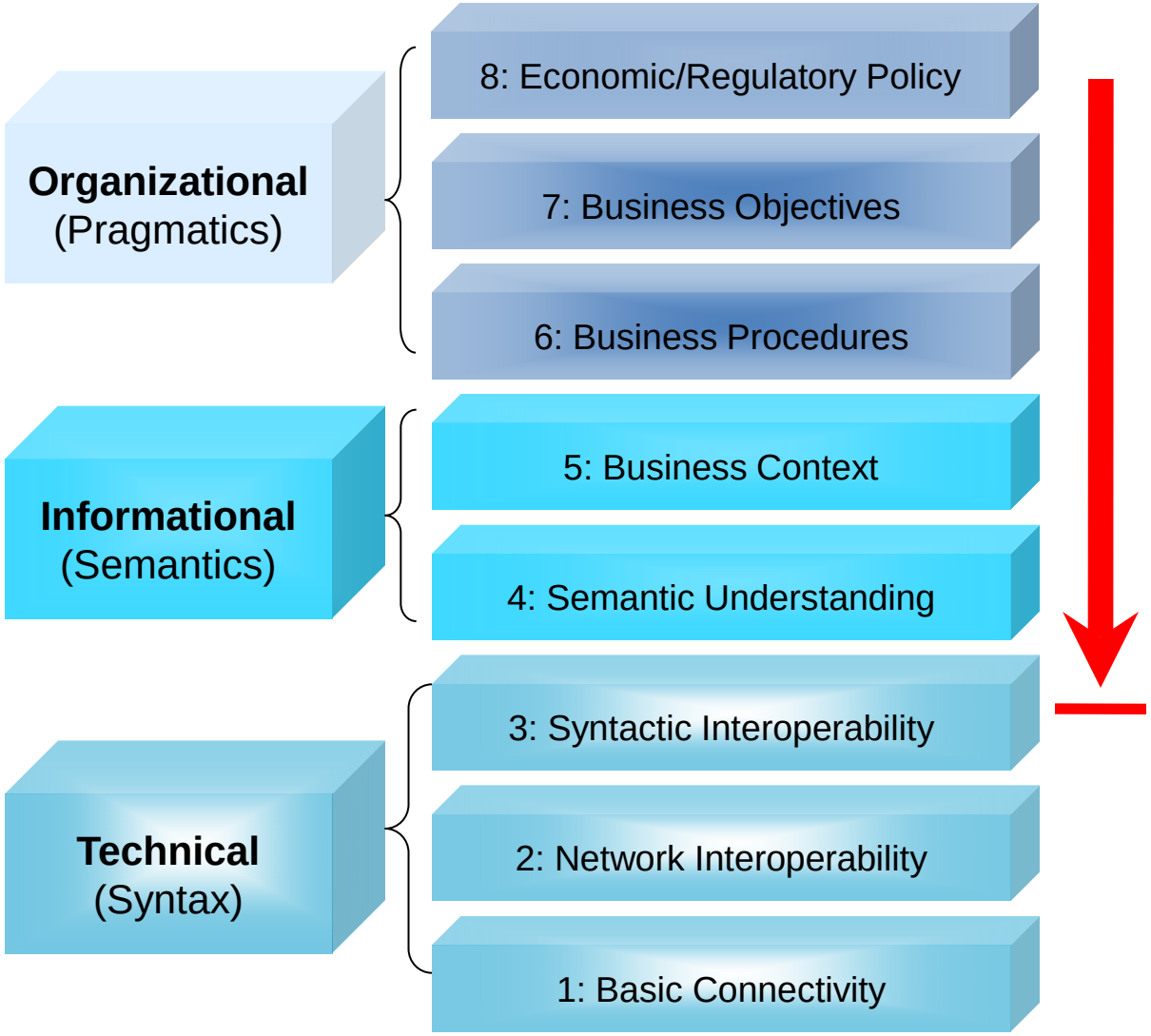
Smart Grid Conceptual Model



PNW Transactive Control Incentive Flow



The GWAC Stack and the Model



Analyze interoperability at key inter-system points in the use case paths through the Conceptual Model …

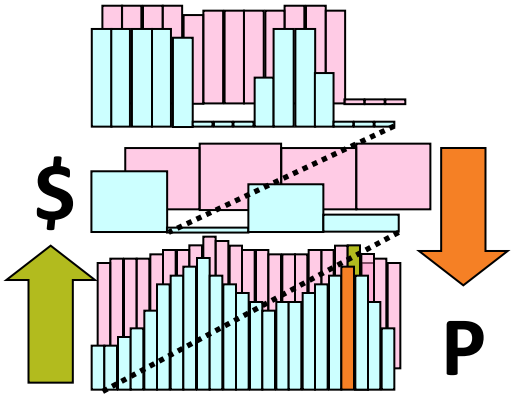
… using the GWAC Stack top-down to define lowest layer that must be addressed

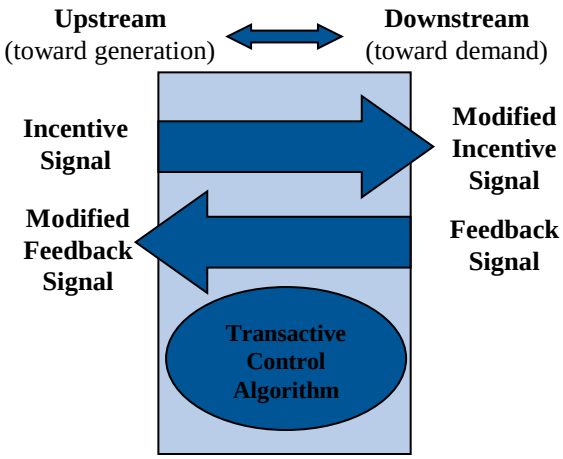
Transactive Control

Transactive Control of Energy Delivery

- What is it?
  - A flexible method for combing multiple objectives and constraints (both economic and operational) using uniform incentive and feedback signals throughout an electricity grid.
- Incentive and feedback signals
  - An incentive signal is used by electricity assets to make decisions about their future consumption pattern – it is a forward signal with information for the next few days.
  - A feedback signal represents that consumption pattern over the same few days, in response to the incentive.
- Creation of the incentive signal
  - Regional and local objectives are "monetized" and incorporated into the signal as it flows through each transactive control node in the system
  - Objectives could include:
    - Congestion Relief
    - Encouraging the use wind power
    - Reducing peak load
    - Reducing phase imbalance on a transformer
    - Avoiding overloading a distribution line

Signals forecast several days



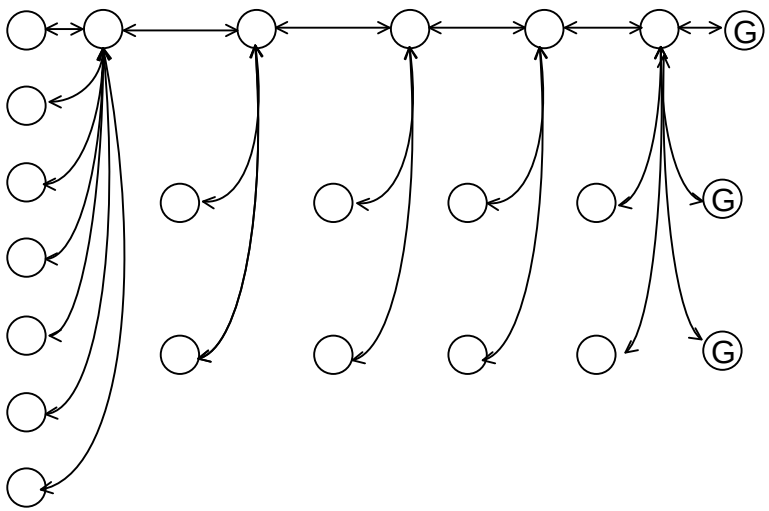


Some Definitions

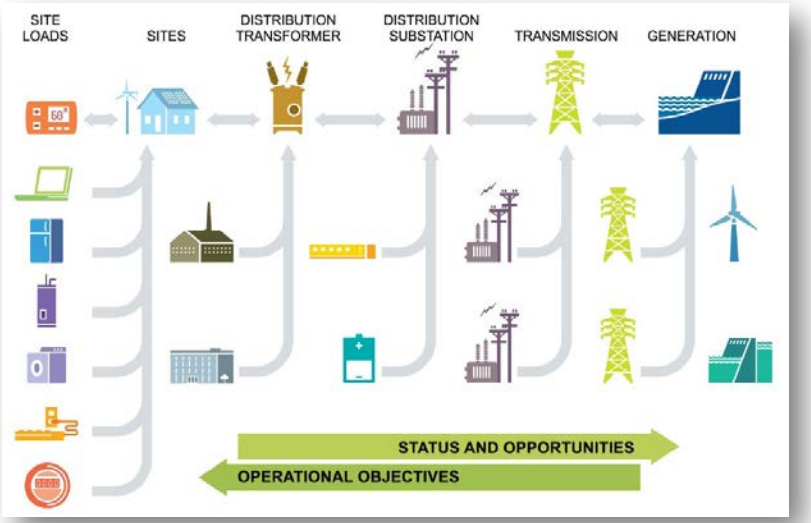
- Transactive Control
  - A single, integrated, smart grid incentive signaling approach utilizing an economic signal as the primary basis for communicating the desire to change the operational state of responsive assets.
- Transactive Incentive Signal (TIS)
  - A representation of the actual delivered cost of electric energy at a specific system location (e.g., at a transactive node). Includes both the current value and a forecast of future values.
- Transactive Feedback Signal (TFS)
  - A representation of the net electric load at a specific system location (e.g., at a transactive node). Includes both the current value and a forecast of future values.

Propagation of the incentive and feedback signals

Incentive signals and feedback signals propagate through an information network (the transactive control system) that overlays the electrical network



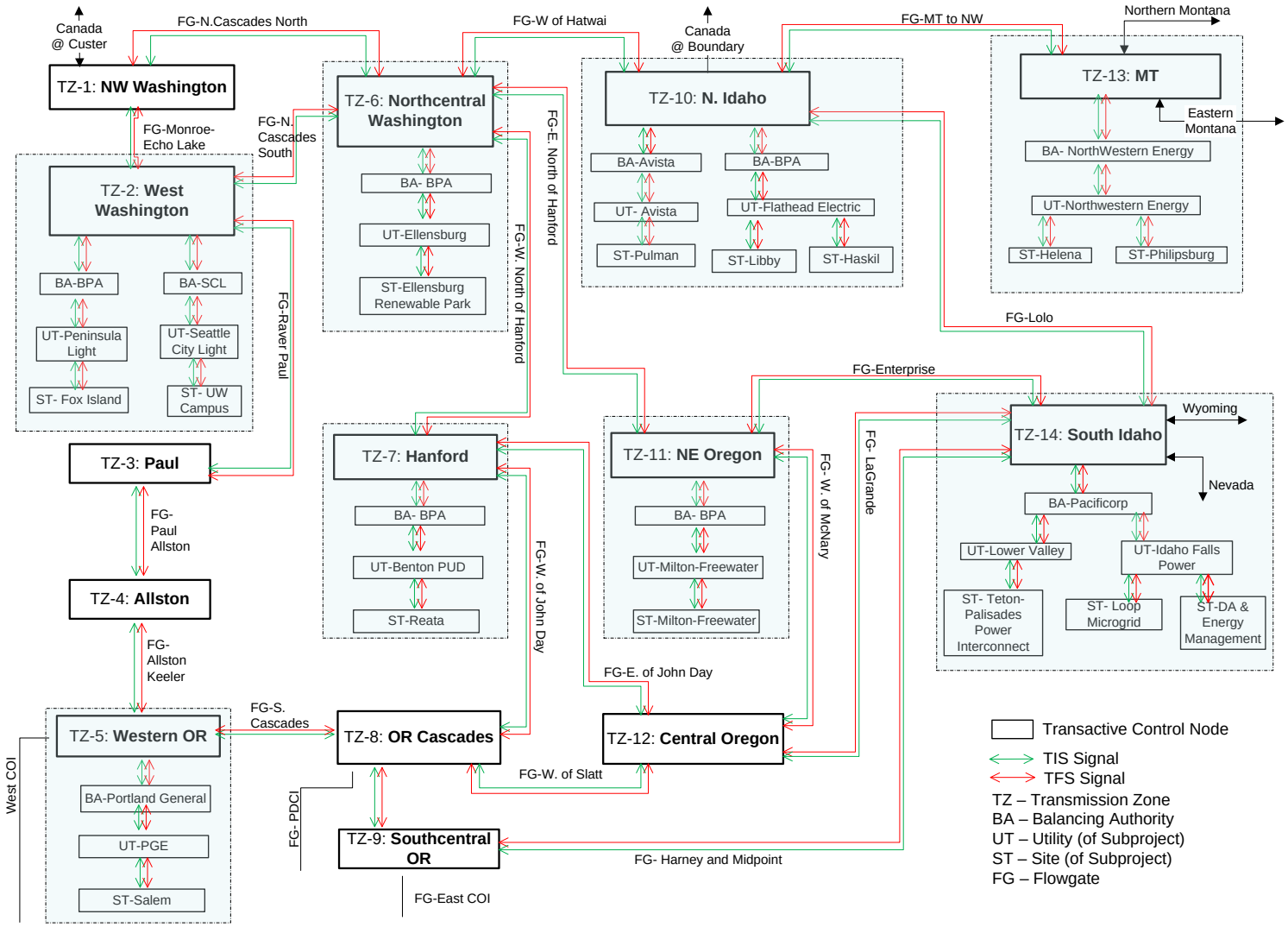
Information Network



Physical Network

Transactive Node Structure for Demo

Regional and Subproject Transactive Control Nodes & Network Topology



ISO/IEC 18012 interoperability framework

IBM Internet-scale Control Systems (iCS)

Now we're at 10,000 feet

Interoperability View

Pacific Northwest SMART GRID DEMONSTRATION PROJECT

Invest – Standardize

Semantic

Syntactic Interop

Find & accept off the shelf solutions

Network Interop

Basic Conn.

iCS* / TC Nodes

iCS Event Bus

Message Transports

WAN / IP

TIS / TFS flows between TC Nodes and is processed within TC Nodes

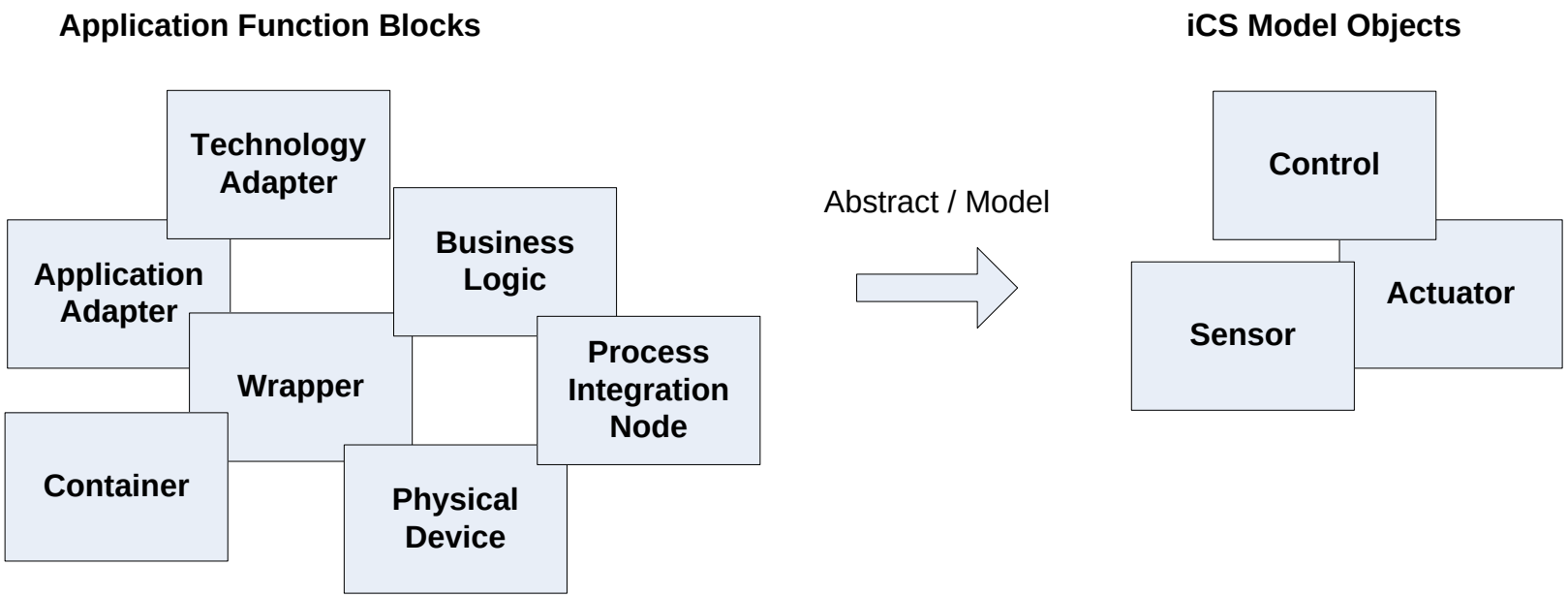
High level ability to transmit messages between nodes

Packaging (packetization) and physical transmittal of messages

* iCS (Internet-scale Control Systems) – IBM reference implementation of ISO/IEC 18012-2 draft international standard

Simple model of sensor/actuator/control objects is used to abstract different application and technology functions

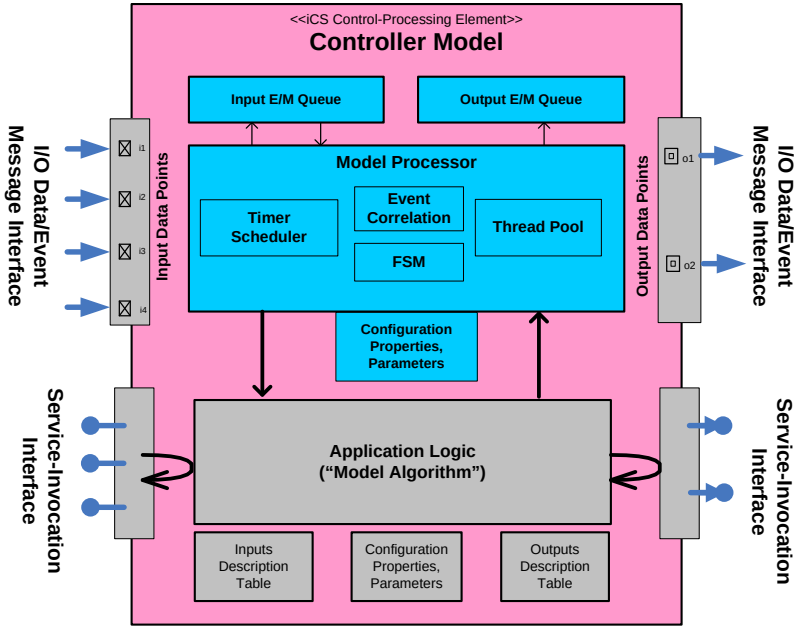
- iCS creates unified view and integration layer that
  - *can encapsulate different underlying application and technology functions*
  - uses XML in a declarative fashion to describe heterogeneous distributed control applications
  - uses runtime to manage lifecycle of objects (creation, initialization, binding, etc.)



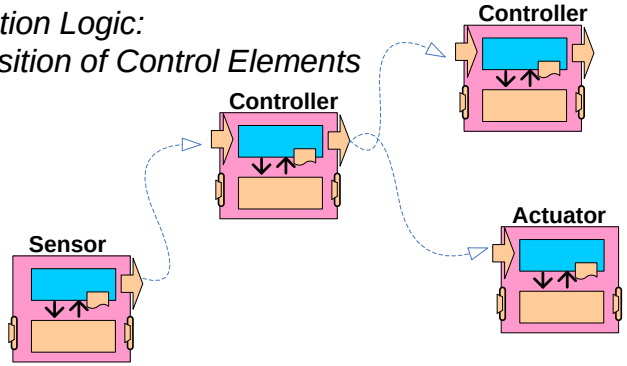
© 2009 IBM Corporation

Component Model: Control Element (iCS Building Block)

Construct of Control Element:



Application Logic:
Composition of Control Elements

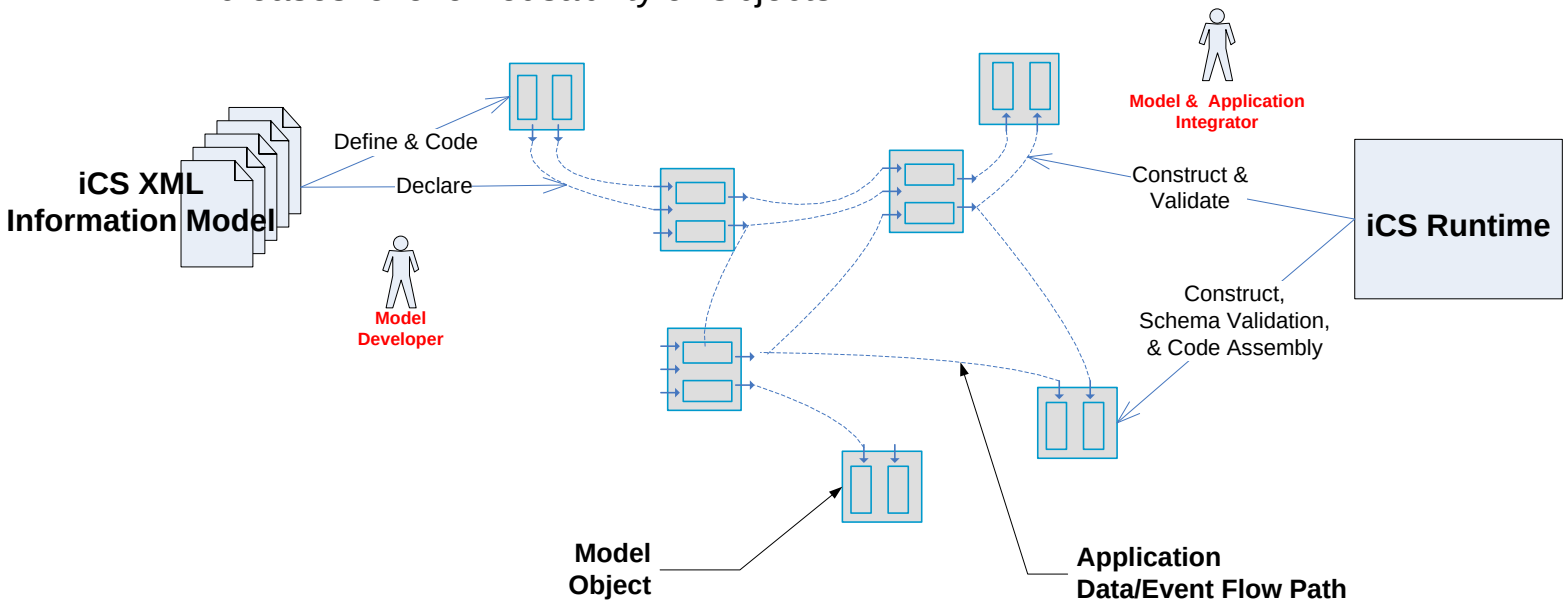


- Simple application abstraction spans from business processes to physical devices
- Runtime implementation scales from embedded to enterprise server environment
  - Designed and implemented with Java Micro Edition (ME) as minimum target
- ***Establishes a hybrid application programming model to integrate both enterprise (business) and embedded control (operational) domains***:
  - Asynchronous event programming
  - Service Oriented Architecture transactional programming

© 2009 IBM Corporation

Important to establish certain separations in the programming abstraction

- ***Separation of Object Implementation from Composition and Integration***
  - Minimizes interference between the two communities of programmers:
    - Component Developers
    - System Integrators
  - Allows event flow of application to be modified with no impact on Object Implementation
- ***Separation of application topology from device topology***
  - Minimizes impact of device evolution and reconfiguration
  - Simplifies reconfiguration of application for tuning, etc.
  - Increases level of reusability of Objects

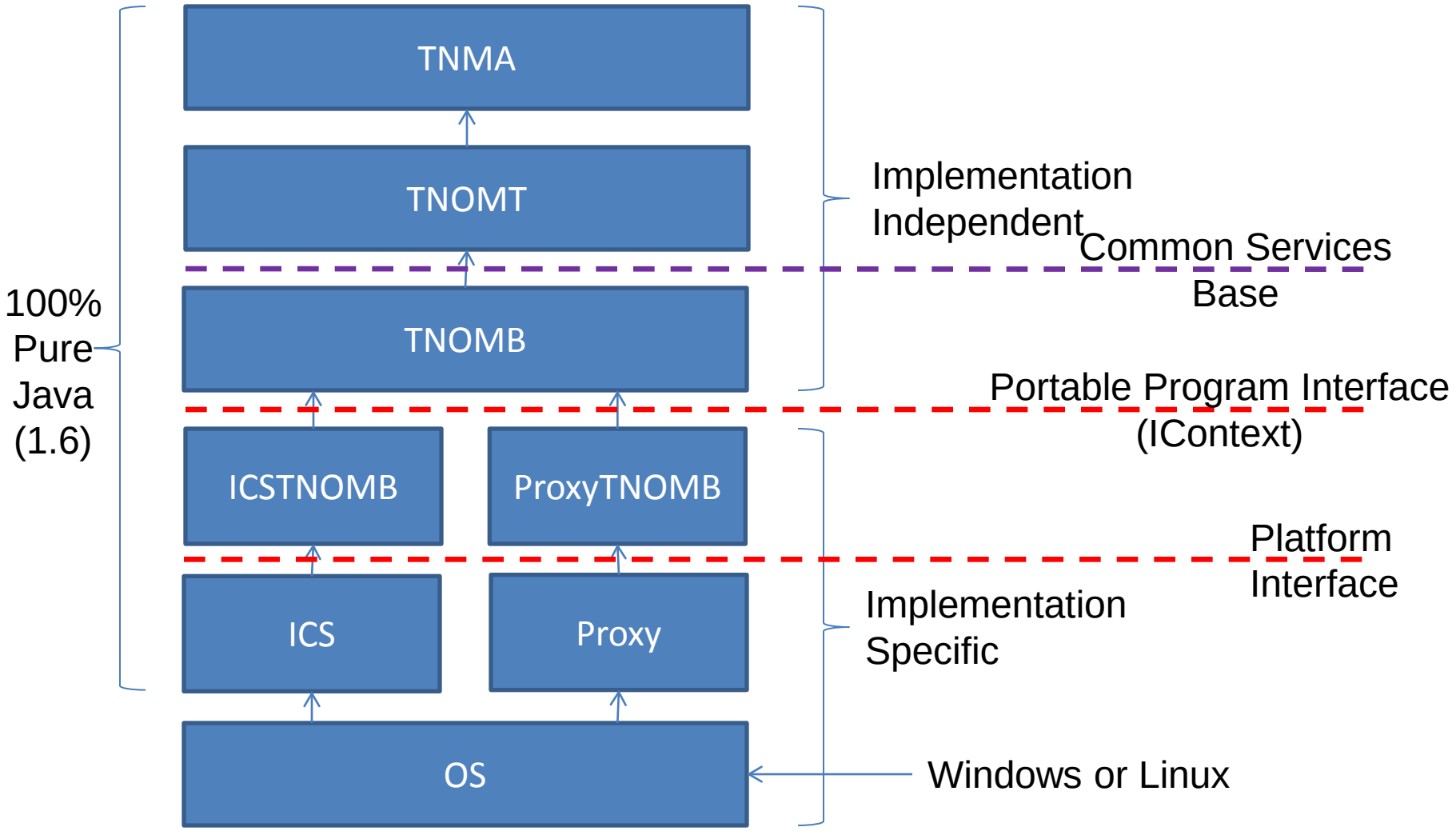


© 2009 IBM Corporation

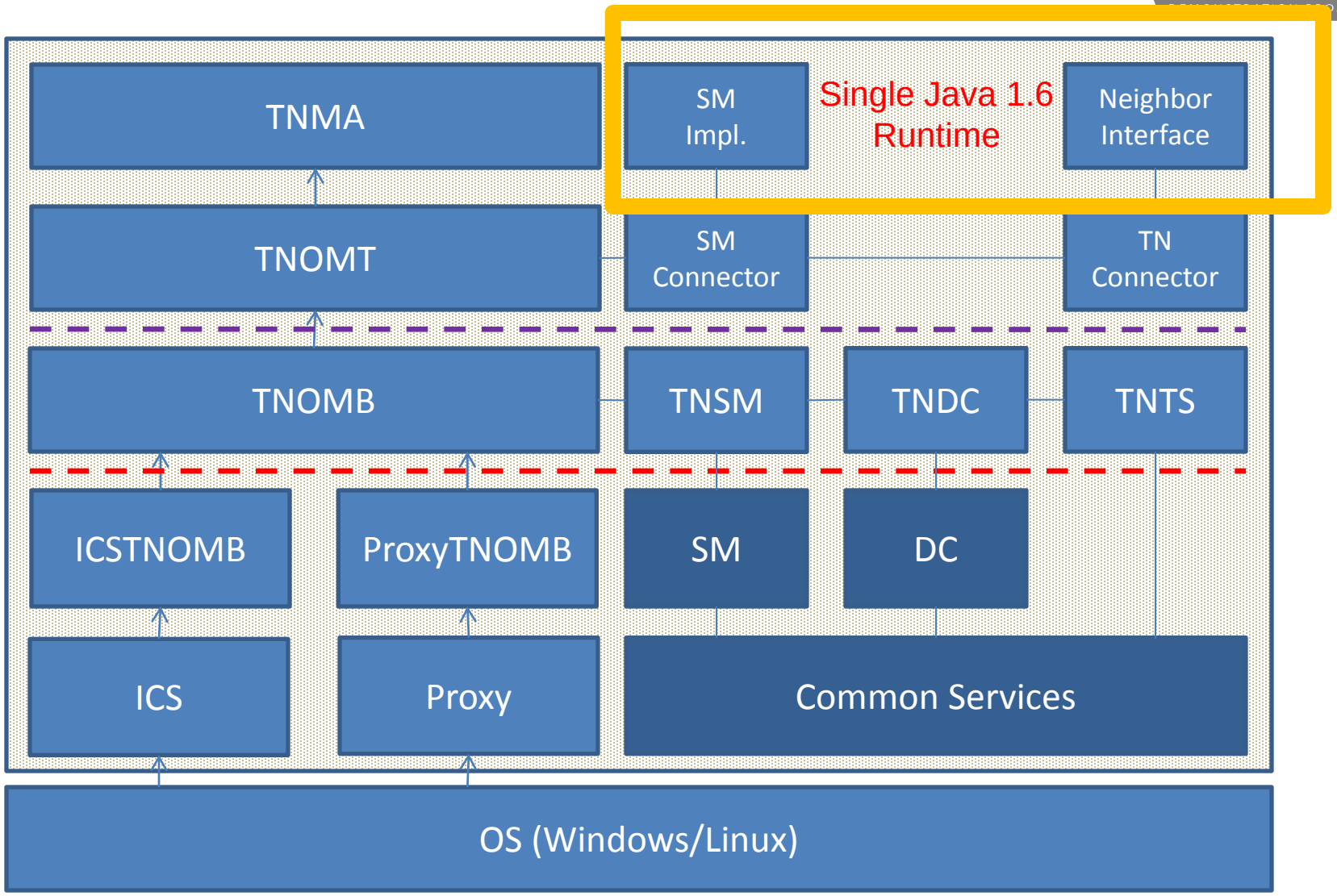
Transactive Node Object Model (TNOM)

Now we're at 1,000 feet

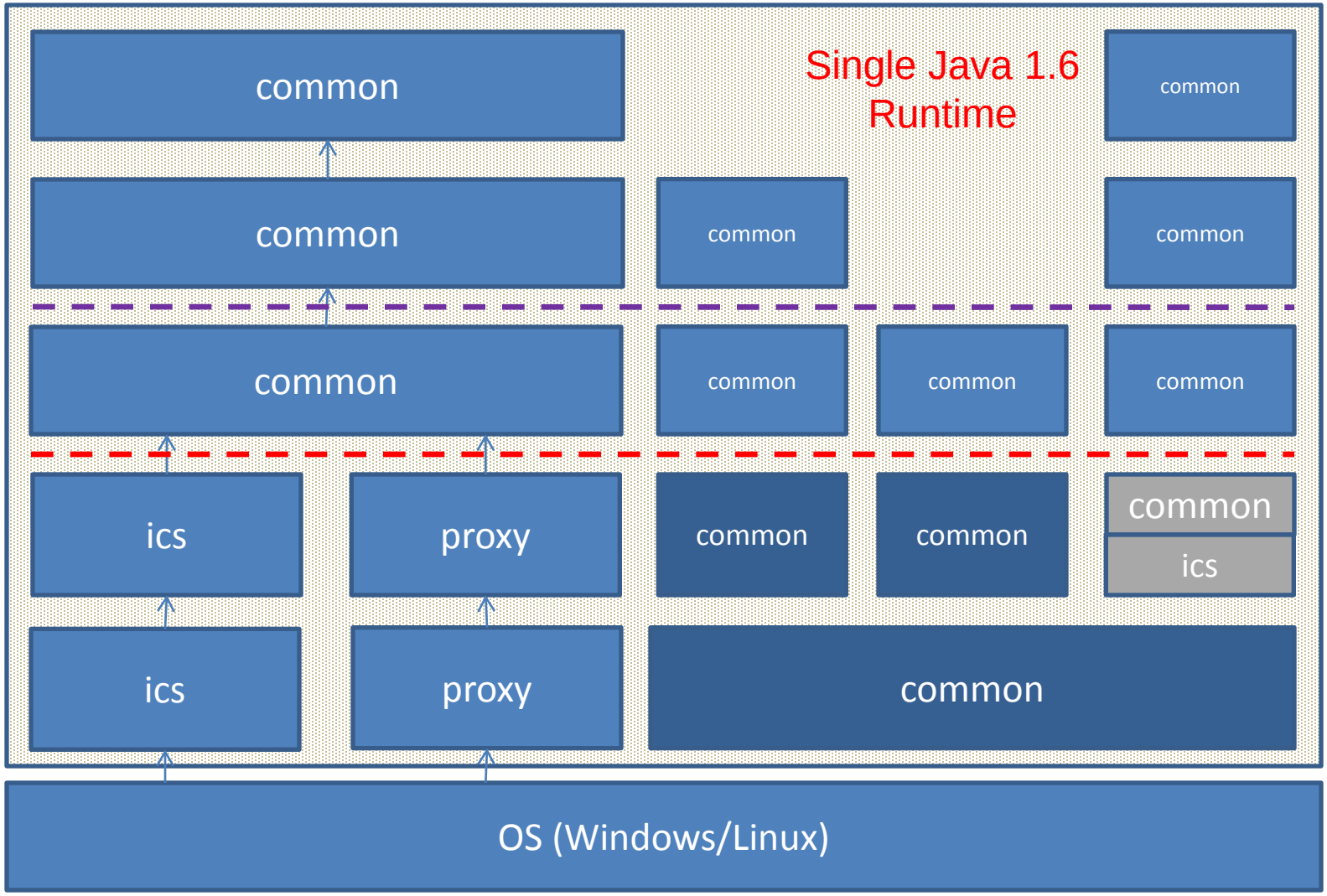
Stack Structure



Stack Structure



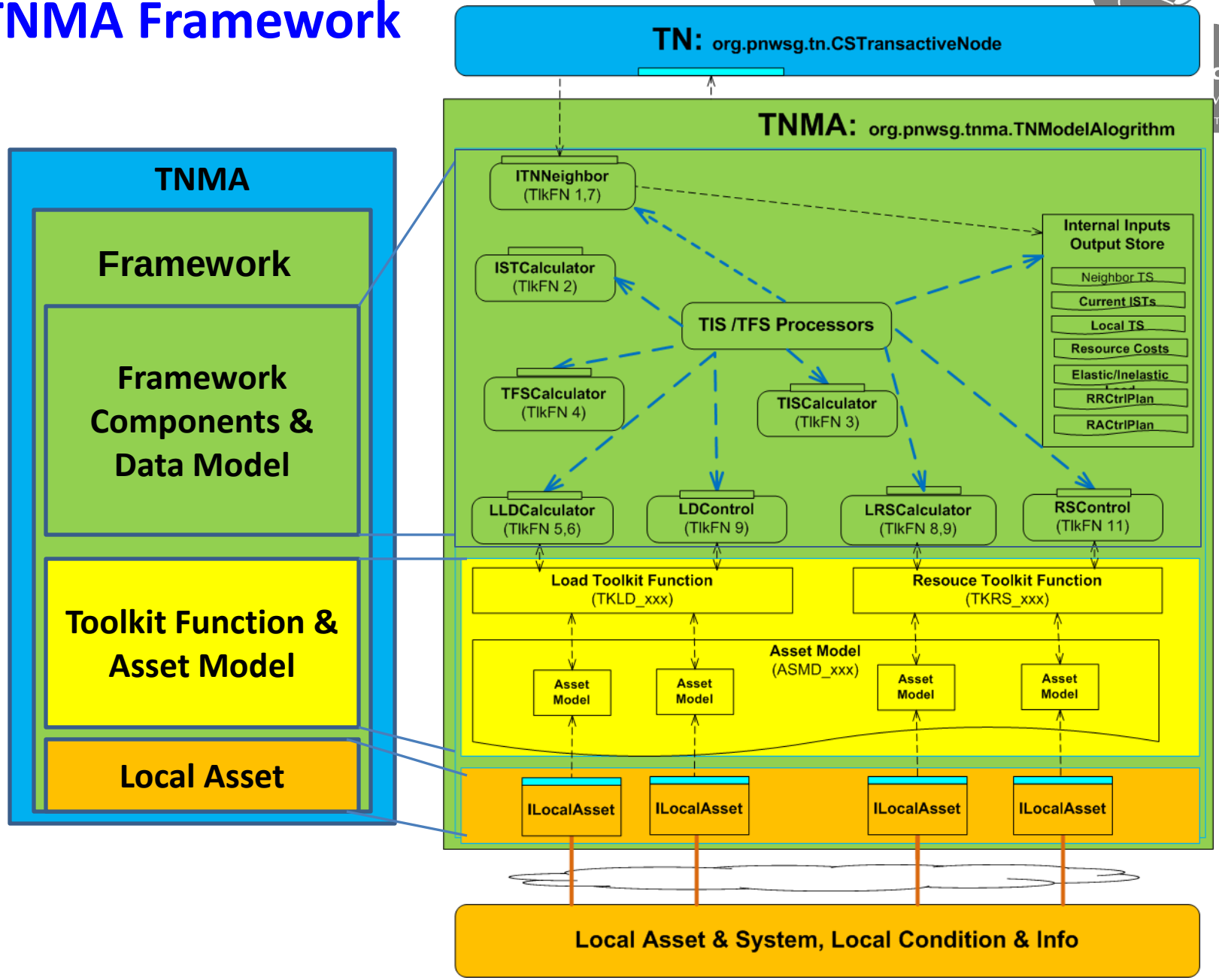
Stack Structure



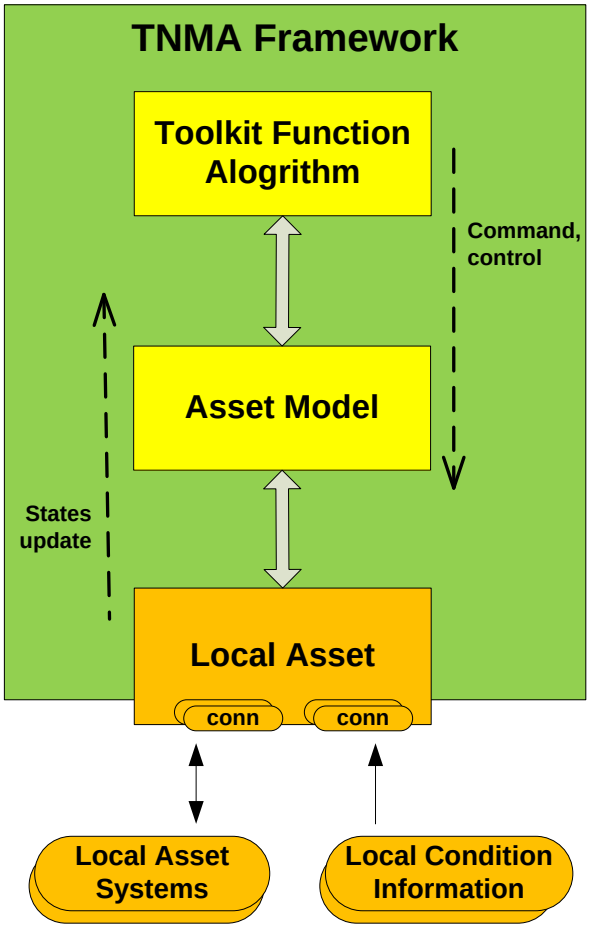
Transactive Node Model Algorithm (TNMA), Transactive Toolkit Functions and local responsive assets

Finally down in the weeds

TNMA Framework



Foundation of TIS/TFS Computing of Transactive Node:
Toolkit Function, Asset Model and Local Asset



Toolkit Function & Asset Model

Toolkit Function

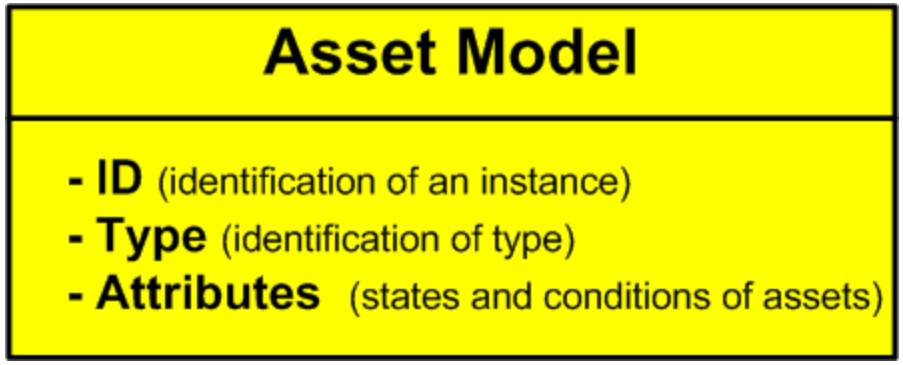
- A mathematic algorithm or control model that takes set of input data to compute output data needed for TIS/TFS calculation for a specific type of local asset or asset-system.
- The input data set of a Toolkit function includes:
  - **Current states and condition of local asset**
  - **Local information related to local asset**
  - Current states of Transactive Node
- The output data set of a Toolkit function is used by TNMA Framework for TIS/TFS calculation.
- There are two categories of Toolkit Function: Resource and Load
- Created and managed by Framework; uniquely identified by Toolkit Function **ID** (type and version)
- Algorithm Code is represented and captured JAVA object class in Framework

Toolkit Function

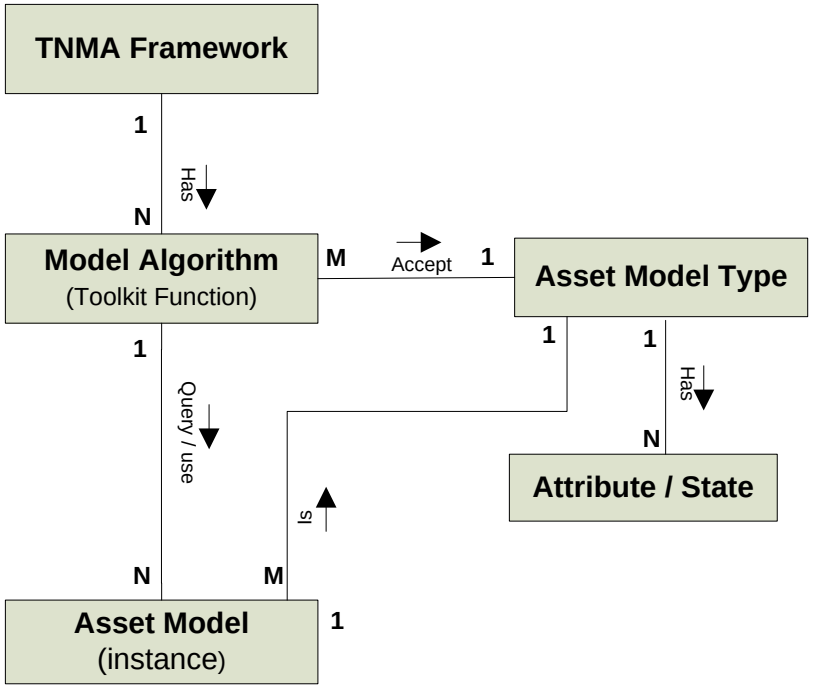
- **ID** (identification of type, version, creator)
- **Inputs** (Local asset states, conditions, etc.)
- **Outputs** (for local TIS/TFS calculation)
- **Algorithm** (code)

Asset Model

- **Asset Model**
  - A software object represents current states, local condition or static characteristics/configuration of a local asset of an asset-system
  - A data abstraction model viewed by a Toolkit Function of local asset or asset-system
  - It is the input data of a specific Toolkit Function object
  - Each Asset Model is typed or associated with a JAVA object class in Framework
  - Each instance of an Asset Model object is referenced and used by two objects in Framework:
    - A Toolkit Function object
    - A Local Asset object



Relation of Toolkit Function & Asset Model



- **Asset Model Relation to Toolkit Function**
  - Each Toolkit Function can accept, take or associate with only one **Type** of Asset Model
  - One **Type** of Asset Model can be accepted by, taken by, or associated with multiple Toolkit Functions
  - Each instance of a Toolkit Function object can work on multiple instances of an Asset Model objects of same **type**

Local Asset

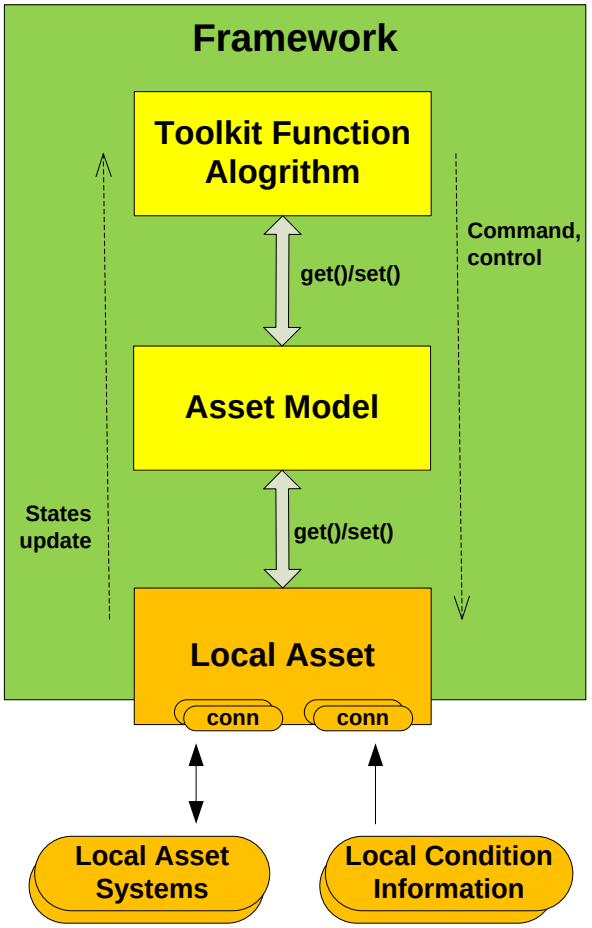
- **Functions and Responsibility**
  - Represents one asset or asset-system associated with a Transactive Function.
  - Based on the current state of local asset/asset-system, provides state update to Asset Model object used by a Toolkit Function Algorithm.
  - Accepts and understands outputs (control command or signals) from Toolkit Function to control or influence local asset/asset-system
  - Manages one or more connections to data sources required for updating the Asset Model object
  - Interprets / translates local data to formats understood by the Framework and vice versa.

Local Asset

- **ID** (identification of an instance)
- **Asset_Model_Type** (type of Asset Model)
- **Asset_Model** (reference to an Asset Model instance)
- **Connections** (to data source)
- **Code** (code, object class)

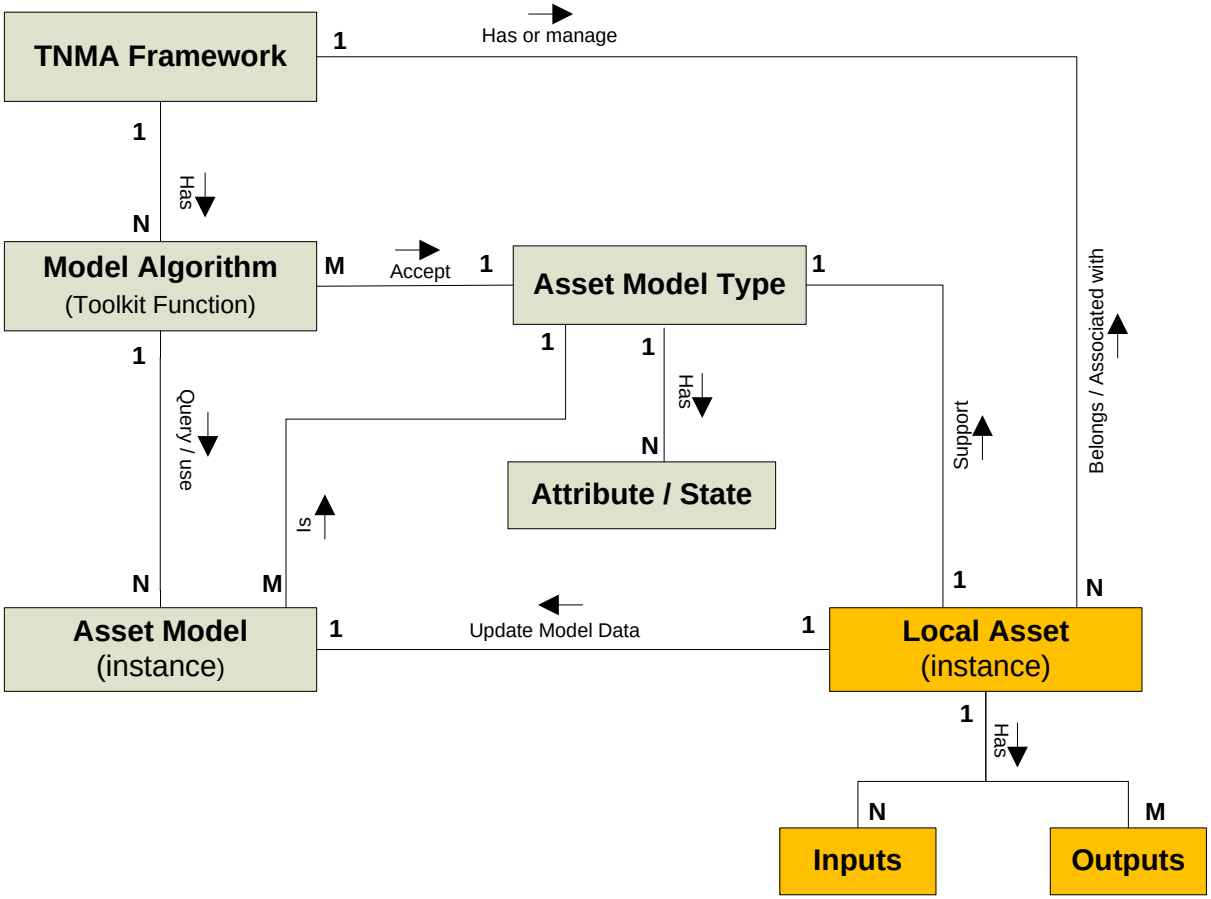
Local Asset

- **Relation to Framework:**
  - Local Asset is developed and owned by a Transactive Node owner/manager
  - Loaded and managed by the Framework
  - Local Asset and Toolkit Function forms data supply-and-consume relationship through the Asset Model object
  - Each Asset Model update consists of the following actions
    - Update the model object
    - Notify framework (through callback)
    - Accept Transactive Control command from Toolkit Function and send to local asset



Local Asset

- Relation to Asset Model and Toolkit Function



Contact

Ron Ambrosio
Global Research Executive
Energy & Utilities Industry

Ron Ambrosio/Watson/IBM@IBMUS
rfa@us.ibm.com

+1 914-945-3121

IBM T.J. Watson Research Center
P.O. Box 218
1101 Kitchawan Rd. / Route 134
Yorktown Heights, NY 10598

© 2010 IBM Corporation

© 2010 IBM Corporation