Data Exchange

Agenda

1

> Define your Data Exchange/Management scenario

2

> 3 exchange secnearios
> Project and model management
> Blueprint of your solution

3

> Model API starting guide

> Knowing the basic concepts
> Model objects and data traversing
> Optimization
> Resource and help

Data Exchange Scenarios

> File Exchange

> Read/Write data to various file formats
> Industry standards
> Proprietary formats

> Database & Repository Exchange

> Read/Write data to external Database systems

> Direct-Link Exchange

> Exchanging model data in real time with other applications
> Direct integration with other API, SDK

Scenario 1- File Exchange

USE CASE

> Formats not supported in Tekla Structures
> Contractual agreement or IPD process requirement

> Industry Standards

> IFC, ISO-15926, AecXML

> Industry Initiatives & Projects

> CSI Classification Systems:
OminClass, Master/Unit Formats
> FM and Commissioning - COBie2
> Other Industry consortium

> Proprietary formats

> Other software native formats
Fabtrol KISS, CNC Fabrication formats, ...
aSa rebar fabrication format
> General CSV file exchange
> General XML file exchange

Scenario 2- Database Exchange

USE CASE

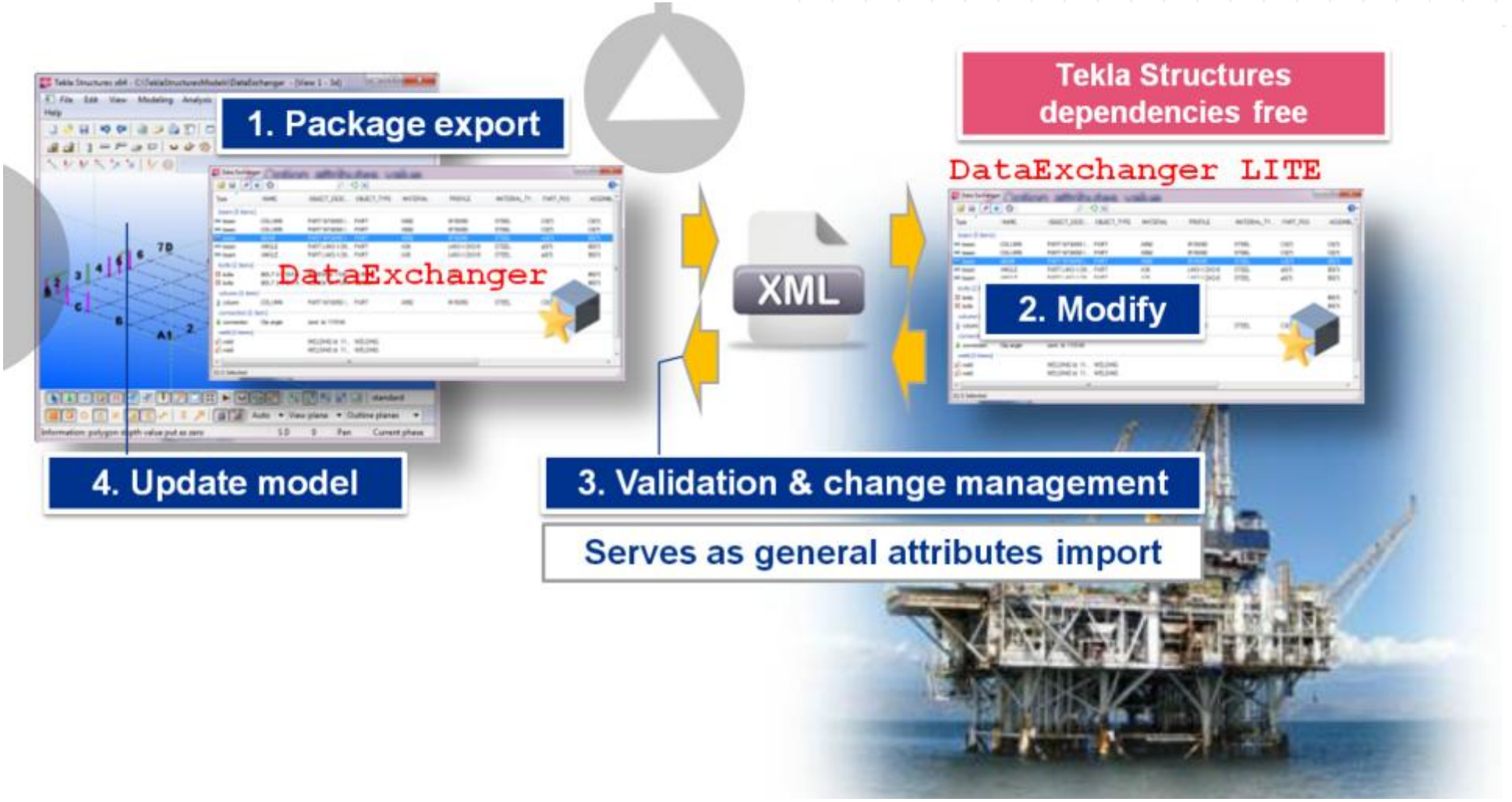
> Linking to your ERP or MIS Systems
> Create daily dashboard-like summary
> Integration with in-house DB solution
> Project Data on mobile or handhold devices

EXAMPLE

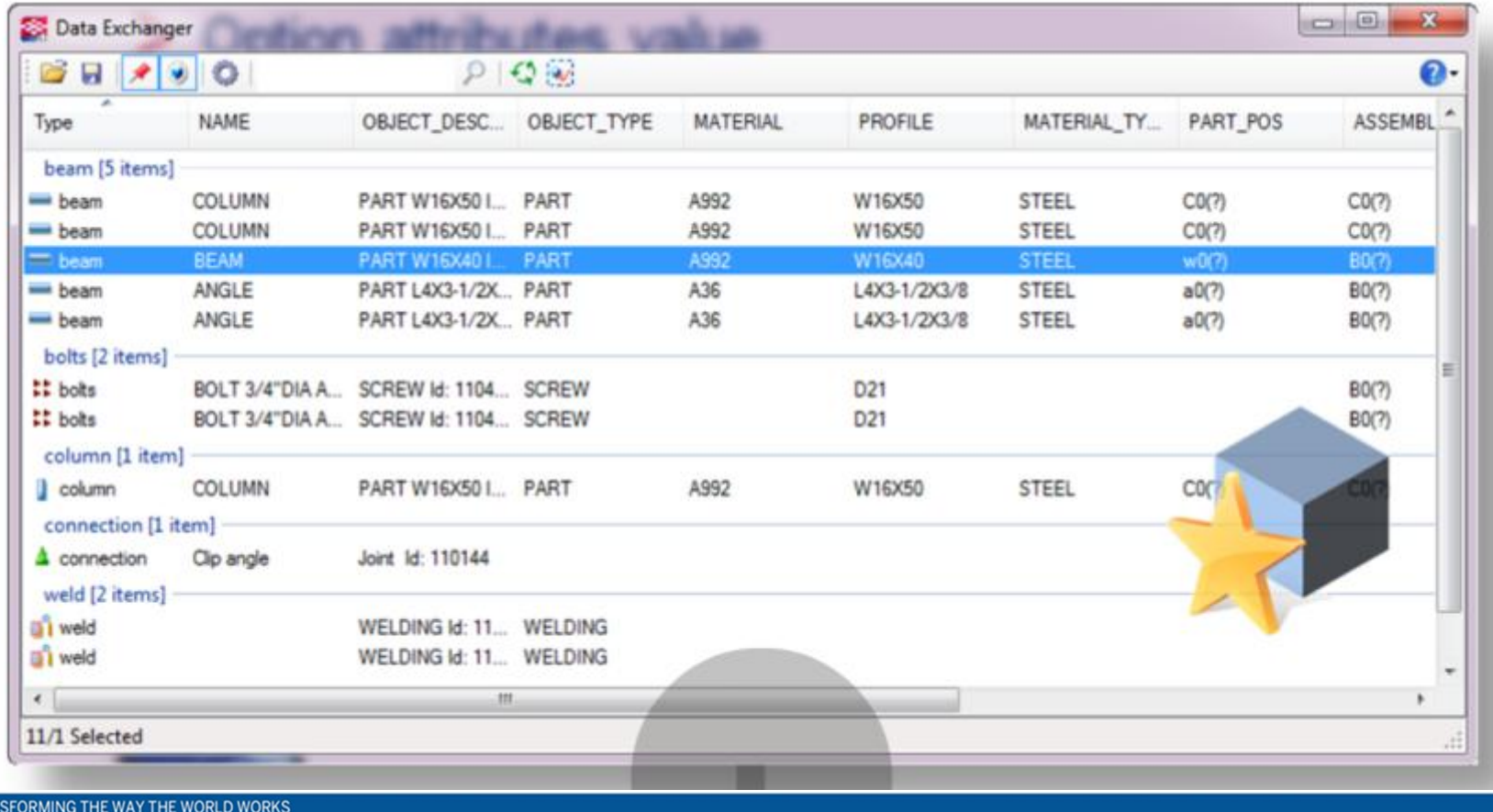
> Example extensions from US Solution Team
  - UDA List
  - Data Exchanger

more detail>

> UM2010 Presentation-"Steel Design Work Process"

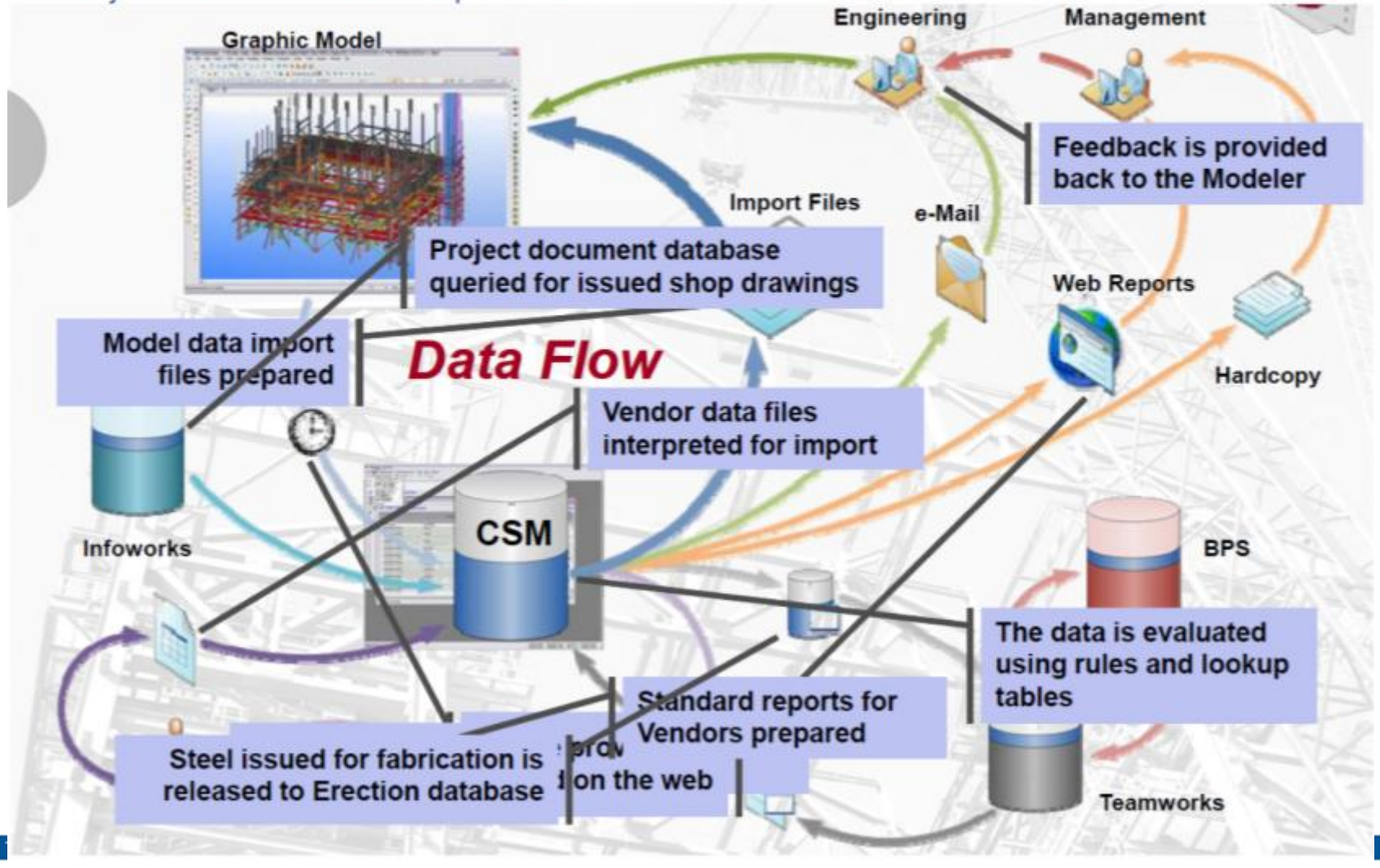

1. Package export
DataExchanger
XML
Tekla Structures
dependencies free
DataExchanger LITE
2. Modify
4. Update model
3. Validation & change management
Serves as general attributes import

Data Exchanger



Multiple Database and In-house ERP System Integration

> Tekla User In-house Database Integration
Curtesy of Bechtel Tekla UM10 presentation



Scenario 3- Direct-Link Data Exchange

USE CASES

> Integrating Tekla Structures with other software
> Runtime API-to-API data exchange

> Example extensions from US Solution Team
- Layout Manager

Project Management & Workflow

> IPD process requirement — Model based communication
> Model content management
> Status visualization — construction/fabrication status

USE CASES

> Workflow Management & Visualization

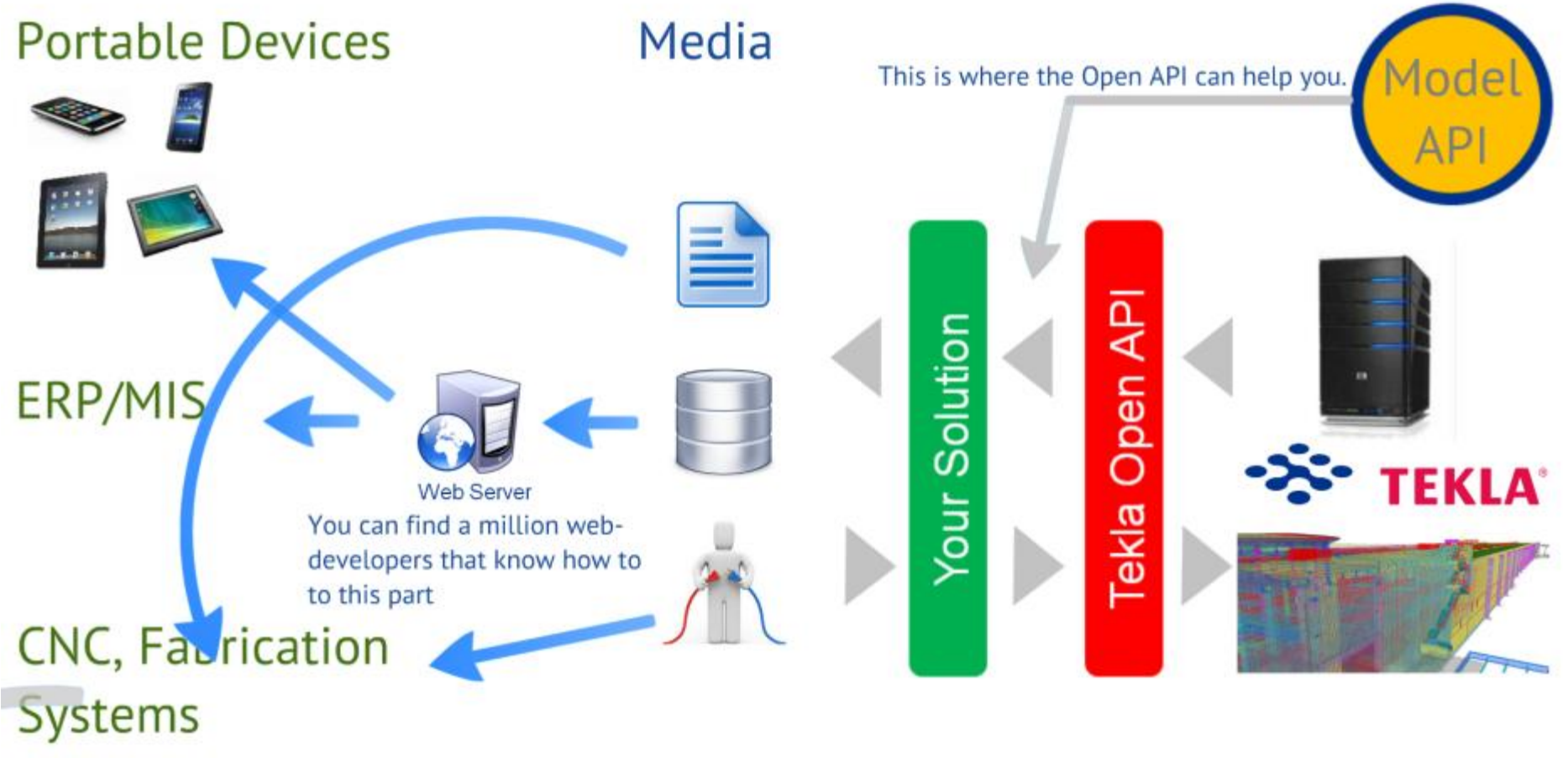
> In-Model Reviewer

> Project & Model Management

> Component Manager
> Component Variables

MPLE

Blueprint of your Data Exchange and Project Management Solution



Linking to Your ERP or MIS System

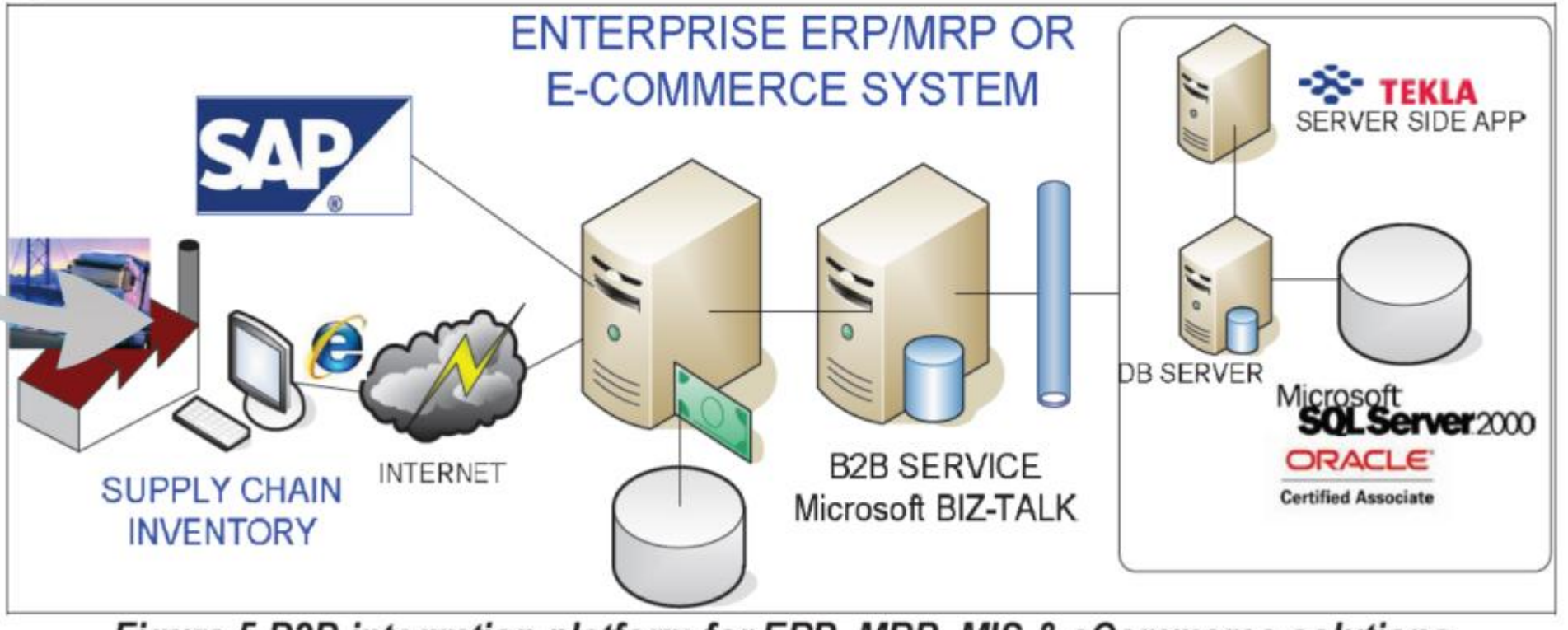
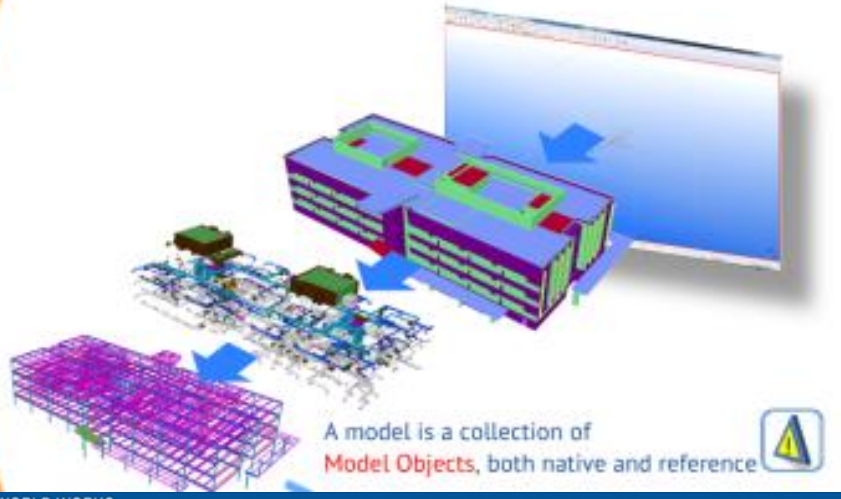
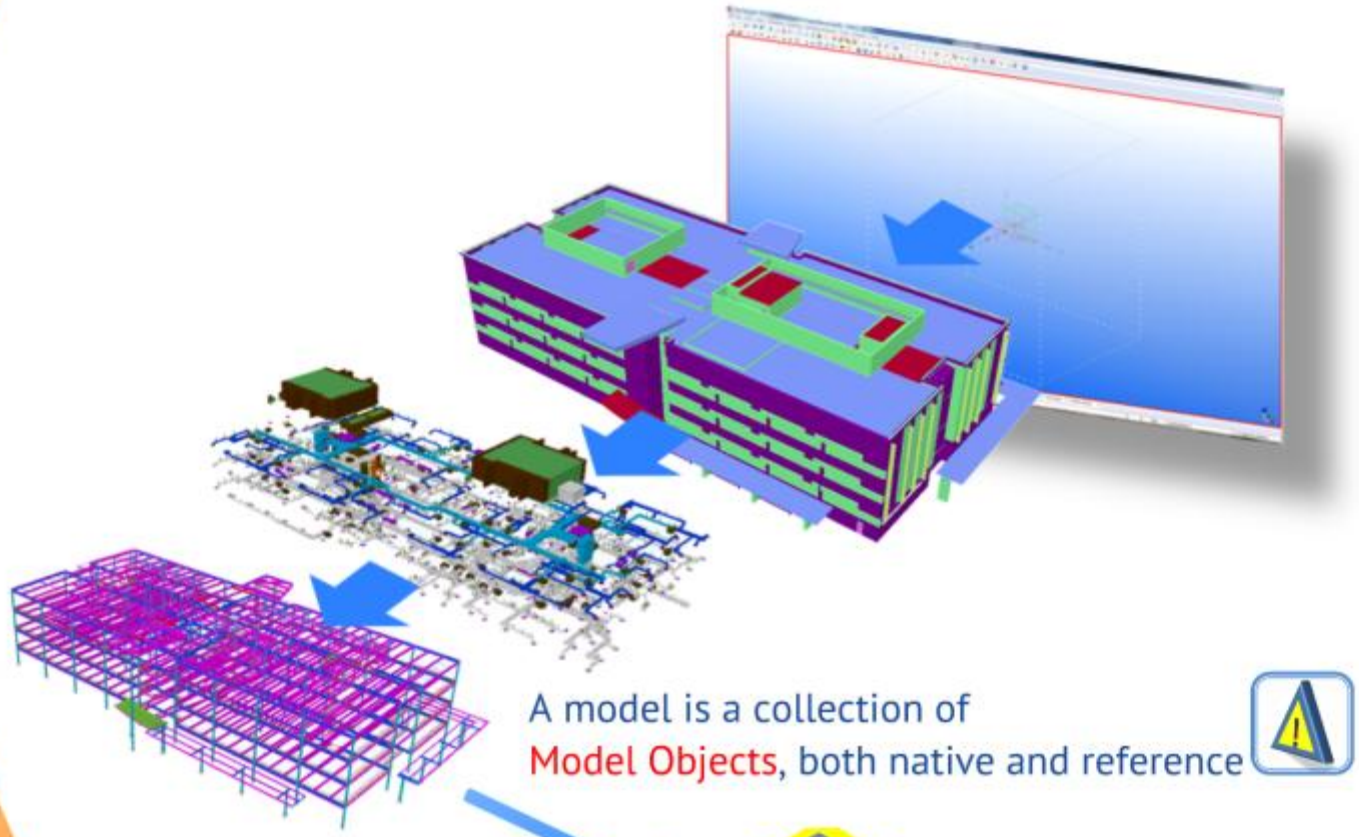


Figure 5 B2B integration platform for ERP, MRP, MIS & eCommerce solutions

Knowing the Concepts

> Simple data structure in Model API



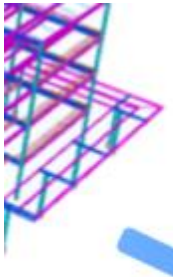


A model is a collection of
Model Objects, both native and reference

Model Object

1 Object Types
Sub-types of model objects in Model Open API

2 Object Properties
- User Property (UDA) e.g. User Field 1
- Report Property
- [API] Class Properties e.g. Part Solids



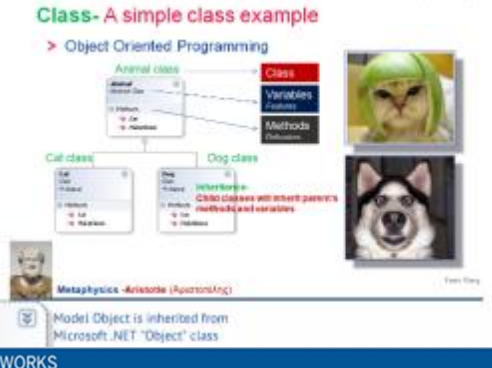
A model is a collection of Model Objects, both native and reference

Model Object

1 Object Types

Sub-types of model objects in Model Open API

Polymorphism & Inheritance



2 Object Properties

- User Property (UDA)

- Report Property

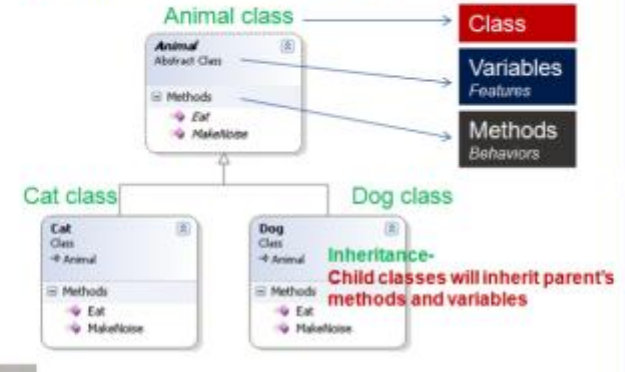
1 Object Types

Sub-types of model objects in Model Open API

Polymorphism & Inheritance

Class- A simple class example

> Object Oriented Programming

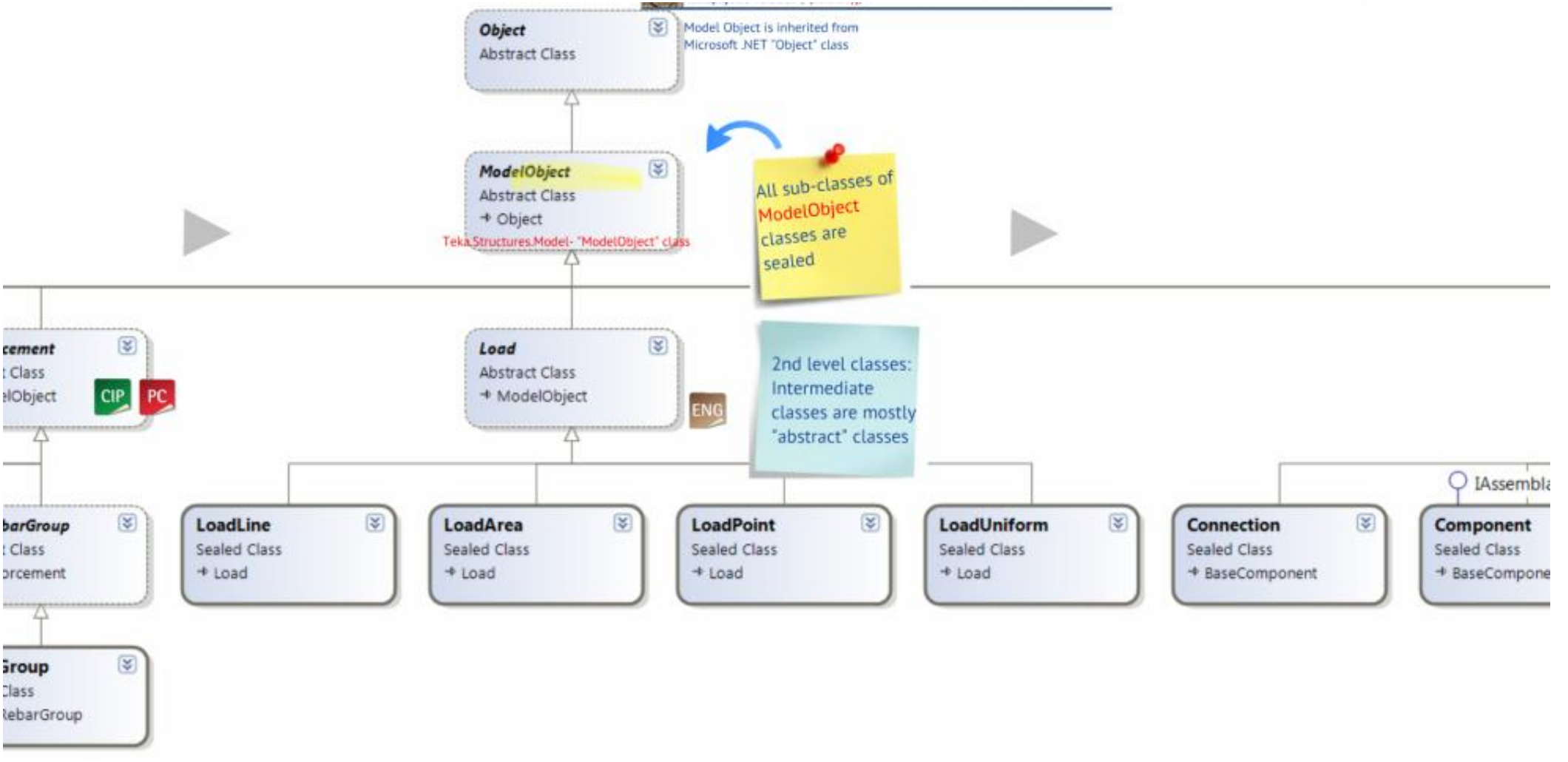


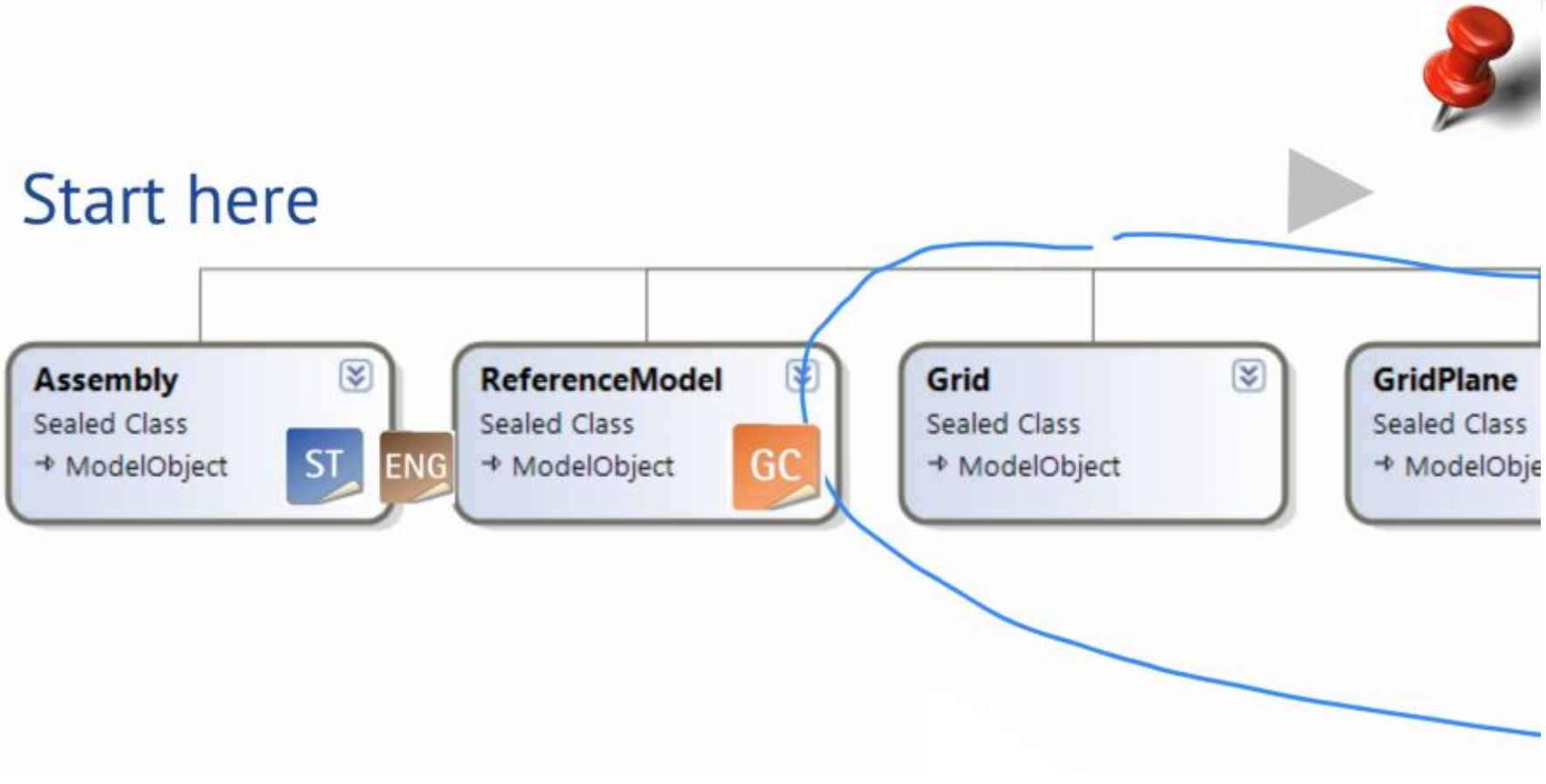
Frank Wang

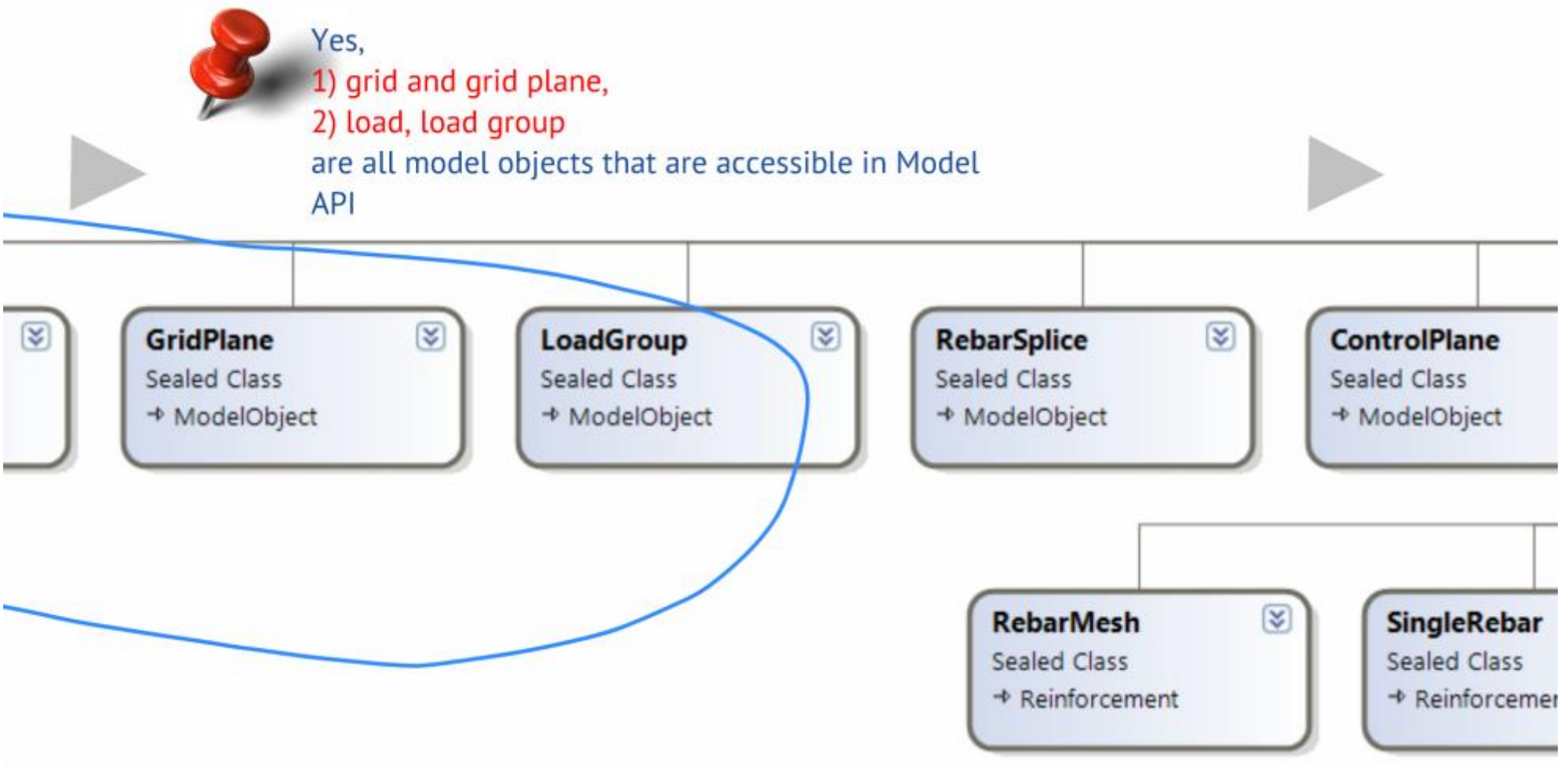
Metaphysics -Aristotle (Ἀριστοτέλης)

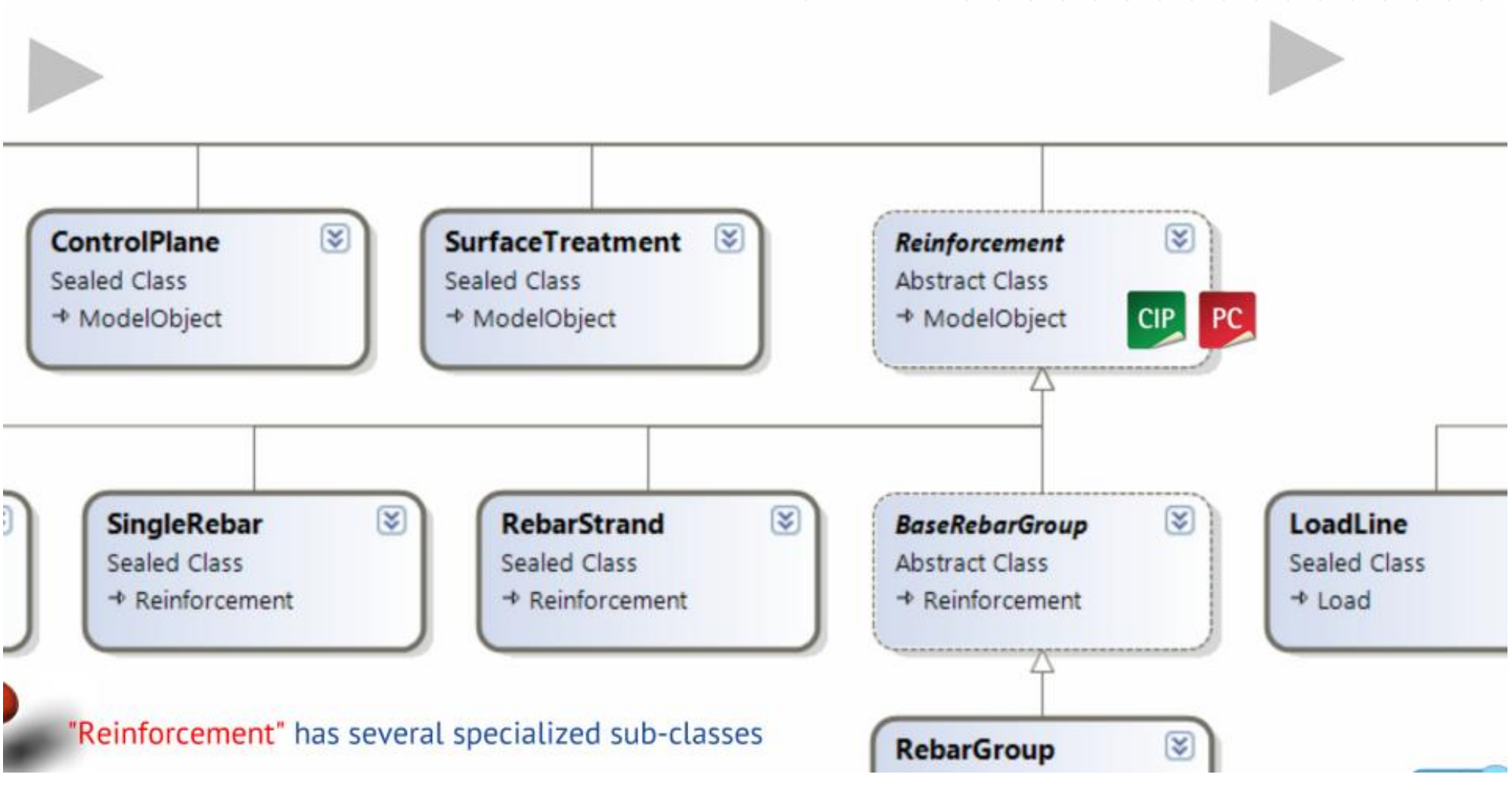
Object
Abstract Class

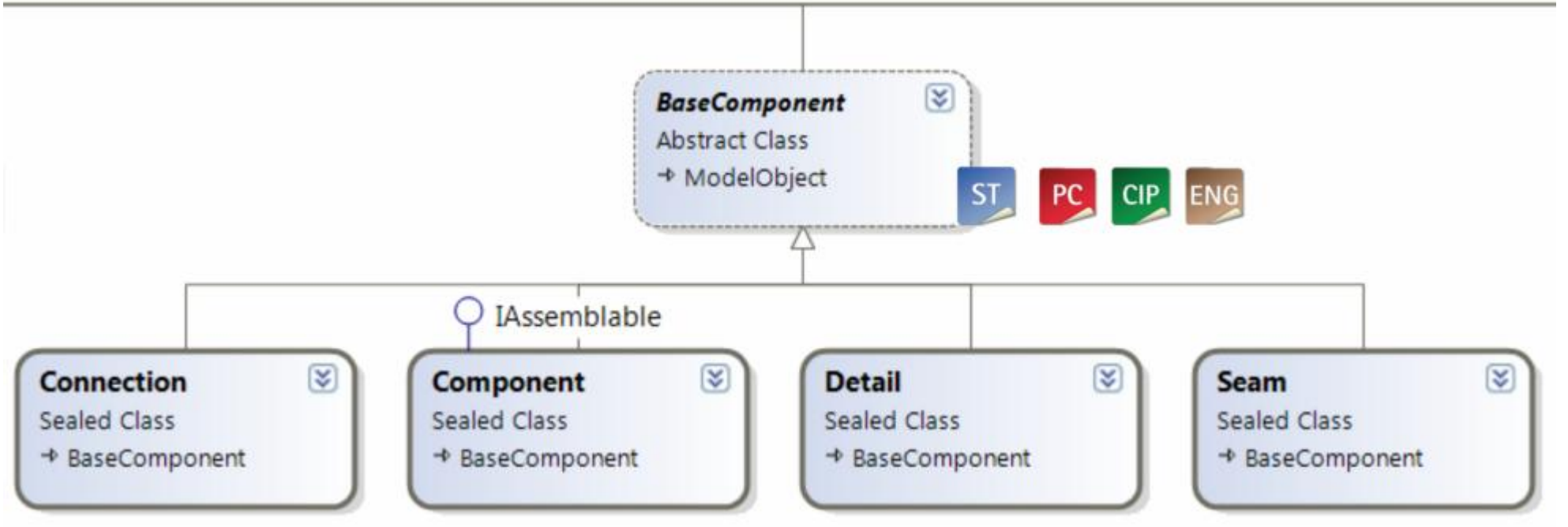
Model Object is inherited from Microsoft .NET "Object" class

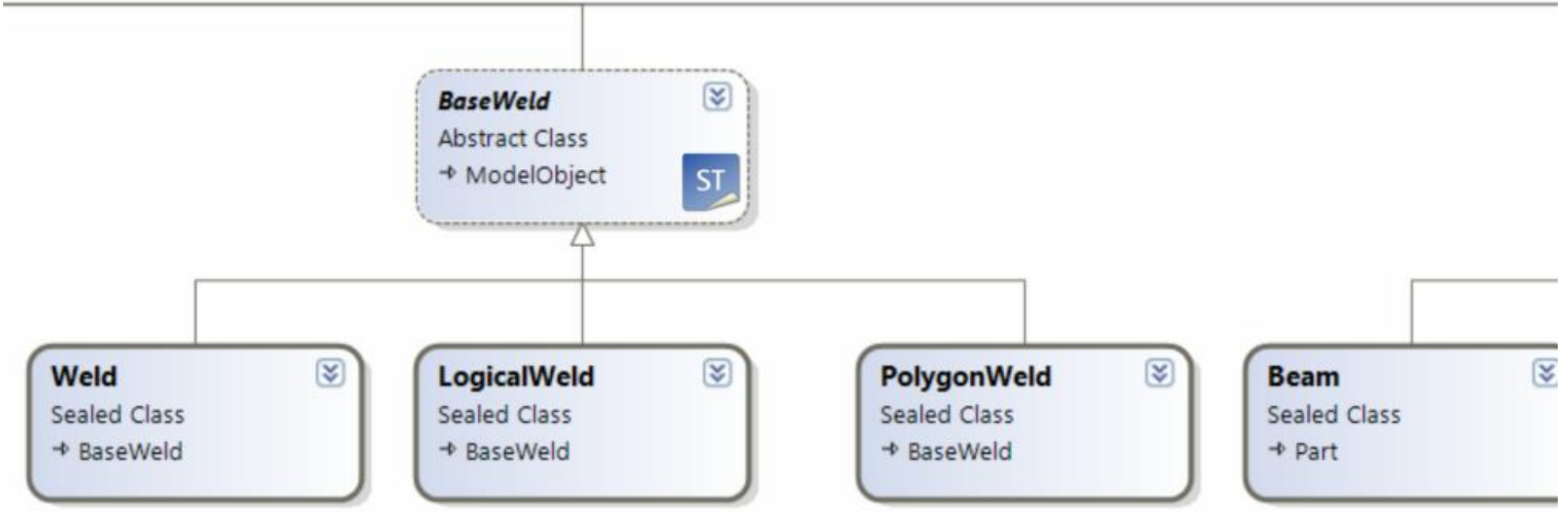

Object
Abstract Class
Model Object is inherited from Microsoft .NET "Object" class
ModelObject
Abstract Class
Object
Tekla.Structures.Model- "ModelObject" class
All sub-classes of ModelObject classes are sealed
Load
Abstract Class
ModelObject
ENG
2nd level classes: Intermediate classes are mostly "abstract" classes
CIP
PC
LoadLine
Sealed Class
Load
LoadArea
Sealed Class
Load
LoadPoint
Sealed Class
Load
LoadUniform
Sealed Class
Load
Connection
Sealed Class
BaseComponent
Component
Sealed Class
IAssembla

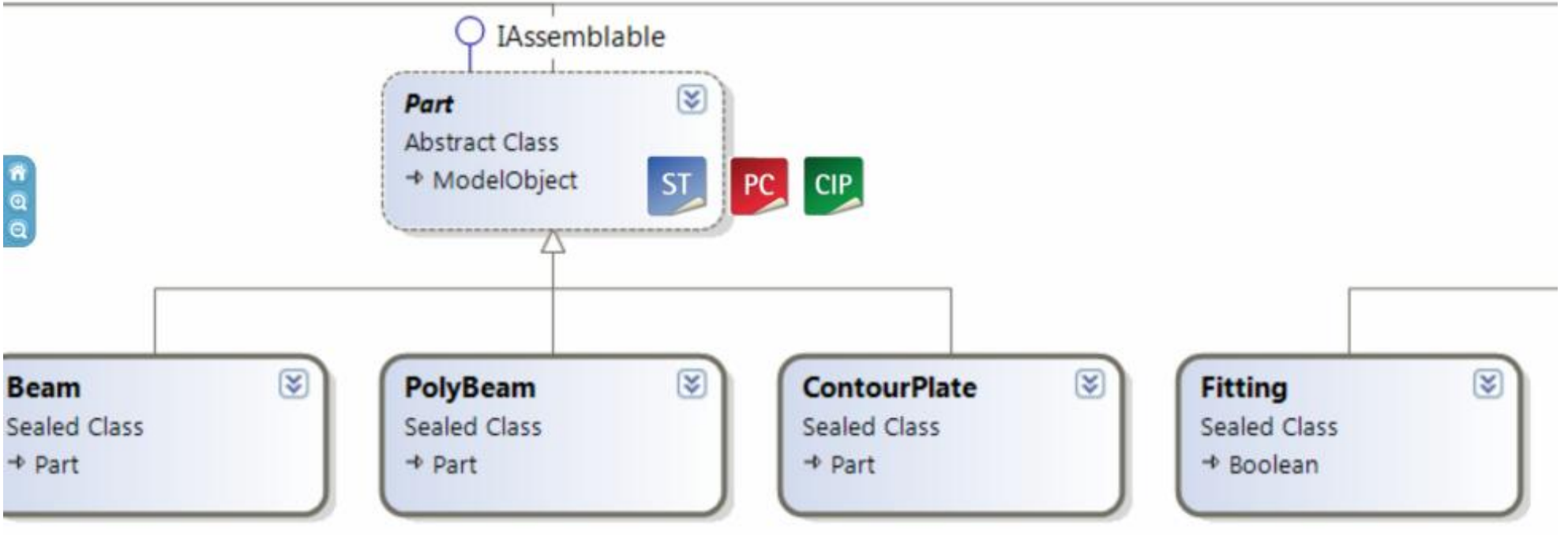

Start here
Assembly
Sealed Class
ModelObject
ST
ENG
ReferenceModel
Sealed Class
ModelObject
GC
Grid
Sealed Class
ModelObject
GridPlane
Sealed Class
ModelObje


Yes,
1) grid and grid plane,
2) load, load group
are all model objects that are accessible in Model API
GridPlane
Sealed Class
ModelObject
LoadGroup
Sealed Class
ModelObject
RebarSplice
Sealed Class
ModelObject
ControlPlane
Sealed Class
ModelObject
RebarMesh
Sealed Class
Reinforcement
SingleRebar
Sealed Class
Reinforcemer


ControlPlane
Sealed Class
ModelObject
SurfaceTreatment
Sealed Class
ModelObject
Reinforcement
Abstract Class
ModelObject
CIP
PC
SingleRebar
Sealed Class
Reinforcement
RebarStrand
Sealed Class
Reinforcement
BaseRebarGroup
Abstract Class
Reinforcement
LoadLine
Sealed Class
Load
RebarGroup
"Reinforcement" has several specialized sub-classes


BaseComponent
Abstract Class
ModelObject
ST
PC
CIP
ENG
IAssemblable
Connection
Sealed Class
BaseComponent
Component
Sealed Class
BaseComponent
Detail
Sealed Class
BaseComponent
Seam
Sealed Class
BaseComponent

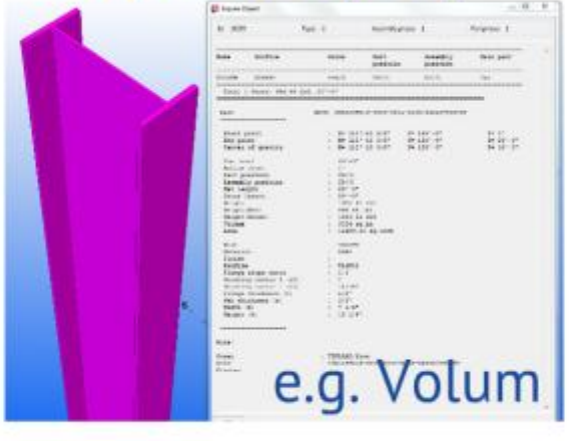

BaseWeld
Abstract Class
ModelObject
ST
Weld
Sealed Class
BaseWeld
LogicalWeld
Sealed Class
BaseWeld
PolygonWeld
Sealed Class
BaseWeld
Beam
Sealed Class
Part


IAssemblable
Part
Abstract Class
ModelObject
ST
PC
CIP
Beam
Sealed Class
Part
PolyBeam
Sealed Class
Part
ContourPlate
Sealed Class
Part
Fitting
Sealed Class
Boolean

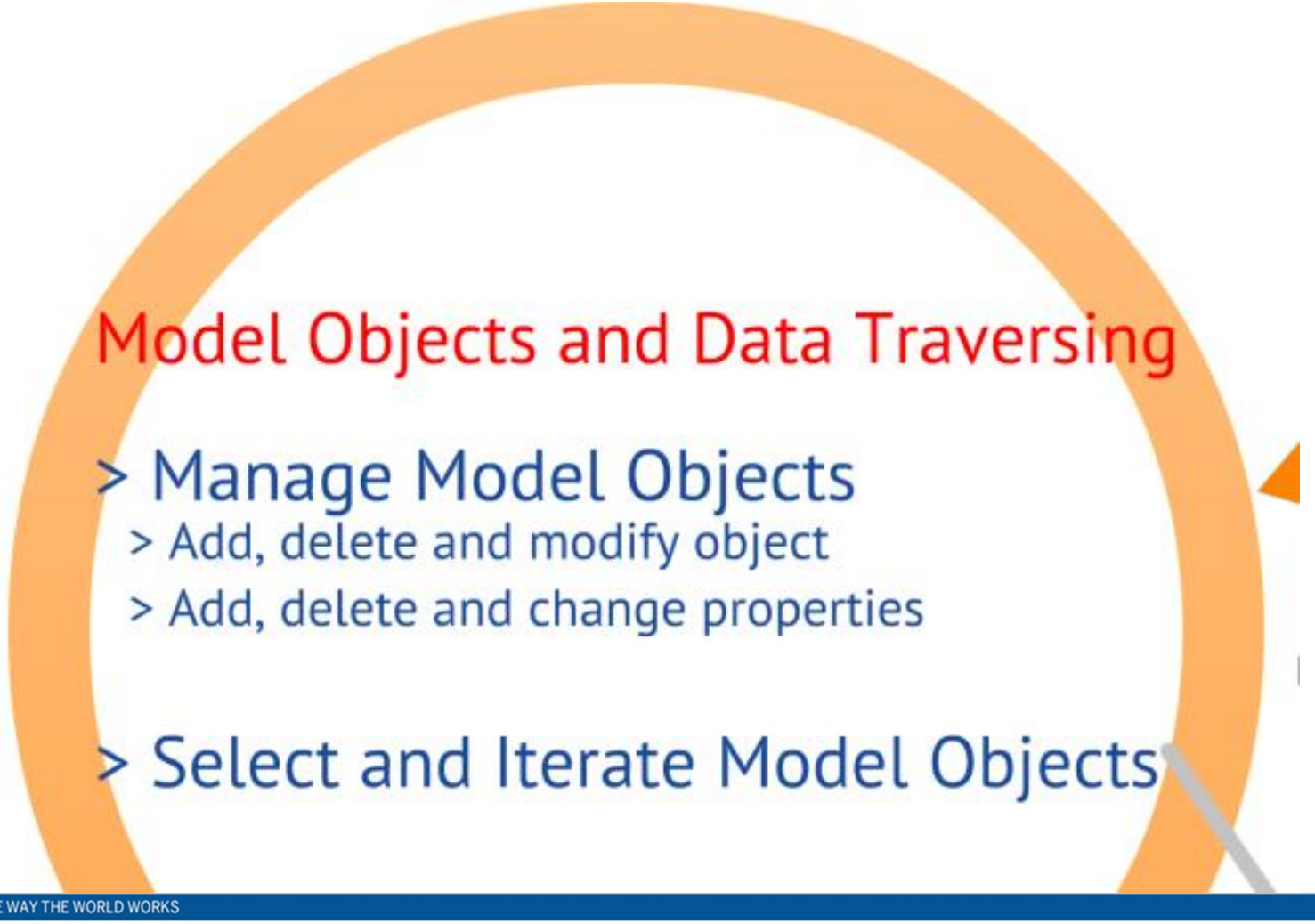
2 Object Properties

- User Property (UDA)

- Report Property



- [API] Class Properties


Model Objects and Data Traversing
> Manage Model Objects
> Add, delete and modify object
> Add, delete and change properties
> Select and Iterate Model Objects

Select and Iterate Model Objects

Model API provides consistent methods to:

- Select and gather objects
  - Using existing filter(s) or
  - Select by object type(s)
- Easy iteration method

Simple Instruction

- Create a selector
- Define the selector's behavior
- Iterate through returned objects

Filters

standard
Cip_All
Cip_Beam
Cip_Caisson
Cip_Column
Cip_Hardware
Cip_Joist
Cip_PadFooting
Cip_Pilecap
Cip_Retaining_Wall
Cip_Shear_Wall
Cip_Slab
Cip_StripFooting
Cip_Wall
Concrete_All
External_Assembly Code
External_Assembly Description
External_Fire Rating
External_Level
External_Load bearing
External_Material
External_Name
External_Object_type
External_Profile name
External_Reference

All

Code Example

C#

Copy

```
using Tekla.Structures.Model;
using System;
using System.Windows.Forms;

public class Example
{
    public void Example1()
    {
        Model Model = new Model();

        ModelObjectEnumerator ObjectEnum = Model.GetModelObjectSelector().GetAllObjects();
        ObjectEnum.SelectInstances = false; // Set the "SelectInstances" to false to speed up the enquiry; possible because only report properties are asked.

        string Result = "CHECKED BY, CHECKED DATE, OBJECT LOCKED \n";
        while(ObjectEnum.MoveNext())
        {
            if(ObjectEnum.Current != null)
            {
                Beam BeamObject = ObjectEnum.Current as Beam;
                if(BeamObject != null)
                {
                    string CheckedBy = "";
                    double DateCheckedValue = 0.0;
                    int LockedStatus = -1;

                    DateTime DateChecked = new System.DateTime(1970, 1, 1);

                    BeamObject.GetUserProperty("CHECKED_BY", ref CheckedBy);
                    BeamObject.GetUserProperty("CHECKED_DATE", ref DateCheckedValue);
                    BeamObject.GetUserProperty("OBJECT_LOCKED", ref LockedStatus);
                    if(CheckedBy.Length > 0 || DateCheckedValue > 0.0 ||
                        LockedStatus != -1)

                    DateChecked = DateChecked.AddSeconds(DateCheckedValue);
                    Result += CheckedBy;
                    Result += ", ";
                    Result += DateChecked.ToString("dd.MM.yyyy");
                    if(LockedStatus == 1)
                    {
                        Result += ", Locked\n";
                    }
                    else
                    {
                        Result += ", Not locked\n";
                    }
                }
            }
        }
        MessageBox.Show(Result);
    }
}
```

Code Example

I am processing large amount of data, how can I improve the speed of my app?

How to deal with large models?

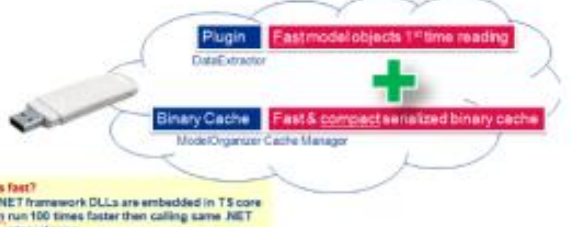
Optimization!

Tips for Beginners

> Gether objects by types and filters
> Avoid explicilty "Select" object
- Use report/user property if aviable
- Set the "SelectInstance" to "false" as defulat

Tips for Advance Users

> Consider doing the heavy work in plugin
> Utilize caching/serialization
- Make your cache accessible by your apps



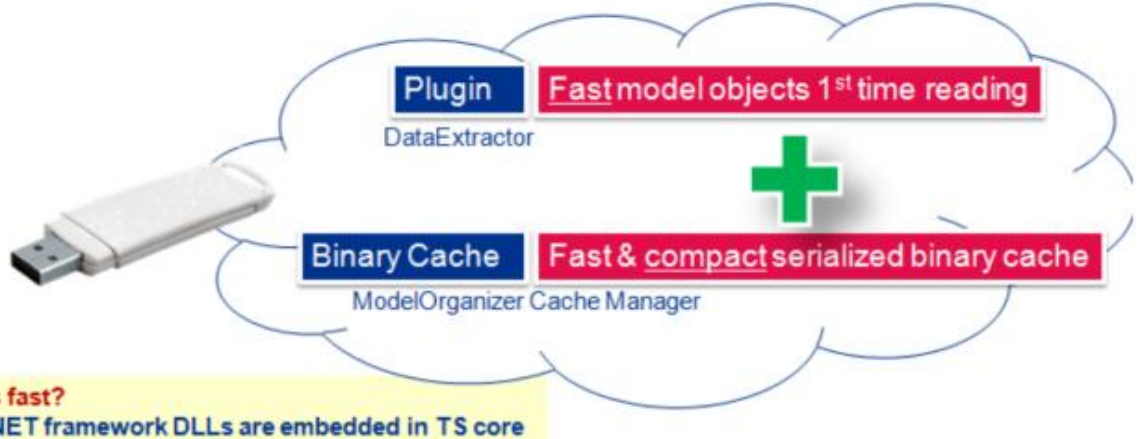
Why plugin is fast?
Several MS .NET framework DLLs are embedded in TS core makes Plugin run 100 times faster then calling same .NET DLL through external apps

Tips for Advance Users

> Consider doing the heavy work in plugin

> Utilize caching/serialization

- Make your cache accessible by your apps



Why plugin is fast?
Several MS .NET framework DLLs are embedded in TS core makes Plugin run 100 times faster then calling same .NET DLL through external apps

Thank You