



BUILDING AN IOT SOLUTION

Ben Stob

Software Development Manager, ThingWorx®

June 8, 2016

liveworx.com | #LIVEWORX



AGENDA

- Scenario: Acme Garden Tractor Co.
- The Model – Best Practices
- Connectivity - Edge Device Communication with ThingWorx®
- Apps – Rapid Application Development
- Creating an Extension using the Eclipse Plugin
- Best Practices for Team Development

AGENDA

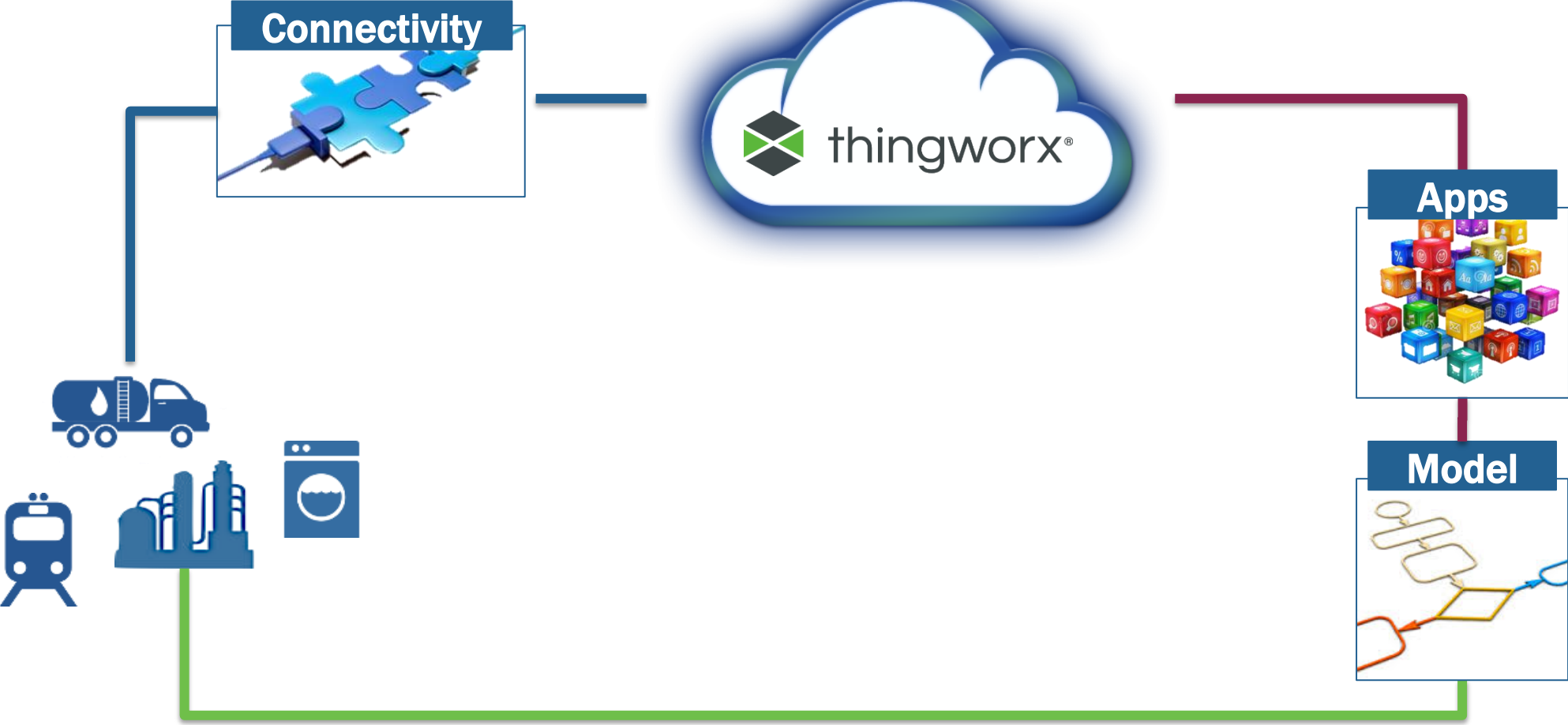
- Scenario: Acme Garden Tractor Co.
- The Model – Best Practices
- Connectivity - Edge Device Communication with ThingWorx[®]
- Apps – Rapid Application Development
- Creating an Extension using the Eclipse Plugin
- Best Practices for Team Development

BUILDING AN IOT SOLUTION - SCENARIO



- Acme Garden Tractor Co. – Provider of residential and commercial lawn tractors
 - Problem Statement
 - Would like a Connected Service offering allowing customers to receive **personalized feedback** about their tractor and upload tractor data to their dealer to **improve customer interaction and satisfaction**.
 - Personas
 - **Customers** – access to performance data, maintenance information, recall and service notifications, and feedback
 - **Dealers** – make it easier for customers to find and engage them for service, aggregate reliability information across their customers
 - Use Scenarios
 - **Customer** takes delivery of their tractor, installs mobile app, and registers the tractor, which instantiates a representation of that tractor in the IoT solution
 - **Customer** uses tractor for session, tractor establishes secure connection to mobile app and transmits usage data
 - **Dealer** views usage data across registered customers to identify marketing opportunities and to report on reliability metrics to the manufacturer

BUILDING AN IOT SOLUTION - SCENARIO

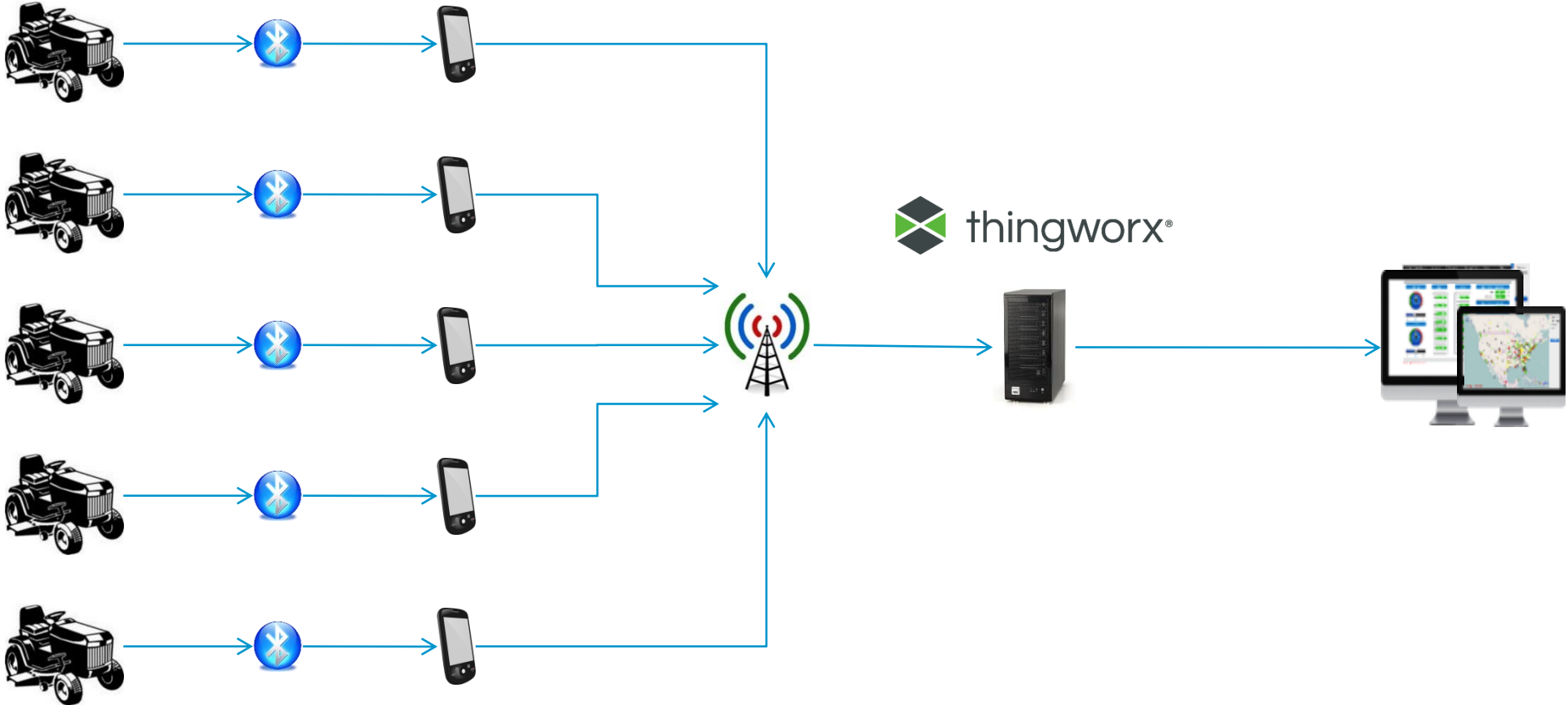


BUILDING AN IOT SOLUTION - SCENARIO



- Acme Garden Tractor Co.
 - Use Bluetooth® to connect to a mobile app and use ThingWorx® SDKs to communicate to ThingWorx® server via HTTPS over carrier network
 - Tractor data:
 - engine hours, engine temperature, peak engine temperature, engine RPM sampling, etc.
 - App data:
 - DIY services performed, e.g. oil change or blades replaced, location

BUILDING AN IOT SOLUTION - SCENARIO



AGENDA

- Scenario: Acme Garden Tractor Co.
- The Model – Best Practices**
- Connectivity - Edge Device Communication with ThingWorx®
- Apps – Rapid Application Development
- Creating an Extension using the Eclipse Plugin
- Best Practices for Team Development

BUILDING AN IOT SOLUTION – THE MODEL



- Some basic questions to ask when modeling:
 - How do we represent the real-life device/smart product in ThingWorx®?
 - How can the model remain flexible and maintainable?
 - What is the data that we want to capture? How often?
 - What do we want to do with the data?
 - E.g., trigger alerts, subscribe to events to initiate a downstream process, perform data analysis, etc.
 - What capabilities does ThingWorx® provide?
 - What capabilities do we need to add to ThingWorx® (e.g., via an extension)?

BUILDING AN IOT SOLUTION – THE MODEL



- ThingWorx® Model Best Practices
 - Thursday, June 9 – 9:30am – 10:15am
 - Bob Elam, Product Owner, ThingWorx®
- ThingWorx® Solution Architecture
 - Thursday, June 9 – 10:30am – 11:15am
 - John Schaefer, Senior Vice President, ThingWorx®

BUILDING AN IOT SOLUTION – THE MODEL



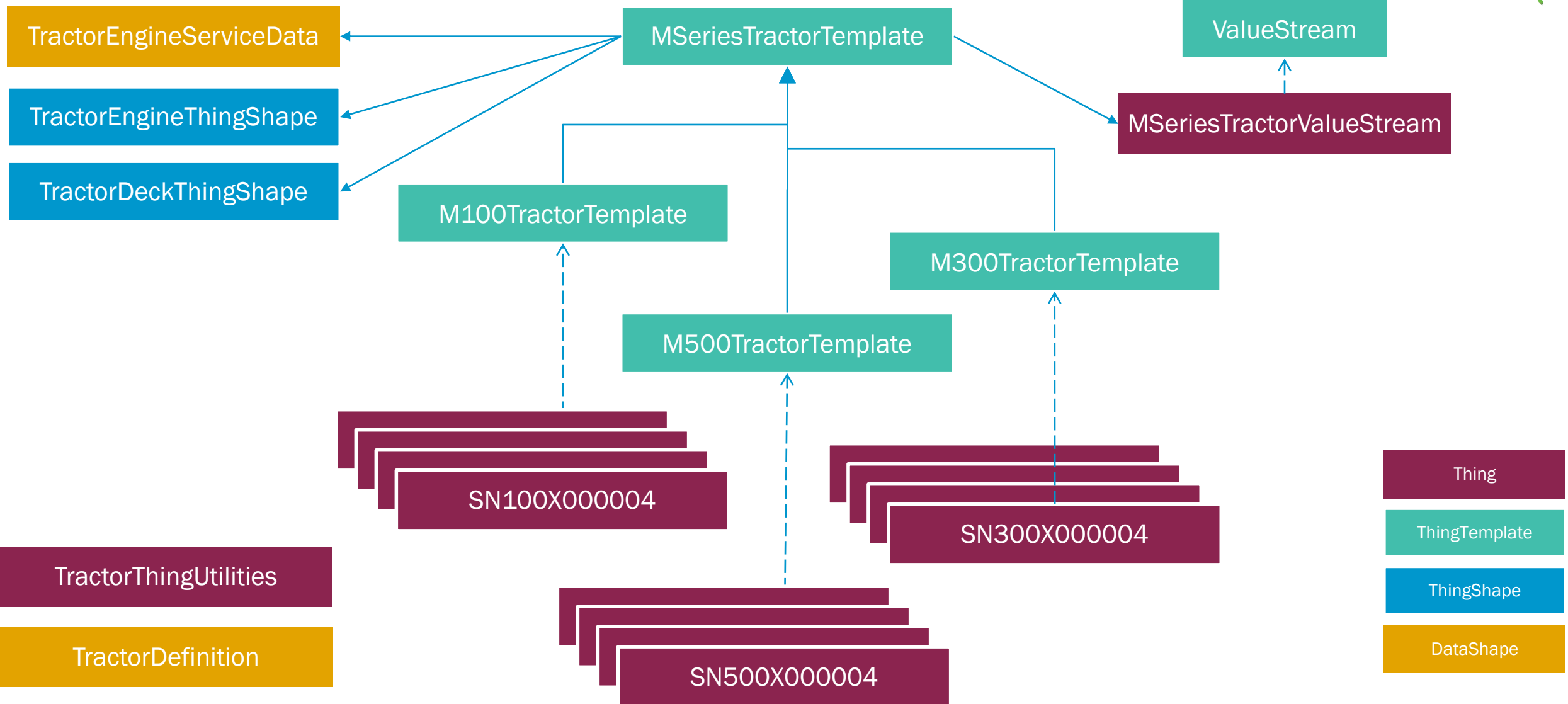
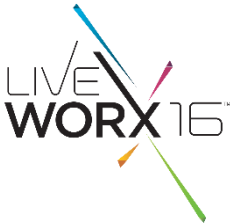
- Modeling Basics
 - Objects in ThingWorx® are referred to as “entities”
 - Many of the entity types can have
 - Properties
 - Services
 - Events
 - Subscriptions
 - Configuration

BUILDING AN IOT SOLUTION – THE MODEL



- Thing:
 - An instance of an actual asset/device/real-world physical thing
 - E.g., the specific Acme Tractor sitting in my garage
 - Can extend ThingTemplates
 - Can derive additional characteristics from ThingShapes
- ThingTemplates:
 - The general definition of the asset/device/real-world physical thing being modeled
 - E.g., all Acme Tractors of a certain model will have the same definition
 - Can extend other ThingTemplates
 - Can derive additional characteristics from ThingShapes
- ThingShapes:
 - Define services, properties, events, that are applicable across different types of Things or ThingTemplates

BUILDING AN IOT SOLUTION – THE MODEL



DEMO – THE MODEL

The screenshot shows the ThingWorx APTC Business web interface. The top navigation bar includes a search field, a search button, and menu items for New Entity, Import/Export, Monitoring, Help, Learning Connector, and a user profile (BStob). The left sidebar contains a navigation menu with categories: MODELING (Things, Thing Templates, Thing Shapes, Data Shapes, Networks, Projects, Model Tags), ANALYTICS (Data Analysis Definitions), VISUALIZATION (Mashups, Masters, Gadgets, Dashboards, Menus, Media, Style Definitions, State Definitions), and DATA STORAGE (Data Tables, Streams, Value Streams, Data Tags). The main content area displays a list of 'Things' filtered by 'Exclude System Objects'. The search criteria is 'SN'. The list shows 12 items, each with a checkbox, a search icon, a gear icon, a serial number, a description, a timestamp, and a lock icon.

SN	Description	Timestamp
SN100X000001	An M100 tractor	2016-06-02 16:04:01.587
SN100X000002	An M100 tractor	2016-06-02 16:04:01.618
SN100X000003	An M100 tractor	2016-06-02 16:04:01.649
SN100X000004	An M100 tractor	2016-06-02 16:04:01.680
SN300X000001	An M300 tractor	2016-06-02 16:04:01.711
SN300X000002	An M300 tractor	2016-06-02 16:04:01.743
SN300X000003	An M300 tractor	2016-06-02 16:04:01.774
SN300X000004	An M300 tractor	2016-06-02 16:04:01.805
SN500X000001	An M500 tractor	2016-06-02 16:04:01.836
SN500X000002	An M500 tractor	2016-06-02 16:04:01.867
SN500X000003	An M500 tractor	2016-06-02 16:04:01.899
SN500X000004	An M500 tractor	2016-06-02 16:04:01.930

AGENDA

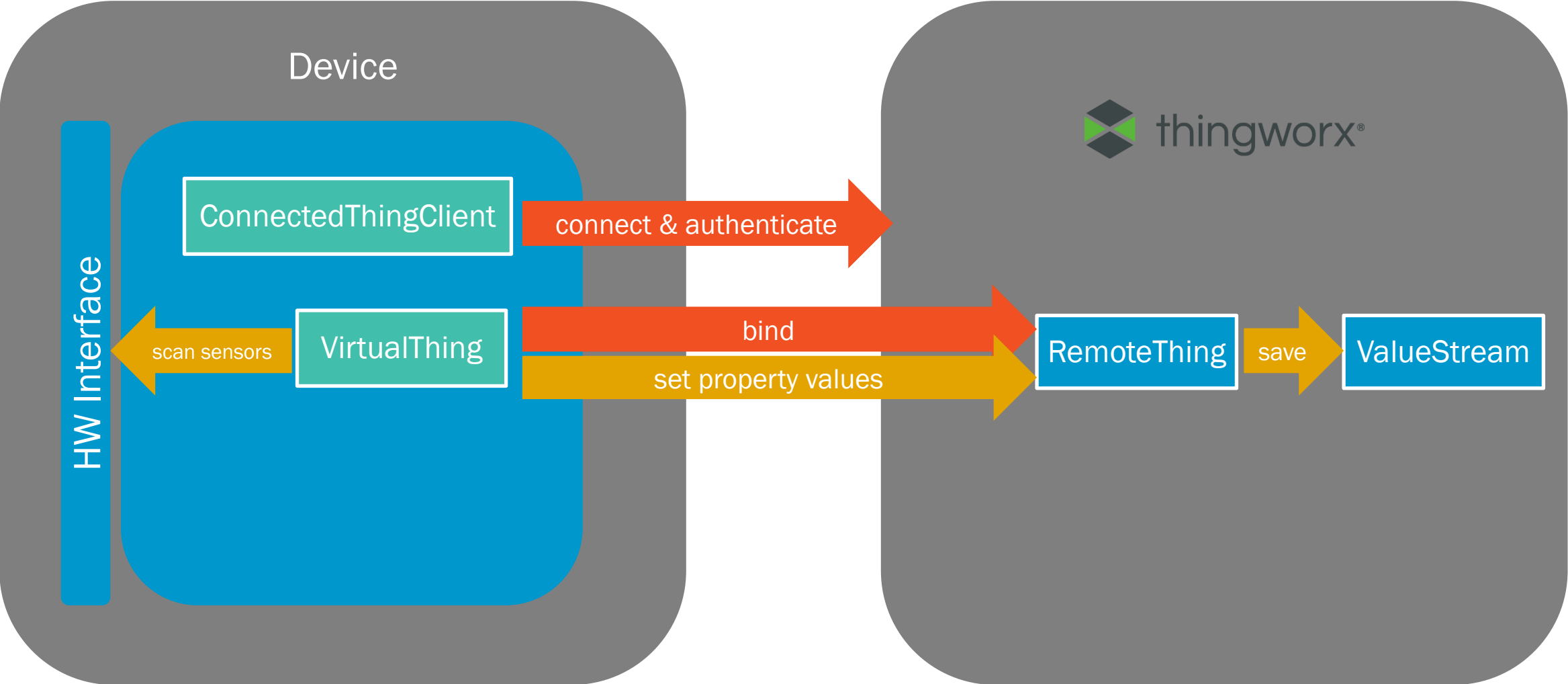
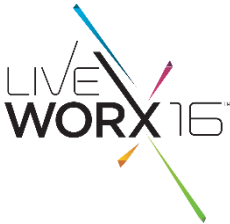
- Scenario: Acme Garden Tractor Co.
- The Model – Best Practices
- Connectivity - Edge Device Communication with ThingWorx[®]**
- Apps – Rapid Application Development
- Creating an Extension using the Eclipse Plugin
- Best Practices for Team Development

BUILDING AN IOT SOLUTION - CONNECTIVITY



- Develop custom application to interface with the device and ThingWorx[®]
- Options for interfacing to the ThingWorx[®] Platform:
 - ThingWorx[™] WS EMS (WebSocket-based Edge Micro Server)
 - HTTP RESTful APIs
 - Lua Script
 - ThingWorx[™] Edge SDK
 - Java, C, .NET, Android, iOS, Protocol Adaptor
- Options for interfacing with device hardware
 - Use hardware-specific libraries

BUILDING AN IOT SOLUTION - CONNECTIVITY



BUILDING AN IOT SOLUTION - CONNECTIVITY



- Model must have a RemoteThing for each device being connected
 - Usually have same name as the VirtualThing on the device, but not required
 - It needs to have RemoteProperties defined that correspond to the properties defined on the VirtualThing
 - These don't need to be the same name but must be bound to each other
- Manually creating remote things for thousands of devices is obviously not practical
 - The best practice is to create service(s) that programmatically create them
 - The service(s) can be invoked by external systems based on the business process for bringing the devices on-line / into production

- On a custom Thing utility entity or on a Java-based Resource entity, create a Service that:
 1. Invokes EntityServices.CreateThing(), passing in the ThingTemplate that
 2. Enables the Thing
 3. Restarts the Thing

JavaScript snippet:

```
var params = {
  thingTemplateName: "RemoteThing" /* THINGTEMPLATENAME */,
  description: "MyThing" /* STRING */,
  name: serialNumber /* STRING */
};

Resources["EntityServices"].CreateThing(params);
Things[params.name].EnableThing();
Things[params.name].RestartThing();
```

Java snippet:

```
EntityServices es = new EntityServices();
es.CreateThing(thingName, desc, null, "RemoteThing");
Thing myThing = (Thing) EntityUtilities.findEntity(
    thingName, ThingworxRelationshipTypes.Things);

myThing.EnableThing();
myThing.RestartThing();
```

DEMO – TRACTOR THING CREATOR AND SIMULATOR



The screenshot shows the ThingWorx user interface. At the top, there is a navigation bar with the ThingWorx logo, a search bar, and several menu items: '+ New Entity', 'Import/Export', 'Monitoring', 'Help', 'Learning Connector', and 'BStob'. Below the navigation bar is a sidebar with a tree view containing categories like MODELING, ANALYTICS, VISUALIZATION, and DATA STORAGE. The main content area is titled 'Things' and has a search filter set to 'SN'. Below the search bar are buttons for '+ New', 'View', 'Edit', 'Duplicate', 'Delete', and 'Permissions'. The main area displays a table of tractor entities with columns for selection, search, settings, SN, description, creation date, and permissions. The table is filtered to show 12 items, all created on 2016-06-02.

<input type="checkbox"/>	<input type="checkbox"/>		SN100X000001	An M100 tractor	2016-06-02 16:04:01.587	
<input type="checkbox"/>	<input type="checkbox"/>		SN100X000002	An M100 tractor	2016-06-02 16:04:01.618	
<input type="checkbox"/>	<input type="checkbox"/>		SN100X000003	An M100 tractor	2016-06-02 16:04:01.649	
<input type="checkbox"/>	<input type="checkbox"/>		SN100X000004	An M100 tractor	2016-06-02 16:04:01.680	
<input type="checkbox"/>	<input type="checkbox"/>		SN300X000001	An M300 tractor	2016-06-02 16:04:01.711	
<input type="checkbox"/>	<input type="checkbox"/>		SN300X000002	An M300 tractor	2016-06-02 16:04:01.743	
<input type="checkbox"/>	<input type="checkbox"/>		SN300X000003	An M300 tractor	2016-06-02 16:04:01.774	
<input type="checkbox"/>	<input type="checkbox"/>		SN300X000004	An M300 tractor	2016-06-02 16:04:01.805	
<input type="checkbox"/>	<input type="checkbox"/>		SN500X000001	An M500 tractor	2016-06-02 16:04:01.836	
<input type="checkbox"/>	<input type="checkbox"/>		SN500X000002	An M500 tractor	2016-06-02 16:04:01.867	
<input type="checkbox"/>	<input type="checkbox"/>		SN500X000003	An M500 tractor	2016-06-02 16:04:01.899	
<input type="checkbox"/>	<input type="checkbox"/>		SN500X000004	An M500 tractor	2016-06-02 16:04:01.930	

AGENDA

- Scenario: Acme Garden Tractor Co.
- The Model – Best Practices
- Connectivity - Edge Device Communication with ThingWorx®
- Apps – Rapid Application Development**
- Creating an Extension using the Eclipse Plugin
- Best Practices for Team Development

- Web apps can be quickly and easily created using the ThingWorx[®] Mashup and Visualization tools in Composer.
- The Mashup editor provides a rich set of widgets that can be bound to the various services from the model to display data, all without having to do any coding

DEMO – ACME TRACTOR MASHUPS

ThingWorx A PTC Business

Type to search system

+ New Entity Import/Export Monitoring Help Learning Connector BStob

Mashups Type to filter list... Advanced Clear

+ New View Edit Duplicate Delete Permissions

Filtering by: Exclude System Objects Showing: 7 items

<input type="checkbox"/>									
<input type="checkbox"/>		OwnerMashup	view for the owner a single tractor	2016-06-03	10:03:26.017				
<input type="checkbox"/>		DealerMashup	Mashup view for Tractor Dealers	2016-06-03	10:02:48.513				
<input type="checkbox"/>		ServiceRecordMashupPart		2016-06-03	08:52:29.176				
<input type="checkbox"/>		EngineMaintenanceMashup	Has the Relevant information for Engine Maintenance	2016-06-02	16:34:09.595				
<input type="checkbox"/>		EngineTemperatureMashupPart	The Engine Temperature Gauge used for	2016-06-02	16:28:03.703				
<input type="checkbox"/>		TractorDetailMashupPart	Shows the details of the tractor, like Purchase Date, Owner, etc.	2016-06-02	13:27:53.295				
<input type="checkbox"/>		AcmeTractorMasterMashup	Main Mashup for all Acme Tractor Views	2016-06-02	12:59:16.612				

AGENDA

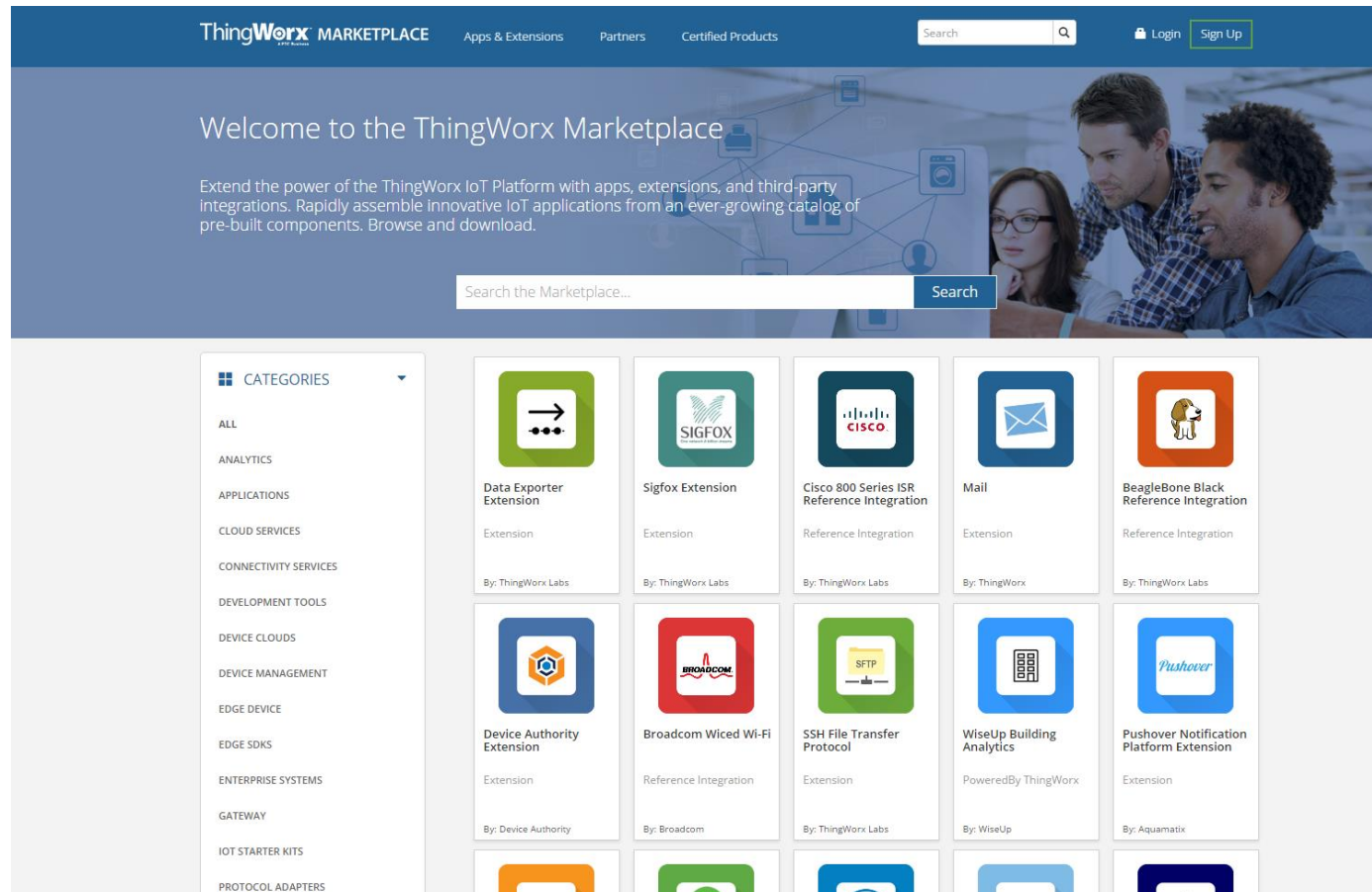
- Scenario: Acme Garden Tractor Co.
- The Model – Best Practices
- Connectivity - Edge Device Communication with ThingWorx®
- Apps – Rapid Application Development
- Creating an Extension using the Eclipse Plugin**
- Best Practices for Team Development

- What is an Extension?
 - Collection of entities, resources, and widgets used to expose new functionality in the ThingWorx Platform.
 - Packaged into a .zip file which can be imported to any ThingWorx® Platform
 - E.g., the Mail extension adds the ability to send emails from ThingWorx®
- Why create Extensions?
 - The best practice for building your solution is to do as much as possible in ThingWorx® Composer and Mashup Builder using the tools it offers.
 - In some cases, it's necessary to build extensions to:
 - Use 3rd-party Java libraries not part of the platform
 - Create global service(s) on a custom Resource Java-based entity
 - Build a custom JavaScript widget to use in mashups
 - Build custom authenticators and/or directory services

BUILDING AN IOT SOLUTION - CREATING EXTENSIONS



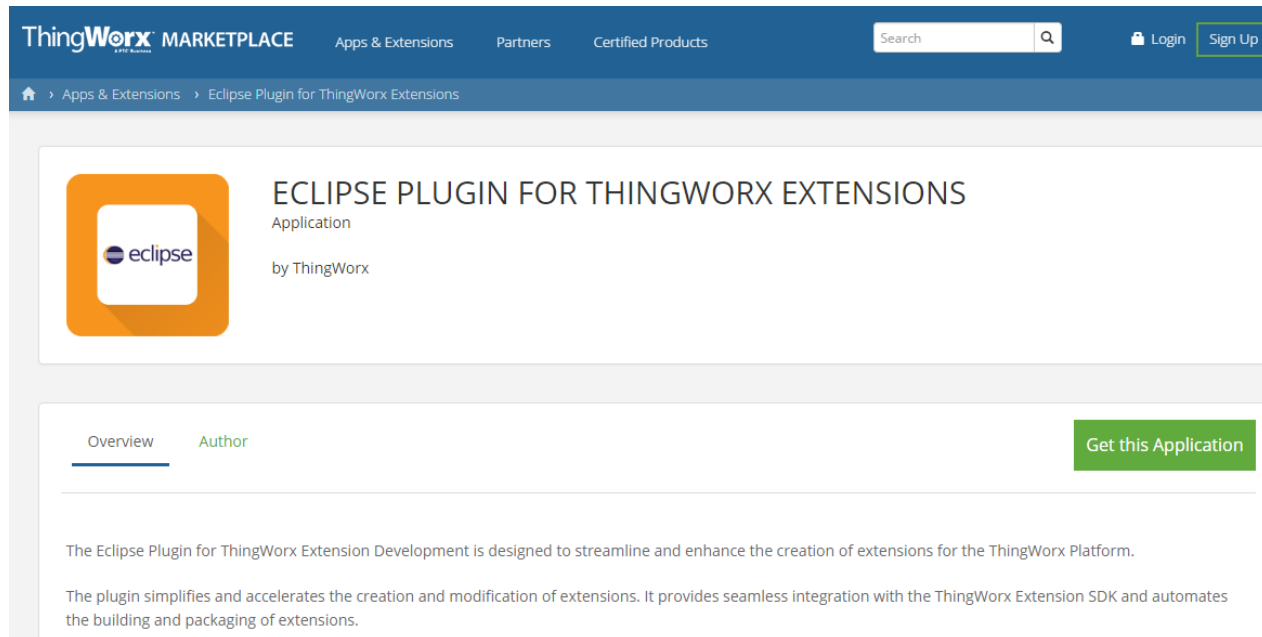
- There is a large collection of Extensions in the ThingWorx[®] Marketplace:
 - <http://marketplace.thingworx.com/>



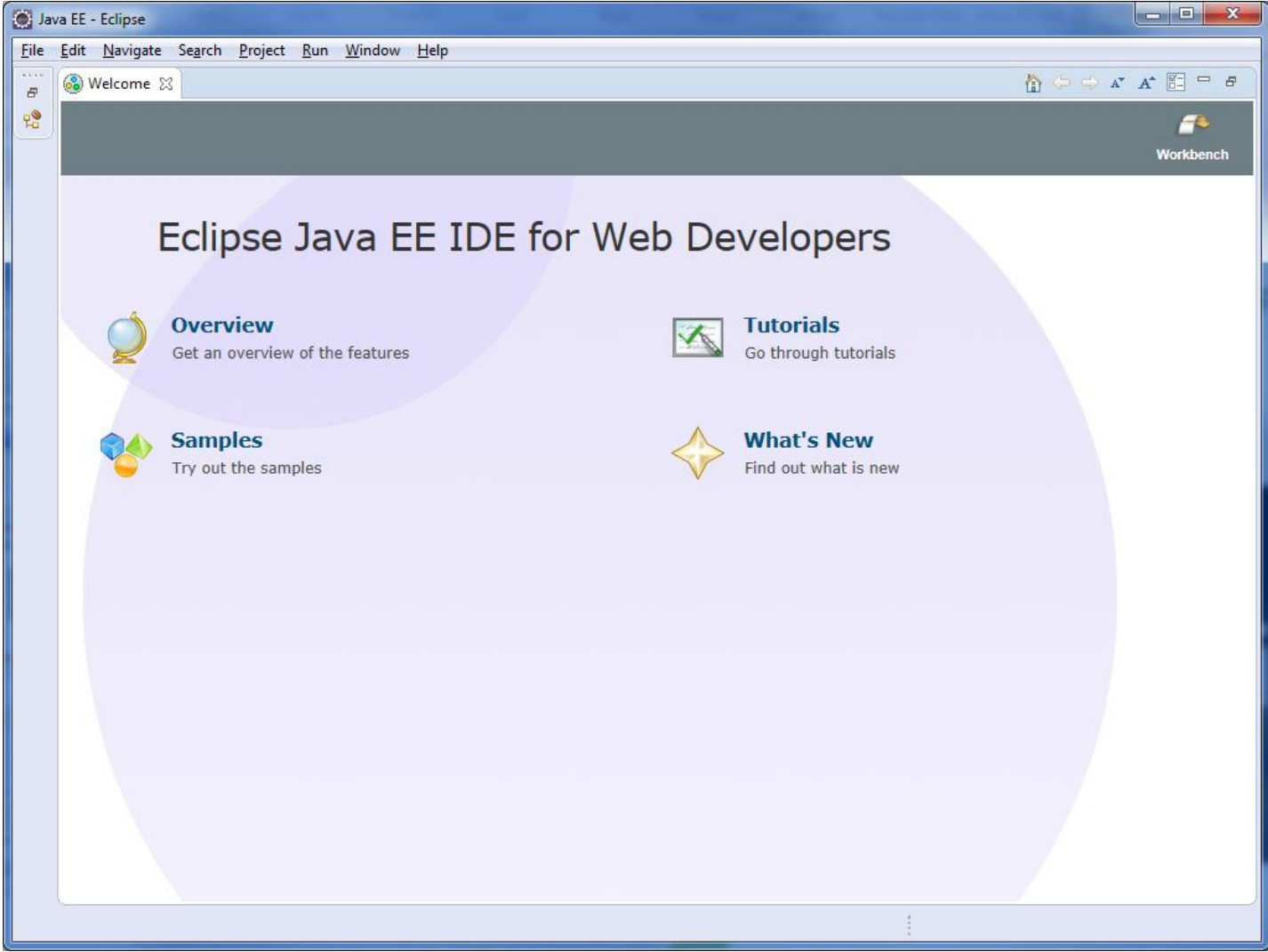
BUILDING AN IOT SOLUTION - CREATING EXTENSIONS



- The following resources are available to help creating extensions:
 - ThingWorx® Extension Development Guide
 - Eclipse Plugin for ThingWorx® Extension
 - Extension SDK
- Available on the ThingWorx Marketplace:
 - <http://marketplace.thingworx.com/Items/eclipse-plugin-for-thingworx-extensions>



DEMO – ECLIPSE PLUGIN FOR EXTENSION DEVELOPMENT



AGENDA

- Scenario: Acme Garden Tractor Co.
- The Model – Best Practices
- Connectivity - Edge Device Communication with ThingWorx®
- Apps – Rapid Application Development
- Creating an Extension using the Eclipse Plugin
- Best Practices for Team Development**



- Projects
 - Provide a grouping capability for keeping a collection of entities together
 - Used in the export/import process to
 - Move the application through the landscape (development, QA, production, etc., environments)
 - Upload to source control
 - Add the ability to add dependencies between projects



- Exporting to Source Code Repository
 - Exports organized individual XML files in folders by project and entity type
 - Individual developers should develop against a local ThingWorx instance and import/export from a local directory that is set up for source control
 - Helps to avoid conflicting changes made at the same time on a server
 - Process:
 - Create a Thing using the “SourceControlRepository” Thing Template.
 - Configure the “rootPath” setting of the thing to point to the source control directory
 - Can be anywhere on server’s file system



- Diff Tool
 - During import, can be used to compare the contents of the import with what is already in the platform
 - Based on the differences, the user can select/deselect what to import

DEMO – PROJECTS, EXPORT FOR SOURCE CONTROL, DIFFERENCES TOOL

The screenshot shows the ThingWorx APTC Business web interface. The top navigation bar includes 'Type to search system', '+ New Entity', 'Import/Export', 'Monitoring', 'Help', 'Learning Connector', and 'BStob'. The left sidebar contains a navigation menu with categories: MODELING (Things, Thing Templates, Thing Shapes, Data Shapes, Networks, Projects, Model Tags), ANALYTICS (Data Analysis Definitions), VISUALIZATION (Mashups, Masters, Gadgets, Dashboards, Menus, Media, Style Definitions, State Definitions), and DATA STORAGE (Data Tables, Streams, Value Streams, Data Tags). The main content area shows a search for 'SN' with a filter set to 'Exclude System Objects'. Below the search bar are buttons for '+ New', 'View', 'Edit', 'Duplicate', 'Delete', and 'Permissions'. The table displays 12 items, each with a checkbox, a search icon, a gear icon, a serial number, a description, a timestamp, and a lock icon.

SN	Description	Timestamp
SN100X000001	An M100 tractor	2016-06-02 16:04:01.587
SN100X000002	An M100 tractor	2016-06-02 16:04:01.618
SN100X000003	An M100 tractor	2016-06-02 16:04:01.649
SN100X000004	An M100 tractor	2016-06-02 16:04:01.680
SN300X000001	An M300 tractor	2016-06-02 16:04:01.711
SN300X000002	An M300 tractor	2016-06-02 16:04:01.743
SN300X000003	An M300 tractor	2016-06-02 16:04:01.774
SN300X000004	An M300 tractor	2016-06-02 16:04:01.805
SN500X000001	An M500 tractor	2016-06-02 16:04:01.836
SN500X000002	An M500 tractor	2016-06-02 16:04:01.867
SN500X000003	An M500 tractor	2016-06-02 16:04:01.899
SN500X000004	An M500 tractor	2016-06-02 16:04:01.930

ADDITIONAL RESOURCES

BUILDING AN IOT SOLUTION – ADDITIONAL RESOURCES



- ThingWorx[®] Developer Zone
 - <http://www.thingworx.com/developer>
- ThingWorx[®] Community
 - <https://community.thingworx.com/welcome>
- ThingWorx[®] Marketplace
 - <http://marketplace.thingworx.com/>
- ThingWorx[®] Reference Documents
 - https://support.ptc.com/appserver/cs/doc/refdoc.jsp?p=browse_results&Product=ThingWorx
- PTC Support
 - <https://support.ptc.com>

BUILDING AN IOT SOLUTION – CONTACT US



- We'd love to hear from you!
- Ben Stob – Ben.Stob@thingworx.com
- Dave Larson - Dave.Larson@thingworx.com
- Jesse Docken - Jesse.Docken@thingworx.com
- Thearon Helgeson – Thearon.Helgeson@thingworx.com

QUESTIONS

The image features several colorful geometric shapes, primarily triangles and lines, scattered across the background. A large, multi-colored triangular shape is prominent on the right side, composed of various shades of blue, green, yellow, orange, pink, and purple. Several thin, colored lines (blue, pink, green, orange) radiate from the center towards the edges of the frame. The text 'LIVE WORX 16' is centered in the upper half, with 'LIVE' in a thin, outlined font and 'WORX 16' in a bold, solid black font. A small 'TM' trademark symbol is positioned to the right of the '16'.

LIVE
WORX 16™

TAKE A FRESH LOOK AT THINGS

liveworx.com