GETTING MOBILE WITH THINGWORX AND ANDROID

William Reichardt

Technical Fellow, R&D Engineering, Edge Connectivity

Date 6/3/16

THINGWORX ANDROID SDK IS HERE!

- General Availability Release 1.0 this past April, 2016
- Get it on the ThingWorx MARKETPLACE

  http://marketplace.thingworx.com
- Built on top of the ThingWorx Java SDK
- “Always On” Protocol
- New Features!
- Great Examples
- IDE Ready – Build Great User Interfaces
- Quick Start

WHY DO I NEED AN ANDROID SDK?

- ThingWorx already has a Java SDK
- Android's VM, Dalvik is JDK 6 Compatible
- ThingWorx Java SDK uses Netty (http://netty.io/) for its Websockets
- Netty is based on Java New I/O (NIO) which is part of JDK 7
- Netty was replaced with Autobahn for Android (http://autobahn.ws/)
- Build and App Examples specifically for Android

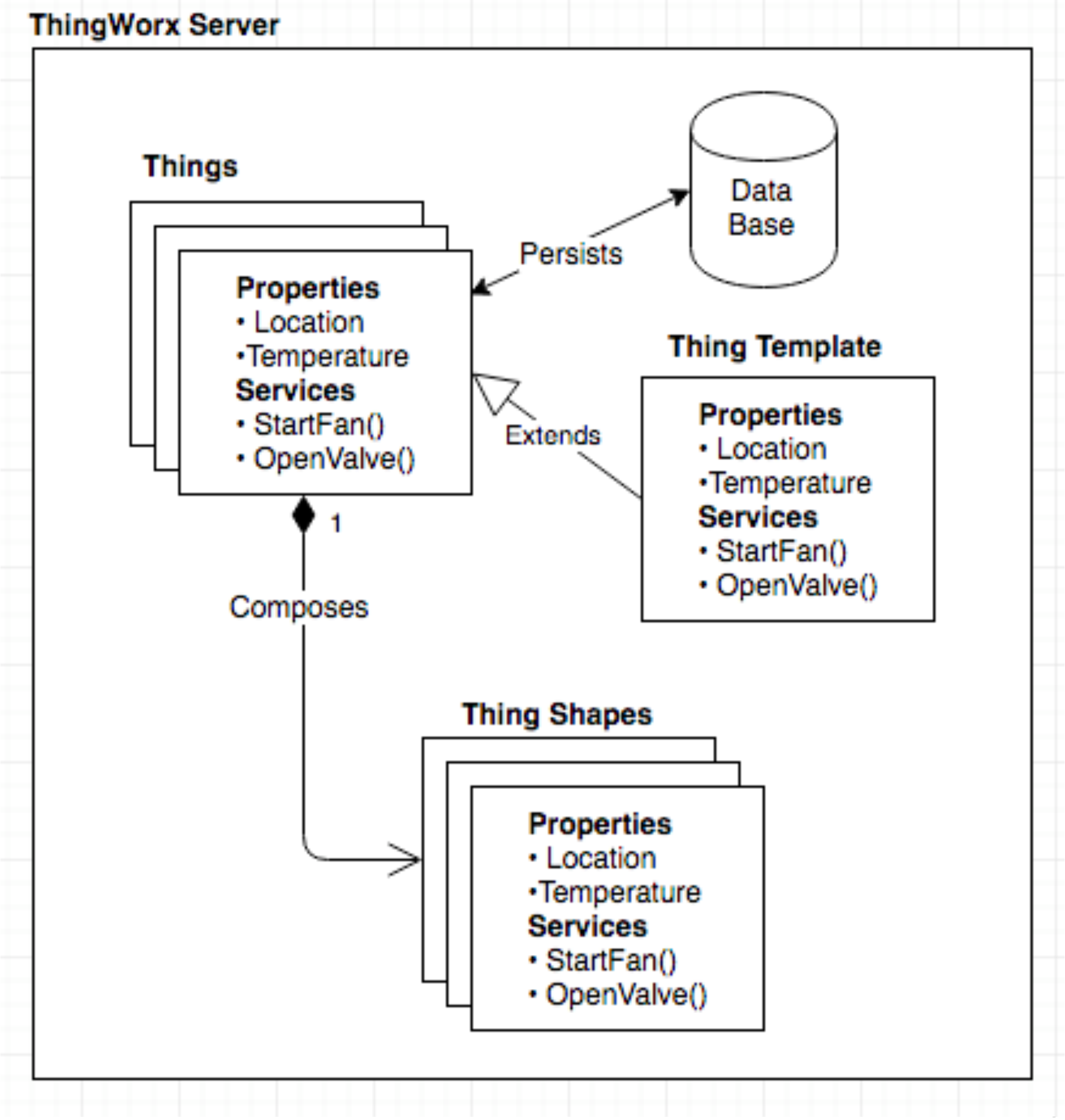
WHAT IS “**ALWAYS ON**” PROTOCOL AND WHY DO I NEED IT?

- Thingworx is REST web service compliant, Any Thing can be modified using an HTTP request
- HTTP connections have connection overhead to establish connections
- “Always On” is Websocket based, two-way “Full Duplex” continuous connection over HTTP
- Great for delivering and receiving continuous data streams on an Edge device with low latency
- It is self-monitoring
  - Automatically re-establishes connections when broken
  - Allows uptime to be configured to conserve network usage
- Provides a Thing Synchronization API between the server and Edge devices
- Provides two way access to Thing Services and Properties callable both from the Server and the Edge

You never know when someone will pull the plug

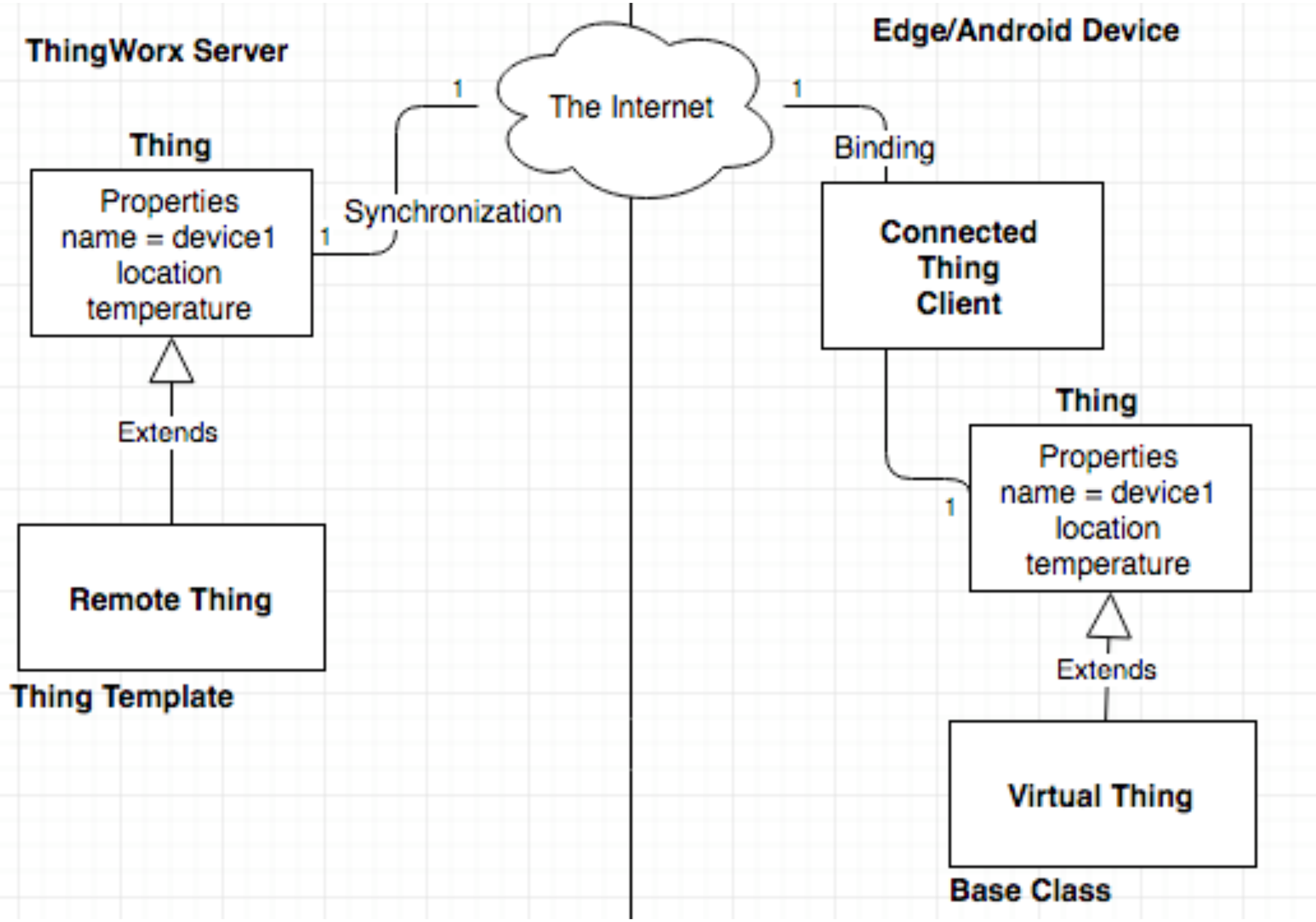
HOW DOES THINGWORX DATA MODELING WORK?

- Modeling in ThingWorx in 30 seconds…
- Things = Objects
- Thing Template = Base Class (Inheritance)
- Thing Shapes = Aggregation Of Functionality
- Properties & Property History can be stored to a database
- Events are omitted



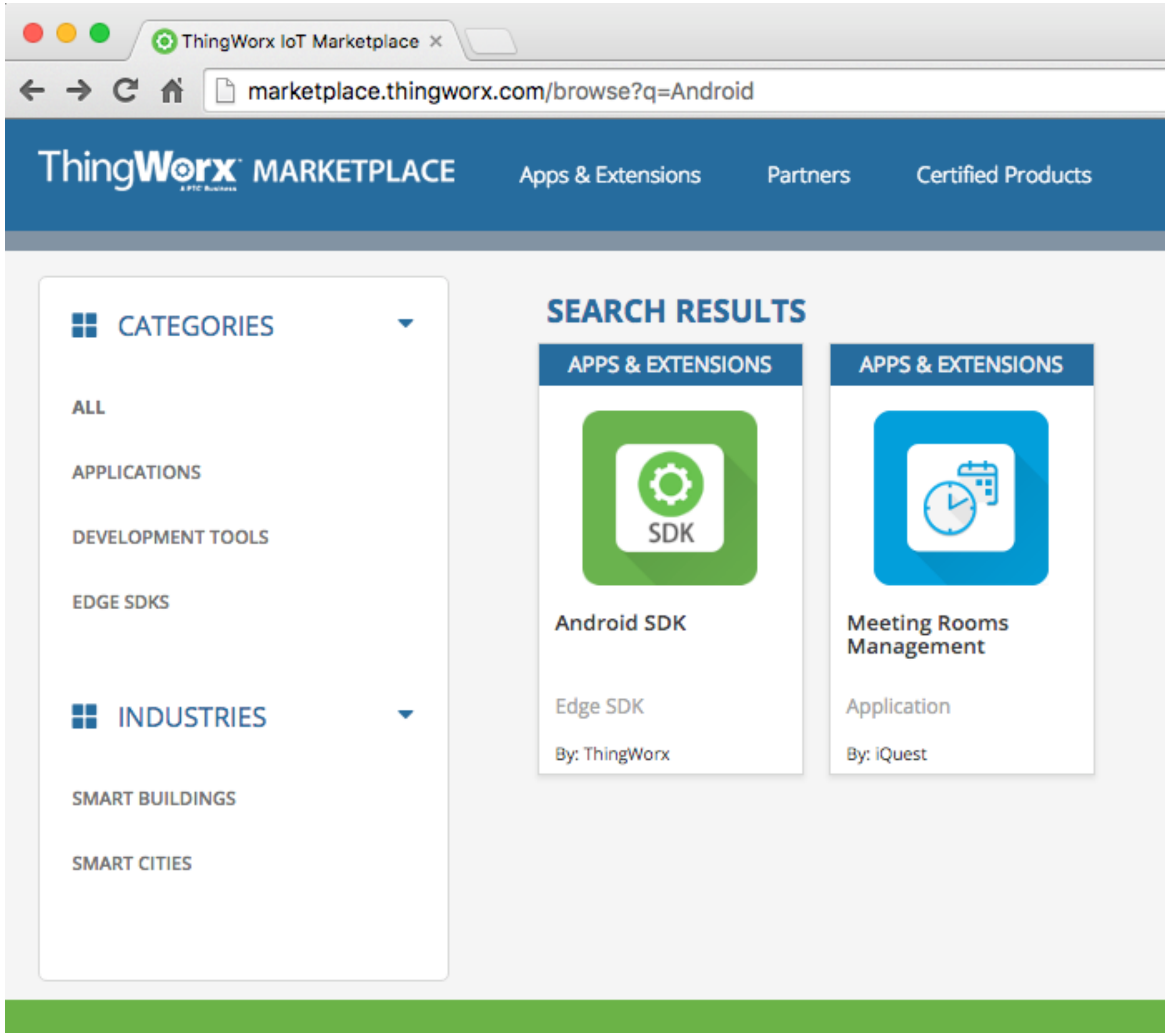
HOW DOES "ALWAYS ON" DATA DELIVERY WORK?

- An Object/Thing is created on your mobile device (In its native language)
- A Thing is created on the ThingWorx server
- Device Thing registers with ConnectThingClient
  - Called "Binding"
- ConnectedThingClient
  - Maintains connection to server
  - Manages Synchronization
  - updateSubscribedProperties()
- Remote and Server Thing "name" properties must match for synchronization to occur



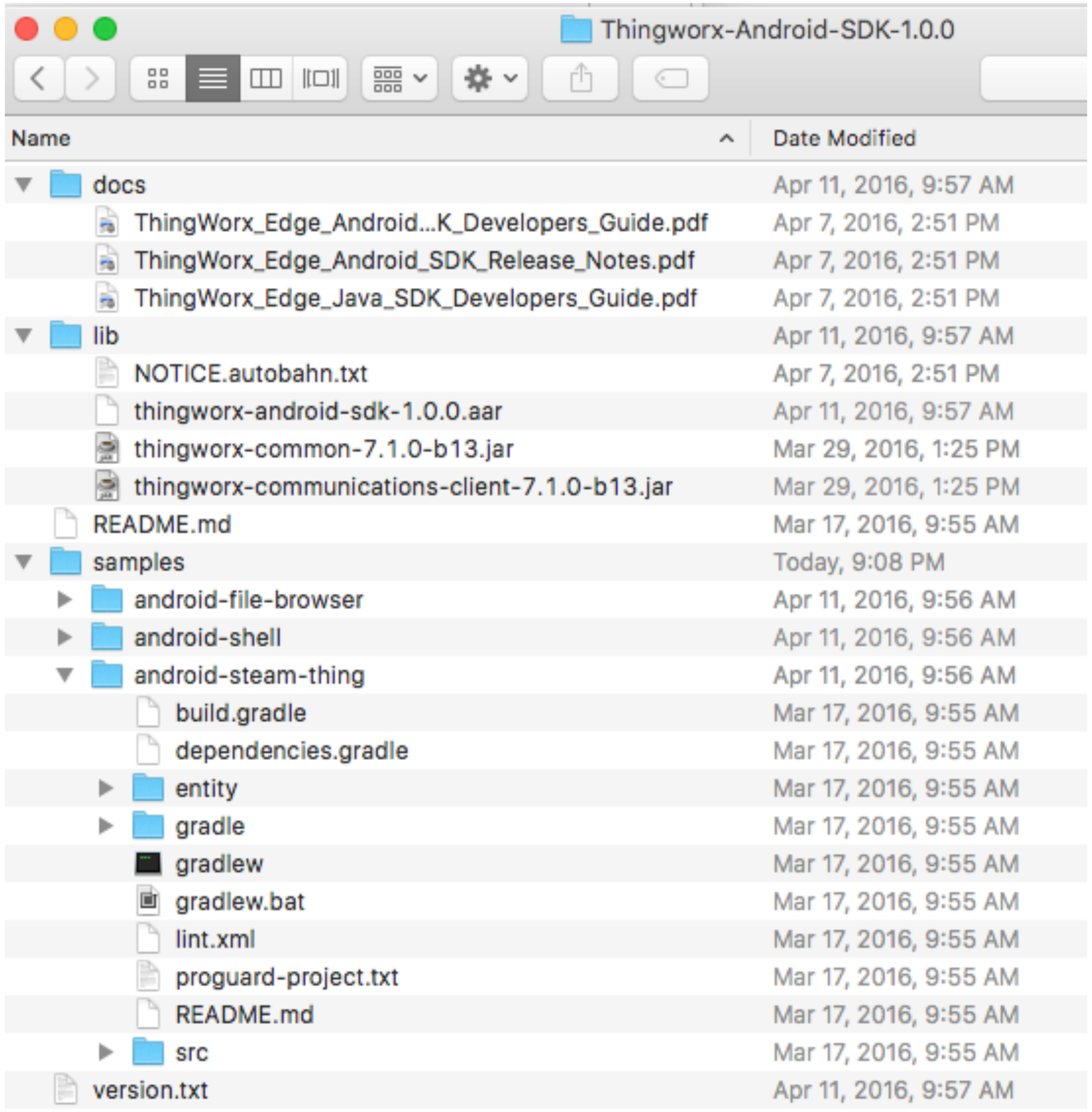
WHAT’S IN THE ANDROID SDK?

- Get it at http://marketplace.thingworx.com/



SEARCH RESULTS

APPS & EXTENSIONS	APPS & EXTENSIONS
Android SDK	Meeting Rooms Management
Edge SDK	Application
By: ThingWorx	By: iQuest

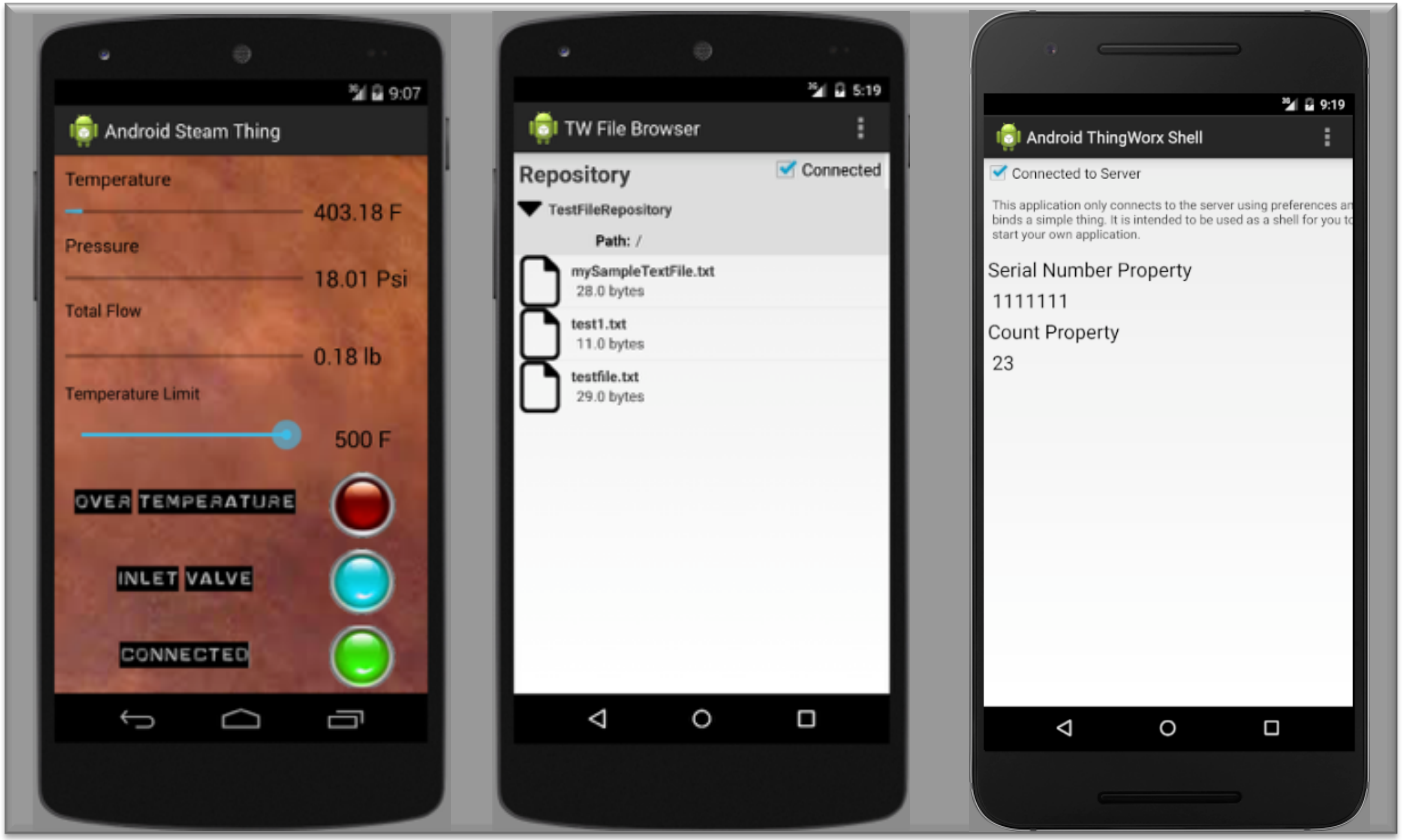


Name	Date Modified
docs	Apr 11, 2016, 9:57 AM
ThingWorx_Edge_Android...K_Developers_Guide.pdf	Apr 7, 2016, 2:51 PM
ThingWorx_Edge_Android_SDK_Release_Notes.pdf	Apr 7, 2016, 2:51 PM
ThingWorx_Edge_Java_SDK_Developers_Guide.pdf	Apr 7, 2016, 2:51 PM
lib	Apr 11, 2016, 9:57 AM
NOTICE.autobahn.txt	Apr 7, 2016, 2:51 PM
thingworx-android-sdk-1.0.0.aar	Apr 11, 2016, 9:57 AM
thingworx-common-7.1.0-b13.jar	Mar 29, 2016, 1:25 PM
thingworx-communications-client-7.1.0-b13.jar	Mar 29, 2016, 1:25 PM
README.md	Mar 17, 2016, 9:55 AM
samples	Today, 9:08 PM
android-file-browser	Apr 11, 2016, 9:56 AM
android-shell	Apr 11, 2016, 9:56 AM
android-steam-thing	Apr 11, 2016, 9:56 AM
build.gradle	Mar 17, 2016, 9:55 AM
dependencies.gradle	Mar 17, 2016, 9:55 AM
entity	Mar 17, 2016, 9:55 AM
gradle	Mar 17, 2016, 9:55 AM
gradlew	Mar 17, 2016, 9:55 AM
gradlew.bat	Mar 17, 2016, 9:55 AM
lint.xml	Mar 17, 2016, 9:55 AM
proguard-project.txt	Mar 17, 2016, 9:55 AM
README.md	Mar 17, 2016, 9:55 AM
src	Mar 17, 2016, 9:55 AM
version.txt	Apr 11, 2016, 9:57 AM

LETS SEE SOME EXAMPLES

- Examples are the best way to learn, this SDK comes with three ready-to-use android applications

- Steam Sensor
  - Properties
  - Services
  - Events
  - DataShapes
- File Browser
  - File Transfers between the Phone and the Server
- Shell Application
  - A starting point for your applications



Temperature 403.18 F

Pressure 18.01 Psi

Total Flow 0.18 lb

Temperature Limit 500 F

Repository ☑ Connected

TestFileRepository

Path: /

mySampleTextFile.txt 28.0 bytes

test1.txt 11.0 bytes

testfile.txt 29.0 bytes

☑ Connected to Server

This application only connects to the server using preferences ar binds a simple thing. It is intended to be used as a shell for you to start your own application.

Serial Number Property 1111111

Count Property 23

BUILDING EXAMPLES

- All sample apps open directly in Android Studio
  http://developer.android.com/sdk/index.html
  Free to download

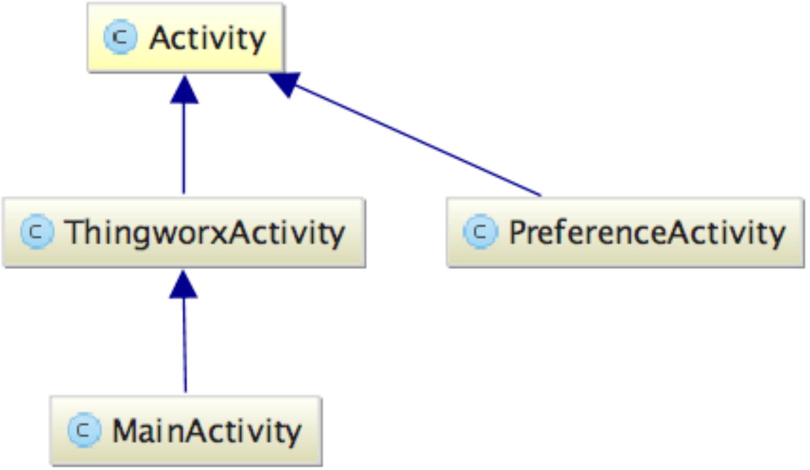
- All sample files also build from the command line using Gradle

DEMOS

- Building the Steam Sample using Gradle
- Building the Shell Example using Android Studio

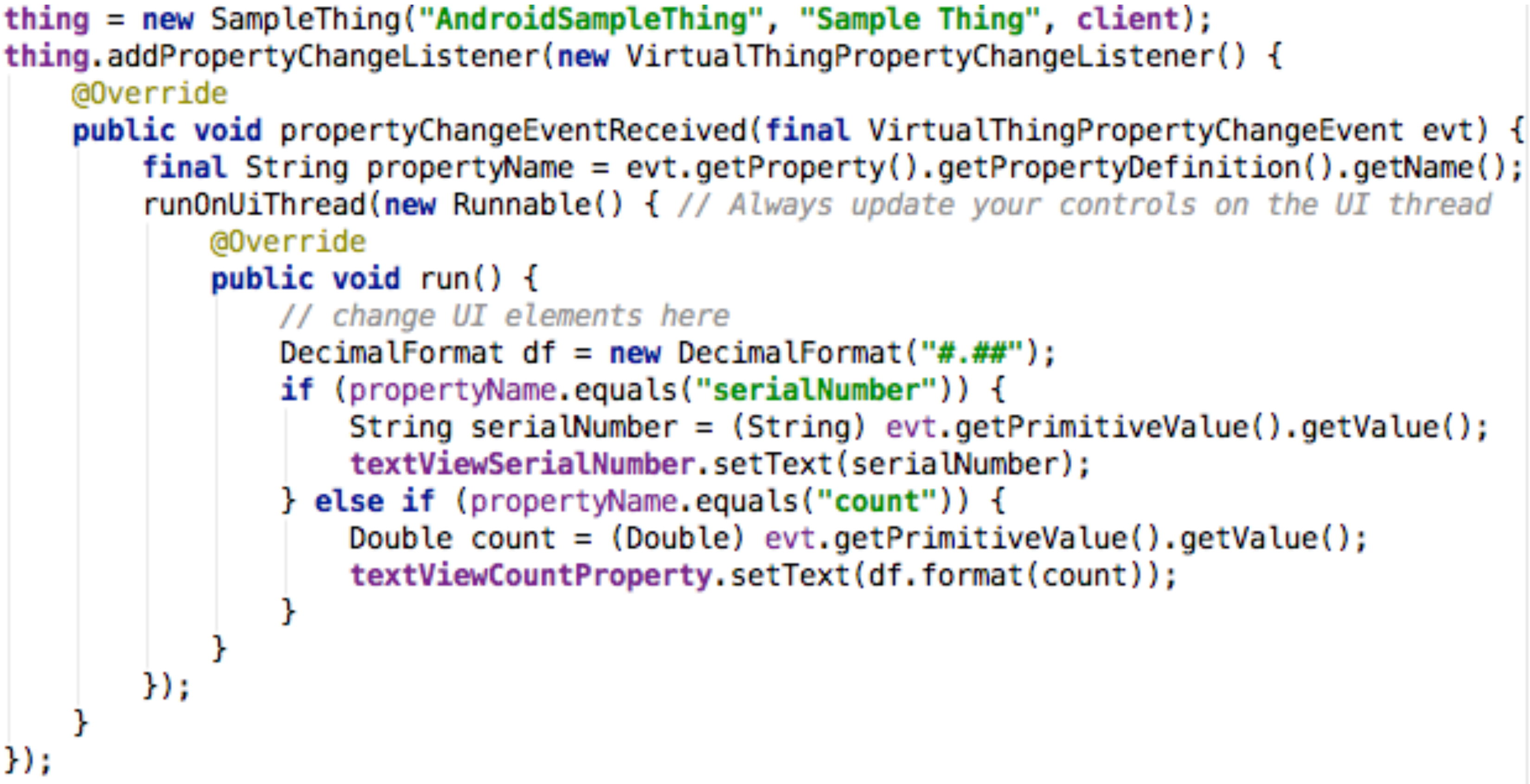
BASE CLASSES MAKE YOUR LIFE EASIER

- An Activity Class Represents a Single Screen in an Android Application
- The ThingWorx Android SDK Examples Extend Activity
- Demonstrate How to Build and Manage a Connection
- ThingworxActivity – Manages your Connection State
- PreferenceActivity – Stores Your Required Server Settings
- MainActivity – Hosts Your User Interface
- Source is Included
- Ready to be Re-used by You



PROPERTY CHANGE LISTENERS ARE JUST LIKE “BINDINGS”

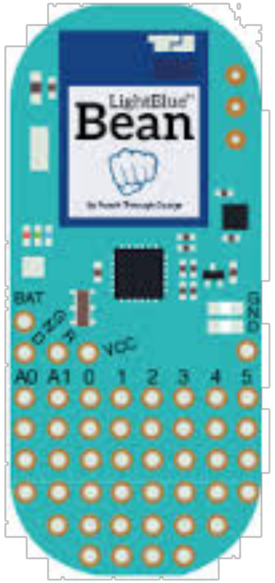
- Android Developers are used to using “Listeners” to be notified of UI Changes
- What if VirtualThings could notify you when their properties change
- Locally or on Your ThingWorx server



```
thing = new SampleThing("AndroidSampleThing", "Sample Thing", client);
thing.addPropertyChangeListener(new VirtualThingPropertyChangeListener() {
    @Override
    public void propertyChangeEventReceived(final VirtualThingPropertyChangeEvent evt) {
        final String propertyName = evt.getProperty().getPropertyDefinition().getName();
        runOnUiThread(new Runnable() { // Always update your controls on the UI thread
            @Override
            public void run() {
                // change UI elements here
                DecimalFormat df = new DecimalFormat("#.##");
                if (propertyName.equals("serialNumber")) {
                    String serialNumber = (String) evt.getPrimitiveValue().getValue();
                    textViewSerialNumber.setText(serialNumber);
                } else if (propertyName.equals("count")) {
                    Double count = (Double) evt.getPrimitiveValue().getValue();
                    textViewCountProperty.setText(df.format(count));
                }
            }
        });
    }
});
```

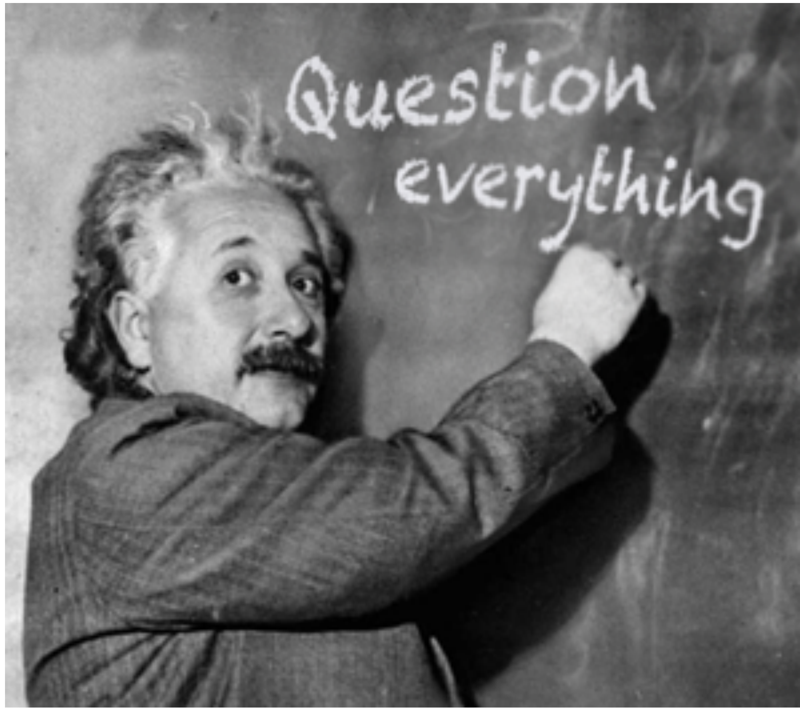
LETS EXPOSE A NEW PROPERTY (DEMO 2)

- Walkthrough of Current Shell Application
- Add a temperature property
- LightBlue Bean (https://punchthrough.com/bean)
  A Bluetooth LE Enabled Arduino with Built-in Temperature Sensor
- Steps
  - Add LightBlue Bean SDK
  - Modify SampleThing to Receive Temperature Readings
  - Add Annotations to Create Temperature Property
  - Update Temperature Property when Data Comes In
  - Create Temperature Text Field in Activity
  - Bind the VirtualThing Property to the Android Activity
- This Example will be posted in the ThingWorx developer community



QUESTIONS?

bill.reichardt@thingworx.com



LIVE WORX 16™

TAKE A FRESH LOOK AT THINGS

liveworx.com