MTConnect 101

Fundamentals of MTConnect

Joel Neidig of ITAMCO

Dave Edstrom of MTConnect Institute

Background on Joel and Dave

- **Joel Neidig**
  - ITAMCO
  - System Engineer
  - MTAG member
  - Wrote first MTConnect mobile app
- **Dave Edstrom**
  - MTConnect President/Chairman
  - 33 years in the computer industry

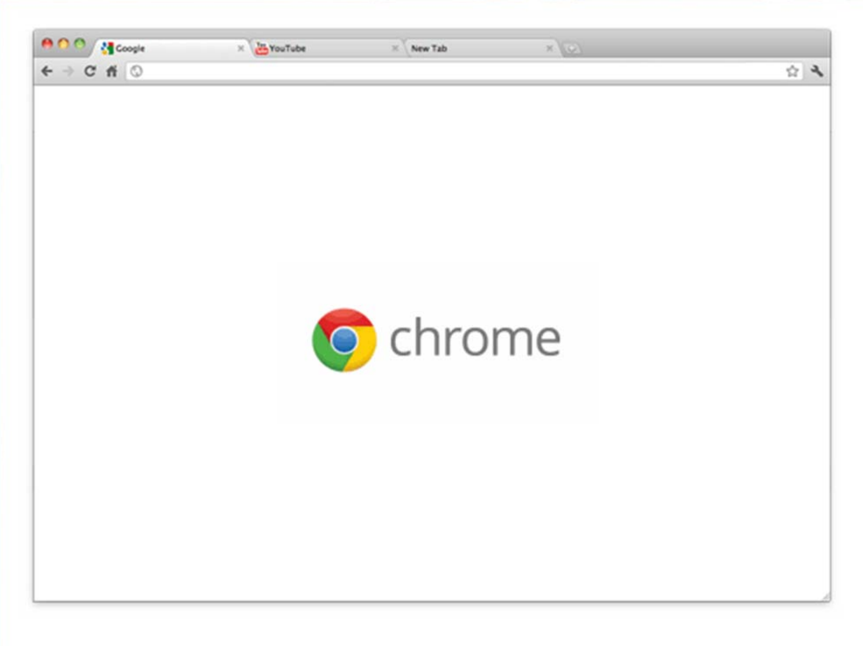
HUGE thanks to Will Sobel who created the basis of this workshop!

Agenda

- Goals and Non Goals of this workshop
- HTTP and XML
- Big Picture of MTConnect
- MTConnect Agent
- Schema
- Probe, Sample, Current and Asset Commands
- Documents versus Streaming
- XML namespaces and path
- Demo of Simple Web Client
- Using Excel for MTConnect
- Demo of ITAMCO’s Mobile App
- Where and How To Keep Learning About MTConnect
- Summary

Prerequisites For MTConnect 101

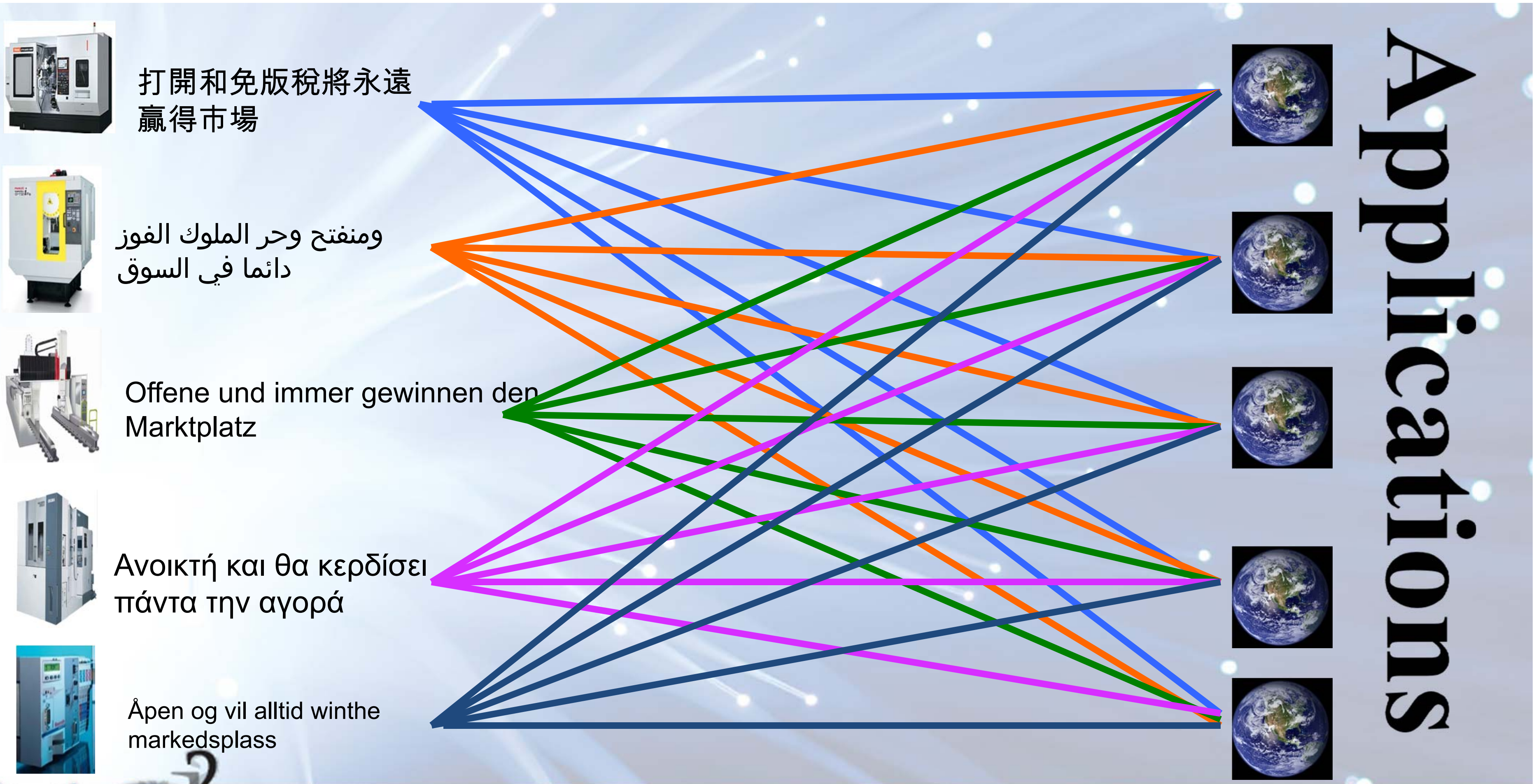
- Chrome or Firefox
- Notebook Computer
- Desire to learn



Goals and Non-Goals

- **General goals of workshops: accelerate your ability to start using and leveraging MTConnect**
  - tutorial-style introductions to the fundamental aspects of MTConnect architecture
  - hands-on lab exercises using real or simulated manufacturing equipment
  - this lab will move at a good pace, with emphasis on the hands-on sections
  - will provide a solid foundation to understand fundamentals of MTConnect
- **Non-goals: 100% depth and 100% breadth**
  - won't make you an expert
  - won't convey every detail of the official MTConnect Specification
- **This workshop will help set the stage for the next two workshops on Agents/Adapters and MTConnect Hello World**

BEFORE MTConnect



打開和免版稅將永遠
贏得市場

ومنفتح وحر الملوك الفوز
دائما في السوق

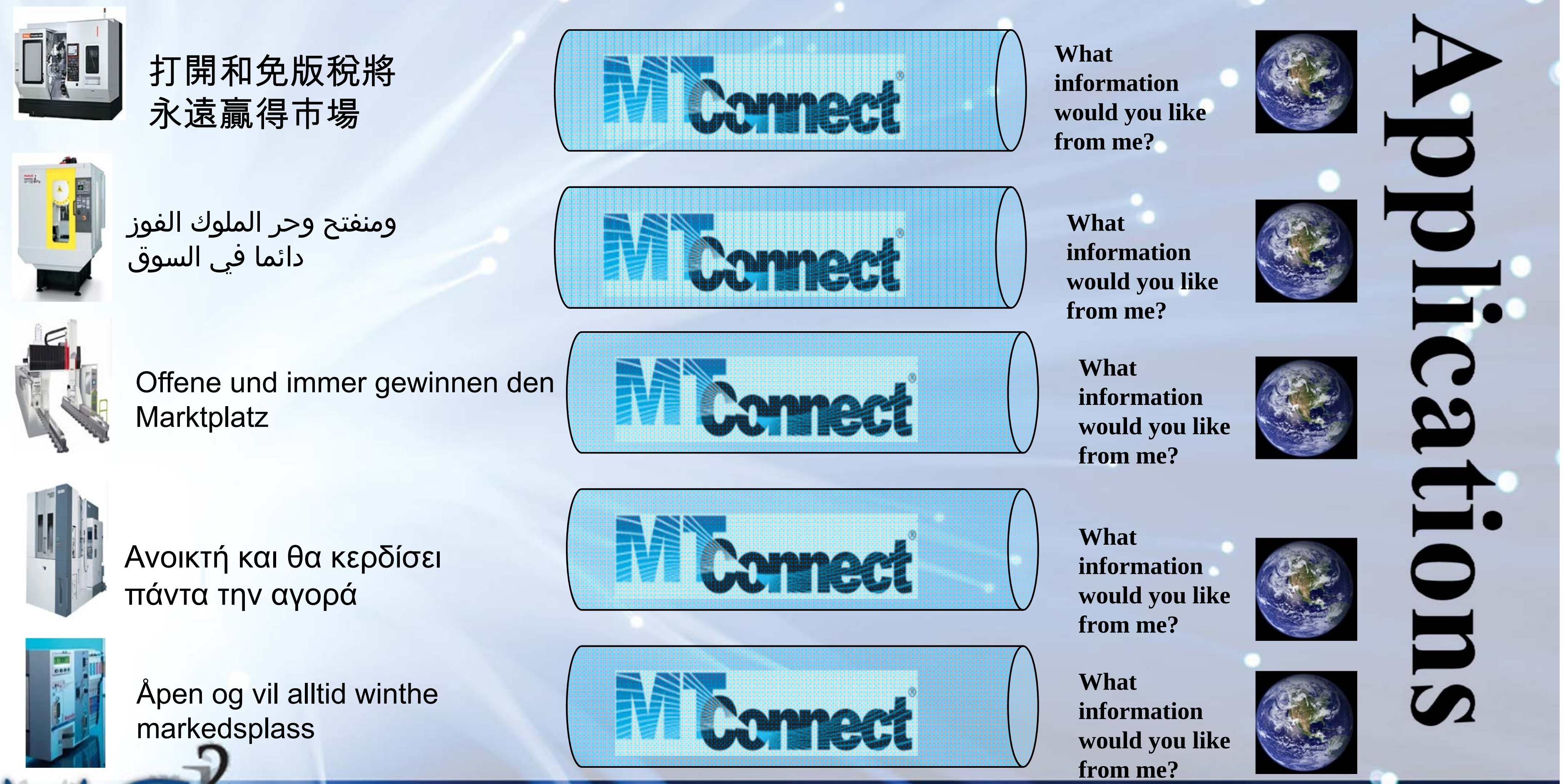
Offene und immer gewinnen den
Marktplatz

Ανοικτή και θα κερδίσει
πάντα την αγορά

Åpen og vil alltid winthe
markedsplass

Applications

Think of MTConnect the Bluetooth For Connecting Manufacturing Equipment to Applications



打開和免版稅將
永遠贏得市場

ومنفتح وحر الملوك الفوز
دائما في السوق

Offene und immer gewinnen den
Marktplatz

Ανοικτή και θα κερδίσει
πάντα την αγορά

Åpen og vil alltid winthe
markedsplass

What information would you like from me?

What information would you like from me?

What information would you like from me?

What information would you like from me?

What information would you like from me?

Applications

Reminder of How HTTP works

- HTTP example: downloading a white paper from MTConnect website
  - user-agent (browser) opens TCP/IP connection to www.mtconnect.org
  - User-agent transmits to server via TCP/IP...

```
GET /media/MTConnectWhitePaper.doc  HTTP/1.1
```

Server replies...

```
HTTP/1.1  200  OK
Content-type: application/x-msword
Content-length: 19171
```

...actual content follows....

MTConnect Data Representation: Review of XML

- eXtensible Markup Language
- Markup language that can represent hierarchical data
- XML *document* is **the** unit of “exchanging data” on the internet today
- XML Schema defines what is “allowed” in a particular “vocabulary” of XML
  - e.g.: “this field must be numeric”
  - e.g.: “there must be exactly one instance of this field; that other field is optional”
  - e.g.: “only the following specific values are allowed for this field” [like country names, e.g.]

Data Representation: XML

- eXtensible Markup Language—represent textual, semi-structured, possibly hierarchical data
  - Shares common ancestor (SGML, Standard Generalized Markup Language) with HTML
- All modern Web browsers can display XML directly

```
<instructors>
  <person name=”Joel">
     <institution>ITAMCO</institution>
     <windows_user/>
  </person>
  <person name=”Dave" title="lecturer">
     <institution type="public">

     </institution>
  </person>
</instructors>
```

Open tag

Attribute & value of “person” tag

Value of “institution” tag

Some tags may have no associated value (only their presence matters)

Whitespace usually not significant

Close tag

Extensibility Is The Key With XML

- Every major protocol in history has undergone post-deployment revision
  - Corollary: more important to get it *extensible* than to get it *right* (since you won't get it right)
- **MTConnect approach**: leverage existing standards & protocols that have *proven* their extensibility
  - HTTP for communication
  - XML for data representation
  - MTConnect Working Groups define MTConnect vocabulary

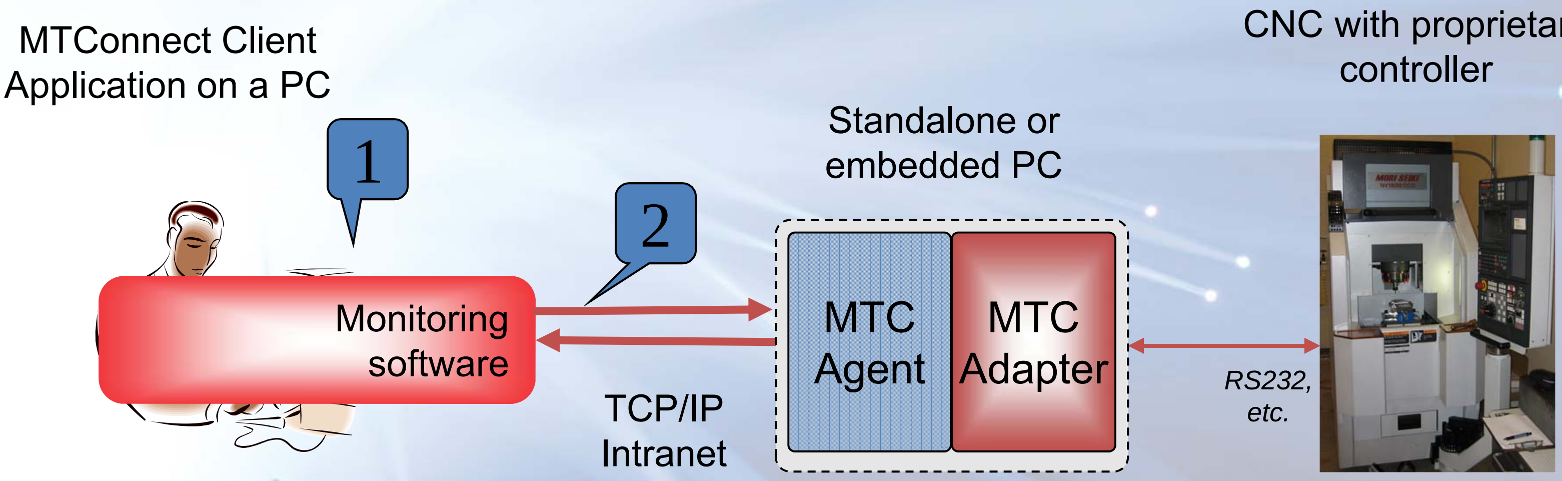
Very Simple Protocol

1. Request to an MTConnect Agent is encoded in URI
2. URI transmitted to the Agent as *HTTP* request

Machine data is “just another” thing or website on the Web.

In fact you can use a Web browser to examine it directly.

3. **MTConnect is NOT an application.** MTConnect is a protocol



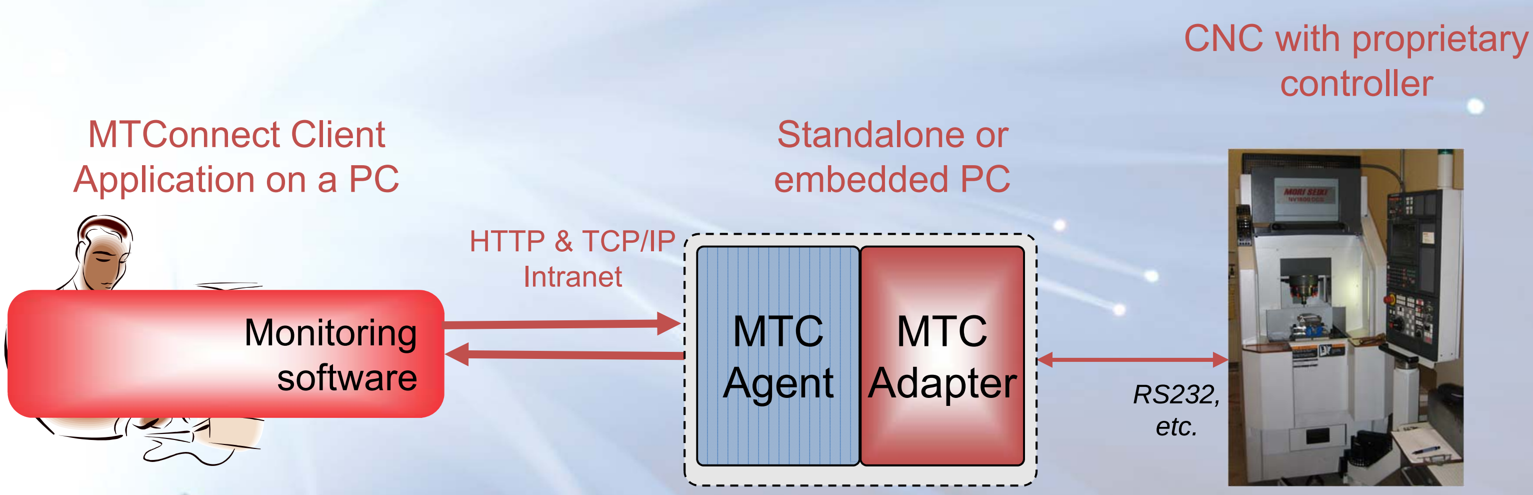
MTConnect Client Application on a PC

Standalone or embedded PC

CNC with proprietary controller

MTConnect From 10,000 Feet

- *Client application* sends a command/request to one or more *MTConnect Agents* via TCP/IP network
- Agent "represents" device in the MTConnect environment
  - Agent software may be built into controller, or running on an embedded device like a black box or standalone PC
  - Adapter is optional and comes into play if CNC does not natively speak MTConnect. Adapter is the software that sits between the agent and the manufacturing equipment
- Agent sends back results



CNC with proprietary controller

MTConnect Client Application on a PC

Standalone or embedded PC

HTTP & TCP/IP Intranet

Monitoring software

MTC Agent

MTC Adapter

RS232, etc.

MTConnect Has Minimal Compliance

- Minimal requirements to become MTConnect compliant
- To be compliant the device only needs to report on the power status
  - **Power Status is either ON or OFF**
- Everything else is optional

What MTConnect Must and Must Not Do

- MTConnect must
  - ✓Collect data from devices
  - ✓Normalize the units
  - ✓Deliver the data in a standard format
- MTConnect must not
  - □Analyze the data
  - □Derive additional meaning

Key Concepts: Agent & Client

- ***Agent* “represents” one or more MTConnect-compliant machines**
  - Makes machine data available in uniform MTConnect representations
  - Responds to MTConnect commands
  - Allows *clients* to specify/select which data is of interest
- ***client [application]* wants to collect data**
  - view of the world: a set of MTConnect Agents
  - A plain Web browser is the simplest client
  - Excel is a simple client
  - MTConnect *discovery* mechanism allows clients to ask: “what Agents are in this environment?”
    - often, clients will already know the answer

Agent Software

- Agent software may be part of Controller' s software
- Or, Agent software may run separately (on a separate PC or black box):

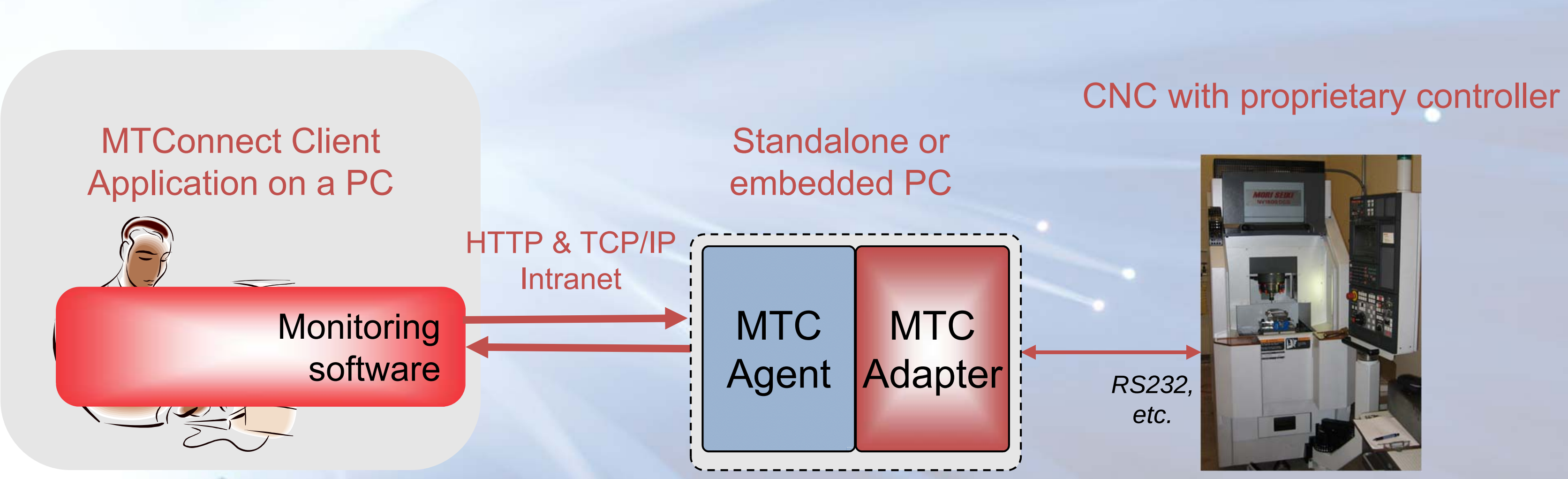
  ...using an *adapter* to communicate with proprietary Controller on specific machine

  ...using an *adapter* to communicate with a *collection of machines* that understand a different standard (e.g. MTConnect-OPC UA, CAMX)

- MTConnect uses sequences numbers to track and order requests/samples

Basic Scenario

- Client application may be a Web-based app or “standalone” app
  - Many compelling applications can be authored to run entirely inside a Web browser
  - We will do hands-on exercises directly from a Web browser



MTConnect Client Application on a PC

Monitoring software

HTTP & TCP/IP Intranet

Standalone or embedded PC

MTC Agent

MTC Adapter

RS232, etc.

CNC with proprietary controller

MTConnect General Workflow

1. Agent(s) corresponding to piece(s) of equipment *register* themselves with a *name service.* This makes them *discoverable.*
2. *Applications* start up and (optionally) query name server to discover Agent(s).
    - Application may already know where agent(s) are, e.g. in a relatively static deployment scenario
3. Applications communicate *requests* to Agents (via HTTP) and receive *responses* (in XML).

The “Bucket” model

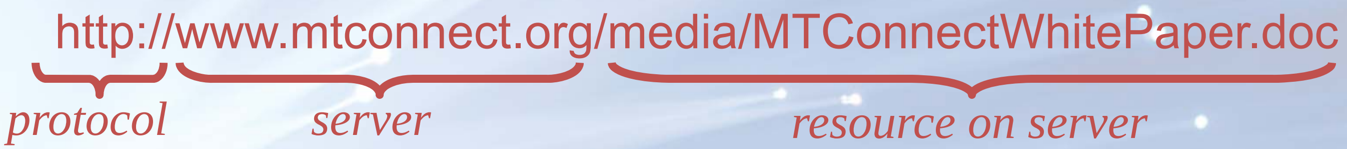
- Think of Agent as a *bucket* that controller deposits data into
- Each application “reaches into the bucket” at the rate it wants
- As bucket fills up, older measurements “leak out” the bottom
  - whether or not anyone has seen them!
- Capacity of bucket depends on Agent implementation

How/Where Does Client Find Data?

- How does Client know where to find the Agent?
  - LDAP (a/k/a Active Directory) discovery
- How does Client know *what data* a particular machine can report, in what units, with what sampling frequency...?
  - *probe* command reports this
- How does client specify *how to collect* data (sample rate, what subset of measurements, how much to gather...)?
  - *sample* and *current* commands allow this

Putting Commands into URI’s

- URI: Universal Resource Identifier
  - a string that follows particular formatting rules, used to name a *resource* (file, image, data set, etc.) that can be served by an HTTP *server*
  - in most people’s experience, it “names” a page or other resource on Web



http://www.mtconnect.org/media/MTConnectWhitePaper.doc

- Think of machine data as “resources on the Web”
  - need to know “verb” (what to do) + “nouns” (what data to operate on)
  - Example:

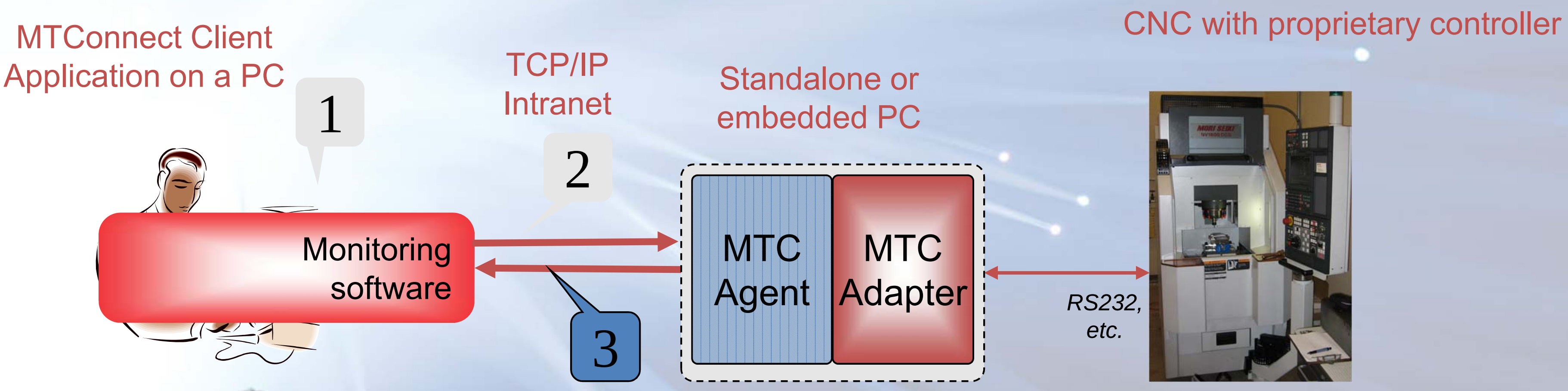
```
http://agent.mtconnect.org/sample?start=1&count=1000&path=//Ax
   es/Linear[@name='X']
```

What to do

Arguments

How MTConnect Uses These Technologies

1. Request to an MTConnect Agent is encoded in URI, *using XPath to indicate what data (measurements) are desired.*
2. URI transmitted to the Agent as HTTP request
3. Response (data) is returned as *XML document;* additional XPath queries used to extract data of interest


MTConnect Client Application on a PC
1
Monitoring software
TCP/IP Intranet
2
3
Standalone or embedded PC
MTC Agent
MTC Adapter
RS232, etc.
CNC with proprietary controller

MTConnect Clients

- Any Web browser* can be an MTConnect client!
  - Properly construct the correct URI by hand
  - Type it into address bar of your browser, preceded by server name of MTC Agent

  * Newer browsers display XML properly. Some old ones don't.
- Very easy to write Web browser based clients
  - The protocol and data representation are already "baked in" to most Web programming frameworks
- What if client app is *not* Web based?
  - *Client libraries* (MTConnect "SDK") allow simple apps to largely avoid dealing with HTTP, XML or XPath – Hello World Workshop will discuss this
  - Ironically, easy to do because all major languages & frameworks *already support* HTTP, XML, ...

MTConnect and XML

- A shop floor environment is naturally hierarchical
  - Floor→device→axes→linear:X
  - Recall: XPath is a way of referencing particular XML document elements meeting some specified criteria
- The MTConnect schema defines a specific hierarchy of XML *elements*
- Each element has certain defined *attributes*
- Some elements are *required* (must be present in XML representation) for MTConnect compliance
- Most elements are *optional*

4 kinds of XML Responses in MTConnect

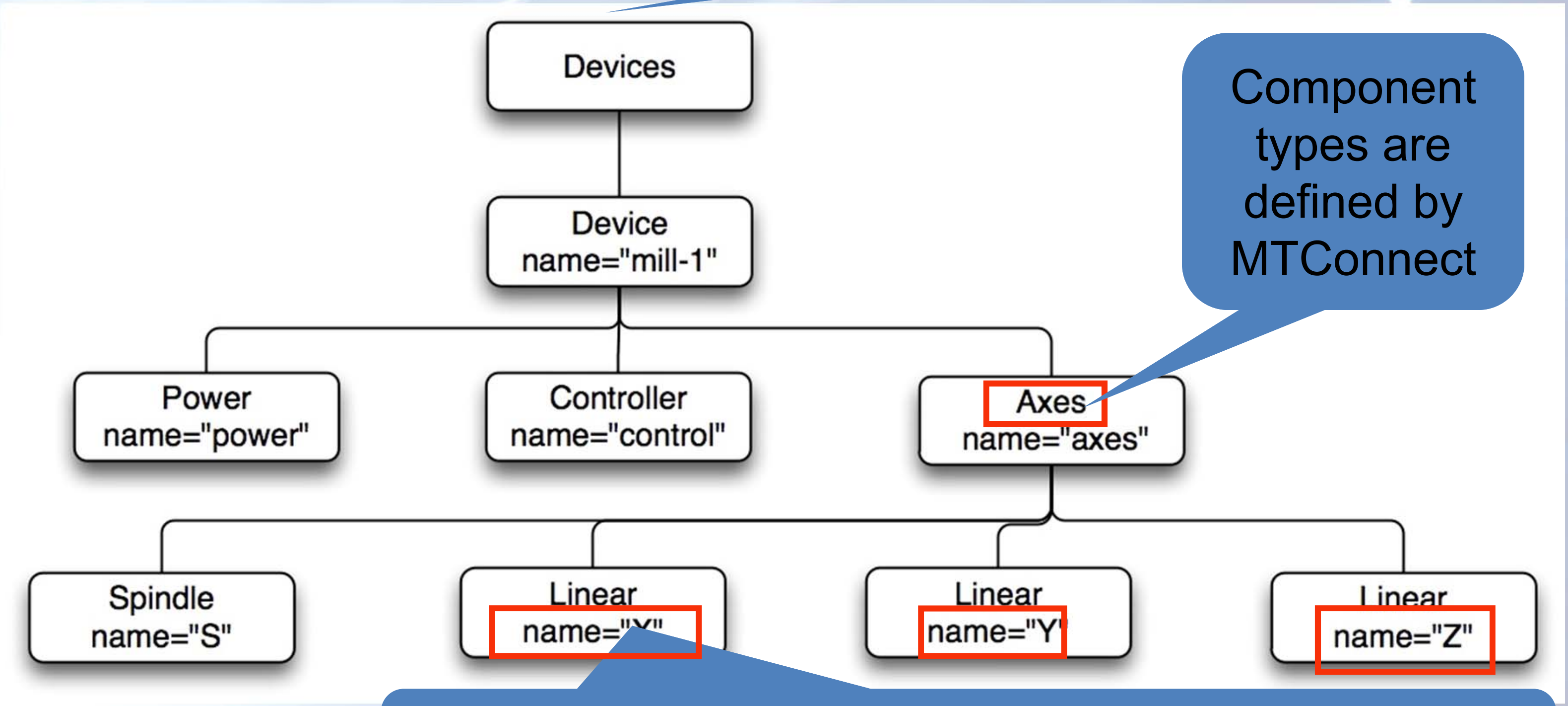
- **Devices**
  - descriptive information about the configuration of the machine(s) and what data can be delivered
  - returned by *probe* command issued to Agent
- **Streams**
  - Data samples and events from the device(s)
  - returned by *sample* or *current* command issued to Agent
- **Assets**
  - Retrieve information on mobile assets
- **Error**
  - Returned when an error occurs that prevents further processing
  - Caveat: most things that don't work as you expect aren't necessarily errors

The Device

- An Agent represents one or more *devices*
- A device (“machine”) is a collection of *components*
  - A component can have sub-components, allowing representation of arbitrarily complex devices
  - In fact, the Device itself is one type of Component
- There’s a fixed set of Component types
  - Special components can be used to extend the specification; we won’t discuss these further today
- Each component also has a *name,* and possibly some other attributes

Example: Simple 3-axis mill

There’s always a *Devices* node—even if only a single *Device* inside of it



Component types are defined by MTConnect

Siblings of the same type can be distinguished by their names (and, as we’ll see, their *id*)

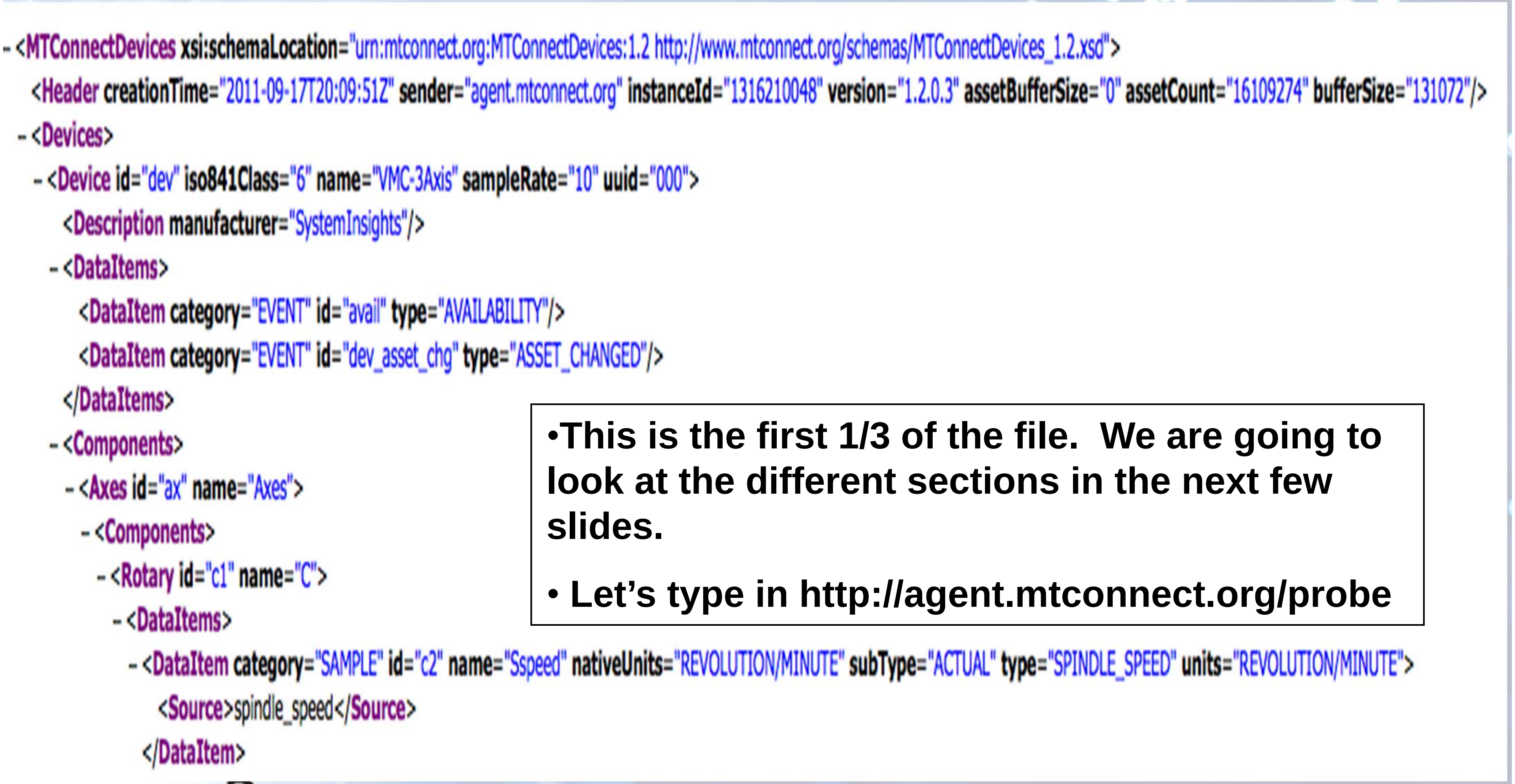
Try It: Issue a *probe* command

- The MTC-TAG runs a publicly-accessible Agent for developer testing
  - Agent represents a (simulated) vertical CNC mill
  - uses open-source, Linux-based *EMC2* machine controller from LinuxCNC.org
  - Always running at `http://agent.mtconnect.org/`

Let's do probe now (type in below):

`http://agent.mtconnect.org/probe`

Let’s Look at the XML

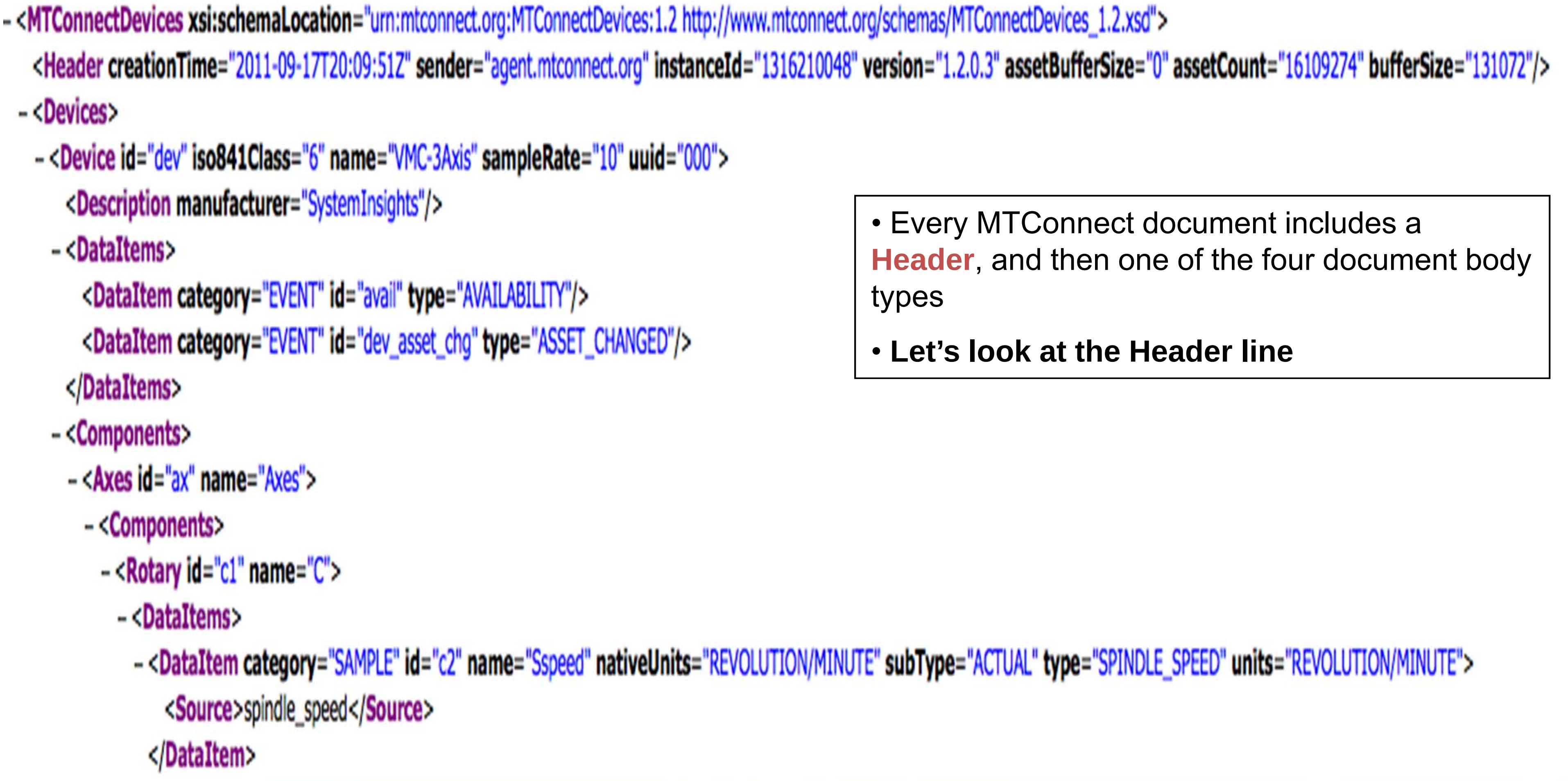


```
- <MTConnectDevices xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.2 http://www.mtconnect.org/schemas/MTConnectDevices_1.2.xsd">
    <Header creationTime="2011-09-17T20:09:51Z" sender="agent.mtconnect.org" instanceId="1316210048" version="1.2.0.3" assetBufferSize="0" assetCount="16109274" bufferSize="131072"/>
  - <Devices>
    - <Device id="dev" iso841Class="6" name="VMC-3Axis" sampleRate="10" uuid="000">
        <Description manufacturer="SystemInsights"/>
      - <DataItems>
          <DataItem category="EVENT" id="avail" type="AVAILABILITY"/>
          <DataItem category="EVENT" id="dev_asset_chg" type="ASSET_CHANGED"/>
        </DataItems>
      - <Components>
        - <Axes id="ax" name="Axes">
          - <Components>
            - <Rotary id="c1" name="C">
              - <DataItems>
                - <DataItem category="SAMPLE" id="c2" name="Sspeed" nativeUnits="REVOLUTION/MINUTE" subType="ACTUAL" type="SPINDLE_SPEED" units="REVOLUTION/MINUTE">
                    <Source>spindle_speed</Source>
                  </DataItem>
```

•This is the first 1/3 of the file. We are going to look at the different sections in the next few slides.

• Let’s type in http://agent.mtconnect.org/probe

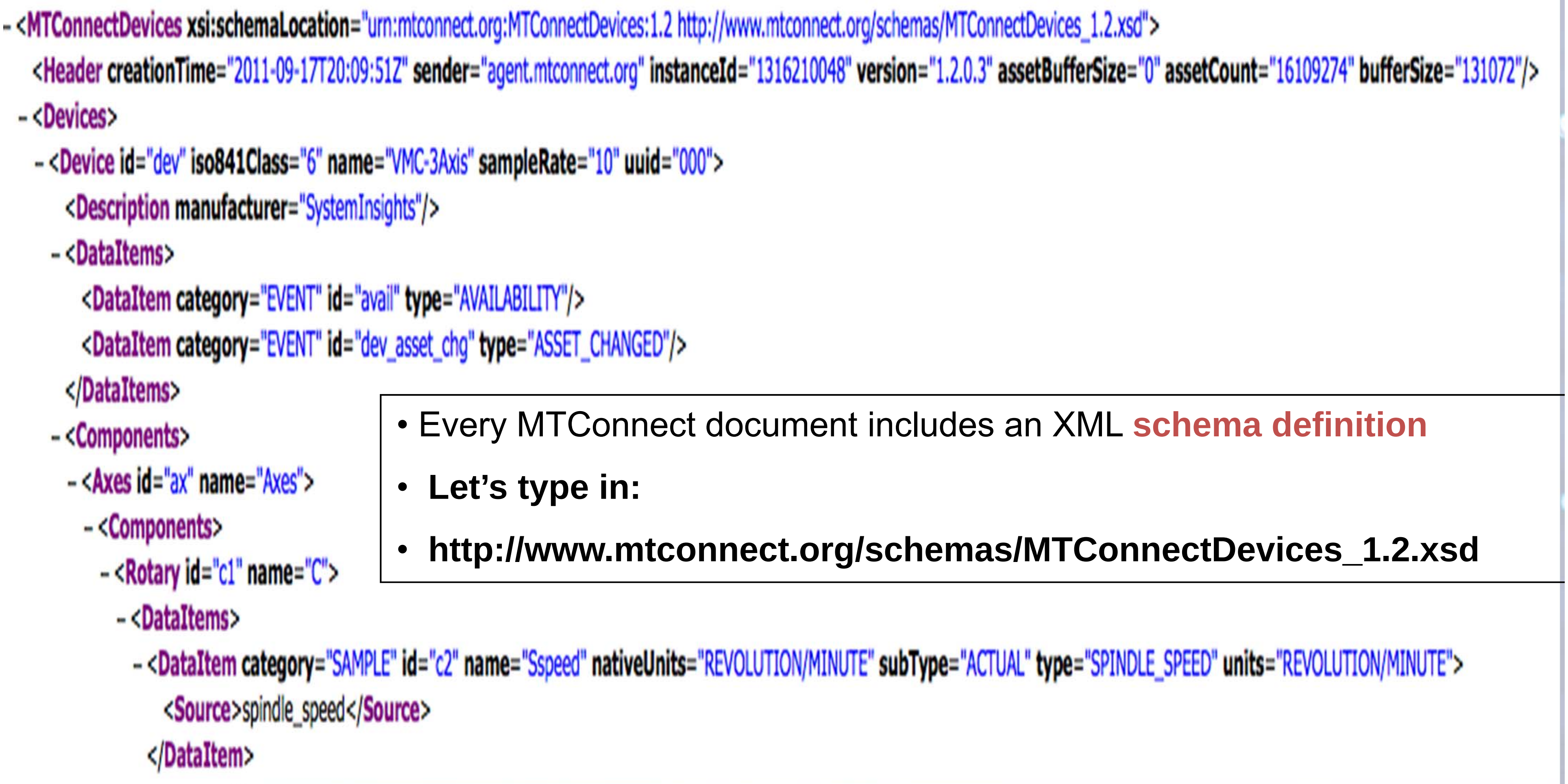
MTConnect Header



```
- <MTConnectDevices xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.2 http://www.mtconnect.org/schemas/MTConnectDevices_1.2.xsd">
   <Header creationTime="2011-09-17T20:09:51Z" sender="agent.mtconnect.org" instanceId="1316210048" version="1.2.0.3" assetBufferSize="0" assetCount="16109274" bufferSize="131072"/>
  - <Devices>
    - <Device id="dev" iso841Class="6" name="VMC-3Axis" sampleRate="10" uuid="000">
        <Description manufacturer="SystemInsights"/>
      - <DataItems>
          <DataItem category="EVENT" id="avail" type="AVAILABILITY"/>
          <DataItem category="EVENT" id="dev_asset_chg" type="ASSET_CHANGED"/>
        </DataItems>
      - <Components>
        - <Axes id="ax" name="Axes">
          - <Components>
            - <Rotary id="c1" name="C">
              - <DataItems>
                - <DataItem category="SAMPLE" id="c2" name="Sspeed" nativeUnits="REVOLUTION/MINUTE" subType="ACTUAL" type="SPINDLE_SPEED" units="REVOLUTION/MINUTE">
                    <Source>spindle_speed</Source>
                  </DataItem>
```

- Every MTConnect document includes a **Header**, and then one of the four document body types
- **Let’s look at the Header line**

XML Schema Definition



```
- <MTConnectDevices xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.2 http://www.mtconnect.org/schemas/MTConnectDevices_1.2.xsd">
   <Header creationTime="2011-09-17T20:09:51Z" sender="agent.mtconnect.org" instanceId="1316210048" version="1.2.0.3" assetBufferSize="0" assetCount="16109274" bufferSize="131072"/>
 - <Devices>
   - <Device id="dev" iso841Class="6" name="VMC-3Axis" sampleRate="10" uuid="000">
       <Description manufacturer="SystemInsights"/>
     - <DataItems>
         <DataItem category="EVENT" id="avail" type="AVAILABILITY"/>
         <DataItem category="EVENT" id="dev_asset_chg" type="ASSET_CHANGED"/>
       </DataItems>
     - <Components>
       - <Axes id="ax" name="Axes">
         - <Components>
           - <Rotary id="c1" name="C">
             - <DataItems>
               - <DataItem category="SAMPLE" id="c2" name="Sspeed" nativeUnits="REVOLUTION/MINUTE" subType="ACTUAL" type="SPINDLE_SPEED" units="REVOLUTION/MINUTE">
                   <Source>spindle_speed</Source>
                 </DataItem>
```

- Every MTConnect document includes an XML schema definition
- **Let’s type in:**
- **http://www.mtconnect.org/schemas/MTConnectDevices_1.2.xsd**

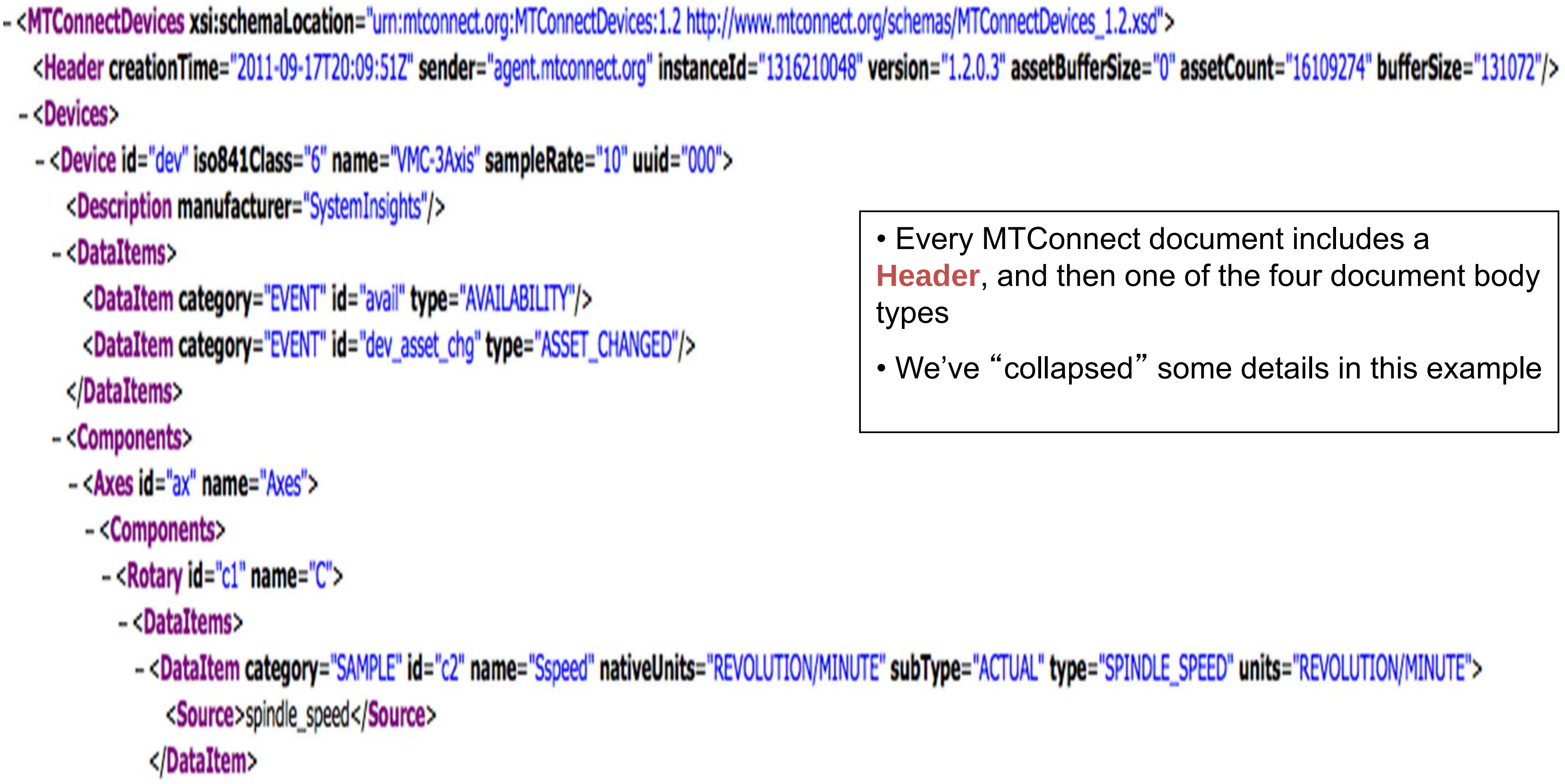
MTConnectDevices_1.2.xsd

```
<xs:element name='MTConnectStreams' type='MTConnectStreamsType'>
    <xs:annotation>
      <xs:documentation>
        The root node for MTConnect
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:simpleType name='SenderType'>
    <xs:annotation>
      <xs:documentation>


The date and time the document was created
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base='xs:dateTime'/>
  </xs:simpleType>
  <xs:simpleType name='SequenceType'>
    <xs:annotation>
      <xs:documentation>
        A sequence number
      </xs:documentation>
    </xs:annotation>
     <xs:restriction base='xs:integer'>
       <xs:minInclusive value='1'/>
       <xs:maxExclusive value='18446744073709551615'/>
```

Notice that we are defining the MTConnect sequence number, which is an integer as well as what the minimum sequence maximum numbers can be.

Components and Data Items



```
- <MTConnectDevices xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.2 http://www.mtconnect.org/schemas/MTConnectDevices_1.2.xsd">
    <Header creationTime="2011-09-17T20:09:51Z" sender="agent.mtconnect.org" instanceId="1316210048" version="1.2.0.3" assetBufferSize="0" assetCount="16109274" bufferSize="131072"/>
  - <Devices>
    - <Device id="dev" iso841Class="6" name="VMC-3Axis" sampleRate="10" uuid="000">
        <Description manufacturer="SystemInsights"/>
      - <DataItems>
          <DataItem category="EVENT" id="avail" type="AVAILABILITY"/>
          <DataItem category="EVENT" id="dev_asset_chg" type="ASSET_CHANGED"/>
        </DataItems>
      - <Components>
        - <Axes id="ax" name="Axes">
          - <Components>
            - <Rotary id="c1" name="C">
              - <DataItems>
                - <DataItem category="SAMPLE" id="c2" name="Sspeed" nativeUnits="REVOLUTION/MINUTE" subType="ACTUAL" type="SPINDLE_SPEED" units="REVOLUTION/MINUTE">
                    <Source>spindle_speed</Source>
                  </DataItem>
```

- Every MTConnect document includes a **Header**, and then one of the four document body types
- We've "collapsed" some details in this example

Anatomy of a Component

- Attributes
  - uuid - A globally-unique id for the component (required for a Device; optional for other component types)
  - **name** - The name of the component
  - sampleRate - An optional sample rate
- Description
  - Manufacturer, serial number, free-form descriptive text
- DataItems
  - *What can be reported* by this component (we' ll describe shortly)
- Components: subcomponents (recursively defined!)
  - A component may have *its own* data items in addition to those of its sub-Components
  - example: PATH_FEEDRATE for Axes component, *in addition* to POSITION or SPINDLE_SPEED for each axis

What is a DataItem?

- A *DataItem* describes *something that a Component can report.*
- A DataItem has:
  - a TYPE e.g. POSITION, PATH_FEEDRATE
  - sometimes, a SUBTYPE e.g. ACTUAL, COMMANDED
  - UNITS (if numeric), e.g. MILLIMETERS, REVOLUTIONS_PER_MINUTE
  - a CATEGORY, describing what “kind” of measurement data this is: EVENT or SAMPLE
- These *enumerations* are part of the MTConnect Specification—and defined in terms of XML Schema

Data Items and their Types

- The type of a specifies what (kinds of) values its corresponding Samples or Events may take
  - POSITION is a numerical value
  - EXECUTION is one of IDLE, PAUSED, EXECUTING
  - DIRECTION is one of CLOCKWISE or COUNTER_CLOCKWISE
- etc...
- legal units are codified in the MTConnect Specification

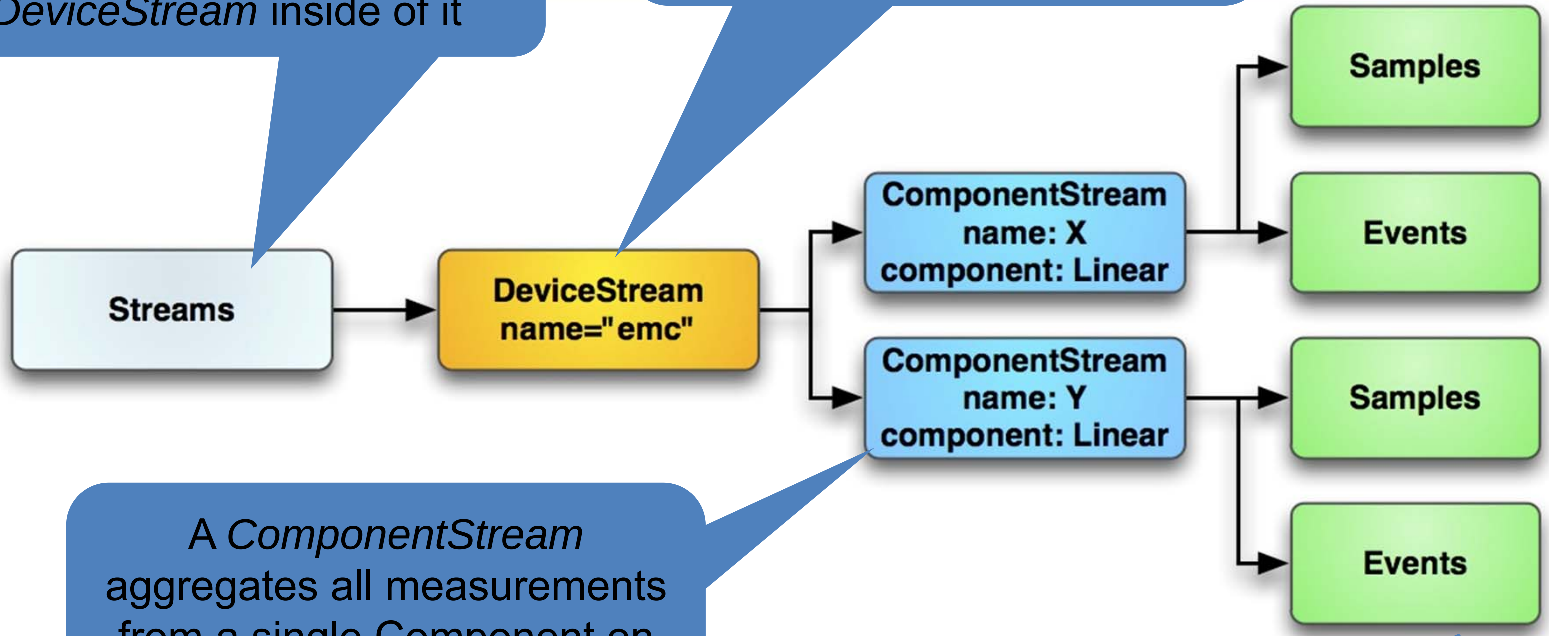
The Actual Data: Streams

- What is a *stream?*
  - A collection of samples and events
  - organized hierarchically by device & component
- Rationale: more efficient data delivery & parsing
  - moves *common information* to a higher level in the XML document
  - e.g., if all samples returned are for components with a common parent
- Provides simple structure for XML parsing (data is easily aggregated)

Stream Structure

There’ s always a *Streams* node—even if only a single *DeviceStream* inside of it

A *DeviceStream* aggregates all measurements from a single Device represented by this Agent


Streams
DeviceStream name="emc"
ComponentStream name: X component: Linear
ComponentStream name: Y component: Linear
Samples
Events
Samples
Events

A *ComponentStream* aggregates all measurements from a single Component on this Device

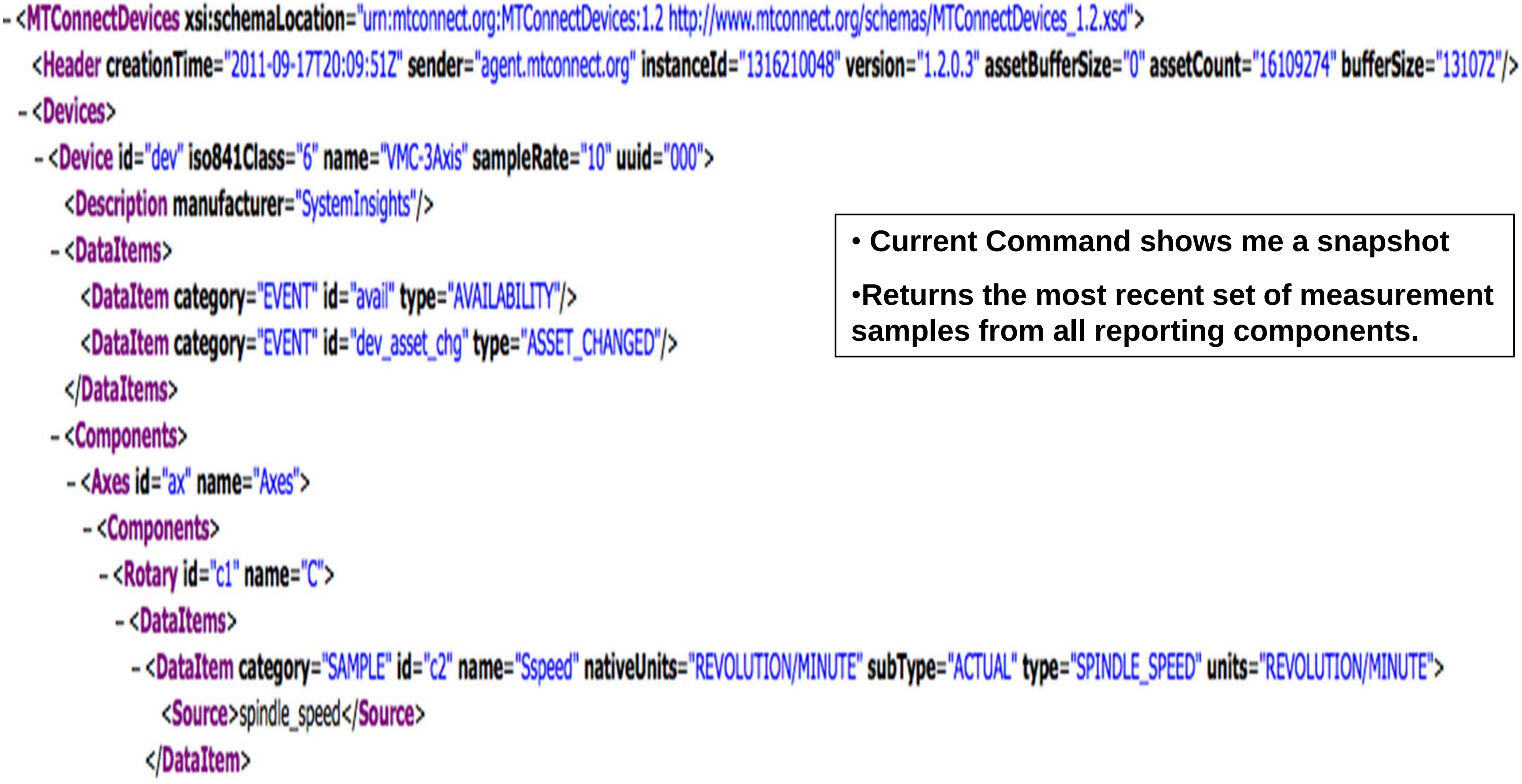
Each *ComponentStream* can include one or both of *Samples* or *Events* (recall the difference...)

How do you get Streams?

- The *probe* command returns *Devices* (metadata)
- The *current* and *sample* commands return *Streams* (data)
- There are two variants of *current* and one variant of *sample*
- Let' s try the simpler *current* variant now:

  `http://agent.mtconnect.org/current`

Current Command



```
- <MTConnectDevices xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.2 http://www.mtconnect.org/schemas/MTConnectDevices_1.2.xsd">
   <Header creationTime="2011-09-17T20:09:51Z" sender="agent.mtconnect.org" instanceId="1316210048" version="1.2.0.3" assetBufferSize="0" assetCount="16109274" bufferSize="131072"/>
 - <Devices>
   - <Device id="dev" iso841Class="6" name="VMC-3Axis" sampleRate="10" uuid="000">
       <Description manufacturer="SystemInsights"/>
     - <DataItems>
         <DataItem category="EVENT" id="avail" type="AVAILABILITY"/>
         <DataItem category="EVENT" id="dev_asset_chg" type="ASSET_CHANGED"/>
       </DataItems>
     - <Components>
       - <Axes id="ax" name="Axes">
         - <Components>
           - <Rotary id="c1" name="C">
             - <DataItems>
               - <DataItem category="SAMPLE" id="c2" name="Sspeed" nativeUnits="REVOLUTION/MINUTE" subType="ACTUAL" type="SPINDLE_SPEED" units="REVOLUTION/MINUTE">
                   <Source>spindle_speed</Source>
                 </DataItem>
```

- Current Command shows me a snapshot

•Returns the most recent set of measurement samples from all reporting components.

Probe and Current Summary

- *probe* returns metadata
  - a *Devices* document describing Devices, Components, and their DataItems
  - describes *which* data can be reported and *how* to interpret it
- *current* returns data
  - actual values of “most recently” observed measurements
  - *not all DataItems may report a measurement*

Sampling and XPath

- Remember: the path parameter in the command refers to the *component* structure (not the stream structure)
- XPath allows arbitrarily complex expressions

Representing a
Desired Data Item In XPath

- //Linear
  - Retrieves all the Linear axis data
- //Linear[@name=“X”]
  - Retrieves all the X axis data
- //Spindle//DataItem[@type=“SPINDLE_SPEED”]
  - Retrieves the spindle speed
- //Controller//DataItem[@type="LINE"]|//DataItem[@type="EXECUTION"]

  - Retrieves the Line and Execution state for the controller

XPath: Specifying Descendants

- A *descendant* occurs *anywhere below* some node in an XML document
  - person and institution are both descendants of instructors
  - institution is a descendant of both person and instructors
  - every child of *x* is a descendant of *x,* but not vice versa
- XPath lets you use // to specify *descendants*

  `/instructors//institution`

  `//institution` (any descendant of the root node)

  `//institution[@type="public"]`

```
<instructors>
  <person name=”Joel">
     <company>
       ITAMCO
     <company/>
     <windows_user/>
  </person>
  <person name=”Dave"
          title="lecturer">
     <institution type=”Non-
Profit">
       MTConnect_Institute
     </institution>
  </person>
</instructors>
```

Embedding the XPath into a URI

- Simple example:

```
http://agent.mtconnect.org/
  /sample?path=//Linear[@name='X']
```

- General HTTP convention: to pass “arguments” in a URL, add a question mark to end, then `arg1=val1&arg2=val2&`....
- Some browsers may require you to “escape” certain characters:

```
  http://agent.mtconnect.org/sample?path=//L
  inear[@name=%22X%22]
```

XML Document Parsing - SAX vs DOM

SAX	DOM
Both SAX and DOM are used to parse the XML document. Both has advantages and disadvantages and can be used in our programming depending on the situation.	
Parses node by node	Stores the entire XML document into memory before processing
Doesn't store the XML in memory	Occupies more memory
We cant insert or delete a node	We can insert or delete nodes
Top to bottom traversing	Traverse in any direction.
SAX is an event based parser	DOM is a tree model parser
SAX is a Simple API for XML	Document Object Model (DOM) API

import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
	import javax.xml.parsers.*;
	import org.w3c.dom.*;

doesn't preserve comments	preserves comments
SAX generally runs a little faster than DOM	SAX generally runs a little faster than DOM
If we need to find a node and doesn't need to insert or delete we can go with SAX itself otherwise DOM provided we have more memory.	

Let’s Try A Few

- Only the Spindle
- Any axes
- Only Controller & Power information
- Only things named "Z"
- Actual values only (not commanded) for Linear axes
- Question: are attribute names and component names case sensitive?

Let’s Try Some

- How would we display the Spindle?
- http://agent.mtconnect.org/current?path=//Axes//Rotary
- How do we display all the Axes?
- http://agent.mtconnect.org/current?path=//Axes
- How would we display Controller and Power Information?
- http://agent.mtconnect.org/current?path=//Controller|//Power

Let’s Try a Few More

- How would we only display the Z axis?
- http://agent.mtconnect.org/current?path=//Axes//Linear[@name='Z']
- How would we display Actual values only (not commanded) for Linear Axes?
- http://agent.mtconnect.org/current?path=//Axes//Linear//DataItem[@subType='ACTUAL']
- Question: are attribute names and component names case sensitive? *Yes!*

Assets in MTConnect 1.2

- Very important enabling addition to MTConnect
- An Asset is something that is associated with the manufacturing process that is NOT a component of the device AND can be removed.
- Concrete Examples of Assets:
  - Cutting Tools, Workholding Systems, and Fixtures.
  - ISO 13399 – standard for describing product data regarding cutting tools, independent from any particular system
- Assets is in Part 4 of the MTConnect standard and will concern itself with the modeling of these assets and the management and communication of asset data using MTConnect.

Assets

- Request for all the assets in the *Agent:*
- *url: http://agent.mtconnect.org/assets*
- *Returns all available MTConnect assets in the Agent. MTConnect **MAY return a limited set if there are too many asset records. The assets MUST be added to the beginning with the most recently modified assets.***

XPath Summary

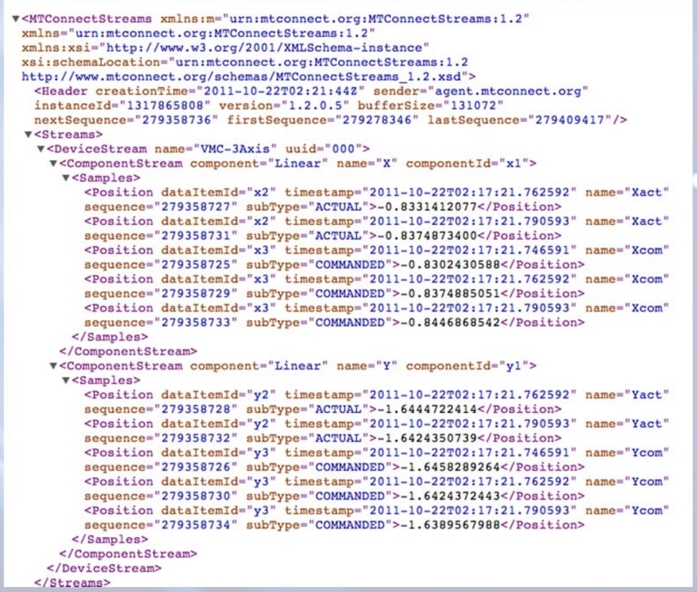
- MTConnect environment represented as a *hierarchy*
  - DataItems describe what can be reported
  - Samples/Events are the actual reported data
  - *Document* (unit of exchange) wraps these
- XPath syntax can be used to identify which *elements* in the hierarchy are of interest
- “Bucket model” decouples machines from clients
  - Device reports new data to Agent (“fill the bucket”)
  - Apps request data at frequency & granularity desired (“draw samples from the bucket”)

The `sample` Command

- Retrieves a series of data starting from a position and return up to the requested number of samples or events
- Allows application to retrieve all data without missing anything
- Can stream data as it arrives
- Can be thought of as a window into the stream of data

The Sample Command

- The Request
- http://agent.mtconnect.org/sample?count=1000
- Agent Responds



```
▼<MTConnectStreams xmlns:m="urn:mtconnect.org:MTConnectStreams:1.2"
 xmlns="urn:mtconnect.org:MTConnectStreams:1.2"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:mtconnect.org:MTConnectStreams:1.2
 http://www.mtconnect.org/schemas/MTConnectStreams_1.2.xsd">
   <Header creationTime="2011-10-22T02:21:44Z" sender="agent.mtconnect.org"
   instanceId="1317865808" version="1.2.0.5" bufferSize="131072"
   nextSequence="279358736" firstSequence="279278346" lastSequence="279409417"/>
 ▼<Streams>
   ▼<DeviceStream name="VMC-3Axis" uuid="000">
     ▼<ComponentStream component="Linear" name="X" componentId="x1">
       ▼<Samples>
          <Position dataItemId="x2" timestamp="2011-10-22T02:17:21.762592" name="Xact"
          sequence="279358727" subType="ACTUAL">-0.8331412077</Position>
          <Position dataItemId="x2" timestamp="2011-10-22T02:17:21.790593" name="Xact"
          sequence="279358731" subType="ACTUAL">-0.8374873400</Position>
          <Position dataItemId="x3" timestamp="2011-10-22T02:17:21.746591" name="Xcom"
          sequence="279358725" subType="COMMANDED">-0.8302430588</Position>
          <Position dataItemId="x3" timestamp="2011-10-22T02:17:21.762592" name="Xcom"
          sequence="279358729" subType="COMMANDED">-0.8374885051</Position>
          <Position dataItemId="x3" timestamp="2011-10-22T02:17:21.790593" name="Xcom"
          sequence="279358733" subType="COMMANDED">-0.8446868542</Position>
        </Samples>
      </ComponentStream>
     ▼<ComponentStream component="Linear" name="Y" componentId="y1">
       ▼<Samples>
          <Position dataItemId="y2" timestamp="2011-10-22T02:17:21.762592" name="Yact"
          sequence="279358728" subType="ACTUAL">-1.6444722414</Position>
          <Position dataItemId="y2" timestamp="2011-10-22T02:17:21.790593" name="Yact"
          sequence="279358732" subType="ACTUAL">-1.6424350739</Position>
          <Position dataItemId="y3" timestamp="2011-10-22T02:17:21.746591" name="Ycom"
          sequence="279358726" subType="COMMANDED">-1.6458289264</Position>
          <Position dataItemId="y3" timestamp="2011-10-22T02:17:21.762592" name="Ycom"
          sequence="279358730" subType="COMMANDED">-1.6424372443</Position>
          <Position dataItemId="y3" timestamp="2011-10-22T02:17:21.790593" name="Ycom"
          sequence="279358734" subType="COMMANDED">-1.6389567988</Position>
        </Samples>
      </ComponentStream>
    </DeviceStream>
  </Streams>
```

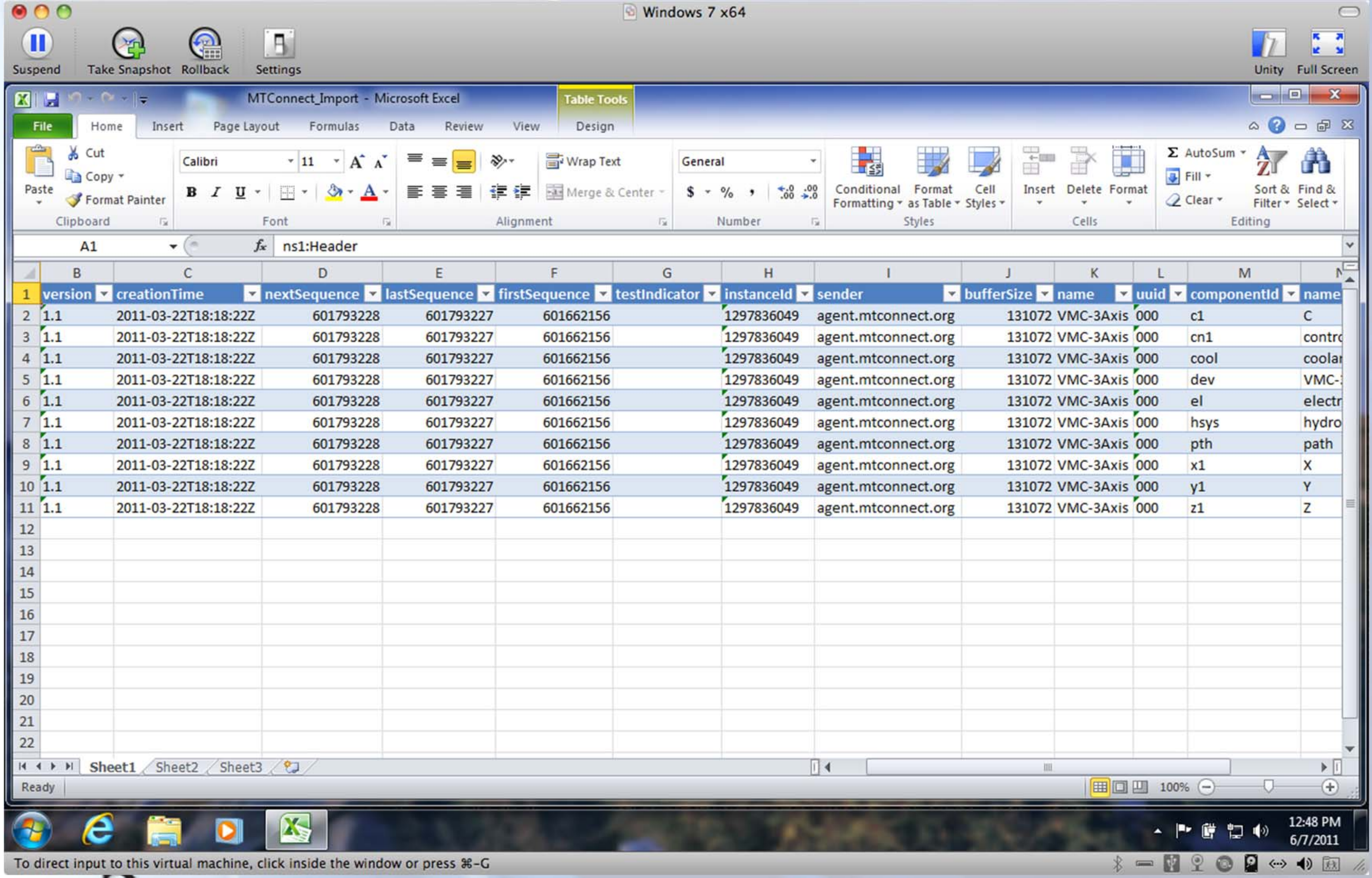
Probe, Current, Sample, Asset Summary

- Probe reports metadata: what is reportable, units, etc.
- Current takes a “snapshot” of the data items for which the agent has data
  - possibly constrained matching some specified Xpath
  - Result will include one value for each sample or event
- Sample returns a bundle of measurements given a sequence number and count
- Current snapshots can also be *streamed* (continuously delivered at some coarse-grained frequency)
- An Asset is something that is associated with the manufacturing process that is not a component of a device

Easy Excel With MTConnect

- Start Excel
- Go into the Data selection on the upper tab
- Select From Web
- New Web Query will come up
- In the browser bar that pops up, put in your MTConnect http url (for example):
  - http://agent.mtconnect.org/sample?count=1000
- Hit Import, you will hit import again, then select import into existing worksheet.
- You can go to:
  - http://tinyurl.com/ExcelMTConnect
  - to see this demonstrated

Easy Excel and MTConnect



A1 ns1:Header

1	version	creationTime	nextSequence	lastSequence	firstSequence	testIndicator	instanceId	sender	bufferSize	name	uuid	componentId	name
2	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	c1	C
3	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	cn1	contro
4	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	cool	coolar
5	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	dev	VMC-3
6	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	el	electr
7	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	hsys	hydro
8	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	pth	path
9	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	x1	X
10	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	y1	Y
11	1.1	2011-03-22T18:18:22Z	601793228	601793227	601662156		1297836049	agent.mtconnect.org	131072	VMC-3Axis	000	z1	Z

ITAMCO MTConnect App

For the demo please browse to:
http://www.itamco.com/iphone4/iphone.asp

Where/How To Keep Learning About MTConnect

- Attend the Agent/Adapter Workshop Nat Frampton is teaching
- Attend the MTConnect Hello World First Application Chris Tacke is teaching
- Having a very good understanding of XML is important - W3 Schools have very good online tutorials
- Download MTConnect at https://github.com/mtconnect
- Download Joel's MTConnect app and try it
- Go back through the labs and try your own queries

Summary of What We Covered

- Goals and Non Goals of this workshop
- HTTP and XML
- Big Picture of MTConnect
- MTConnect Agent
- Schema
- Probe, Sample, Current and Asset Commands
- Documents versus Streaming
- XML namespaces and path
- Demo of Simple Web Client
- Using Excel for MTConnect
- Demo of ITAMCO's Mobile App
- Where and How To Keep Learning About MTConnect
- Summary

THANKS!

MTConnect 101

Fundamentals of MTConnect

Joel Neidig of ITAMCO

Dave Edstrom of MTConnect Institute