2010 XNAT Workshop: Day 2

9:00 XNAT Inside the Box

9:30 Customizations

11:00 Tuning, Optimization, & Monitoring

1:00 Pipelines

2:30 Harvard Customizations & Pipelines

3:00 Breakouts

XNAT

XNAT: Inside the Box

What makes XNAT tick?

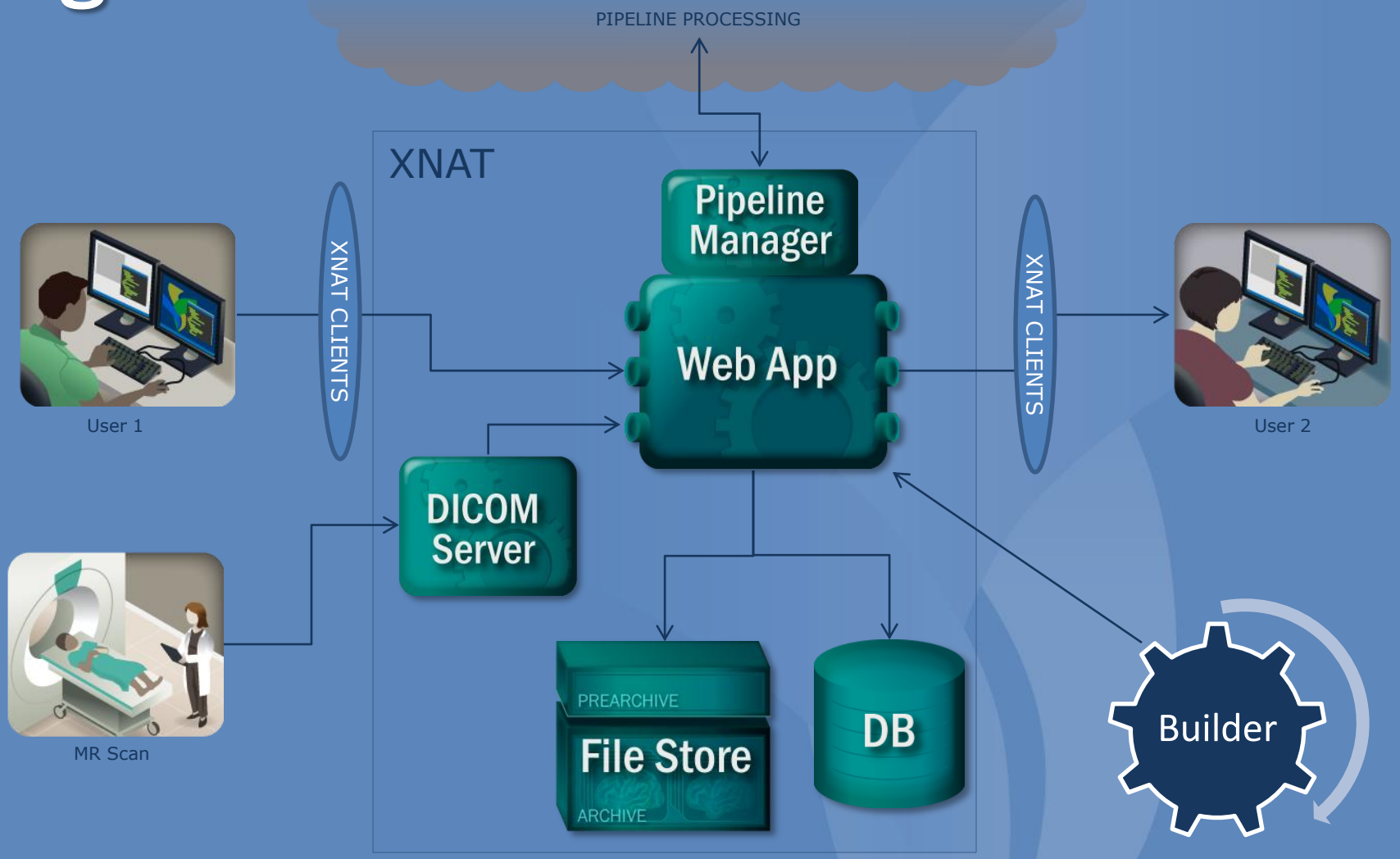
Tim Olsen

XNAT

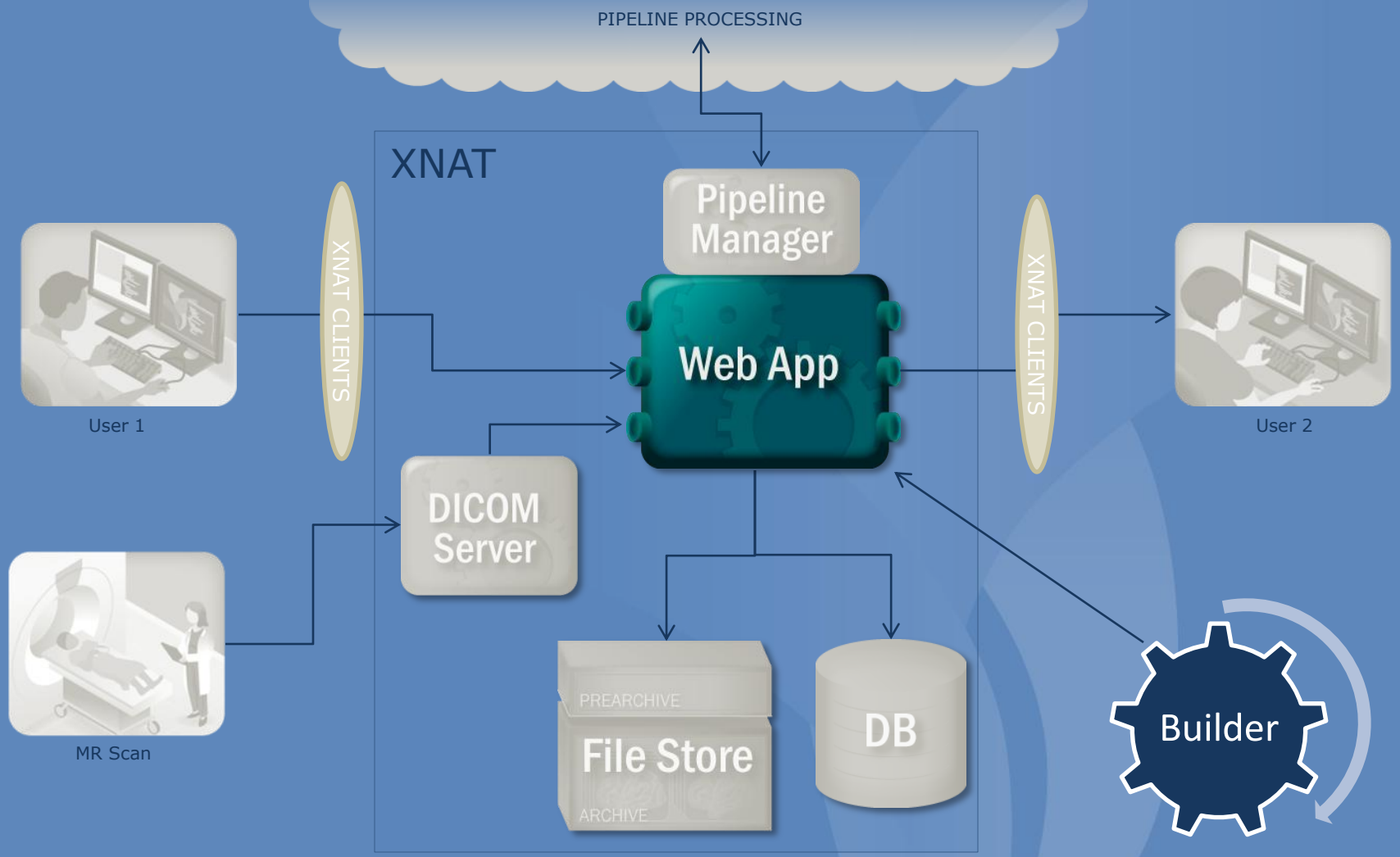
Goals

- Provide introduction to some of the major tools/components of XNAT
- Provide insight into the history of XNAT development

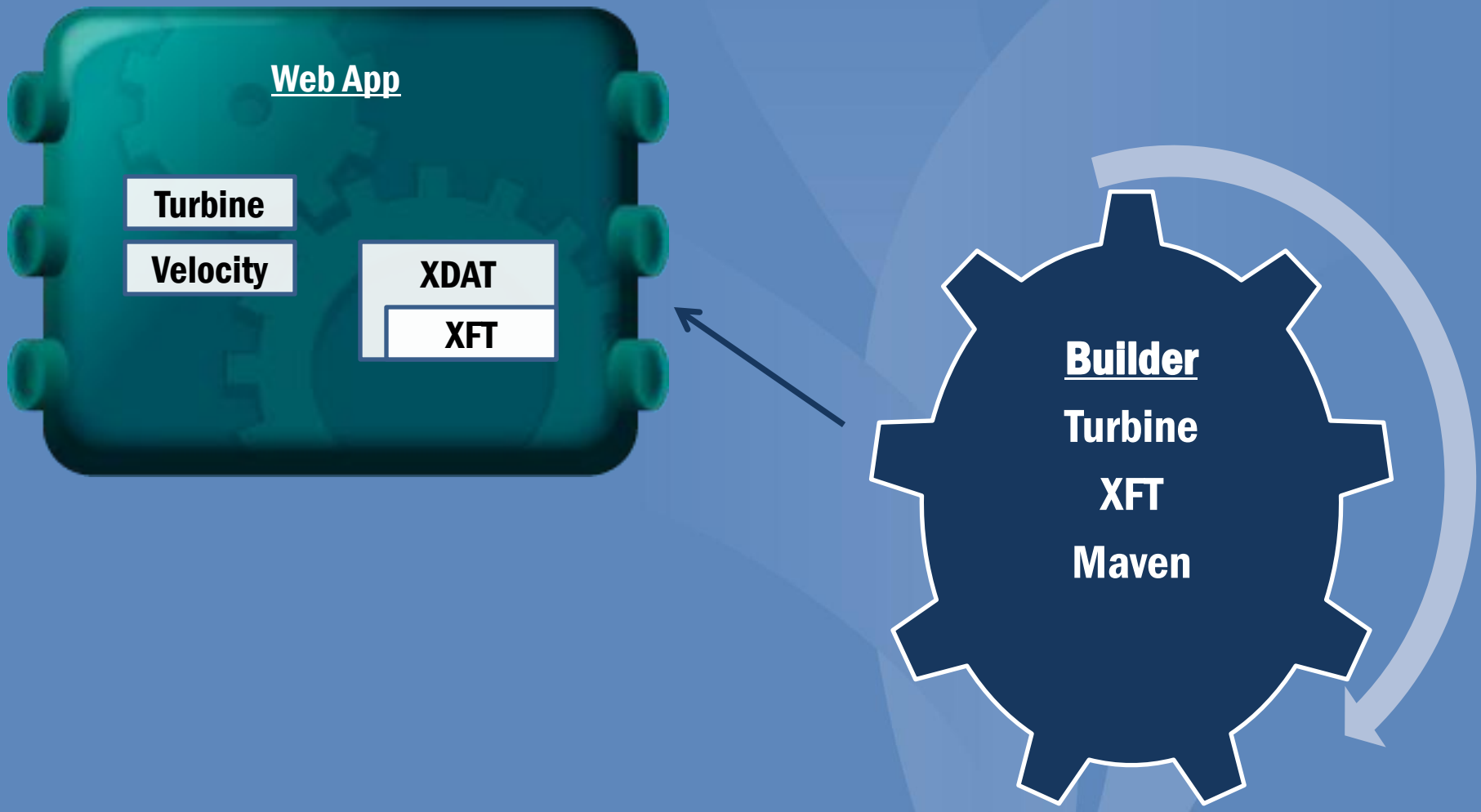
Big Picture


PIPELINE PROCESSING
XNAT
Pipeline Manager
Web App
DICOM Server
PREARCHIVE
File Store
ARCHIVE
DB
XNAT CLIENTS
XNAT CLIENTS
User 1
User 2
MR Scan
Builder

XNAT: Inside the Box



XNAT: Inside the Box



XNAT

Plan

- Turbine
- Velocity
- Display Documents
- XFT
- XDAT
- Maven

In the beginning…

… there was the CNDA.

- Built using Turbine 2.2
- Velocity
- Torque
- Simple (Torque compliant) CNDA schema

XNAT

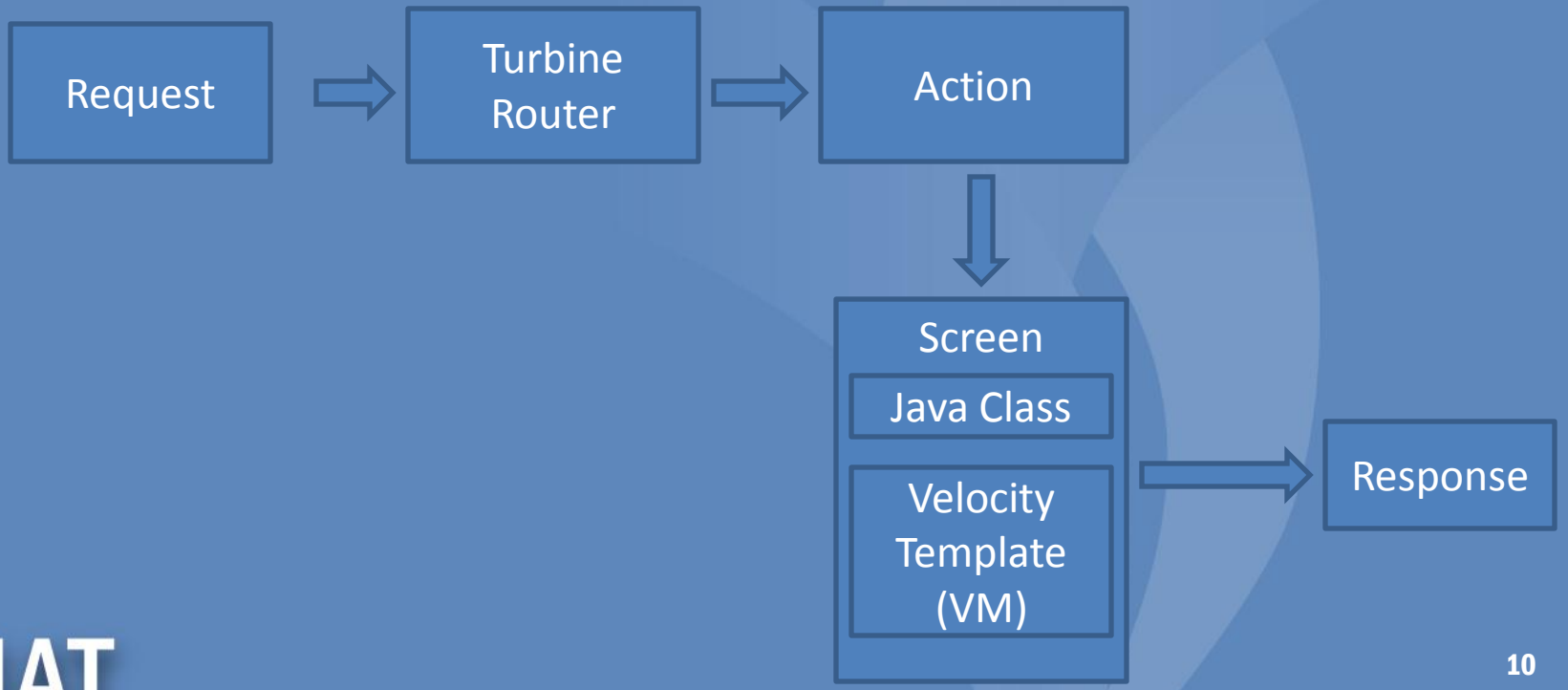
Apache Turbine

- Platform for building applications (not just running them)
- MVC Pattern (Model-View-Controller)
- Similar to Struts
- Native support for Velocity (and JSP)
- Combines a collection of frequently used APIs and manages the creation of web applications

http://turbine.apache.org/turbine/development/turbine-2.3/index.html

Apache Turbine (2)

- Uses Action classes to perform tasks
- Routes to Screens to display data



XNAT

Apache Turbine: Router

- Turbine Servlet
- URL contains the name of the action

http://localhost/xnat/app/action/DisplayItemAction

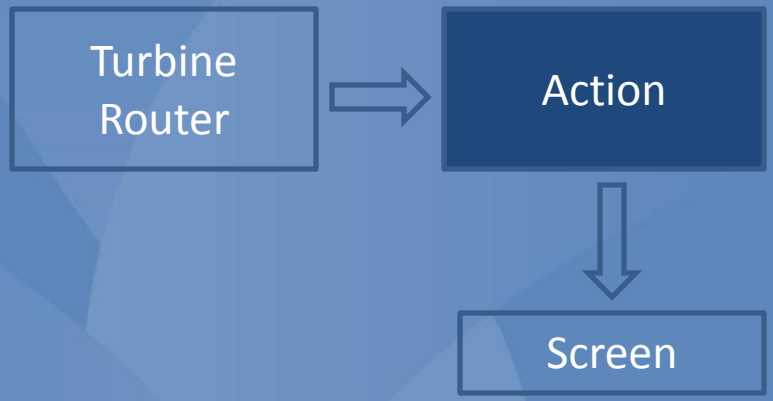
- XDAT classes support passing Items to actions by search parameters:
  - search_element=xnat:mrSessionData
  - search_field=xnat:mrSessionData/ID
  - search_value=LOCAL_E00012

http://localhost/xnat/app/action/DisplayItemAction/search_element/xnat:mrSessionData/search_field/xnat:mrSessionData.ID/search_value/LOCAL_E00012

XNAT

Apache Turbine: Action

- Used to perform tasks
  - Storage
  - Retrieval
  - Data manipulation
  - Calculations
  - Routing to screens



- Uses RunData object to handle Request level variables ($data)
  - Passed parameters
  - Screen Template destination
- Uses Context to pass parameters to Template ($context)

XNAT

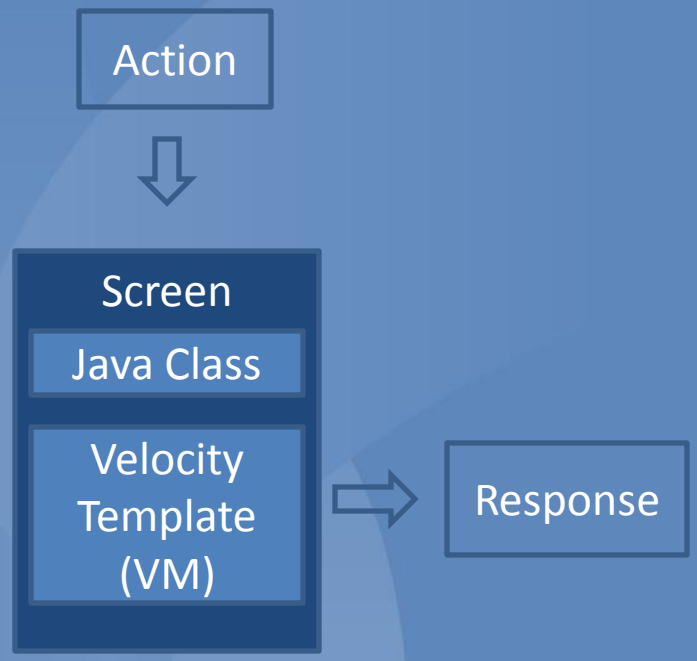
Apache Turbine: DisplayItemAction

```
public class DisplayItemAction extends SecureAction {
    static Logger logger = Logger.getLogger(DisplayItemAction.class);
    public void doPerform(RunData data, Context context) throws Exception
    {
        preserveVariables(data,context);
        try {
            logger.debug("BEGIN DisplayItemAction");
            final SchemaElementI se = TurbineUtils.GetSchemaElementBySearch(data);

            if (se != null)
            {
                final String templateName = GetReportScreen(se);
                data.setScreenTemplate(templateName);
                logger.info("Routing request to '" + templateName + "'");
            }else{
                logger.error("No Element Found. ");
                TurbineUtils.OutputPassedParameters(data,context,this.getClass().getName());
                data.setMessage("No Item Found.");
                data.setScreenTemplate("Error.vm");
            }
        } catch (Exception e) {
            logger.error("",e);
            TurbineUtils.OutputPassedParameters(data,context,this.getClass().getName());
            data.setMessage(e.getMessage());
            data.setScreenTemplate("Error.vm");
        }
        logger.debug("END DisplayItemAction");
    }

    public static String GetReportScreen(final SchemaElementI se)
    {
        String templateName = "/screens/XDATScreen_report_" + se.getSQLName() + ".vm";
        if (Velocity.templateExists(templateName))
        {
            templateName= "XDATScreen_report_" + se.getSQLName() + ".vm";
        }else
        {
            templateName="DefaultReport.vm";
        }
        return templateName;
    }
}
```

XNAT

Apache Turbine: Screen

- Renders View for user
- VM + Java

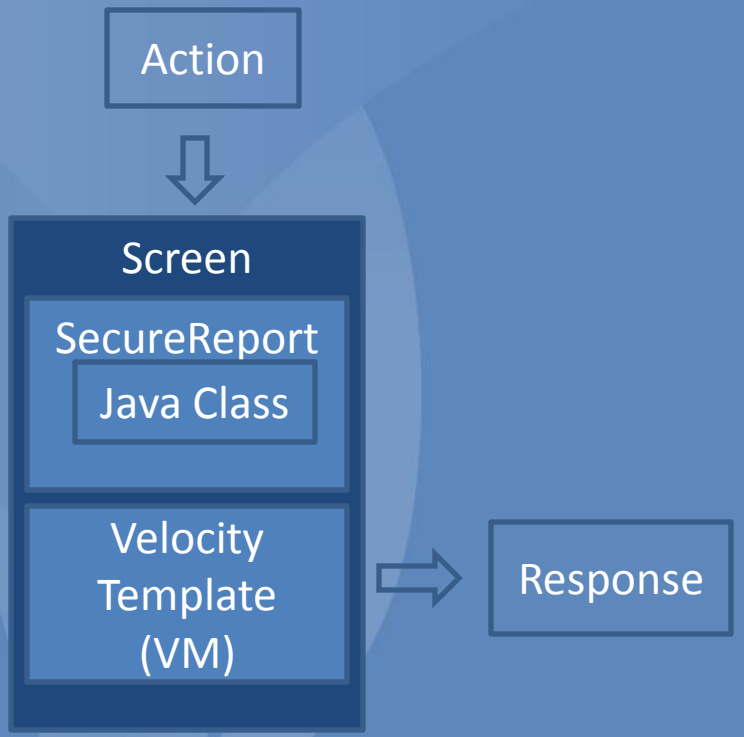


- Java file contains logic to build objects used in display
- Velocity template contains design specific content

XNAT

Apache Turbine: XNAT SecureReport

- Extends SecureScreen
- Loads Item data ($item)
  - passed item (data_item)
  - passed search parameters
- Loads ORM version of item
  - org.nrg.xdat.om.X ($om)
- Context object
  - used to pass objects to VM



XNAT

Apache Turbine: SecureReport

```
if(item==null){
    item = TurbineUtils.getDataItem(data);
}

if (item== null)
{
   try {
        ItemI temp = TurbineUtils.GetItemBySearch(data,preLoad());
        item = temp;
    } catch (IllegalAccessException e1) {
        logger.error(e1);
        data.setMessage(e1.getMessage());
        noItemError(data,context);
        return;
    } catch (Exception e1) {
        logger.error(e1);
        data.setMessage(e1.getMessage());
        data.setScreenTemplate("Error.vm");
        noItemError(data,context);
        return;
    }
}

if (item == null)
{
    data.setMessage("Error: No item found.");
    noItemError(data,context);
}else{
    try {
        context.put("item",item.getItem());
        context.put("user",TurbineUtils.getUser(data));

        context.put("element",org.nrg.xdat.schema.SchemaElement.GetElement(item.getXSITy
        if(data.getParameters().getString("search_element")!=null)
            context.put("search_element",data.getParameters().getString("search_element"
        else
            context.put("search_element", item.getXSIType());

        if(data.getParameters().getString("search_field")!=null)
            context.put("search_field",data.getParameters().getString("search_field"));
        else
            context.put("search_field", item.getXSIType() + ".ID");

        if(data.getParameters().getString("search_value")!=null)
            context.put("search_value",data.getParameters().getString("search_value"));
        else
            context.put("search_value", item.getProperty("ID"));

        om = BaseElement.GetGeneratedItem(item);
        context.put("om",om);

        finalProcessing(data,context);


    } catch (Exception e) {
        data.setMessage(e.getMessage());
        logger.error("",e);
    }
}
```

XNAT

imageSessionData_report.vm

```
<!-- BEGIN imageSessionData/xnat_imageSessionData_report.vm -->
#if($project)

#else
  #set($project=$om.getProject())
#end
<script type="text/javascript" src="$content.getURI("scripts/FileViewer.js")"></scri
<script type="text/javascript">
document.title = "Session: $!item.getProperty('label')";
</script>
#set($subject = $mr.getSubjectData())
#if($item.needsActivation())
<div id="quarantine_div" class="error" style="background-color:#FFCC33;">This sessio
#end
#parse($turbineUtils.getTemplateName("_title",$om.getXSIType(),$project))
<STYLE>
TH.underscore{
border-bottom-width:2px;
border-bottom-color:rgb(255,153,51);
border-bottom-style:solid;
}
</STYLE>
#if($project)

#else
#set($project=$mr.getProject())
#end
<BR>
#parse("/screens/workflow_alert.vm")
<table width="100%" align="left" border="0">
<tr>
  <td align=left>
#parse($turbineUtils.getTemplateName("_details",$om.getXSIType(),$project))
    </td>
  </tr>
  <tr>
   <td colspan=2>
#parse("/screens/ReportProjectSpecificFields.vm")
  </td>
 </tr>
 <tr>
  <td align=left colspan=2 valign=top>
<BR/><BR/>
#parse($turbineUtils.getTemplateName("_scans",$om.getXSIType(),$project))
  </td>
 </tr>
 <tr>
  <td align=left colspan=2 valign=top><BR/><BR/></td>
 </tr>
```

XNAT

imageSessionData_scans.vm

```
#set ($scanCounter = 0)
#foreach($scan in $om.getSortedScans())
#set ($scanID =$!scan.getProperty("ID"))
#set ($scanQC = $!scan.getManualQC())
<tr valign=top border=0>
<td border=0 align=left NOWRAP>
<a name="LINK${scanCounter}" href="#LINK${scanCounter}" onClick=" return blocking($scanCounter);">
<img ID="IMG$scanCounter" src="$content.getURI("images/plus.jpg")" border=0>
$!scan.getProperty("ID")
</a>
</td>
<td border=0 align=left NOWRAP>$!scan.getProperty("type")</td>
<td border=0 style="font-weight:bold; text-align:left;"  NOWRAP>
#if ($!om.getManualQC())
#if ($!scanQC.getStringProperty("pass").equals("1"))
<span style="color:green">Passed</span>
#elseif($!scanQC.getStringProperty("pass").equals("0"))
<span style="color:red">Failed</span>
#else
<span style="color:gold">
Unknown
#if ($!scanQC.getStringProperty("pass"))
($!scanQC.getStringProperty("pass"))
#end
</span>
#end
#else
<span style="color:
#if ($!scan.getProperty("quality").equalsIgnoreCase("usable"))
green
#elseif($!scan.getProperty("quality").equalsIgnoreCase("questionable"))
gold
#else
red
#end
">$!scan.getProperty("quality")
</span>
#end
</td>
<td border=0 align=left NOWRAP>#if($scan.getFile().size()>0)<div id="scans_$!scan.getId()_stats"><a
<td border=0 align=left NOWRAP>$!scan.getProperty("note")</td>
</tr>
```

XNAT

Apache Velocity

- Templating engine
- Java based
- Simple, but powerful language
- Used to generate any file format
  - Commonly HTML
- Still on version 1.3.1

Previous ORM

- Torque
  - Included by default with Turbine
  - Similar to Hibernate (much older)
- Problems
  - Searching problems
    - Unable to build searches that queried multiple tiers of the XNAT Data Hierarchy in one query
  - Required very specific schema definition
  - Difficult to customize for accessory tables

Search Engine

- Developed by NRG ~ 2004
- Built to address Searching problems
- Internalizes Schema structure
- Builds SQL based on XML Path references

  xnat:mrSessionData/project
- Identified relationships between elements
- Built Super Searches (searches across data types)

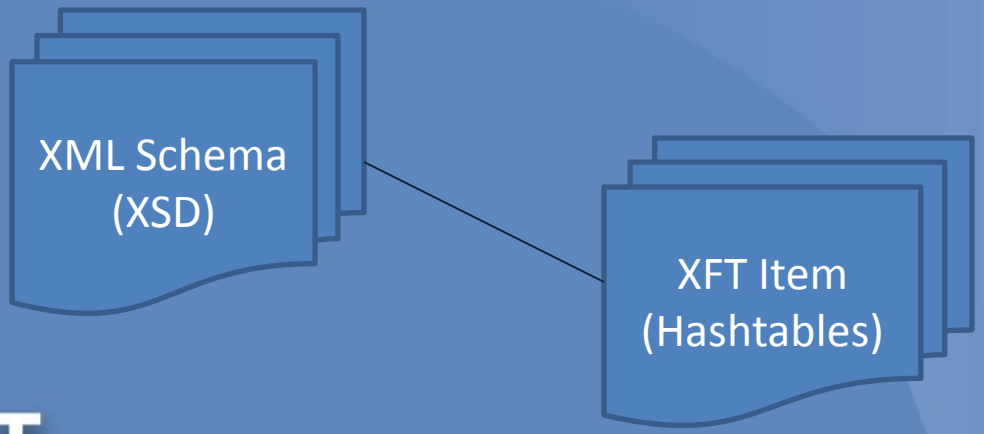
Display Documents

- Built to customize available search fields
  - Formatting
  - Derived values
  - Complicated SQL views

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by Tim Olsen (Washington University) -->
<Displays xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../xdat/display.xsd" schema-element="xnat:mrSessionData" full-description="MR Session"
brief-description="MR Session">
    <Arc-Definition id="ASSESSOR">
        <CommonField id="EXPT_ID" type="STRING"/>
        <CommonField id="ASSESSED_EXPT_ID" type="STRING"/>
        <Bridge-Element name="xnat:mrSessionData" field="SESSION_ID"/>
        <Filter field="EXPT_ID" filterType="distinct"/>
        <Filter field="ASSESSED_EXPT_ID" filterType="equals"/>
    </Arc-Definition>
    <Arc name="PARTICIPANT_EXPERIMENT">
        <CommonField id="PART_ID" local-field="SUBJECT_ID"/>
        <CommonField id="DATE" local-field="DATE"/>
        <CommonField id="EXPT_ID" local-field="SESSION_ID"/>
    </Arc>
    <DisplayField id="DATE" header="Date" visible="true" searchable="true" data-type="date" sort-order="DESC">
        <DisplayFieldElement name="Field1" schema-element="xnat:mrSessionData.date"/>
        <description>MR Session Date (DATE)</description>
    </DisplayField>
    <DisplayField id="DATE_CHAR" header="Date" visible="true" sort-by="DATE" sort-order="DESC">
        <DisplayFieldElement name="Field1" schema-element="xnat:mrSessionData.date"/>
        <Content type="sql">CAST(@Field1 AS VARCHAR)</Content>
        <description>MR Session Date (String)</description>
    </DisplayField>
    <DisplayVersion versionName="listing" default-order-by="DATE" dark-color="DEDEDE" light-color="FFFFFF" default-sort-order="DESC">
        <DisplayFieldRef id="LABEL"/>
        <DisplayFieldRef id="DATE"/>
        <DisplayFieldRef id="PROJECT"/>
        <DisplayFieldRef id="SUBJECT_LABEL" element_name="xnat:subjectData"/>
        <DisplayFieldRef id="GENDER_TEXT" element_name="xnat:subjectData"/>
        <DisplayFieldRef id="AGE"/>
    </DisplayVersion>
    <ViewLink alias="ORDERED_WORKFLOWS">
        <Mapping TableName="ORDERED_WORKFLOWS">
            <MappingColumn rootElement="xnat:mrSessionData" fieldElement="xnat:mrSessionData.ID" mapsTo="id"/>
        </Mapping>
    </ViewLink>
    <SQLView name="ordered_workflows" sql="SELECT wrk_workflowdata.id, status, wrk_workflowdata_id,
     CASE pipeline_name
        WHEN 'Transfer'::text THEN 'Archive'::text
       ELSE
         CASE xs_lastposition('/'::text, pipeline_name::text) WHEN 0 THEN pipeline_name ELSE
         substring(substring(pipeline_name::text, xs_lastposition('/'::text, pipeline_name::text) + 1), 1, xs_lastposition('.'::text, substring(pipeline_name::text, xs_lastposition('/'::text, pipeline_name::text) +
1)) - 1)
      END END AS pipeline_name
   FROM wrk_workflowdata
   RIGHT JOIN ( SELECT wrk_workflowdata.id, max(wrk_workflowdata.launch_time) AS launch_time
         FROM wrk_workflowdata
        GROUP BY wrk_workflowdata.id) wrk_max ON wrk_workflowdata.id::text = wrk_max.id::text AND wrk_workflowdata.launch_time = wrk_max.launch_time"/>
</Displays>
```

XFT: eXtensible Formatting Toolkit

- Used internal XSD representation
- Developed by NRG ~2005
- Dynamic Object model
  - Generic Object structure (Hashtables)
  - Data structure loaded from XSD



XNAT

XFT

- Manages Item level transactions
  - Storage (~2005)
  - Retrieval (~2004)
  - Reformatting (XML, SQL, CSV, HTML, etc)
- Generates schema (~2005)
  - SQL Database schema
  - Previously Torque schema
- Generates Java files

XFT

- Storage
  - Uses unique fields to match to pre-existing rows in the database
  - Adds missing rows/ modifies existing rows
  - Deletes missing fields (allowDataDeletions=true)
- Retrieval
  - Item Functions
  - Direct SQL generation

XFT: XML to XFTItem

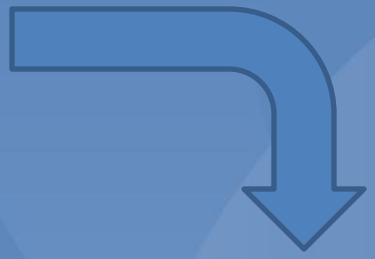
```
<xnat:Subject ID="LOCAL_S00002" project="TEST"
group="control" label="TEST1">
   <xnat:demographics
xsi:type="xnat:demographicData">
      <xnat:yob>1975</xnat:yob>
      <xnat:gender>male</xnat:gender>
   </xnat:demographics>
</xnat:Subject>
```

```
XFTItem {
   element_name:"xnat:subjectData",
   properties:hashtable{
      ID:"LOCAL_S00002",
      group:"control"
      project:"TEST"
      label:"TEST1"
      demographics: XFTItem{
         element_name:"xnat:demographicData"
         properties:hashtable{
            yob:"1975",
            gender:"male"
         }
      }
   }
}
```

XNAT

XFT: XFTItem – SQL/CSV

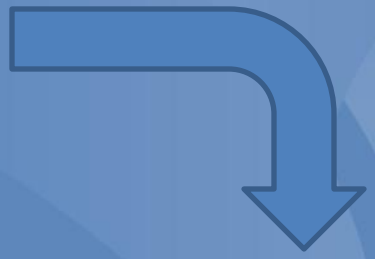
```
XFTItem {
   element_name:”xnat:subjectData”,
   properties:hashtable{
      ID:”LOCAL_S00002”,
      group:”control”
      project:”TEST”
      label:”TEST1”
      demographics: XFTItem{
         element_name:”xnat:demographicData”
         properties:hashtable{
           yob:”1975”,
           gender:”male”
         }
      }
   }
}
```



SQL

Database TABLE

ID	PROJECT	GROUP	LABEL
“LOCAL…”	“TEST”	“control”	“TEST1”



CSV

CSV File

ID	PROJECT	GROUP	LABEL
“LOCAL…”	“TEST”	“control”	“TEST1”

XNAT

XDAT: eXtensible Data Archiving Toolkit

- Developed by NRG ~ 2005-2006
- Security
  - Needed security in multiple places
    - Java (loading of Items from XML, CSV, etc)
    - SQL (loading of query data)
  - Needed way of managing users, groups, data-types
- Navigation structure (Data-types, Actions)
- Search Engine

Maven 1.0.2

- Build tool
- Added in migration to Turbine 2.3
- Dependency management
  - This failed for us (relocated jars)
- Plugin architecture with rich Ant support

Maven 2

- Introduced recently to manage XNAT related code
  - XDAT_CORE, DICOM Tools, Custom Tools
- All new development is moving to Maven 2
  - Encourages coding/design conventions
  - Integrated build/testing support
  - More versatile dependency management

Questions

- ???

XNAT