

XML Application Components (XACs)

*Rafah A. Hosn
IBM Research
May 11th, 2007*

WWW 2007

W3C[®] TRACK

Outline

- **Backplane and Web 2.0**
- **Web 2.0 Widgets & Composite Apps**
- **XACs, Component Model for Widget Composition**
- **Demo & Code Walkthrough**

Agenda

- **Backplane and Web 2.0**
- **Web 2.0 Widgets & Composite Apps**
- **XACs, Standard Composition Model**
- **Demo & Code Walkthrough**

Backplane & Web 2.0

■ Web 2.0

- No Longer A Buzzword ?!
- Main themes
 - Data-centric
 - Collaborative
 - Simple to author, simple to use
 - Remixable

■ How can the Backplane help?

- Standardize the programming model
 - Easily snap various namespaces together
 - Events
 - Data model lifecycle
 - Choreography
 - Component Composition model

Rich Web Application Backplane

Interaction Namespaces and Frameworks

XHTML

SVG

XForms

VoiceXML

SMIL

Composite Apps

Rich Web Application Backplane

Compound
Document
Framework

Data
Model

Model-view
binding
and events

Submission object
(XMLHTTP++)
and behavior

Choreography
(SCXML)

Component Model
and Composition

Platform technologies

DOM Parser

XPath, XSLT

DOM Events

JS

Security

Agenda

- **Backplane and Web 2.0**
- **Web 2.0 Widgets & Composite Apps**
- **XACs, Standard Composition Model**
- **Demo & Code Walkthrough**

Web 2.0 Widgets & Composite Apps

- **Web 2.0 Widgets today**

- Lots of various platforms & widget component models
- Google, Yahoo, Adobe, Clearspring, Netvibes, Microsoft...

- **Composing widgets**

- Different per platform/widget provider

- **No standards to date**

- Interoperability is not seamless (if at all possible)
- Mashing up widgets from different platforms is difficult
- Web 2.0 value-add proposition is weakened
 - Simplicity
 - Collaboration
 - Remixability

Agenda

- Backplane and Web 2.0
- Web 2.0 Widgets
- XACs, Standard Composition
- Demo & Code Walkthrough

XML Application Components (XACs)

■ XACs

- Loosely coupled web components
- Assembled using Backplane infrastructure
 - XML Events
 - XML/JSON data model
 - SCXML controller

■ Features of XACs

- Aggregation
- Composition
- Customization
- Re-use

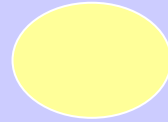
XML Application Components (XACs)

- **Anatomy of an XAC**
 - **Any web component that supports**
 - **A public data model**
 - **Method to initialize its public data model**
 - **Method that returns a portion of (or its entire) data model to caller**
- **Support for both scriptable and declarative interface**

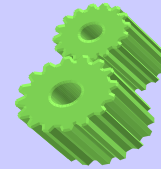
Create Document 2.0 widgets using XAC Component Model

Compound document

Root language,
e.g. XHTML, XUL



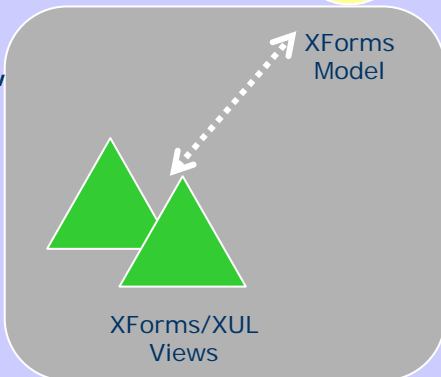
Data Model
Ex: JSON, XForms



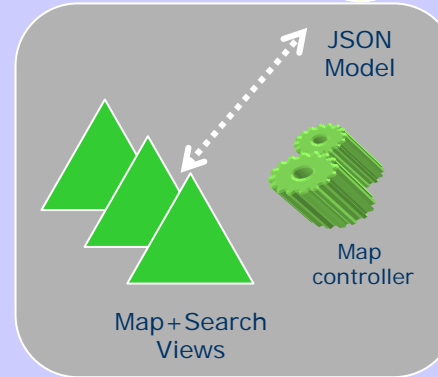
Top-level App Controller
Coordinates between map and
XHTML/XForms frames

XHTML View

Weather XAC
Component



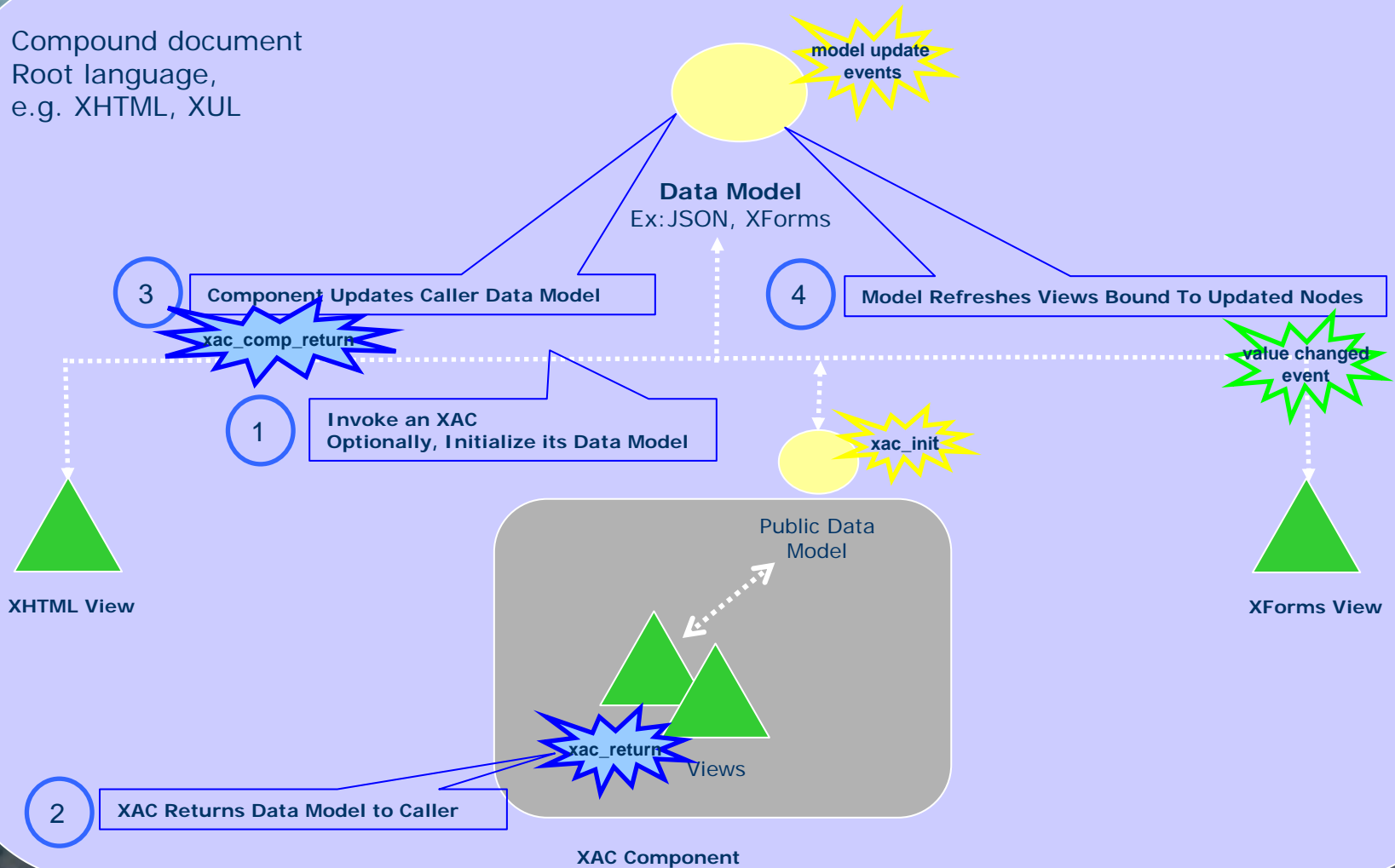
Map XAC
Component



XForms View

XAC Lifecycle

Compound document
Root language,
e.g. XHTML, XUL



Agenda

- Backplane and Web 2.0
- Web 2.0 Widgets
- XACs, Standard Composition
- Demo & Code Walkthrough



Thank You !

Pictures courtesy of: www.math.unc.edu/Faculty/petersen/Pics/Banff03/

WWW 2007

```
<xul:window
  id = "weathermap" title = "Weather Map" width = "600" height = "400"
  xmlns=http://www.w3.org/1999/xhtml xmlns:xac="http://www.ibm.com/2006/xac"
  xmlns:xf=http://www.w3.org/2002/xforms xmlns:ev=http://www.w3.org/2001/xml-events
  xmlns:xul="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
  persist="screenX screenY" orient="horizontal" >
```

```
<!-- Page data -->
<xf:model id="appDM">
  <xf:instance id="inst-app" xmlns="">
    <instanceData>
      <selectCity/>
    </instanceData>
  </xf:instance>
</xf:model>
```

```
<!-- Page Views & XAC Components →
```

```
<xul:hbox>
  <!-- weather component to display the weather panel -->
  <xac:comp id="weatherPanel" src="weather.xml" ref="selectCity/city/zip" />
  <!-- map component to show a map of USA and allow user to initial the markers -->
  <xac:comp id="mapPanel" src="map.xml" ref="selectCity">
    <!-- Initialize all the cities to be display on the map -->
    <xf:output> My first XAC-based mash-up </xf:output>
  </xul:hbox>
```

BACK

```
<xac:xac xmlns="http://www.w3.org/1999/xhtml" ... >
```

```
<!-- XForms Public Data Model -->
```

```
<xf:model id="inDM" xac:public="true">
```

```
<!-- input data -->
```

```
<xf:instance id="city" xmlns="">
```

```
<instanceData>
```

```
<selectCity>
```

```
<city>
```

```
<zip/>
```

```
</city>
```

```
</selectCity>
```

```
</instanceData>
```

```
</xf:instance>
```

```
</xf:model>
```

```
<!-- Private Data Model -->
```

```
<xf:model id="wDM" >
```

```
<xf:instance id="weather" xmlns="">
```

```
<weather/>
```

```
</xf:instance>
```

```
</xf:model>
```

```
.....
```

```
<xf:output model="inDM" class="getweather" value="selectCity/city/zip" />
```

```
<xul:vbox id="area">
```

```
<xf:output model="wDM" class="size9 bold" value="loc/dnam"/>
```

```
<xf:output model="wDM" class="size9 bold" value="instance('weather')/cc/t"/>
```

```
<xul:hbox id="weather">
```

```
<xul:image id="pic" src="weather/weather15.png"/>
```

```
<xf:output model="wDM" class="huge bold" value="concat(instance('weather')/cc/tmp, 'F')"/>
```

```
</xul:hbox>
```

[BACK](#)