



Multi-modal and Multi-device Services

3G Deployments based on reusing existing standards



Stéphane H. Maes, IBM
smaes@us.ibm.com



Definitions / Positioning

- ▶ **Channel:** a particular user agent, device, or a particular modality - characterized by a delivery context
- ▶ **Mono-channel applications:** applications designed for access through a single channel
- ▶ **Multi-channel applications:** applications designed for access through different channels
- ▶ **Multi-modal or multi-device applications:** A particular channel, where several modalities or devices are available and synchronized sequentially or simultaneously:
 - ▶ Different granularity of synchronization:
There are numerous commonalities between interactions using **several devices (multi-device interaction)** and **several modalities (multi-modal interaction)**
Related to coordination of web services
- ▶ **Conversational application: Multi-modal or voice**
 - ▶ Free-flow / mixed initiative
 - ▶ Dialog management and Disambiguation
 - ▶ Context / history management
 - ▶ NLP: NLU / NLG

Multi-channel / Mono-Channel Scenarios

- Multiple access devices
- One interaction mode per device

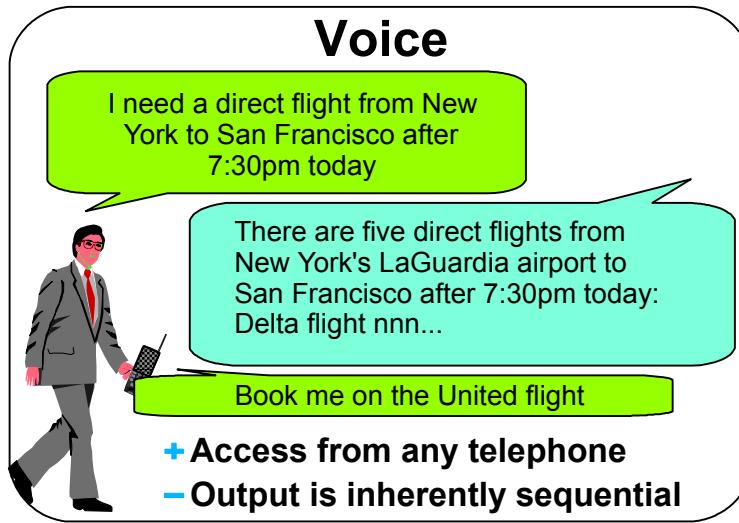
PC



Flights Hotels Cars Packages Cruises Maps
EXPRESS SEARCH
Departing from: _____ Going to: _____
When are you leaving? When are you returning?
[Dec] [31] [Noon] [Jan] [1] [Noon]
Tip: We have many more [flight](#), [hotel](#), and [car](#) options.
WHAT'S NEW
[Ski Travel: Choose from more than 80 ski destinations](#)
[Cruise Travel: Take a virtual tour of select cruise ships](#)

+ Standardized rich visual interface
- Not suitable for mobile use

Voice



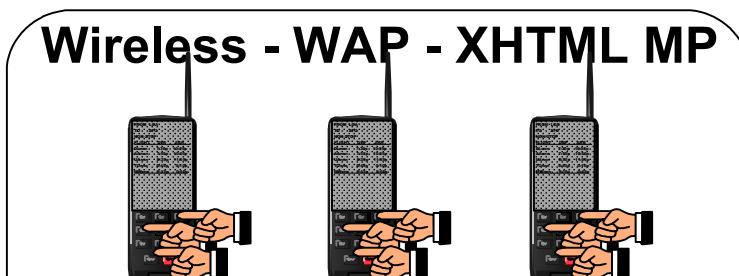
I need a direct flight from New York to San Francisco after 7:30pm today

There are five direct flights from New York's LaGuardia airport to San Francisco after 7:30pm today:
Delta flight nnn...

Book me on the United flight

+ Access from any telephone
- Output is inherently sequential

Wireless - WAP - XHTML MP



From: LGA _____
To: _____
Date: _____

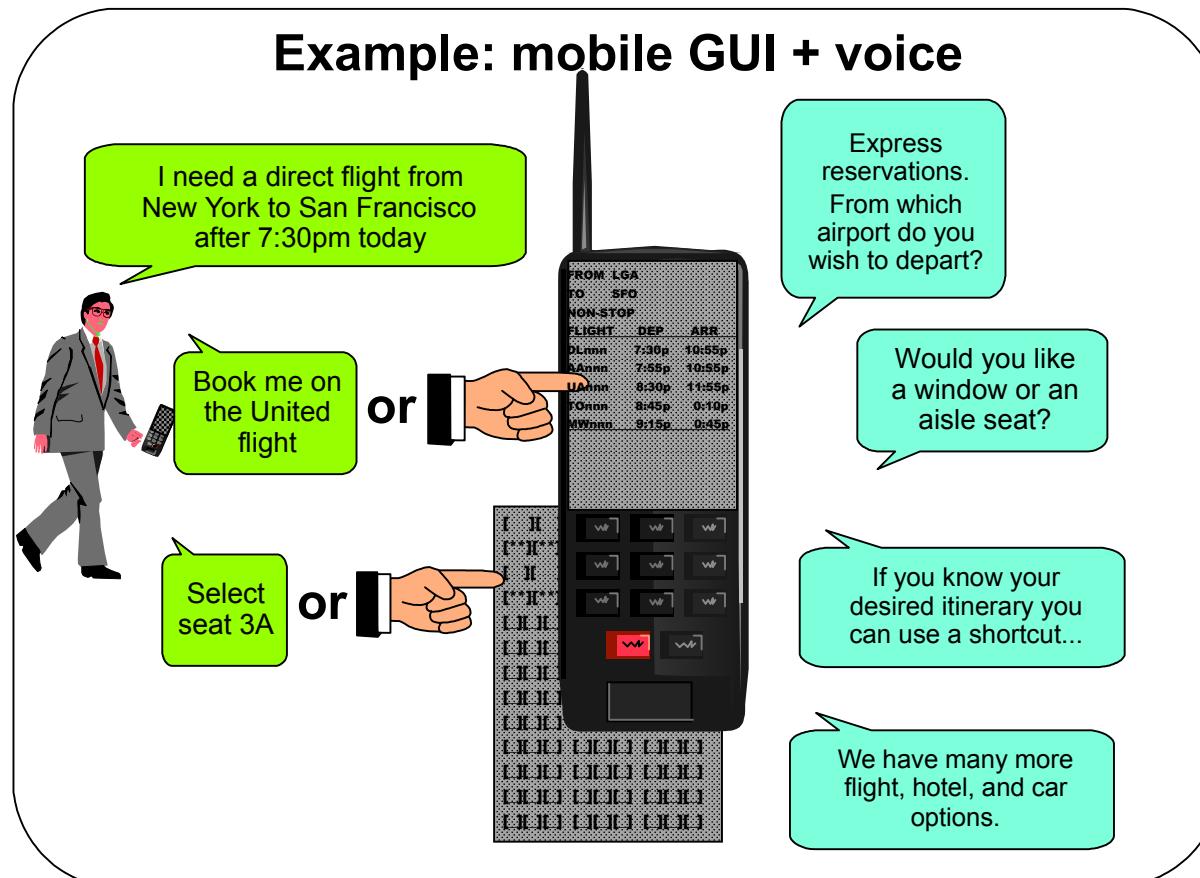
From: LGA _____
To: SFO _____
Date: _____

From: LGA _____
To: SFO _____
Date: 00/12/11

+ Mobile and becoming ubiquitous
- Hard to enter data

Multi-modal Interactions

- User can select at any time the preferred modality of interaction
- Can be extended to selection of the preferred device (multi-device) and other modalities



Additional examples:

- display seat selection chart (not simply "window or aisle")
- use voice or keys to enter PIN code and performs speaker verification
- use audio or voice for notifications
- information can be saved for later use
- Suspend and Resume Scenarios
- Other modalities: handwriting (tablets), video, ...
- Multi-device (kiosks, PDA, Phone, remote control)

- User is not tied to a particular channel's presentation flow
- Interaction becomes a personal and optimized experience
- Multi-modal output is an example of multi-media where the different modalities are closely synchronized.

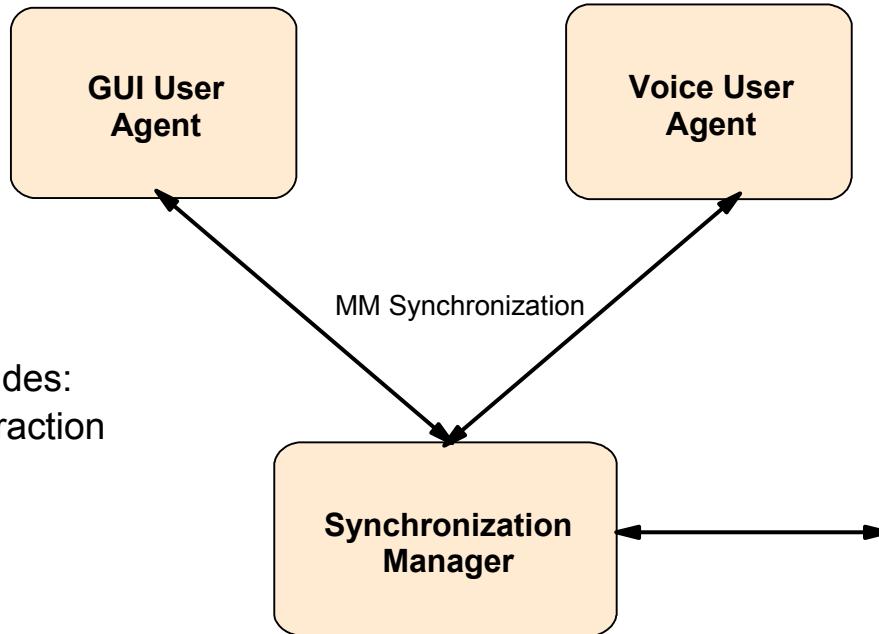
Multi-modal Computing Value Proposition

Multi-modal Computing value proposition

- **easily enter and access data using small devices**
 - ▶ by combining multiple input & output modes
- **choose the interaction mode that suits the task and circumstances**
 - ▶ input: key, touch, stylus, voice...
 - ▶ output: display, tactile, audio...
- **reliable access:**
 - ▶ User are no more blocked by limitations / mistakes of a given interaction mode at a given moment
- **use several devices in combination**
 - ▶ by exploiting the resources of multiple devices
- **In the future: conversational extension**
 - ▶ Interacts with and across application as in an every day dialog: free form, mixed initiative and context management

Basic Architecture - Model View Controller (MVC)

Each piece is distributable, including speech engines

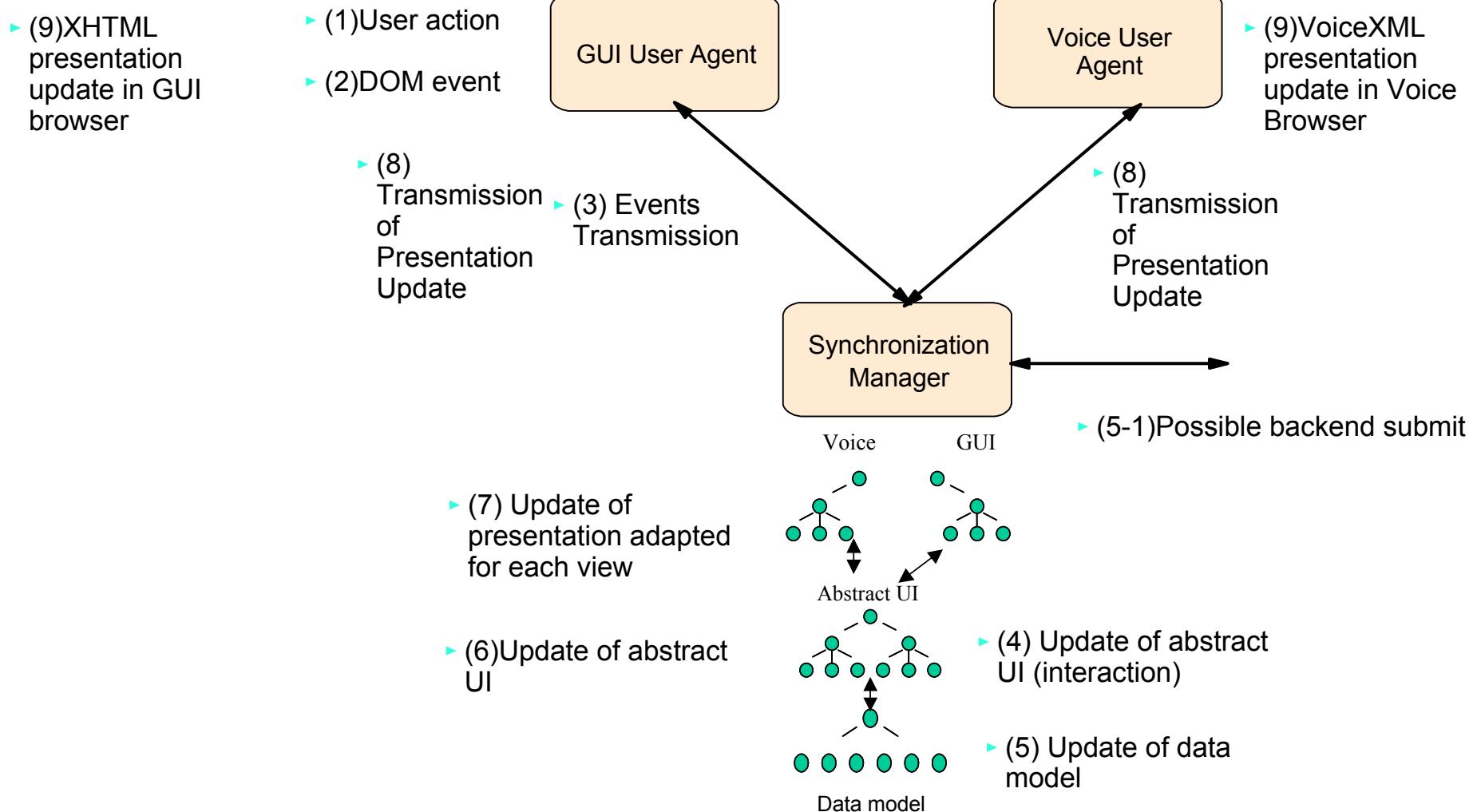


- ▶ Synchronization manager includes:
 - ▶ XForms processor with interaction state
 - ▶ Presentation UI adaptation
 - ▶ Synchronization
 - ▶ ...
- ▶ The model can be distributed.

- ▶ Requirements:
 - ▶ Multi-modal Synchronization:
 - ▶ interaction events
 - ▶ user agent presentation update
 - ▶ Remote synchronization when views are distributed
 - ▶ Discovery/registration/initiation when views are distributed
 - ▶ When voice engines are distributed, it may be advantageous to use:
 - ▶ Distributed Speech Recognition (DSR)
 - ▶ Speech Engine Remote Control (SERC)
 - ▶ End-to-end deployment requirements to be addressed (security, quality of service, etc...)

MVC Multi-modal Browser - Application Model and Flow

Interaction: Assuming GUI interaction & XForms authoring



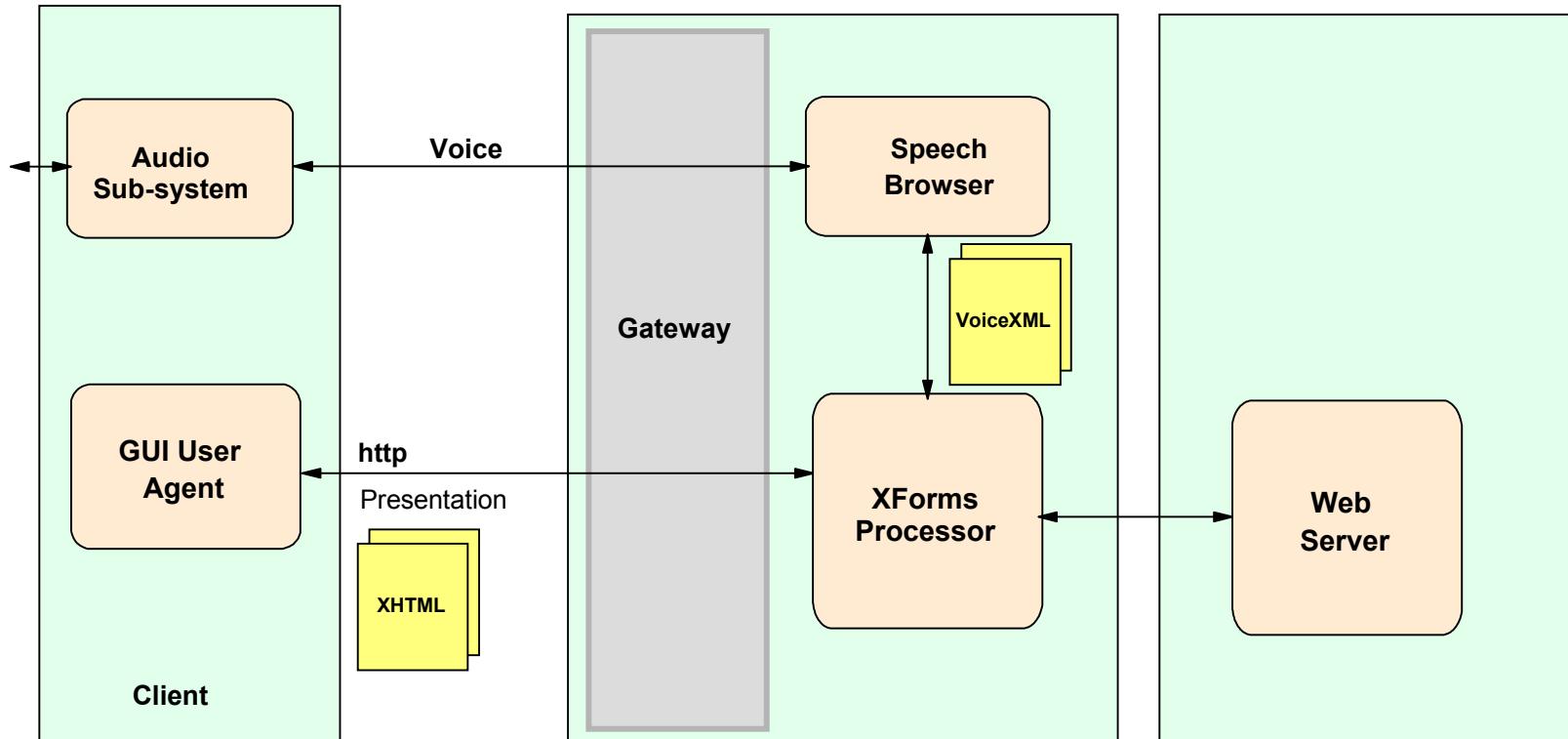
Note: instead of XForms, the application can be manually authored (XHTML and VoiceXML or XHTML+Voice) and bound to XForms data model.

Related framework

- ▶ WSXL - Web Service eXperience Language submitted to OASIS WSIA and WSRP
 - ▶ <http://www-106.ibm.com/developerworks/library/ws-wsxl/index.html>
 - ▶ Synchronization, coordination , combination of web services

Interoperable Deployment Configurations

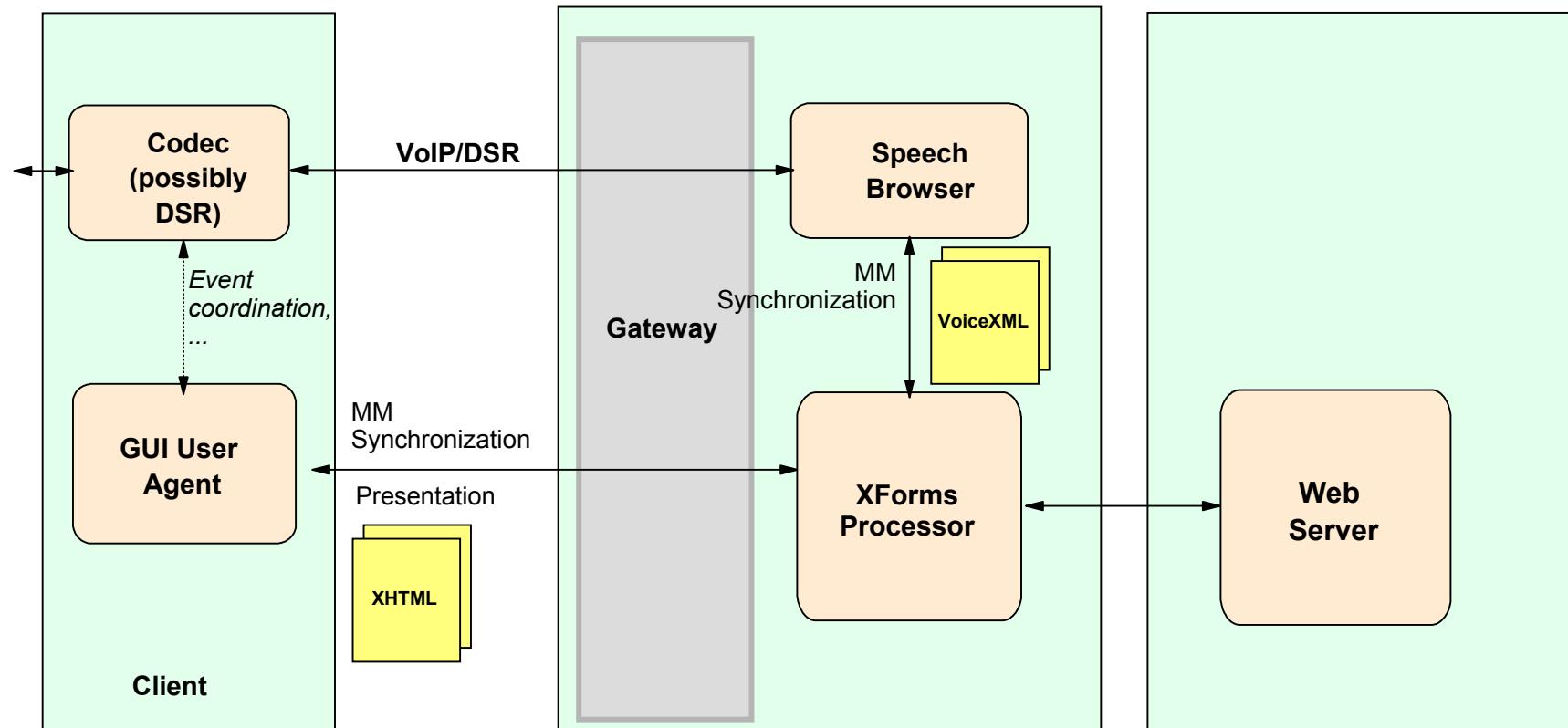
Sequential: (no voice and data support)



Interoperable Deployment Configurations

Thin Client Configuration (voice and data support)

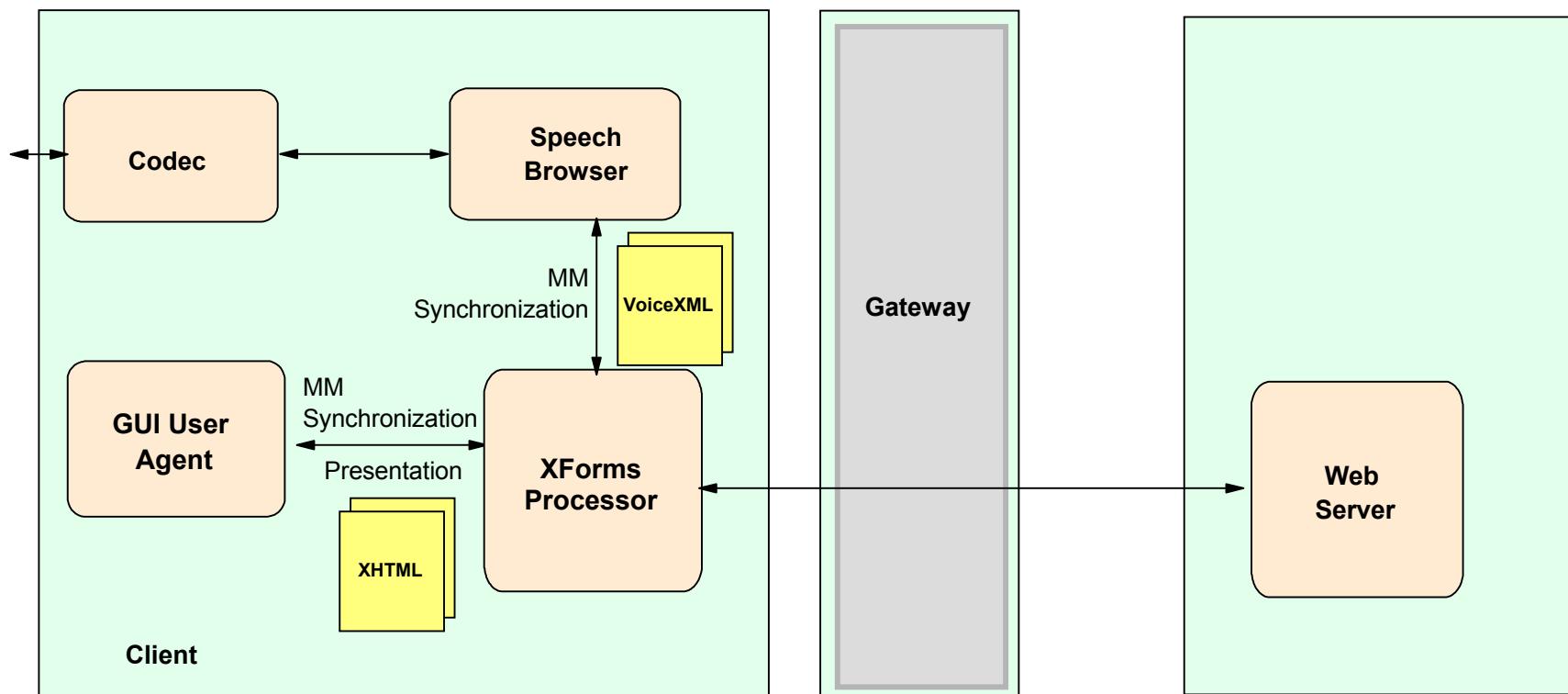
Engines can be local or distributed to the voice browser



Interoperable Deployment Configurations

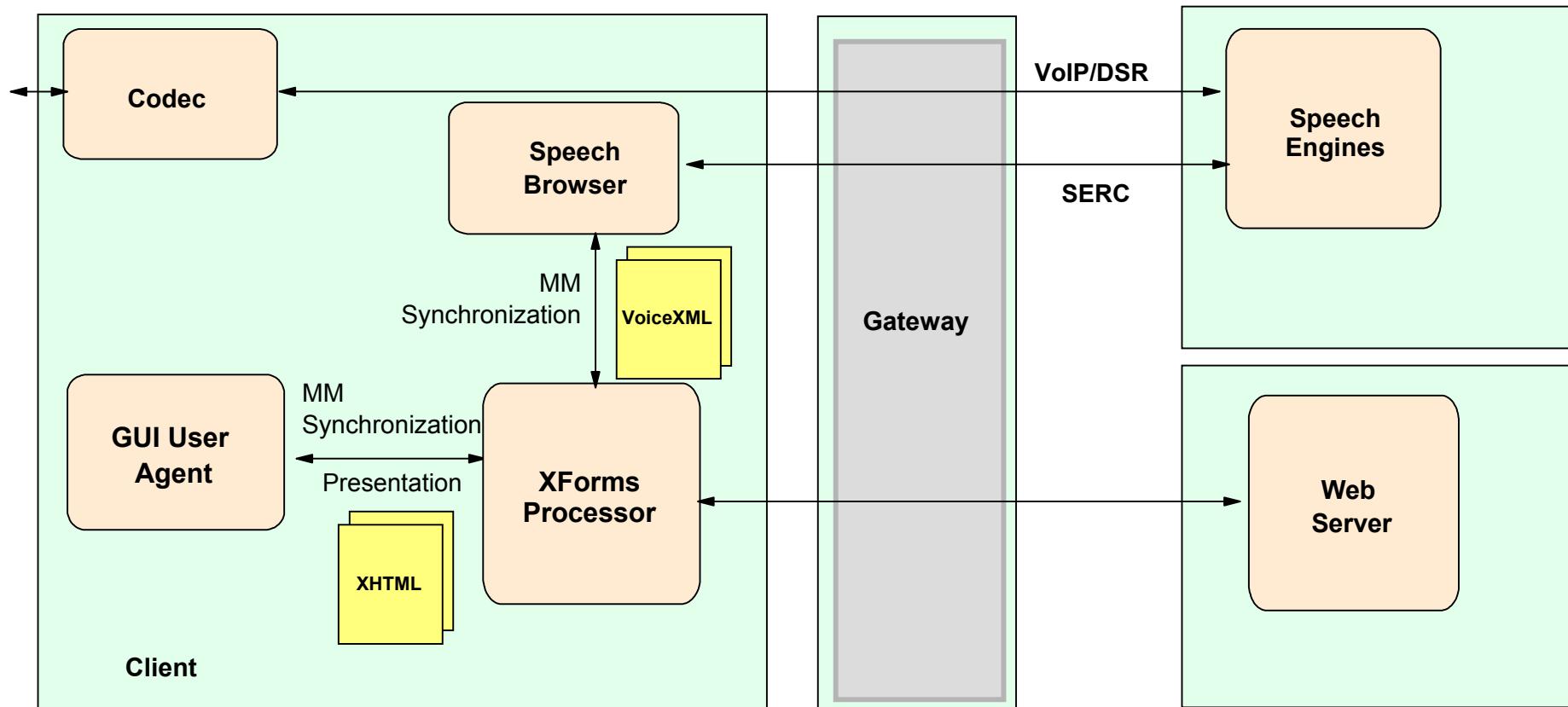
Fat client configuration with local speech engines

This can be the internal architecture of a browser implementation



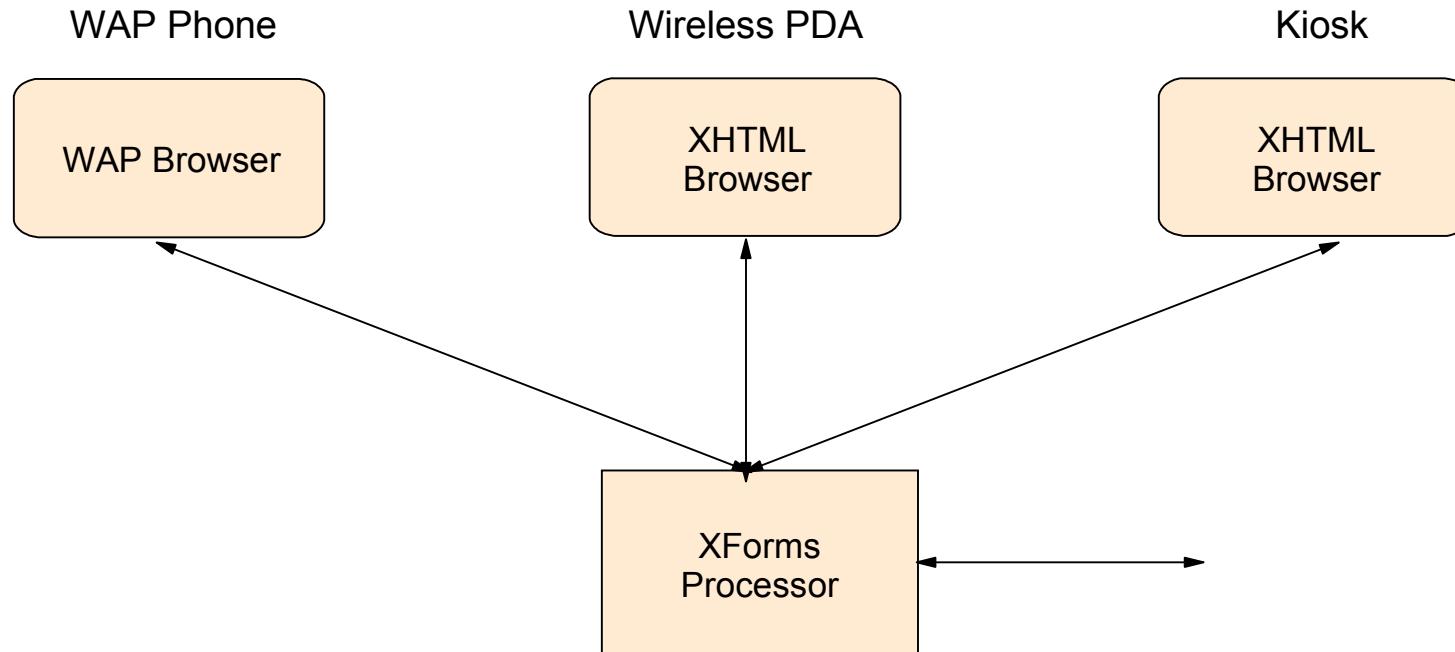
Interoperable Deployment Configurations

Fat client configuration with remote speech engines



Interoperable Deployment Configurations

Multi-device:

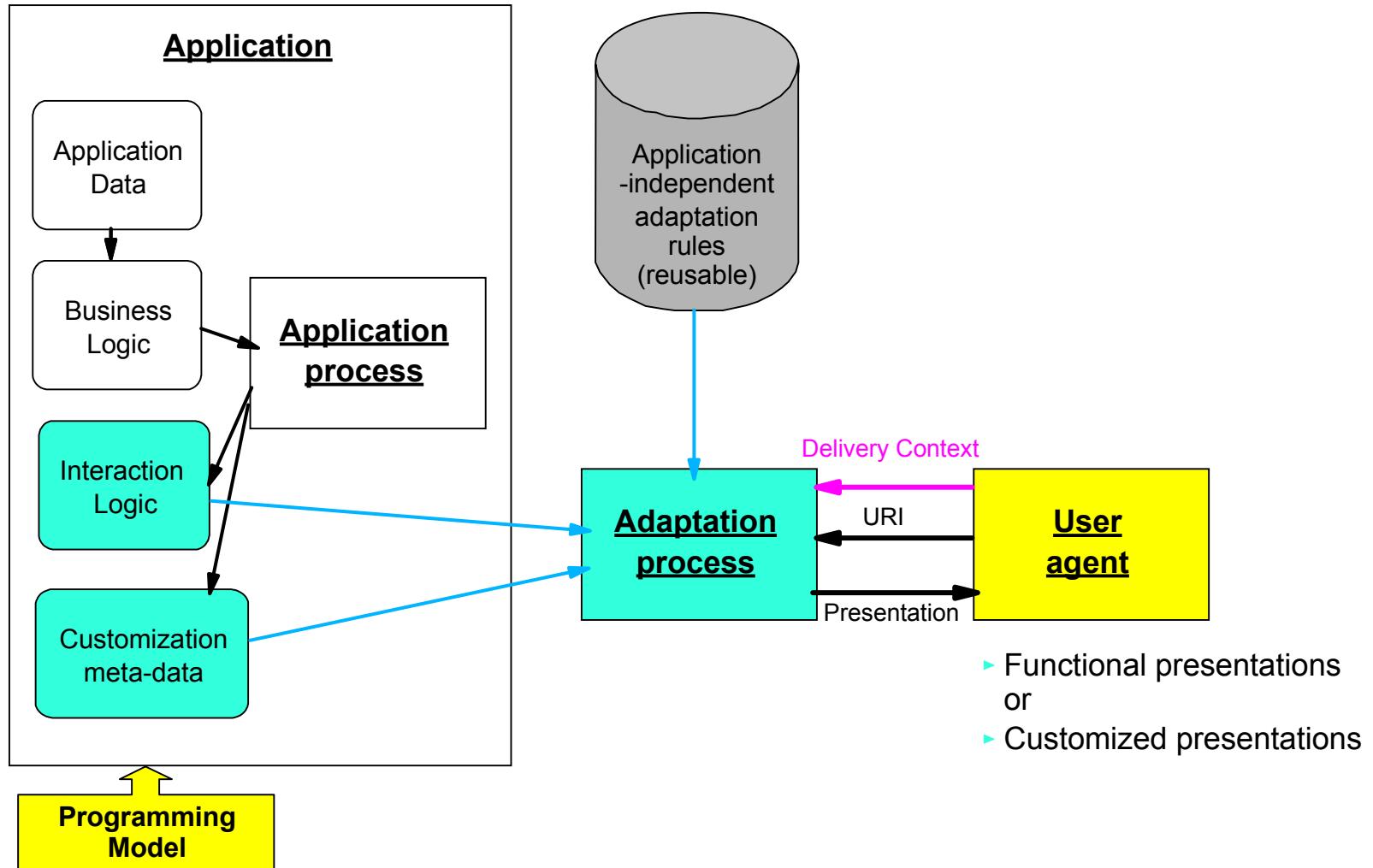


- ▶ With Server-side synchronization
 - ▶ e.g. Kiosk / phone synchronization, Navigation system and phone
 - ▶ e.g. multiple user / shared applications
- ▶ With Spontaneous networking and synchronization performed among the devices.
 - ▶ e.g. remote console for accessibility to appliance (e.g. NCITS V2)
 - ▶ e.g. remote control
 - ▶ PDA (PIM) and phone

Main Architecture Features

- ▶ Interoperable and extensible:
 - ▶ Across runtime configurations
 - ▶ Across content
 - ▶ Across different runtime / infrastructure capabilities (client, server, infrastructure)
- ▶ Support:
 - ▶ Thin and fat configurations with local or remote speech engines
 - ▶ Suspend and resume
 - ▶ Dynamic multi-device or multi-modal sessions:
 - ▶ more than 2 views
 - ▶ changing views and configurations:
 - ▶ e.g. hybrid thin/fat client configuration.
- ▶ Relies on:
 - ▶ Existing standard interfaces and protocols practices (to be defined for some channels)
 - ▶ Evolution of web (web services, intermediaries, etc...)
 - ▶ Fit evolution of wireless infrastructure
- ▶ Directly compatible with evolution of W3C authoring specifications and reusing existing authoring standards
 - ▶ XHTML + Voice
 - ▶ XForms

Generic DI Authoring



<http://www.w3c.org/DI>

XFORMS-based DI AUTHORING

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<html>
<head>
<model id="dob">
    <instance>
        <person>
            <birthdate xsi:type="xsd:date"/>
        </person>
    </instance>
</model>
</head>
<body>
    <xforms:input ref="/person/birthdate" class="dobui">
        <xforms:caption>Enter your date of birth:</xforms:caption>
    </xforms:input>
</body>
</html>
```

Abstract UI
To be customized

Multi-modal & Multi-device Browsing Deployment

Abstract DOB Information (XForms)

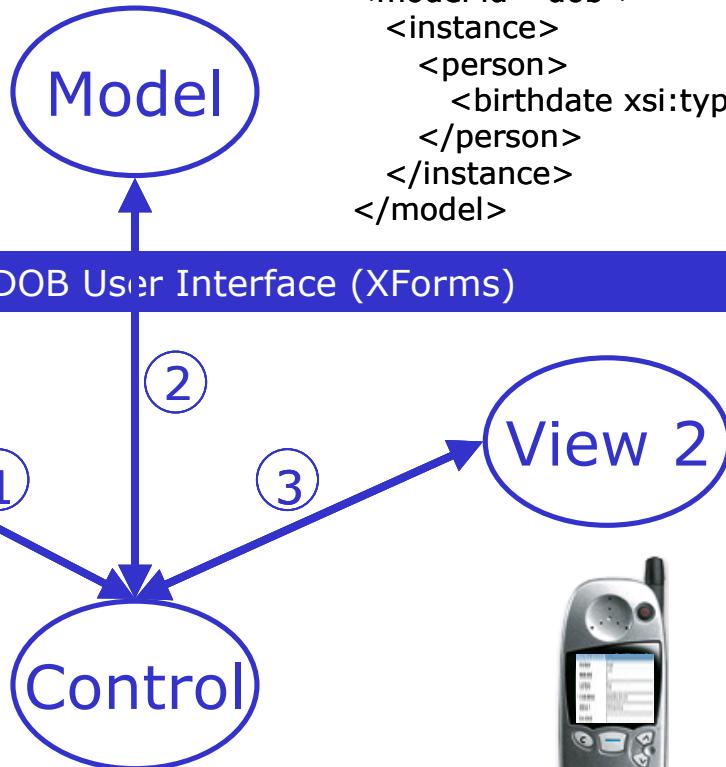
User interaction results in the **control** updating the **model**, and these updates being reflected in the various **views**.

```
<model id="dob">
<instance>
  <person>
    <birthdate xsi:type="xsd:date"/>
  </person>
</instance>
</model>
```

Abstract DOB User Interface (XForms)

The screenshot shows a Microsoft Internet Explorer window with the title "My developerWorks profile - Microsoft Internet Explorer". The page displays a form titled "Please tell us about yourself" with fields for First Name (Angel), Middle Initial (L), Last Name (Diaz), E-mail address (aldiaz@us.ibm.com), Address 1 (158 Duke Drive), and Date of birth (10/30/2001). The browser interface includes a menu bar (File, Edit, View, Favorites, Tools, Help) and a toolbar.

HTML 4.1 Forms



WML input card

XFORMS AUTHORING ALTERNATIVES

- ▶ Alternatives or intermediate steps towards DI authoring:
 - ▶ The presentation can also be manually authored and bound to data model
or
 - ▶ The presentation may be adapted by transformation of an abstract UI using manually authored stylesheets (instead of re-usable stylesheet + customization as in DI authoring)

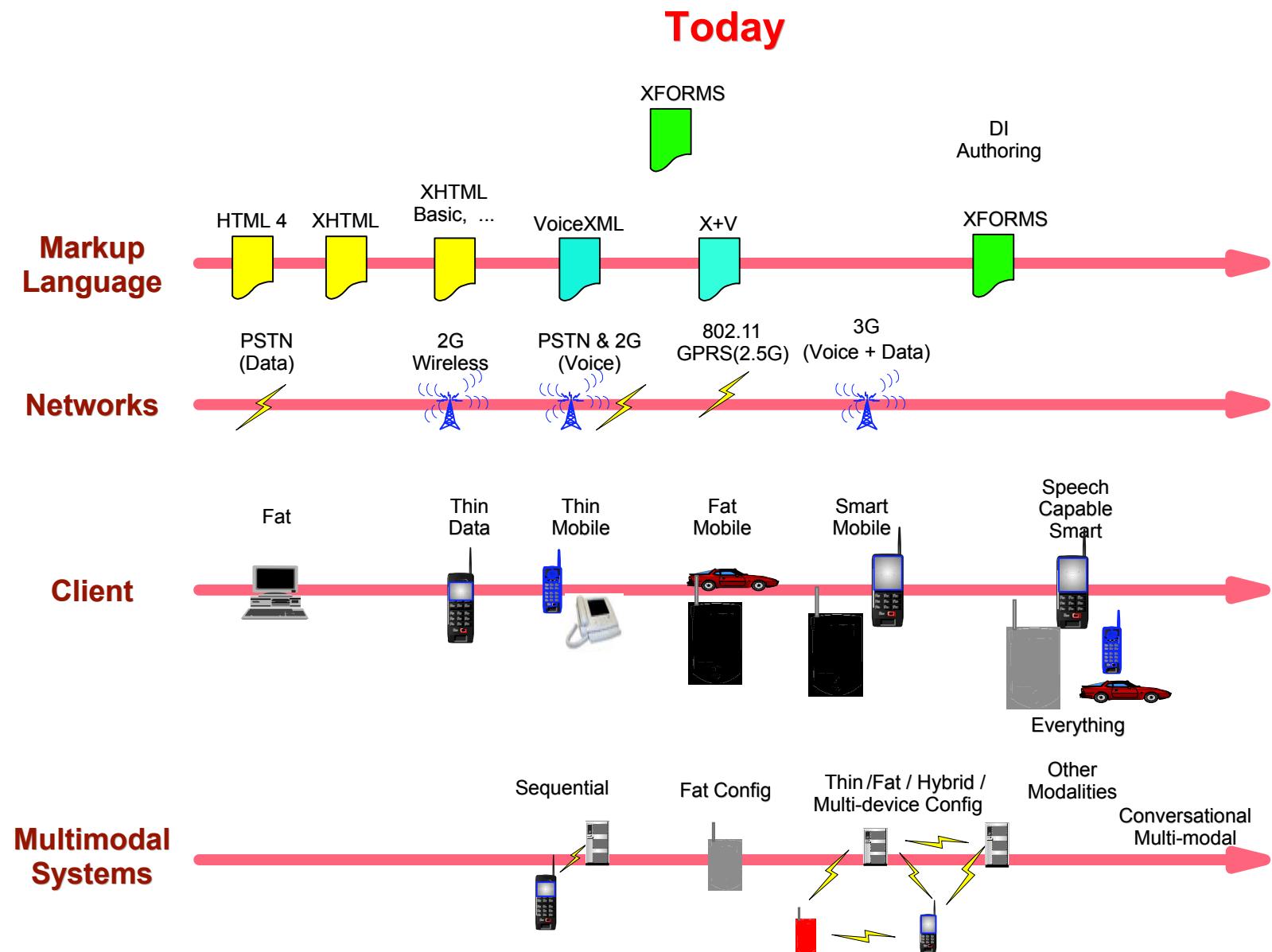
XHTML + Voice Profile Authoring

- ▶ XHTML+VXML Profile authoring for fat client Voice/GUI configurations:
 - ▶ See <http://www.w3.org/TR/xhtml+voice/>
 - ▶ Relies on XML events, VoiceXML modularization
 - ▶ Fits as a XHTML module
- ▶ The profile includes the XHTML modules and the following Voice XML 2.0 modules:
 - ▶ Speech and non-speech audio Output
 - ▶ Command And Control
 - ▶ Speech Grammars
 - ▶ Voice XML Event Types
 - ▶ Voice XML Event Handlers
- ▶ XHTML+VXML Language may be extended by other W3C recommendations, or by private extensions. For these extensions, the following rules must be obeyed:
 - ▶ All elements introduced in extensions must have a skip-content attribute if it should be possible that their content is processed by XHTML+VXML user agents.
 - ▶ Private extensions must be introduced by defining a new XML namespace.
- ▶ Specs provide examples of conversational multi-modal application

XHTML + Voice Profile Authoring Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml+voicexml10.dtd">
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:vxml="http://www.w3.org/2001/voicexml20"
      xmlns:ev="http://www.w3.org/2001/xml-events">
  <head>
    <title>Skeleton XHTML+VXML Document</title>
  </head>
  <body>
    <!-- first declare the voice handlers. -->
    <h1>Skeleton XHTML+VXML Document</h1>
    <vxml:form id="sayHello">
      <vxml:block>Hello World</vxml:block>
    </vxml:form>
    <!-- done voice handlers -->
    <p ev:event="mouseclick" ev:handler="#sayHello">
      This is a sample document designed
      to illustrate the markup structure of a conformant
      XHTML+VXML document. Notice that the default XML
      namespace is XHTML --and consequently, standard HTML
      element names do not need a namespace prefix. We can
      add voice-interaction specific elements from the VoiceXML 2.0 namespace using prefix
      <code>vxml</code>. We can attach event handlers using prefix <code>ev</code>.
      Clicking anywhere on this paragraph results in a welcome
      message being spoken on account of attaching a
      <code>vxml:block</code> handler to this paragraph.
    </p>
  </body>
</html>
```

Evolution of Multi-modal Deployments



Conclusions

- ▶ Multi-modal and multi-device authoring can be done today using the W3C stack of application model standards.
- ▶ Multi-modal and multi-device browsers can be deployed today based on the MVC architecture.
- ▶ The MVC architecture fits the evolution of the web and the wireless infrastructure.
- ▶ Richer deployments will soon be possible as the infrastructure matures.
- ▶ Applications and deployments are interoperable across wide range of configurations and evolution of capabilities.