



Information Technology Industry Council
Leading Policy for the Innovation Economy

July 22, 2011

Ms. Marlene Dortch
Secretary
Federal Communications Commission
445 Twelfth Street, S.W.
Washington, D.C. 20554

Re: Ex parte report regarding meeting on CG No. 10-213, WT No. 96-198, CG No. 10-145

Dear Ms. Dortch:

On July 20, 2011, I once again had the pleasure of leading a delegation of representatives from members of the Information Technology Industry Council (“ITI”) to the offices of the Federal Communications Commission (hereafter, “the Commission”), to discuss with key staff issues of importance relative to the Commission’s Notice of Proposed Rulemaking implementing the Twenty-First Century Communications and Video Accessibility Act of 2010.

The ITI delegation consisted of Ms. PJ Edington of IBM; Ms. Paula Boyd of Microsoft; Mr. Tom Wlodkowski and Mr. Chris Blouch of AOL; Ms. Lisa Lindgren of Adobe Systems; Mr. Jim Morgan of Sony; and myself. Participating by phone were Mr. Alex Li of Microsoft, and Mr. Daniel Tang of Cisco Systems.

Representing the Commission were Ms. Elizabeth Lyle, Mr. Brian Regan and Mr. Jeffrey Tignor of the Wireless Telecommunications Bureau; Ms. Karen Strauss, Ms. Rosaline Crawford, Mr. Jamal Mazrui and Mr. Eliot Greenwald of the Consumer and Governmental Affairs Bureau; Ms. Janet Sievert of the Enforcement Bureau; Mr. Craig Elin of the Office of the Managing Director; and Mr. Walter Johnston of the Office of Engineering and Technology. Two FCC interns also

attended: Mr. Doug Brake and Mr. Vijay Pattisapu, who are working with the Wireless Telecommunications Bureau.

Consistent with comments that ITI filed on April 25, 2011¹ and July 8, 2011,² the parties discussed the complexity of the product “stack,” i.e., the various components necessary to provide value and functionality, including accessibility, and the advisability of focusing manufacturer and service provider accountability on components and services within their control.

At this session, the ITI team provided demonstrations that emphasized the importance of standards in enabling advanced communications services (“ACS”) via interoperability between diverse components in the “stack.” For example, the W3C/WAI Accessible Rich Internet Applications, commonly referred to as WAI-ARIA,³ has gained significant traction in the marketplace by facilitating the smooth interaction between the various product components, which are often provided by diverse suppliers and developers. Three examples were highlighted, utilizing different assistive technologies (in this case, screen readers that were either built in, available at a nominal cost (i.e., free) via download, or as a commercial offering for purchase), different web browsers and different operating systems. All three combinations produced a common accessibility experience on the same notebook computer due to the manufacturers’ implementation of the WAI-ARIA standard. We also noted that HTML5 will incorporate some of the same interoperability features currently enabled via the WAI-ARIA standard.

There was a discussion regarding whether the adoption and effective implementation of standards such as WAI-ARIA that facilitate interoperability between information and communications technology (“ICT”) and assistive technology (“AT”) devices could be deemed a

¹ See Comments of the Information Technology Industry Council in response to the Federal Communications Commission Notice of Proposed Rulemaking on Advanced Communications Services, GH Docket No. 10-145 (<http://fjallfoss.fcc.gov/ecfs/comment/view.action?id=6016478346>), April 25, 2011.

² See *Notice of Ex Parte*, Comments of Ken J. Salaets (Information Technology Industry Council) in response to the Federal Communications Commission Notice of Proposed Rulemaking on Advanced Communications Services, GH Docket No. 10-145 (<http://fjallfoss.fcc.gov/ecfs/comment/view.action?id=6016826922>), July 8, 2011.

³ For detailed information regarding WAI-ARIA, see <http://www.w3.org/WAI/intro/aria>.

“safe harbor.” We also discussed whether the Commission could frame the eventual regulation in such a way so as to encourage reliance on such standards. This led to questions about how the Commission could identify and delineate such standards – indeed, whether it even has the statutory authority to do so – and, as a corollary, whether it would be more appropriate to place the emphasis on standards development bodies rather than on individual standards. ITI participants reiterated a previous suggestion that the Commission consider facilitating the development of an advisory group consisting of representatives from consumer, ICT and AT organizations or businesses, to review and provide the Commission with advice on an ongoing basis regarding emerging standards in the marketplace that facilitate accessibility. Such an organization might even fit under ongoing Commission initiatives and thereby include participants from various agency bureaus. There was general agreement that the idea was worth exploring.

As in previous sessions, ITI also reiterated its views on the importance of limiting the definition of “existing” peripheral devices and customer premises equipment to products that are currently being sold, pointing out that most legacy devices and software that are no longer sold often are not “serviced” by manufacturers and developers (e.g., via product and firmware updates), and typically cannot take advantage of the WAI-ARIA standard and similar innovations that are or will be available in the marketplace. It would be extremely expensive if not impossible to in effect retrofit such systems so that they can utilize new and emerging technologies.

ITI also returned to the issue of the statutory exemption for enterprise software, due to its highly customizable and configurable nature. Although a similar enterprise product may be sold to multiple customers, how that product is configured within a given environment (e.g., existing hardware, software, databases, etc.) varies with every customer implementation, whether it is a Federal agency, a corporation or even a small business or school district. Such variability introduces myriad unknowns that can impact the accessibility features built into enterprise products. The manufacturer of such products cannot possibly anticipate and address every possible scenario. ITI reiterated its support for Congress’ decision to exempt such products from the requirements of the Accessibility Act.

Ms. Marlene Dortch
July 22, 2011
Page 4

This concludes my report on the July 20 *ex parte* meeting between the aforementioned Commission staff and the ITI delegation. I am including a set of slides that were used during the aforementioned demonstrations. We welcome any inquiries regarding the views expressed herein. Please direct any questions to the undersigned.

Respectfully,

Ken J. Salaets
Director
Information Technology Industry Council
202.737.8888

ATTACHMENT

Ex parte meeting on

CG No. 10-213

WT No. 96-198

CG No. 10-145

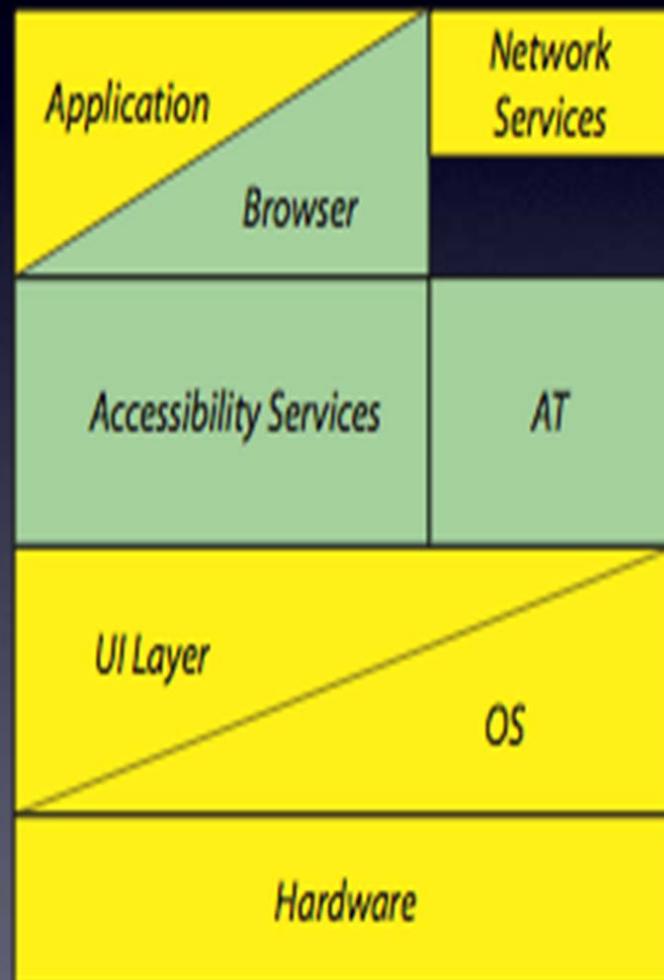


Information Technology Industry Council

Leading Policy for the Innovation Economy

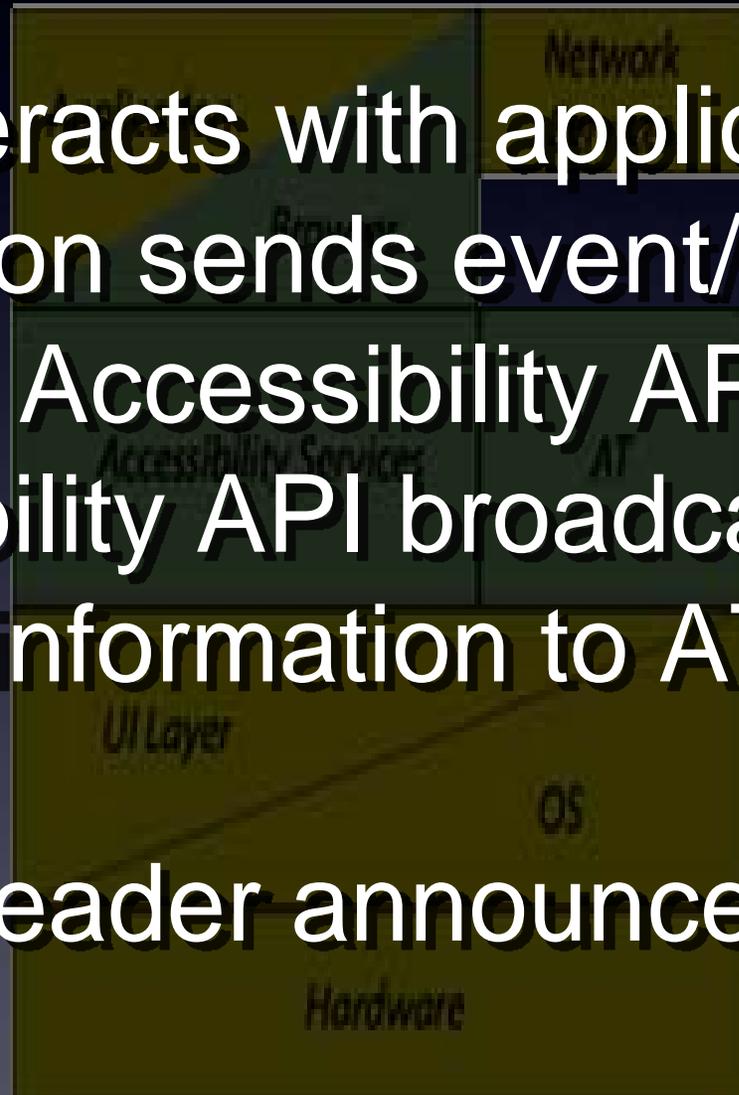
July 20, 2011

Layers



Layers

1. User interacts with application
2. Application sends event/state changes to Accessibility API
3. Accessibility API broadcasts contextual information to AT (screen reader)
4. Screen reader announces information



Focus on Web Applications

Define some common terminology

HTML

Markup which puts content and controls into a hierarchy of objects such as lists, paragraphs or even generic boxes - the DOM

CSS

Rules which apply styles
such as color, position or
sizing to objects

JavaScript

Methods or functions which react to user input and change the HTML or CSS

jQueryUI

jQuery is a JavaScript library which makes it easier for developers to implement rich internet applications

jQueryUI

Smooths browser differences and minimizes code required to build functionality

jQueryUI

Changes this:

```
<div id="slider-range"></div>
```

jQueryUI

Into this:

```
<div id="slider-range" class="ui-slider ui-slider-horizontal ui-widget ui-  
widget-content ui-corner-all">  
  <div class="ui-slider-range ui-widget-header" style="left: 15%; width:  
45%; "></div>  
  <a href="#" class="ui-slider-handle ui-state-default ui-corner-all"  
style="left: 15%; "></a>  
  <a href="#" class="ui-slider-handle ui-state-default ui-corner-all"  
style="left: 60%; "></a>  
</div>
```

jQueryUI

Rendering as:

Price range:

\$75 **\$300**

A jQueryUI price range slider widget. It consists of a horizontal track with a light gray background and a white border. Two gray square handles are positioned on the track, one at the left end and one at the right end. The text "\$75" is displayed in orange above the left handle, and "\$300" is displayed in orange above the right handle. The track is currently empty, indicating no selection has been made.

Great, but...

The generated markup
lacks any semantics or hints
to tell the screen reader
what it is or what state
things are in

Enter ARIA

Web standard which adds attributes to HTML giving meaning to piles of markup

ARIA

```
<div id="slider-range" class="ui-slider ui-slider-horizontal ui-widget ui-widget-content ui-corner-all">
```

```
  <div class="ui-slider-range ui-widget-header" style="left: 15%; width: 45%; "></div>
```

```
  <a href="#" class="ui-slider-handle ui-state-default ui-corner-all" aria-valuenow="75" aria-valuetext="75 $" title="price range min" role="slider" aria-valuemin="0" aria-valuemax="300" style="left: 15%; "></a>
```

```
  <a href="#" class="ui-slider-handle ui-state-default ui-corner-all" aria-valuenow="300" aria-valuetext="300 $" title="price range max" role="slider" aria-valuemin="75" aria-valuemax="500" style="left: 60%; "></a>
```

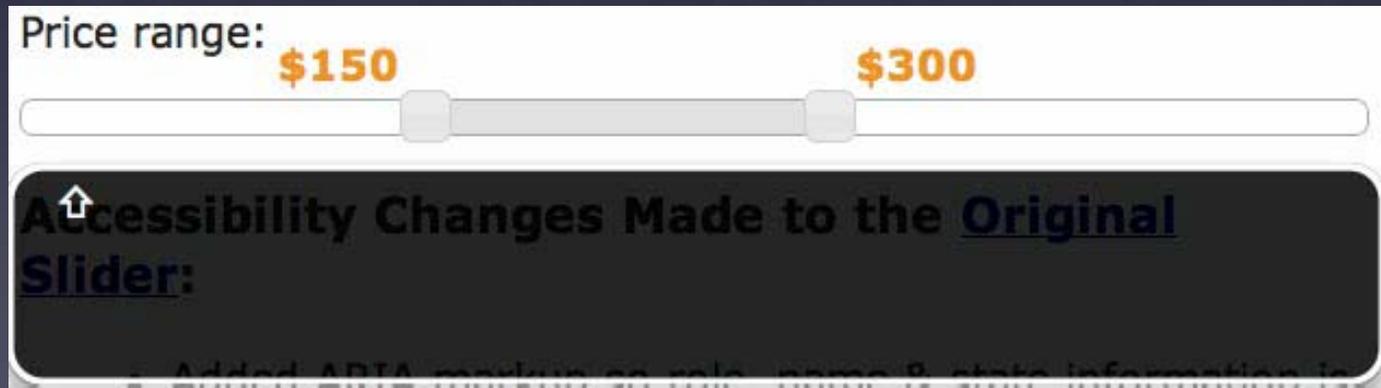
```
</div>
```

Results

Mac OSX

Safari

VoiceOver Screen Reader



Results

Windows XP

Firefox

NVDA Screen Reader

Price range:

\$150

\$300



Results

Windows 7

Internet Explorer

Jaws Screen Reader

Price range:

\$150

\$300



Results

Because of web standards like ARIA, CSS, HTML and JavaScript we get functionally identical behavior over multiple OSes, browsers and screen readers

Under The Hood

How exactly did this all work?

Under The Hood

1. Javascript hooked to the keydown event connected to the slider thumb.
2. Arrow key pressed
3. Javascript updates CSS to reposition slider thumb.

Under The Hood

4. Javascript updates aria-valuenow and aria-valuetext attributes

5. Browser passes changed aria attributes to accessibility API

Under The Hood

6. Screen reader receives events from accessibility API and generates text equivalent

Inspection (win)

AVIEWER

Select Display

MSAA ARIA HTML IA2

Interface	Value
MSAA	
-Name	price_range_min
-Value	150 \$
-Role	slider
-State	floating , focusable , focused
-Description	(none)
IAccessible2	
-Name	price_range_min
-Value	150 \$
-Role	slider
-State	Horizontal, Opaque
-Description	(none)
-Relation Type	(none)
ARIA	
-Role	slider
Attributes	
-aria-valuenow	150
-aria-valuetext	150 \$
-aria-valuemin	0
-aria-valuemax	300

AVIEWER

Select Display

MSAA ARIA HTML IA2

Interface	Value
MSAA	
-Name	price_range_min
-Value	151 \$
-Role	slider
-State	floating , focusable , focused
-Description	(none)
IAccessible2	
-Name	price_range_min
-Value	151 \$
-Role	slider
-State	Horizontal, Opaque
-Description	(none)
-Relation Type	(none)
ARIA	
-Role	slider
Attributes	
-aria-valuenow	151
-aria-valuetext	151 \$
-aria-valuemin	0
-aria-valuemax	300

Inspection (osx)

<AXSlider: "price range min">

Attributes:

AXRole: "AXSlider"

AXRoleDescription: "slider"

AXHelp: "price range min"

AXTitle: "price range min"

AXValue (W): "75"

AXFocused (W): "true"

AXEnabled: "true"

AXMinValue: "0"

AXMaxValue: "300"

AXOrientation: "AXHorizontalOrientation"

AXValueDescription: "75 \$"

<AXSlider: "price range min">

Attributes:

AXRole: "AXSlider"

AXRoleDescription: "slider"

AXHelp: "price range min"

AXTitle: "price range min"

AXValue (W): "76"

AXFocused (W): "true"

AXEnabled: "true"

AXMinValue: "0"

AXMaxValue: "300"

AXOrientation: "AXHorizontalOrientation"

AXValueDescription: "76 \$"

Desktop

Using the free open source NVDA screen reader we can interact with an existing desktop application that interfaces with windows accessibility APIs

Desktop



Questions?