



GateIn Mobile: Device Detection and Redirection

**Matt Wringe
15/03/2012**

Agenda

- Brief Mobile Overview
- Site Detection
- Site Redirection and Mirroring
- Future Mobile Work and the Other Mobile Specifications



GateIn Mobile Specs






- Main Mobile Spec Page:

community.jboss.org/wiki/GateInMobileSpecifications

- Contains links to the various other mobile specs (in varying degrees of completeness)
 - Device Detection
 - Site Redirection
 - Mobile Look and Feel
 - Mobile Portlet Development
 -



GateIn Mobile: Introduction

- Only considering modern 'mobile' devices
 - Standard HTML/CSS/JS    
 - ~~WAP/WML/....~~ 
- 'Mobile' to include phones and/or tablets
 - 'device specific configurations' more accurate
 - Mobile devices are starting to surpass traditional desktops in some areas (resolution, power, ...)
 - Traditional desktops are become more like mobile devices (touch screen, app stores, restrictions, ...)



GateIn Mobile: Introduction

- Phones and Tablets can be very different or similar
 - ~3 inch screen phone vs. 10 inch tablet
 - ~5 inch screen phone vs 7 inch tablet
- Should a tablet display a desktop site or a phone specific site? A tablet specific site?
 - Not always clear and depends on the site
- Rapidly moving space, lots of changes happening



GateIn Mobile

- We need to provide a default configuration that works well across different mobile devices
- But, the admin needs to be able to make their own decisions based on their own sites capabilities (custom skins, portlets, ...)
- We need to provide a highly configurable mobile solution with replaceable and extensible services
- Initial Approach
 - Separate mobile site
 - Redirects will be between sites on the same server



Device Detection



Device Detection

- Traditional Approaches
 - User Agent Strings
 - CSS Media Queries and Javascript Detection
- GateIn Device Detection
 - Approach
 - Configuration



Device Detection:User Agent Strings

- HTTP header (optional, user modifiable)
- Specifies the browser information
- String, complex format convention
- New user agent strings are constantly being produced (new devices, new browsers, ...)
- But:
 - Quick and easy to deal with (forwards based on header)
 - Works very well in some situations
- Third party solutions/services that try to handle detection based on the user agent (eg WURFL,...)



Device Detection: User Agent Strings

- User-Agent: Mozilla/5.0 (Linux; U; Android 2.3.5; fr-fr; **Nexus S** Build/GRK39F) AppleWebKit/533.1 (KTML, like Gecko) Version/4.0 **Mobile** Safari/533.1
- User-Agent: Mozilla/5.0 (Linux; U; Android 4.0.3; fr-ca; **Transformer Prime TF201** Build/IML74K) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Safari/534.30
- User-Agent: Mozilla/5.0 (Android; Linux armv71; rv:10.0.1) Gecko/20120208 Firefox/10.0.1 Fennec/10.0.1



Device Detection: Media Queries and Javascript

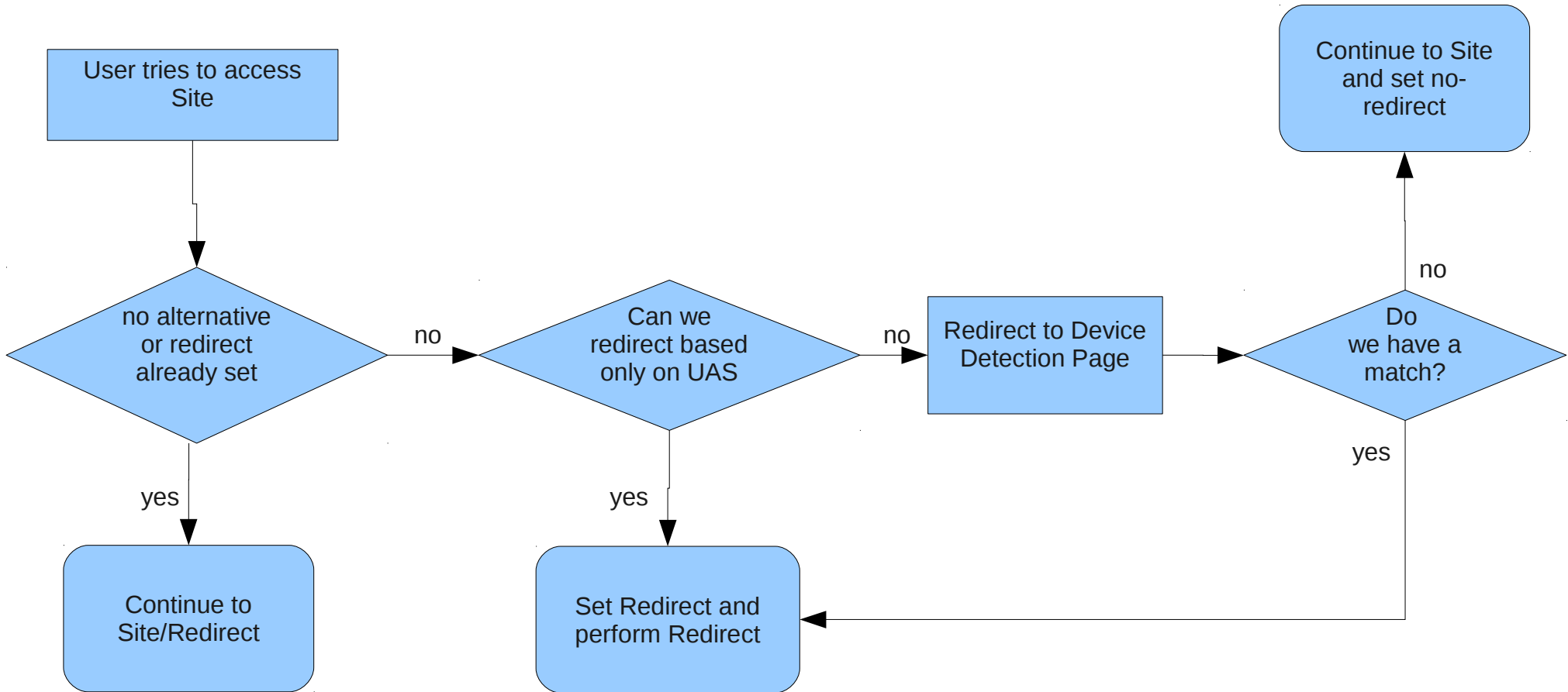
- CSS3 Media Queries
 - Different style sheets based on things like device screen size, resolution, pixel density, ...
- JavaScript
 - Device property detection and page modification
- Can retrieve the browser information without having to know beforehand anything about the browser.
- But:
 - Is determined when the page is loaded, not before



GateIn Device Detection

- Admin can specify multiple conditions for when a particular site should be performed:
 - Only user agent string based
 - Only device property based
 - Hybrid approach of both





```
<device-configuration>
  <device-redirects>
    <device-redirect>
      <origin-site>classic</origin-site>
      <redirects>
        <redirect>
          <redirect-site>test</redirect-site>
          <conditions>
            <condition>
              <user-agent>
                <contains>mozilla</contains>
              </user-agent>
            </condition>
            <condition>
              <device-property>
                <property-name>device.width</property-name>
                <less-than>1000</less-than>
              </device-property>
              <user-agent>
                <does-not-contain>foobar</does-not-contain>
              ...
```



Device Detection: User Agent XML Config

- Simple string based matches:
 - contains
 - Does-not-contain
- Pattern/RegEx based matches: matches

```
<conditions>  
  <condition>  
    <user-agent>  
      <contains>mozilla</contains>  
    </user-agent>  
  </condition>  
  <condition>  
    <user-agent>  
      <does-not-contain>KHTML</does-not-contain>  
      <matches>.*Chrome.*</matches>  
    </user-agent>  
  </condition>  
</conditions>
```



Device Detection: Device Property XML Config

- Device-name
- Comparators:
 - less-than, greater-than, equals, matches

```
...  
<conditions>  
  <condition>  
    <device-property>  
      <property-name>device.width</property-name>  
      <less-than>1200</less-than>  
      <greater-than>800</greater-than>  
    <user-agent>  
  </condition>  
</conditions>  
...
```



Device Properties: Device.jsp

- Device detection and re-re-direction page
- User configurable to add their own customizable device properties.
 - Just need to add custom javascript and use the `addParameter(name, value)` js method.
 - Free to check for whatever they want
 - Device properties, HTML5 capabilities,
 - Free to use JS detection libraries (like Modernizr)



Device.JSP device-detection.xml DEMO



GateIn Device Detection: TODO

- Still a bit in the prototype phase and in flux as more of the features are actually implemented
- Specifications needs to be updated
- Missing Features:
 - UI configuration & persistent storage to jcr
 - Will post UI mockups to the specifications page
 - More generic testing
 - Automated tests :(
 - Default 'mobile' setup which works well across devices



Site Redirection



GateIn Mobile: Site Redirection

- Initial implementation is only going to handle redirecting between sites on the same portal server
 - Should be able to easily add generic URL redirects
 - Redirects to an external GateIn servers will require some sort of service to check if nodes exists or not.
 - Not sure if this is exactly something we really want to support right now



Site Redirects: Configuration

- Same configuration file as the device detection
- Mapping Types:
 - Explicit mapping
 - Node name mapping
 - If Unknown Node:
 - Common ancestor node name mapping
 - Redirect
 - No-Redirect
 - Redirect to Root path



```
<device-redirect>
  <origin-site>classic</origin-site>
  <redirects>
    <redirect>
      <redirect-site>test</redirect-site>
      <conditions>
        ...
      </conditions>
      <mappings>
        <use-node-name-matching>true</use-node-name-matching>
        <map>
          <origin-path>/sitemap</origin-path>
          <redirect-path>/test-sitemap</redirect-path>
        </map>
        <map>
          <origin-path>/foo</origin-path>
          <redirect-path>/mobile/foo</origin-path>
        </map>
        <unknown-node-mapping>no-redirect</unknown-node-mapping>
      </mappings>
    </redirect>
  </redirects>
</device-redirect>
```



Site Redirects: Explicit Mapping

- Specified mapping to use between a path on the origin site and the redirect site
- Takes priority over other mapping types

```
<map>  
  <origin-path>/sitemap</origin-path>  
  <redirect-path>/test-sitemap</redirect-path>  
</map>
```



Site Redirects: Node Name Matching

- True if we want to map nodes based on node names.
 - If the nodes are called the same on both servers, then they will automatically map
- False if we only want to consider explicitly mapped nodes

<use-node-name-matching>true</use-node-name-matching>



Site Redirect: Unknown Node

- If no match from explicit map and node name matching
- Types
 - Common Ancestor Node Matching
 - Try and matches names of parent nodes
 - Redirect
 - Try and redirect anyways based on the url
 - NoRedirect
 - If no match, continue to the original requested site
 - Redirect to Root
 - Redirects to the default node of the redirect site



Device-detection.xml DEMO



GateIn Site Redirection: TODO (This looks familiar...)

- Still a bit in the prototype....
- Specifications needs to be updated....
- Missing Features
 - UI configuration & persistent storage to jcr
 - Will post UI mockups to the specifications page
 - Automated Tests
 - Support for generic URL redirects
 - Support for redirects to external portals and sites
 - Requires a node service to do matching
 - Could just use the external URL redirect method, but lose node matching....



GateIn Redirect: Manual Redirection Portlets

- Need a way for the user easy move between the various sites
 - User preference
 - Device detection misses
- Still in TODO state



Manual Redirection: Header Redirection Portlet

- User css media queries/javascript to detect if the user is accessing the proper site or not
- If (and only if) the user is accessing the site where an alternative, device specific site exists, then display the portlet
- Link to the other device specific site
- Option to not display again



Imagine Cool Mock-Up Here
(hey, its still a work in progress)



Manual Redirection: Footer Site Selector

- New footer with links to the various other sites
- Always displayed regardless of what device is accessing the page



GateIn Mobile: Future Work

- Main Mobile Spec Page:
community.jboss.org/wiki/GateInMobileSpecifications
- Mobile Look and Feel
 - Improvements needed in skinning service
 - 'responsive' web design, ...
- Mobile Portlet Development
- Future 'mobile' enhancements
 - Gestures, etc...
-





Discussions

