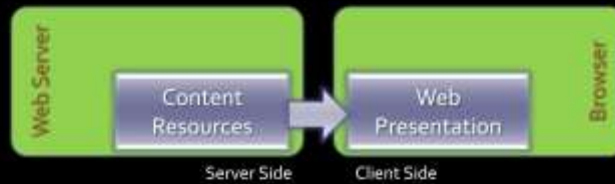




"Will the Web work on my device?" is an increasingly challenging question. Given the growing diversity of Web clients, from mobile devices to billboards, it's apparent that Web content and services must adapt to new environments. Real-time access to device descriptions will enable future adaptive Web servers to meet the challenge. In this short presentation, the W3C's Device Description Working Group (DDWG) chair, Dr Rotan Hanrahan, explains.

At the time of this presentation, Dr Hanrahan is also a member of the Mobile Web Initiative Steering Council, a member of the W3C Advisory Committee (representing MobileAware), a member of the Ubiquitous Web Applications Working Group and a member of the OpenAjax Alliance Mobile Taskforce.

How it should be?

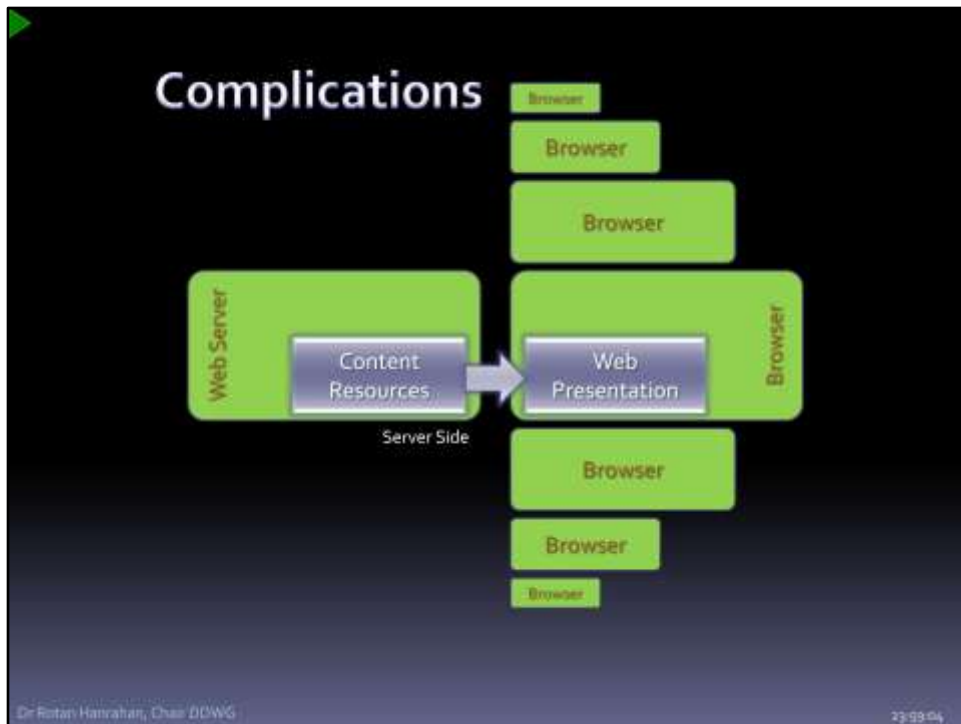


ONE WEB

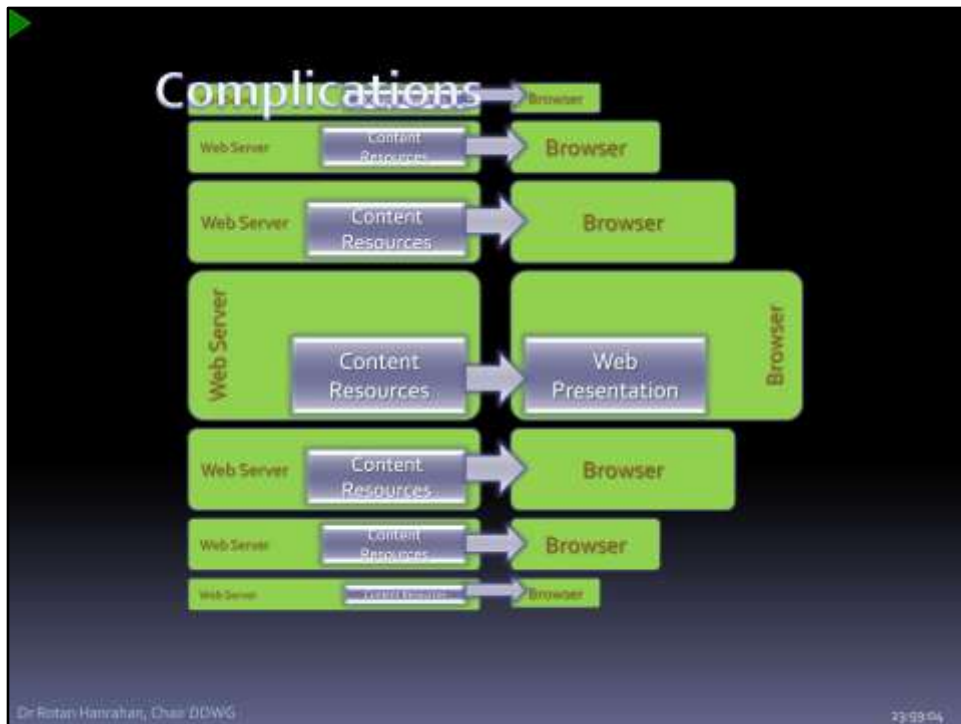
Dr. Ratan Hanrahan, Chair DDWG

23/09/07

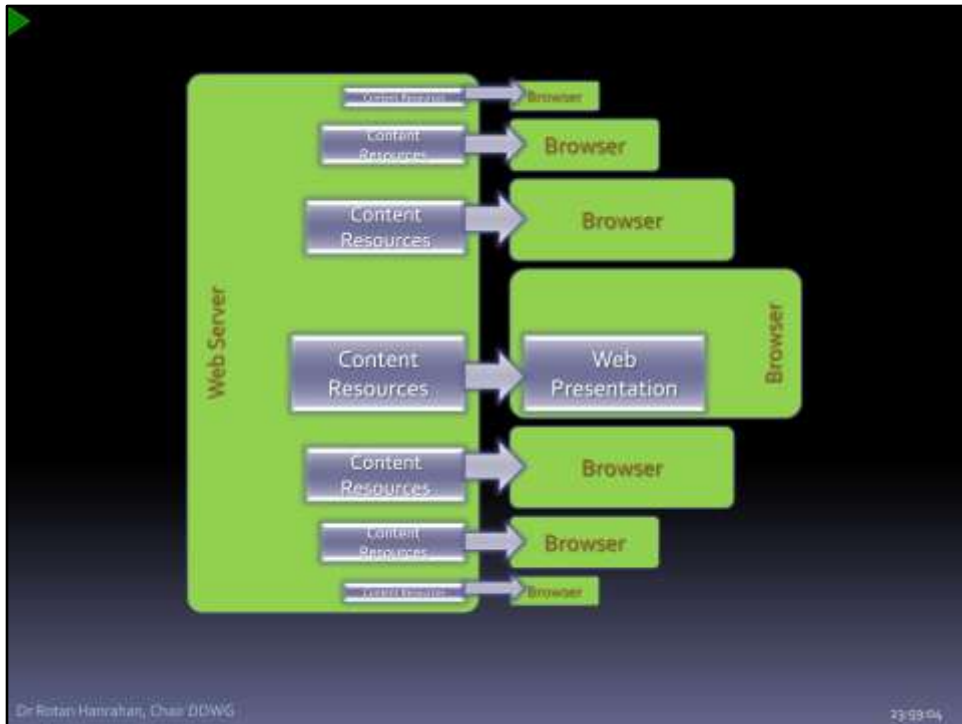
The idea of setting up a single Web server with a single version of content for delivery to a single version of browser on a common device is a developer's Utopia. This situation only existed on the Web for a very short period after Tim Berners-Lee invented it. From the day that a second browser was developed, this Utopia vanished. Forever.



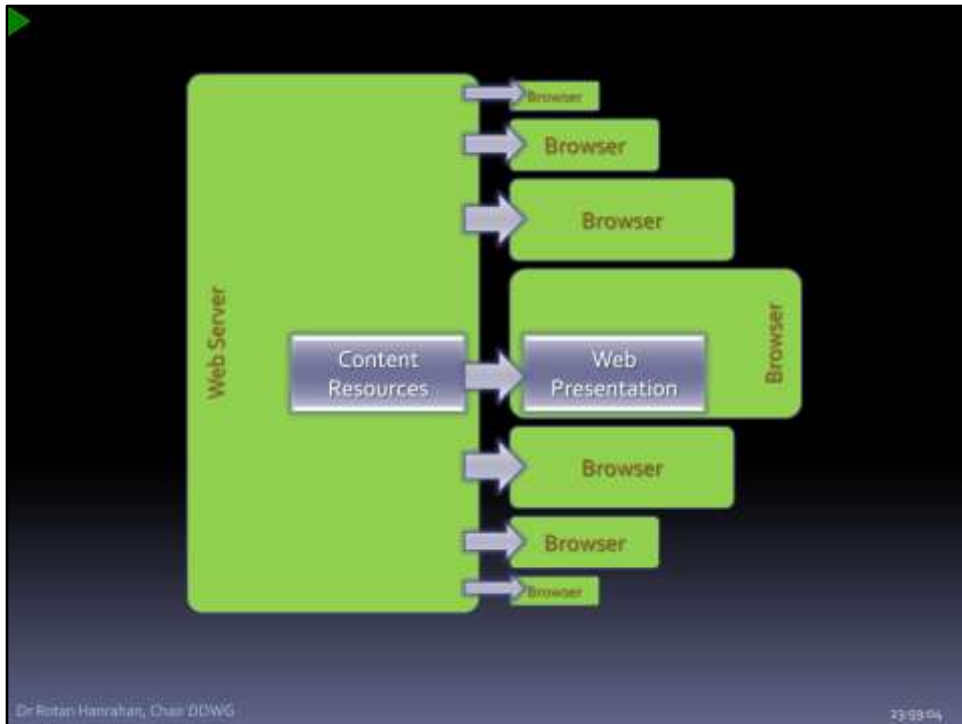
The reality is that there are now many browsers, of many different sizes and capabilities, running on an ever-increasing variety of platforms. The diverse demands of end-users has created a market for diverse browsers and devices. A single Web server will only be able to deliver successfully to a subset of these clients. If the presentation resources (comprising markup and images and other resources) on the server do not match the features of the client, the delivery will fail. Possibly in spectacular ways.



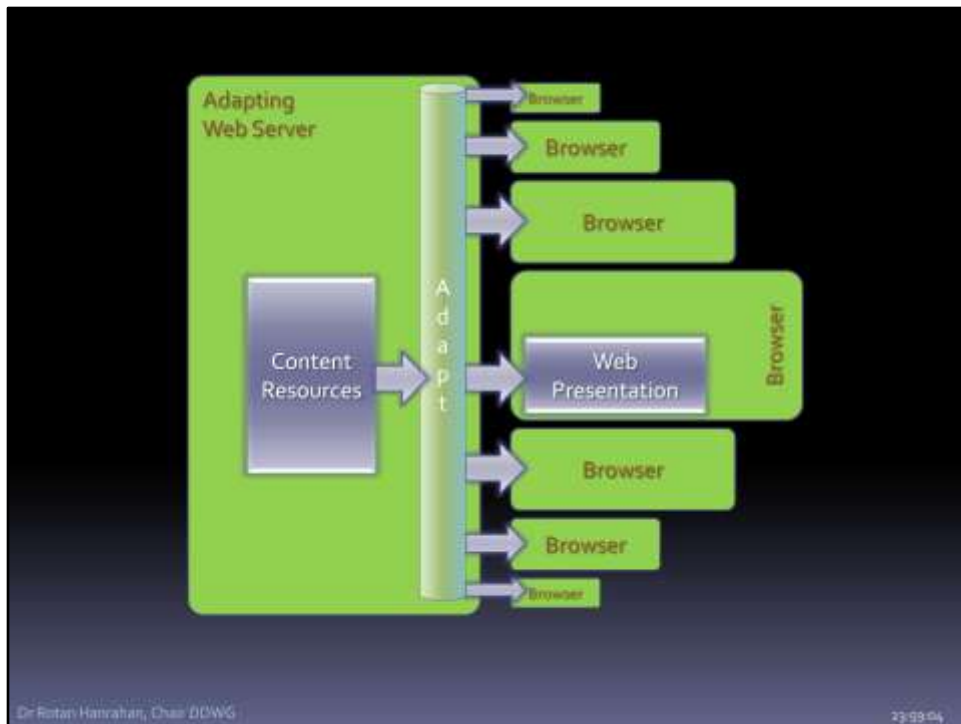
One might think that having a separate server for the mobile phones would be useful. Then perhaps you add one for the PDAs. And one for the hybrid devices. And the smartphones. And the flipphones. And so on, until you have many different servers for the different channels. You now have a major problem managing all these systems behind your Web presence, which itself is confusing because you have to have separate entry points (URLs) for each type of device.



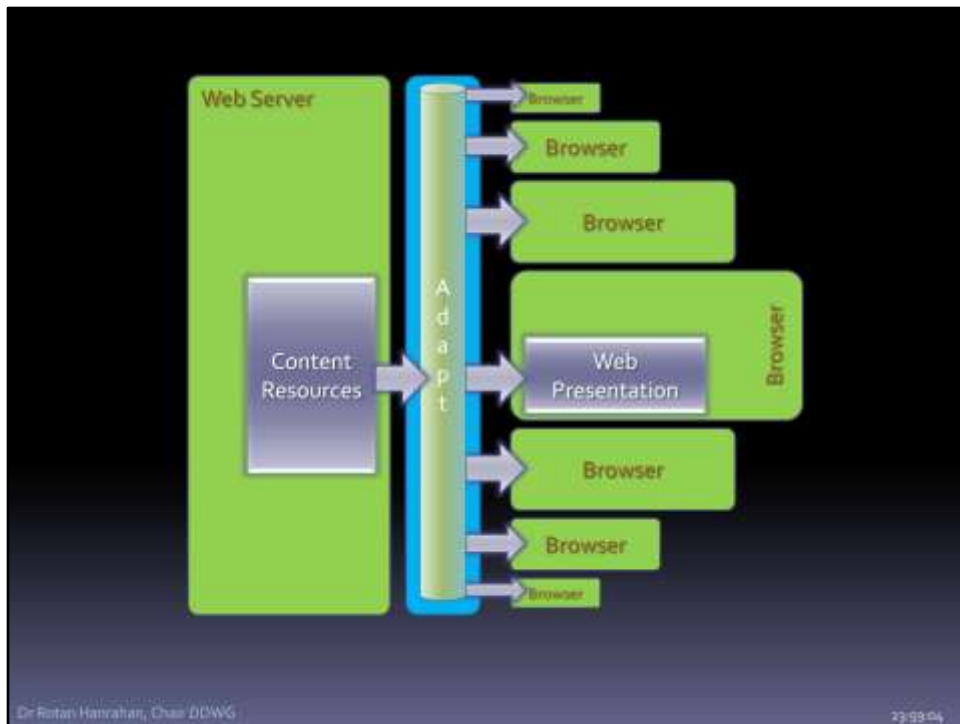
To avoid the multiple server problem, you could consolidate into a single server that supports multiple hosts. This does not alleviate the management problem for the many sites on this server, and the many forms of content that each site requires. Additionally, you have still not addressed the problem of multiple entry points, which is seriously diluting your Web presence.



It is better that all of these delivery channels can tap into the same set of content resources. This avoids much of the unnecessary duplication, and the danger of diverging sites, which would compromise the stability of your brand.



The ideal situation is a solution that has a single set of content resources and can service all devices via a single entry point, giving your site a proper focus and brand stability. This solution involves an Adapting Server. You control both the content and the adaptation process, so you can ensure that your mobile presence is exactly the way you intend it to be.



Dr Rofan Hamzah, Chair DDWG

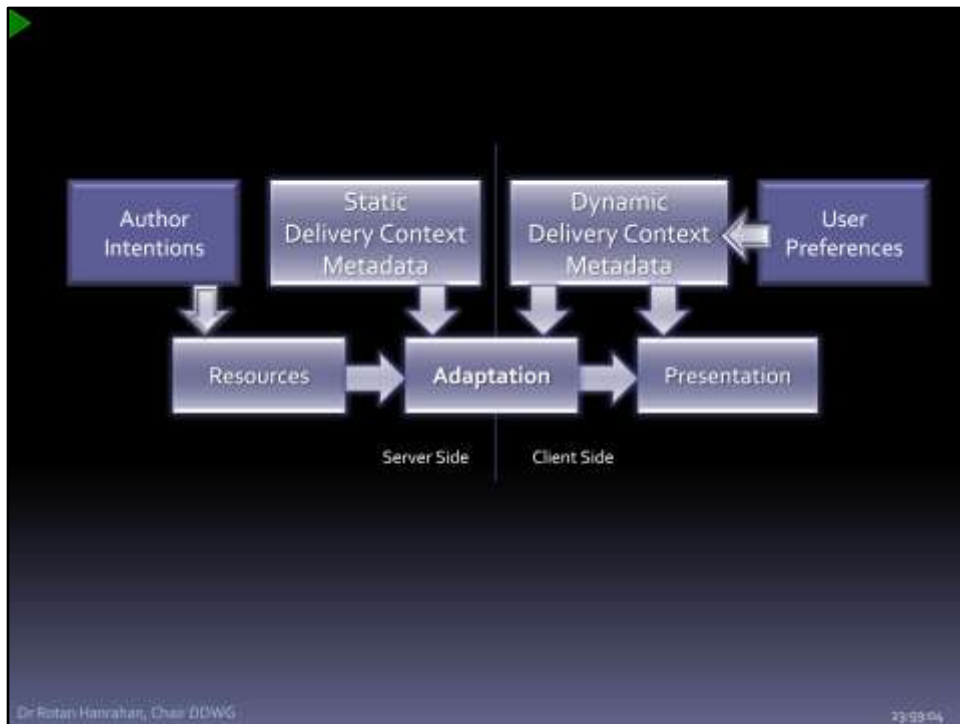
23/03/04

An alternative architecture is one where the adaptation process is separate from the content server. If the adaptation process and the content are under your control, then this architecture can provide similar results to the combined adapting server. Such an approach can be useful where the adaptation technology is significantly different from your existing Web server technology. This approach is also known as the Proxy Model.

Unfortunately, if the adaptation process is inserted into the delivery chain in a manner where neither the content author (the main Web server) or the client have any substantial influence over the behaviour of the adaptation process, the results can be far from pleasing. Nevertheless, this approach has been seen to be somewhat successful when the input content is from legacy sources (i.e. the “Long Tail”) that do not have, and never will have, any guidance to offer to the adaptation process.

The worst case scenario is where the intermediate adaptation proxy masquerades as a desktop browser in order to retrieve content from the origin server, which itself is an adaptation server. In this case the adaptation server is fooled into delivering desktop content instead of content adapted for the real client. The intermediate proxy then treats the desktop content like any other Long Tail source, and produces a result that is far less suitable. Furthermore, the adapting server may be doing more than merely adapting pages; it may in fact be adapting the application workflow. The introduction of an uncooperative intermediate adapting proxy will destroy many opportunities to deliver optimal end-user experiences.

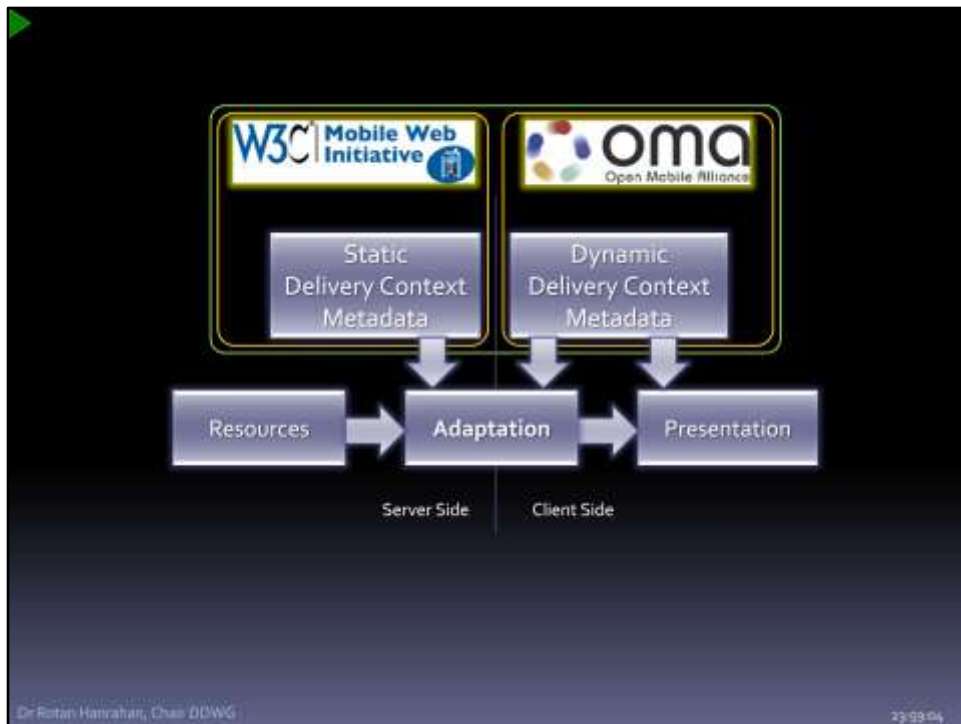
Efforts are ongoing to ensure that the community of proxy providers can work in cooperation with adapting and non-adapting sites to avoid these negative scenarios.



Adaptation is a complex process. The author provides the raw resources (markup, images, workflow, scripts etc.), which are then selected, transformed and transcoded according to whatever is known about the client. This information is usually obtained from the metadata in the request that came from the client. It is possible that other metadata is available, such as the user's preferences or dynamic states of the device (e.g. sound is turned off).

As the author is not going to be available at the time of adaptation, the intentions of the author must be captured in advance. For example, a page that is intended to convey a particular message to the user should have that message unambiguously identified, so that an adaptation process will know that this piece of information is not to be removed if the page content is being "pruned". The author can also be given the power to decide, based on discovered conditions, what content is to be used, and what content is not to be used.

Similarly, the end user can contribute to the metadata. For example, a device that is capable of displaying high resolution pictures might be presented with monochrome because that is the user's preference. (That preference could be related to the cost of delivering high resolution images, or perhaps some eyesight constraint.)



Metadata is the key to successful adaptation. This is what motivates the W3C and the OMA to create common standards to manage and convey device metadata. The W3C's Mobile Web Initiative has created the Device Description Working Group to define the technology that will enable static device information to be accessed by adaptation systems. Similarly the OMA is defining technologies to access dynamic device information. Both are contributing to a common ontology (a formal set of definitions) being managed by the W3C's Ubiquitous Web Applications group, to ensure that the various technologies can work in harmony.



It's a complex problem...

- Portrait or Landscape?
- Do you know if rotation is possible?
- Do you know the current orientation?
- What happens if the screen rotates?
- Do you send a new version of the page?
- Can you adapt the page on the client?
- How do you discover the screen size?

Today, even this last question is hard to answer.

© R. Raman, OMA, 2004

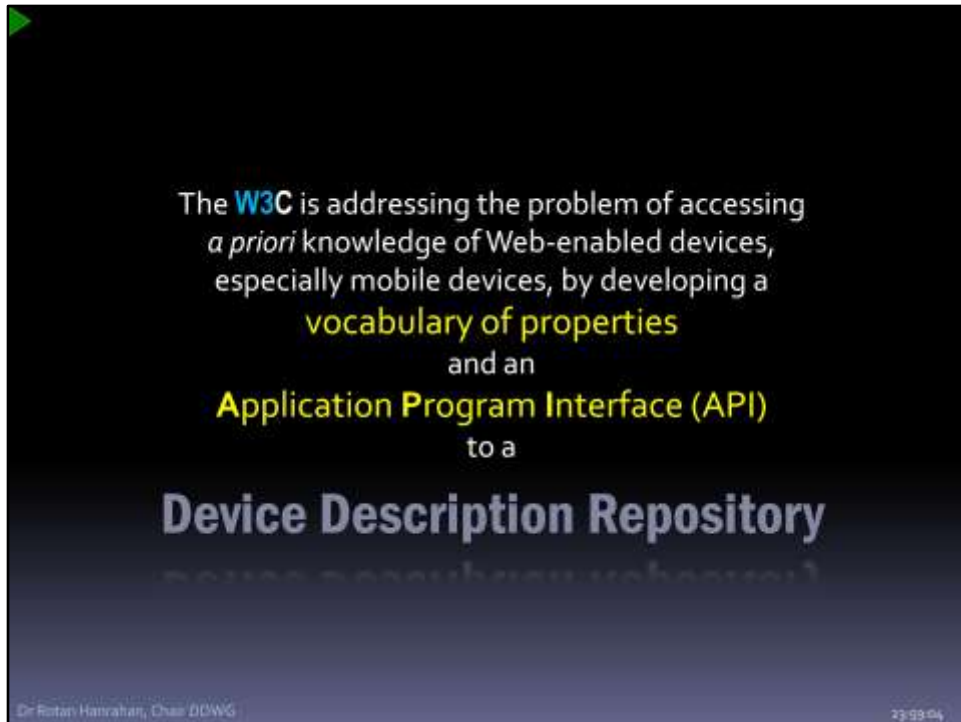
The relationship between static and dynamic device information is best explained through an illustration. The illustration will also show that the split into static and dynamic is not as clear as one might think. The various groups are aware of these subtleties and are taking them into account.

In this example, the adaptation technology must deal with a device that is capable of displaying a Web page in either a portrait or a landscape mode. The fact that the device is capable of doing this is known in advance. This information does not depend on any particular interaction with the client. It is a fact that can be recorded in advance, and therefore can be held as a static property in a repository. The adaptation technology merely has to be certain about which device is involved, so that it can check the property in the repository.

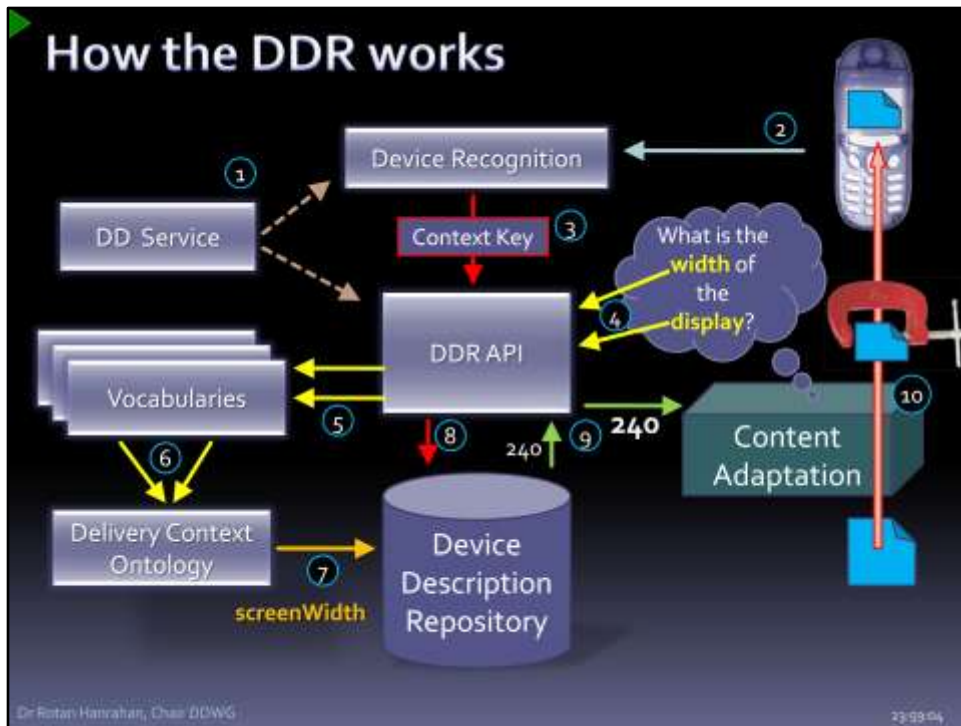
At the point in time when the client makes a request, the screen will be in one of the two possible orientations. The actual orientation at that point in time is not something that can be known in advance. The current orientation is a dynamic property. Identifying the current orientation is something that the OMA will consider.

It is possible for the user to change the orientation during a session. There are many ways in which this event could be processed. The server could be informed, so that a new page can be delivered. Alternatively, a script embedded in the page could handle the event by adjusting some of the layout of the page elements. This kind of dynamic processing can be handled by script, possibly in cooperation with the server, using a design pattern now popularly known as Ajax. Of course, the adapting server needs to know in advance that the device can use Ajax so that the right scripts can be delivered with the page.

All of these different uses of the screen orientation involve different communities. The W3C, OMA and Ajax communities in particular. It would be unfortunate if their various technologies could not interoperate. It would be a bad idea for one group to represent screen orientation as Portrait and Landscape, while another uses Tall and Wide, and yet another uses 278456GF and 6635QK. Instead, a common ontology is being created (under the guidance of the UWA) so that all of the conceptual representations will be compatible.



The Device Description Working Group is part of the W3C's Mobile Web Initiative. The group will not be creating a repository of its own, but will be defining a technology to enable existing and future repositories to interoperate, and to provide a common means for these to be used by adaptation solutions.



[1] A device description service is used to set up a Device Recognition service and an interface to the Device Description Repository (DDR).

[2] A mobile device makes a request for a Web resource (e.g. an image for use as a banner on a page).

[3] The recognition service uses information from the request to create a key to represent the context of the interaction with the mobile device. This key will enable the DDR to identify properties that are appropriate to the context.

[4] The adaptation needs to know the width (a property) of the display (an aspect of the device to which the notion of width can apply).

[5] The DDR Application Program Interface (API) passes the compares the property and aspect against terms it knows about in the one or more vocabularies it understands.

[6] The property and aspect map to a concept that has been recorded by the W3C Ubiquitous Web Applications ontology. In this case, the concept is known to the ontology as "screenWidth".

[7] The screenWidth identifier is passed as a query parameter to the repository.

[8] The context key refines the query so that only the screenWidth that is appropriate to this context is retrieved.

[9] The value "240" is returned, and passed back to the adaptation system as the answer to its query.

[10] The adaptation system uses the value of 240 to adjust the image so that it is suitable as a banner across the width of the device.

This is an extreme simplification of the DDR mechanism. It is presented here to give a very high-level overview of the solution, and to illustrate the role of the different actors.

The first implementations of the DDR are expected to appear during 2008.

All of the technologies developed by the [W3C](#) MWI DDWG are open specifications and are free to be implemented by anyone.

The work of the DDWG can be viewed on the Web and the group has a public mailing list for comments and questions.

<http://www.w3.org>

Thank you for your attention.

Rotan Hanrahan, on behalf of the DDWG.

A link to the DDWG can be found on the W3C home page.