Scalable architectures and services for ubiquitous Web access

Michele Colajanni
University of Modena
colajanni@unimo.it

Riccardo Lancellotti
University of Modena
lancellotti.riccardo@unimo.it

Philip S. Yu
IBM T.J. Watson Research Center
psyu@us.ibm.com

1 Tutorial Overview

The success of the Web in the last decade has caused an evolution of the resources that are disseminated through the Internet. The initial static contents have been enriched by an increasing amount of multimedia and dynamically generated resources. This evolution has shifted the research focus from a nowadays mature content delivery scenario to a Web service generation and delivery scenario. The overall complexity is further increased by the so-called ubiquitous Web access that aims to allow the users to access Web-based services from any location through every class of devices. The key feature of the ubiquitous Web is represented by the content adaptation services that tailor the Web content and the Web-based services to the characteristics of the client devices and to the preferences of the users. This feature of the ubiquitous Web introduces new performance and security problems to the infrastructure that has to generate and disseminate the resources, but it also offers a wide range of novel service opportunities. The tutorial is divided in three parts: the first related to the services for the ubiquitous Web access, the second to the architectures to build scalable services, the third to the presentation of some case studies.

In the first part, we present the adaptation services by differentiating transcoding from personalization services. Transcoding services tailor Web resources to the capabilities of the client and network infrastructure, while personalization requires more sophisticated services that aim to adapt the content to (a combination of) user preferences, locations and behaviors. We also provide a classification of adaptation services, where the classification is based on the information used for the content adaptation service. For the design and deployment of content adaptation architectures the main difference occurs between services relying on persistent data (e.g., user profiles), and services using just volatile information. We discuss the characteristics of each class of content adaptation services providing examples of both research studies and commercial products.

The second part of the tutorial is devoted to the analysis of the architectures for the deployment of content adaptation services with a special focus on system scalability and issues related to data consistency and privacy. We define the design of a content adaptation architecture as a mapping problem. The functions of content adaptation, data management and connection management must be assigned to the nodes composing the infrastructure. We identify different solutions: the *client-based* architecture that places every adaptation function directly on the client device; the *server-based* architecture that follows an opposite approach and utilizes a powerful system to provide ubiquitous Web access to heterogeneous clients; and the *intermediary-based* architecture that places adaptation services on intermediate edge servers that are closer to the clients than to the origin servers. There is a wide space of architectural, coordination and algorithmic options for the nodes of an intermediate infrastructure. We describe and evaluate the architectures having in mind the main requirements that are necessary for supporting most adaptation services, especially scalability, data consistency, identity and location management, information security and user privacy.

In the third part of the tutorial, we present some interesting prototypes and systems for the support of content adaptation services. The presentation criteria is based on two parameters: whether the infrastructure

supports ubiquitous Web access for the whole Web or a limited set of Web sites (*Web scope*) and whether the service is provided to any user or to a subset of registered users (*User scope*). Hence, we identify four classes of content adaptation systems. For each class we describe the typical content adaptation services, the available architectural solutions and discuss some examples.

2 Learning objectives

1. To introduce participants to the content adaptation services and to the infrastructures for supporting ubiquitous Web access.
2. To provide an insight on the architectures available for the deployment of content adaptation systems.
3. To provide significant examples of ubiquitous Web access systems.

3 Content adaptation services

Ubiquitous access to information and services is a broad field of research. In this tutorial, we distinguish two branches of proposal that address this problem:

- *Pervasive computing,* which aims to provide access to *information* and *services* for everyone, at all times and everywhere, as shown in Figure 1. Pervasive computing is focused towards a central role of connectivity among heterogeneous devices and is characterized by a special attention to everyday's tasks and to how these tasks can be carried out in a more efficient way with the help of information technology [37].
- *Ubiquitous Web access,* which is a subset of pervasive computing that focuses on allowing access to services and information from a *Web-based interface*, as shown in Figure 1. In 1997 G. C. Vanderheiden [40] presented access to the Web "Anywhere, anytime, for anyone" as a key challenge for the current Web generation. This definition of next generation Web services is well suited to ubiquitous Web access.

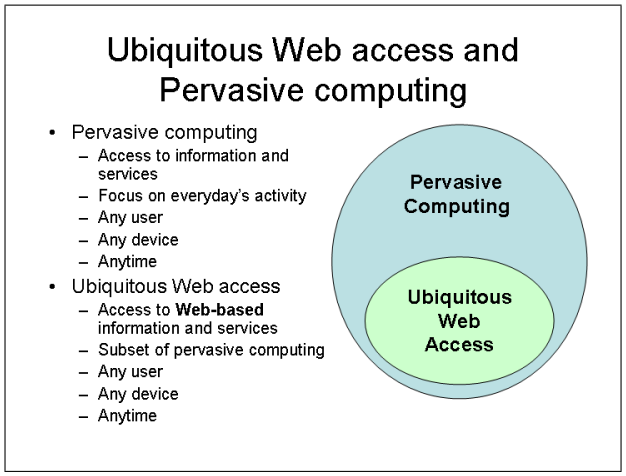

Figure 1: Ubiquitous Web access and pervasive computing

This tutorial will focus on ubiquitous Web access, but we will also provide some insights related to the wider research field of pervasive computing to better explain the highly innovative potential of ubiquitous services.

In the first part of this tutorial, we present an overview of the so called adaptation services that can be deployed to provide ubiquitous Web access.

3.1 Transcoding and personalization

The *adaptation service* term spans various types of functionalities, that for the goal of this tutorial we classify into two main categories based on the implicit/explicit promoter of the request and the consequent adaptation service that is, the user or the client device. Throughout this tutorial we will call:

- *Transcoding*, the service of tailoring Web content to the capabilities of the client device and the network connection.
- *Personalization*, the adaptation of the content to (a combination of) user preferences, locations and behaviors.

The adaptation process accepts an input Web resource (a Web resource may be a single Web object as well as a whole Web page consisting of the HTML container and its embedded objects) and produces a Web resource adapted to the user/client needs. Transcoding and personalization are not mutually exclusive, but we can combine them as shown in the example of Figure 2. In this instance, the adaptation of the Web resource is based on information regarding the user profile, the network status and the client device characteristics.

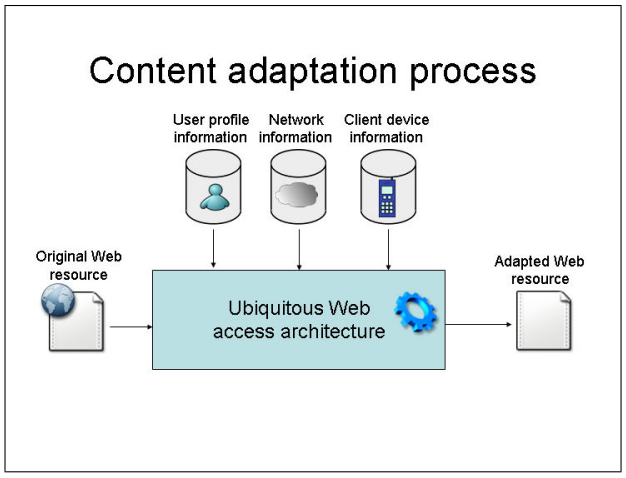

Figure 2: Adaptation Process

3.1.1 Transcoding services

The so-called pervasive computing devices that characterize the ubiquitous Web show an enormous variability in processing power, storage, display, and connectivity capabilities [10]. The main consequence is that Web resources (including video, image, audio and text) may need to be summarized, translated and converted because of two main requirements related to the client platform that is, the features of the physical devices and the characteristics of their connection to Internet. As shown in Figure 3, we distinguish the following transcoding services:

- *Device-oriented transcoding* that is mainly related to the characteristics of the user device interface. The goal is to provide the user with the best quality for content access and Web-based services that are compatible with the physical capabilities of the device (eg., display size and available number of colors).
- *Network-oriented transcoding* that is mainly due to the medium and protocol characteristics of the connections of the devices to the Internet.

It is worth to note that the previous classification is not a partition of the transcoding services, because adaptation based on device capabilities may impact network resource usage and, vice versa, adaptation to network features may impact the Web resource representation. Due to the heterogeneity of devices, services and network protocols, it is mandatory to find techniques and innovative systems that are capable to customize the same multimedia content and/or the Web-based service to different client devices, still preserving its semantic attributes.

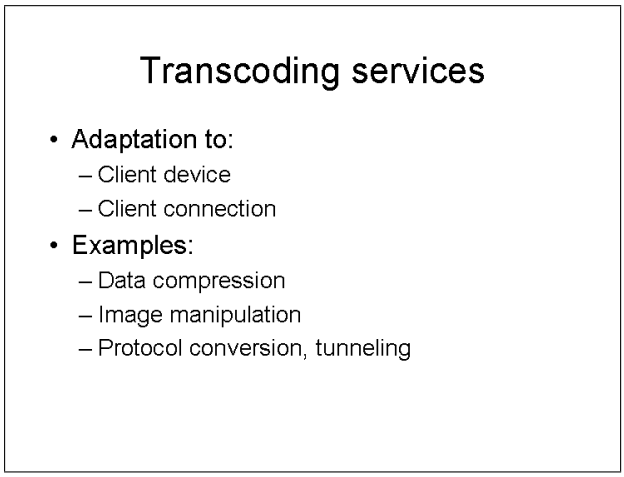

Figure 3: Transcoding services

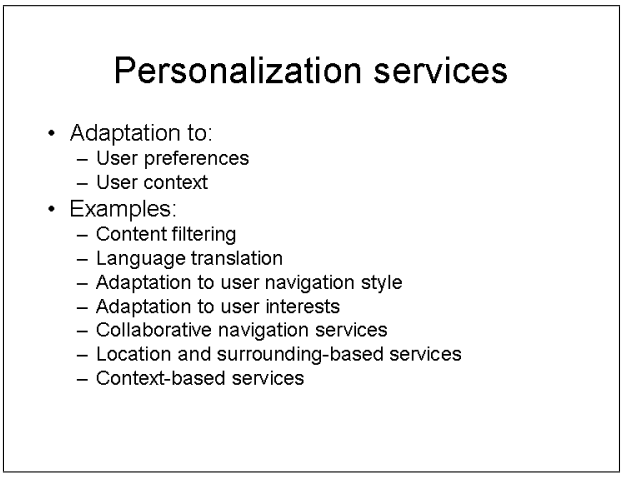

Figure 4: Personalization services

The transcoding tasks are implemented through software applications which are able to filter and convert information. A non-exhaustive set of transcoding services includes:

- *Data compression*. A compression service may reduce download time by reducing the size of transmitted data through algorithms that decrease redundancy. Gzip, Bzip, or Compress algorithms are widely used in Unix systems to reduce disk space utilization and may be easily applied to Web resource transfer [18].
- *Video/Image transcoding*. A video transcoder can resize, change color palette, format and compression type (e.g., from MPEG to AVI or from GIF to JPEG), transmit only selected spots, single frames or part of them.
- *Text transcoding*. A text transcoder may summarize the textual content, apply various style sheets to XML documents [38, 4], and transform HTML objects into WML objects for wireless devices [26]. An image removal service may discard images or transcode them into textual description if the information has to be delivered to a device that can only visualize text [21].

Network-oriented transcoding aims to adapt the user Web experience to the network status, which may be subject to changes in very short period of time. As a consequence network-oriented transcoding services

are more dynamic in nature with respect to device-oriented transcoding services. Examples of network-oriented transcoding services include:

- Tunneling/conversion of HTTP protocol over/to different protocols (e.g., in [12], the authors propose a system that provides HTTP tunneling over a GPRS-optimized transport layer based on UDP);
- Adapting object conversion parameters (e.g., compression level in images or bit rate in multimedia streams) according to network capability of the device and to the network status;
- Adapting conversion semantics to the connection characteristics (e.g., changing from a video stream to a sequence of images when an UMTS smartphone switches to a GPRS connection) [7].

3.1.2 Personalization services

Personalization services are directed towards the user. As shown in Figure 4, the personalization is oriented towards two main goals:

- *Personalization oriented to suit user preferences*. The interests communicated by the user and information inferred from its typical behaviors contribute to create a *user profile* which directs the content adaptation process.
- *Personalization oriented to suit user context*. This type of personalization considers the user within a specific context. The context information contains data such as the current user activity, the user geographic location and the identity of other interacting users.

Personalization services are extremely heterogeneous and several interesting services are not directly related to address user mobility. We provide a non-exhaustive list of services aiming to improve the user Web access experience, both from standard clients (i.e., PCs) and from mobile devices as a significant set of examples to outline the potential of personalization.

- *Content Filtering*. The request/response interaction of the user with Web servers is monitored to identify potentially harmful contents. Content filtering may be used to prevent access to Web content considered unsuitable for the user, or to intercept and remove virus/spyware from downloaded documents.
- *Language translation*. The textual information is translated according to user preferences. The user communicates its preferred language and, if the language is not within the set of available texts, the infrastructure may try to execute some run-time translation process.
- *Adaptation to the user navigation style*. The presentation of the resources is dynamically rearranged to help the user reaching the information of interest as soon as possible [17]. For example, the analysis of user click history allows the infrastructure to identify typical behavior patterns, such as paths in the navigation graph that leads from a page to another (or from a Web site to another Web site) through a specific sequence of links. Once a known pattern is recognized, the service delivery infrastructure can open directly the target page, thus saving some user clicks.
- *Adaptation to the user interests*. The system can figure out the preferences of the user and tune services. For example, this may modify content presentation and insert banners, according to the user interests. Another example of adaptation to user interests is provided by personalized portal pages that are composed by aggregating information from different and heterogeneous sources, such as XML-RSS news feeds.

- *Collaborative navigation services*. Collaborative navigation [5] is another example of service that enriches Web access allowing users to leave notes on Web resources. Other users can access both the Web resources and the meta-information left by other users. These social-friendly personalization services are gaining popularity [6] and are likely to represent a significant evolution of the Web and the Internet.

On the other hand, some personalization services are explicitly focused towards mobile users. Significant examples of such services are:

- *Location and surrounding-based services*. They achieve content personalization on the basis of the user geographic location. The user position is compared with geographic data that is managed by the service provider. The generation and delivery of static and dynamic content (e.g., queries) is carried out based on the user location and may be combined with some user preferences. Examples of surrounding-based services are typical in the field of pervasive computing, such as the system proposed by Console et al. [14] for tourism applications or applications for commercial information meeting, blind dates, banner insertion and advertisements.

- *Context-based services*. Context-based services aim to provide services that adapt Web resources to the user *current* interests and needs. For example, the ContextPhone platform [35] provides a flexible framework for mobile-phone applications, where the framework allows to adapt the offered services with data from external sensors (e.g., mobile phone camera, microphone) aiming to provide information about the user context. Such context-awareness may be merged with the connectivity module of the ContextPhone platform to provide context-aware access to network services. For example, the system can modify the rendering option of a multimedia stream depending on the ambient: video color and contrast can modified depending on the ambient light (e.g., dark room or outdoor environment), while audio volume can be adapted to the noise level perceived by the mobile device microphone. Context-aware service has been also proposed to provide distraction-free environment to maximize user productivity. For example, this approach has been widely explored in the Aura project [3, 19] which aims to provide a pro-active environment that takes care of low-level details of user activity, especially those related to information access from any location and with every device.

3.2 Classification of content adaptation services

For the goal of this tutorial, which is focused on the architectures to support ubiquitous Web access, we classify content adaptation services on the basis of the information that is used to provide content adaptation. We will see that information management has a major impact on the architectural choices to provide ubiquitous Web access. As shown in Figure 5 it is important to distinguish the type of information that is necessary for the content adaptation service:

- *Volatile information.* The information used to carry out the content adaptation is embedded within the request or may be explicitly communicated but it is managed on a per-session basis, without storing any information. These content adaptation services are not based on a previously stored information.

- *Persistent information.* Content adaptation services based on persistent information require some data about the user that is stored somewhere. We will refer to this data as *user profile*. The presence of a stored user profile is more important for the architectural choices than the presence of per-session information, hence in this category we consider also services (such as location-based services) that require on-the-fly aggregation of previously stored and per-session information [1].

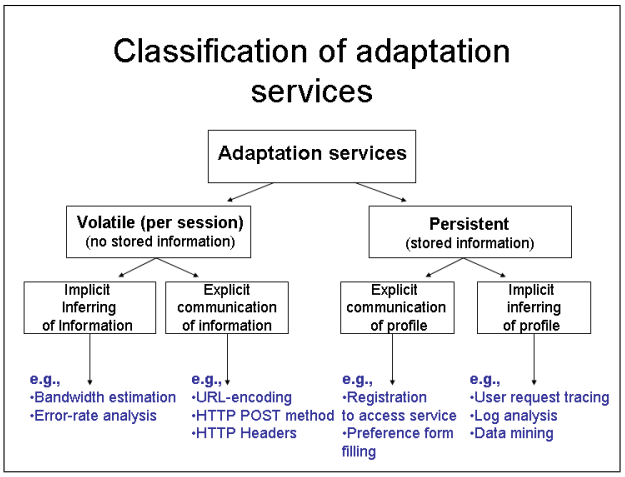

Figure 5: Content adaptation services

For any content adaptation service, a further and important level of classification is represented by the way through which information reaches the architecture for ubiquitous Web access. We distinguish between *explicitly communicated* or *implicitly inferred* information.

In the case of volatile information, explicit communication may occur in the forms of:

- URL-encoding/POST submission. This is the case where the information for the content adaptation services are directly available in the URL or in the request body of a POST HTTP request. For example, a search engine can enable or disable content filtering in the search results depending on some key-value pair sent in the URL or in the request body.

- HTTP headers. Standard headers can be used to select which content adaptation services should be enabled to provide ubiquitous Web access. For example, the *User-agent*, *Accept*, and *Accept-language* headers can be used to identify mobile devices, to enable transcoding options, and to enable automatic translation services, respectively

In the case of volatile information, implicit information inference may occur in the forms of:

- Bandwidth estimation. The ubiquitous Web access system can evaluate the available bandwidth for client downloads. As bandwidth changes dynamically, this information must be periodically refreshed and moving-average can be used to filter out the high variability in the status of network links.

- Error-rate analysis. Similarly to the case of bandwidth estimation, if the content adaptation system can access low-level network information, the error rate can be used to evaluate the opportunity of protocol conversion.

For adaptation services based on persistent information, we assume that the necessary information is stored in a user profile. The user profile can be obtained by:

- Explicit communications coming from the user. The typical example is a fill-in form to add/edit user preferences. This profile communication may occur when the user registers itself for the ubiquitous Web access services or may be filled/modified later through a Web-based service.

- Implicit inference of user profile. There are several ways to analyze the user behavior. Log files, cookies, user click history, trojans and spyware outputs may be subject to data-mining in order to infer user interests and profile information [17, 15, 28]. With the available information collection technologies it is possible to extract interesting information related to the user including sensitive data, such as political, physical and sexual features. Furthermore, most techniques are almost transparent to the user which is often completely unaware. Unauthorized user information collection occurred in the last years, for example by the doubleclick.com commercial advertisement service. Several commercial services, including search engines, were associated with doubleclick.com. The commercial sites used cookies to monitor their visitor's activities, and any information collected were stored in doubleclick.com databases. This user profiles were then used by doubleclick.com to select the advertisement banners more suitable for the users.

The different types of content adaptation services (in particular, the type of information they are based on) introduce different issues that will be analyzed in the following of this tutorial.

3.3 Issues in ubiquitous Web access architectures

Content adaptation services are extremely heterogeneous, ranging from simple image resizing to location-aware, social-oriented services. Most transcoding services are based on volatile information extracted directly from the client request, while personalization services tend to require more sophisticated information that are stored in the user profile. However, it is not possible to draw a clear connection between the transcoding/personalization classification and the type of required information.

For the goals of this tutorial, that analyzes the issues and the solutions for the design of ubiquitous Web access architectures, we will distinguish between services based on volatile or persistent information. This is the most relevant classification from an architectural point of view, because services based on persistent information must address all the issues related to the user profile management. These problems do not exist in the case of volatile information. It is worth to observe that the management of user profiles requires solutions to store and retrieve information (possibly from different sources) that must guarantee scalability, data consistency and privacy.

The design of ubiquitous Web access architectures aiming to provide advanced content adaptation services (thus allowing the deployment of services based on persistent information) must address the two issues of *performance* and *privacy*.

The need for architectures that satisfy the requirements of high performance combined with privacy of sensitive information results in a clear trade-off between distributed and centralized solutions. Increasing performance by means of highly distributed architectures is a common trend in Web content delivery, while privacy requirements suggest a more controllable, secure and centralized architecture. Proposing winning solutions to the performance/privacy trade-off is likely to be one of the main challenges for the success of ubiquitous Web access.

3.3.1 Performance issues in ubiquitous Web access

The performance and scalability requirements for ubiquitous Web access are motivated by the Web evolution towards a growing amount of multimedia contents, which require (and will require even more in the future) an increasing amount of computational power to enable ubiquitous Web access. Furthermore, the complexity of services required by ubiquitous Web access requires on-the-fly content adaptation. The pre-generation of formats for every combination of client device and user preferences set is clearly unfeasible.

The solution already known to address performance requirements in the standard Web access cannot be directly applied to the ubiquitous Web access. Indeed, traditional Web access performance is related to the

time required to retrieve from disk or generate Web contents and to the time required by the content delivery. On the other hand, on-the-fly content adaptation requires time that usually overweights the time for content retrieval/generation and delivery. In a scenario where the service time for content adaptation is the main contribution to the response time, the standard approaches for high performance Web must be thoroughly revised.

Content adaptation, besides the inherent complexity of the provided services, is characterized by significant requirements of computational power, as shown in the example in Figure 6. This figure compares the 90 percentile of the service times of different adaptation services with the standard Web page service time. The message from the multiple experiments is twofold:

- Content adaptation is a computationally expensive task, to the extent that it may take one to three orders of magnitude more than the service of standard Web resources.
- The computational requirements of adaptation services are highly variable and depend on the provided service. Service time ranges from milliseconds for some simple HTML manipulations up to seconds for transcoding of multi-media resources.

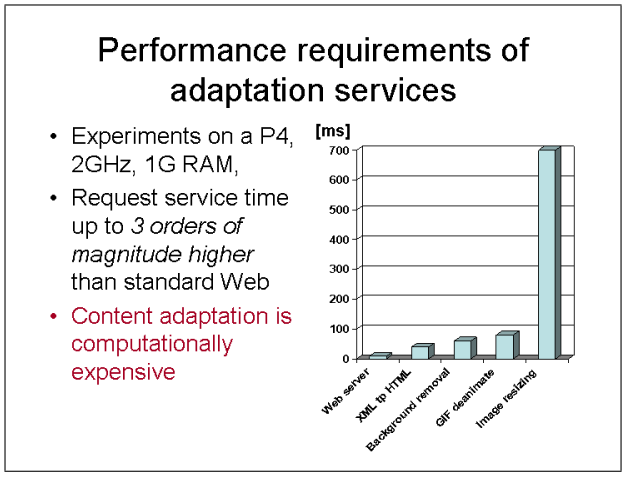

Figure 6: Performance of adaptation services

3.3.2 Security and privacy issues in ubiquitous Web access

The management of sensitive information composing the user profile opens another set of novel and important issues about adaptation services. This issue is typical of adaptation services based on *persistent information* where the presence of the user profile introduces the following requirements:

- It is necessary to guarantee the privacy of the sensitive information in the user profile.
- It is necessary to respect user privacy even in the case where implicit user profile communication is used (that is when user profile is inferred through user behavior analysis). Indeed, in most cases the users are unaware of the effort from Web service providers to model their behavior. Furthermore, the user has a limited freedom with respect to this data collection activities because some tools (e.g., cookies) used for information collection may be necessary to guarantee a better Web-based service.

The privacy issues impose important security requirements in the design of ubiquitous Web access architectures. Preserving the privacy of user-related information may be particularly complex in the case of location-aware services, where the service requires the knowledge of the user physical location. Multiple studies have been proposed to address this issue [22, 8], however, the performance impact of the cryptographic solutions proposed has not been yet widely explored.

A further important issue, gaining popularity in these days, is whether and how to inform users about personal data collection. Concerns about privacy due to log data mining and cookie analysis [1] motivate the efforts of defining novel mechanisms to negotiate what information can be derived from user behavior and how they are to be used. The doubleclick.com case is a clear example of the risks introduced by current data analysis technologies.

The Platform for Privacy Preferences (P3P) [32], shown in Figure 8, is an example of a proposal aiming to address this issue. Such proposal from the W3C confirms the strategic nature of privacy management in Web personalization.

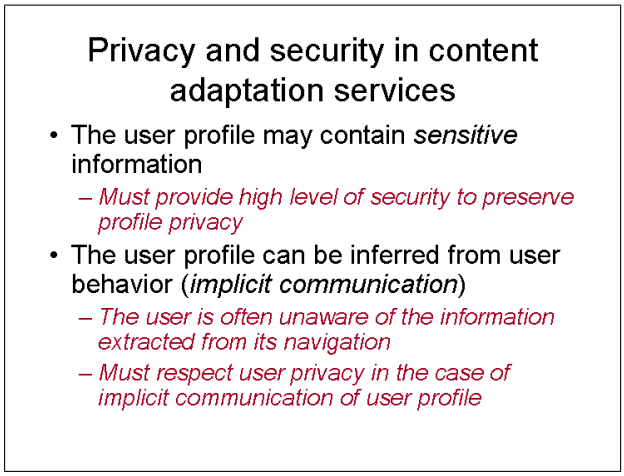

Figure 7: Privacy issues in content adaptation

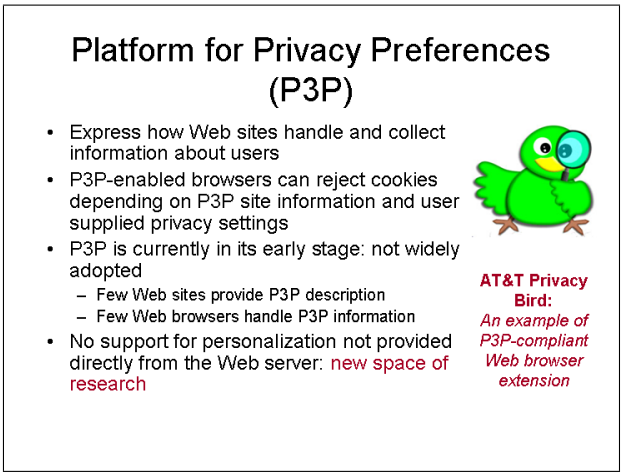

Figure 8: P3P platform proposal

4 Architectures for adaptation services

We now analyze the architectures for supporting ubiquitous Web access with a special focus on issues related to system performance and data privacy.

4.1 Classification

Multiple architectures have been proposed for the support of ubiquitous Web access. Figure 9 proposes a classification based on where content adaptation occurs:

- *Client-based architectures* where the content adaptation service process is provided directly by the client device.
- *Server-based architectures* where content adaptation is provided by the Web server of the content provider (the so-called *origin servers*).
- *Intermediary-based architectures* where content adaptation is provided by one or more intermediary nodes placed in the network, such as access points or proxies.

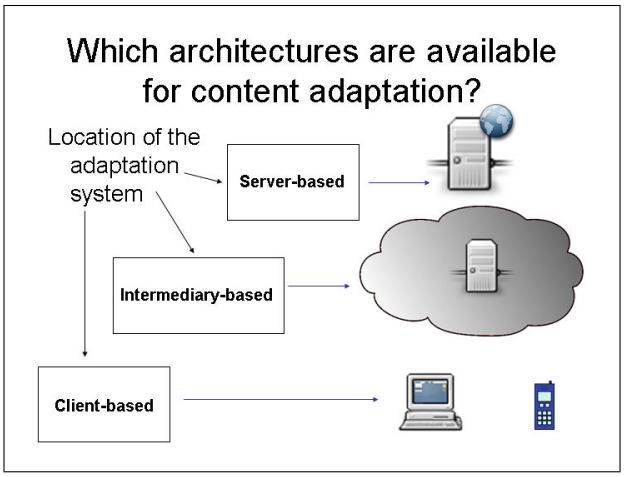

Figure 9: Classification of content adaptation architectures

In the tutorial we will present the main characteristics of the client- and server-based architectures that are outlined in Figures 10 and 11, respectively. In these tutorial notes we will focus just on intermediary-based solutions.

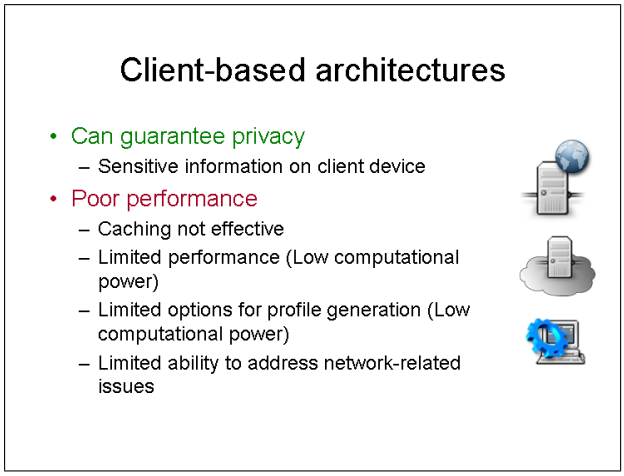

Figure 10: Client-based architectures

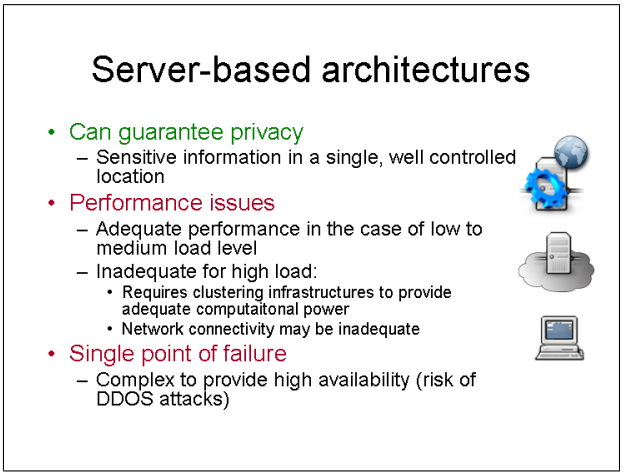

Figure 11: Server-based architectures

4.2 Intermediary-based architectures

Intermediary-based architectures come from the early period of the Web, when proxy servers were used as gateways and for performance reasons. More recently, these intermediate servers can provide also content adaptation along the client request path between the client device and the origin server, as shown in Figure 12 and 13. Intermediary-based architectures are appreciated because they can take advantage of existing servers, such as proxies or access points, that are often necessary to access the Web from mobile clients. In general, an intermediary-based architecture reduces the load on the origin servers, thus simplifying the design of the origin server platform and, when consisting of multiple servers, it may achieve good performance thanks to its inherent scalability. Indeed, intermediary-based infrastructures open a wide space of different alternative, especially if we refer to intermediary architectures consisting of distributed nodes. Some exam-

ples of intermediary-based architectures for ubiquitous Web access are provided by iCAP [23] and OPES (Open Pluggable Edge Services) proposals [31].

On the other hand, we should consider that intermediary-based architectures introduce significant issues on the privacy side, because the intermediary nodes cannot be considered trusted components as a client device or an origin server. Most intermediary nodes are operated by third parties (e.g., ISP) and, in general, it is difficult to guarantee security when the infrastructure is composed by thousands of nodes spread around the main point-of-presence of the Internet.

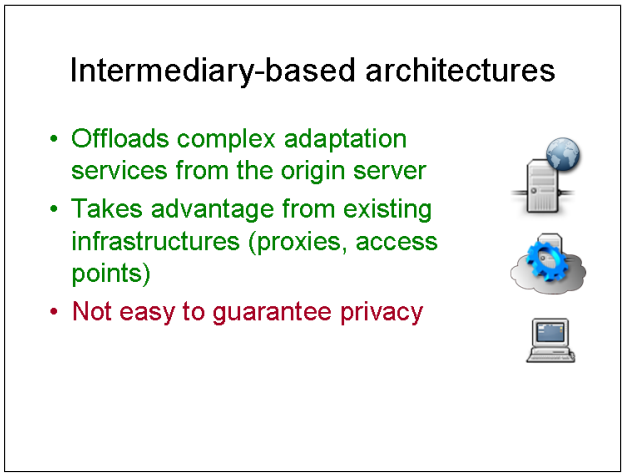

Figure 12: Intermediary-based architectures

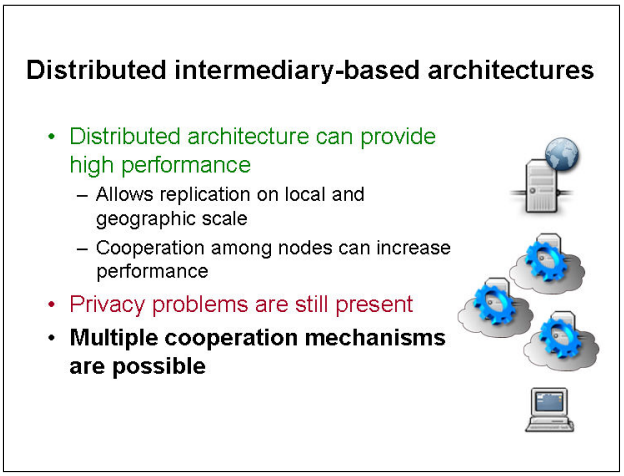

Figure 13: Distributed intermediary-based architecture

4.3 Main functions

An architecture for ubiquitous Web access implements and manages many functions that are related to accessibility and content adaptation. In Figure 14, we identify three main classes of functions: *Content adaptation* (already described in Section 3), *Data management*, and *Connectivity*.

Data management functions are related to storage and retrieval of information, such as caching of already adapted Web resources and management of user profiles and user-related meta data.

Connection functions provide the basic connectivity to the client devices and with the origin servers. In particular, *edge function* represents the front-end service of the intermediate architecture, including the possibility to receive the client request, to identify the client requirements and, once the requested resource is get from some source, to deliver it back. A *fetch function* is used to retrieve the resource(s) from the origin server, when the requested item is not found in any node of the intermediate distributed architecture.

To design a distributed intermediary-based system we need to map Content adaptation, Data management, and Connectivity functions on the nodes of the infrastructure (Figure 15). The services for enabling ubiquitous Web access are so heterogeneous that it is impossible to identify a "one size fits it all" solution. A large number of alternatives exist, some of which will be discussed in the tutorial. Here, we give an example by taking into account the most challenging personalization services that are based on persistent information about the user profile.

Content adaptation services based on persistent information are more complex than services based on volatile information, because they require a suitable profile management. This function adds the necessity of considering privacy and security to the traditional performance-related issues. Two topologies can be used to provide content adaptation services based on persistent information: *Flat topology* and *Two-level topology*.

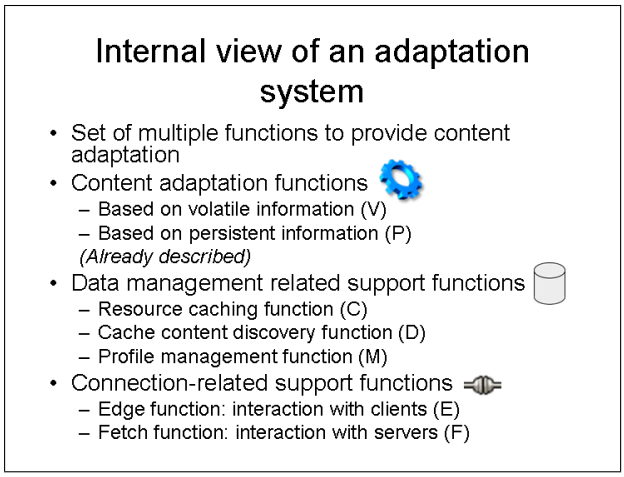

Figure 14: Functions of a content adaptation system

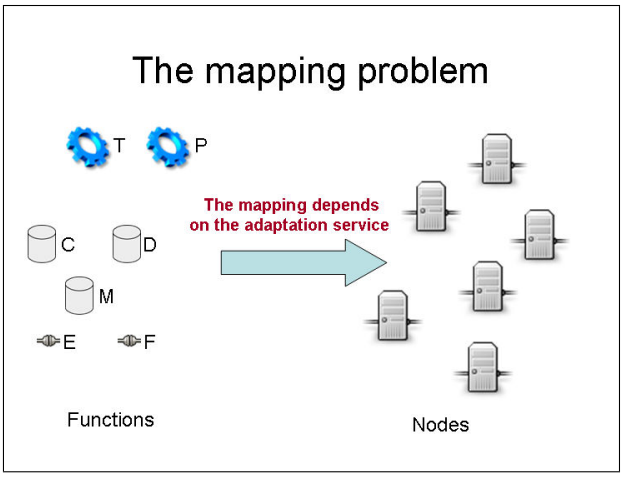

Figure 15: The functions/nodes mapping

Figure 16 shows an architecture based on a flat topology for content adaptation services requiring persistent information. In this architecture each node provides every function and acts as a reverse proxy for the origin Web server. The number of intermediary nodes is reduced because it is not affordable to provide a high level of security when the number of nodes and locations is too high. Hence, a more centralized architecture where few powerful nodes (possibly clusters, to provide adequate computational power) is preferable with respect to highly distributed architectures. Network-related delays can still be kept to an acceptable level by locating intermediary nodes in well-connected locations (the network core).

Figure 17 shows the architecture based on the two-level topology for content adaptation based on persistent information [11]. In the two-level topology the nodes are divided in two *levels*: *edge nodes* are located close to clients and *inner nodes* located in the network core, that is, in well-connected Autonomous Systems that have a large number of BGP peerings with other Autonomous Systems. The functions are distributed between edge and internal nodes as follows:

- Edge nodes host lightweight functions that do not require neither high computational power nor high security standards. In particular, edge nodes provide edge and discovery functions.
- Inner nodes provide the most critical functions of profile management, caching, content adaptation and fetch functions.

5 Ubiquitous-enabling systems

In the third part of the tutorial, we examine some examples of prototypes and real systems that support ubiquitous Web access. As shown in Figure 18, we classify the proposals by considering the user scope and the content scope of the adaptation services:

- The *User scope* denotes the users that may access the ubiquitous Web-based services. There are two main scenarios:
  - *Any user* may access the services related to the ubiquitous Web, because no registration/login is required. From an architectural point of view, a similar system may be subject to unpredictable workload. Moreover, it is impossible to exploit explicit profile communication from the user.

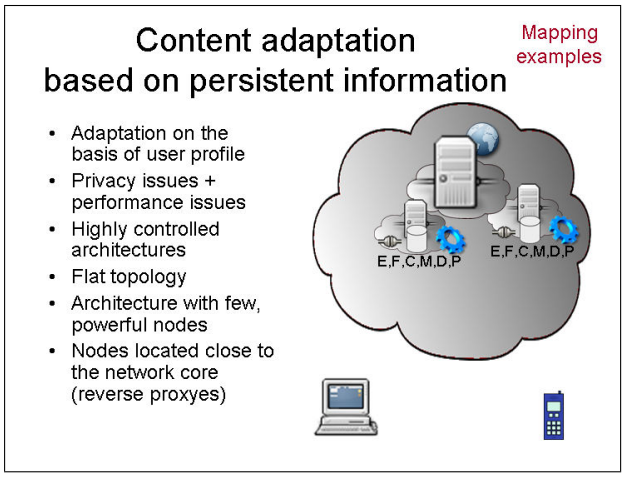

Figure 16: Flat architecture for content services based on persistent information

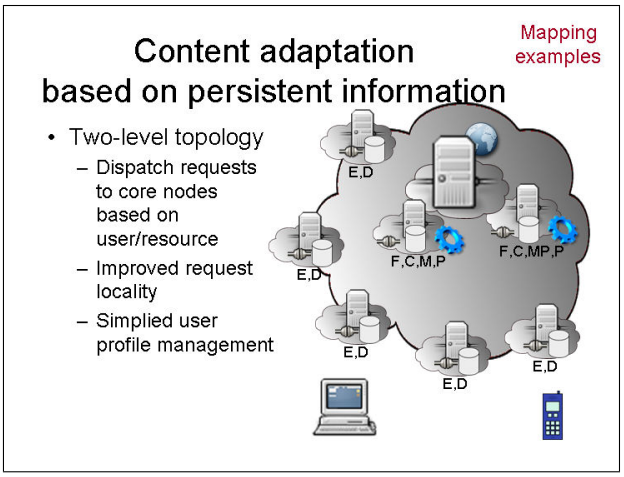

Figure 17: Two-level architecture for content services based on persistent information

While services based on volatile information remains unaffected, services based on persistent information are more difficult to deploy.

– *Partial set of user.* Only pre-registered users may access the system. In this case, the maximum load and the typical load are more easily predictable. It is likely to expect that a typical user fills in his/her profile form, that simplifies the deployment of advances personalization services.

- The *Content scope* denotes the set of Web sites that may be accessed through some infrastructure for ubiquitous Web access. Two main scenarios are possible:

  – *Any Web site.* The infrastructure for the ubiquitous Web access may be used for the whole Web. In this instance, the content adaptation system cannot rely on any information on the type of accessed Web resources. Not knowing the nature and the semantics of the provided information/services typically hinders the possibility of offering advanced services.

  – *Limited set of Web sites.* The infrastructure for the ubiquitous Web access refers to a limited set of Web sites. In this instance, it is possible to think to a tight interaction between the entity operating the content adaptation system and the provider of the original information/services. For example, this scenario is typical to support context-aware services.

The two scope criteria are orthogonal, which means that they identify four main scenarios (Figure 19):

- *Any user/Any site.* Ubiquitous Web access for any user (without login requirements) to any Web site. This is the typical case of ubiquitous enabled proxy servers.

- *Any user/Partial set of sites.* Ubiquitous Web access for any user to a subset of ”ubiquitous access friendly” Web sites. This is typical of Content Delivery Networks.

- *Partial set of users/Any site.* Ubiquitous Web access for a subset of registered users to any Web site. This is typical of a Web intermediary (portal-like) that enables ubiquitous access to its customer (the registered users).

- *Partial set of users/Partial set of sites.* Ubiquitous Web access for a subset of registered users to a subset of Web sites. This is typical of vertical solutions where a set of Web sites allows ubiquitous access to their customers.

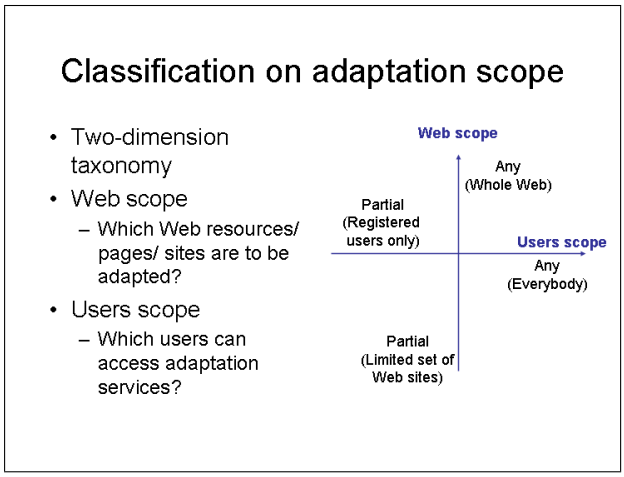

Figure 18: User scope and content scope

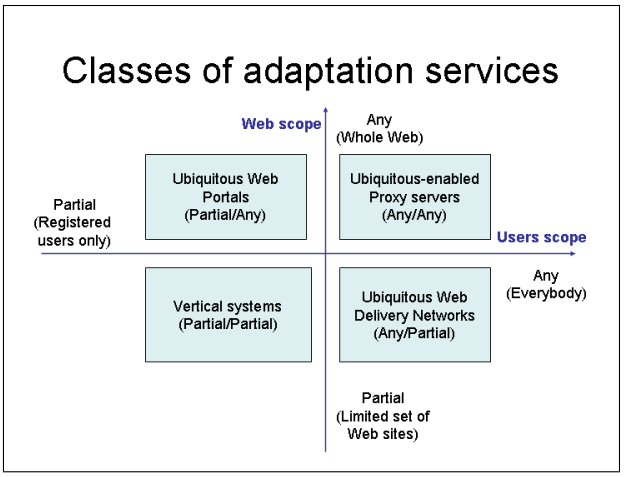

Figure 19: Ubiquitous-enabling systems

Let us consider each of these scenarios by showing the typical content adaptation services, the most suitable architectural solutions and some examples of prototypes and real installations.

5.1 Any user/Any site

The *Any user/Any site* class of ubiquitous-enabling systems is mainly composed by Web intermediaries (usually called proxies or edge servers) with capabilities enabling ubiquitous Web access (Figure 20). This is the most popular (and old) class of content adaptation systems and is characterized by:

- user login is not required, hence the provided services are the same for every user; no persistent information is stored.
- every Web site can be accessed through the Any user/Any site class of infrastructures.

From an architectural point of view, most systems belonging to this class are based on an architecture of intermediary nodes, without cooperation among the multiple servers. Due to the lack of knowledge about accessed Web sites and users issuing requests, the provided services are often simple, typically related to resource transcoding.

Ubiquitous-enabling systems in the Any user/Any site class ranges from simple proxies to rich frameworks. Examples of proxies that offer just basic services for enabling ubiquitous Web access are:

- *Muffin* [30] and *RabbIT* [34]. Muffin and RabbIT are proxy servers that provide a (non-distributed) intermediary-based architecture for content adaptation. Both programs aims to reduce bandwidth utilization to improve Web performance in the case of slow last-mile links. The proxies support transcoding services (in particular JPEG image quality reduction and data compression). Furthermore, they allow more complex content adaptation services, such as banner removal. This last service is provided through a black-list of advertisement sites. HTML tags requiring images from a blacklisted site are removed from the Web page.
- *Privoxy* [33] and *Junbkuster.* [25]. Junkbuster is a proxy aiming to preserve user privacy and filter unwanted contents. Its main function is to remove from Web pages the resources coming from black-listed Web sites. The rules for defining site blacklists and whitelists are more flexible than in the case of RabbIT and Muffing. Furthermore, the proxy allows to selectively filter cookies from requests

and responses, thus enabling a higher level of privacy for the users. Privoxy is a project based on Junkbuster that provides additional content filters such as pop-up and Javascript code removal.

The main example of a rich framework that provides extensible, programmable and composable services is:

- *Web Based Intermediary (WBI)* [27]. The WBI framework is a set of Java classes that allows the composition of a content adaptation chain providing advanced services. The idea of composable services in WBI is similar to the OPES specification [31]. The framework can be extended to implement per-user advanced services, however, in the WBI system the focus is on proxy administrators/programmers that build the content adaptation chain.

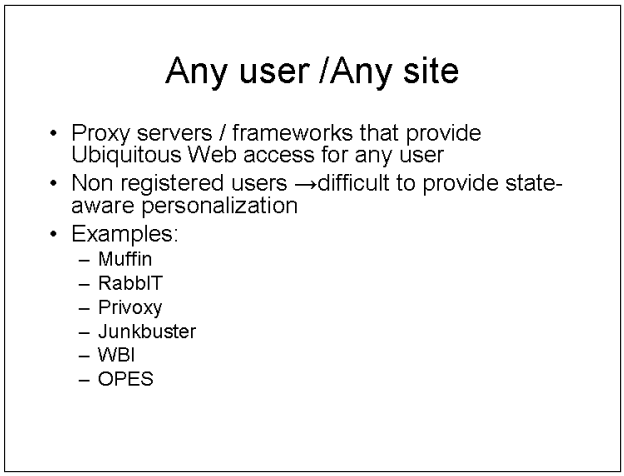

Figure 20: Any user/Any site

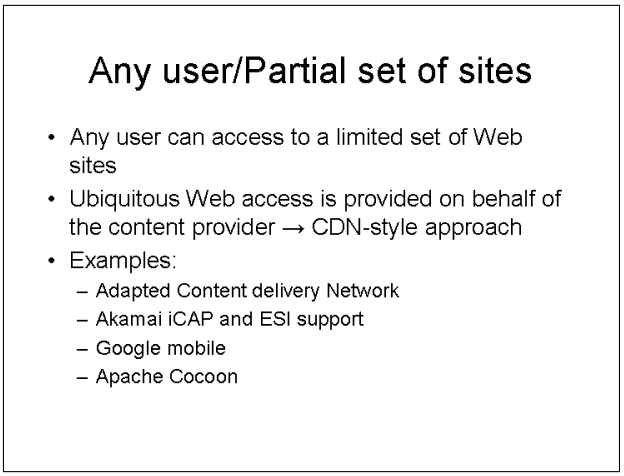

Figure 21: Any user/Partial set of sites

5.2 Any user/Partial set of sites

This class of *Any user/Partial set of sites* is of specific interest for CDN-like infrastructures that may serve adapted resources of a limited set of customer Web sites (Figure 21).

Systems belonging to this class typically use large distributed infrastructures consisting of very large numbers of nodes. These architectures are borrowed from the CDNs model. Nodes are organized in two (or more)-levels with edge servers in the ISP points of presence (close to the clients) and inner nodes that address the issues of data management (e.g., data consistency in the case of content/service replication)

Providing ubiquitous access to a subset of Web sites could allow advanced content adaptation services because the semantics of the Web site accessed is known. However, the lack of knowledge about user identity hinders the implementation of really advanced services. As a consequence, the provided services are usually related to transcoding or simple personalization based on the user request (e.g., banners personalized to suit the user preferred language). Some examples of systems belonging to the Any user/Partial set of sites class are reported below:

- Buchholz *et al.* [9] propose an Adapted Content Delivery Network that provides the functions of a CDN and is also capable of providing content adaptation services for ubiquitous Web access.

- Akamai [2] is a leader company in the CDN world. While the traditional services of the Akamai CDN are related to static Web content, the CDN has been involved in the proposal of the iCAP (Internet Content Adaptation Protocol) and ESI (Edge Side Includes) protocols [23, 16] and supports these protocols in its infrastructure for content adaptation services.

- The mobile.google.com portal [29] is another example of content delivery where every user can obtain ubiquitous Web access to a set of Web sites (in this case, the Google Web search sites).
- The Cocoon [4] technology from the Apache foundation allows to store page templates as XML files. The Cocoon engine is responsible for transforming the XML documents into Web pages (e.g., HTML or WML documents) through XSLT rules to enable the fruition of contents through different client devices.

5.3 Partial set of users/Any site

The main difference of the *Partial set of users/Any site* class of ubiquitous-enabling systems with the previous class is the business model (Figure 22). Now, the provider of the services for ubiquitous Web access may be an ISP or a network carrier that aims to provide ubiquitous Web access to its users. Unlike the CDN case, it is the user side that pays for ubiquitous Web access instead of the content/service provider. The user side may refer to different instances, such as a *single user* that needs advanced Web access features or a *corporate user* that requires this service for its employees.

The user identification procedure allows the deployment of content adaptation services based on persistent information. For example, each user may provide his/her own profile to build a personalized content adaptation service by composing simpler services. However, the content adaptation provided for every accessed site (with no direct knowledge on the accessed information) may limit the ability to provide advanced personalization services.

The Partial set of users/Any site class is often based on intermediary-based architectures that are already described in Section 4, with the addition of client-based extensions that implement services, such as protocol tunneling (e.g., the Avant-Go [24] system by i-Anywere).

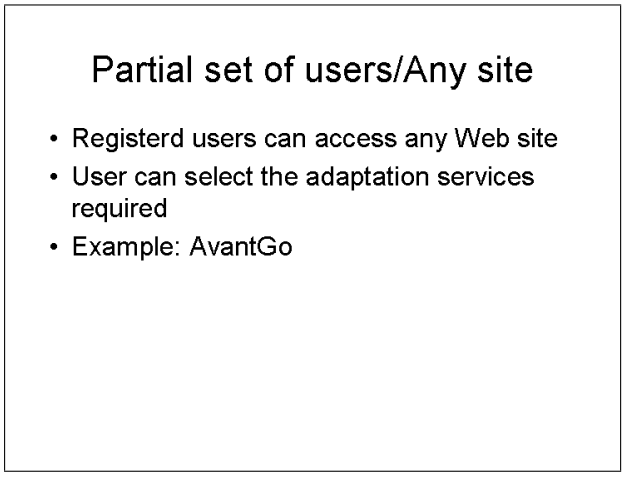

Figure 22: Partial set of users/Partial set of sites

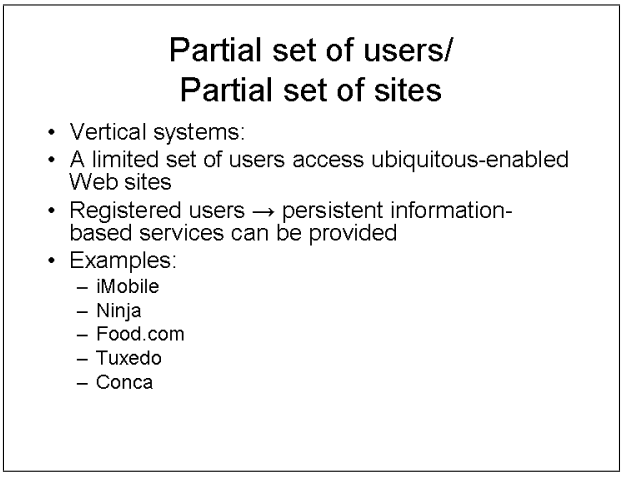

Figure 23: Partial set of users/Any site

5.4 Partial set of users/Partial set of sites

This class of systems supports content adaptation services for a limited set of registered users that access a limited set of Web sites (Figure 23). Several of these systems belong to vertical or enterprise solutions, such as portals with ubiquitous access for subscribers or employees.

These systems deploy the most complex content adaptation services. Indeed, the knowledge of both user and type of accessed information allows the infrastructure to provide highly personalized services.

The heavy use of persistent information in content adaptation services introduces the issues related to information privacy, that has been discussed in Section 3. For this reason, most systems belonging to this class are based on infrastructures with a limited number of nodes, that rely more on local replication (clustering) than on geographic replication. Some examples of systems belonging to this class are indicated below:

- The iMobile [36, 13] system contains a framework for providing ubiquitous access to information and services. The Web-based interface is one of the main interfaces for accessing information and services but iMobile supports also interfaces based on SMS and other protocols.
- The Ninja [20] project aims to achieve goals similar to the iMobile system, with a distributed infrastructure for seamless information and service access from mobile devices.
- The Food.com portal [15] is an example of location-aware service. The user profile supplied to the portal contains the geographic location of the user (its Zip code). The portal collects information about nearby restaurants and provides the user with a choice of daily menus.
- The Tuxedo and CONCA frameworks [39, 38] support mobile user access through a distributed P2P infrastructure. Their design is focused on performance and does not consider privacy issues related to the user.

6 Conclusions

Throughout this tutorial we present an overview of the issues and solutions to provide ubiquitous Web access, by focusing on main issues and some feasible solutions.

Although some research prototypes and even commercial products already exists, the available services do not exploit the full potential of the future ubiquitous Web. Hence, one message from this tutorial is that the research space about ubiquitous Web access is far from being fully explored.

We expect in the near future innovative ubiquitous-enabled services to appear. In particular, the class of services for registered users accessing to vertical applications shows interesting potential for the introduction of novel, personalized, and location-aware services.

A parallel research area is related to the design of novel infrastructures that may integrate solutions related to performance and privacy that have been addressed separately even by the recent literature. We claim that solutions addressing both performance and privacy issues are one of the main reasons for the success or failure of the ubiquitous Web-based services.

Glossary

Content adaptation. The process of tailoring Web resources (pages and objects) to a specific user, device, connection.

Pervasive computing. To access information and services from any device, from every location bu every user.

Pervasive computing devices. Mobile devices with networking and computing capabilities.

Ubiquitous-enabling system. A specific software package or an infrastructure that provides ubiquitous Web access.

Ubiquitous Web access. To access the Web from any device, from every location. Ubiquitous Web access is provided through *content adaptation*

Ubiquitous Web access architecture. An organized set of software and hardware components that enables ubiquitous Web access

User profile. The set of information related to the user preferences, typical behaviors and user context used to provide personalization services.

References

[1] A. Agostini, C. Bettini, and D. Riboni. Loosely coupling ontological reasoning with an efficient middleware for context-awareness. In *Proc. of Mobiquitous 2005*, S. Diego, CA, Jul. 2005.

[2] Akamai Inc., 2005. http://www.akamai.com.

[3] J. ao Pedro Sousa and D. Garlan. Aura: an architectural framework for user mobility in ubiquitous computing environments. In *Proc. of the 3rd Working IEEE/IFIP Conference on Software Architecture*, pages 29–43, Göteborg, SW., Aug. 2002.

[4] Apache Foundation. the Cocoon project, 2005. http://cocoon.apache.org.

[5] M. Barra, P. Maglio, A. Negro, and V. Scarano. Gas: Group adaptive system. In *Proc. of Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference*, Malaga, SP, May 2002.

[6] R. Beale. Supporting social interaction with smart phones. *IEEE PErvasive computing*, 4(4):35–41, Apr.–Jun. 2005.

[7] P. Bellavista, A. Corradi, R. Montanari, and C. Stefanelli. Context-aware middleware for resource management in the wireless internet. *IEEE Transactions on Software Engineering*, 29(12):1086–1099, Dec. 2003.

[8] C. Bettini, X. S. Wang, and S. Jajodia. Protecting privacy against location-based personal identification. In *Proc. of the 2nd VLDB Workshop on Secure Data Management*, pages 185–199. Springer-Verlag, 2005.

[9] S. Buchholz and T. Buchholz. Replica placement in adaptive content distribution networks. In *Proc. of the 2004 ACM Symposium on Applied Computing (SAC)*, Nicosia, Cyprus, Mar. 2004.

[10] M. Butler, F. Giannetti, R. Gimson, and T. Wiley. Device independence and the web. *IEEE Internet Computing*, 6(5), Oct. 2002.

[11] C. Canali, V. Cardellii, M. Colajanni, R. Lancellotti, and P. S. Yu. A two-level distributed architecture for web content adaptation and delivery. In *Proc. of The IEEE/IPSJ Symposium on Applications and the Internet (SAINT 2005)*, Jan./Feb. 2005.

[12] R. Chakravorty, A. Clark, and I. Pratt. Gprsweb: Optimizing the web for gprs links. In *Proc. of ACM/USENIX First International Conference on Mobile Systems, Applications and Services (ACM/USENIX MOBISYS 2003)*, pages 317–330, San Francisco, USA, May 2003.

[13] Y.-F. Chen, H. Huang, R. Jana, T. Jim, M. Hiltunen, R. Muthumanickam, S. John, S. Jora, and B. Wei. imobile ee - an enterprise mobile service platform. *ACM Journal on Wireless Networks*, 9(4):283–297, Jul. 2003.

[14] L. Console, S. Gioria, I. Lombardi, V. Surano, and I. Torre. Adaptation and personalization on board cars: a framework and its application to tourist services. In *Proc. of Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 112–121. Springer Verlag, 2002.

[15] M. Eiriniaki and M. Vazirgiannis. Web mining for web personalization. *ACM Transaction on Internet Technology*, 3(1), 2003.

[16] Edge Side Includes, 2002. http://www.esi.org.

[17] S. Flesca, S. Greco, A. Tagarelli, and E. Zumpano. Mining user preferences, page content and usage to personalize website navigation. *World Wide Web*, 8(3):317–345, Aug. 2005.

[18] A. Fox, S. D. Gribble, and Y. Chawathe. Adapting to network and client variation using active proxies: Lessons and perspectives. *IEEE Personal Communications*, 5(4):10–18, Aug. 1998.

[19] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste. Project aura: toward distraction-free pervasive computing. *IEEE Pervasive computing*, 1(2):22–31, 2002.

[20] S. D. Gribble, M. Welsh, R. von Behren, E. A. Brewer, D. Culler, N. Borisov, S. Czerwinski, R. Gummadi, J. Hill, A. Joseph, R. Katz, Z. Mao, S. Ross, and B. Zhao. The ninja architecture for robust internet-scale systems and services. *Computer Networks*, 35(4):473–497, 2001.

[21] R. Grieco, D. Malandrino, F. Mazzoni, and V. Scarano. Mobile Web Services via Programmable Proxies. In *Proc. of the IFIP TC8 Working Conference on Mobile Information Systems - 2005 (MOBIS)*, pages 139–146, Leeds, UK, December 2005.

[22] U. Hengartner and P. Steenkiste. Access control to people location information. *ACM Transaction on Information and System Security*, 8(4):424–456, Nov. 2005.

[23] Internet Content Adaptation Protocol, 2005. http://www.i-cap.org/.

[24] "iAnywhere Inc.". AvantGo, 2005. http://www.avantgo.com/.

[25] The Internet JunkBusters, 2005. http://www.junkbusters.com.

[26] E. Kaasinen, M. Aaltonen, J. Kolari1, S. Melakoski1, and T. Laakko. Two approaches to bringing internet services to wap devices. In *Proc. of the 9th international World Wide Web conference*, Amsterdam, NL, May 2000.

[27] P. Maglio and R. Barrett. Intermediaries personalize information streams. *Communications ot the ACM*, 43(8), Aug. 2000.

[28] P. Merialdo, A. Atzeni, and G. Mecca. Designa and development of data intensive Web sites: the ARANEUS approach. *ACM Transaction on Internet Technology*, 3(1), 2003.

[29] Google mobile, 2004. http://mobile.google.com/.

[30] Muffin World Wide Web filtering system, 2005. http://muffin.doit.org/.

[31] Open Pluggable Edge Services, 2005. http://www.ietf-opes.org/.

[32] Platform for Privacy Preferences (p3p) project, 2005. http://www.w3.org/P3P/.

[33] Privoxy, 2005. http://www.privoxy.org/.

[34] RabbIT proxy for a faster Web, 2005. http://www.khelekore.org/rabbit/.

[35] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. Contextphone: a prototyping platform for contxt-aware mobile applications. *IEEE Pervasive computing*, 4(2):51–59, Apr.–Jun. 2005.

[36] H. Rao, Y. Chen, D. Chang, and M. Chen. imobile: a proxy-based platform for mobile services. In *Proc. of the 1st workshop on wireless mobile Internet (WMI2001)*, 2001.

[37] D. Saha and A. Mukherjee. Pervasive computing: A paradigm for the 21st century. *IEEE Computer*, 36(3), Mar. 2003.

[38] W. Shi and V. Karamcheti. Conca: An architecture for consistent nomadic content access. In *Proc. of Workshop on Caching, Coherence, and Consistency (WC3 '01)*, Sorrento, IT, June 2001.

[39] W. Shi, K. Shah, Y. Mao, and V. Chaudhary. Tuxedo: a peer-to-peer caching system. In *Proc. of the 2003 Int'l Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'03)*, Jun. 2003.

[40] G. C. Vanderheiden. Anywhere, anytime (+ anyone) access to the next-generation www. *Computer Networks and ISDN Systems*, 8(13), Sep. 1997.