

Mobile Personal Website

Johan Wikman*
Nokia Research Center

Ferenc Dósa Rácz†
Nokia Research Center

1 INTRODUCTION

Most, if not all, smartphones of today are, in terms of processing power and the amount of available memory, on the same level as or even surpassing the servers of the early web. In the beginning of 2004 we started a project at Nokia Research Center with the aim of exploring what implications it would have if it were possible to host a website on a mobile phone and if that website were addressable in and accessible from the Internet, using the operator networks of today.

2 WEB SERVER

We set out with the intention of putting a full-fledged web server, that implements the full HTTP protocol, on a mobile phone. Due to the modest size of the project, creating a web server from scratch was not a realistic alternative and instead we chose to port Apache httpd[2].

Apache is implemented on top of APR[4] — *Apache Portable Runtime* — which is a software library whose intention is to provide a predictable and consistent interface to an underlying platform-specific implementation. The porting friendliness of Apache, including APR, stems from two sources: firstly they are modular in general, which allows you to remove functionality that is not needed, and, secondly, it is easy to replace a common implementation of some functionality with a platform specific implementation.

Apache modules are written in C or in Symbian's[6] case also in C++. While that makes it possible to access all functionality of Symbian and in our case S60[7] it requires a certain amount of diligence. Fortunately, while we were working with Apache, another group at Nokia Research Center were in the process of porting Python to S60[5][1], so we proceeded to port *mod.python*[3] — an Apache module that closely integrates Python with Apache — to Symbian as well. Consequently, content generating scripts can now be written in Python, which obviously lowers the threshold for developing web applications by a fair amount.

3 CONNECTIVITY

As our goal was to make a web server on a mobile phone accessible from any regular browser on the Internet at any time, there was really no alternative but to provide the connectivity using the cellular network. However, that is not quite as straightforward as it may seem as operators typically employ Network Address Translation and firewalls that effectively prevent a device on the inside of the firewall to be reached from the outside.

The restriction can be circumvented by arranging for the connection to be created by the device on the inside of the firewall and then deliver traffic to the device over that connection. That is, there is no need to be able to reach the mobile phone directly. In order to create that outbound connection the phone obviously needs something to connect to and for that we created a custom gateway that runs on a computer on the Internet.

To make it appear as if the web server on the mobile phone is directly addressable we employ a generic DNS mapping like:

*.raccoon.net ⇒ Gateway IP Address

Thus, when someone browses to, for instance, `http://john.doe.raccoon.net` and the browser does a domain name lookup it will resolve to the IP address of our gateway. However, when the browser subsequently opens a connection to the gateway and submits the request, the header will still contain the original host name `http://john.doe.raccoon.net`.

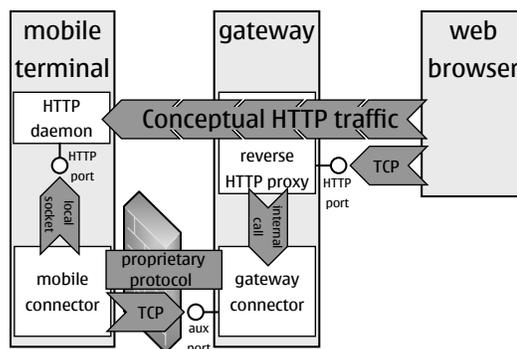


Figure 1: Cellular accessibility.

From the header the gateway knows that the request is intended for the mobile phone known as *john.doe* and if it is connected, the request can be sent forward over the connection opened by the phone. Finally the request can be delivered to the actual web server. Thus, as shown in Figure 1, it seems as if there would be a direct connection between the web browser and the mobile web server, although there are a few extra components in between.

4 WEBSITE ON A MOBILE PHONE

Although current smartphones are quite powerful, the processing power, the amount of memory and the available network bandwidth will, for the foreseeable future, be inferior to that of regular computers. Consequently, it would probably be a mistake to simply move what could be characterized as a “regular” website to a mobile phone. However, a phone equipped with a web server is quite different from a phone without one and a website on a mobile phone—or *mobsite*, as we call it—is quite different from a regular website, and if these specific aspects are taken into account, a number of new possibilities emerge.

4.1 A Phone with a Web Server is Different

Even though the user interfaces of mobile phones are quite user friendly, they are still rather constrained compared with the possibilities offered by the big screens and proper keyboards of regular PCs. If a web interface is created for the core applications of a mobile phone, it would mean that whenever you have access to a PC—any PC, not necessarily your own—with Internet connectivity you would be able to access the functionality of your phone using a proper keyboard and a big display.

*e-mail: johan.wikman@nokia.com

†e-mail: ferenc.dosa@nokia.com

If you have a web server running on the mobile phone then you can have web service providers as well. That opens up new possibilities for making mobile phones even more general purpose than they currently are. For instance, with a web service interface to the calendar it would be easy to create a distributed calendar application without the presence of any centralized server.

Currently the means for communicating with the phone are basically defined by standards, phone manufacturers and operators. In practice, you can call a phone or send it an SMS or MMS message. With a generic HTTP connectivity to the phone it is possible to create new messaging concepts without a-priori standardization or support from the operator.

Disseminating any data from a mobile phone requires currently that the phone owner actively, for instance, sends an SMS or an MMS, or uploads a picture to a website. With a web server on the mobile phone the situation can be reversed; for instance, instead of sending MMSs to your friends when you are on a vacation, you could provide them with a pointer to a picture gallery on your phone that they can browse at their leisure.

4.2 A Website on a Phone is Different

A regular website can be personal but the personal content is largely explicitly created and has to be manually uploaded, because the website does not reside on a personal device. Contrary to that, a mobile phone is personal and both contains and collects personal data that easily can be used for (semi)automatically creating a mobile home page.

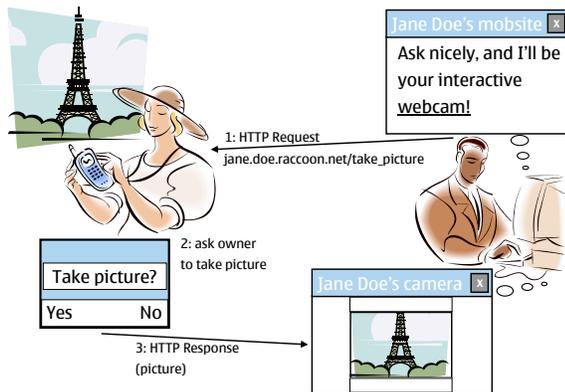


Figure 2: Interactive website.

Furthermore, a website on a personal device has always its “administrator” nearby who can actively participate in the content generation. For instance, Figure 2 describes a web application that, when activated, beeps and displays a dialog box with the question *Take Picture?* and two buttons *Yes* and *No*. If *Yes* is pressed, then a picture is taken, converted to a JPG and returned. That is, the fact that the website resides on a device that the owner most of the time carries with him means that websites can now become interactive.

In traditional websites the geographical location of the actual webserver lacks meaning since it never changes and it has no impact on the returned content. With a mobsite this is no longer the case as the returned content may depend on the geographical location and surrounding context. An example of this is a concept we call *mobsite hopping*, which is illustrated in Figure 3.

By utilizing the device discovery functionality of Bluetooth[8] we can dynamically find and create a list of other mobile websites in the proximity of a particular mobsite. It should be noted that the

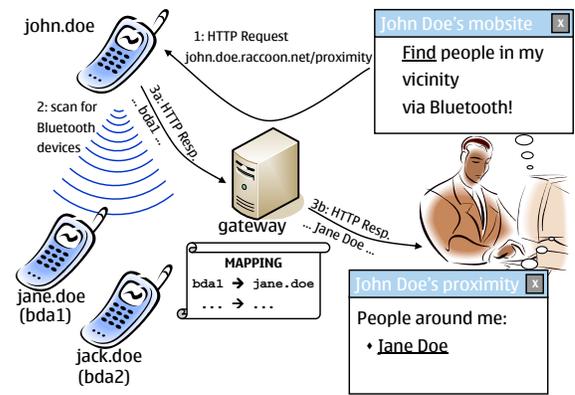


Figure 3: Mobsite hopping.

use of Bluetooth is restricted to the finding of other websites while the actual accessing takes place directly. That is, the access of a second mobsite is not dependent on the first mobsite *remaining* in the proximity of the second.

The fact that a website on a mobile phone is *mobile* implies that “traditional” means for linking websites are applicable but not sufficient. The constantly changing context must be taken into account.

5 DEMONSTRATION

In the demonstration we will show, among others, the application concepts described in this extended abstract, both from a development and application point of view. We will have a number of mobile phones and access the mobile websites using browsers on our or the conference attendees’ laptops and mobile phones.

For the demonstration we would need: a poster stand, WiFi/Ethernet connection to the Internet at our booth/table, GPRS connectivity, 3-4 power sockets at our booth/table, and preferably a data projector or a big external monitor.

REFERENCES

- [1] Nokia Research Center. Python for s60. <http://opensource.nokia.com/projects/pythonfors60/index.html>.
- [2] Apache Software Foundation. Apache httpd web server. <http://httpd.apache.org>.
- [3] Apache Software Foundation. Apache/python integration. <http://www.modpython.org>.
- [4] Apache Software Foundation. Apr. <http://apr.apache.org/>.
- [5] Python Software Foundation. Python. <http://www.python.org>.
- [6] Symbian Ltd. Symbian os. <http://www.symbian.com>.
- [7] Nokia. S60. <http://www.s60.com>.
- [8] Bluetooth SIG. Bluetooth. <http://www.bluetooth.com/bluetooth>.