

## Towards a 'Service Socket'



Aart van Halteren

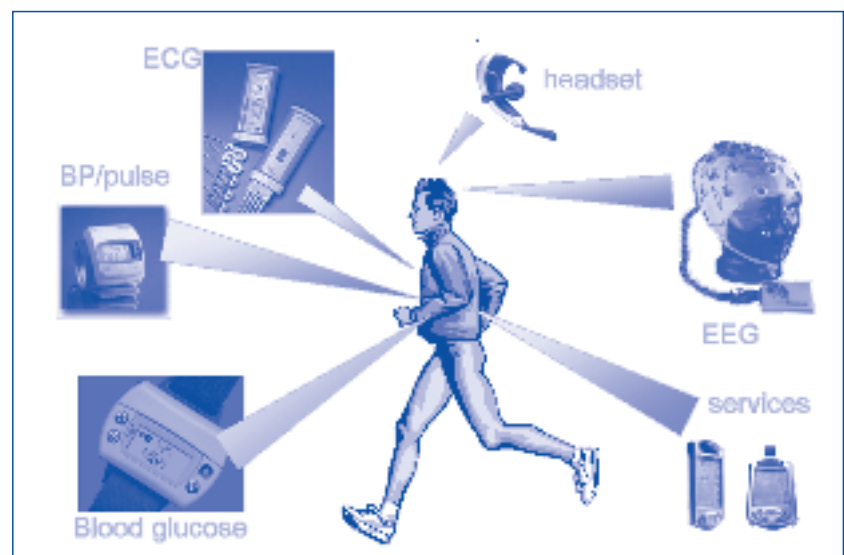
Instead of a network socket, Aart van Halteren would like to see a 'services socket' on the wall, not just giving access to the physical network but providing personalized information services to users. He is developing middleware that negotiates with both the user and the network, so that optimal services can be offered independent from the network environment. He currently applies his 'service socket' principle in the area of mobile healthcare. How to efficiently and reliably send personalized medical data from the pocket computer of a patient over a public network to a healthcare center?, Aart wants to know.

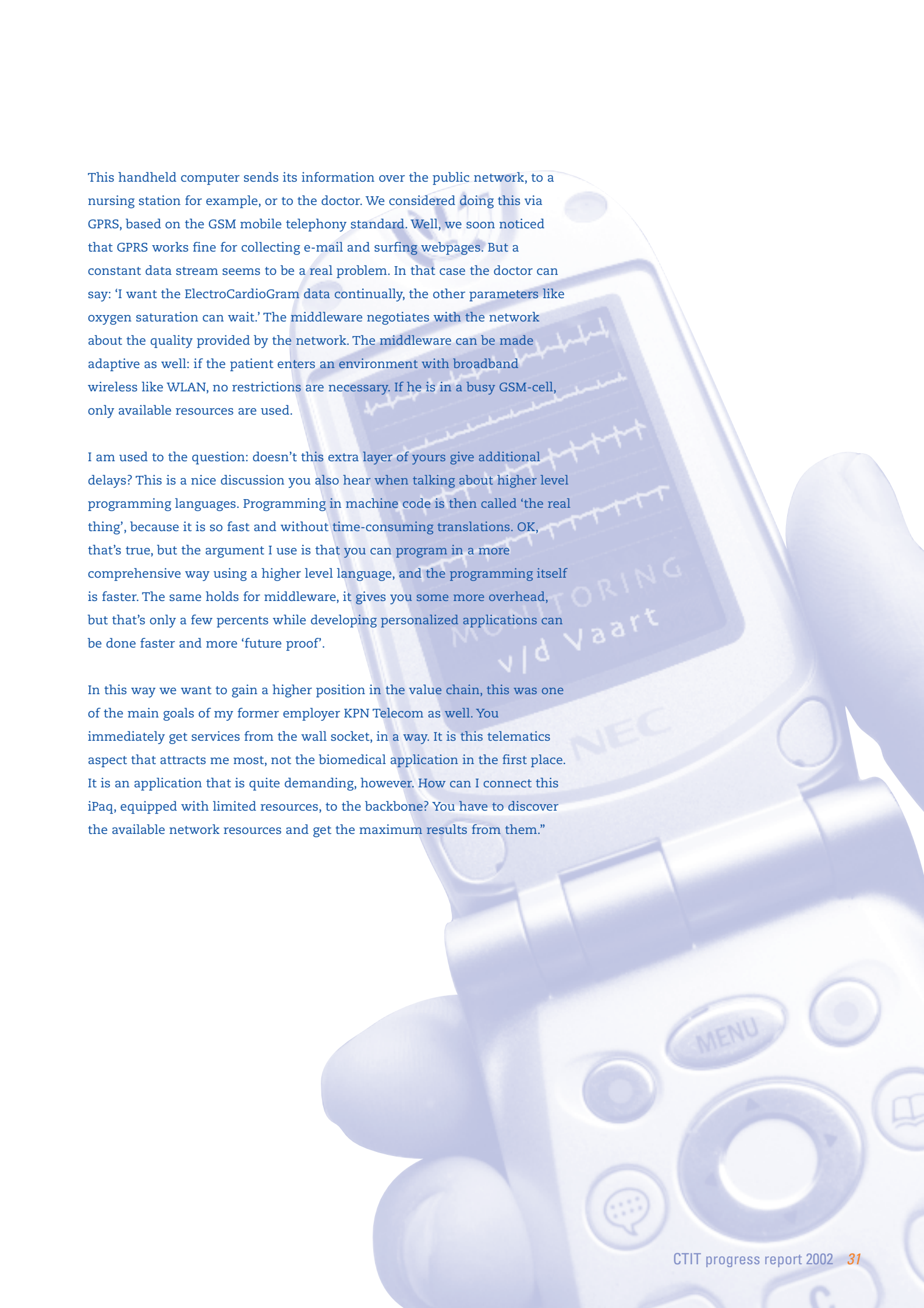
"Middleware is rather confusing terminology, you hear a lot of different interpretations. My idea is that it is a layer you create between the user and the network. Applications or users do not directly negotiate with the network, but they make agreements with the middleware. As a user, you are not interested if your application currently runs via a WLAN, GPRS or UMTS network. If you make a phone call, you don't want to know if it is transported via copper or glass. You have certain requirements on performance and costs. Using our method you can define your requirements on a software level, the middleware does the job, and you don't have to worry about network details and limitations. Reaching Quality of Service (QoS) agreements is usually done on the network level, the challenge is to move QoS agreements to the level of personalized information services. In that way, we lift it one level higher."

"Our MobiHealth application is a nice test case in that respect. A patient will wear a network of sensors, a Body Area Network, on his or her body. The sensors communicate wirelessly with each other and with an iPAQ computer.

### Examples of current projects:

- MobiHealth: Mobile Health Care (EU / FP5)
- SEACORN: Simulation of Enhanced UMTS Access and Core Networks (EU / FP5)
- WASP: Web Architectures for Services Platforms (TI)
- End-to-end Quality of Service in next-generation networks (Ministry of Economic Affairs)
- Beyond 3 G (Ministry of Economic Affairs)





This handheld computer sends its information over the public network, to a nursing station for example, or to the doctor. We considered doing this via GPRS, based on the GSM mobile telephony standard. Well, we soon noticed that GPRS works fine for collecting e-mail and surfing webpages. But a constant data stream seems to be a real problem. In that case the doctor can say: 'I want the ElectroCardioGram data continually, the other parameters like oxygen saturation can wait.' The middleware negotiates with the network about the quality provided by the network. The middleware can be made adaptive as well: if the patient enters an environment with broadband wireless like WLAN, no restrictions are necessary. If he is in a busy GSM-cell, only available resources are used.

I am used to the question: doesn't this extra layer of yours give additional delays? This is a nice discussion you also hear when talking about higher level programming languages. Programming in machine code is then called 'the real thing', because it is so fast and without time-consuming translations. OK, that's true, but the argument I use is that you can program in a more comprehensive way using a higher level language, and the programming itself is faster. The same holds for middleware, it gives you some more overhead, but that's only a few percents while developing personalized applications can be done faster and more 'future proof'.

In this way we want to gain a higher position in the value chain, this was one of the main goals of my former employer KPN Telecom as well. You immediately get services from the wall socket, in a way. It is this telematics aspect that attracts me most, not the biomedical application in the first place. It is an application that is quite demanding, however. How can I connect this iPaq, equipped with limited resources, to the backbone? You have to discover the available network resources and get the maximum results from them."