

# BlueSpot System– Infrastructure for Mobile Services

Ulrich Dümichen, Uwe Baumgarten

Technical University of Munich  
Boltzmannstr. 3  
85748 Garching / Munich  
Germany  
Tel.: +49 (89) 289-18592  
Fax.: +49 (89) 289-18557  
e-mail: [duemiche@in.tum.de](mailto:duemiche@in.tum.de)  
[baumgaru@in.tum.de](mailto:baumgaru@in.tum.de)

## Abstract:

This paper describes the BlueSpot system, a system which provides an infrastructure for mobile users. Every participant can access offered Mobile Services which present different information. The transmission of information occurs via Bluetooth. Therefore small hotspots are installed to build up a radio connection to the device of a user. The whole system can be run in two different modes which will be described in this paper. To keep the actual set of information there is a server in the background. The functionality of the BlueSpot system is implemented in a middleware. This middleware has to be present on every participant of the system. One function of the middleware is to manage all connections to other involved devices. A Mobile Service connects to an interface the middleware provides. With help of this interface the Mobile Service an access all available connections and communicate requested data. The kind of information distributed by a Mobile Service is kept as flexible as possible. In this paper are described two example Mobile Services, but also future developments.

## Introduction

In the last years the development of mobile technologies was very rapid. In order to market their products the industry of mobile devices and telecommunication needed to invent new features to convince customers to purchase the new devices. One of these features is Mobile Services. They are used to provide any kind of data to a mobile user, i.e. a user can access a Mobile Service from almost everywhere and retrieve nearly any information he wants to. So the Mobile Service is used to make personalized and filtered information available which fits to the current user.

This paper introduces a new system, the BlueSpot system. It allows providing several different Mobile Services on an arbitrary extendable area. Main implantation targets are large buildings or areas like the campus of a university. Especially the scenario of an application on a campus is of great interest. E.g. students could get informed about last minute changes, like the change of a room or the delayed beginning of a lecture. In addition to that a Mobile Service can provide maps to users which show their current position and the way to their destination.

A Mobile Service can also be location based. E.g. if the system is built up in an art gallery each painting could be equipped with a communication spot which carries the important information to the painting. Once a user gets within range the communication spot transmits the information to the user's wireless device. It wouldn't be necessary to purchase a brochure anymore and the user would be able to filter the information according to his demands.

A third scenario could be the public transport system. While waiting for the train a user can receive information concerning the whereabouts of the train, like when it arrives too late, etc. In this case also application based Mobile Services are imaginable, i.e. a provided Mobile Application could calculate the fastest connection to the destination.

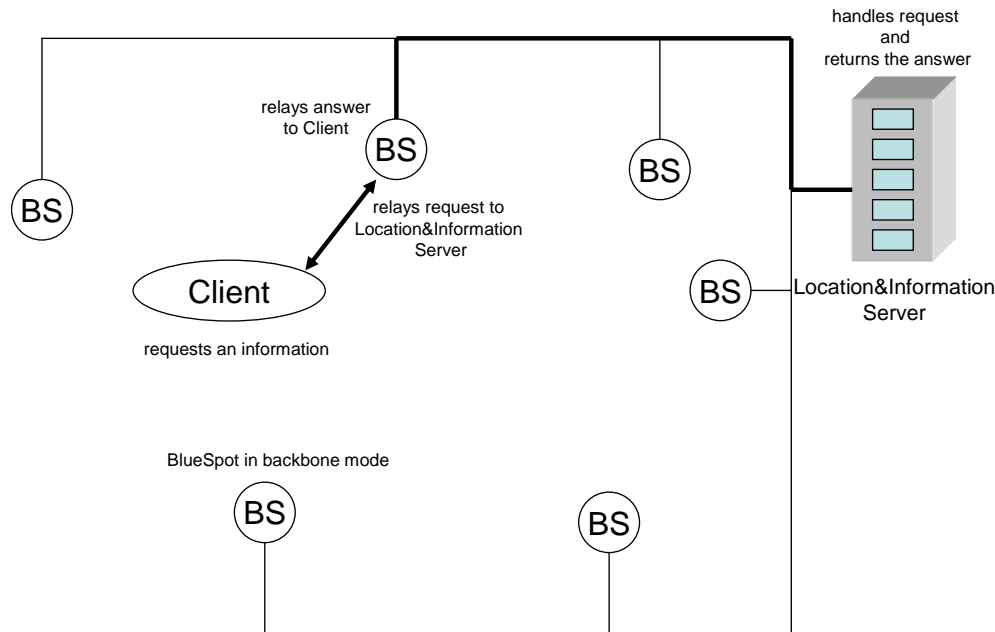
All in all, several different fields of application are thinkable, and this was the main motivation to develop such a communication system.

## Description of the BlueSpot system

As mentioned above, the BlueSpot system implements an infrastructure to provide Mobile Services. These Mobile Services can be used by every participant of the system. Each Mobile Service is specialized in the kind of information it provides. The interface between the BlueSpot system and the mobile users are hotspots, called *BlueSpots* (BS). These BlueSpots are very small and cheap devices which are equipped with a Bluetooth radio and an optional Ethernet device. In addition to that they have a small memory to keep a copy of the information provided and a middleware which is used to support the BlueSpot system. This middleware must also be present on the mobile device of the user, called *Client* (C) in the following. Its job is to manage the connections between participants and to provide a platform where the Mobile Services can be run.

There is also a server, called *Location & Information Server* (LIS). Within this server all data provided by the Mobile Services is stored. It is globally defined that only the data available on the Location & Information Server is the current data. Since it is possible to interact with users by Mobile Services the received data must be updated on the Location & Information Server before it is available to other participants.

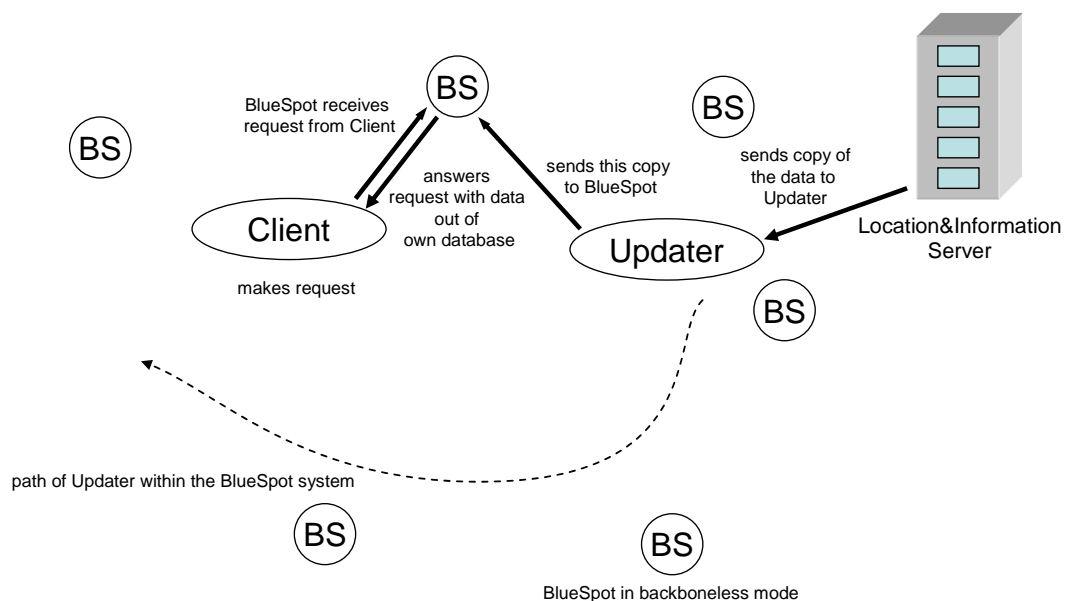
The BlueSpot system can be run in two different modes. Most commonly used is the *Backbone Mode*. In this configuration every BlueSpot must be equipped with an Ethernet device. By this device all *BlueSpots* are connected to the Location & Information Server. When a Client gets within range of a BlueSpot the BlueSpot builds up a connection automatically. Running in Backbone Mode the BlueSpot serves as router for requested information between the Client and the Location & Information Server, so it just redirects any



**Figure 1: the BlueSpot system running in Backbone Mode**

data. **Fehler! Verweisquelle konnte nicht gefunden werden.** shows the structure of a BlueSpot system in Backbone Mode.

The second mode is called *Backboneless Mode*. This mode is used when a connection between the Location & Information Server and the BlueSpots is not possible. For this mode a new participant is needed in the system, the *Updater* (U). It is also a mobile device and is featured with a Bluetooth radio and an Ethernet device. Once connected to the Location & Information Server via Ethernet it copies the whole set of available information. Then it moves through the BlueSpot system and *updates* all the BlueSpots by copying the data into their memory via Bluetooth. The setup can be seen in Figure 2:



**Figure 2: the Updater updates all the BlueSpots within the system**

By this the whole system is kept up-to-date.

The disadvantage of the second mode is that no direct interaction between two clients is possible. E.g. if one Mobile Service implements a standard chat all participants can

communicate on the fly if the selected mode is Backbone. Using the other mode this is not possible and just static content is available because the update by an Updater takes too long.

## **Detailed information on used hard- and software**

The last paragraph gave an overview of the function of the devices involved. In the following this paper will talk about the underlying hard- and software technologies. First, it will describe the Location & Information Server, then the BlueSpot and the other mobile devices will be explained.

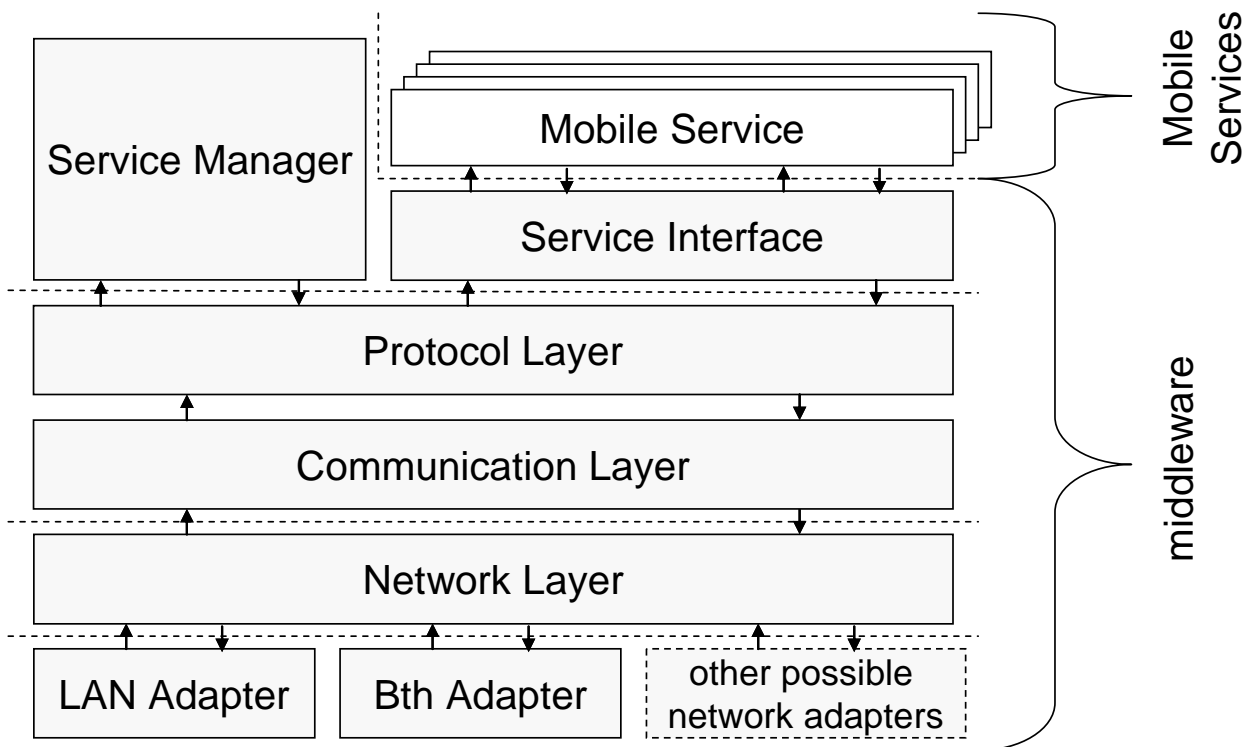
The Location & Information Server is implemented on a Microsoft Windows Server operating system. The underlying hardware is a standard x86 compatible computer. The BlueSpot system application on this server is implemented in C# using the .Net Framework. The server must be equipped with an Ethernet device which enables the communication with the BlueSpots in Backbone Mode and with the Updater in Backboneless Mode. While developing the server application the server was supposed to be equipped with a USB Bluetooth dongle, too. With this dongle the server would have been able to communicate with the Updater by Bluetooth as well. But the only available Bluetooth stack at that time was from Widcomm Inc., an US enterprise [Wid04]. The costs for their stack had been too high, so that the concept had to be changed and the Updater now has to communicate via Ethernet with the server. The BlueSpot works with the aid of an eBox distributed by Treckstore [Trk04]. This is a small personal computer with 200MHz Intel Pentium processor on a Sis 550 chipset with 128MB ram. Its size is 14 to 10 to 6 cm and it costs about 200€ The operating system is Microsoft Windows CE .Net 4.2. In order to have full support for the eBox it was necessary to build an own derivate of the OS with the help of the Microsoft Platform Builder. This software allows customizing Windows CE .Net with device drivers and functionalities needed. It is necessary to boot the eBox with an USB memory dongle with a DOS operation system to start it. Then a program gets started which enables the download of the Windows CE image from the server and its start. In future it will be possible to boot the eBox directly without booting DOS first. Then the image will be downloaded with the aid of the Etherboot protocol and a DHCP server. Then no more mass storage devices like the USB memory dongle are needed. By plugging in a USB Bluetooth dongle the eBox obtains a Bluetooth radio. The software component for the BlueSpot system running on the eBox was implemented in Embedded Visual C++ with the use of the MFC (Microsoft Foundation Classes) for creating GUIs. The Updater and the Client have also been eBoxes in the test scenario.

At this moment we are working on the migration of the middleware to a HP IPaq PDA and a XDA 2. A disadvantage of the IPaq is that HP doesn't use the Bluetooth stack developed by Microsoft for Windows CE .Net. They use a third party stack implemented by Widcomm where as the desktop version the SDK needed for developing Bluetooth applications is as expensive.

The standard way in the BlueSpot system to transmit data to a mobile device is via Bluetooth. But the system is not limited to Bluetooth. So the transmission via WLAN or even GSM is also thinkable. To have support a corresponding driver has to be implemented to the BlueSpot middleware, which every participant needs.

## **Service middleware and protocol architecture**

All devices are equipped with a middleware which is necessary in the BlueSpot system. This middleware is standardised by a predefined layer structure which forms an abstraction within each layer from the bottom to the top. Figure 3 shows the structure of the middleware.

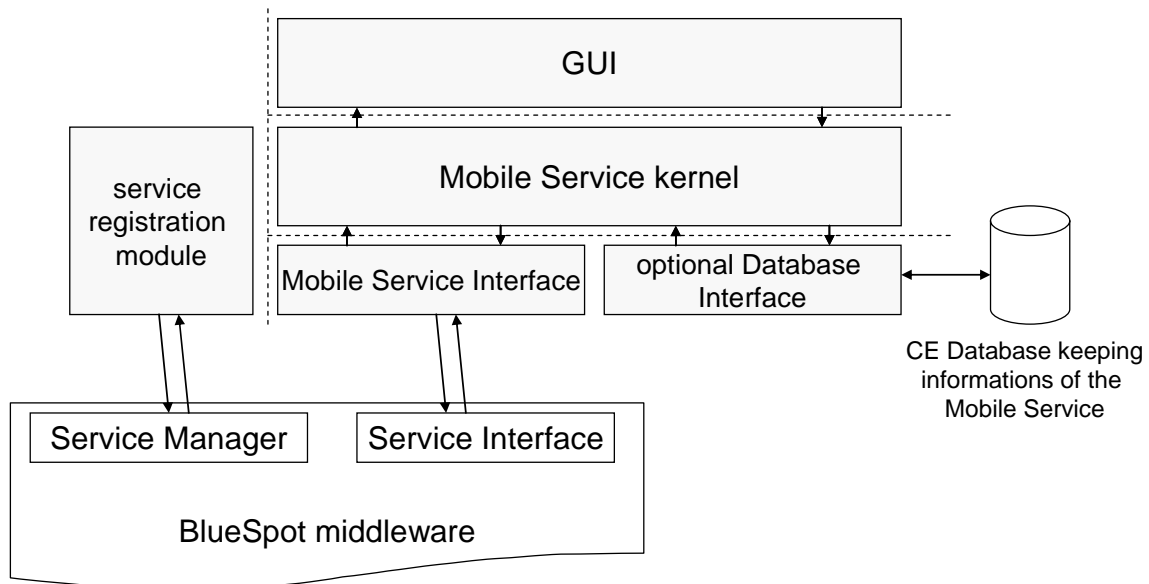


**Figure 3: the structure of the BlueSpot middleware**

The lowest layer is the Network Layer. It performs all network activities via the connected adapters, e.g. it builds up a connection, closes a connection and sends or receives data. One layer above is the Communication Layer. It inherits an abstraction of the information about existing connections to other devices from the Network Layer and manages them. With this information it can provide the upper layer, called Protocol Layer with the currently established connections. The Protocol Layer is the most important layer. Within this layer a user-defined protocol is implemented which offers several kinds of message types. These messages can be sent to other communication partners. A small selection of possible message types is `msgConnectDevice`, `msgDisconnectDevice` or `msgSendData`. These messages are used to provide an exactly defined way to communicate with other devices. In the future the BlueSpot middleware will be available to other operating systems like Symbian OS [Sy04], so a standard communication protocol is necessary. Above the Protocol Layer sits the Service Manager and the Service Interface. Both are needed to enable the communication with the Mobile Services. The Service Manager is able to locate the Mobile Services available for the device. The manager registers a found service and starts it. The Service Interface is used to enable the communication between the Mobile Services and the BlueSpot middleware. The communication is bidirectional since it is necessary to notify the Mobile Service of new available information and to let the service send information to other participants of the BlueSpot system like the Location & Information server. It must be kept in mind that a Mobile Service can only be used if both sides of communication own the Mobile Service code package and this package is registered by both sides and started by the Service Manager. This is the reason why it was important to be able to include a new Mobile Service on a device without the need to restart the BlueSpot middleware. The following paragraph about Mobile Services describes the way this procedure was realized.

## Mobile Services – Integration and Usage

Mobile Services are located in the system as a dynamic link library (DLL) which is registered in the BlueSpot middleware via COM. COM stands for Component Object Model [MS04]. By dint of COM a Mobile Service can be registered globally in the registry of the Windows CE system. The Service Manager of the BlueSpot middleware searches a predefined directory for the corresponding Mobile Services DLL. When a new service has been found the manager calls its registration method. After a successful registration COM enables the creation of an instance of the Mobile Service and simultaneously provides a marshaller for address translation. So the Mobile Service can be started and used like a standard object as if it were placed in the middleware. In order to support all kinds of components of the BlueSpot system like a BlueSpot, a Client, or an Updater, all functions must be packed into one DLL. By this the Mobile Service can be made available to another device which copies the DLL containing the Mobile Service. The BlueSpot middleware on the new device is running in the corresponding mode and uses the functions offered by the Mobile Service. An example structure of a Mobile Service is shown in Figure 4.



**Figure 4: example structure of a Mobile Service**

The service registration module consists of the methods needed to control the Mobile Service. These are the registration method, the deregistration method and a method to obtain the representing name of the service. The Mobile Service Interface contains the implementation of the communication interface to the BlueSpot middleware. It provides methods which can be called by the Mobile Service kernel. This kernel represents the main part of the service. If the service needs to store data for offline accessibility it can use the existing Windows CE Database. Therefore an optional Database Interface is available.

## Implemented Mobile Services

Today two implemented Mobile Services exist. The first service offers magazines with articles. A web based application called 'Redaktionsystem' enables authors to create magazines and articles. Clients can receive these articles with the aid of the BlueSpot system and the *BlueSpace* Mobile Service. The content of the articles is transmitted as HTML, so the

viewer needed is the Pocket Internet explorer coming together with Windows CE .Net. All magazines and articles are stored by the Location & information server. If the BlueSpot system is run in Backbone Mode the requested articles are directly routed to the Client by the BlueSpots. In Backboneless Mode the Updater makes a copy of all available articles and sends them to all the BlueSpots which store them in their Database. If a user requests an article the connected BlueSpot looks for the content in the copy of the dataset.

The second available Mobile Service is a positioning service called *BluePos*. By means of this service a user can define his position within the BlueSpot system. The service shows a map of the whole area and marks the position of the connected BlueSpot with a red dot. The map is available as a bitmap file which can be transmitted to the user. The positions of all BlueSpots are kept in form of X and Y coordinates in a database on the Location & Information server. If the system is run in Backboneless Mode the map and the coordinates are sent to the BlueSpot by the Updater. The BlueSpot stores them and makes them accessible to any Client.

At the moment several further Mobile Services in development. The most interesting approach is to enable users to communicate with others by voice telephone. For that purpose the speex codec developed by the Xiph group [Xi04] could be used. This codec was implemented to encode and decode voices and to transmit them via internet. The main area of application of this codec is Voice over IP. But also the BlueSpot system could be an interesting domain. Of course this Mobile Service will be available only in Backbone Mode.

## Resume

There is still a lot of work to do. First of all it is necessary to implement further Mobile Services to make the BlueSpot system more attractive. Another construction site is the migration of the middleware to other operating systems like Symbian OS. But also a Java version is planned. The aim is to support as many different hardware platforms as possible. But with the migration to operation systems other than Windows CE, the interfaces where the Mobile Services can be connect to must be newly specified. This is because COM is not and will never be available on e.g. Symbian OS. Even the underlying programming language will be different from C++ like the support of Java. So our further efforts will be to find a way to implement Mobile Services in an abstract language and to let them be compiled by the target device on the fly before using them.

## References and related Work

- [Wid04] Website: Widcomm Inc., <http://www.widcomm.com/>, 2004
- [Trk04] Website: Trekstore Deutschland, <http://www.trekstor.de/slimpc.shtml>, 2004
- [Sy04] Website: Symbian Ltd., <http://www.symbian.com/>, 2004
- [MS04] Website: Microsoft Resources, <http://www.microsoft.com/Com/resources/comdocs.asp>, 2004
- [Xi04] Website of the Xiph group offering the speex codec, <http://www.speex.org/>, 2004

Diploma thesis: Ulrich Dümichen, Title: Personalisierte Informations- und Positionierungsdienste für mobile Endgeräte auf der Basis von Bluetooth und CE .Net, <http://www13.in.tum.de/personen/duemichen/index.shtml>, 2004

Other approaches:

Information Society Technologies (IST) Programme, AmbieSense, <http://www.ambiesense.com/>, 2004

Institut für Kommunikation und Navigation, Heywow, <http://www.heywow.de/>, 2004