University of Stuttgart
Research Group Nexus

Final Report of Design Workshop (10.-15.7.2000)

Daniela Nicklas (Editor)

v2.1 (final)

Participants and Authors:

Martin Bauer ....IPVR (VS)
Marco Boldt ....IND
Peter Coschurba ....IPVR (VS)
Bernd Gloss...IND
Matthias Großmann ....IPVR (AS)
Christian Hauser...IND
Darko Klinec ....IFP
Uwe Kubach...IPVR (VS)
Alexander Leonhardi ....IPVR (VS)
Jens Messmer...IPVR (AS)
Daniela Nicklas..IPVR (AS)
Monika Sester ....IFP
Steffen Volz..IFP

Table of Contents

1 Introduction

There's no sense in being precise about something when you don't even know what you're talking about.
(John von Neumann)

1.1 About this Document

This document is a report of the Research Group NEXUS that is carried out at the University of Stuttgart.

The goal of this research group is the examination of concepts and methods for the support of location-aware application with mobile users. It will lead to the development of a global platform for spatially-aware applications. The main purpose of the NEXUS infrastructure is to maintain a distributed, dynamic model of the real world which is augmented by virtual objects.

The working party "architecture" held a meeting from 10th to 15th July to discuss the basic concepts of the platform design.

This document is the report of this week. It still contains inconsitencies and a lot of open issues (but - hey - it's a research group, would be bad if there weren't be any), but it conserves the status of discussion and is therefore a point from which we can go on.

Some parts of this document are in German. Because this is an internal work draft, we wanted to spend only as much effort as necessary in writing and so we didn't translate chapters which were in German already.

1.2 Process of the Design Workshop

The group started with the presentations of the previous work and the current understanding of the design.

After the discussion of the overall architecture (Chapter 4 “Overview of the Architecture” on page 21) we went deeper in the functions of the components (Chapter 5 “Components of the NEXUS Platform” on page 23).

We verified the architectur using different scenarios (Chapter 6 “Scenarios” on page 45).

Another topic was the Augmented World Model, which is the integrated view on the Augmented World. A presentation took place about the requirements of the model from previous work and a continuous collecting of requirements of the model (Chapter 3 “Augmented World Model” on page 13).

Security is an important design goal for the NEXUS Platform. Nevertheless, in this early stage of discussion, we only scratched the surface of this topic. A thouroughly analysis of security and privacy issues will follow. In Chapter 7 “Security” on page 47, the current state of the discussion is presented.

Since the NEXUS Project is a research project funded by the DFG, we want to show our sponsors the work in a prototype which we call the demonstrator. At the last day of the workshop, we discussed which parts of the architecture will be shown in the prototype (Chapter 8 “Demonstrator” on page 49).

The last day also covered topics of project organisation and scheduling, since there is no project manager doing this work. Chapter 9 “Project Plan” on page 53 and Chapter 10 “Organisation of the Project (German)” on page 55 are documenting this.

The last chapter of the document is called “Open Issues”. It covers all those questions that came up during the discussion and could not be finally answered yet. This chapter is a snapshot of a project internal document called "Liste offener Fragen (LOF)" (list of open questions) with this content.

1.3 The Vision of NEXUS

The goal of this research group is the examination of concepts and methods for the support of location-aware application with mobile users. It will lead to the development of a global platform for spatially aware applications, whose main purpose is to maintain a distributed, dynamic model of the real world augmented by virtual objects.

The real world objects will have to be stored in distributed, spatially enabled databases. Since different applications need different resolutions of digital data, multiple representations of spatial objects have to be provided.

Virtual objects are associated with a geographical position or a certain real world object in the Augmented World Model and contain links to existing information spaces like the WWW. For a convenient access to this platform a basic communication service with a high availability is required, which integrates different mobile communication systems, provides a management of the quality of service and has the ability of geographic addressing.

The support of disconnected operations permits an interaction with the platform even in areas with low or temporarily unavailable network connection. By integrating different sensor technologies (e.g. GPS and inertial systems) in combination with image analysis, the platform will be able to gather as much information about the position and environment of its users as possible.

For an efficient access to the model and the various information spaces attached to it, suitable techniques for the storage of the model and the optimized execution of spatial database queries will have to be developed. An important aspect is the creation of new federation techniques, as the special information databases developed here will have to be integrated with existing information sources.

2 Use Cases

2.1 Purpose of this chapter

The following use cases are partial functionalities of applications based on the Nexus platform. A scenario would consist for example of a sequence of concrete instances of these use cases. For the architecture working party we should commit to a series of use cases which we want to support with the Nexus platform and which form the foundation for a further determination of the Nexus platform’s functionality. They can help to identify missing or perhaps redundant functionalities. In the first instance a use case could later be removed while in the latter it could be added or modified.

2.2 Use Cases

As a basis for discussion we have collected several use cases we can imagine for the Nexus platform:

Usecase 1: Passive environment information

Importance:
Fundamental

Specification:
A user moving through a given environment (e.g. city, trade fair, shopping mall) is being shown a changing list of topics (VITs) relevant for his current geographical position and matching his user profile (e.g. in Stuttgart about the "Neue Schloss" or the market place). See VIT prototype.

Source:
IPVR.VS

Usecase 2: Active environment information

Importance:
Fundamental

Specification:
A user wants to be notified when passing by a sports store.

Source:
IPVR.VS

Usecase 3: Determination of a user’s current location and movement vector

Importance:
Fundamental

Specification:
A user requests a specific user's location and possibly also the user's speed and direction.

Source:
IPVR.VS

Usecase 4: Navigation

Importance:
Fundamental

Specification:
Navigation is a recurring subject in our scenarios. A user wants to know how to get from his current location to a destination point. Alternatively he might want to know how to get from a given starting point to a destination. Since navigation is a very complex interaction it might be sufficient just to supply the application with a path from the starting point to the destination. The visualization or the "right/left" instructions, respectivly, can then be built by the application itself.

Source:
IPVR.VS

Usecase 5: GeoMail

Importance:
Fundamental

Specification:
There are tickets available for a sold out opera performance. The theater wants to send a message with this information to all tourists in downtown Stuttgart.

Or there is an accident on the A8. All traffic on the A8 and on roads leading towards it have to be informed.

Source:
IPVR.VS

Usecase 6: Querying the service in an area

Importance:
Fundamental

Specification:
A user would like to receive a list of all Italian restaurants in downtown Stuttgart. Other possible queries would be mailboxes, post offices, ATMs and so on.

Source:
IPVR.VS

Usecase 7: Retrieving information by pointing at objects

Importance:
Strongly desired

Specification:
A user can receive information (e.g. a web site) about an object (e.g. a building) by pointing at it with an appropriate device (telepointer).

Source:
IPVR.VS

Usecase 8: Location-aware Memo

Importance:
Strongly desired (corresponding to use case 2)

Specification:
Shopping list at the supermarket. When something one wants to buy comes to one's mind it can be added to a shopping list. When the appropriate store is being entered or approached, the user receives a reminder of the items on the shopping list.

Source:
IPVR.VS

Usecase 9: Redirecting of communication requests to a terminal device close to the user's current location

Importance:
Strongly desired

Specification:
When there are attempts to communicate with a user (by telephone, fax, E-mail or by other means) the system tries to bring the attempt as close as possible, or, if there is more than one device, as appropriate as possible, to the user (assumed he has allowed these requests). When a call is being received the system tries to redirect it to a phone close to the user. If no phone is available, the user is being notified of the call for example by mail and is then able to call back or reply to the caller with a (speech-generated) message and so on. Asynchronous communication is being stored and can be retrieved by the user on any terminal device. Different setups allow for the system's features to be used without interaction.

Source:
IPVR.VS

Usecase 10: Virtual warning sign/triangle

Importance:
Desired

Specification:
If an accident occurs on a freeway, a virtual warning sign can be placed in a certain distance on the lane leading in this direction. All vehicles moving towards the scene of the accident are being informed about the accident by their Nexus-enabled vehicle navigation system and can either be warned to reduce the speed or asked to leavdisplaye the freeway.

Source:
IPVR.VS

Usecase 11: Virtual ticket

Importance:
Desired

Specification:
A Nexus user buys his admission ticket for a trade fair at a virtual ticket office. When he enters the trade fair, the system checks for a valid ticket and informs the staff if he does not possess one. This can be repeated for subsections of the trade fair. Food and drinks will be automatically deducted from his account.

Source:
IPVR.VS

Usecase 12: Display plumbing/wiring

Importance:
Desired

Specification:
The application is supposed to project the wires or plumbing onto an augmented reality display enabling the user to view the corresponding pipes/wires when looking at a part of the wall or the ground.

This use case has implications on the precision of the sensors and data's level of detail rather than having new demands on functionality.

Source:
IPVR.VS

Usecase 13: Control of real objects via their model representation

Importance:
Strongly desired

Specification:
A change in the state of an object in the model world can trigger a change of the corresponding object in the real world and vice versa. A user would for example use his PDA to control a model train or to program the VCR.

Source:
IfP

Usecase 14: Change Object

Importance:
Fundamental

Specification:
An authorized user realizes that the color of his house has been modified and changes this in the model so that the model is consistent with the real world.

Source:
IfP

Usecase 15: Identify Object by image processing

Importance:
Desired

Specification:
A user transmits an image of a real-world object. The image is being matched with the augmented world and if possible the object is being identified.

Source:
IfP

3 Augmented World Model

3.1 Introduction

In NEXUS, the Augmented World Model plays a major role. All of the data used by more than only one NEXUS component is delivered by the Augmented World Model. The Augmented World Model is nothing else than an abstract formation of hierarchically structured objects. The components access that data by using the Augmented World Model Language (yet not specified), which describes the objects in a machine understandable manner. The intention of the following sections is to provide an overview over requirements, major problems, and some important considerations concerning the Augmented World Model.

In section two of this chapter, the integration of the Augmented World Model is discussed. Therefore, the interface to the NEXUS components communicating with the Augmented World Model are examined more closely.

In section three of this chapter, one major difficulty, namely linking objects with different detail levels, is considered. Methods are needed for linking entrance doors of buildings to the corresponding entrances on the outside of the building. But in the representation of the different views (outside and indoors) different scales are used.

In section four, a list of significant attributes which are common for all NEXUS Objects is provided.

In section five, there is an accumulation of requirements concerning the Augmented World Model.

3.2 Interface between Global Federation and Augmented World Model

In the NEXUS architecture (see figure below) the Augmented World Model is placed at the same level as the global federation. Thus, the communication seems to be performed mainly by the federation. This could lead us to the possibly erroneous assumption that there is only one interface between global federation and AWM.

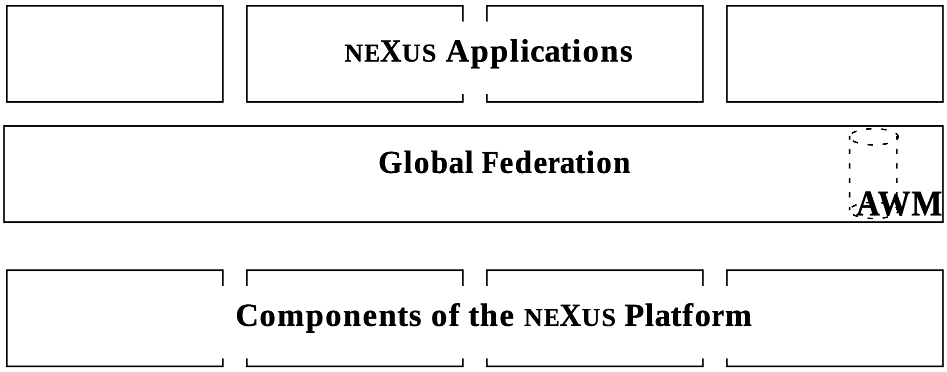


Figure 1: The logical place of the Augmented World Model

In fact there are also some other quite relevant possible interfaces which could refer to the AWM.

First of all, the SpaSe (Spatial Model Service) Component handles all the 2D, 2.5D and 3D information of static objects. The exact graphical representation that one needs for map creating e.g. is completely stored on a SpaSe. That means, when a user starts a query for a route from Stuttgart Hauptbahnhof to Killesberg Messe (compare use case 3 and 4) the LS (Location Service) transfers the localization information to the global federation that demands the corresponding mapping data from the SpaSe. Up to this point AWM itself doesn't seem to be necessary at all. Assuming that there are lots of one way streets on the way from Stuttgart Hauptbahnhof to Killesberg Messe and that our navigation tool should be able to deliver a textual route description as well, we definitely need a place where traffic and transportation information is stored. The AWM is able to deliver information like this. But the query for this information has to be started from the Global Federation (the global interface between all components), and will be handled in the Spatial Model Server, because only this component possesses the necessary algorithms for pathfinding and navigation.

Other possible use cases (2, 3 e.g.) implicate the necessity of querying AWM information from the LS component. The best solution is to distribute the AWM information to the responsible components. This architecture results in the following picture:

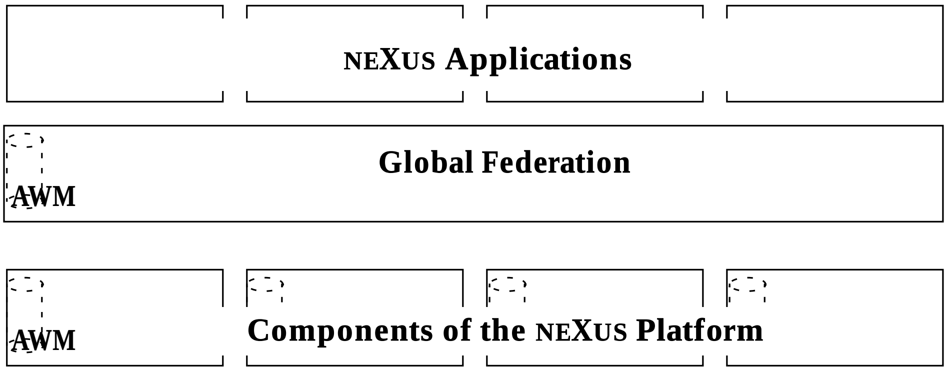


Figure 2: Distribution of different AWMs over the NEXUS Platform

The distribution of the NEXUS objects can be implemented as follows: due to the hierarchical structure of the AWM and the NEXUS objects respectively, the type of the objects (mobile or static) can be determined at depth 4 of the NEXUS Object Tree (as shown below).

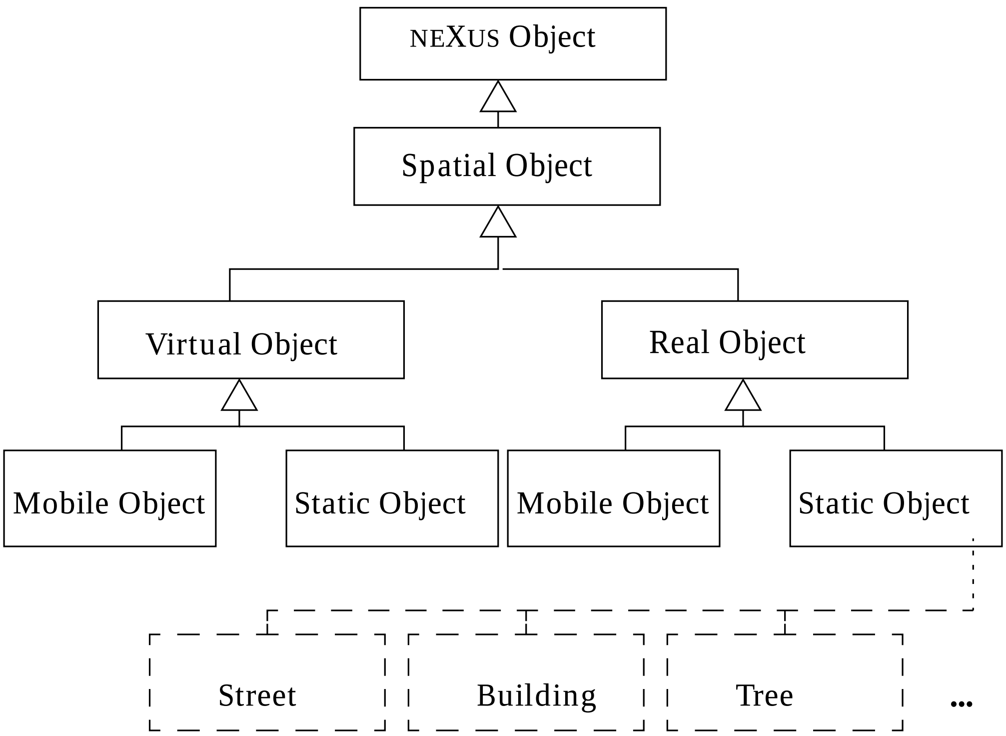


Figure 3: Basic Inheritance of the AWM

The unsolved question of how to access the AWM data remains. If the queries were performed in SQL99, we would need a special architecture layer which translates the query into AWM understandable syntax of the implementation language of the AWM objects. Another solution is to use data structures that are closely related to the implementation language of the components, java classes for a java implemented SpaSe for instance.

3.3 Linking NEXUS Objects residing on different detail levels

Another interesting issue is the question about object identifiers and hierarchical structures of NEXUS Objects. The need for hierarchical structures mainly occurs by linking different models of different detail levels. The problem is to find entry points for models of lower detail level that are derived from models of higher detail level. To give an example: you have a building with lots of rooms all related to this building. The rooms consist of doors that link rooms with halls and so on. The hierarchical structure is as follows: **building - room - door**. You can describe this relation in reverse order by using a *is_part_of relation* (for a better understanding **objects** are printed in bold an *relations* in italic).

Now imagine that you are in an office of a company with two separate buildings **A** and **B**. In NEXUS the indoor data for buildings is stored on another detail level than the data for city maps. There is also a difference in the resolution of the different detail levels. That means if you want to have the route from office **A.176** in building **A** to office **B.453** in building **B**, you have to switch the detail level twice on your way from one office to the other. To do so, you have to follow up the ***building*** *-is_part_of->* ***city*** relation and you need an entry point for the city map.

One possible solution for storing the entry point could work like this: on detail level city to store not only buildings, streets etc. but also entry points like doors that are strictly speaking elements of a higher detail level. Another solution would be a little bit more inexact by handling the whole building as an entry point. But this will only work for small buildings with only one or two entrances. In any case you need an additional link object for building directional or non-directional typed relations.

A third, but, rather unhandy solution would be to handle links by deducing the kind of relationship out of the NEXUS Object identifier. In this implementation one would lose the ability of setting up typed links and directional links. All links would be of the same kind.

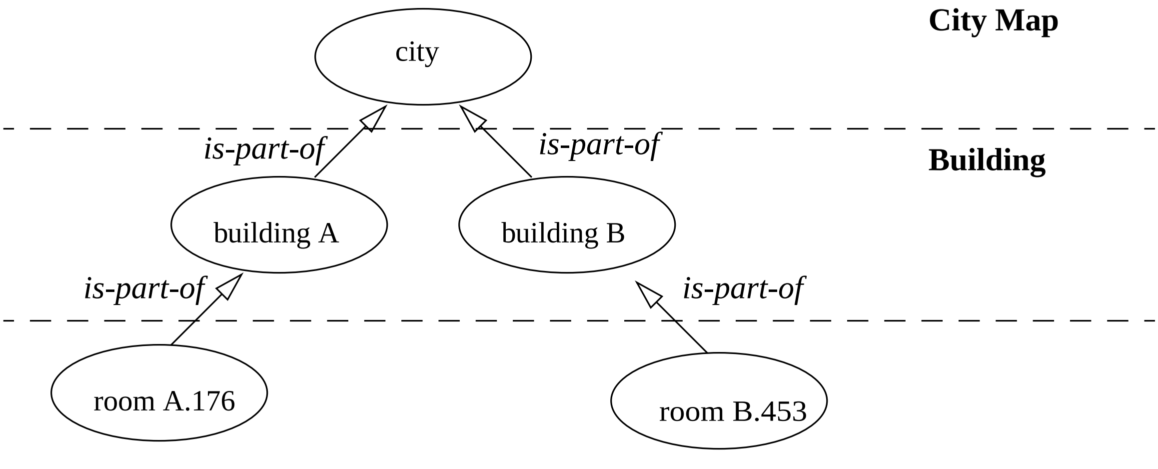


Figure 4: object hierarchy of rooms linked with buildings

Let us have a closer look on the second solution: by using link objects between models of different detail-level, typed links have to be set on typed model objects. We want to find a generic way for setting up links between different detail-levels. In our example with the two offices **A** and **B** there is an object called **building A** and a relation *is_part_of* **model** of detail-level building. Then, there is a second object called **surrounding area** and a relation *is_part_of* **model** of detail-level city-map. Last, there is the third object named **building B**. Another *is_part_of relation* between object **model** of detail-level building and object **model** of detail-level city-map provides us with the following transitive relation: **office A.176** *-is-part-of->* **A** *-is-part-of->* **model** detail-level building *-is-part-of->* **model** detail-level city-map.

This is one possible solution for handling different detail-levels in AWM. Maybe there also is a solution by omitting the model object. But this depends on the requirements of the already existing SpaSe components and the planned data handling.

3.4 Attributes in the Augmented World Model

This chapter gives an impression of the AWM's major object attributes particularly with regard to default values and necessity.

Attr. Id	Attribute Name	Mandatory	Default	Comment
A 1	object id	Y		primary key
A 4	graphical representation	Y	geometry(extension, position)	extension and position are one way (in this case the default) to describe an object's representation extension and position are dependant on the detail level (map scale)
A 5	object class	Y		class name respectively type (class attributes can be deduced by following the object tree structure, e.g. virtual/real)
A 7	linkable	N		AWM requirements AW11[a]
A 8	scope	N		AWM requirements AW23[b]
A 9	usable by	N		AWM requirements AW25[c]
A 10	importance	N		similar to A7 but expressed by a continuous function (requirements AW19+23)[d]
A 11	class specific attributes	N		attributes that are specified by the class type, e.g. paper is a material you can use for writing

a. AW11: Links between objects of different detail level must be possible
b. AW23: Scopes (e.g. objects that are only visible in certain areas or at certain periods of time)
c. AW25: Usability for objects has to be specified (e.g. streets are usable by mobile objects with wheels)

d. AW23: Modelling of functional dependencies between different objects

3.5 Requirements to the Augmented World Model

In the current version the list of requirements is intended as base for discussions because many requirements are not very precise. Some of the requirements are the result of the NEXUS Use Cases which have to be looked up in the corresponding documents.

R.#..............................number of the requirement (will never be changed)
Type............................type of requirement, there are the following types:
Fnk ....functional requirement
NF ....non-functional requirement (Usa, Sec, Tec):
Usa. ....Usability
Sec......................Security
Tec......................Technical
Source ....source of requirement (paper, partner, initials of project members)
UC #...........................number of the use case which makes this requirement necessary. If there is no number provided, this requirement may be deduced by all use cases
Date............................creation date of the requirement

R.#	Type	Description	Source	UC #	Date
AW1	Fnk.	the AW contains real and virtual objects.			
AW2	Fnk.	the AW contains static and mobile objects.			
AW3	Fnk.	The mobile objects in the AW will have information about their location, their speed an their direction.	VS	3	14.6.00
AW4	Fnk.	in the AW, virtual objects can be connected to other static or mobile objects (e.g. virtual PostIt)		1,10	17.5.00
AW5	NF	extensibility and maintainability of the AWM (primarily important for the AWML)	AS		5.7.00
AW6	NF	flexibility of the AWM: any insertion of new NEXUS Objects should be feasible	JM		5.7.00
AW7	Fnk.	scalability of NEXUS Objects (i.e. users, locations, objects etc.)			5.7.00
AW8	Fnk.	access restrictions depending on user groups			6.7.00
AW9	Fnk.	querying the NEXUS infrastructure leads to optimization of transmitting information	VS, IND		6.7.00
AW10	Sec.	anonymity of users	IND		6.7.00

R.#	Type	Description	Source	UC #	Date
AW11	Fnk.	links between objects of different detail level must be possible		3,4	6.7.00
AW12	Fnk.	integration of differently detail-levelled models		3,4	6.7.00
AW13	Tec.	methods for creating objects and for changing their state	DN, JM		5.7.00
AW14	Fnk.	querying the class schema	VS	1,2,6,7	10.7.00
AW15	Fnk.	modelling of uncertainty concerning the exact localization of mobile objects depending on sensory resolution	DN, IFP		5.7.00
AW16	Fnk.	non-ambiguous NEXUS Object identifiers			11.7.00
AW17	Fnk.	relative coordinate system (needed for mobile objects which provide indoor functionalities, like navigation inside trains)	PC		13.7.00
AW18	Fnk.	geometry should be available in 2D, 2.5D, and 3D	IFP		13.7.00
AW19	Fnk.	modelling of functional dependencies between different objects	DN	5,7,8, 9,13, 14	5.7.00
AW21	Fnk.	different hierarchical structures of NEXUS Objects (an object linked with several others viewed as subobjects, e.g. room with doors, windows, tables etc.)			5.7.00
AW22	Fnk.	it should be possible to trigger events depending on conditions linked to objects	VS	2,5,8, 9,10, 11	13.7.00
AW23	Fnk.	scopes (e.g. objects that are only visible in certain areas or at certain periods of time)	VS	1,2,5, 7,8,10	
AW24	NF	objects have mandatory or non-mandatory attributes which can have default values			13.7.00
AW25	Fnk.	usability for objects has to be specified (e.g. streets are usable by mobile objects with wheels)	JM		10.7.00
AW26	Fnk.	aggregations of objects do possibly need a new id which determines the ids of the aggregated objects	VS		11.7.00
AW27	Fnk.	links between objects can be typed	DN		20.7.00
AW28	Fnk.	the modelling of the spatial attributes must follow the OPEN GIS-Standard	IFP		20.7.00

4 Overview of the Architecture

This chapter gives an overview of the architecture of the NEXUS-system. A more detailed description of the different components can be found in chapter five. There are three different types of hosts used by the components:

4.1 Client side

The client is the users portable terminal and the host the applications are running on. The client-sided parts of the NEXUS-system are the interface for the applications and parts of the sensor component and the communication components:

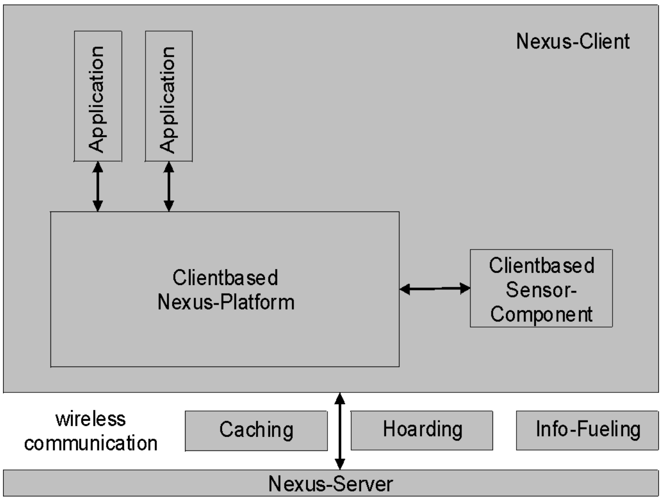


Figure 5: Client side of the architecture

4.2 Server side

The server is a host the client contacts via a wireless communication interface. The server hosts – besides the server-sided parts of the communication components – the Global Federation, which integrates the different services and provides the interface the applications see through the communication components. The Global Federation may use additional services, e.g. a Map Service to transform parts of the Augmented World Model into a graphics format, running on the same server. As these components do not keep any data themselves, the client may contact any server, regardless of the data he requests. The Global Federation has to contact other servers to get the data necessary to fulfil the client’s request.

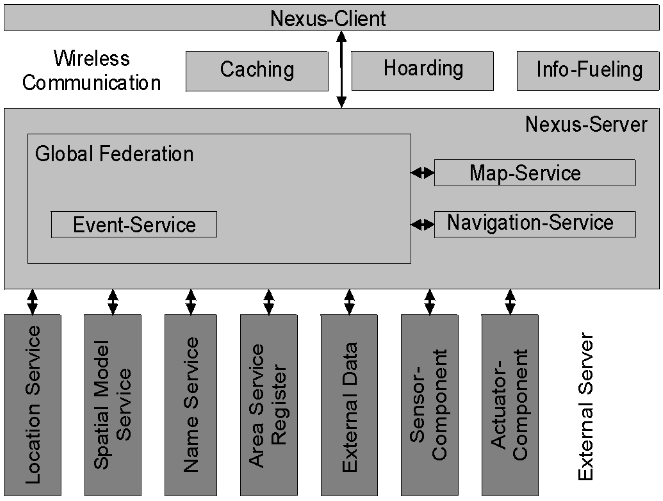


Figure 6: Server side of the architecture

These additional servers are the hosts for the data-keeping components of the NEXUS-system, which are:

- Location Service: keeps position, direction and possibly other variable data of mobile objects
- Spatial Model Service: stores information about static real-world and vitual objects
- Area Service Register: provides the information which Spatial Model Servers and Location Servers are responible for a certain area
- Name Service: maps common names of objects on unique object-ids, that can be used to query informatin about specific objects from other services

Furthermore, the Global Federation has the opportunity to contact servers that are not part of the NEXUS-system to integrate external data. In a similar way, server-sided sensors and actuators are connected to the Global Federation. The NEXUS-system also provides an Event Service to notice applications of events triggered by mobile objects and a GeoCast-component allowing messages to be sent to users within certain areas.

The wireless communication component between client and server integrates different communication services like GSM or wireless LAN and provides a consistent interface. It has the ability to opaquely switch between those services and features mechanisms to cope with the variable, comparatively small bandwidth to ensure best quality of service:

- Caching
- Hoarding: prefetching data while the user stays within an area of higher bandwidth
- Info-Fueling: delaying certain requests until the user enters an area of higher bandwidth

Every component can contact a PKI Service to authenticate servers requesting data and to check if this is allowed to get certain data.

5 Components of the NEXUS Platform

5.1 Spatial Model Service

5.1.1 Overview

The tasks of the Spatial Model Servers (SpaSes) mainly concern the management and query handling of spatial data in multiple representations within a distributed environment.

5.1.2 Function

A SpaSe has to provide operators in order to solve spatial queries, e.g Within Distance, Buffering, Intersection, etc. Furthermore, queries on networks have to be possible. To carry out these analyzes, the topology of a network has to be generated. This is a time consuming process and thus a big problem taking into account distributed network data, where topologies will have to be generated online. Moreover, building topology on differently resoluted networks will be a challenge for research. Another major task concerning the combined use of different data sets deals with the homogenization. Identical features have to be identified in the different data sets in order to be able to merge them into one, on which the processing can be performed. That also has to be done by the SpaSes. To present the results of queries visually, Spatial Model Servers have to be able to generate maps. For thin clients, only images will have to be prepared. Mobile devices with a higher computing power could also download data to their local machine and process some less demanding operations on their own.

5.1.3 Interaction with other components

SpaSes have to interact with many other components of the platform. First of all, the Spatial Model Servers have to register themselves at the Area Service Register, delivering meta information about the data they store. When a query is triggered, SpaSes receive the query parameters via the Global Federation. They return results or data extracts either to the federation or also to other SpaSes. Positioning information is obtained by communicating with the Location Service. Spatial Model Servers also generate the areas which are necessary for the Location Service in order to support Spatial Events. SpaSes act as event producers for the Event Service. Moreover, they are informed by the actuator component if the states of real world objects change. Since spatial models will be needed for object recognition within images, SpaSes also have a connection to the sensor component. In order to define, where a message has to be sent to, the GeoCast component must also provide an interface to Spatial Model Servers.

5.1.4 Responsibility for data

SpaSes are responsible for all real and virtual objects of the Augmented World which need a spatial representation. They store 2D, later also 3D data in Vector format. The data have to be available in multiple representations, concerning geometry as well as semantics. For some tasks, also the topology has to be provided. Textures of surfaces are held as raster data. Within the DBMS that represent the SpaSes, also the appropriate indexing structures for spatial data have to be provided. Concerning the thematics, all attributes that are required by the Augmented World Data Model have to be generated.

Possibly, also the user specifications or the descriptions of mobile objects, respectively, will have to be stored within Spatial Model Servers, as well as the links of objects to external information sources.

5.1.5 Distribution

Since it is assumed that the system will have to serve for large areas, also the appropriate data sets have to be stored within the platform. This requirement rises the demand for a high degree of distribution, since geographic information generally demands a lot of disk space. Distribution and common usage of distributed spatial information addresses a major research topic since there are still many problems concerning the combination of different data sets for query purposes. Especially the homogenization of different data sets (each of which possibly having a different underlying data model) and their topologies, reveals big difficulties. Query strategies will have to consider that many components of the NEXUS platform have to be involved in the processing of a spatially related task. Therefore, these strategies will have to provide sophisticated query sequences in order to guarantee an optimal response time.

5.1.6 Open Questions

It has not been decided yet, if the SpaSes will also have to deal with the management of external information or with the storage of mobile objects specifications.

5.2 Management of External Data

5.2.1 Overview

Data sources that have a spatial aspect, i.e. information that is assigned to a place, should also be included in the NEXUS Platform. It is not yed decided, if there is an extra component for the Management of External Data (called Information Service or *Informationsservice*) or not. Perhaps it is just a Spatial Model Service, which does not emphasize the spatial aspects of data but other information.

5.2.2 Function

Integration of External Data Sources to the NEXUS Platform

5.2.3 Interaction with other components

see Chapter 5.1 “Spatial Model Service” on page 23.

5.2.4 Responsibility for data

see Chapter 5.1 “Spatial Model Service” on page 23.

5.2.5 Distribution

see Chapter 5.1 “Spatial Model Service” on page 23.

5.2.6 Open Questions

Is the interface of the Information Service identical to the Spatial Model Service?

5.3 Location Service

5.3.1 Overview

The Location Service as part of the NEXUS platform is responsible for the management of the dynamic location information of mobile objects (users, car, etc.)

The Location Service manages the location information of the mobile objects in the Augmented World as accurate as possible (depending on the available sensors, network bandwidth and delay as well as on server capacity). It provides the functionality to query this location information in various ways. It can for example return the current location for a certain object as well as all objects that are currently inside a given area. For queries to the NEXUS platform concerning the location of mobile objects, the Global Federation forwards corresponding subqueries to the Location Service and integrates the results with those of other components.

5.3.2 Function

Queries can be sent to any location server and are routed internally to the responsible location server(s). The following query and event mechanisms are supported by the Location Service:

locOfObj: position query that returns the current location of a given mobile object

objAtLoc: range query that returns all mobile objects currently inside a given area

nearestObj: returns the mobile object that is nearest to a given location

events: using the Event Service of the NEXUS platform the Location Service supports a number of spatial events concerning the position of mobile objects (e.g., mobile object enters/leaves certain area)

continuous location updates: if requested by another NEXUS component, the Location Service can send continuous location updates (similar to event notifications)

Control messages:

register/deregister: registers or deregisters a mobile object with the Location Service

update: updates the location information for a mobile object

5.3.3 Interaction with Other Components

The Location Service interacts with the following components of the NEXUS platform:

Sensor component: Sensor components for location information send newly acquired location information, according to a given update protocol, to the Location Service.

Global Federation: The Global Federation extracts the elements concerning location information from the queries to the NEXUS platform and forwards them to the Location Service.

Spatial Model Server: If necessary, the Location Service can update the location information in a Spatial Model Server, periodically.

Event Service: The Location Service needs to cooperate closely with the Event Service, as the Event Service provides a part of the functionality of the Location Service and probably requires to have some components integrated into the Location Service.

Area Service Register: If other components (e.g., the Event Service) require knowledge about the internal distribution of the Location Service, it can register its servers with the Area Service Register.

5.3.4 Responsibility for Data

The Location Service is responsible for the location information of all objects, which are not permanently stored on a certain Spatial Model Server.

5.3.5 Distribution

A single server of the Location Service stores the location information for all mobile objects inside a certain area with a given accuracy. However, any Location Server can be queried about any mobile object or area. Which server is responsible for a certain mobile object can be determined internally through a Object Register that also stores additional data for theses objects. Which server is responsible for a certain area can be determined through the Area Service Register. For reasons of optimization, hierarchical structures can be introduced, where a server in a higher level is responsible for less accurate location information for a larger area.

5.3.6 Open Questions

There are the following open questions concerning the Location Service:

The way, the Location Service is distributed on a number of services is not yet decided in all detail. However, any Location Server can be queried about any mobile object or area. It is not clear, if other components need to have knowledge about the way the Location Service is distributed for optimization purposes.

An open question is also the way, the Location Service cooperates with the Spatial Model Servers. On the one hand, the Global Federation could decompose queries to the platform in elements which are forwarded to the Location Service and in elements for Spatial Model Servers. On the other hand, all queries could be sent to the Spatial Model Servers, which for mobile objects store a link to the Location Service and query the location information from the there, if necessary. In the description above, we have assumed the former alternative.

5.4 Area Service Register

5.4.1 Overview

The Area Service Register acts as a "Spatial Search Engine", storing the assignment of Augmented Area Models to Spatial Model Servers.

5.4.2 Function

The Area Service Register receives those parameters via the Global Federation, that define which spatial data sets have to be involved in the processing of a query. Generally, the most important criterion in order to find a spatial data set is the definition of the spatial extent that the query demands. For this reason, the Global Federation must calculate this information before it addresses the ASR. But, of course, it also has to be specified which resolution and which object classes are needed. Within the metadata, information about the projection, etc. has to be given, too. Furtheron, details on the relationship of data sets must be provided.

In a first step, the Area Service Register finds all the servers that are responsible for the specified area. Then, by perfoming a search on the metadata, it chooses, which data sets or which servers, respectively, are most appropriate for the task. It also has to determine, if the interoperability of the different data sets can be guaranteed. Finally, the addresses of the servers are transmitted to the Global Federation.

5.4.3 Interaction with other components

At first, the ASR receives the registrations of the SpaSes. If there is an update of a SpaSe which is of relevance, the ASR has to be informed. SpaSes lose their connection to the ASR if they are not used within a predefined period of time. During a query, the component communicates with the Global Federation, on the one hand receiving parameters for the search of data, on the other hand returning locations/addresses of data sets to the Federation. It could also be possible that other components directly address the ASR, thereby avoiding the Global Federation.

5.4.4 Responsibility for data

The Area Service Register stores the assignment of spatial data to Spatial Model Servers. Also the metadata of the different data sets are saved, including remarks on the relationships (spatial overlap, spatial refinement, etc.) between them.

5.4.5 Distribution

In the first phase, a distribution of the Area Service Register is not planned.

5.4.6 Open Questions

It has not been decided yet if the Area Service Register is belonging to the Global Federation or if it is an extra component.

5.5 Sensor Component

5.5.1 Overview

The Sensor Component provides a connection between the real and virtual world. For this purpose different sensors are utilized. The measurements of these sensors are processed and provided to other components. For example the actual position of a mobile user can be processed using the different measurements. In this way it is possible to provide these measurements to the NEXUS components to enable services of higher order.

5.5.2 Function

The Sensor Component primarily provides positioning information, which is extracted from the different positioning sensors. Until now the following sensors are used:

- DGPS (GPS)
- Digital compass
- Pedometer
- Digital camera
- …

Directly measured as well as derived coordinates are available. DGPS provides coordinates directly with this sensors. The other sensors yield different measurements and the position only can be processed indirectly. Position information has to be provided in indoor and outdoor areas. At first only outdoor areas will be supplied with positioning information. The quality of the positioning depends on the used sensor.

In addition to the positioning sensors which provide location information at any time and different quality, there also exist sensors which are able to record the environmental factors, e.g. temperature, image information, etc. .

5.5.3 Interaction with other components

The Sensor Component interacts with the following components of the NEXUS platform:

Location Service: Transfers (periodically) position information to the Location Service.

Global Federation: Communication with the Global Federation to provide interaction with other components.

or: Direct communication with the NEXUS components (‡ v. open questions).

SpaSe: Provides spatial models for the object recognition within images, which is a part of the Sensor Component.

5.5.4 Responsibility for data

The individual sensors of the Sensor Component are responsible for the data by themselves.

5.5.5 Distribution

Distribution is possible.

5.5.6 Open Questions

Which privacy aspects must be considered in combination with the Sensor Component?

Further sensors for the Sensor Component?

Interaction with other components: Only communication with the Global Federation or direct communication with the NEXUS components?

5.6 Actuator Component

5.6.1 Overview

Actuators are elements, which control the environment in the real world. If there is a change in the model or there is changed something well direct, so the Actuator Component effects the action in the real world. Examples for Actuator elements are control of luminance in a room, the temperature, etc. .

5.6.2 Function

Actions in the model effect a reaction in the real environment. Changes can be undertaken in the following way: An adjustment or change in the model initiates a reaction in the environment, e.g.

- controlling of luminance
- controlling of heating
- open/close doors
- …

5.6.3 Interaction with other components

Global Federation: The Global Federation extracts events (if necessary in combination with the Event Service) and changes in the model (which can be controlled by the Actuator Component). The communication between the Actuator Component and the Global Federation causes, that changes in the model are transferred in the real environment. Executing the event by the corresponding Actuator element does the transfer.

SpaSe: The Actuator Component informs the SpaSe, if the states of the real world objects change.

5.6.4 Responsibility for data

The Actuator Component controls the status of the environment, if an event occurs. It has no responsibility for data.

5.6.5 Distribution

none

5.6.6 Open Questions

It has not been decided yet which Actuator Elements should be used.

5.7 Name Service

5.7.1 Overview

The Name Service (NS) allows the retrieval of the NEXUS-Object ID for a specific object.

5.7.2 Function

The name service provides the appropriate Object ID for a given name.

5.7.3 Interaction with other components

Global Federation: The application will direct a request to the Global Federation. The Federation will forward this request to the name service. It also will forward the answer to the client application.

Other Components: It might be that there is a interaction with the SpaSe in order to retrieve the Object ID. The same might be true for the Location Service.

5.7.4 Responsibility for data

The Name Service is responsible for the data it needs to connect the name with an object ID.

5.7.5 Distribution

At least for the prototype there will be no distribution.

5.7.6 Open Questions

It is unclear what kind of service the name service provides and how these functions are realised. Especially the following questions are not answered yet:

- How does a name look like? (What naming scheme will be used)
- How is the connection between name and ID built?
  - Automatic: There could be an automatic mechanism that retrieves the Object ID from the Spatial Model Service.
  - Manually: Someone (the user?) has to enter the name and the object-id once. The remaining question is, how the user retrieves the id in the first place.

5.8 Global Federation

5.8.1 Overview

The Global Federation provides the interface of the NEXUS platform to the NEXUS client and the logical place for the Augmented World Model. It generates the strategies for distributed queries.

5.8.2 Function

(Spatial) Queries. The NEXUS client can query the NEXUS platform to get information about the Augmented World. Since the information is distributed over different Spatial Model Servers and Location Servers, the Global Federation must analyse the queries and distribute it over the components.

(Spatial) Events. The NEXUS platform also supports spatial events. An important part of the Global Federation is the Event Service. This component is described in Chapter 5.9 “Event Service” on page 31.

Integration. The information that is gathered from the components is integrated to a combined view (an object or a part of the Augmented World). This also includes the generation of maps.

Prognosis. The Global Federation is able to give predictions about the size of the result of a query.

Data Management. The Global Federation provides an integrated data management interface for objects of the Augmented World.

5.8.3 Interaction with other components

The Global Federation has possible interactions with all other components.

5.8.4 Responsibility for data

The Global Federation has no persistent memory.

All data is distributed among the Spatial Model Service, the Location Service and the Area Service Register.

For the NEXUS-Client, the GloF is responsible for the Augmented World Model.

5.8.5 Distribution

There is a Global Federation component on every NEXUS node.

5.8.6 Open Questions

Registry of Applications – does the Global Federation know which NEXUS clients are connected?

5.9 Event Service

5.9.1 Overview

The Event Service provides event-based commincation for the NEXUS platform. It is responsible for the registration of predicates describing events, the observation of the events described by the predicates and the notification of the interested NEXUS components that the event has actually occurred.

5.9.2 Function

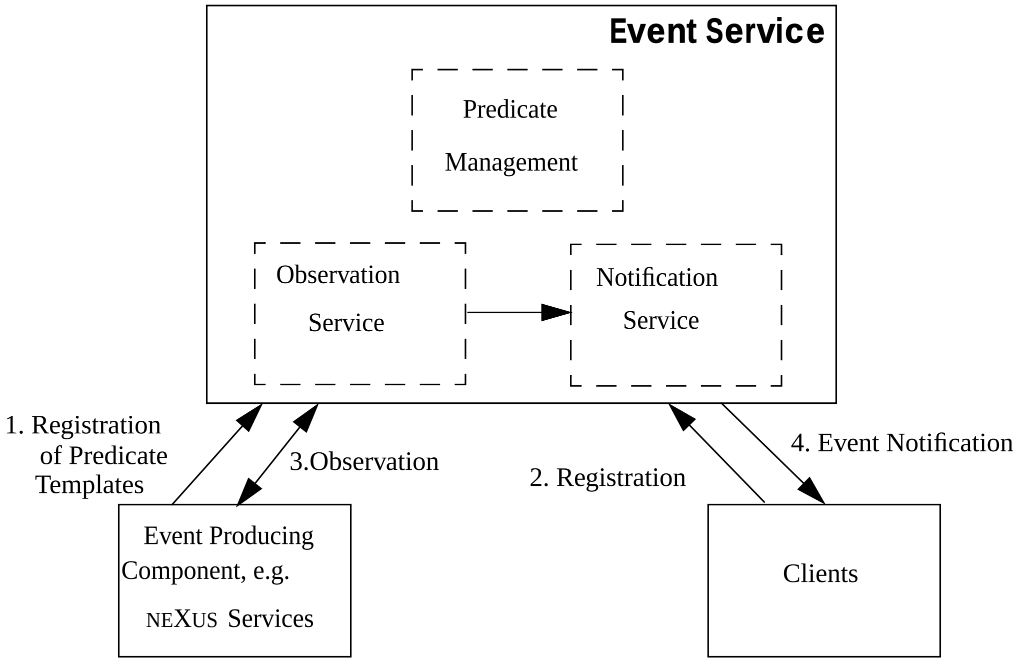


Figure 7: Architecture of Event Services

Registration of Predicate Templates

Event producing components (e.g. the Location Service) register predicate templates with the Predicate Management Component. Predicate templates are non-instantiated descriptions of events that can be observed by the event producing component. The predicate template contains variables that have to be instantiated when an event is actually to be observed. For example, the location service offers an onEnter event. The onEnter event has two varaibles, the area and a description of the objects that may enter the given area to make the event occur.

Registration of Events

NEXUS components that are interested in receiving event notifications for a particaular event, register the respective predicates with the Event Service. The predicates are instantiations of the predicate templates. For example, a component might register a predicate onEnter(CS department, professor).

Event Observation

The Observation Service component of the Event Service registers basic events with the event producing components and is responsible for observing higher level events, e.g. combinations of different basic events.

Event Notification

The Event Notification Service component of the Event Service is responsible for delivering event notifications about events that have occurred to the NEXUS components interested in the particular event.

5.9.3 Interaction with other components

Basic Events

NEXUS services provide basic events to the Event Service, e.g. the Location Service (onEnter, onLeave, onMeeting, ...) and the Spatial Model Service (onAddPostIt, ...)

Plug-in trigger

Plug-in trigger contain code for the observation of events. The NEXUS services provide an interface that allows the trigger code to observe complex events in the interior of the respective NEXUS Service.

Communication Infrastructure

The event service makes use of the NEXUS communication infrastructure, e.g. unicast, multicast or the Geocast component.

NEXUS services can also use the Event Service

5.9.4 Responsibility for data

The Event Service is responsible for all the data pertaining to the observation of events and the notification of interested NEXUS components,

5.9.5 Distribution

The Event Service will be distributed. Every “NEXUS node” will have at least certain components of the Event Service

5.9.6 Open Questions

How will the interface for plug-in triggers look like?

What kind of higher level events can be supported?

How many different notification mechanisms will be supported? What delivery semantics do they offer?

What is the most reasonable way to distribute the observation of events?

5.10 Communication

The Communication component of the NEXUS system architecture is responsible for exchanging data between different host systems belonging to the NEXUS system platform. In this chapter, the basic structure of the communication component is described, its functionalities are outlined, and interactions with other components of the NEXUS system are described.

5.10.1 Overview

The basic requirements of the communication component are to interconnect the various systems belonging to the NEXUS system, to provide access to mobile terminals by several network technologies and to support mobile applications with additional communication functionalities (e.g. informations about the actual network facilities and Quality of Service support). The defaults for the communication component are to adopt the Internet Protocol (IP) as the basic network protocol, to rely on established fixed network infrastructures like the Internet and on mobile networks like GSM or GPRS, where possible, and to use established network technologies like Ethernet, Wireless LANs, etc. In the following part of this section, the NEXUS communication system is structured and the main components are named.

The hosts building the NEXUS system which are interconnected by the communication component can be classified into two groups based on their communication capabilities:

1. fixed hosts (i.e. servers), running one or more NEXUS services; fixed hosts are normally attached to the network infrastructure by wire based network interfaces
2. mobile hosts (i.e. mobile terminals), which are carried arround by their users and which are usually attached to a network infrastructure by one or more wireless interfaces

Communication between these hosts will be handled by various network technologies which sometimes will coexist at one place, e.g. there might be two or more wireless networks at one place.

Due to the structure of the Internet, on which NEXUS will be based on, the communication infrastructure can be divided into access networks and a core network. Since requirements of connections to the network infrastructure are very different for fixed and mobile hosts, access networks can be subclassified into wire based and wireless access networks. Resulting from this classification and structuring, the communication infrastructure of NEXUS can be described as a set of interconnected access networks as depicted in figure 8.

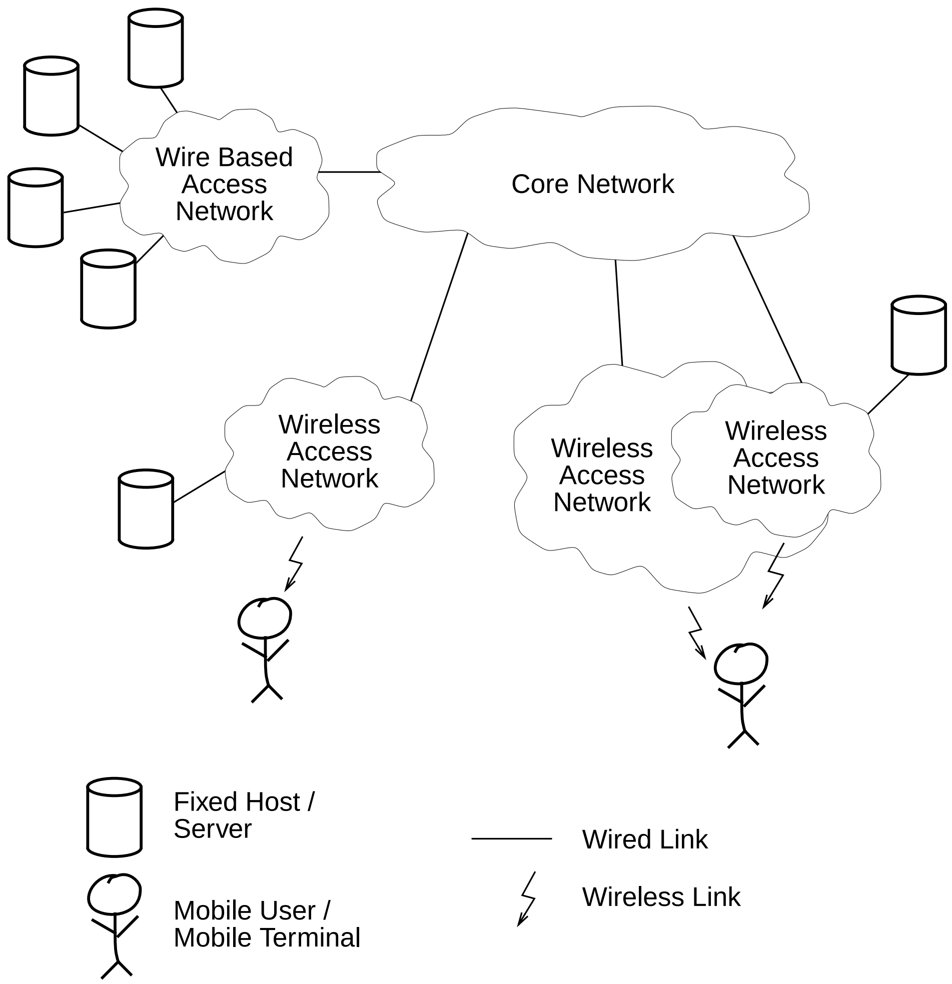


Figure 8: Simple network structure of the NEXUS system platform embedded into the Internet

The access networks attach host systems to the network infrastructure while the core network is responsible for interconnecting the various access networks. The core network will be the Internet with its typical structure of interconnected networks as depicted in figure 9. In this picture the Internet core network comprises the *Backbone*, *Network access points (NAPS)*, *Regional networks*, and *Local ISPs*. *User organization* networks correspond to the access networks defined above.

The core network as a worldwide deployed network infrastructure will not be a subject of research within the communication component. There might be enhancements to Internet technology, e.g. concerning the mobility support for mobile hosts, which mostly will be deployed at the edges of the core network or within the access networks. Those enhancements will be described later in chapter 5.10.2.

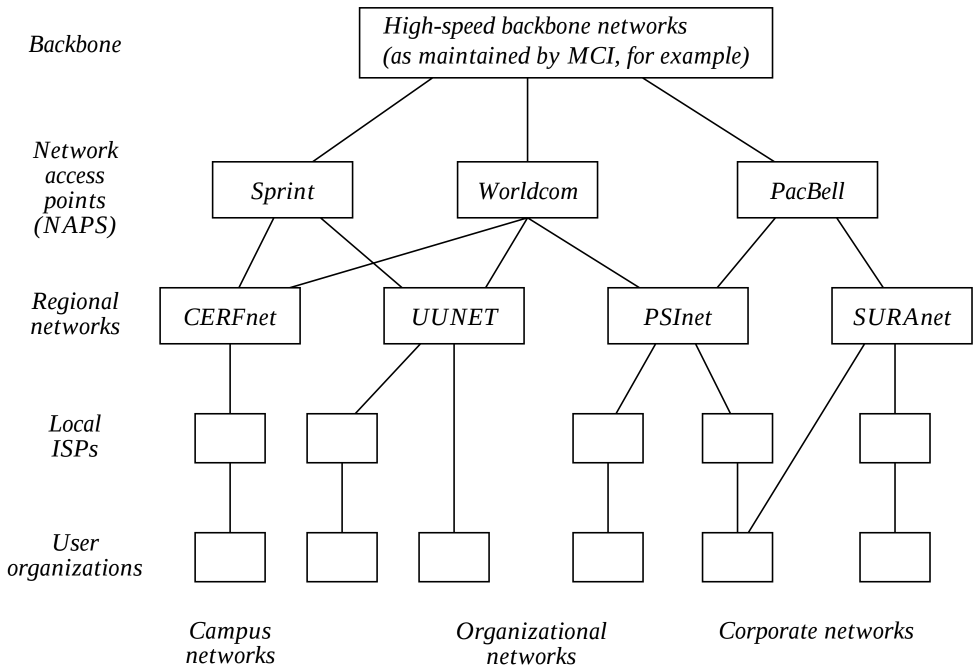


Figure 9: The Internet hierarchy (picture taken from [Kosiur 1998])

Based on communication technologies and the requirements to network links (as bandwidth, mobility, reachability, availablility, path of data packets), the communication relations between the various peers can be classified into the following four categories:

1. communication between mobile hosts and fixed hosts
2. communication between fixed hosts
3. communication between mobile hosts using a fixed network infrastructure
4. communication between mobile hosts without using a fixed network infrastructure (ad-hoc networking)

Based on this topology, the communication component will provide mobile communication support to the hosts of NEXUS. Based on current mobility supporting protocols, this support will be handled in the network layer and therefore will be transparent for applications. Additionally, there will be further mobility support (e.g. signalling of actual network conditions to applications or influencing choice of used network infrastructure), which will be offered to applications and to other components dealing with communication as Caching/Infofueling, Hoarding or GeoCast.

5.10.2 Functionality

The Communication component will provide several features and functionalities to provide communication and mobility support for the NEXUS system. The main features and functionalities are outlined in the following sections grouped into some areas. Parts of those functionalities are still research topics and may therefore change slightly.

Network technologies

The Communication component intends to offer communication services to mobile terminals everywhere, indoor as well as outdoor. So at least one indoor and one outdoor communication technology between which the mobile terminals can shift will be supported. At this time of the project, GPRS and a wireless LAN interface based on IEEE 802.11b specification (e.g. Lucent WaveLAN/ORiNOCO) are our favourite technologies. Other upcoming network technologies like UMTS, Bluetooth and HIPERLAN/2 will be observed.

The basic network protocol (IPv6)

For not to invent the wheel again, the communication component of NEXUS will mostly be based on the well established and yet widely deployed Internet Protocol (IP) enhanced by several mobility add-ons. To offer some new aspects in communications and due to the decision of the „Systems Architecture Group S2“ of the 3rd Generation Partnership Project (3GPP) in May 2000 to use IP Version 6 (IPv6) for 3rd generation mobile systems, IPv6 shall be used for the NEXUS system where possible.

Mobility Support

Mobility support of the NEXUS system comprises enhancements to the network layer protocol IP to support mobile hosts. Those mobility enhancements may already be implemented as with GPRS or may have to be implemented separately to support mobility e.g. to wireless LANs [Ramjee et al. 1999], [Valkó et al. 2000]. Due to latest advances in mobility support of IP, more than one mobility supporting protocol will be used in a hierarchical way with Mobile IP [Perkins 1996]. The mobility support at network layer will be transparent to the applications both on mobile and on fixed hosts. On mobile hosts, there will be some additional functions within the Communication API supporting additional mobility features which are discussed in the next section.

Communication API of mobile hosts

On mobile hosts, there will be more than one network interface, which can be used for transmitting data to the fixed network infrastructure or to other mobile hosts. At first, the Communication API has to hide the complexity of those network interfaces and the different functionalities they offer from applications. The Communication API should offer a unique view of the network to applications as Berkley Sockets or WinSockets are doing it now.

Due to the fact, that there are communication related impacts to the applications which the applications want to deal with, the Communication API has to provide additional functions to the applications. For instance, the applications (or the user) may want to choose, which network interface should be used for data transmission based on circumstances only known by the application (e.g. price conditions, etc.) or the applications have to react on changed network conditions by changing picture sizes of video streams.

The basic structure of the Communication API and its interaction with other components is depicted in figure 10. The detailed specification of the Communication API is not yet done.

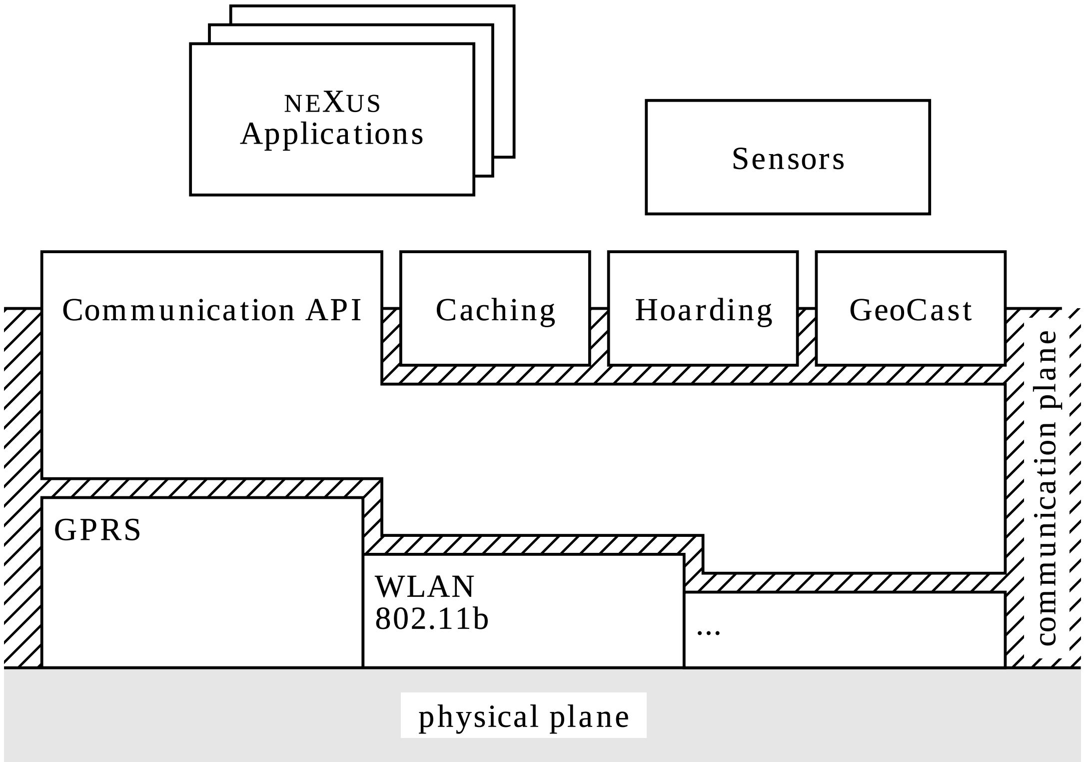


Figure 10: Communication components of a NEXUS-Station

5.10.3 Interaction With Other Components

The communication components will affect all other components, at least those, which want to communicate with other components over a network infrastructure. To provide communication service, a programming interface offering communication facilities will be provided on fixed and mobile hosts. The basic version of this interface will be the well known „socket interface" either of IPv4 or IPv6.

Host systems (fixed hosts as well as mobile hosts) providing communication facilities have to adopt device drivers for the used network devices and protocol stacks to provide the network and transport layer services of IP, either for IPv4 or IPv6. These protocol stacks might be enhanced by mobility supporting facilites like „Mobile IP Binding Update" support.

The placement of fixed hosts in the network can be optimized refering to the spatial relation in relation to the network topology of the services running on those hosts. For example: It might be useful to attach an exhibition information system (running on a host system) directly to the feeder net of the WLAN providing the exhibition site with communication service.

On mobile hosts there will be additional mobility supporting functionalities considering aspects of the fast changing network conditions a mobile host will suffer. Those functionalities provide information to the applications running on a mobile host and to other components of the NEXUS system as Hoarding, Caching and GeoCast-support.

5.10.4 Responsibility for Data

The Communication system of the NEXUS system platform operates on some data that reflect the network topology and which is related to the delivery of data packets (e.g. network addresses of hosts, routing tables). Those informations are data interact not so much with NEXUS objects and can easily be mapped on names (e.g. the Domain Name Service) or IDs.

Based on the network topology, the communication component can deal with spatial information about network coverage over ground. This information, e.g. to support Hoarding and Caching mechanisms, might be provided by the network operator while deploying the network, or might be gathered by special tools and applications running in the NEXUS environment.

5.10.5 Distribution Aspects

Communication Service will be distributed to a high degree in order to offer ability of exchanging data everywhere.

5.10.6 Open Issues

Mobility supporting protocols used to support mobility for users are not yet specified. Mobile IP and enhancements for Mobile IP will be a good choice. Extra support to use more than one network type at once have to be added to the mobility supporting protocols and are still research topics.

The communication interfaces on the mobile terminal to the NEXUS applications and to the additional network related services as Hoarding, Caching and GeoCast have to be defined.

5.11 GeoCast

5.11.1 Overview

GeoCast makes use of the location information for the means of communication. It seems to be useful to be able to send messages to a group of receivers in a specific region. The possible usages include the warning of catastrophes (like flooding or earthquakes), sending e-mail to all in a specific region, local service announcements, provision of slides, ... Especially for warnings, it is essential that users moving into the area do get the message. For this purpose, a message gets a lifetime. During this lifetime the message will be sent periodically.

5.11.2 Function

The user/client application has to provide the target area (which can be defined using geographic coordinates or NEXUS objects), and the message. Additional parameters would be the lifetime of the message, the groups that are to receive the message and the priority that the message has. If the information is provided, the message will be forwarded to all clients in the target region.

5.11.3 Interaction with other components

SpaSe, Location Service: In order to get the geographic coordinates for objects used to denote a target area, the GeoCast components access the Spatial Model Service and the Location Service.

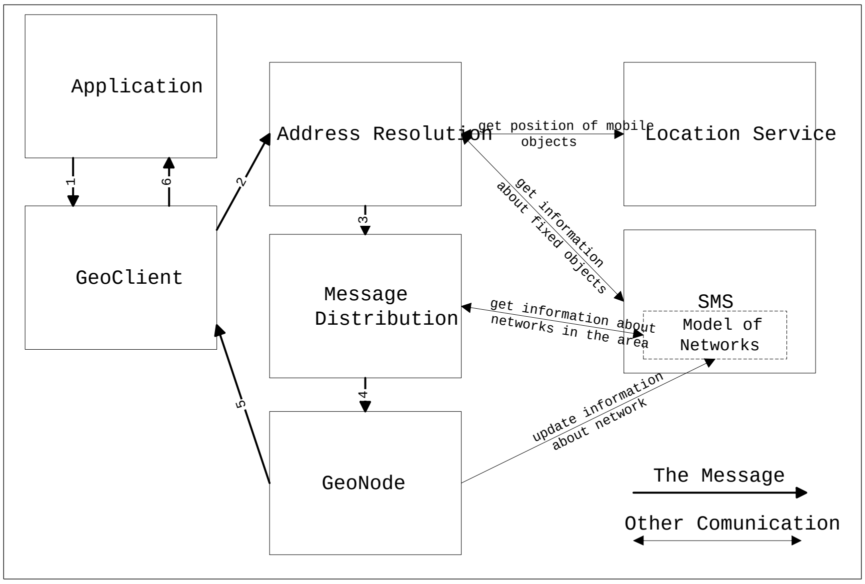


Figure 11: Basic Architecture of GeoCast and Interaction with NEXUS

Others: The implementation will make use of an information source that provides the information about the network topology. If this information (or parts of it) is needed by other components as well, the information might be located at a common component (SpaSe?)

5.11.4 Responsibility for data

GeoCast is responsible for the data that it needs to resolve the addresses and to forward the messages. It is not clear yet, how this data looks like. For one implementation the information is needed which networks cover which area and what GeoNodes are responsible for them. If this implementation is chosen, GeoCast will be responsible for this data.

5.11.5 Distribution

In order to provide GeoCast, several components will be needed:

Client: The client has to have a component called GeoClient in order to receive the messages. This component will naturally be highly distributed.

Server: In order to provide an efficient service, the server component will be distributed as well. The idea is to have a GeoCast component on every NEXUS-Server.

GeoNode: In each IP-Subnet that is to participate a GeoNode is needed. This component therefore will be distributed widely by definition.

5.11.6 Open Questions

- It is not clear yet, what the exact semantics will be, that GeoCast will support.
- How the user can specify a NEXUS object as target region.
- How the network information is gathered that is needed is another question that still has to be answered.
- How groups will be supported.
- SECURITY! Who is allowed to send messages of what length in what area

5.12 PKI Service

The PKI-Service is a kind of placebo-component standing for the security as a whole. It is still to discuss if NEXUS has to implement an own PKI-Service or if we just use any PKI-Service we trust, e.g. the one from RUS. A PKI-Service alone surely is not capable of satisfying all our security-needs. We will need a lot more security mechanisms. Since the security-group has not started yet, the required mechanisms and components are not fixed. The PKI-Service-component in our architecture just means, that we are aware of security and privacy problems and that we know this being a topic of great interest.

5.13 Info Fueling

5.13.1 Overview

Today, communications can take place in nearly any place with rather low bandwidth by using GSM or even by using GPRS. By only using these cellular communication networks, it is very expensive, to exchange great amounts of data. It is obvious, that this cannot be the situation, we wish to persist. So we introduce some “islands“ of high bandwidth, called hot spots. It is very likely, that 4th generation mobile systems also will strike this way of integrating several different network technologies.

This can be achieved e.g. by installing a wireless LAN with a maximum bandwidth of about 10 Mbit/s covering these islands. In the WLAN there is an access point connecting it to the global Internet. Introducing this topology, communications will become location dependant, because at different places, there are different network capabilities.

By providing a so called vertical handover between these different technologies and networks, one can communicate over the high-bandwidth networks when being in one of these islands and when being outside, one still can exchange data over the GPRS network. By introducing this scenario, it is possible to reduce costs and to increase the transmission’s quality of service simultaneously.

How does a typical communication using this vertical handover take place? At first, the user contacts the Info-Fueling system over the low-bandwidth GPRS network, indicating the files, he wants to load down. Then Info-Fueling decides, how to serve this wish. If the user is somewhere on the Swabian Alb for example where there are not many hot spots, transmission of the files just using high-bandwidth islands will last too long. So the files are transmitted only by GPRS or by the few passed hot spots in addition to GPRS. If communications take place somewhere on a highway or in a city, Info-Fueling perhaps decides to use only hot spots to push files on the mobile host. To make this scenario work, Info-Fueling needs to have some knowledge about where hot spots are located and perhaps about the user’s route, too. So Info-Fueling has two parts, one on the mobile host and another one, stationary somewhere attached to the Internet.

One occurring problem is that the link from the source-databases to the hot spots serving the mobile user, must have at least the same bandwidth as the wireless link from the hot spot up to the mobile host. Another problem is, that stationary parts have to be aware of Info-Fueling. To alleviate these drawbacks, a kind of proxy is introduced in the architecture described above. So the stationary part communicates with this proxy and is not aware of Info-Fueling. Then just the links from proxy to the hot spots have to provide high bandwidth. The proxy can collect data over the slow Internet and then push it quickly towards the hot spots when a user is passing them.

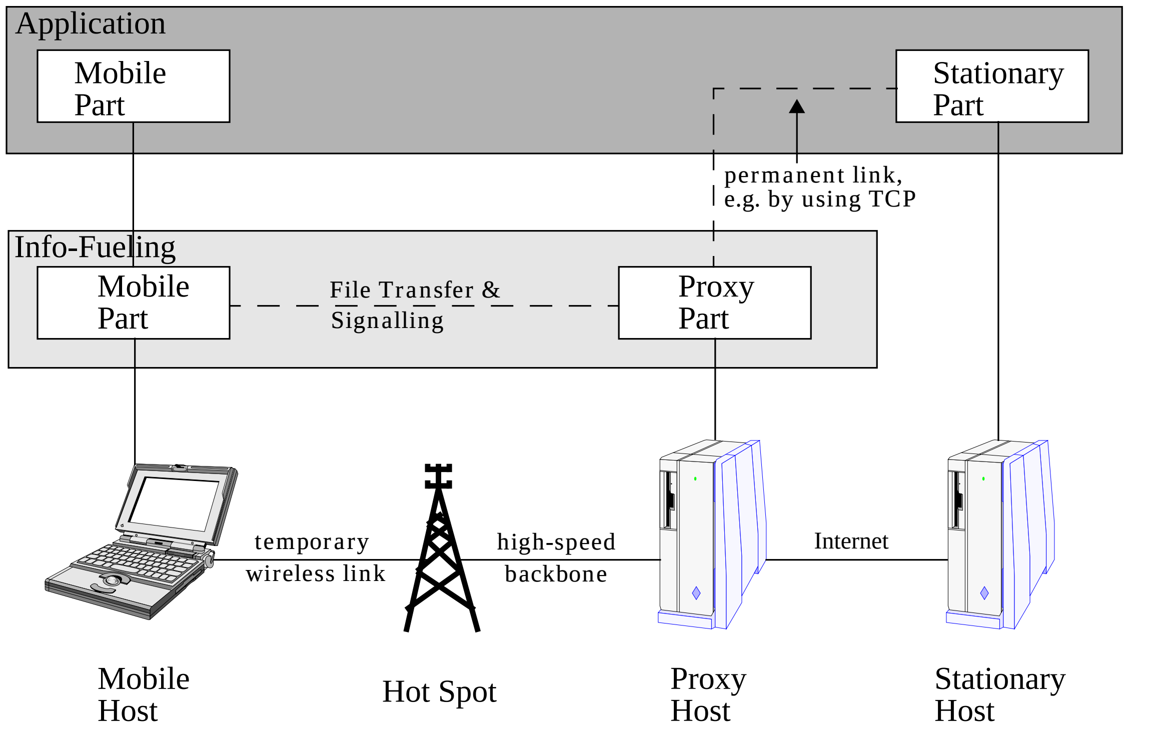


Figure 12: Architecture of Info-Fueling

The basic architecture of Info-Fueling is shown in Figure 12: “Architecture of Info-Fueling”. Each application has its own mobile part, running on the mobile host, its own stationary part, running somewhere in the world and its own proxy. Of course it is possible, that different proxies are running on the same host.

5.13.2 Functionality

One scenario, where the user wants to load some data down from somewhere in the Internet, will be depicted in this paragraph. The user is signalling his planned transmission over a low-bandwidth link, which is available in (nearly) all places. Then the Info-Fueling system is preparing data and is moving it towards the user, when he enters a hot spot.

Of course, a user can also load some data up to the Internet. Likewise, a host of the Internet is able to load files up to or down from the mobile host. Additionally these scenarios can be divided in those with location dependent data and those with data independent of location.

On the one hand side, the Info-Fueling part running on the mobile host can be accessed by applications and on the other hand side the parts running on the proxy hosts in the network infrastructure can be accessed. Either they can be accessed directly by an own Info-Fueling API or they can be hidden under the API of the Communication Component being transparent to the applications.

5.13.3 Interaction with other components

It is still to examine, with which components Info-Fueling has to interact. For example, it is likely to interact with the Location Service for locating some users or with the SpaSe to get some information in order to improve delivery to the user.

In any case, it has to interact with applications, e.g. for accessing a user's route, as well as with the Communication Component. Maybe there will be offered a special interface to the applications for using the enhanced service or the functionality will be hidden for applications, making Info-Fueling running somehow in the background.

5.13.4 Responsibility for data

Info-Fueling is responsible for the data, it needs to improve transmissions. There is for example the location of the hot spots, their coverage area and perhaps the load in the Wireless LANs as well. Perhaps some of this data will also appear in the Augmented World Model and can be accessed there. Until now, it is not decided, how this data will look like.

5.13.5 Distribution

As depicted above, the Info-Fueling system consists of a mobile part running on the mobile host and a stationary part running on the proxy host in the network infrastructure.

5.13.6 Open Questions

There are a lot of open issues to investigate. For example it is not sure, if an application using Info-Fueling can reserve some bandwidth in the hot spots and how big amounts of data, e.g. one large file, will be distributed over more than one hot spot, the mobile user will pass.

Other open questions are with which location aware services the Info-Fueling system will interact, e.g. GeoCast or city guides, and in which way Info-Fueling will use the Location Service and the Spatial Model Service of NEXUS.

5.14 Hoarding Component

5.14.1 Overview

The hoarding component is needed to hide the disadvantages of mobile information access over wireless long distance networks from the user to the greatest possible extent. These disadvantages in particular include low bandwidth, delays and frequent connection losses. The basic idea is to transmit the information to the mobile device before the user actually accesses it. The so called "information hoarding" takes place at info stations. These are basically proxy servers that can be accessed via a wireless LAN.

The hoarding component is a service that can be used by the NEXUS platform but is not critical for its functionality.

5.14.2 Functionality

On a user's arrival at an info station the station determines the information the user will most likely access in the future. This decision is based on different criteria. The info station is also responsible for the information to be transmitted to the mobile device. The following criteria will be taken into account:

- the user's future location
- the frequency a certain information is accessed with
- the user's profile (speed of the user's movement, age, device type,...)
- the link structure of the information space

5.14.3 Interaction with other components

The hoarding component interacts with at least the following components:

- sensor component: Supplies the client-side hoarding component with the user's current location
- area-service-register: Manages information on which particular info station is responsible for a particular geographical area.

The requirements the hoarding component needs to fulfil in order to be used with the NEXUS system have been discussed at length in the hoarding presentation in the workgroup (see slides).

5.14.4 Responsibility for data

The hoarding component only handles internal data no other components are relevant for.

5.14.5 Distribution

Each info station is responsible for a geographically limited area. An info station only transmits information in advance that is being needed in this particular area.

5.14.6 Open questions

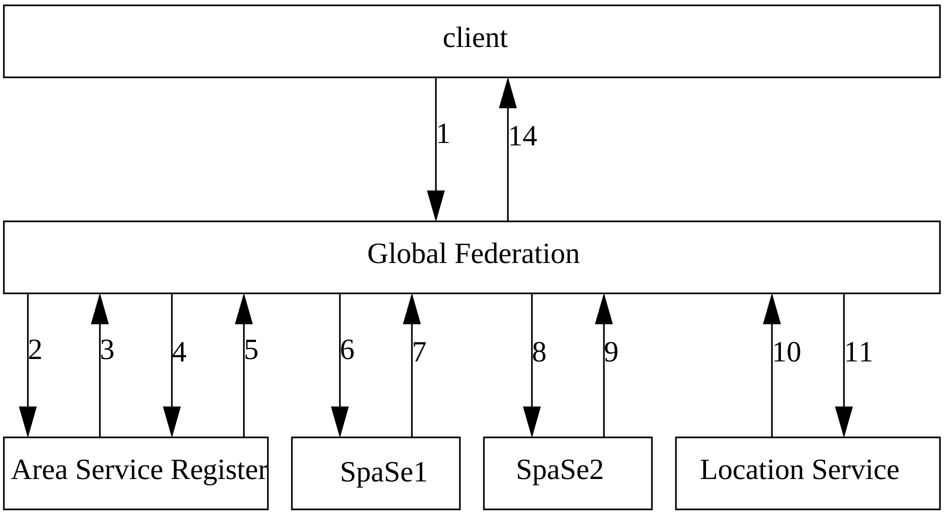
The interfaces to other NEXUS components are not yet exactly specified.

6 Scenarios

In this section we will consider two typical scenarios of applications of the neXus system. The first one describes the interaction of the different servers more detailed, whereas the second focuses on an architectual view on the system.

6.1 Request for a map of Stuttgart

A travelling group visits Stuttgart. A member of the travelling group requests a map of Stuttgart containing streets and buildings etc., Virtual Information Towers (VITs) and all members of the group.



1. The client locates a neXus-server running a Global Federation via its wireless communication interface. It sends a request for a map of the area of an object named "Stuttgart" containing static real-world objects, VITs and all objects that belong to the travelling group to this server.
2. Global Federation requests the adresses of the SpaSes responsible for static real-world objects in Stuttgart from the Area Service Register
3. Area Service Register replies with the address of SpaSe1
4. Global Federation requests the adresses of the SpaSes responsible for VITs in Stuttgart from the Area Service Register
5. Area Service Register replies with the address of SpaSe2
6. Global Federation requests all static real-world objects from SpaSe1
7. SpaSe1 replies with list of objects
8. Global Federation requests all VITs from SpaSe2
9. SpaSe2 replies with list of objects
10. Global Federation requests the positions of the members of the travelling group from the Location Service
11. Location Service replies with positions

12. Global Federation integrates the data of static real-world objects, VITs and travelling group members into one object
13. Global Federation generates a map from this object
14. Global Federation sends the map back to the client

6.2 Passive information about environment of a user

A mobile user moves through a certain environment, e.g. a city center, an exhibition or a shopping centre. His neXus client displays a varying list of VITs according to his position and his profile.

1. There are two alternatives for the application displaying VITs to get the necessary data:
   - It uses the Event-Service and subscribes an event that ist generated every time the client enters the visibility area of a VIT or
   - It just checks periodically if it is in the visibility area of a VIT. The application may use caching in this case to reduce the amount of data transferred.
2. The VITs are filtered according to the profile of the user.
3. The application requests content, graphical representation etc. for the requested VITs from the Spatial Model Service.
4. The application updates its list of VITs.

7 Security

In NEXUS we have to handle a lot of data concerning the privacy of users. In fact, we collect a lot of data, especially very exact location data, of users. So we need to proceed very carefully concerning security and privacy.

At first we have to assure integrity and availability of systems as well as of components. Furthermore confidentiality issues have to be regarded as well as accountability. We have to provide for actions being legally binding to assure non-repudiation. There are several classic security objectives like confidentiality, integrity and access control, but we also have to observe some new issues like the fine granularity of location information. So our main focus will be on the conjunction of location, identity and service usage.

A great threat, which is known for long, is user tracking and collection of data for building user profiles. On the one hand side, we have to encrypt data, so that just authorized parties can read the information. On the other hand side we have to assure, that nobody can reveal secure information by observing exchanged data. There are two common approaches to tackle this second issue: Trust in providers or complete distrust. In the latter case we have to mask location information totally e.g. by using MIX-concepts.

What is new in NEXUS, is the wish to reveal location information to some services. The danger coming along is unwished conjunction of location- and identity-information. So we have to economize data on the least amount needed. A NEXUS-user should always have complete control over the data he is offering. Possible approaches hereby are pseudonym-mechanisms or dynamic definition of trusted areas.

At first the security group has to examine our use cases in order to find the threats. So we have to determine the persons taking part in NEXUS, like users, providers or general public, and to look on the interests they have. After that, we will have to build a model about what informations are generated, what informations are needed and what informations have to be guarded against whom. On the basis of this model and the determined use cases, we discover the essential data streams. By allocating security interests to the data, we will find security objectives and threats in NEXUS, built up by exchange and storage of information in different areas. Examples of threats are recording, manipulation or correlation of data.

One possible approach is the division in several physical or logical domains being separated from each other to assure their security objectives. Possible examples for domains are the NEXUS-Station, data structures of the augmented world model or log-data. Then these areas can be delimited by physical, logical or cryptographic schemes like filters, anonymizers, trusted third parties or public key crypto systems.

Concluding, we can say, that we have to consider the security-needs of both, the providers as well as the users. So we have to provide multilayered security in NEXUS and therefore have to observe lots of security objectives.

8 Demonstrator

At the review of the research group NEXUS by the DFG we want to show a demonstrator of the platform. The requirements and prerequisites for this demonstrator and the way it is to be created will be described in this chapter. It is planned to take an incremental approach to the development of the demonstrator, where a working version of the platform is available at all times for demonstrations and tests. This version will be enhanced with further functionality continuously.

8.1 Goals for Developing the demonstrator

For the review of the research group NEXUS through the DFG in the middle of 2002, a simple demonstrator shall be available with which selected functionality of the NEXUS platform can be shown. A main purpose of the demonstrator is the presentation of first results of the research done in the research group.

As an application for the demonstrator of the NEXUS platform it is planned to create a location-aware information system (e.g. for a city or a fair), which contains at least the following elements:

- searching for stationary and mobile objects
- displaying virtual and mobile objects
- a map for the current environment of the user
- sending geographical messages
- simple spatial events

8.2 ViLiS-System

The ViLiS system was developed at the Distributed Systems department of the IPVR as the result of a study project. It is a simple location-aware system and can be seen as a first prototype of the NEXUS platform. It will be used as the starting point for an incremental development of the platform.

The ViLiS system is based on the idea of Virtual Information Towers (VITs, see [Leonardi et al. 1999b]), which are attached to a certain geographical position in the real world and have a given area of visibility. Items of information (posters), which are relevant for this location and are of interest inside its area of visibility, can be attached to such a Virtual Information Tower. Using a client program a mobile user always has a direct access to the nearest Virtual Information Towers and therefore to the information that is of interest for his current geographical location.

The ViLiS system, which manages the Virtual Information Towers and provides access to the information stored on the VITs, consists of the components shown in Figure 13: "Architecture of the ViLiS System", which are described in the following sections.

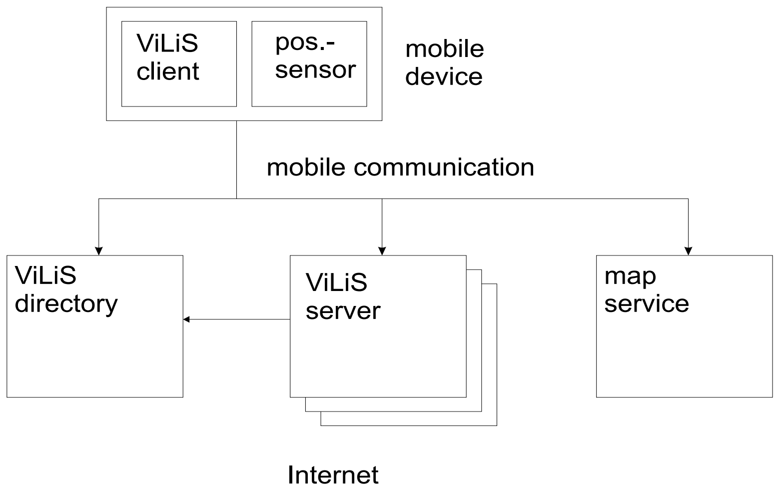


Figure 13: Architecture of the ViLiS System

8.2.1 ViLiS Client

A ViLiS client runs on a mobile device and allows its user to access the ViLiS system. Through a local positioning system (e.g., GPS) the client determines the current location of its user. It shows a list with the VITs visible from this location, which it loads periodically from the ViLiS directory using mobile communication. Additionally, certain posters of a certain VIT (equivalent to Web pages) can be loaded from the corresponding VIT server and a map of the current environment of the user, which is provided by the map service, can be displayed.

8.2.2 ViLiS Directory

The ViLiS directory contains links to all available VITs and is used to find the VITs that are visible from a certain geographical location. The answer to such a query, contains besides the name of the VIT a link to the corresponding ViLiS server, where the client can query the contents of the VIT.

8.2.3 ViLiS Server

A ViLiS server is based on a Web server and manages the table of contents and if necessary the posters for one or more VITs. At a ViLiS server the table of contents and posters of a VIT can be queried. If available, it also provides access to the black board of a VIT. The VIT server registers the VITs that are managed by it at the VIT directory.

8.2.4 Map Service

The map service returns an image of a map for a given geographical position, extent and level of detail.

8.3 Development Phases

Till the review of the research group in the middle of 2002, the demonstrator of the NEXUS platform will be created from the ViLiS system in the following phases:

8.3.1 Phase 0 (Cebit), End of February 2001

In the first phase, the ViLiS system will be enhanced by the following components. The first items are mandatory, while the last items are nice to have:

1. a simple indoor positioning component, probably based on the Cybercodes of a Sony Vaio
2. integration of a simple Spatial-Model-Server, which replaces the map service of the current ViLiS system (prerequisites are maps of the Cebit halls and if possible of Hannover)
3. support for a number of fixed location-aware queries
4. integration of the ViLiS directory into the functionality of a Spatial-Model-Server

8.3.2 Phase 1, End of September 2001

The second phase, which shall be finished in the end of September 2001, will consist of the following components:

- a first version of the final application interface
- a centralized location service
- rudimentary functionality for geocast
- a Spatial Model Server with a first version of the final application interface
- a version of the ViLiS Client, which has been converted to run with these additional components

8.3.3 Phase 2, End of February 2002

The last phase of the demonstrator in the first period of support by the DFG, shall consist of the following components:

- at least one Spatial-Model-Server: desirable are a number of servers
- at least a central location service; desirable is a distribution on a number of servers
- an area service register with basic functionality
- basic functionality for the event service
- a global federation that integrates queries for the Spatial Model Servers and the Location Service or passes them through
- functionality for GeoCast
- two different types of sensors for out-doors and in-doors (GPS and a client based sensor system that works indoors)
- communication shall be possible over two different networks as well as switching between them without loosing connection (GPRS and Wireless LAN according to IEEE 802.11)
- a location-aware information system (see above) as an application of the NEXUS platform

9 Project Plan

One goal of the workshop was to agree on a project plan, which covers the main milestones both of the research process and the demonstrator.

Of course research cannot be planned like an engeneering project but we managed to define the next steps to be taken and dates when they should be accomplished.

The next important milestone is the 26.9.2000 where the working group meets again and discusses the first version of the interfaces between the components.

Several task forces (as described in Chapter 10 “Organisation of the Project (German)” on page 55) will provide input to this meeting to prepare certain decisions.

9.1 Timetable

Task	Begin	End
Interface Language (XML, SQL, ...)		1.9.2000
Database System		26.9.2000
Operating System		26.9.2000
Nexus-ID		26.9.2000
Version 1.0 of the interfaces		26.9.2000
Augmented World Model		30.11.2000
Demonstrator, Stufe 0	1.09.2000	1.3.2001
Demonstrator, Stufe 1	1.4.2000	1.10.2001
Demonstrator, Stufe 2	1.11.2001	1.3.2002
Nexus-Client	1.10.2001	1.7.2002

10 Organisation of the Project (German)

As the content of this chapter is only relevant to members of the research group, this chapter will be in German.

10.1 Treffen

10.1.1 Vollversammlungen

Alle sechs Monate finden Vollversammlungen statt, an denen die Mitglieder der Forschergruppe sowie die Leiter der beteiligten Abteilungen teilnehmen. Die Abteilungen wechseln sich in der Organisation der Vollversammlungen ab.

10.1.2 Koordinationstreffen

Koordinationstreffen finden ca. einmal pro Monat statt, passend zu den Meilensteinen. An den Koordinationstreffen nehmen alle Mitarbeiter der Forschergruppe teil. Sie präsentieren und diskutieren die aktuellen Arbeitsergebnisse. Die Koordinationstreffen übernehmen die Aufgaben eines Projektleiters: Sie dienen der Fortschrittskontrolle und der Planung. Die Abteilungen wechseln sich in der Organisation der Koordinationstreffen ab.

10.1.3 Informelle Treffen

Informelle Treffen zwischen zwei Abteilungen finden falls nötig statt. Die Organisation übernehmen die betroffenen Abteilungen.

10.1.4 Task-Forces

Zur Vorbereitung von Entscheidungen zu einzelnen Themen werden Task-Forces eingerichtet. Bisher existieren die Task-Forces

- Betriebssystem: Bernd Gloss (Leitung), Peter Coschurba, Darko Klinec
- Indoor-Positioning-System: Darko Klinec (Leitung), Peter Coschurba, Bernd Gloss
- Nexus-ID: Steffen Volz (Leitung), Jens Messmer, Daniela Nicklas, Matthias Großmann, Alexander Leonhardi
- Schnittstellensprache und Kommunikationsmechnismus: Daniela Nicklas (Leitung), Steffen Volz, Alexander Leonhardi, Christian Hauser

10.2 Dokumentation

Es wird ein gemeinsames Design-Dokument im WWW erstellt. Jedes Kapitel des Dokuments besteht aus (mindestens) einer eigenen Datei, die folgende Information enthält:

- Titel
- Kurzbeschreibung (HTML)
- Änderungshistorie (HTML)
- Modelle

Teil der Dokumentation ist ein gemeinsames Begriffslexikon.

10.3 Versionierung

Folgendes Versionierungsschema wird verwendet: <*Schnittstellenversion*>.<*Komponentenversion*>. Alle Komponenten mit der gleichen Schnittstellenversion sind miteinander kompatibel. Entscheidungen über die Schnittstellenversion werden im Rahmen von Koordinationstreffen gefällt, Entscheidungen über die Komponentenversion trifft der Verantwortliche.

10.4 Verantwortlichkeiten für Komponenten

Die folgende Tabelle enhält die Ansprechpartner für die Einzelkomponenten. Ihre Implementierung wird ggf. von anderen Personen(-gruppen) übernommen.

Thema	Verantwortlich
Location Service	Alexander Leonhardi (VS)
SpaSe	Steffen Volz (IFP)
GeoCast	Peter Coschurba (VS)
Event Service	Martin Bauer (VS ab 1.9.2000)
Kommunikation	Bernd Gloss (IND)
Sicherheit	Christian Hauser (IND)
Hoarding	Uwe Kubach (VS)
ASR	Steffen Volz (IFP)
AWM	Jens Messmer (DA bei Daniela Nicklas, AS)
Name Service	Peter Coschurba (VS)
Globale Föd.	Daniela Nicklas (AS)
Sensorkomp.	Darko Klinec (IFP)
ext. Information	Matthias Großmann (AS)

11 Open Issues

This chapter contains a list of open issues in the design discussion and possible answers. Since this list will be maintained in the further research process it is kept in German.

11.1 Wo liegen die Daten eines mobilen Objekts?

Status: offen

Betrifft: Architektur, Datenverwaltung

Quelle: Architekturwoche 10.-14.7.2000

Möglichkeiten:

- im LoSe oder
- im SpaSe mit Vermerk, daß die Position nicht gültig ist oder
- im SpaSe, der den zugehörigen LoSe kennt

11.2 Wo liegen virtuelle Objekte?

Status: offen

Betrifft: Architektur, Datenverwaltung

Quelle: Architekturwoche 10.-14.7.2000

Möglichkeiten:

- in einem SpaSe, in dem auch normale statische Objekte liegen oder
- in einem extra SpaSe, dessen Modell überlagert wird

11.3 Wie erfolgt die Zuordnung Oid <=> Server?

Status: offen

Betrifft: Architektur, Datenverwaltung

Quelle: Architekturwoche 10.-14.7.2000

Möglichkeiten:

- über ein Object Register oder
- über die Nexus-ID

11.4 Wie sieht die Nexus-ID aus?

Status: offen

Betrifft: Architektur, Datenverwaltung

Quelle: Architekturwoche 10.-14.7.2000

Entscheidung durch Taskforce Nexus-ID

Möglichkeiten:

- Server ist bei allen Objekten in die ID hineinkodiert oder
- Server ist nur bei statischen Objekten in die ID hineinkodiert
- "enthalten"-Beziehung ist hineinkodiert (hierarchisch)
- menschenlesbar
- ewig oder epochal
- eindeutig für jedes reale Objekt oder
- reale Objekte in verschiedenen Repräsentationen haben verschiedene Nexus-IDs und werden über Beziehungen miteinander verknüpft

11.5 Unterscheidet sich ein virtuelles Objekt (vO) prinzipiell von einem realen Objekt (rO)?

Status: offen

Quelle: Architekturwoche 10.-14.7.2000

Möglichkeiten:

- ja: virtuelle Objekte sind entweder ortsbezogen (VITs) oder objektbezogen (vPostit)
- nein: statische vOs (VITs und vPostIts an statischen Objekten) liegen in einem SpaSe, mobile vOs (vPostIts an mobilen Objekten) haben einen Positionssensor (z.B. den des zugehörigen rOs) und liegen in einem LoSe

11.6 Registrieren sich Anwendungen? Gibt es Profile für Anwendungen?

Status: offen

Quelle: Architekturwoche 10.-14.7.2000

Betrifft: Architektur, Datenverwaltung

Damit ist die Frage verbunden, ob die Globale Föderation ein Gedächtnis hat.

11.7 Wo wird die "enthaltenen-sein"- Beziehung von mobilen Objekten verwaltet?

Status: offen

Quelle: Architekturwoche 10.-14.7.2000

Betrifft: Architektur, Datenverwaltung

Beispiel: ich setze mich in den Zug.

11.8 Wer garantiert die Privacy bei den Lokationsdaten?

Status: offen

Quelle: Architekturwoche 10.-14.7.2000

Betrifft: Architektur, Datenverwaltung, Security

11.9 Wie wird der Funktionsumfang eines bestimmten SpaSe spezifiziert (beim Area Service Register)?

Status: offen

Quelle: Architekturwoche 10.-14.7.2000

Betrifft: Architektur, Datenverwaltung

11.10 Hat der Demonstrator nur 2D oder auch 3D-Daten?

Status: offen

Quelle: Architekturwoche 10.-14.7.2000

Betrifft: Demonstrator

12 References

[Bauer2000]

Martin Bauer: Event-Management für mobile Benutzer. Universität Stuttgart, Fakultät Informatik, Diplomarbeit Nr. 1836 (2000). English.

[Hohl et al 1999]

F. Hohl, U. Kubach, A. Leonhardi, K. Rothermel, M. Schwehm: Next Century Challenges: Nexus - An Open Global Infrastructure for Spatial-Aware Applications, Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, Washington, USA, August 15-20, 1999, T. Imielinski, M. Steenstrup, (Eds.), ACM Press, 1999, pp. 249-255.

[Kosiur 1998]

D. Kosiur; „Building and Managing Virtual Private Networks"; Wiley Computer Publishing; 1998

[Leonhardi et al 1999a]

A. Leonhardi, U. Kubach: An Architecture for a Universal, Distributed Location Service, Proceedings of the European Wireless '99 Conference, Munich, Germany, pp. 351-355, ITG Fachbericht, VDE Verlag, 1999

[Leonardi et al. 1999b]

A. Leonhardi, U. Kubach, K. Rothermel, A. Fritz: Virtual Information Towers - A Metaphor for Intuitive, Location-Aware Information Access in a Mobile Environment, Proceedings of the Third International Symposium on Wearable Computers (ISWC'99), San Fransisco, CA, USA, IEEE Press, 1999

[Perkins 1996]

C. Perkins; „IP Mobility Support"; RFC 2002; October 1996, IETF

[Ramjee et al. 1999]

R. Ramjee et al.; "IP micro-mobility support using HAWAII"; Internet Draft, work in progress; Juni 1999; Lucent Bell Labs

[Schützner 1999]

Johannes Schützner: Entwicklung einer Server-Komponente für ein räumliches Modell in Nexus. Universität Stuttgart, Fakultät Informatik, Diplomarbeit Nr. 1768 (1999). Deutsch.

[Valkó et al. 2000]

András G. Valkó et al.; "Cellular IP"; Internet Draft, work in progress; Januar 2000; Columbia University, Ericsson