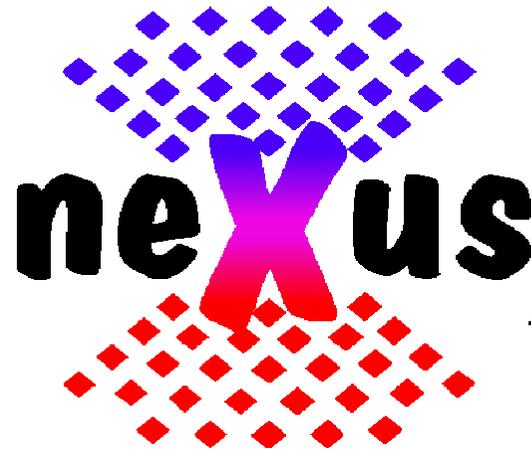


---

## 2.5 Das Nexus Projekt



# Nexus

---

Nexus ist ein Projekt der Universität Stuttgart an dem folgende Lehrstühle beteiligt sind:

- Institut für Photogrammetrie  
Professor Fritsch
- Institut für Nachrichtenvermittlung und Datenverarbeitung  
Professor Kühn
- Institut für Parallele und Verteilte Höchstleistungsrechner  
Professor Mitschang (Abteilung Anwendersoftware)  
Professor Rothermel (Abteilung Verteilte Systeme)

Zeitlicher Rahmen: 1.1.2000 bis 31.13.2002

## Nexus - Motivation

---

Durch die immer weiter wachsende Datenflut wird es immer schwieriger die Daten zu erhalten, die in diesem Moment relevant sind.

Lösung: **Location Based Services** (ortsbezogene Dienste)

Problem: bisherige Anwendungen sind Insellösungen, d.h. jede Anwendung besitzt einen eigenen Datenbestand und eigene Sensoren. Daher ist die Einführung neuer ortsbezogener Dienste mit einem sehr hohen Aufwand verbunden, wobei die Funktionalität durch den vergleichsweise kleinen Datenbestand eher beschränkt ist.

Lösung: Nexus

## Nexus – Ziele

---

Das Ziel von Nexus ist es eine Plattform für ortsbezogene Dienste zu entwickeln, die diesen ein detailliertes Modell der realen Welt zur Verfügung stellt, sowohl für die Anwendung im Freien als auch in Gebäuden. Dieses Modell wird *Augmented World Model* genannt und besteht sowohl aus Vertretern für Dinge aus unserer ‚realen‘ Welt als auch aus virtuellen Objekten, die eine Verknüpfung zu externen Informationsquellen, wie zum Beispiel dem WWW, darstellen. Dieses Modell wird durch integrierte Sensoren ständig aktualisiert, um z. B. den Aufenthaltsort von mobilen Objekten ständig zu korrigieren.

# Nexus

---

Um die Funktionalität und die Akzeptanz des Nexus-Systems sicherzustellen, muss die Plattform einige grundlegende Eigenschaften besitzen:

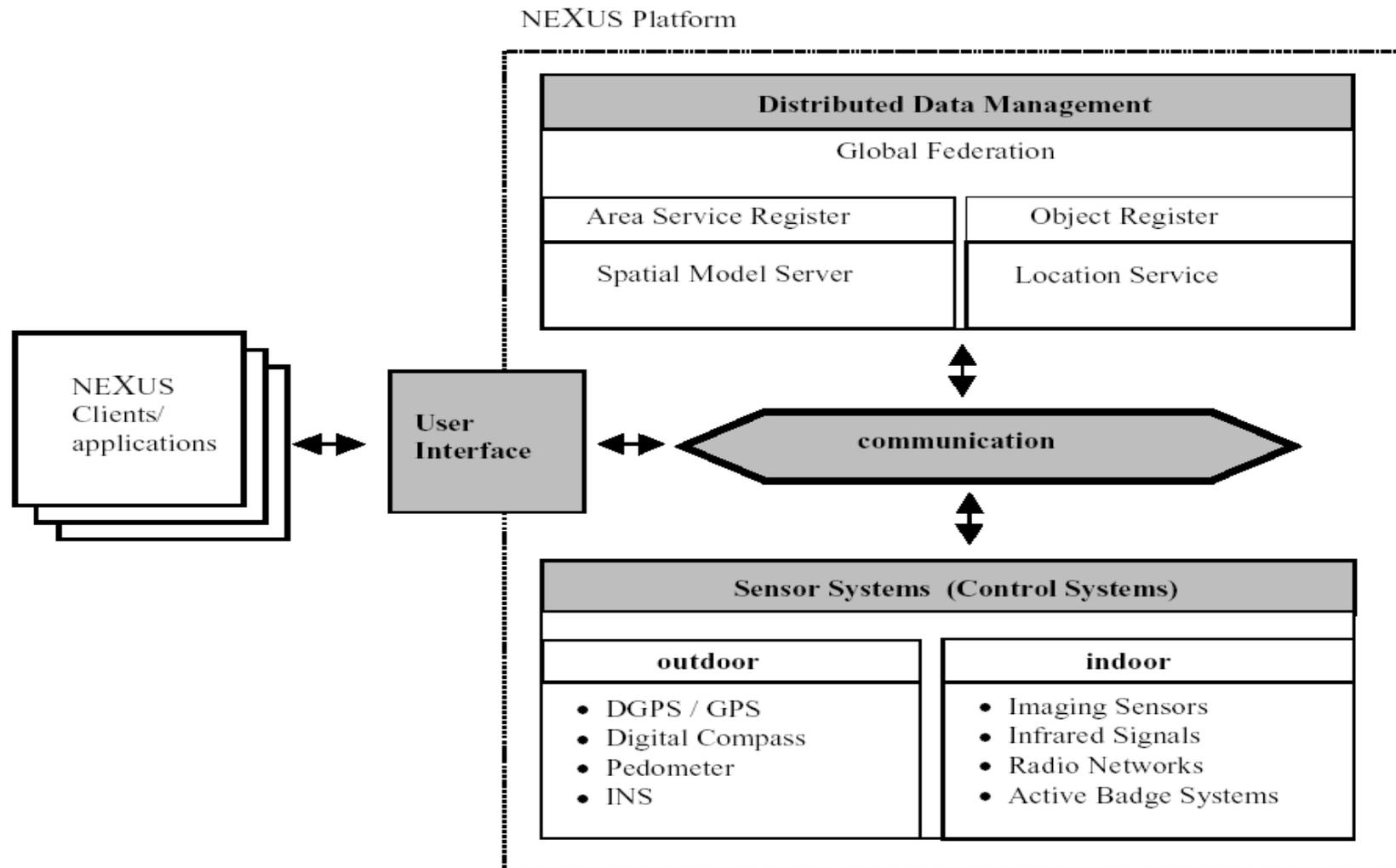
- Skalierbarkeit
- Datensicherheit
- Adaptivität
- Interoperabilität
- Offene Infrastruktur
- Plattformunabhängigkeit

## Nexus – Anwendungsgebiete

---

- Informationssysteme für Messen, Museen oder Städte
- Einkaufsführer für Kaufhäuser oder Ladenpassagen
- Navigationssysteme
- Stau- oder Unfallwarnsysteme
- Virtuelle Schwarze Bretter
- Virtuelle Ticket Schalter

# Nexus - Architektur



# Nexus

---

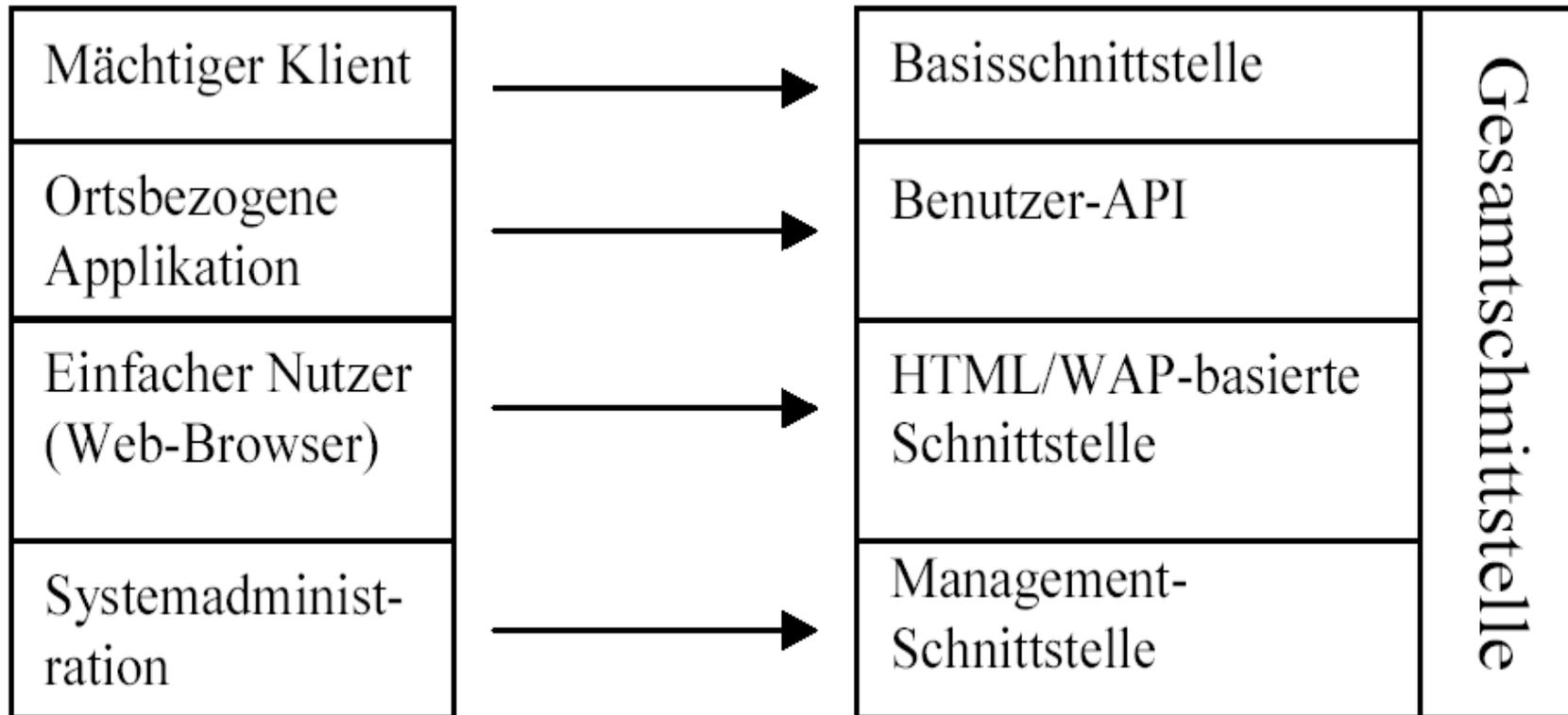
## 2.5.1 User Interface

- Verschiedene Arten von Endgeräten (PDAs, Notebooks, Wearables)
- Unterschiedliche Rechenleistung, Speicherkapazitäten, Displays



## Nexus – User Interface

---



Verschiedene Endgeräte benötigen individuell angepasste Schnittstellen

## 2.5.2 Communication

Probleme:

- Es gibt eine Vielzahl von Drahtlosen Datenübertragungstechniken (GSM, GPRS, UMTS, IEEE 802.11, Hiperlan, Bluetooth)

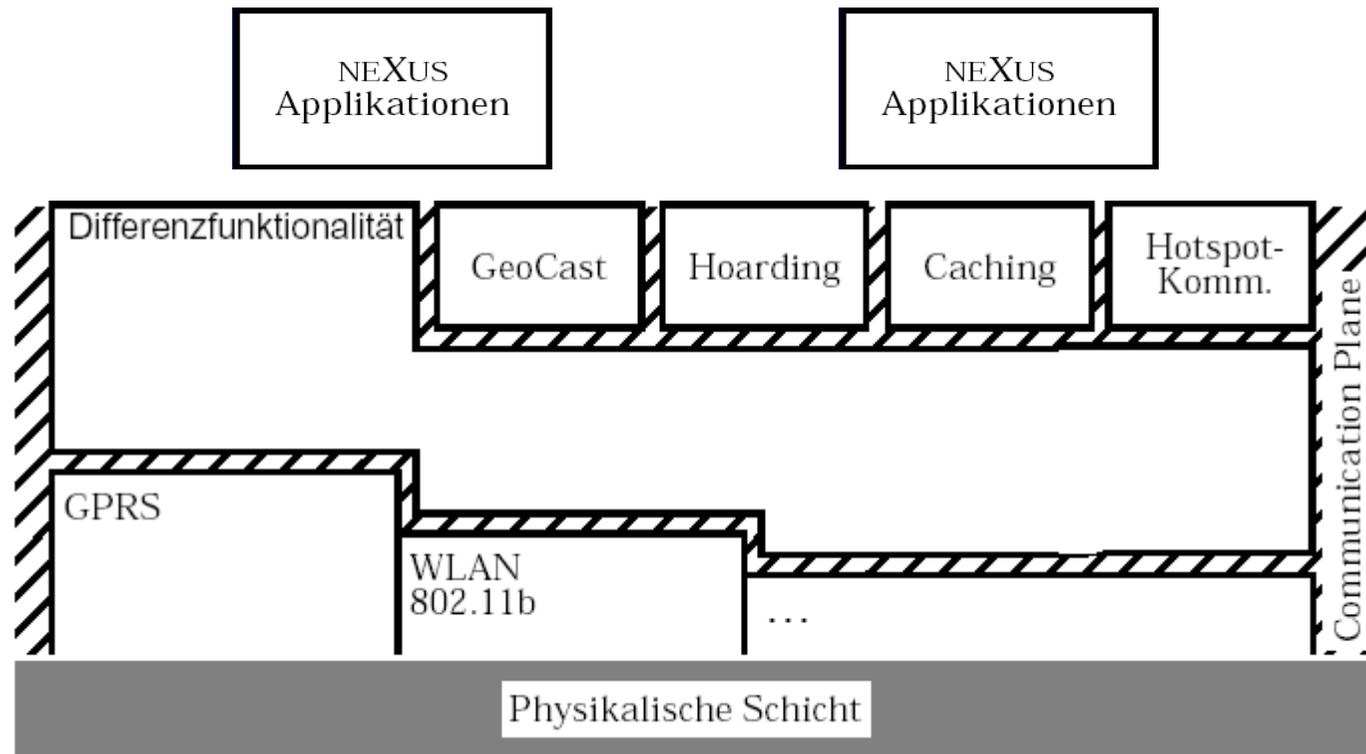
⇒ keine einheitliche Technik, mit der alle mobilen Endgeräte kommunizieren können.

- Schmalbandige Übertragungstechniken sind meistens nicht ausreichend und zu dem teuer
- Breitbandige Übertragungstechniken sind nicht flächendeckend verfügbar

Lösung:

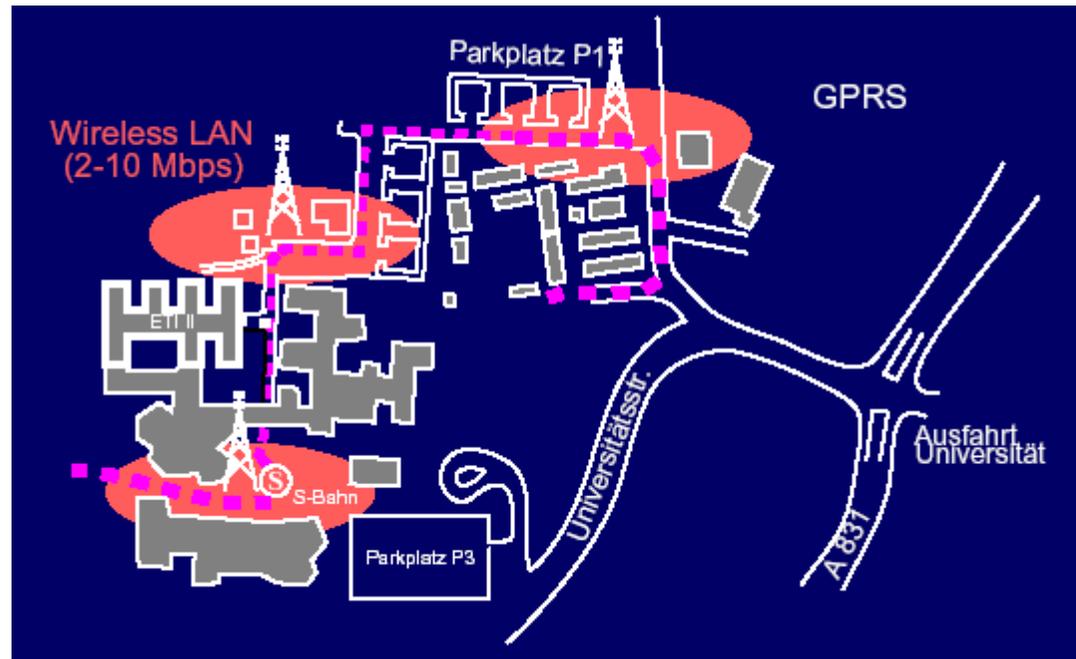
Nexus unterstützt alle gängigen Übertragungstechniken und benutzt eine Mischung aus schmal- und breitbandigen Übertragungstechniken

## Nexus - Kommunikation



- Anhebung unterschiedlicher Technologien auf ein gemeinsames Funktionsniveau
- Integration von Kommunikationsdiensten (Geocast, Hoarding, Caching, Hotspot-Kommunikation)
- Basiskommunikationsprotokoll IP, um zusätzliche Funktionalitäten ergänzt (Mobilität, Sicherheit)

# Nexus - Kommunikation



- Flächendeckende Grundversorgung über GSM / GPRS (UMTS)
- Inseln mit hoher Bandbreite, z.B. Wireless LAN (Hotspots)
- Signalisierung des Übertragungswunsches via GSM / GPRS
- Intelligente Verteilung der Kommunikation auf Hotspots entlang der Route

## 2.5.3 Sensory Systems (Control Systems)

Es gibt viele verschiedene Möglichkeiten die Position eines mobilen Objekts zu bestimmen:

- GSM
- GPS / DGPS
- Digitaler Kompass und Schrittzähler
- Bluetooth
- IR
- Active Badge System
- Kamerabilder

## Nexus - Sensory Systems (Control Systems)

---

Im Rahmen des Nexus Projekts wird auf eine Multi-Sensoren-Lösung gesetzt, denn:

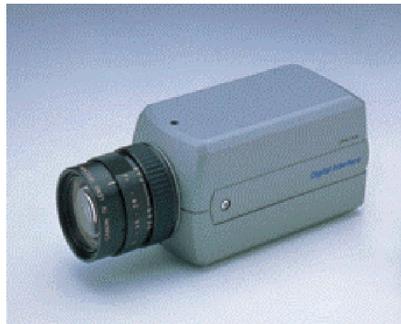
- Es gibt kein System, das sowohl in Gebäuden, als auch außerhalb gute Ergebnisse liefert
- Ein einzelnes System wäre ein Single Point Of Failure

# Nexus - Sensory Systems (Control Systems)

---

## Positionierung im Freien:

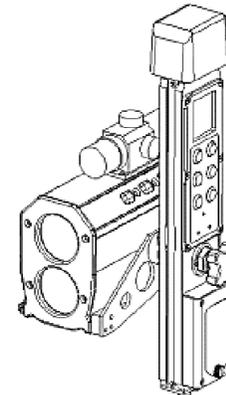
- DGPS (Garmin 25 LP DGPS Empfänger der den ALF Dienst nutzt)
- Digitale Kamera (Sony DFW-V500 mit Fire Wire (IEEE 1394))
- Zusätzliche Sensoren zur Bestimmung von Blickrichtung, Neigung und Distanz zu Objekten
- GSM



Digitale Kamera



LADIS System



Kompassmodul  
MAPSTAR

# Nexus - Sensory Systems (Control Systems)

---

## Positionierung in Gebäuden:

- Kamera
- Zusätzliche Sensoren zur Bestimmung von Blickrichtung, Neigung und Distanz zu Objekten
- Bluetooth

## Nexus - Sensory Systems (Control Systems)

---

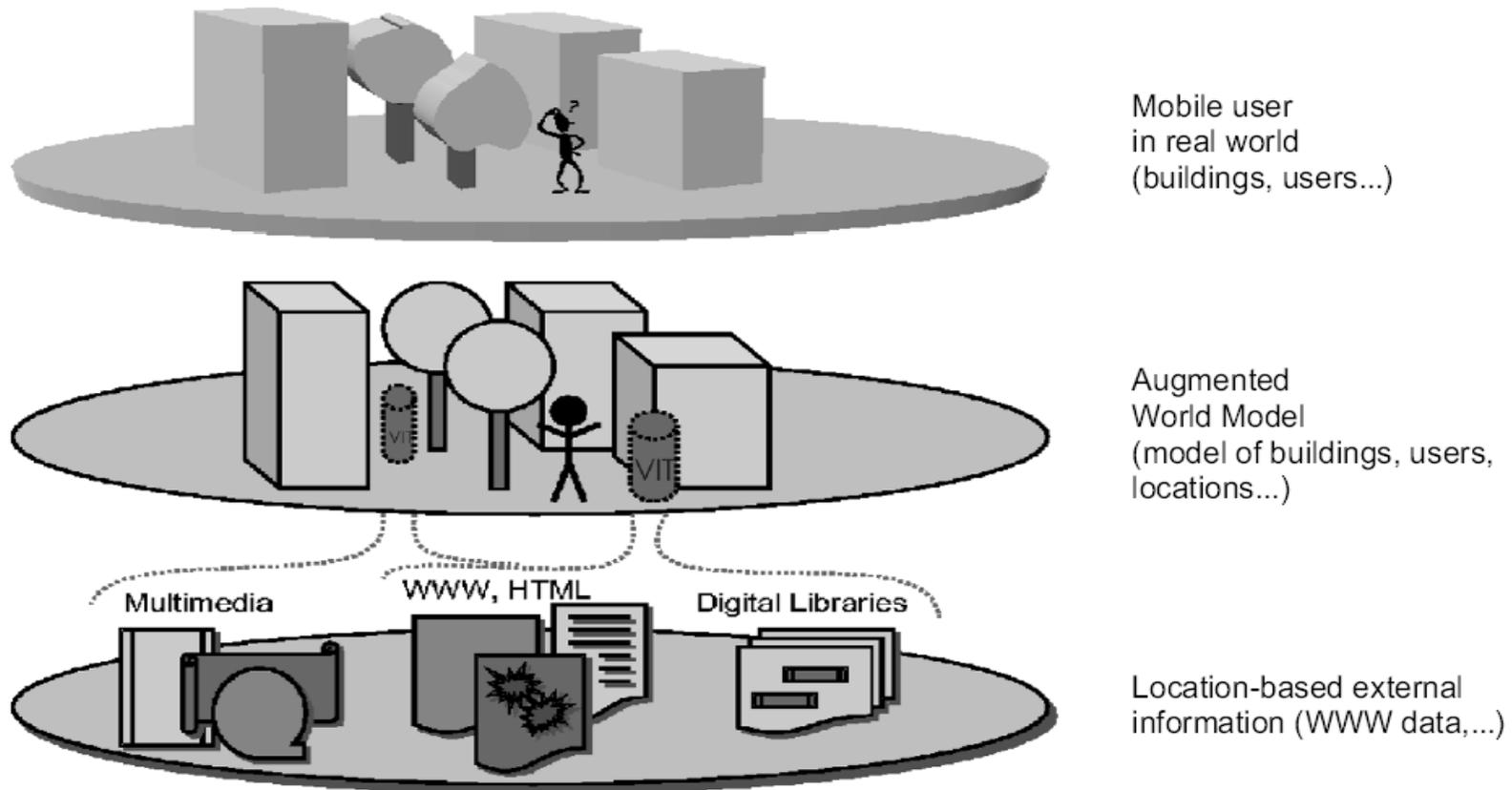
Eine weitere Aufgabe des Sensory Systems ist es die Informationen der verschiedenen Positionierungsdienste zu vereinen und sie in ein einziges Koordinatenpaar zu übersetzen. Dieses wird im WGS84 Format dargestellt, das auch von GPS verwendet wird.

## 2.5.4 Distributed Data Management

Dieser Teil der Nexus Architektur beschäftigt sich mit der Speicherung der Daten der sogenannten Augmented World sowie mit der Aktualisierung dieser Daten und mit der Anfragenbearbeitung der ortsbezogenen Anwendungen.

# Nexus – Distributed Data Management

Was ist eigentlich die Augmented World bzw. wie sieht sie aus?



# Nexus – Distributed Data Management

---

Die Augmented World setzt sich also aus drei verschiedene Objektarten zusammen:

- Geographische Objekte (z.B. Gebäude, Strassen oder Städte)
- Mobile Objekte (z.B. Nexus User)
- Virtuelle Objekte (z.B. ViLis, Virtual PostIts)

Die ganze veränderte Welt (Augmented World) basiert auf einem objektorientierten Ansatz, da dieser sehr leicht erweitert werden kann. So kann man zum Beispiel durch die Erstellung von Unterklassen zusätzliche Funktionalität hinzufügen.

# Nexus – Distributed Data Management

---

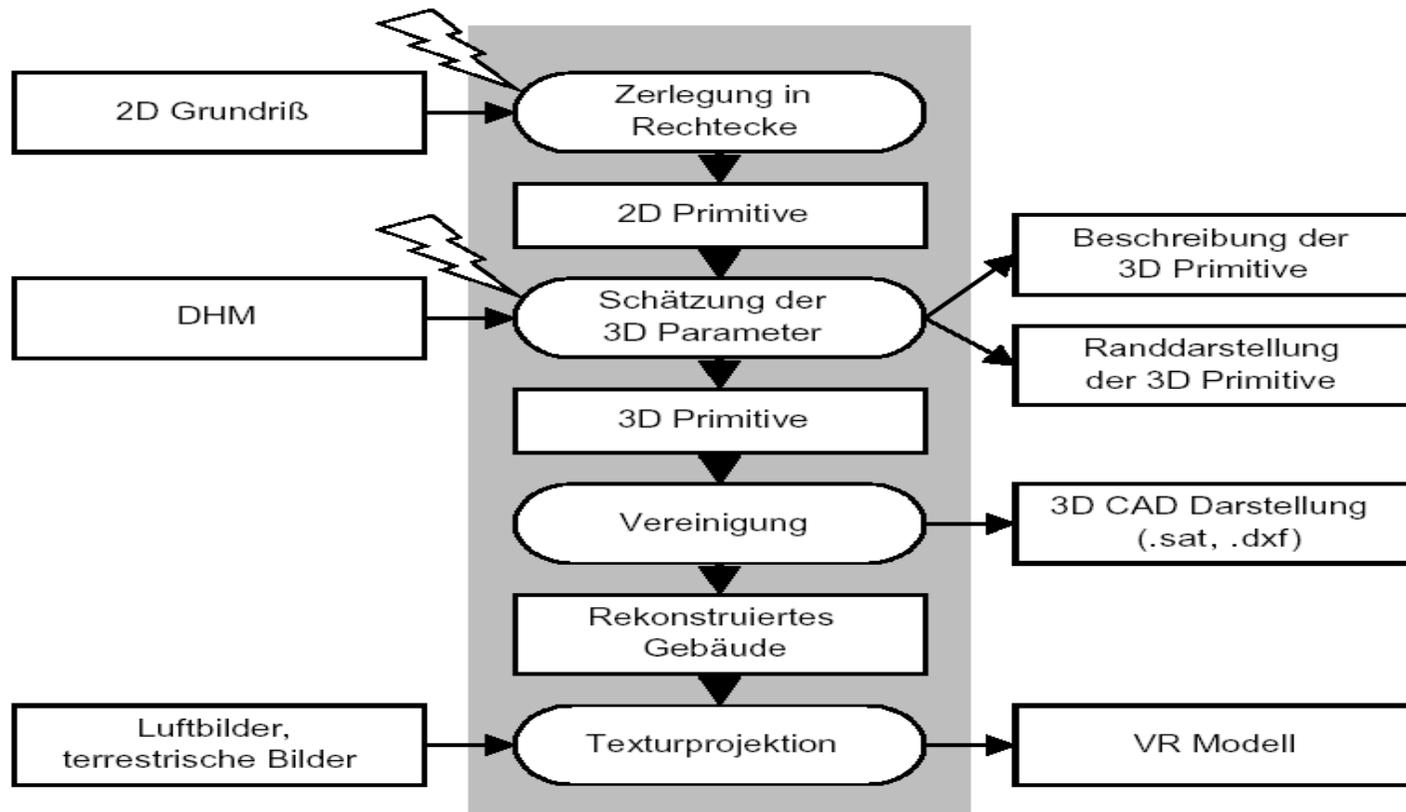
Wie entsteht die Augmented World?

Mögliche Lösung: manuelle Eingabe

Problem: große Datenmenge nicht zu bewältigen

Lösung: Auswertung von Luftbildern bzw. vorhandenen Grundrissen in Verbindung mit digitalem Höhenmodell (DHM)

# Nexus – Distributed Data Management



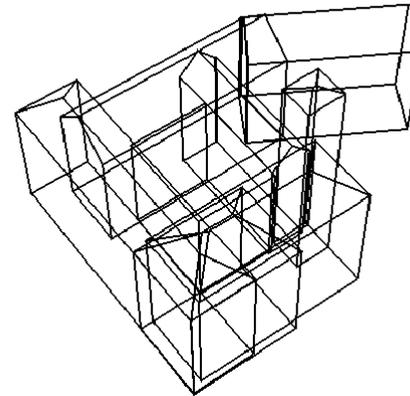
Ablauf der 3D Gebäuderekonstruktion

# Nexus – Distributed Data Management

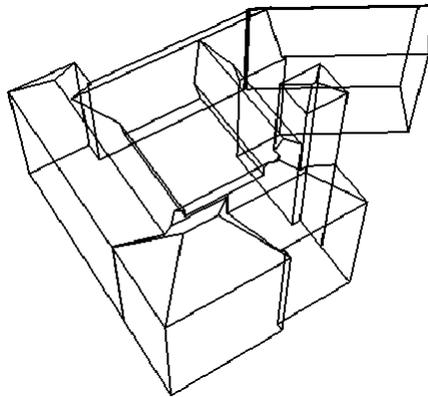
---



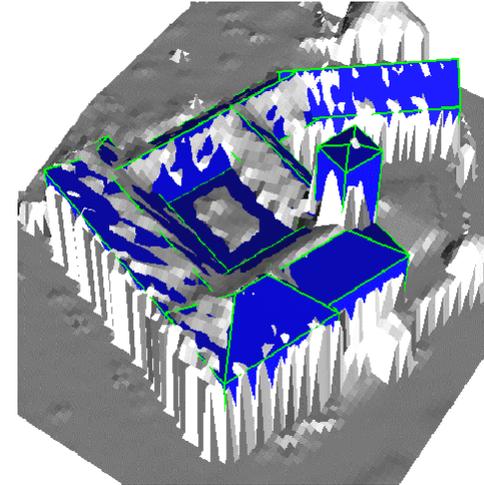
Rechteckszerlegung des Grundrisses



Rekonstruiere 3D Primitive



3D Randdarstellung des Gebäudes



Rekonstruiertes Gebäude und DHM

# Nexus – Distributed Data Management

---



Ergebnis der automatischen Gebäudekonstruktion am Beispiel eines 1,8 km x 2,3 km großen Teilgebiets von Stuttgart mit 5208 Gebäuden

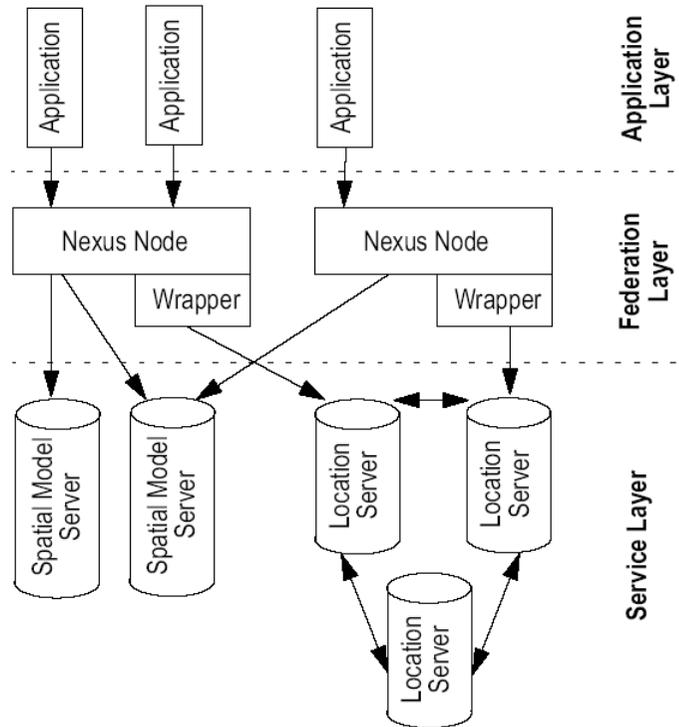
## Nexus – Distributed Data Management



Teil Stuttgarts als 3D Modell mit zum Teil texturierten Flächen. Die Informationen für die Texturen wurden aus Luftbildern oder terrestrischen Aufnahmen gewonnen.

# Nexus – Distributed Data Management

Wie wird nun dieses AW Modell gespeichert ?



# Nexus – Distributed Data Management

---

## *Service Layer*

Hier werden die Daten über die verschiedenen Objekte des AW Modells gespeichert.

Es wird unterschieden zwischen sogenannten:

- *Spatial Model Servern*
- *Location Servern*

## *Spatial Model Server*

Jeder *Spatial Model Server (SMS)* enthält Daten zu statischen geographischen Objekten innerhalb eines gewissen räumlichen Gebiets (z.B. 2D und 3D Ansichten von Gebäuden in Stuttgart).

Die *Spatial Model Servers* werden vom *Area Service Register* verwaltet, dieses ist eine Art geographischer Domain Name Service. Im *Area Service Register* wird gespeichert welcher *SMS* für welches Gebiet zuständig ist.

Die Daten werden in einem herkömmlichen Datenbanksystem mit Erweiterung für Ortsdaten abgelegt (z.B. DBMS)

# Nexus – Distributed Data Management

---

## Location Server

Ein *Location Server* ist das Pendant zu einem *SMS*, im Bezug auf mobile Objekte. Er enthält Informationen über die Registrierung der einzelnen Benutzer in einem bestimmten Bereich. Auch diese werden in einer herkömmlichen Datenbank abgelegt.

Da ein Datenbanksystem nicht auf häufige Updates ausgelegt ist, musste für die Positionsdaten der mobilen Objekte eine andere Lösung gefunden werden. Sie werden im sogenannten *Location Service* gespeichert.

Dieser verwaltet die Daten in einem Quad-tree in seinem Hauptspeicher, außerdem ist er verantwortlich für die Weiterleitung von Anfragen und für Handover zwischen den einzelnen *Location Servern*.

## *Federation Layer*

Das *Federation Layer* ist das Bindeglied zwischen der Anwendungsschicht und dem Datenmanagement. Es wird benötigt um die verteilte Datenspeicherung vor den Anwendungen zu verbergen. Sie leitet Anfragen an die jeweils zuständigen *SMS* oder *Location Server* weiter und berechnet aus den Ergebnissen eine eindeutige Antwort. Diese wird in einer einheitlichen Semantik präsentiert, in der sogenannten *Augmented World Modelling Language (AWML)*. Entsprechend werden die Anfragen der Anwendungen in der *Augmented World Querying Language (AWQL)* gestellt. Beides sind XML basierte Sprachen.

## Praktische Ergebnisse

Bis jetzt wurde ein Prototyp eines *Location Service* gebaut, der pro Sekunde 2000 Updates und 1000 Anfragen auf einer Standard Solaris Workstation bearbeiten kann.

An einem kleinen Prototyp der gesamten Plattform und einigen Beispielanwendungen wird im Moment gerade gearbeitet um die Grundeigenschaften des Systems demonstrieren zu können.