

# Ein Framework für kontextbezogene Anwendungen in der Nexus-Plattform



Matthias Wieland  
Universität Stuttgart  
[matthias.wieland@iaas.uni-stuttgart.de](mailto:matthias.wieland@iaas.uni-stuttgart.de)

# Überblick



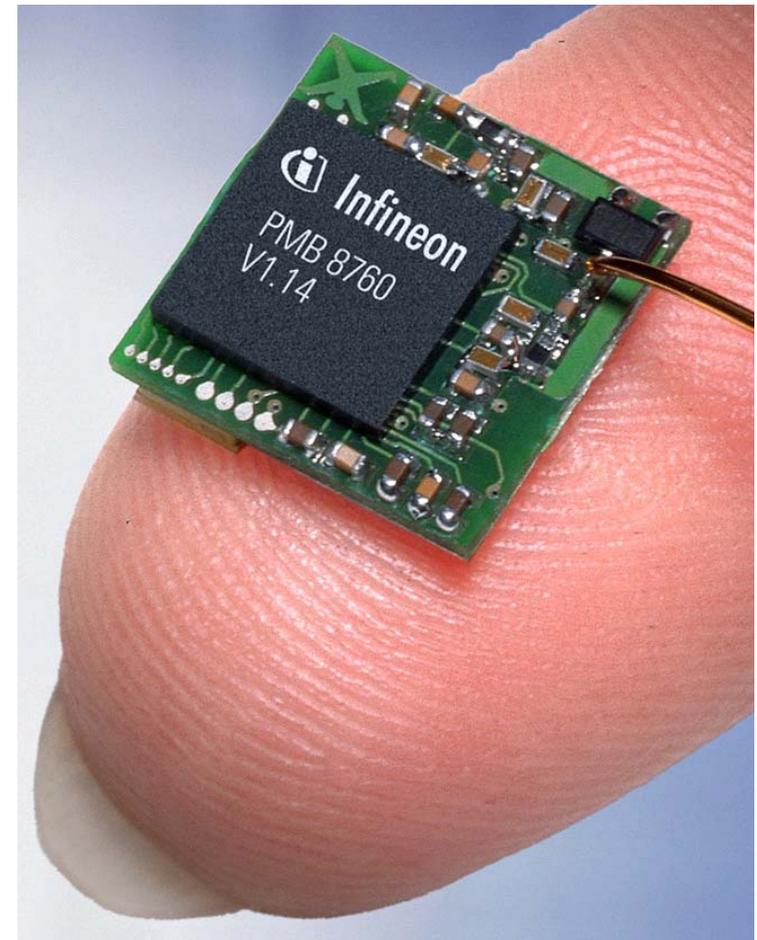
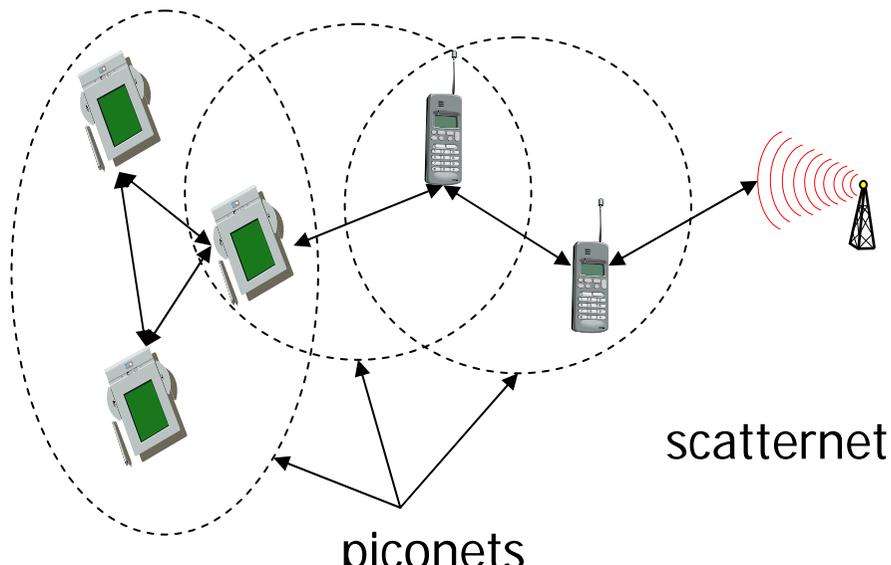
- Motivation
- Nexus Projekt
  - Vision, Forschungsziele
- Nexus-Plattform
  - Architektur, Datenmanagement, Dienste, Zugriffsmethoden
- Anwendungs-Framework
  - Funktionalität, Aufbau, Zweck
- Ausblick
  - Anwendungsunterstützung durch Workflows, Situationen, Debugging, Test



# Trend: Kommunikationstechnologie



- “Ubiquitous Communication”: Wireless WAN, LAN, PAN
- Günstige, kleine Kommunikationsgeräte verfügbar
- Beispiel: UMTS
  - Bis zu 2Mbit/s
- Beispiel: Bluetooth
  - 1 Mbit/s Datenrate
  - ~ 10 m Reichweite
  - Einzelchip-Lösung
    - bei geringem Stromverbrauch: 0.001 W
    - niedriger Preis: < 4 Euro



Quelle: Infineon Technologies

# Trend: Sensorintegration



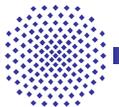
- Große Vielfalt an Sensoren
- Integriert in
  - mobilen Geräten
  - Gegenständen der realen Welt
- Beispiel: Smart-Its
- Beispiel: Smart Dust
  - Vollständige Kommunikations- und Sensorplattform
  - Sehr kleine Größe (vgl. Sandkorn)
  - Günstig: Ausbringen von sehr vielen unabhängigen Einheiten
  - Sensoren für: Licht, Beschleunigung, ...
  - Stromversorgung mit Solarenergie



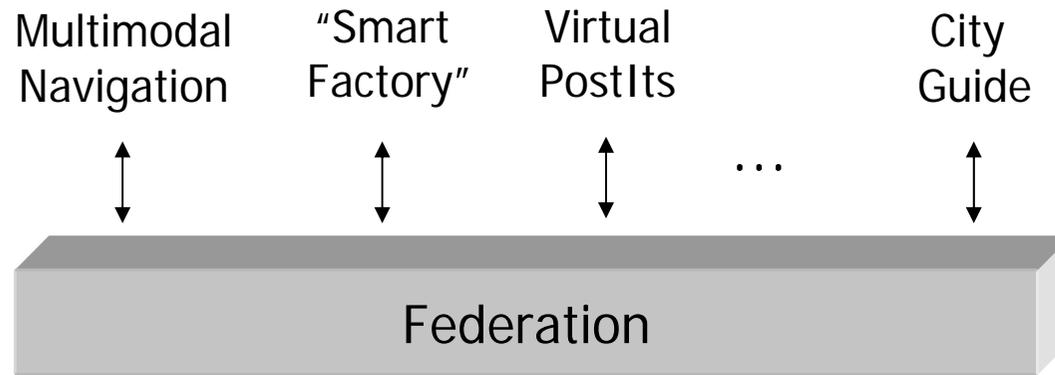
# Kontextbezogene Anwendungen



- Navigation
  - Fahrzeug-, Fußgänger-, multimodal
- Informationssysteme
  - Touristen, Hausführungen, ...
- Kommunikationsdienste
  - GeoCast, bestes Netz, ...
- Lernszenarien/Spiele
  - Rallyes, "mixed reality games"
- intelligente Umgebungen
  - smart room, building, factory
- Erinnerungsdienste
  - Rucksack komplett? Schuhe kaufen?



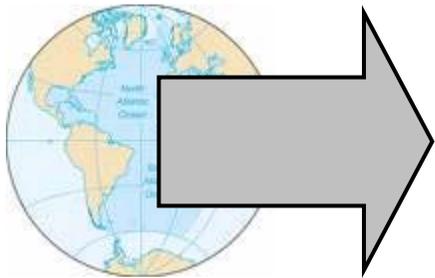
## Vision: Föderierte Umgebungsmodelle



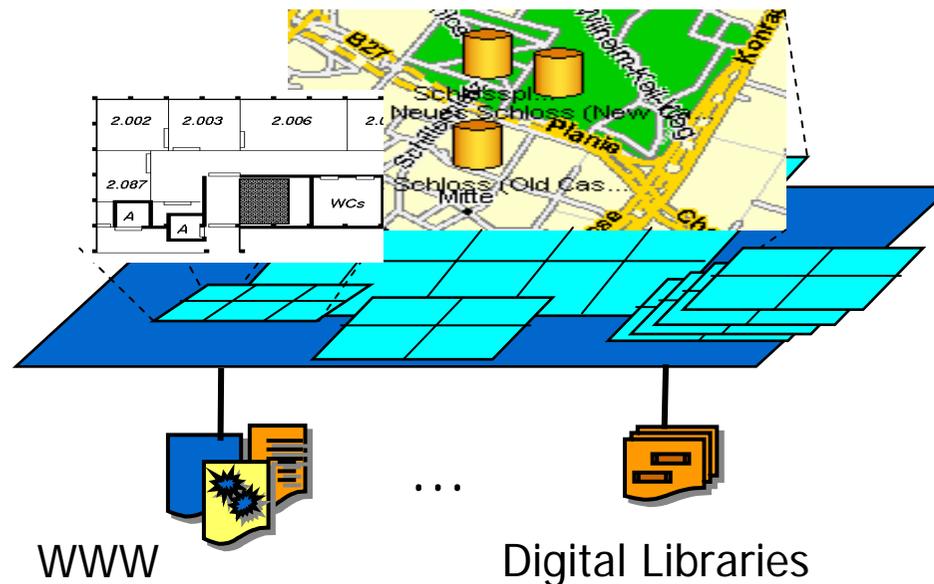
kontextbezogene Anwendungen

Globales Kontextmodell

Daten von



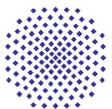
sehr vielen Sensoren



Lokale Kontextmodelle

digitale Informationsräume

=> Verknüpft die physische mit der digitalen Welt

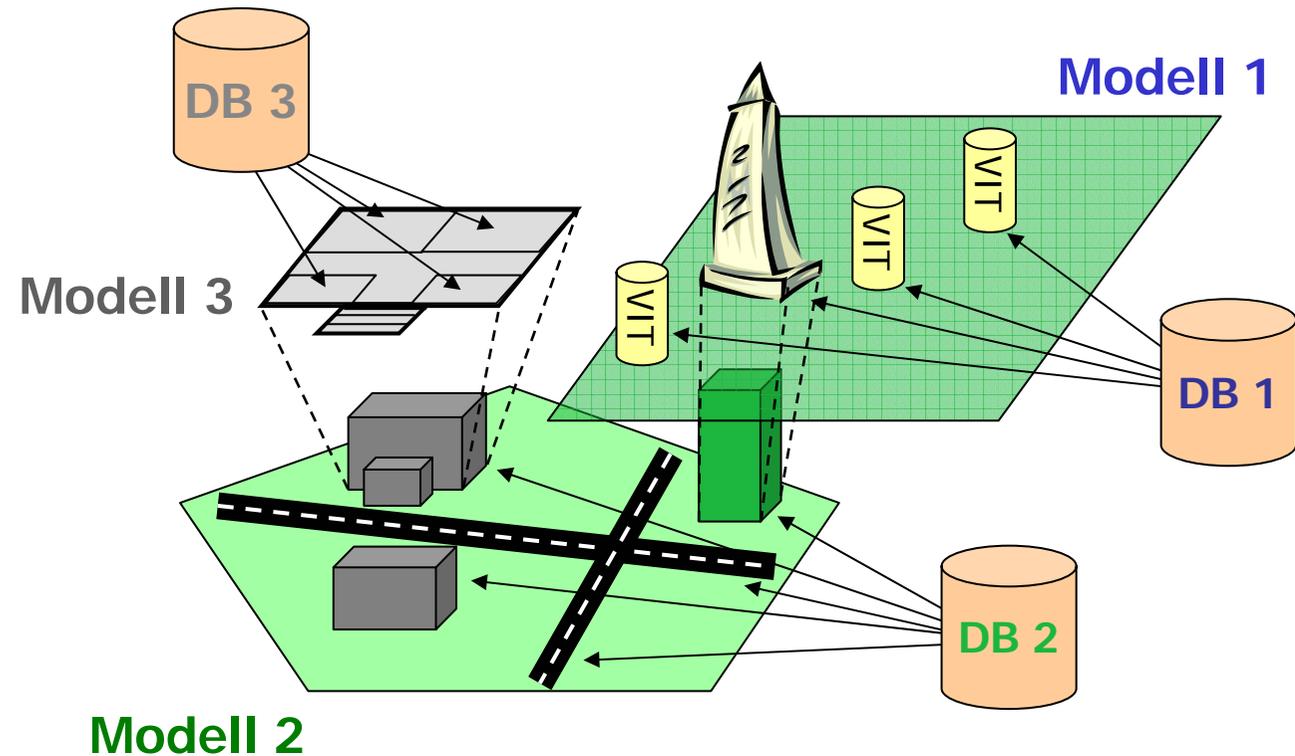


# Lokale Kontextmodelle

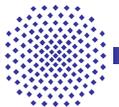


Ein lokales Kontextmodell:

- deckt ein begrenztes räumliches Gebiet ab
- enthält bestimmte Kontextinformation
- wird komplett bei einem Datendienst verwaltet



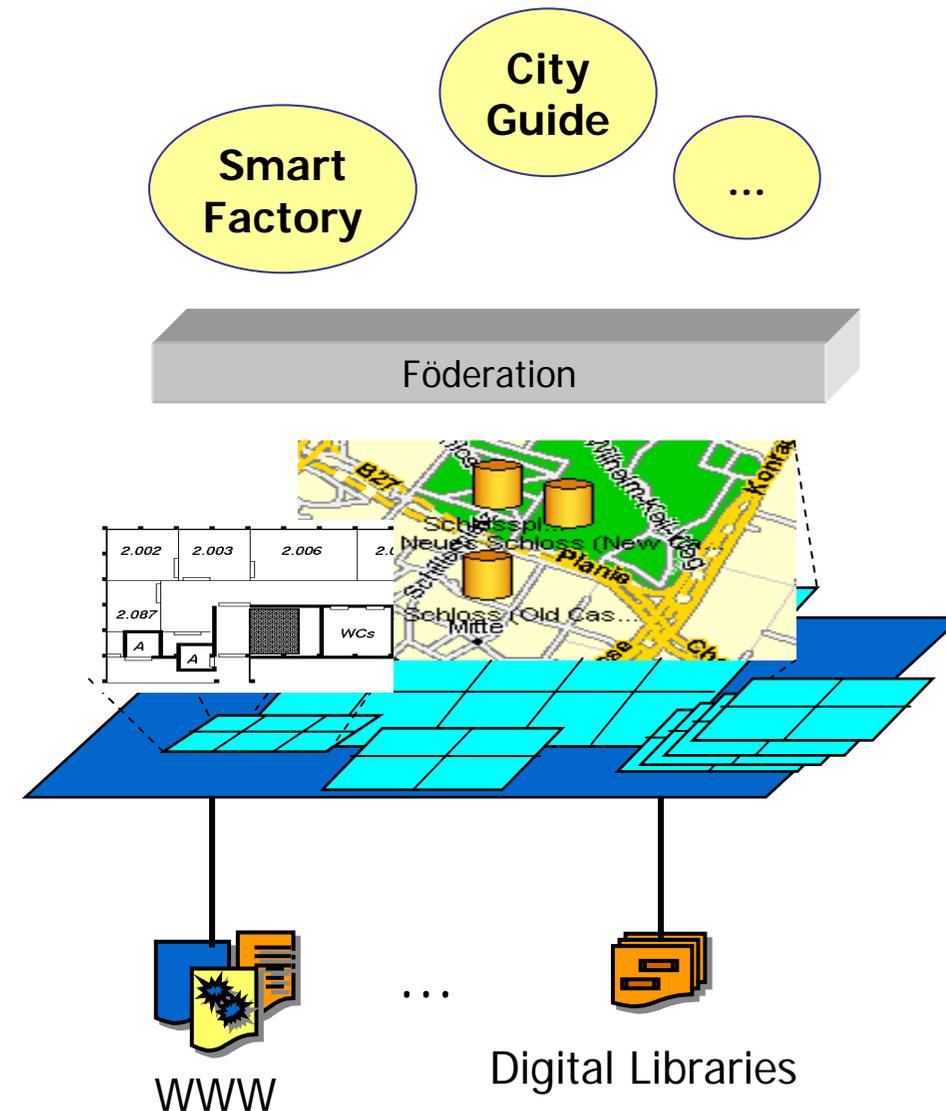
- räumliche und inhaltliche Überlappung:
  - ➔ Mehrfachrepräsentationen von Realweltobjekten



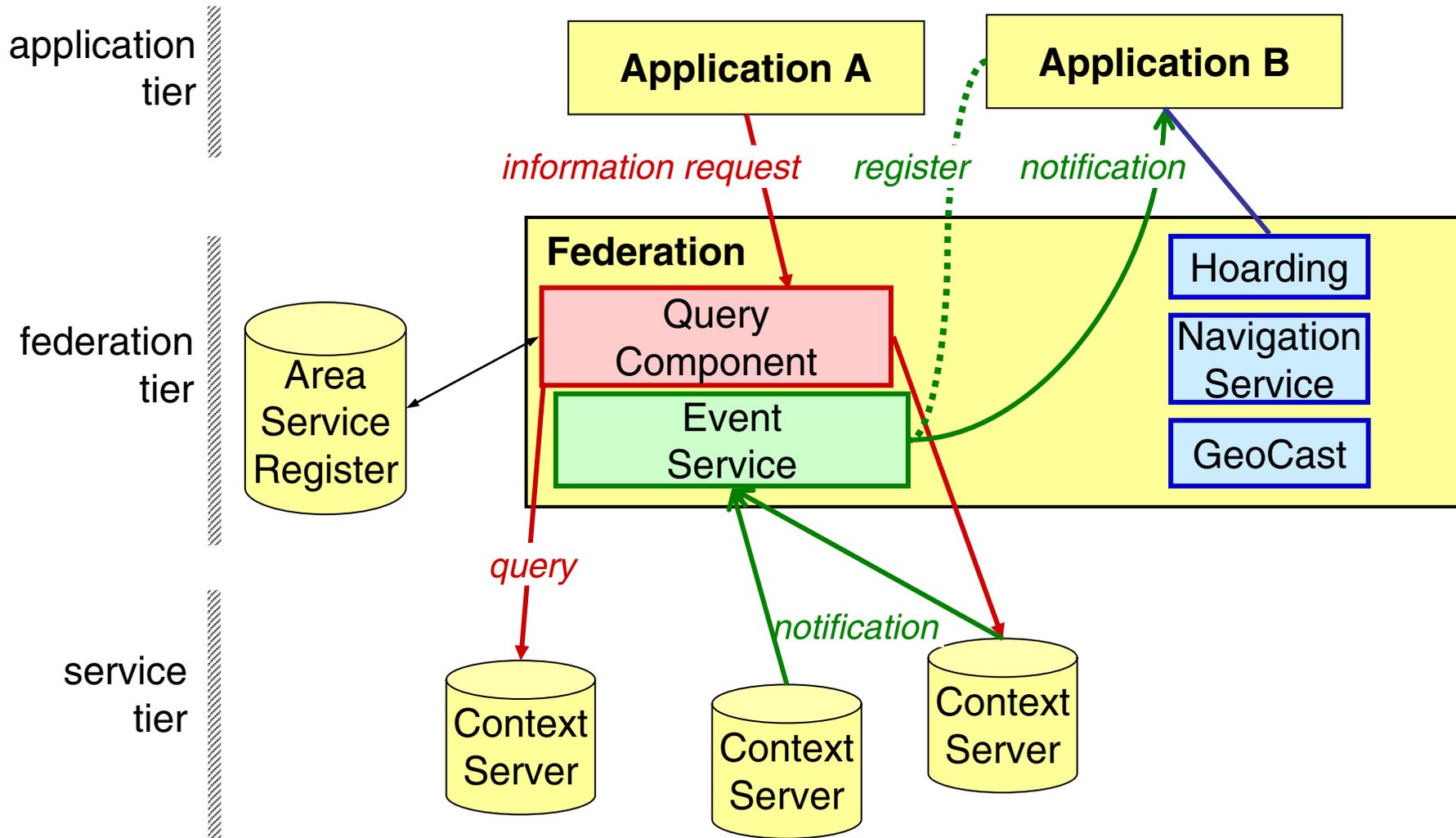
# Globales Kontextmodell



- Föderation lokaler Kontextmodelle
  - Wiederverwendung von Modelldaten, denn Modellierung ist teuer!
  - Datendienste bleiben autonom
- Nutzt räumliche Struktur als Integrationskriterium
  - Globales Integrationsschema
  - Verbindet lokale Kontextmodelle zu einem globalen Bild
- Offen: neue Datendienste, neue Anwendungen

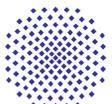


# Zugriffsmethoden auf die Nexus-Plattform



→ **Queries**

→ **Events**



# Überblick



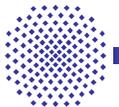
- Motivation
- Nexus Projekt
  - Vision, Forschungsziele
- Nexus-Plattform
  - Datenmanagement, Architektur, Dienste, Zugriffsmethoden
- Anwendungs-Framework
  - Funktionalität, Aufbau, Zweck
- Ausblick
  - Anwendungsunterstützung durch Workflows, Situationen, Debugging, Test



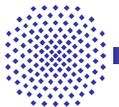
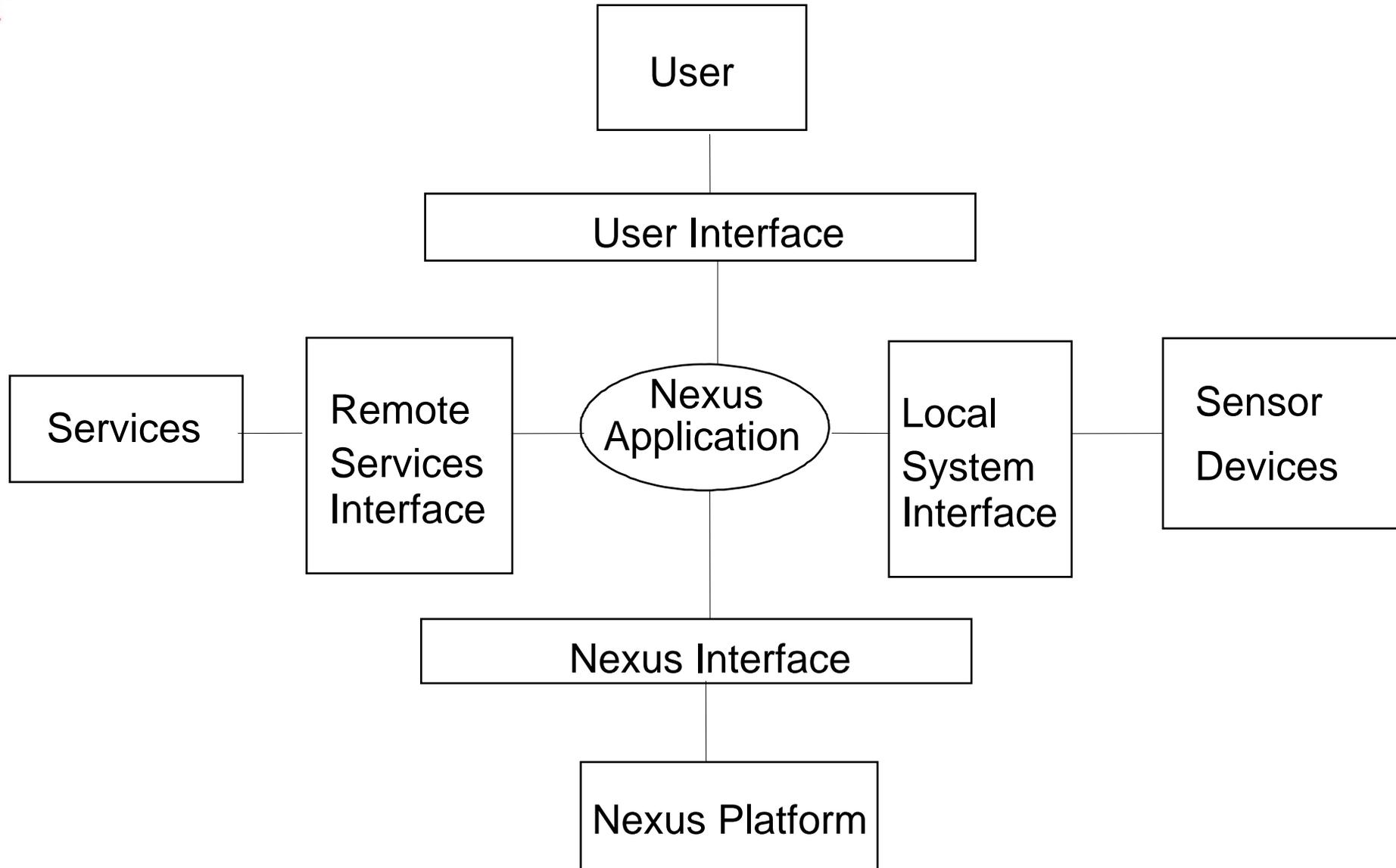
# Motivation für die Entwicklung des Framework



- Durch die Entwicklung mehrerer kontextbezogener Anwendungen hat sich gezeigt, dass:
  - dies komplex ist
  - viel Zeit benötigt wird
  - dabei vieles doppelt implementiert wird
  - zusätzliche Funktionen benötigt werden
  
- Das Framework hilft diese Punkte zu verbessern



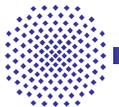
# Aufgabenbereiche des Frameworks



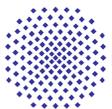
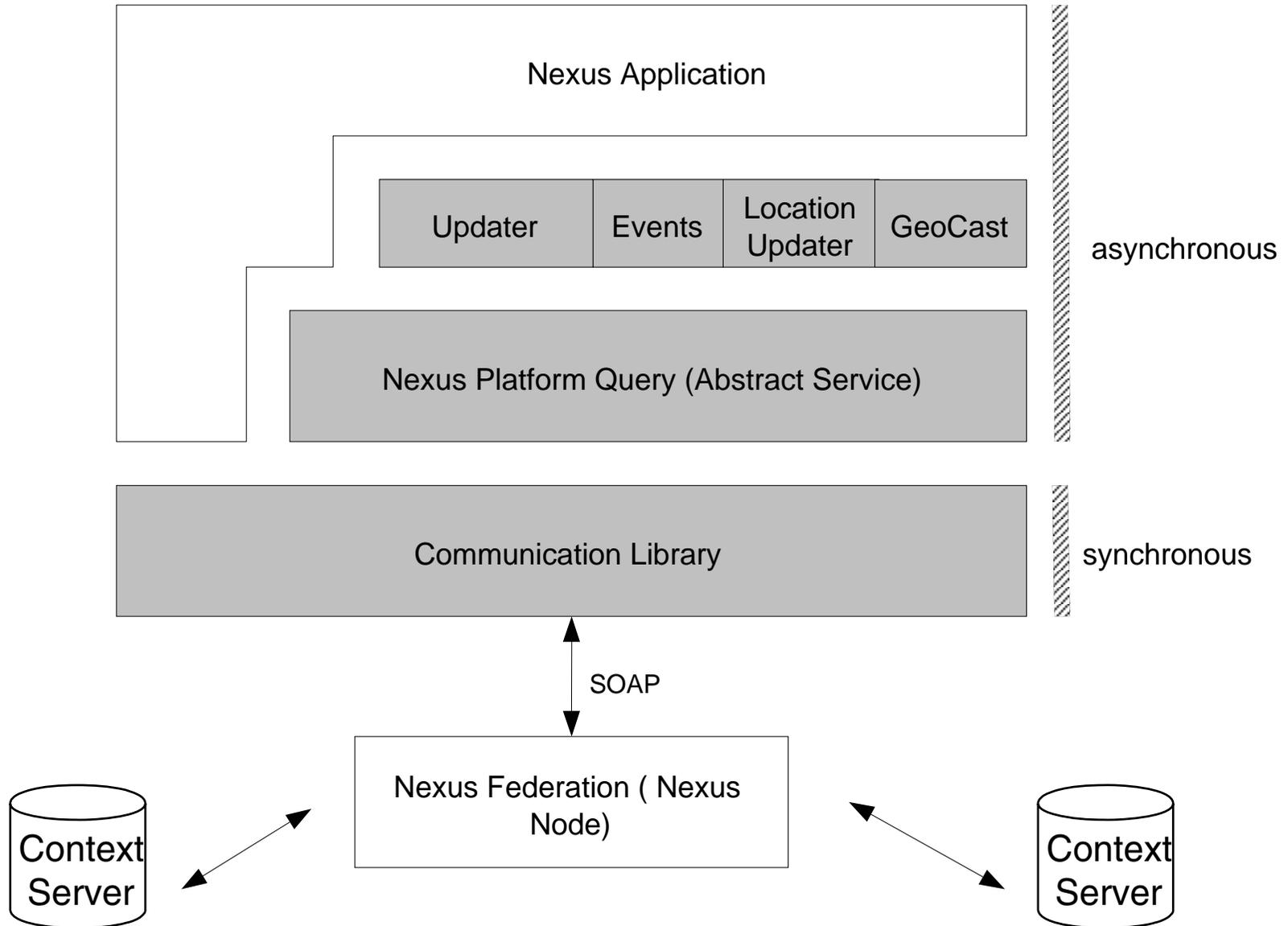
# Konzepte des Frameworks



- **Komponentenbasiertes verteiltes System**
  - Modularität, Kapselung von unbeständigen Implementierungsdetails
  - Wiederverwendbarkeit durch Definition stabiler Schnittstellen
- **Ereignisbasiertes System**
  - Umkehr des Kontrollflusses: „*Don't call us, we call you*“ (Hollywood Prinzip)
- **Modulare Kommunikationsarchitektur**
- **Lokale Kontexterfassung**
  - ereignisbasiert
  - robust
- **Laufzeit-Unterstützung**
  - Konfiguration
  - Verwaltung von Komponenten und Anwendungen

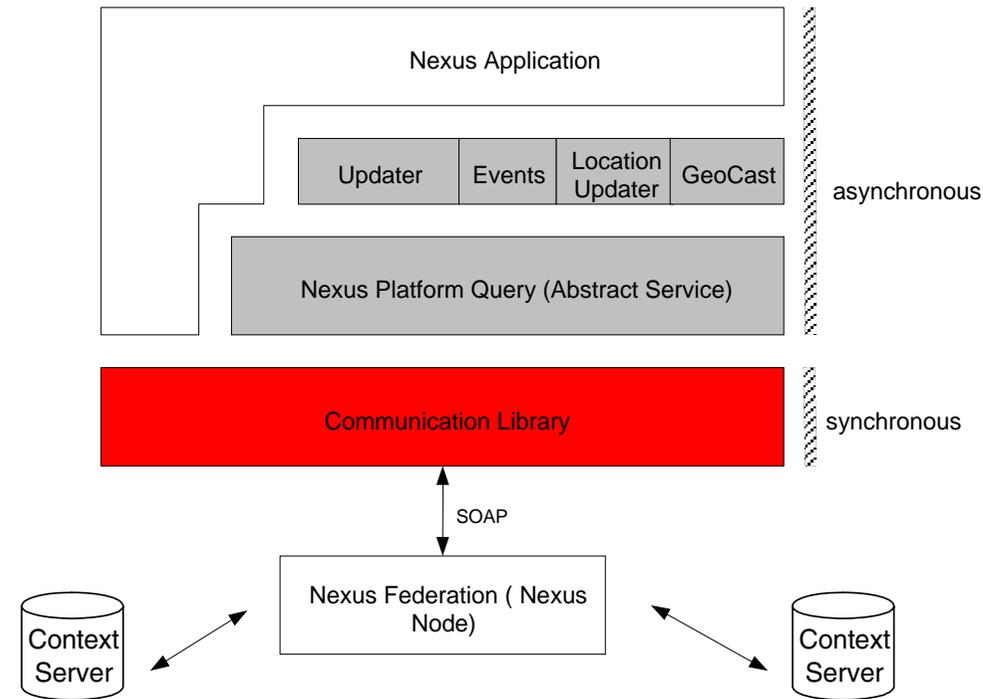


# Kommunikationsarchitektur des Frameworks



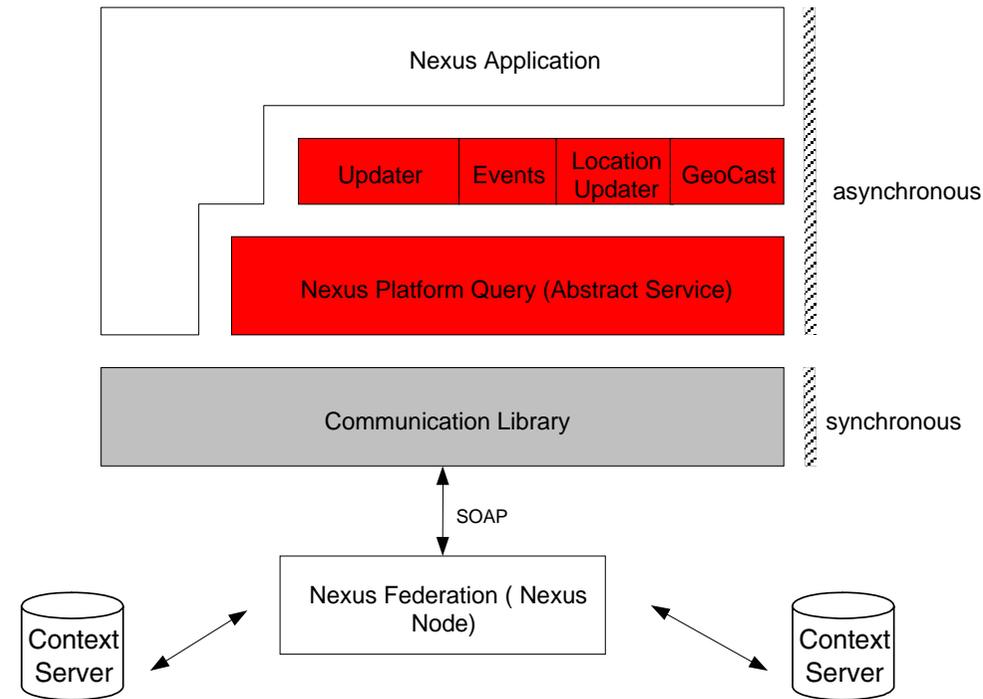


- synchroner Zugriff auf Nexus-Server
- Auswahl der Kommunikationsmethode
- Server-Stubs
- dynamisches Aufrufen von Webservice-Methoden
- Server-Schnittstellen



## Asynchrone Kommunikation

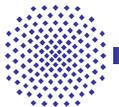
- asynchroner Zugriff auf Nexus-Dienste
- Benachrichtigungssystem
- Umgebungsmodell-Anfragen (delete, insert, update, query)
- Map, Navigation Service
- Event Service, Geo Cast
- kontinuierliche Aktualisierung
- persistente Datenobjekte



# Paket *sensors* – Akquise von Sensordaten



- Kapselung der Sensoren
  - Einheitlicher Zugriff
  - Gleichzeitiger Zugriff
- Positionierungssystem
  - Auswahl der zu nutzenden Sensoren
  - Robustes System
  - Zugriff auch über Prozessgrenzen hinweg
- Event-System
  - Asynchronität
  - Sensorereignisse



# Definierte lokale Sensor-Ereignisse



## Basis-Ereignisse:

- onPosUpdate
- contPosUpdate
- onAccuracyChanged

## Erweiterte Ereignisse:

- onEnterArea(Ort)
- onLeaveArea(Area)
- distPosUpdate(Distance)
- onRouteLost(Route)
- onTimeInterval(Duration)



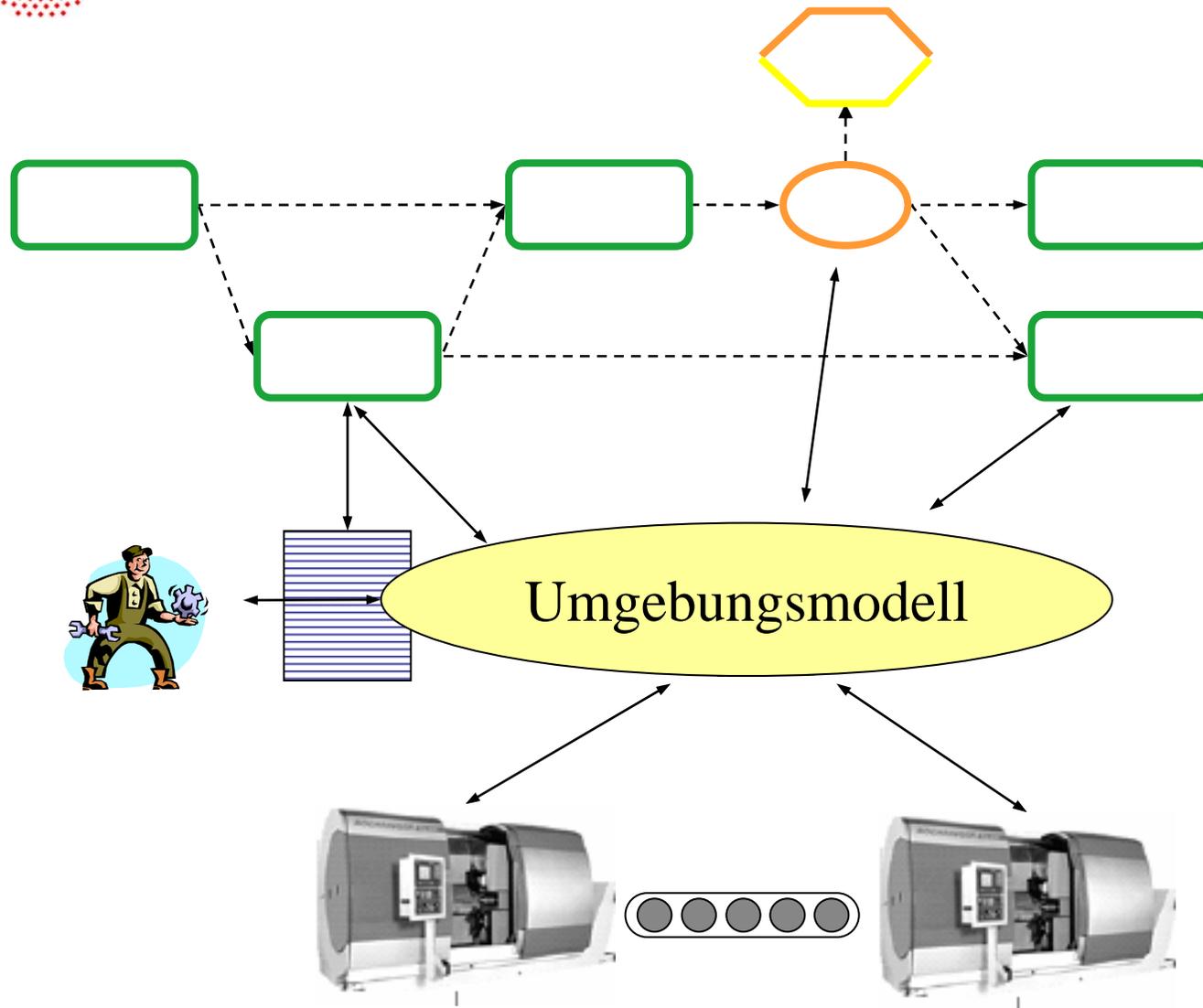
# Zusammenfassung und Ausblick



- Es hat sich gezeigt
  - Anwendungsunterstützung wichtig für die Entwicklung kontextbezogener Anwendungen
  - Das Framework reicht noch nicht aus
- Gründung eines neuen Projektbereichs - Anwendungsunterstützung mit den Teilprojekten
  - Kontextbezogene Workflows
  - Entwicklungswerkzeuge für kontextbezogene Anwendungen (Nachwuchsgruppe)
  - Verteilte Bestimmung von Situationen auf der Basis bewerteter Kontextdaten



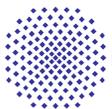
# Unterstützung von Produktionsprozessen



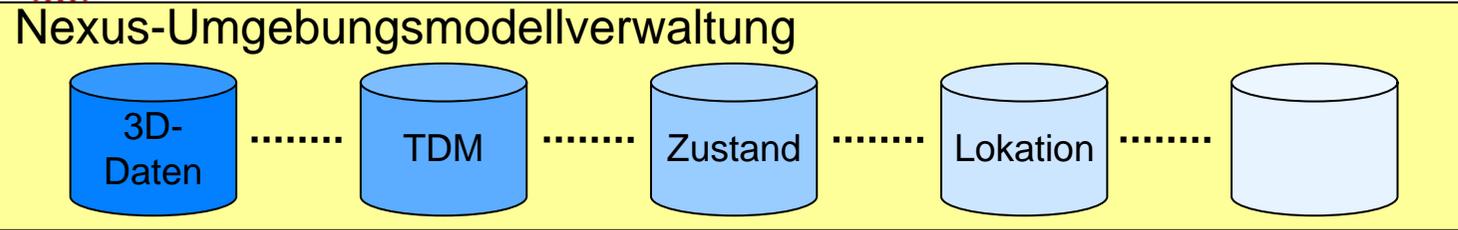
Geschäftsprozesse

mobile Agenten  
(Arbeitsabläufe)

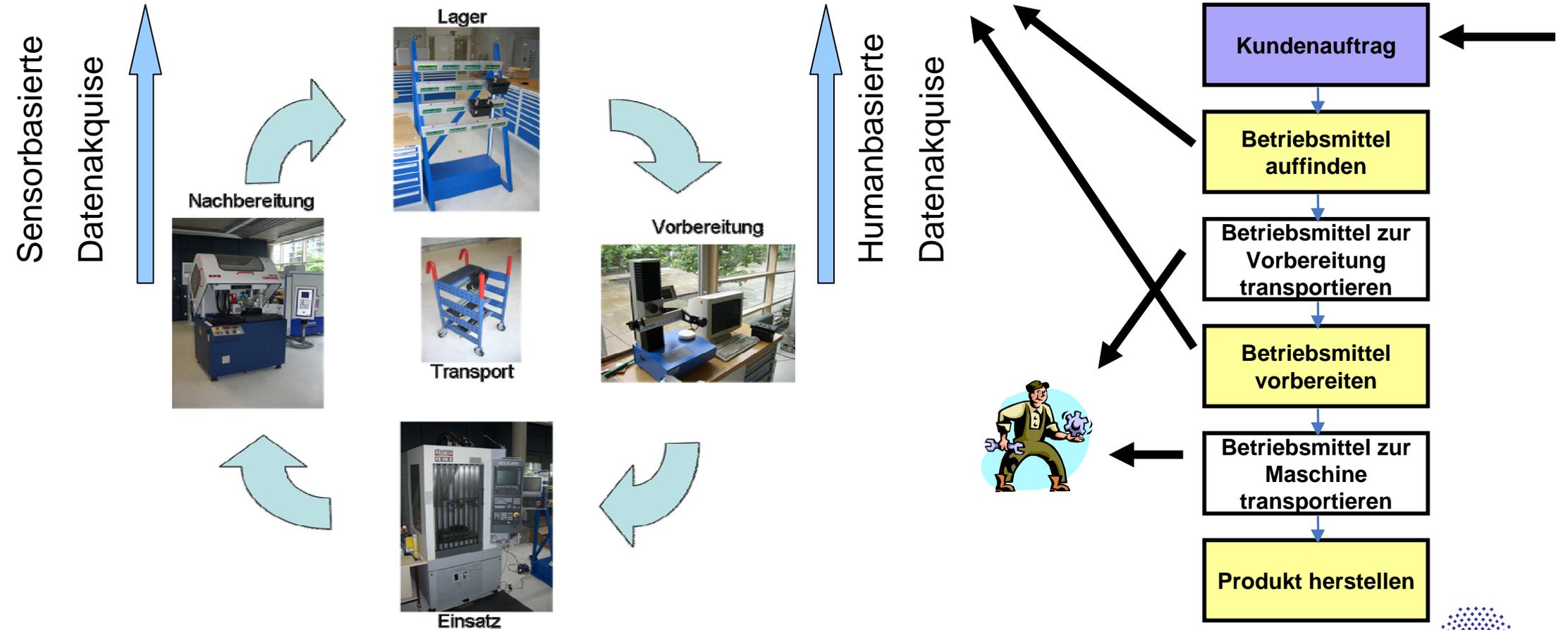
technische Prozesse  
(Produktion)



# Prototyp eines kontextbezogenen Workflows



- Produktionsumfeld ferne Teilnehmer (Kunden) und Systeme
- Funktionalitäten des Betriebsmittelmanagement der Smart Factory (Nexus-Teilprojekt)





[www.nexus.uni-stuttgart.de](http://www.nexus.uni-stuttgart.de)