Eine nachrichtenbasierte Architektur für Smart Homes

Dennis Hollatz
Hochschule für Angewandte Wissenschaften Hamburg

30.06.2010

Gliederung

- Smart Homes: Wer, wie, was?
- Living Place Hamburg
- Bisherige Realisierung
- Architekturvorschlag
- Diskussion

Ubiquitous Computing, Pervasive Computing und Ambient Intelligence

Mark Weiser in „The Scientific American“ (1991):

„There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment, instead of forcing humans to enter theirs, will make using a computer as refreshing as taking a walk in the woods.“

Ubiquitous Computing

- Vision:
  - Überall Computer
  - Hochspezialisiert
  - Verwendung wird kaum wahrgenommen werden

Pervasive Computing

- Ubicomp + Mobilität
- Kurz- und Mittelfristige Realisierbarkeit
- Industriell geprägt

Ambient Intelligence

- Wieder mehr Wissenschaft

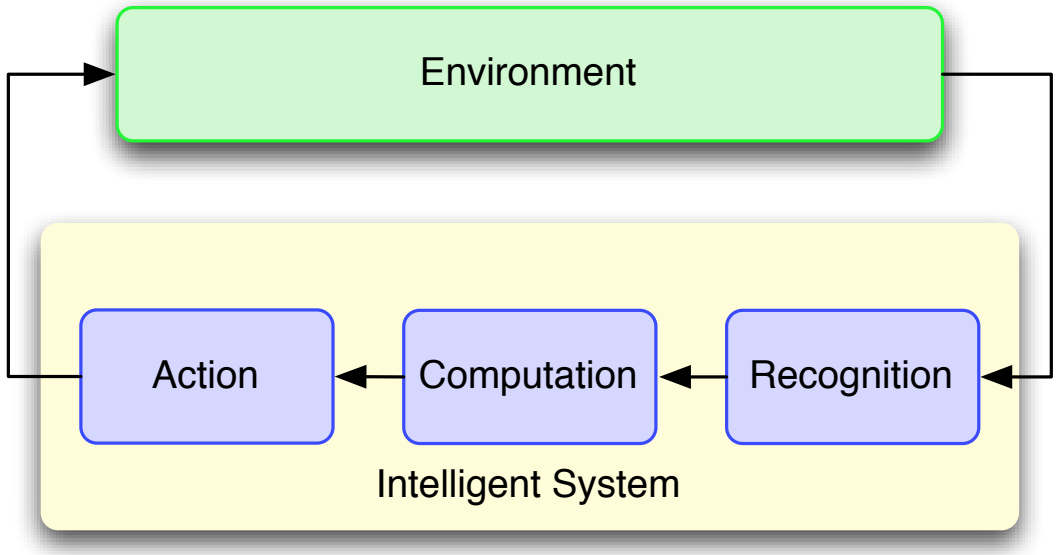
Wichtig:

- Konzepte
- Architekturen

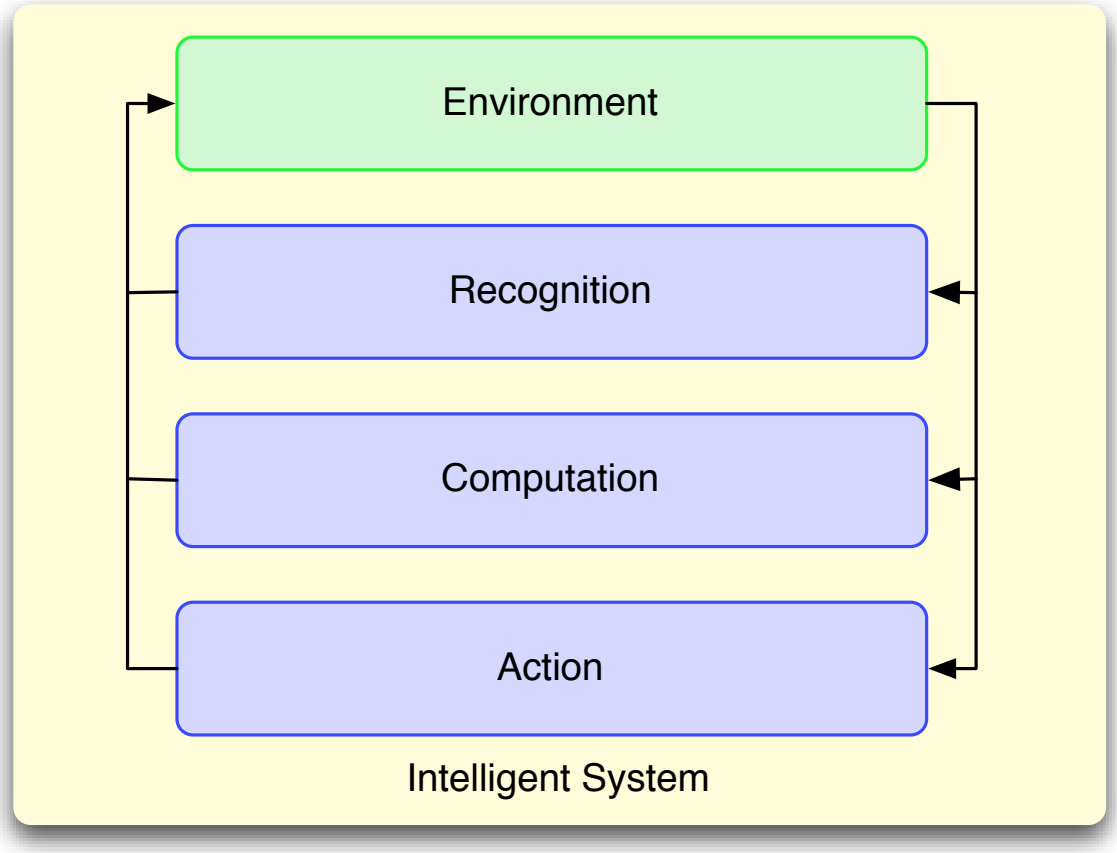
Smart Homes

- Interaktion & Usability
- Fokussierung der Herangehensweise auf den Wohnbereich
- Neu: Einbezug der Umgebung in die Untersuchungen

Interaktion von System und Umwelt - eine veraltete Sichtweise

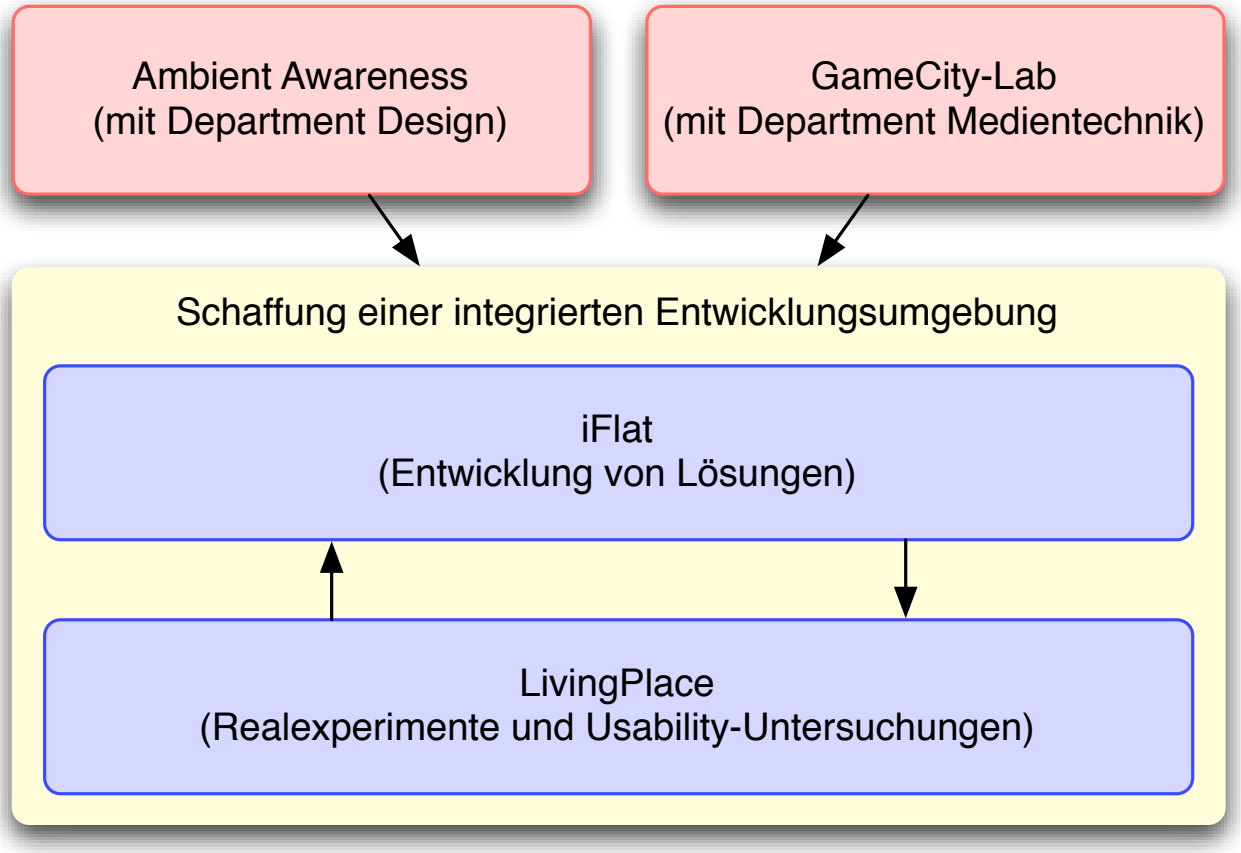


Umwelt als Teil der Systemsicht

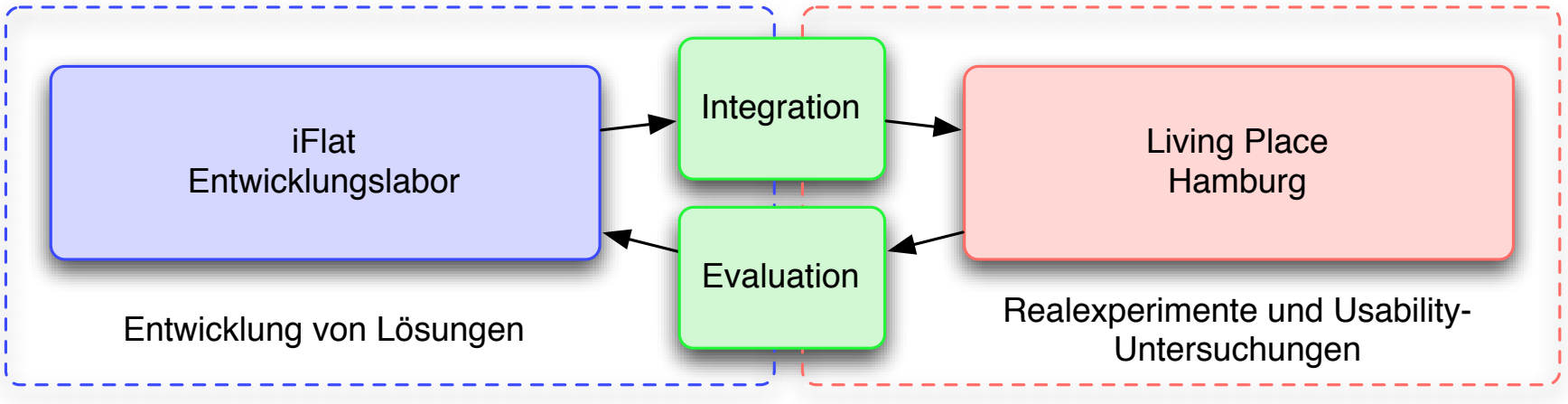


Das Living Place Hamburg

iFlat & Living Place: Einflussbereiche



iFlat & Living Place: Rollen



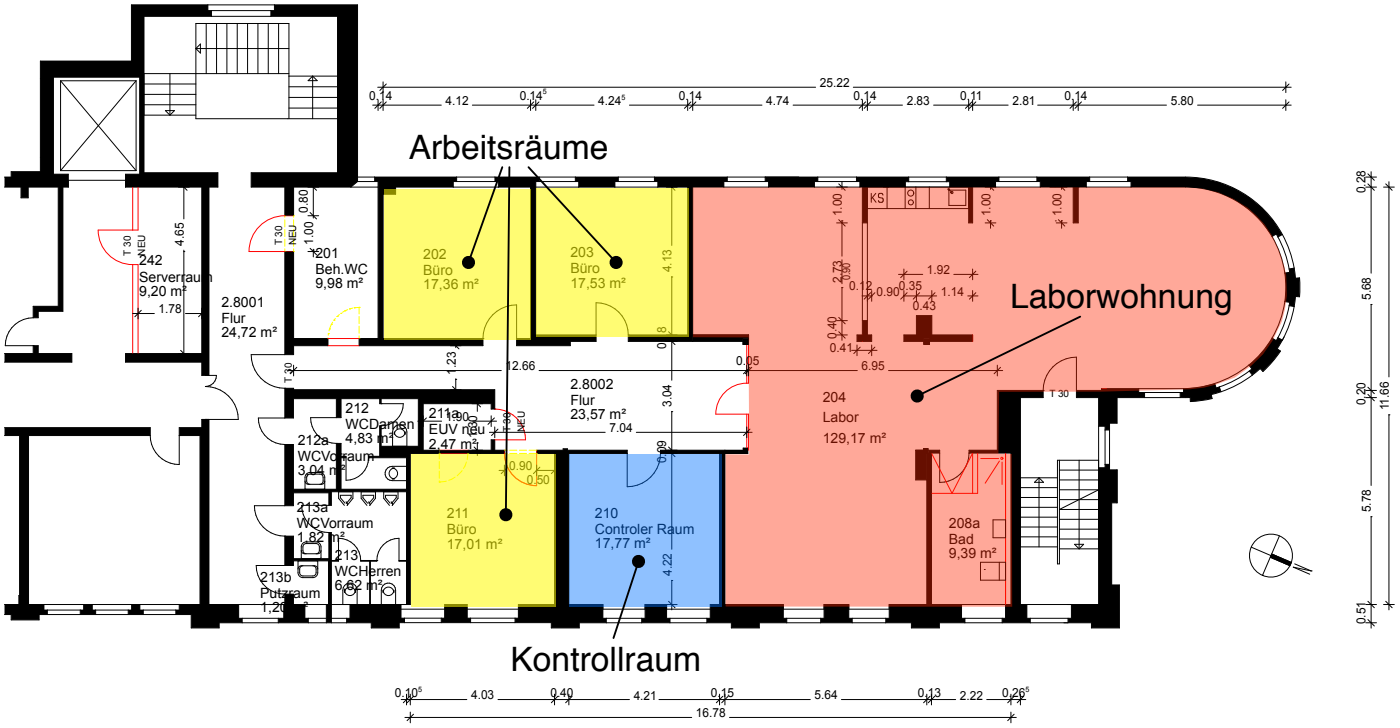
Living Place Hamburg

- iFlat & Living Place

- iFlat:
  - Prototypische Entwicklung
  - Experimente

- Living Place:
  - Integration
  - Langzeit-Usability-Studien

Grundriss des Labors

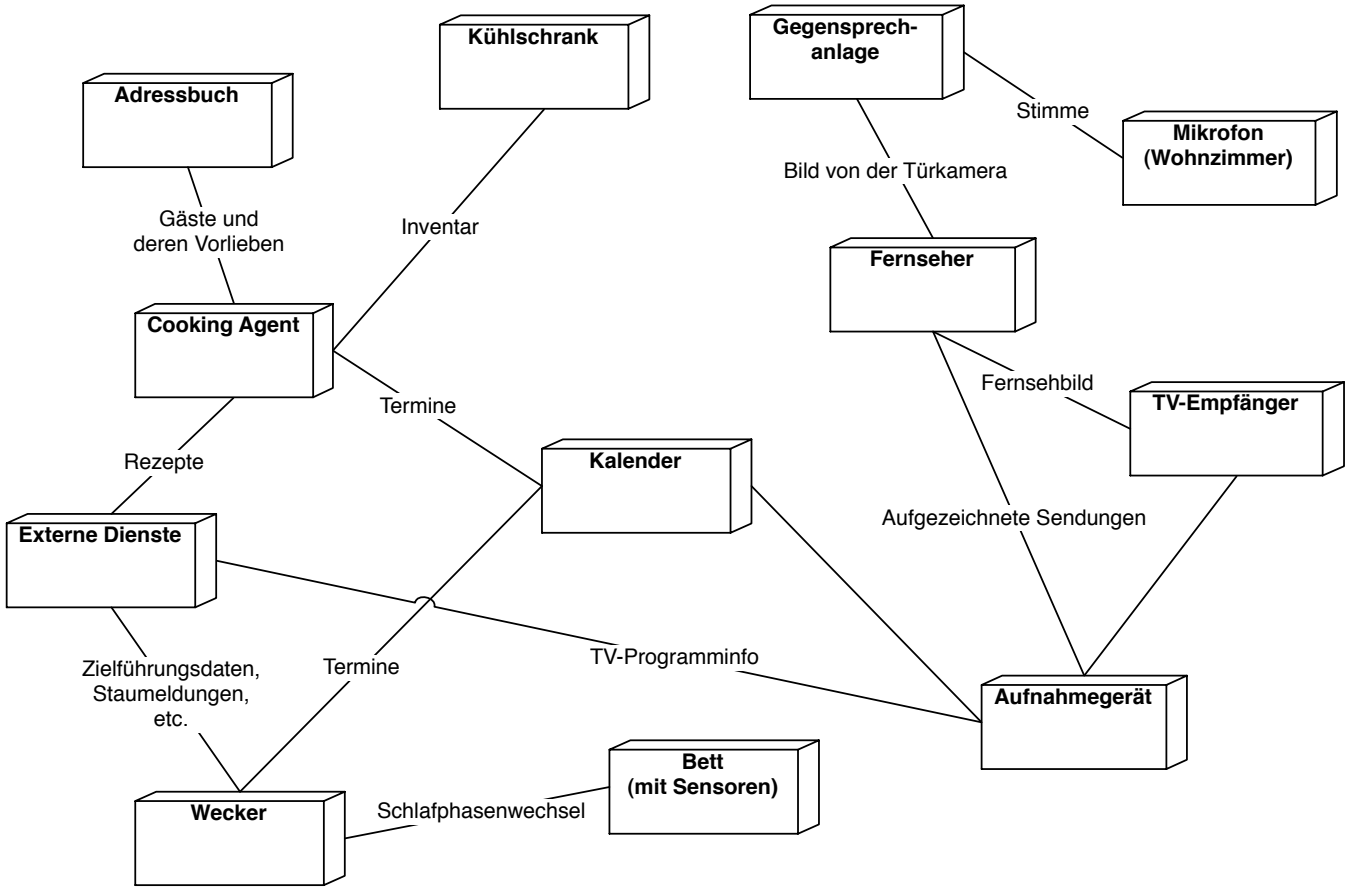


HAW Hamburg -Baumanagement- Berliner Tor 5, 20099 Hamburg			Planbezeichnung: 2.OG Grundriss	
			Liegenschaft / Gebäude: Berliner Tor, Berliner Tor 11	
Bearbeiter: M. Döinghaus	Datum: 09.11.2009	Maßstab: 1:100	Bauherr/Entwurfsverfasser:	Format: DIN A3

Beispielanwendungen

- Cooking Agent
- Wecker 2.0
- Gegensprechanlage

Beispielszenarien: Concept-Map

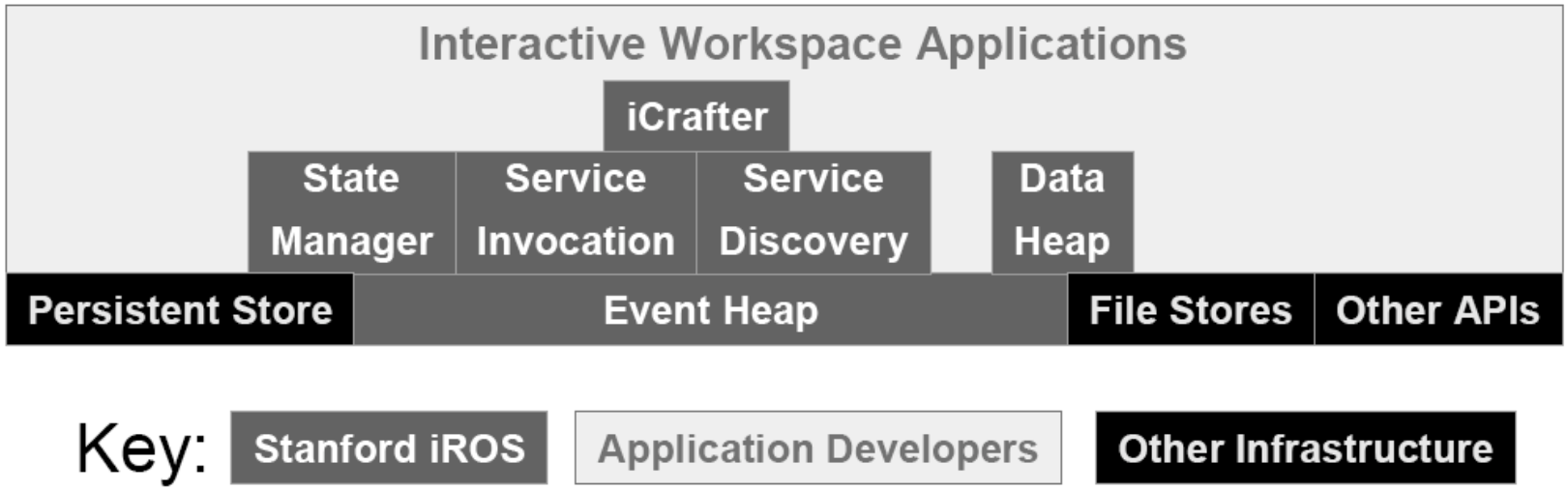


Bisherige Realisierung

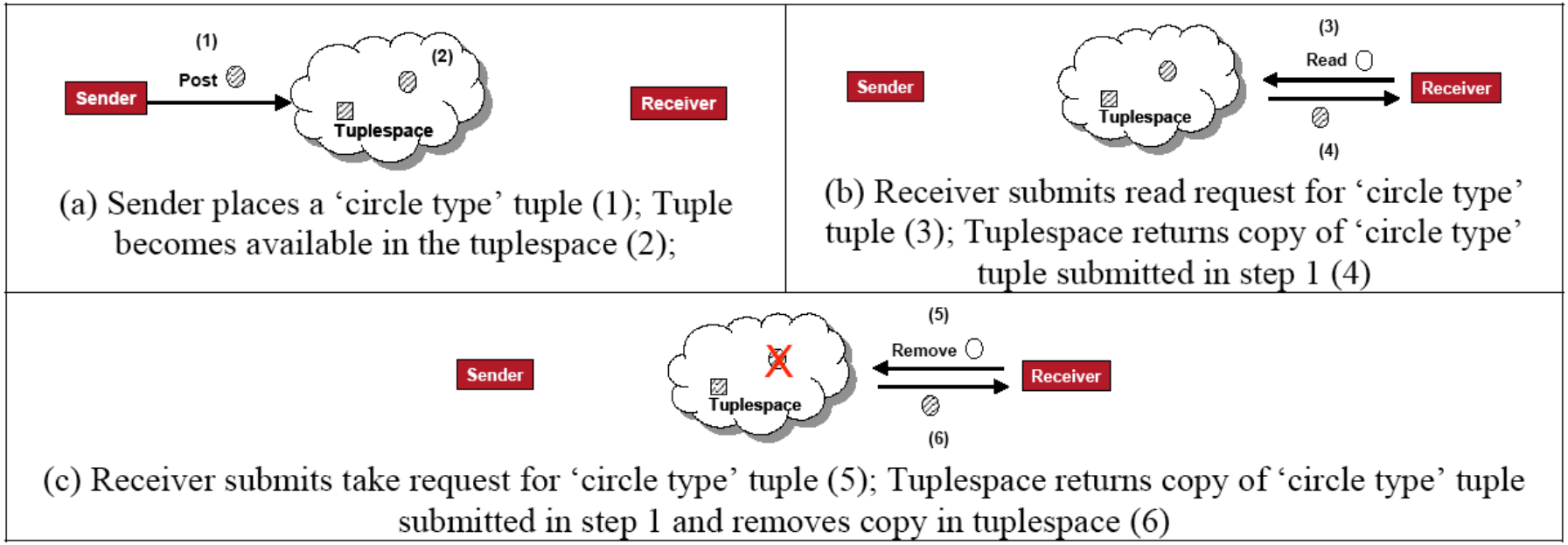
iROS – Event Heap

- Teil des Stanford „interactive Room Operating System“
- Intelligente Endgeräte
- Asynchrone Kommunikation über Events

iROS – Komponenten



Anlehnung an das Tuplespaces-Modell



Erkenntnisse aus der praktischen Realisierung

- Der Event Heap ist...
  - Einfach und
  - Schlank
- Aber auch
  - Unzuverlässig,
  - Langsam und
  - Starr

Weiterentwicklung eingestellt

- Letzte Veröffentlichungen: ca. 2004
- Letzter Stand der Entwicklung: ca. 2006

Andere Projekte?

- OxyGEN (MIT, Philips, Nokia, u.a.)
  - ca. 2005 beendet
- DynAMITE (Fraunhofer)
  - ca. 2006 beendet
- Conected Living e.V.
  - Ziel: Standardisierung von Heimautomation
  - DAI-Labor (TU-Berlin) & Industrie

Zielsetzung

- Aufbau einer flexiblen und leistungsfähigen Infrastruktur für den Laborbetrieb

Das Smart Home als verteiltes System

Herausforderungen verteilter Systeme

- Heterogenität
- Offenheit
- Sicherheit
- Skalierbarkeit
- Fehlerbehandlung
- Nebenläufigkeit
- Transparenz

Was ist wichtig?

- Transparente Modellierung der Nebenläufigkeit
- Ereignisse beobachten und erkennen können
- Verarbeiten von Ereignisinformationen

- Den Entwicklungsprozess unterstützen
- Erweiterbarkeit

Modellierung von Nebenläufigkeit

- Nebenläufigkeit mit Threads ist oftmals schwer verständlich
  - Synchronisationspunkte
  - Semaphoren, Mutexe
  - Etc.
- Gilt ebenso in verteilten Systemen
- Aber: Unsere Welt ist doch ebenfalls höchst nebenläufig!

Aktor-Modell

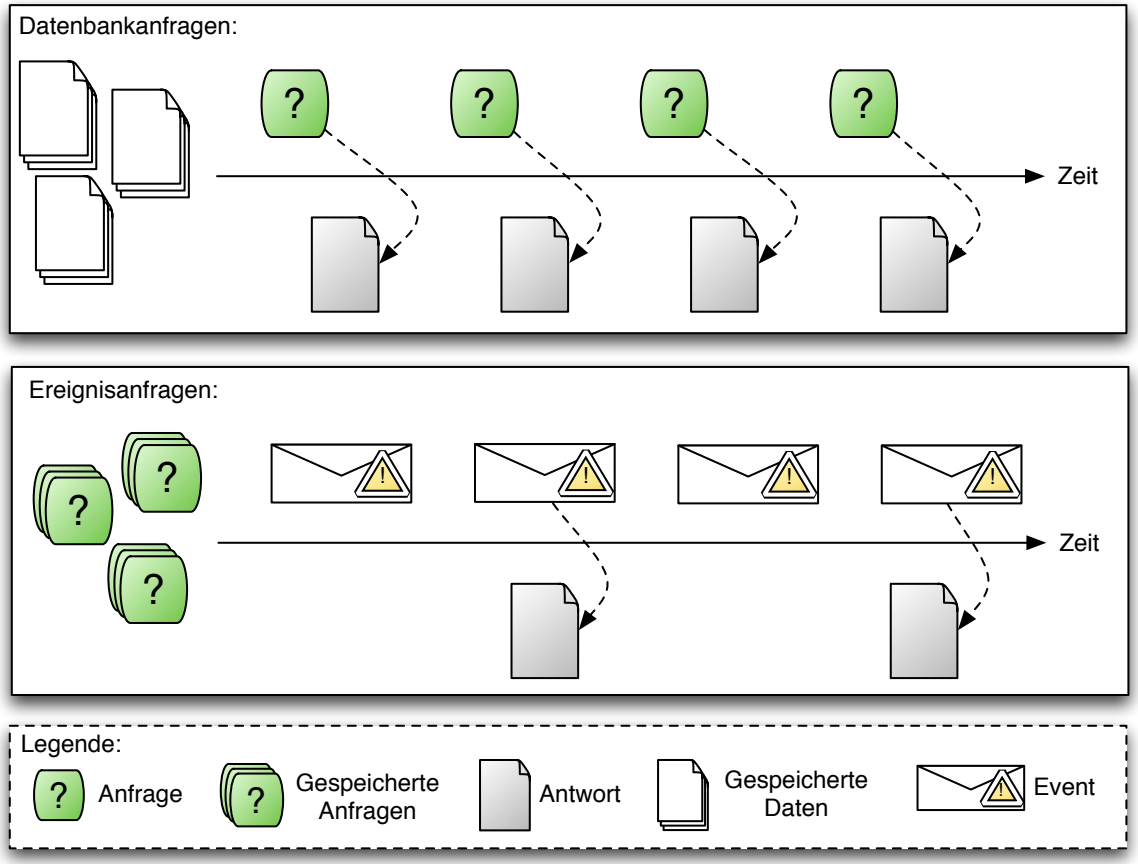
- Lösungsansatz durch verständliche Modellierung
- Aktoren sind vollständig getrennte Entitäten
- Kommunikation asynchron über Nachrichten
- Programmiersprachen: z. B. Erlang oder Scala

Ereignisverarbeitung

- Beobachten von Aktivitäten
- Abbildung durch Events
- Mustererkennung auf Event-Folgen

- Complex Event Processing

Complex Event Processing

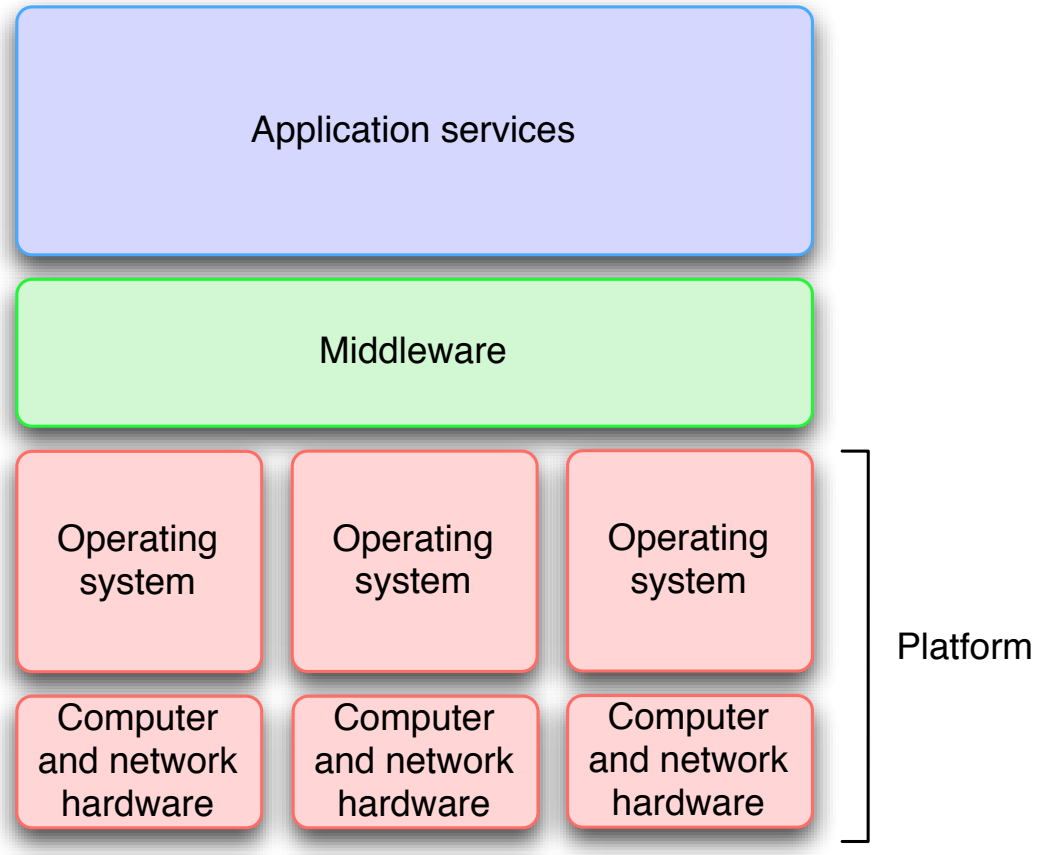


Heterogenität

- Betriebssysteme
- Datenrepräsentation
- Kommunikationsfähigkeiten

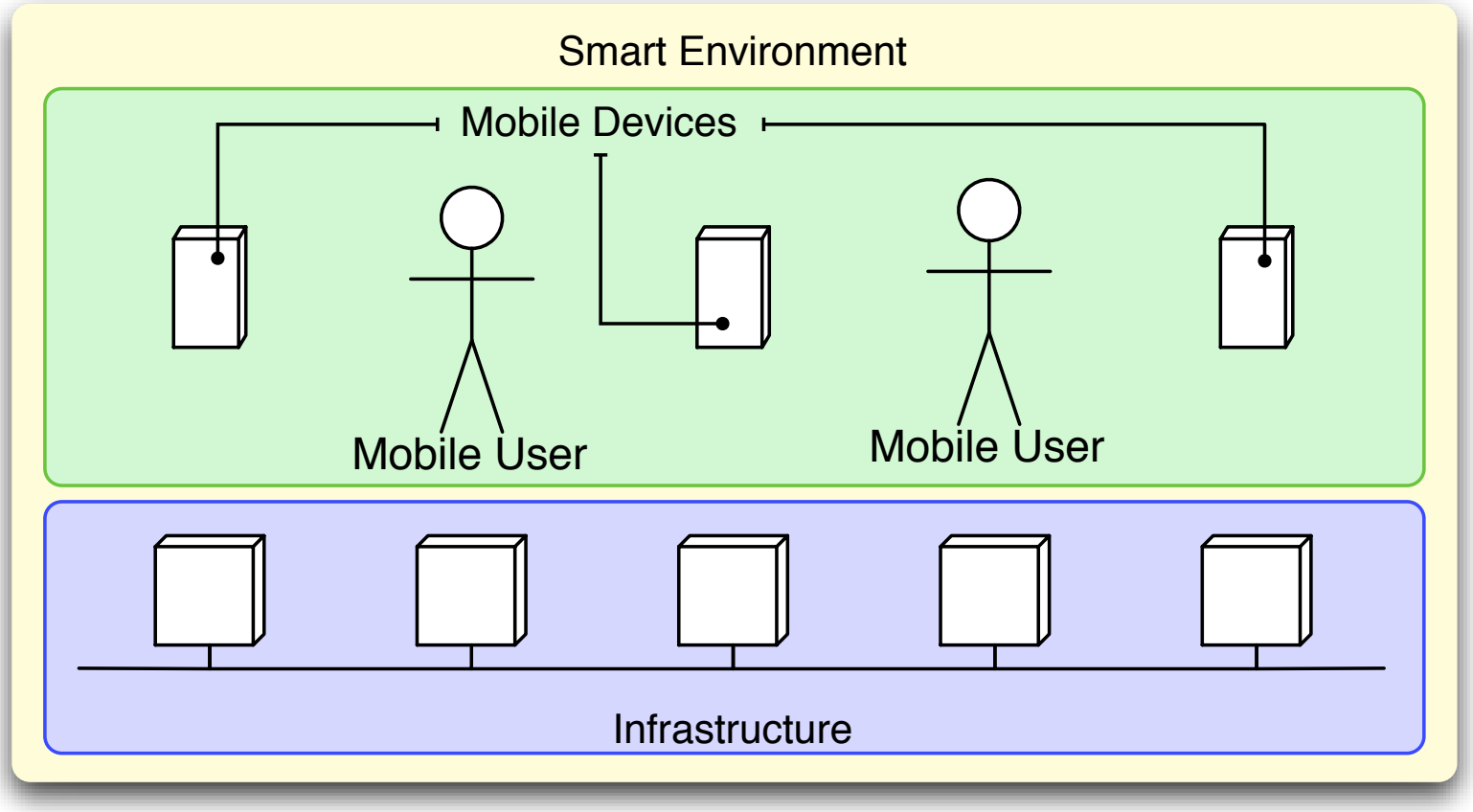
- Leistungsfähigkeit
- Verfügbare Ressourcen
- Mobilität

Middleware in verteilten Systemen

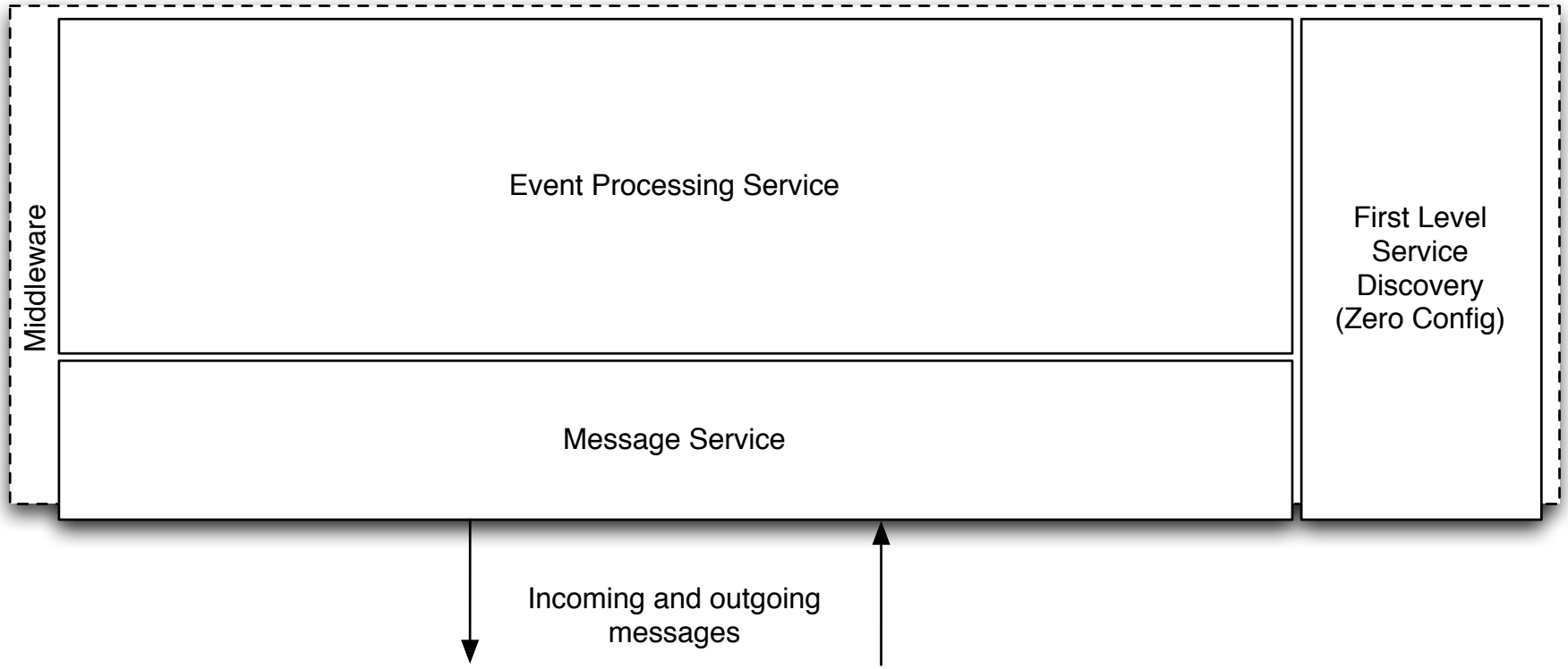


Architekturvorschlag

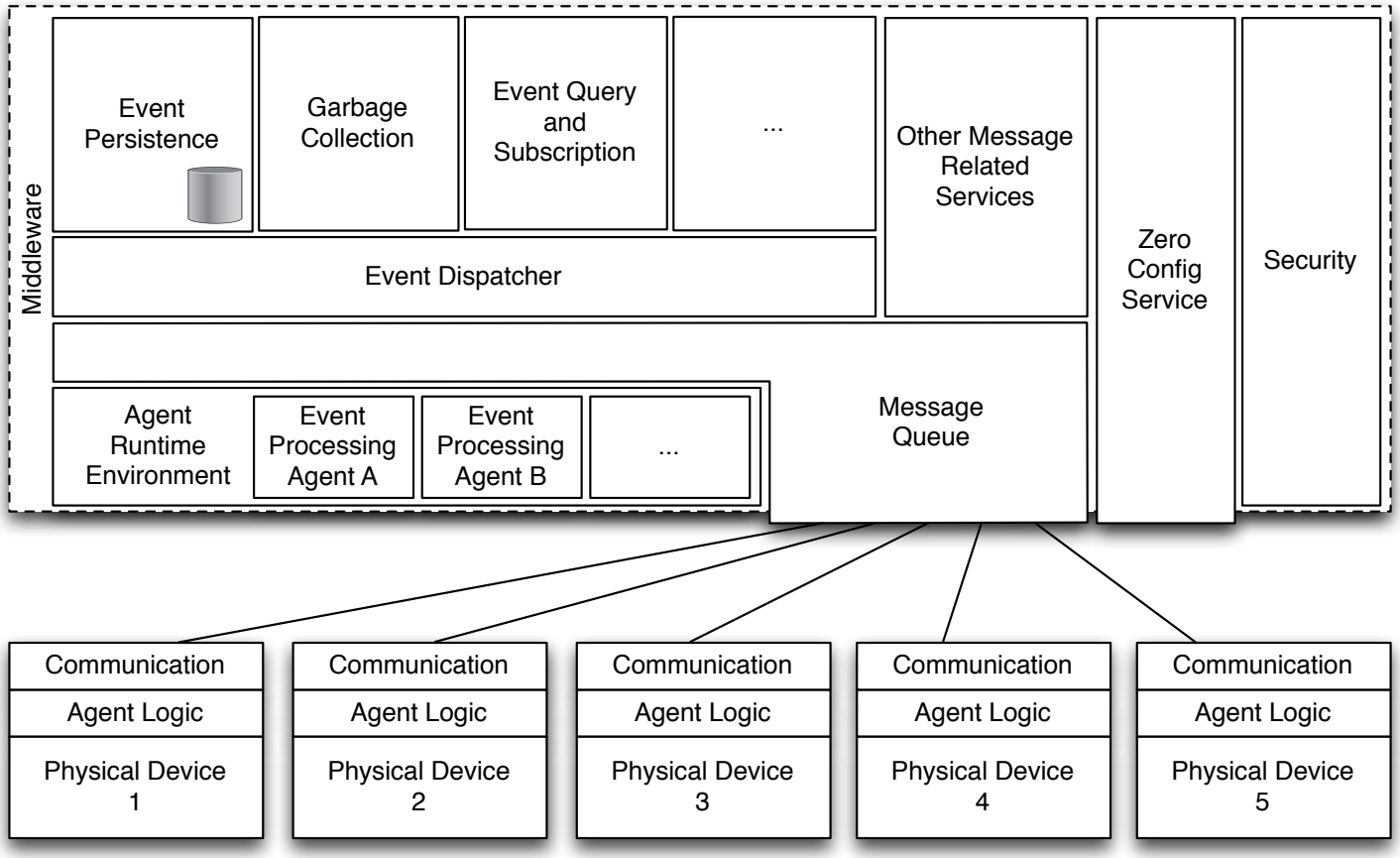
Systemorganisation



Komponentenbasiertes Design



Konkretisiertes Design



Architekturentscheidungen

- Zentrale Middleware
- Trennung von Kommunikation und Nachrichtenverarbeitung
- Intelligente Peers
- Systemverhalten emergent durch Agenten bestimmt
- Asynchrone Kommunikation

Vorteile des gewählten Modells

- Flexibel erweiterbar
- Alternative Kommunikationssysteme

- Entwicklung in getrennten Teams möglich
- Iterative Entwicklung bietet sich an

Implementation

Erste Schritte:

- Wahl eines Komponentenframeworks
- Definition von Schnittstellen
- Definition interner und externer Protokolle

Diskussion