

# Ontologieorientierter Datenbankentwurf für ein föderiertes Datenbanksystem im Projekt SensorCloud

Gregor Büchel, Henning Budde, Georg Hartung, Tobias Krawutschke, Alexander Stec,  
Lothar Thieling

Institut für Nachrichtentechnik  
Fachhochschule Köln  
Betzdorfer Str. 2  
50678 Köln  
gregor.buechel@fh-koeln.de

**Abstract:** Im Rahmen des Projekts SensorCloud wird ein föderiertes Datenbanksystem zur Verwaltung von Sensoren und Aktoren, dem Ansatz einer Ontologie für Sensornetzen folgend, entwickelt. Lokale Sensornetze werden über Gateways, die den Anforderungen einer „Trusted Cloud“ unterliegen, mit den Rechnerknoten der Cloud verbunden. Das föderierte Datenbanksystem enthält hierzu Segmente auf den Gateways als auch in der Cloud.

## 1 Einleitung

Allgemeines Ziel des Projekts SensorCloud<sup>1</sup> ist, Sensornetze, wie sie typischerweise in der Hausautomatisierung oder in der industriellen Anlagensteuerung eingesetzt werden und bisher fast ausschließlich als lokale Netze unter proprietären Konzepten betrieben werden, den Methoden und Architekturen des Cloud Computings zugänglich und damit global steuerbar zu machen. Der Aufbau der Cloud soll dem Konzept serviceorientierter Strukturen folgen, die folgende Anwendungsfälle berücksichtigt: a) Hausautomatisierung in Gebäuden zum Zweck der Energieeinsparung; b) Neue Methoden der Strommessung (z.B. in Privathaushalten der bürgerlichen Mitte oder in Windkraftanlagen); c) Steuerung und Sensorik für industrielle Anlagen, z. B. Portalkräne bzw. Druckstraßen.

Das Projekt SensorCloud unterliegt den Anforderungen einer „Trusted Cloud“, d.h. sie muss sowohl den Aspekten der Datensicherheit (z.B. Ausfallsicherheit bei fehlender Verbindung des lokalen Sensornetzes zur Cloud) und des Datenschutzes (z.B. Schutz vor unbefugten Zugriffen) als auch den Aspekten des Vertrauens genügen. Z.B. kann der Benutzer auswählen, welche Daten lokal bleiben bzw. welche in die Cloud kommen. Traditionell wird im industriellen Maßstab das Problem der Integration lokaler Datenbanken in ein globales Datenbanksystem durch den Einsatz eines verteilten Systems relationaler Datenbanken (RDB) gelöst. Erfahrungen der Betreiber großer

---

<sup>1</sup> Das Projekt SensorCloud wird im Rahmen des BMWi-Forschungsprogramms „Trusted Cloud“, das aufgrund eines Beschlusses des Deutschen Bundestages eingerichtet wurde, seit dem 1.2.2012 für einen Zeitraum von drei Jahren gefördert. Das Konsortium des Projekts wird gebildet von der RWTH Aachen, der QSC AG (Köln), der Firma Symmedia (Bielefeld) und der FH Köln.

Cloud Systeme (Amazon, Facebook, Google u.a.) zeigen aber, dass RDB Systeme in Hinsicht des Problems der horizontalen Skalierbarkeit großer und heterogen strukturierter Datenmengen nicht das Mittel der Wahl sind, und betreiben hierfür die Entwicklung von NoSQL Systemen (Wide Column Stores) [EFHB2010]. Weiterhin legen erste Untersuchungen zur Ontologie von Sensortypen die Verwendung von objektorientierten Persistenzmechanismen nahe.

Die Nachfrage nach Diensten der SensorCloud wird u. a. davon abhängen, wie es gelingt, Gebäude mit kostengünstigen Sensoren und lokalen Rechnern und einer an die Erfordernisse der SensorCloud angepassten Vorverarbeitung auszustatten, so dass die Sensordaten in der Cloud unabhängig von der Sensor-Implementation verfügbar werden. Der nun zunächst angestrebte Ansatz besteht darin, existierende Sensoren über einen preisgünstigen und dennoch leistungsfähigen Einplatinenrechner als Location Master in die SensorCloud zu integrieren, so dass an der Schnittstelle zur SensorCloud allgemeine und vertrauenswürdige Sensordienste unabhängig von deren Ausprägung angeboten werden. Der Location Master soll als festplattenloser Rechner auf Basis des Betriebssystem Linux betrieben werden und mit nichtflüchtigen RAM Speicherkomponenten ausgestattet sein. Im Rahmen des Projekts sollen Stromerfassungssensoren und bildverarbeitende Sensoren (VSfSC<sup>2</sup>) als integrierte Komponenten der Cloud adaptiert werden. Zur Speicherung der Sensordaten auf dem Location Master im Fall fehlender Netzverbindung sind leichtgewichtige DB-Managementsysteme mit ihrer inhärenten Transaktionssicherheit zu prüfen. Die Kombination mit obigen Betrachtungen, die OODB-Systeme und Wide Column Stores als CloudDB-Systeme einschließen, zielen auf den Aufbau eines geeigneten **Persistenzmechanismus** für die SensorCloud, das heterogene Datenbankstrukturen in Form eines verteilten Datenbanksystem integrieren soll. Ein wichtiger Ansatz hierbei ist die Nutzung des Konzepts **föderierter Datenbanksysteme** (FDBS).

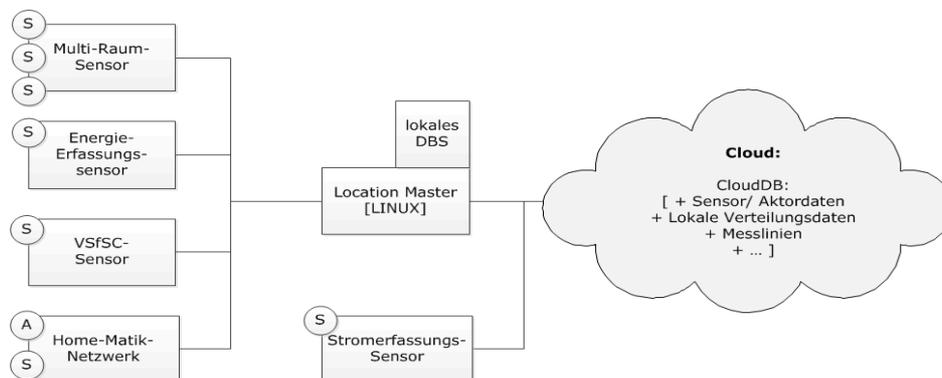


Abbildung 1: Kommunikation zwischen Cloud und lokalem Sensornetz

## 2 Das föderierte Datenbanksystem der SensorCloud – Ziele und Entwicklungsansätze

Aufgrund der allgemeinen Zielsetzung des Projekts und dem Stand der Technik verfolgt die Entwicklung des föderierten Datenbanksystems (FDBS) der SensorCloud folgende Ziele: (1) In der Konzeption föderierter Datenbanken überwiegen Ansätze der Homogenität der Datenmodelle der beteiligten Datenbanken (z.B. der Ansatz: Alle beteiligten Datenbanken sind relational). Aufgrund der Heterogenität der Sensor- und Aktorwelt sind neben relationalen auch objektorientierte Datenbanksysteme, aufgrund der Skalierbarkeitsanforderungen innerhalb einer Cloud sind auch NoSQL-Systeme zu berücksichtigen. Daher wird das Ziel Heterogenität der Datenmodelle beim Aufbau des föderierten Datenbanksystems verfolgt. (2) Aus der Integration von Sensoren verschiedener Hersteller ergibt sich die Notwendigkeit, deren Eigenschaften in einem vom Hersteller unabhängigen Format zu beschreiben. Hierzu bieten sich Ansätze an, die sich auf Ontologien für den Anwendungsbereich Sensoren/Aktoren beziehen. Dieser Ontologie-gestützte Ansatz ermöglicht nicht nur eine einheitliche Benennung gleicher Funktionalitäten unabhängig vom konkreten Sensor, sondern bietet die Basis, um SensorCloud-Dienste allgemein formulieren zu können. (3) Für die lokalen Datenbanken sind Lasttests erforderlich, die sowohl die Performanz als auch Sicherheitsaspekte untersuchen. Erste Arbeitsergebnisse bezüglich der Anforderungsanalyse an ein Entitätenmodell des FDBS liegen vor [BBSL13]. In Hinsicht auf die Lasttests der lokalen Datenbanken sind Kriterien für die Auswahl geeigneter DBMS untersucht worden [AL13].

## 3 Ontologieorientierter Entitätenentwurf

Aus der Integration von Sensoren verschiedener Hersteller ergibt sich die Notwendigkeit, deren Eigenschaften in einem vom Hersteller unabhängigen Format zu beschreiben. Hierzu bieten sich Ansätze an, die sich auf Ontologien für den Anwendungsbereich Sensoren/Aktoren beziehen.

Eine Ontologie kann in ihren Kernbestandteilen durch einen Tupel (C, R, HC, A0) beschrieben werden, wobei C die Konzeptmenge der Ontologie, R die Menge ihrer zweistelligen Relationen, HC die Konzepthierarchie der Ontologie und A0 eine Menge von Axiomen ist, die der Ontologie zu Grunde gelegt werden können. Für gewöhnlich werden C, R, HC, A0 mit Sprachmitteln einer Ontologiebeschreibungssprache, z.B. OWL modelliert. Das Projekt favorisiert folgenden Ansatz: Man kann in äquivalenter Weise C, R, HC und A0 mit Mitteln eines DB-Schemas beschreiben, wenn die Data Definition Language (DDL) vom Datenmodell des DBMS hinsichtlich der Sprachmittel komfortabel ausgestattet ist: Die Menge C und R entsprechen dann der Menge der im DB Schema beschriebenen Entitäts- und Beziehungstypen. Wenn die DDL Vererbungsmechanismen unterstützt, wird HC von den Entitätstypen gebildet, die untereinander in Vererbungsbeziehungen stehen. A0 kann mit den DDL-Sprachmitteln zur Formulierung von Integritätsbedingungen abgebildet werden, wenn die Prädikate der Axiome hinreichend einfach, d.h. in CHECK-Klausel-Syntax formulierbar sind. Im

Moment wird eine Erweiterung der von [BC08] vorgeschlagenen Ontologie für Sensornetze nach diesem Ansatz als konzeptionelles objektorientiertes CloudDB Schema aufgebaut.

Hierbei wird in Betrachtung der Komplexität der Landschaft der in einer SensorCloud aufzunehmenden Sensoren und Aktoren und der Vielschichtigkeit ihrer Vernetzung in einer SensorCloud vom Ansatz der Beschreibung einer Top-Level Ontologie ausgegangen, die zunächst eine Basiskonzeptmenge  $C'$  von notwendigen Entitätstypen enthält, die als persistente DB-Segmente<sup>3</sup> im DBMS der SensorCloud implementiert werden sollen. Später kann unter Benutzung der Vererbung und anderer Mittel der Schemaerweiterung diese Menge  $C'$  zu einer für die Projektanforderung vollständige Konzeptmenge  $C$  erweitert werden. In der Bestimmung der Basiskonzeptmenge wird von folgender Überlegung ausgegangen: In der SensorCloud sollen alle die Informationen persistent verwaltet werden, die dazu dienen

- Objekte  $X$  ( $X =$  Räume, Geräte, Stromnetze, ...)
- mit Sensoren  $S$  zu überwachen,
- um dabei ihre Zustände  $Z$  feststellen zu können, um
- sie mit Aktoren  $A$
- aufgrund von Regeln  $R$  steuern zu können.

Dieser Ausgangspunkt führt zum konzeptionellen Ansatz, sieben Entitätengruppen zu modellieren: (1) Sensoren, (2) Aktoren, (3) Benutzer, (4) Lokation, (5) Messwerte, (6) LokationMaster, (7) Event.

Die Entitätengruppen vereinen verschiedene semantisch zusammengehörende Entitätstypen zu einer Gruppe. Zum Beispiel sind in der Gruppe Sensoren die folgenden Entitätstypen vorgesehen:

Entitätstypen	Beschreibung
Sensor	Charakterisiert einen einzelnen Sensor (Verknüpfung mit Lokation und Nutzer)
Sensortyp	Charakterisiert einen Sensortyp und wird in SDL <sup>4</sup> definiert
Sensorprodukt	Charakterisiert ein bestimmtes Produkt eines Sensortyps
SensorproduktKonfiguration	Charakterisiert die Konfiguration eines Sensorproduktes
SensorKonfiguration	Charakterisiert die Konfiguration eines einzelnen Sensors

<sup>3</sup> Je nach DBMS der zu implementierenden FDBS-Komponente kann das Segment z.B. eine Tabelle, ein Extent einer persistenten Klasse oder eine Column Family sein.

<sup>4</sup> Sensor Definition Language. Zur ontologischen Beschreibung des Sensortyps gehört die grammatische Beschreibung des Messdatensatzes, den der Sensor als Instanz des Sensortyps meldet. Damit liefert der Sensortyp die Parsing-Vorschrift für das lokale Messwertempfangsprogramm, das auf dem LocationMaster arbeitet.

SensorKommunikation	Charakterisiert die Kommunikation mit einem Sensor (IP, RS232, LAN, WLAN, BT, etc)
SensorKomponente	Charakterisiert eine Komponente eines Sensors
SensorEvent	Beschreibt, wann welcher Sensor Alarm melden soll (z.B. Schwellenwert Über-/Unterschreitung)
SensorVerbund	mehrere Sensoren können zu einem Verbund zusammengefasst werden

Nach dem aktuellen Stand der Analyse sind in den sieben Entitätengruppen insgesamt 48 Entitätstypen enthalten.

Der Aufbau der Begriffshierarchie HC kann am Beispiel der Entitätengruppe Lokation verdeutlicht werden: Der Top-Level Entitätstyp Lokation kann je nach Anwendungsbereich der SensorCloud unterschiedliche Substrukturen haben. Für drei typische Anwendungsbereiche (Hausautomation, Anlagenüberwachung, Strommessung in Windmühlen) sind hier verschiedene Substrukturen der Entität Lokation skizziert:

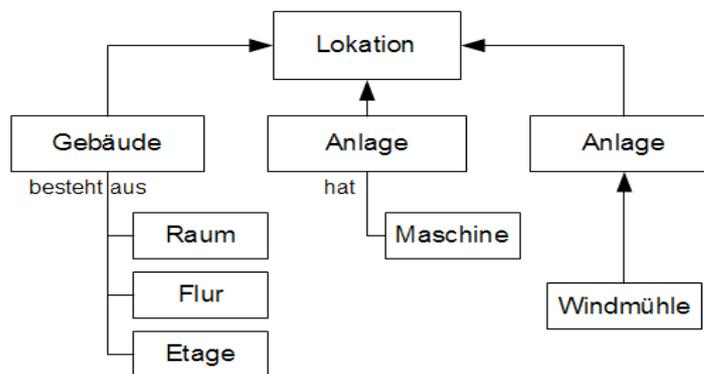


Abbildung 2: Begriffshierarchie des Entitätstyps Lokation

In der nächsten Spezifikationsphase wird die Relationenmenge  $R$  und die nach Analysevorgabe möglichst kleine Menge der Axiome  $A_0$  bestimmt.

#### 4 Implementationsansätze des FDBS der SensorCloud

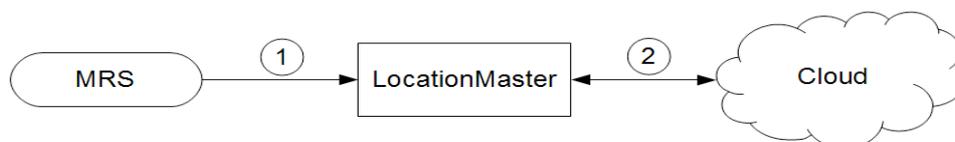
Für den Aufbau des FDBS wird im Moment von folgendem Planungsszenario ausgegangen: Die Entitätstypen der Entitätengruppe Benutzer (Benutzerstammdaten, Produkt- bzw. Servicelinien, Benutzergruppen, Mandanten u.a.m.) werden, an

Datenbeständen der industriellen Projektpartner anknüpfend, voraussichtlich in relationalen Segmenten der CloudDB verwaltet werden. Für die Entitätengruppen Sensoren, Aktoren, Messwerte und Event werden im Moment hinsichtlich der Flexibilität des Datenmodells, der DDL-Unterstützung und der Skalierbarkeit Wide Column Stores (z.B. Cassandra mit Cassandra Query Language) und ein OODBMS (db4o) als alternative DBMS, die für CloudDB-Segmente eingesetzt werden können, untersucht.

Für ihre Tauglichkeit zur Verwaltung leichtgewichtiger FDBS-Segmente auf Gateway-Rechnern vom Typ des LocationMasters werden RDBMS wie SQLite, MySQL und PostgreSQL und nichtrelationale DBMS wie db4o und Berkeley DB untersucht. Vom Mengenaufkommen und von der Attributvarietät werden diese Segmente deutlich kleiner als die entsprechenden Cloud-Segmente sein. Ihre vorrangige Aufgabenstellung lässt sich durch folgende Punkte beschreiben:

- Zwischenspeicherung der Messwerte, die von den Sensoren erhoben wurden, zur Sicherung gegen Verbindungsausfall des Internets;
- Speicherung der lokalen Sensor- und Aktorkonfiguration zur Unterstützung von Wartungsarbeiten und der Autokonfiguration;
- Unterstützung der Vorverarbeitung von Messwerten, wenn nur verdichtete Werte oder komprimierte Bilder von bildgebenden Sensoren in die Cloud übertragen werden;
- Speicherung von Sensor-, Aktor- und Eventwerten, wenn der Benutzer einen lokalen Betrieb seines Sensornetzes sicherstellen will, aber seine Daten nicht in die Cloud weiterreichen möchte.

Zur Transaktion der Sensordaten, von einem oder mehreren Location Masters zur SensorCloud, wird das Konzept der verteilten Transaktionen (Distributed Transaction Processing), nach der X/Open XA Spezifikation der Open Group, untersucht. Die XA Spezifikation erlaubt es eine globale Transaktion über mehrere Ressourcen (z.B. Datenbanken) unter Einhaltung der ACID-Eigenschaften (garantiert durch das Zwei-Phasen-Commit-Protokoll) durchzuführen. Der X/Open XA Standard ist für die objektorientierte Programmiersprache Java, in der Java Transaction API (JTA) implementiert.



typischer Protokollaufbau für MRS bei ① :

MRS-ID	SenTyp	Beweg.- sensor	Luft- feuchte	Temp.	Luft- qualität
		ja/nein [1 Hex]	8-Bit in % [2 Hex]	0° - 50°C [4 Hex]	CO <sub>2</sub> in % [2 Hex]

Abbildung 3: Teststrecke mit Multiraumsensor, LocationMaster und Cloud-Rechnerknoten

Im Rahmen eines Laboraufbaus wurde eine Teststrecke der SensorCloud entwickelt, die aus folgenden Komponenten besteht:

1. Mehrere Sensoren als Teile eines Sensornetzes (z.B. ein Muliraumsensor (=: MRS), der die Luftfeuchte, den CO<sub>2</sub>-Gehalt, die Temperatur und Bewegungen im Raum misst; ein Tür-Sensor; eine beschaltbare Steckdose) übertragen über eine Funkstrecke (in Abb. 3 mit „1“ gekennzeichnet) Messdaten an den LocationMaster des Netzes.
2. Der LocationMaster verfügt über ein lokales FDBS-Segment zur Speicherung von Sensor-, Aktor-, Mess- und Eventdaten.
3. Über eine Internet ist der LocationMaster mit Cloud-Knoten verbunden, die die Cloud-FDBS-Segmente verwalten.

In der CloudDB wird das globale Schema des FDBS verwaltet. Im Fall der Erweiterung oder Änderung bestimmter Attribute von Entitätstypen in den CloudDB-Segmenten ist es erforderlich, diese Erweiterungen bzw. Änderungen auf den Entitätstypen der betroffenen lokalen FDBS-Segmenten nachzuziehen. Hierzu ist ein entsprechender Schema Dienst, ein sogenannter Schema-Monitor zu implementieren. Für die vier Fälle  $CloudDBMS \in \{SQLite, db4o\}$  und lokales  $DBMS \in \{SQLite, db4o\}$  wurde ein prototypischer Schema-Monitor entwickelt, der wechselseitig den entsprechenden ALTER TABLE Code bzw. den Java Code der geänderten db4o Klasse generiert [Li11].

## 5 Stand der Technik

Innerhalb der Klasse verteilter Datenbanksysteme sind föderierte Datenbanksysteme als lose gekoppelte heterogene Datenbanksysteme gekennzeichnet ([Ritt2007]). In einem FDBS werden verschiedene lokale Datenbanken durch ein gemeinsames globales Schema verwaltet ([Ritt2007], [Voss2008]). Typischerweise werden für die Implementierung des FDBS relationale Datenbanken als lokale Datenbanken angenommen (z.B. [KeEi2006], S.451). Vossen sieht die Möglichkeit der Föderation lokaler Datenbanken von unterschiedlichen Datenmodellen. Z.B. können relationale und objektrelationale Datenbanken föderiert werden ([Voss2008], S.452). Er schlägt als gemeinsames Datenmodell für das im Prozess der Datenintegration aufzubauende globale Schema ein objektorientiertes Datenmodell vor (a.a.O. S.454).

Die Schemaarchitektur eines föderierten DBS besteht typischerweise aus folgenden Komponenten (vgl. [Voss2008], [KeEi2006], [Ritt2007], [ÖzVa1999]): Sie besteht aus den Schemata der lokalen Datenbanken (lokale Schemata) und dem globalen Schema, das gemäß dem logischen Datenmodell des FDBS aufgebaut ist. Hinsichtlich der Vermittlung zwischen dem globalen Schema und den lokalen Schemata gibt es unterschiedliche Strategien: Die Vermittlung wird entweder unter dem Aspekt der Fragmentierung und der Allokation betrieben, dann wird aus dem globalen Schema zunächst ein Fragmentierungsschema abgeleitet, aus dem über Allokationsschemata auf die lokalen Schemata zugegriffen werden kann ([KeEi2006], [Ritt2007]) oder die Vermittlung erfolgt unter dem Aspekt eines Multidatenbanksystems mit globalem Schema ([Voss2008]), indem in Form eines Bottom-Up Verfahrens die lokalen Schemata gemäß dem logischen Datenmodell des FDBS in Komponentenschemata

übersetzt werden, die in Hinsicht an die Serviceanforderungen des FDBS in so genannte Exportschemata überführt werden, deren Integration dann das globale Schema ergibt.

Beim beschriebenen Weg der Datenintegration, der von den lokalen Schemata zum Aufbau eines globalen Schemas führt, sind diverse Arten von semantischen Problemen festgestellt worden ([Voss2008], [Ritt2007]). Zur Lösung schlägt [Voss2008] den Einsatz von Ontologien zum Aufbau eines kontrollierten Vokabulars und zur Verwaltung von Synonymen, Homonymien vor (vgl. a. [Ritt2007]). Solche Integrationskonflikte und Lösungsansätze werden auch bei [Conr2002] diskutiert. [Rahm1994] schlägt als Verfahren ein semiautomatisches Schema-Matching vor.

Sollen semantische Analysen mittels des Wissensrepräsentationsmechanismus von Ontologien betrieben werden, um kontrollierte Vokabularien für die datenbankgestützte Verarbeitung von Sensordaten aufzustellen, kann an bereits entwickelte Sensor/Aktor - Ontologien angeknüpft werden (vgl. etwa [BCCo2008], [LCYB2010]). Weiterhin sind umfangreiche Untersuchungen zu Ontologien als Mittel der Wissensrepräsentation, der Definition kontrollierter Vokabularien mit OWL und ihrer Anwendung zum Aufbau von wissensbasierten Systemen (z.B. Knowledge Management Systemen), die der W3C Architektur für das Semantic Web entsprechen, zu berücksichtigen (vgl. [StSt2004], [HCLO2009]).

In Hinsicht auf sensornahe DBMS sind unterschiedliche Aspekte zu betrachten. Zum ersten ist es die Integration von persistenten Sensordaten in Enterprise Services oder allgemein gesprochen in ein „Internet der Dinge“ [Kam2009]. Im Weiteren geht es um geeignete Datenbanktechnologien für bestimmte Klassen von Sensornetzwerken, wie z. B. um Funknetzwerke. Hierfür schlagen Klan und Sattler [KISa2011] eine prototypische Architektur eines Datenstrom Managementsystems auf Basis des DBMS TinyDB mit einer an SQL angelegten Anfragesprache CQL (Continuous Query Language), die die Gruppierung von Daten in einer SELECT Anfrage nach Zeittakten zulässt. In [BSAA2009] wird eine Fokussierung der Speicherstrategie für große Mengen von Sensordaten auf eine Kombination von Data Warehouse Systemen auf Basis von RDB Systemen mit einem System von Flatfiles vorgeschlagen. Bei diesem Ansatz wird von einem nicht verteilten DBMS ausgegangen. Das Kriterium der horizontalen Skalierbarkeit, wie es für Bigtable Konzepte essentiell ist (vgl. [CDGH2006]), wird dort nicht diskutiert.

Sollen Sensordaten auf eingebetteten Systemen vorverarbeitet werden, stellt sich die Frage nach DBMS, die in Hinsicht auf die beschränkten Betriebsmittel eines eingebetteten Systems eine hinreichende Performanz bieten. In [PeRi2009] werden die DBMS Java DB/Derby, SQLite, Berkeley DB und db4o untersucht. Diese vier DBMS sind in ihrem Datenmodell verschieden (z.B. Java DB/Derby ist ein RDBMS, db4o ein OODBMS) und unterstützen unterschiedliche Anfragesprachen, kommen aber alle mit den begrenzten Betriebsmitteln eines eingebetteten Systems aus (z. B. liegt der Speicherverbrauch zwischen 500 KB bis 4 MB) und sind transaktionssicher.

Von großen Anbietern von Cloud Services sind neue Typen von Datenbanksystemen beruhend auf dem Bigtable bzw. Wide Column Store Konzept entwickelt worden, die

vom Row Store Konzept, das dem RDB Modell zu Grunde liegt, abweicht ([Feke2011]). Beispiele solcher Datenbanksysteme sind HBase (Apache Hadoop), SimpleDB (Amazon) und Cassandra (Facebook). Solche Datenbanksysteme bieten Vorteile der horizontalen Skalierbarkeit großer, heterogen strukturierter Datenmengen und der einfachen Erweiterbarkeit der Column Families um Attribute während der Laufzeit ([EFBH2010]). Gleichzeitig bestehen mehrere grundsätzliche Problemstellungen („Big Questions“) in Hinsicht auf diese Cloud Datenbanksysteme, auf die Kossmann und Kraska hinweisen: Hierzu gehören allgemein ungelöste Problemstellungen, wie „How to Partition the Data?“, „Consistency vs. Availability?“ und die Frage nach der Garantie der Konsistenz durch geeignete Kontrollstrukturen ([KoKr2010], S.126ff.). Beiträge zur praktischen Lösung dieser Fragen in bestimmten Anwendungsbereichen sind ein Desiderat der Forschung im Bereich von Datenbanken und Cloud Services.

Klassische Ansätze zur Integration von Hausautomatisierungsnetzwerken (HaNw) in das Internet stellen i.d.R. Funktionalitäten des Netzwerks über ein Gateway als Internet-Dienste zur Verfügung (z.B. Araujo, Gill, Zamora-Izquierdo). Dabei ist der Ausgangspunkt zumeist ein Netzwerkstandard für Hausautomatisierung, wie ZigBee, ZWave, KNX. Sie sind im Rahmen von SensorCloud als Lösung ungeeignet, da die Sensor/Aktorbeschreibungen an die Formate der Hersteller gekoppelt sind. In den Arbeiten der Turiner Gruppe um Bonino et al. zum Thema „Domotic Ontology driven Gateway“ [BC08] wurde ein OWL basiertes System entwickelt, mit dem Komponenten verschiedener HaNw aus dem Internet abgefragt/ferngesteuert werden können.

## Literaturverzeichnis

- [AFMN2004] Araujo, A.; Fraga, D.; Moya, J.M; Nieto-Taladriz, O.: Domotic platform based on multipurpose wireless technology with distributed processing capabilities. In: 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Bd. 2004, S. 3003–3007.
- [AL13] Andreas Lockermann: "Deploymentanalyse", Januar 2013. ([http://vma.web.fh-koeln.de/index.php?option=com\\_content&view=article&id=111&Itemid=108](http://vma.web.fh-koeln.de/index.php?option=com_content&view=article&id=111&Itemid=108))
- [BaCo2001] Balsters, H., Conrad, St.: „Database Schema Evolution and Meta-Modeling“, 9th International Workshop on Foundations of Models and Languages for Data and Objects, Lecture Notes in Computer Science 2065, Berlin, Heidelberg (Springer) 2001.
- [BaGü2008] Bauer, A.; Günzel, H.: „[Data Warehouse Systeme - Architektur, Entwicklung, Anwendung](#)“, Heidelberg (dpunkt) 2008.
- [BBSL13] G. Büchel, H. Budde, A. Stec, A. Lockermann: "Anforderungsanalyse für das föderierte Datenbanksystem der SensorCloud", Januar 2013. ([http://vma.web.fh-koeln.de/index.php?option=com\\_content&view=article&id=111&Itemid=108](http://vma.web.fh-koeln.de/index.php?option=com_content&view=article&id=111&Itemid=108))
- [BCCo2008] Dario Bonino, D.; Emiliano Castellina, E.; Fulvio Corno, F.: “The DOG Gateway: Enabling Ontology-based Intelligent Domotic Environments“, in: IEEE Transactions on Consumer Electronics, Vol. 54, No. 4, NOVEMBER 2008, S. 1656–1664.
- [BSAA2009] Beenken, P.; Schwassmann, St.; Albrecht, M.; Appelrath, H.-J.; Heisecke, S.: „Speicherstrategien für die Sensordaten des Offshore-Windparks alpha ventus“, in: Datenbank-Spektrum 28/2009, S.22-30.

- [CDGH2006] Chang, F.; Dean, J.; Ghemawat, S.; Hsieh, W.; Wallach, D.; Burrows, M.; Chandra, T.; Fikes, A.; Gruber R.: „Bigtable: A Distributed Storage System for Structured Data“ OSDI 2006. <http://labs.google.com/papers/bigtable.html> .
- [Conr1997] Conrad, St.: „Föderierte Datenbanksysteme, Konzepte der Datenintegration“, Berlin, Heidelberg (Springer) 1997.
- [Conr2002] Conrad, St.: „Schemaintegration – Integrationskonflikte, Lösungsansätze, aktuelle Herausforderungen“, in: Informatik – Forschung und Entwicklung, 17 / 2002, S.101-111.
- [EFBH2010] Edlich, St.; Friedland, A.; Hampe, J.; Brauer, B.: „NoSQL – Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken“, München (Hanser) 2010.
- [Feke2011] Fekete, D.: „Optimierungstechniken in Column Stores“, in: Datenbank-Spektrum April 2011, S.43-46.
- [HCLO2009] Heese, R.; Coskun, G.; Luczak-Rösch, M.; Oldakowski, R. u.a.: „Corporate Semantic Web – Semantische Technologien in Unternehmen“, in: Datenbank-Spektrum Oktober 2010, S.73-79.
- [Karn2009] Karnouskos, St.: „Efficient Sensor Data Inclusion in Enterprise Services“, in: Datenbank-Spektrum 28/2009, S.5-10.
- [KISa2011] Klan, D.; Sattler, K.-U.: „AnduIN: Anwendungsentwicklung für drahtlose Sensornetzwerke“ in: Datenbank-Spektrum April 2011, S.15-26.
- [KeEi2006] Kemper, A.; Eickler, A.: „Datenbanksysteme“, (6. erw. Aufl.) München, Wien (Oldenbourg) 2006.
- [KoKr2010] Kossmann, D.; Kraska, T.: „Data Management in the Cloud: Promises, State-of-the-art, and Open Questions“, in: Datenbank-Spektrum Dezember 2010, S.121-129.
- [KSYX2009] Khusvinder Gill, Shuang-Hua Yang Fang Yao and Xin Lu: A ZigBee-Based Home Automation System. In: IEEE Transactions on Consumer Electronics, Vol. 55, No. 2, MAY 2009, S. 422-430
- [LCYB2010] Li, Z.; Chu, Ch.; Yao, W.; Behr, R.: “Ontology-Driven Event Detection and Indexing in Smart Spaces”, in: IEEE International Conference on Semantic Computing 978-0-7695-4154-9/10 2010.
- [Li11] Libi, St.: Fallstudie zum Aufbau eines föderierten Datenbanksystems mit objektorientiertem Schema im Projektumfeld SensorCloud. Master-Thesis FH Köln, Köln, 2011.
- [ÖzVa1999] Özsu, M.T., Valduriez, P.: “Principles of Distributed Database Systems”, Englewood Cliffs, NJ, USA (Prentice Hall) 1999.
- [PeRi2009] Petersen, M.; Ristig, Chr.: „Kleine DBMS für Embedded Systems und Java DB (Derby), Moderne Datenbanktechnologien – Database as a Service“, Seminarvortrag TU Clausthal im WS08/09 am 24. Januar 2009.
- [Rahm1994] Rahm, E.: „Mehrrechner-Datenbanksysteme – Grundlagen der verteilten und parallelen Datenbankverarbeitung“, Bonn, Paris, Reading MA, USA (Addison-Wesley), 1994.
- [Ritt2007] Ritter, N.: „Verteilte und föderierte Datenbanksysteme“, in: Kudraß, Th. (Hg.): „Taschenbuch Datenbanken“, Leipzig (Fachbuchverlag) 2007. S.394-426.
- [StSt2004] Staab, St.; Studer, R. (Hg.): „Handbook on Ontologies“, Berlin, Heidelberg, New York (Springer) 2004.
- [Voss2008] Vossen, G.: “Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme”, (5. erw. Aufl.) München, Wien (Oldenbourg) 2008.
- [ZSGS2010] Miguel A. Zamora-Izquierdo, José SantaAntonio F. Gómez-Skarmeta: An Integral and Networked Home Automation Solution for Indoor Ambient Intelligence. A new home automation system improves on existing technologies by considering user requirements and addressing integration and deployment issues. In: Pervasive Computing, IEEE Journal, October-December, 2010, S. 66–77.