

NoSQL mit XML

Einsatzmöglichkeiten und Vorteile eines dokumentorientierten Repositories

MarkupForum 2012

Jochen Jörg, Senior Consultant, MarkLogic

Agenda

Ein Blick zurück ... Datenbanken und Software-Architektur

NoSQL - Eine Übersicht

Dokumentorientierte Datenbanken

Vorteile einer flexiblen Datenschicht

Ein Blick zurück ...

Datenbanken und Software-Architektur

80er, 90er Jahre ...

Beginn der Ära der relationalen DBMS

Hoch strukturierte Daten

Normalisierung, Vermeidung von Redundanzen

Einfache Applikationen, statische View

Ende 90er – Anfang 2000er

Zunahme der Komplexität der Anforderungen

Funktionale und nicht funktionale Anforderungen

Dynamische Websites, CMS

Erhöhte Last, vertikale Skalierung

Hoher Kopplungsgrad, schwer wartbar

„Wildwuchs“



2005 – 2009

Entkopplung, aber mehr Schichten



Mehr Disziplin u. Qualität in der Software-Architektur

Applikationsserver, Frameworks

ORM, SOA, ESB

Entkopplung der Komponenten (gut!!) ... aber Änderungen, Anpassungen recht aufwendig

2009 - heute

Neue Anforderungen an die Datenschicht

Datenmengen wachsen exponentiell

Daten werden konsumiert und produziert
(Web 2.0)

Mehr semi- und unstrukturierte Daten

2009 - heute

Neue Anforderungen an die Datenschicht

Multiple Datenquellen

Heterogene Daten

Schema- bzw. strukturelle Änderungen

2009 - heute

Neue Anforderungen an die Datenschicht

Neue Kanäle für die Bereitstellung

Web, Smartphones, Apps

Erhöhte Anforderung an Verfügbarkeit und
Performance

Schnelle Umsetzung neuer Anforderungen

2009 – heute

Traditionelle Ansatz der Datenhaltung stößt an seine Grenzen

Klassische Schichtenmodell ist zu statisch

... und das ist erst der Anfang!

Gesucht: neue Ansätze für die Datenschicht

Flexibilität

Skalierbarkeit (scale out, not up)

Strenge Konsistenz ist nicht mehr das Maß aller Dinge

neuer Ansatz: NoSQL

NoSQL - was steckt dahinter

Not only SQL

Nicht relational

Keine oder schwache Schemarestriktionen

Horizontale Skalierbarkeit

NoSQL - Kategorien

Key/Value Stores

Wide Column Stores

Graph Databases

Document Stores

NoSQL – Key/Value Stores

Einfaches Schema: Schlüssel und Werte

Werte: auch Set, Liste

Pro: Schnelle, effiziente Datenverwaltung

Contra: Keine ad-hoc Queries, abhängig von API

Beispiel: Redis, Voldemort, Dynamo, ...

NoSQL- Wide Column Stores

Hybrider Ansatz:

Strukturiertes, aber dennoch flexibles Datenbankmodell

Mehrfach ineinander geschachtelte Schlüssel-Wert Paare

Hochverfügbarkeit durch Replikation, Sharding

Client-Bibliotheken: Java, Python, C#, PHP

Pro: flexibel, schnell, hochverfügbar

Contra: keine ACID Transaktionen

Cassandra, HBase, SimpleDB

NoSQL- Graphen-Datenbanken

Knoten und Kanten (Beziehungen zwischen Knoten)

Eigenschaften (Key Values) an Knoten und Kanten

Verwandtschaft zu Triple Stores

Soziale Netzwerke (“Wer kennt wen”)

Routenberechnung, Flugplanoptimierung

Neo4j, AllegroGraph, Sones

Siehe auch [NoSQL], Kapitel 6

NoSQL- dokumentorientierte Datenbanken

Das Dokument im Mittelpunkt

Schema frei- Verlagerung der Schemaverantwortung in
die Applikations-Schicht

Vorteilhaft:

Agile Web 2.0 Anwendungen

Inhaltsorientierte Anwendungen

Document Stores vs. RDBMS

Document Store	RDBMS
Dokument	Tabellenzeile, Datensatz
Collection, Gruppierung von Dokumenten	Tabelle
XML Element, XML Attribute, JSON Element	Zelle, Attribut eines Datensatzes
XQuery, XSLT, JavaScript, MapReduce	SQL, PLSQL

CouchDB

Schema-frei
Dokumentorientiert
Replication
Noch kein Sharding
Fehlertolerant



Zugriffskontrolle: MVCC
Dokument ID, Revision ID
Zahlreiche Client Libraries
Dokumentanhänge

Speichert JSON Datenstrukturen
RESTful JSON API (HTTP)
Gefilterte Abfragen: MapReduce mit JavaScript,
Keine Ad-Hoc Abfragen
DB für web servers und Mobile Devices

MongoDB

Schema-frei
Dokumentorientiert
Replication
Sharding



Zahlreiche Client Libraries
Dokumentanhänge
BSON ($\leq 4\text{MB}$)
GridFS (Splitten großer Dateien)

Speichert JSON Datenstrukturen
Internes Format: BSON
Dynamische Abfrage
MapReduce für umfangreichere Abfragen
Index auf Dokument-Schlüssel
Keine dokumentübergreifende Transaktionen

MarkLogic Server

Datenbank, Search-Engine, AppServer
aus einem Guss

Schema-frei
XML als internes Datenmodell
auch Text, Binärdaten, JSON
Ad hoc queries
Skaliert horizontal (~PB)

 **MarkLogic**



ACID-Transaktionen
Replication,
Backups,
Disaster Recovery,
Hochverfügbar,
Hadoop-Connector

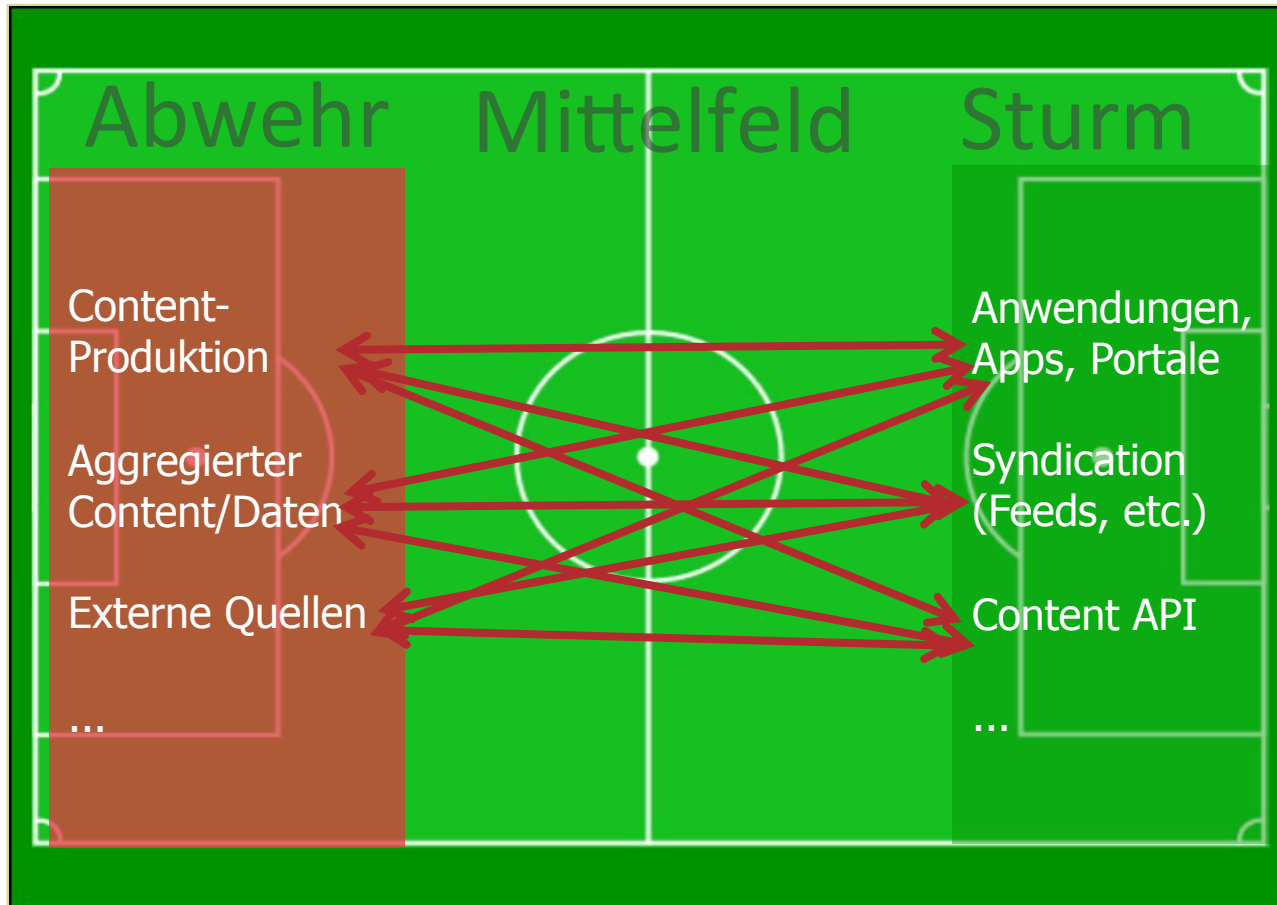
Universal Index == Invertierter Index
Volltext (stemming, wildcards, near, boolean ...)
Dokument-Struktur
Element-, Attributwerte
Geospatial
Co-occurrence, Alerting
Range-Index -> Fassetierte Suche
Real-Time

XQuery, XSLT
REST-API
HTTP, XDBC

Vorteile einer flexiblen Datenschicht als globales Repository

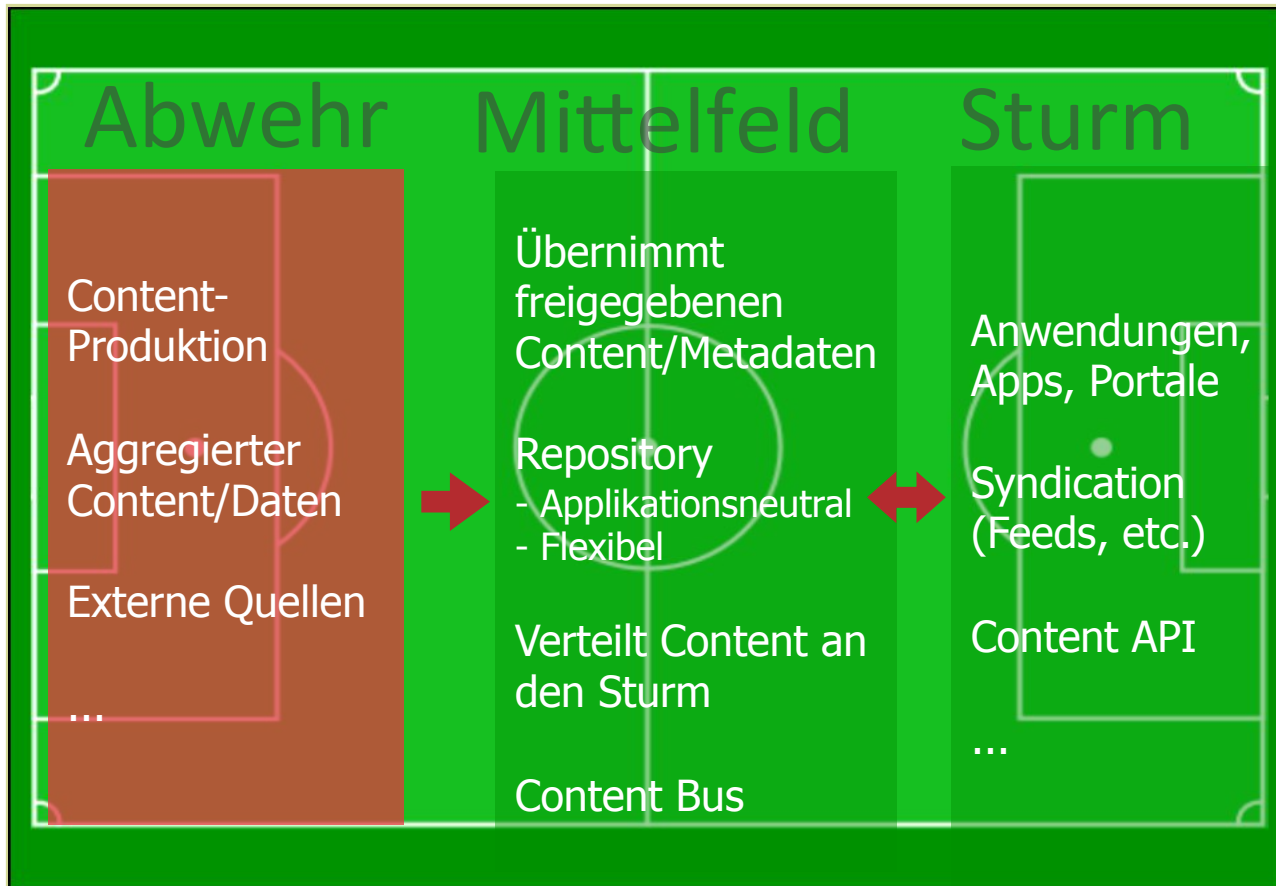
Beispiel: Medien-Unternehmen

Daten- und Information-Silos → erschwerte Bereitstellung von Information



Beispiel: Medien-Unternehmen

Konsolidierung und Entkopplung von Content Produktion und Bereitstellung durch eine flexible Datenschicht als “Mittelfeld”

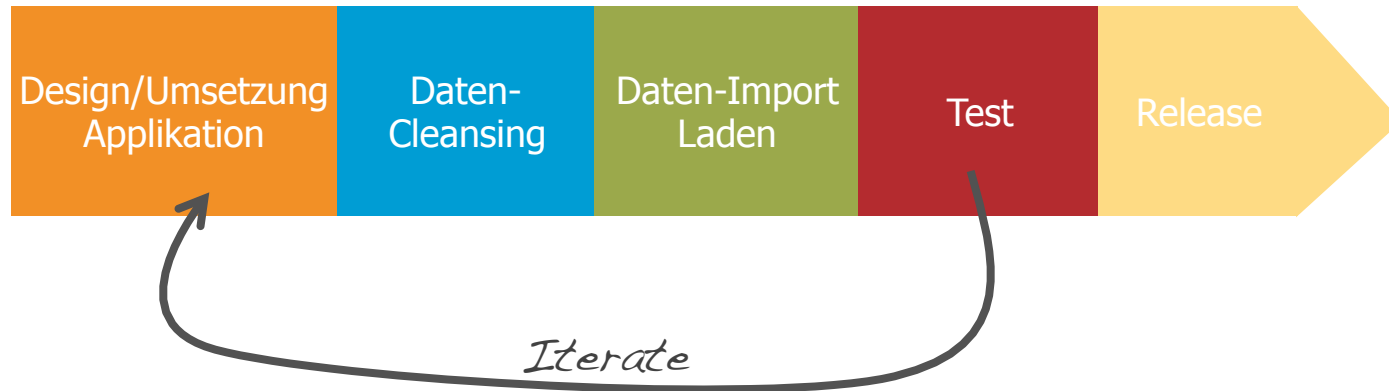


Entwicklungsprozess

Traditioneller Ansatz (RDBMS)

Daten-Cleansing außerhalb der Datenbank
Daten werden an Applikationslogik ausgerichtet (shredding)
Spätes Erkennen von Fehlern
Lange, aufwendige Iterationen

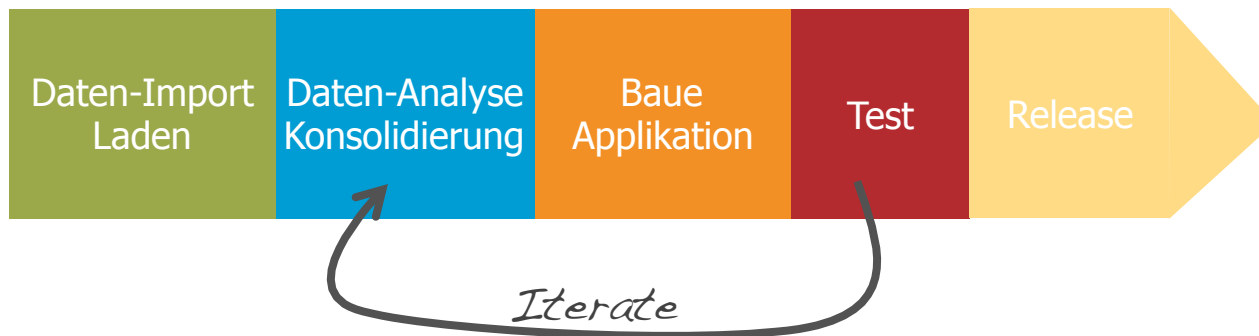
Start



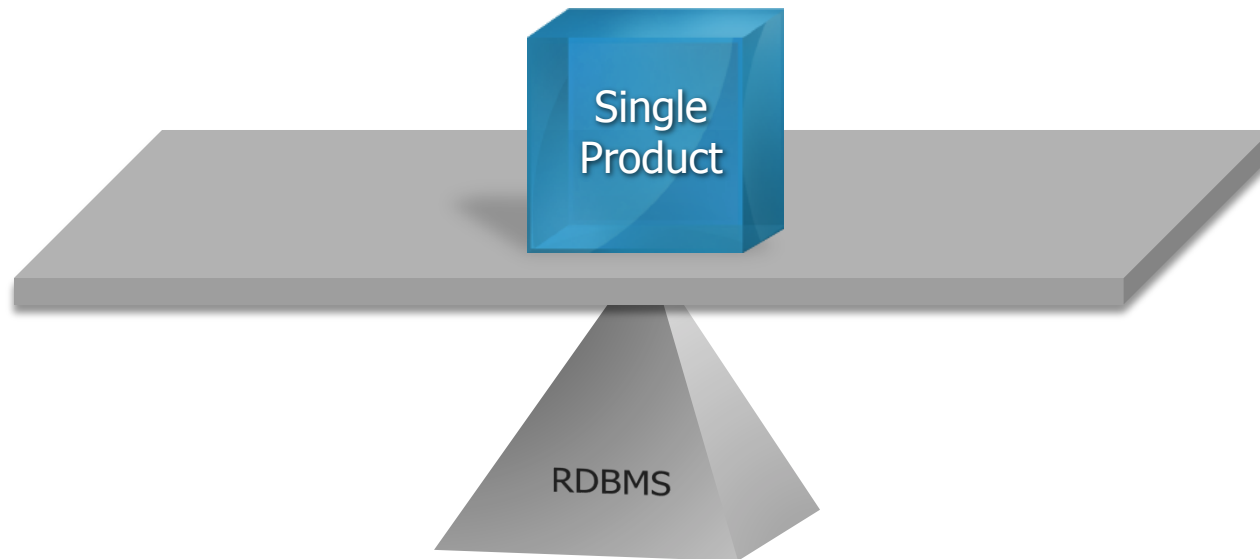
Entwicklungsprozess mit dokumentorientierter DB

Laden der Daten (“accept as is”)
Analyse der Daten
Erstellen der Applikation
Kurze Iterations-Zyklen

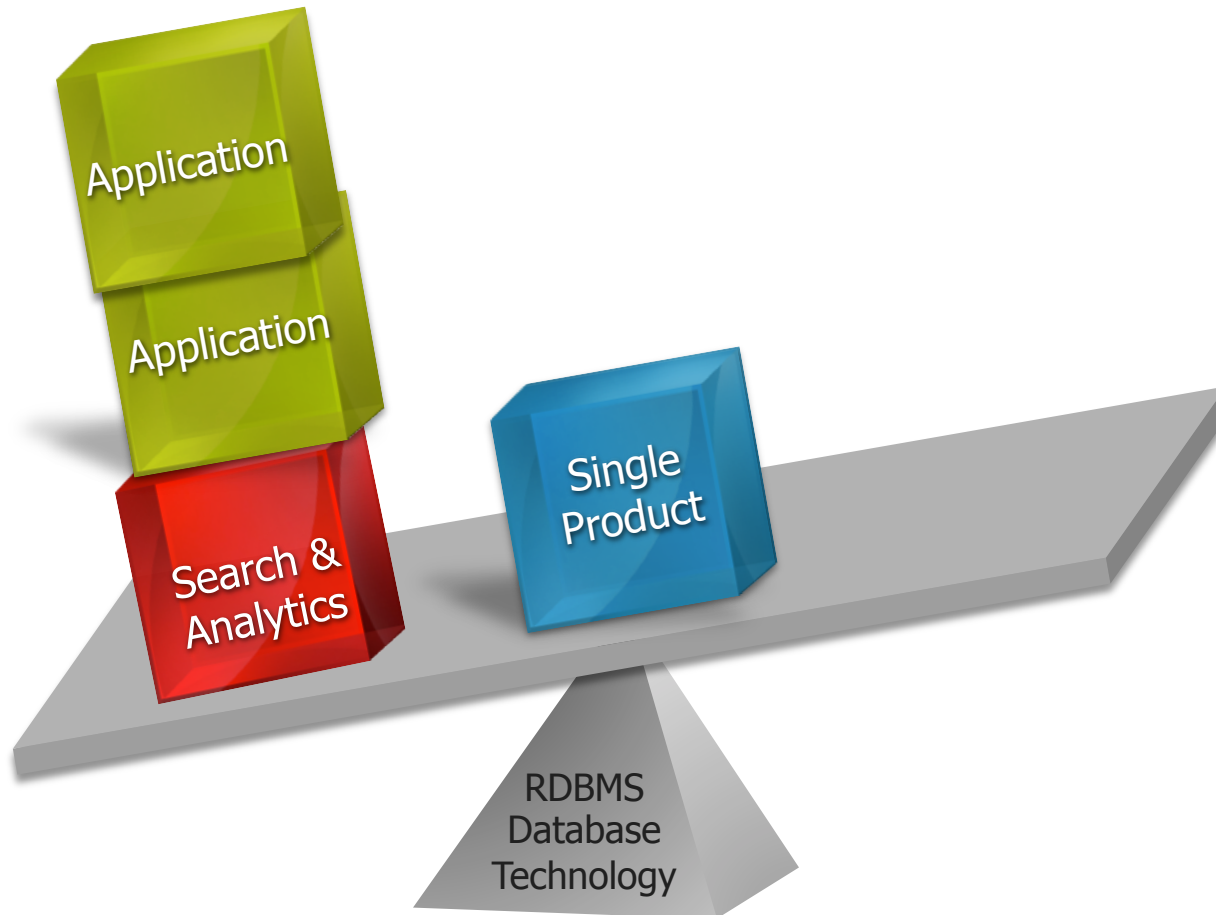
Start

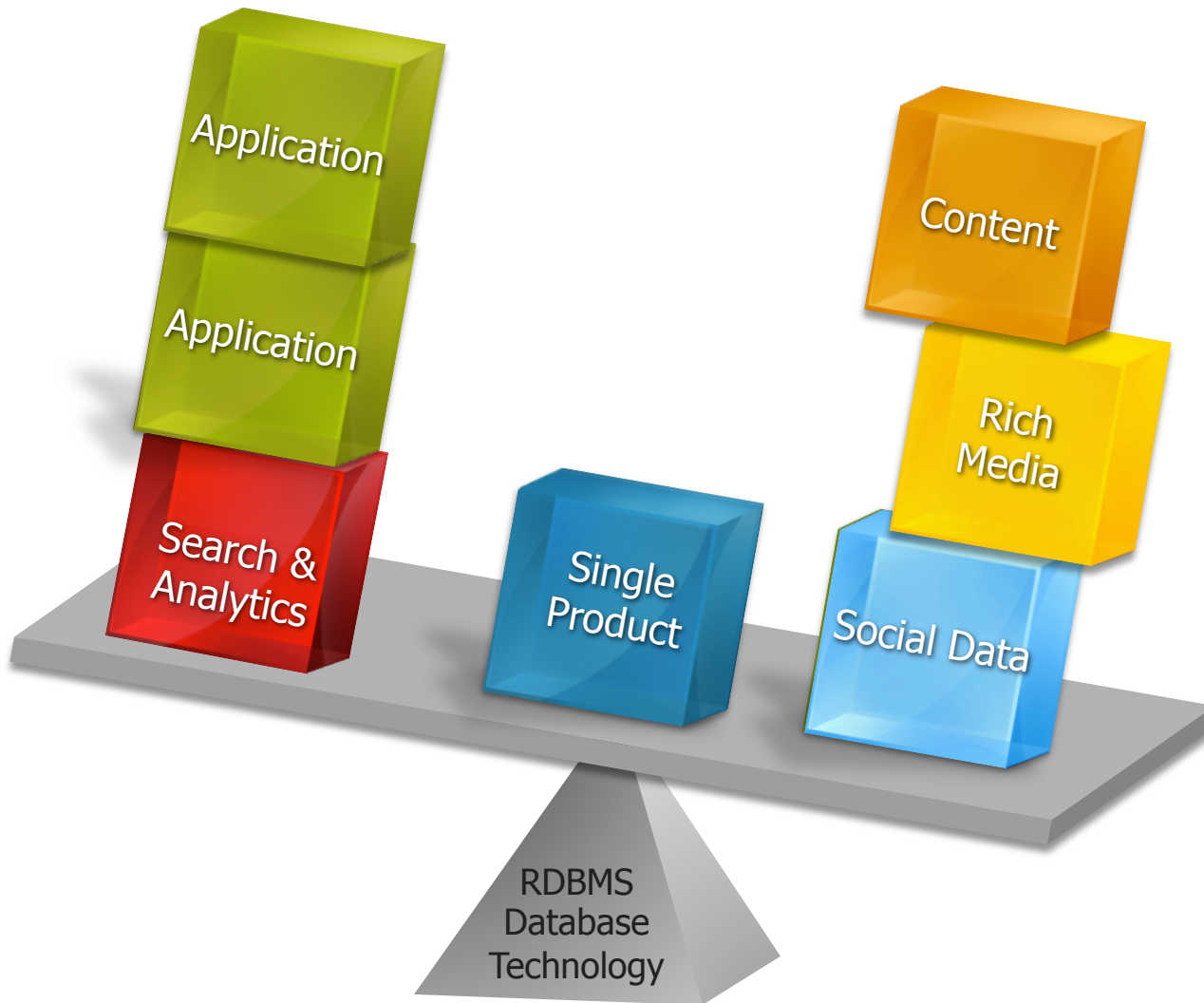


Traditionelle Relationale Datenbanken ist geeignet für “statische” Anwendungsfälle



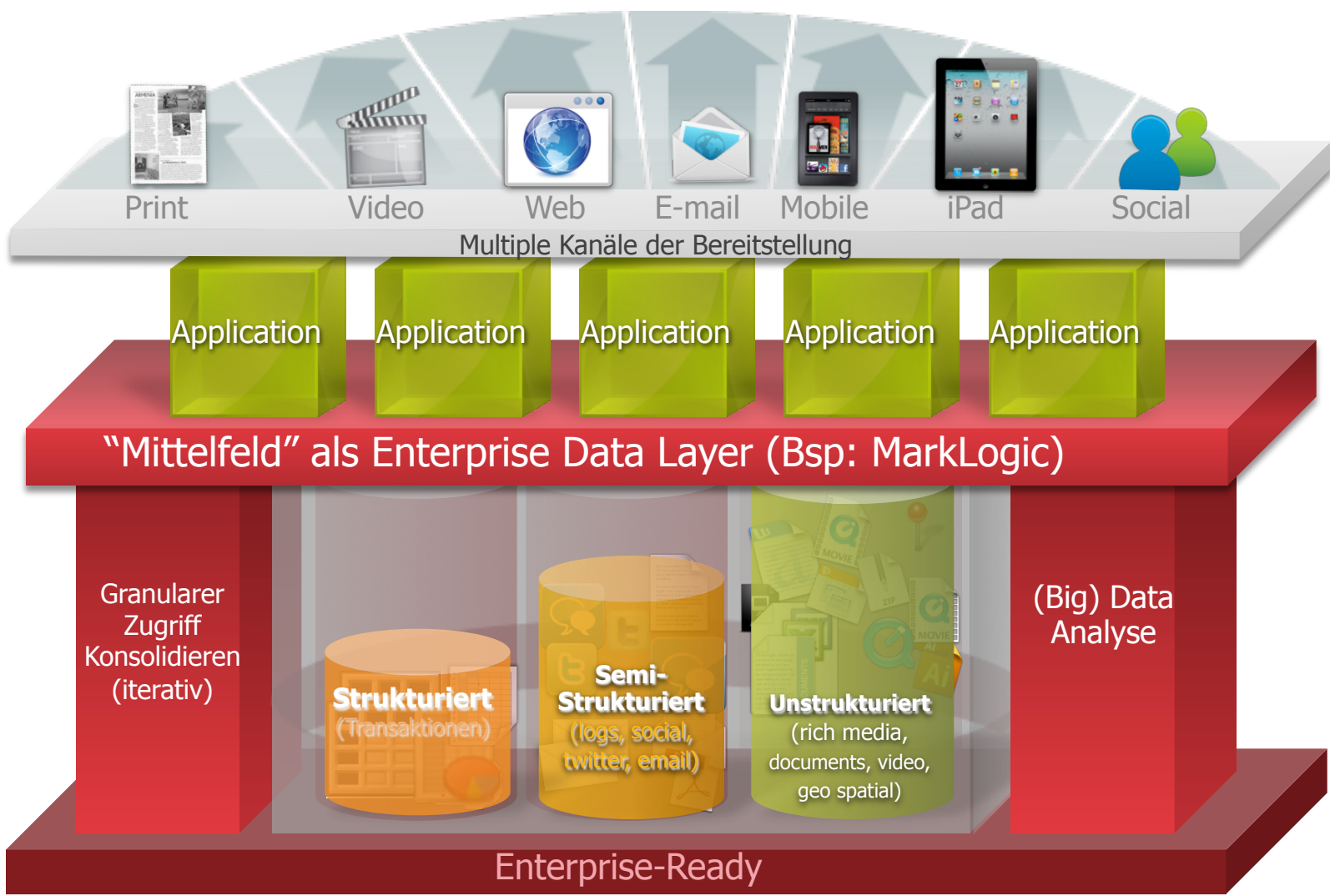
Neue Anforderungen bringen traditionellen Ansatz aus dem Gleichgewicht







Flexibilität in der Datenschicht ist Grundlage für neue Produkte und Geschäftsmodelle



jochen.joerg@marklogic.com
<http://de.linkedin.com/in/jochenjoerg>

www.marklogic.com

MarkLogic Download:
<http://community.marklogic.com/express>

VIELEN DANK!!

Quellen + Links

[NoSQL]: NoSQL Einstieg in die Welt nichtrelationaler Datenbanken,
Edlich, Friedland, Hampe, Brauer - Hanser Verlag

<http://community.marklogic.com/express> (Download free express license)

<http://marklogic.com>

<http://developer.marklogic.com>

<http://hadoop.apache.org>

<http://www.marklogic.com/products-and-services/marklogic-hadoop/>

<http://couchdb.apache.org>

<http://mongodb.org>

<http://de.wikipedia.org/wiki/CouchDB>

<http://de.wikipedia.org/wiki/MongoDB>