Department of Computer Science
University of Cyprus

EPL646 – Advanced Topics in Databases

Lecture 12

Big Data Management II (NoSQL Databases / CouchDB)

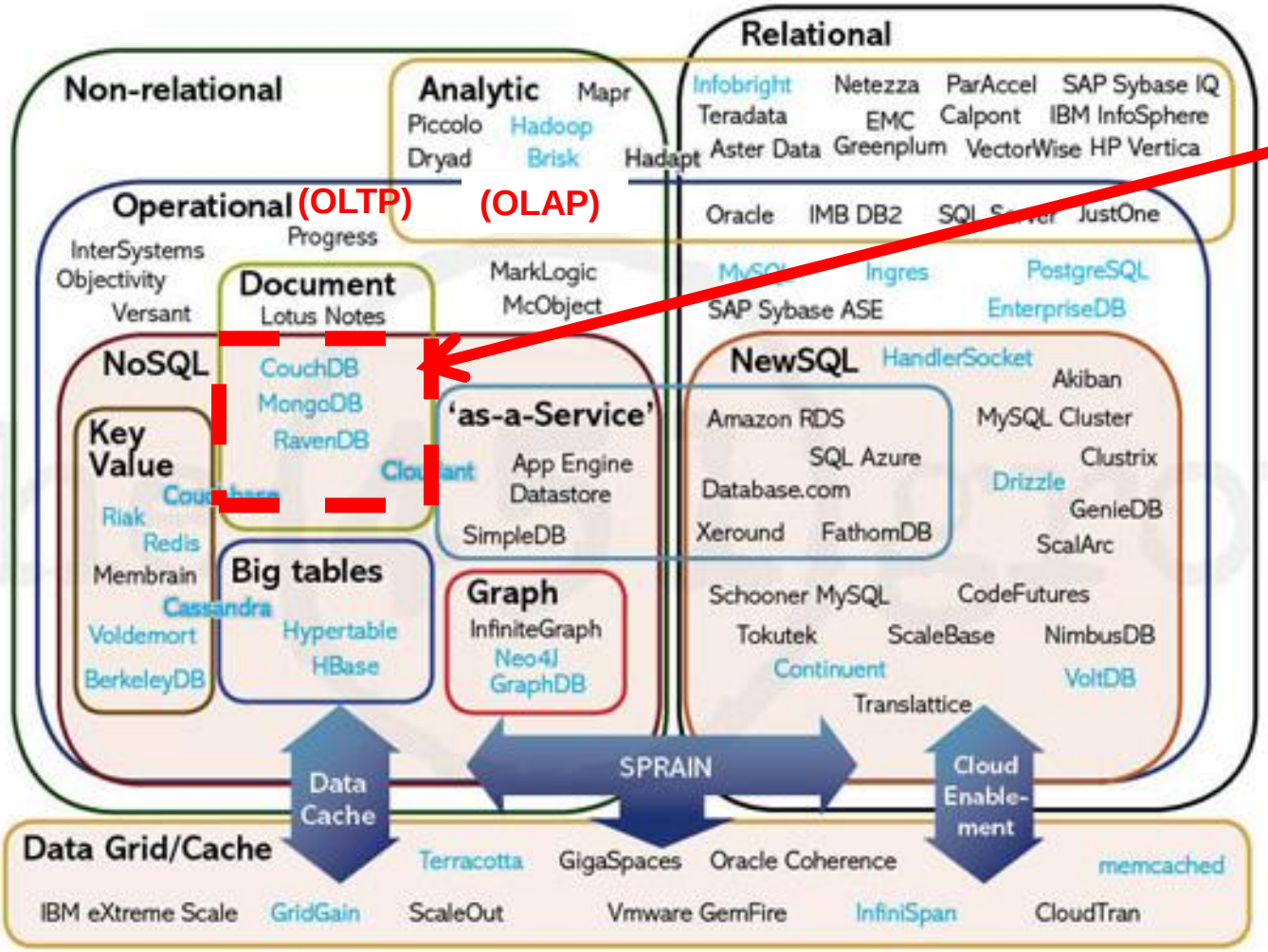
Chapter 20: Abiteboul et. Al.

+ http://guide.couchdb.org/

Demetris Zeinalipour

http://www.cs.ucy.ac.cy/~dzeina/courses/epl646

EPL646: Part B
Distributed/Web/Cloud DBs/Dstores



Lecture Focus

Venn Diagram by 451 group

http://xeround.com/blog/2011/04/newsql-cloud-database-as-a-service

Lecture Outline (Introduction to Semi-structured Data)

- Intro to Web2.0 & JSON Data Interchange Format
- JSON Key-Value Data Model
- CouchDB: A JSON Database (written in Erlang)
  - Using Command Line CURL/ Web-based FUTON
  - CouchDB Architecture (Btrees, Filesystem, Replication)
  - REST Principles
  - Creating DBs, Adding Docs, Updating Docs, Deleting Docs, _ID and _REV issues, Multi-Version CC (MVCC)
  - Querying Data with (Materialized) Views (Map-Reduce style in Javascript)
  - Replication and Scalability Issues

Web 2.0: The Structured Web

DBLP: http://www.informatik.uni-trier.de/

[Numerous sites already allow downloading remote repositories in structured form (e.g., XML)]

Links

- Computer Science Organizations: ACM (DL / SIGMOD / SIGIR), IEEE Computer Society (DL), IEEE Xplore, IFIP, ...
- Related Services: Google Scholar, MS Academic Search, CiteSeer/ CiteSeerX, CS BibTeX (DBLP), io-port.net, CoRR, HAL, NZ-DL, Zentralblatt MATH, MathSciNet, Erdös Number Proj., Math Genealogy Proj., BibSonomy, CiteULike, ScientificCommons, Libra, Arnetminer, RePEc, ...

Schloss Dagstuhl and DBLP join forces

A joint cooperation between Schloss Dagstuhl and Trier University/DBLP aims at strengthening the documentation of research publications in Informatics in a comprehensive, transparent, and open accessible way. ... (news) - (project page)

XML

You may download **DBLP XML records** from http://dblp.uni-trier.de/xml/ - a simple DTD is available. The paper "DBLP - Some Lessons Learned" documents technical details of this XML file. In the appendix "DBLP XML Requests" you may find the description of a primitive DBLP API.

New Design

On the host Trier II we are testing a new design for the DBLP website ...

2 100 000

DBLP now lists more than 2.1 million ($> 2^{21}$) publications. More Statistics about DBLP

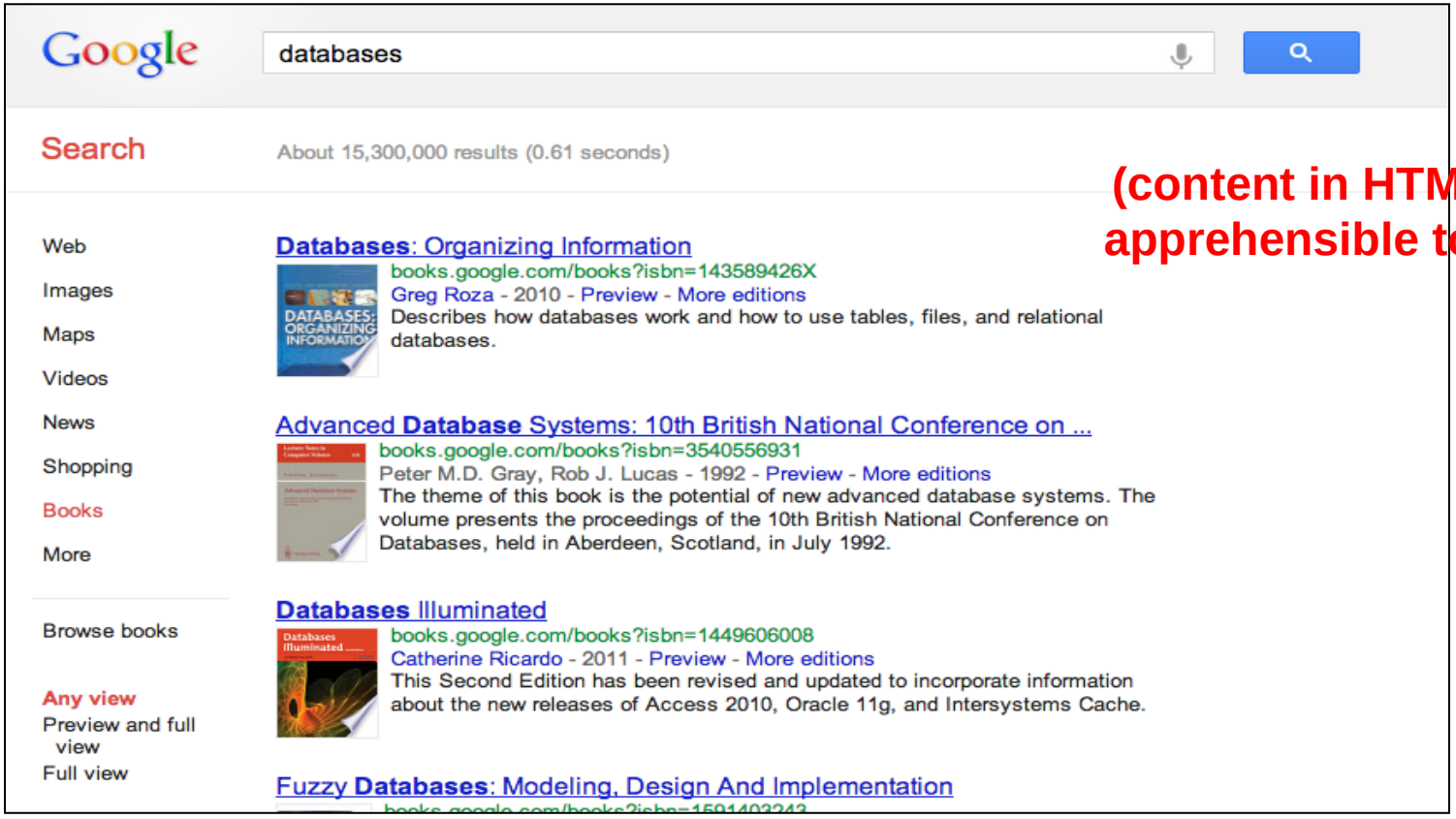
JSON: Web 2.0 Data Interchange Format (JSON: The Fat-free XML)

- The initial vision for XML was to provide a data-**interchange language** to enable **machine-to-machine** communication.
  - However, XML is not well suited to data-interchange as the elements are **taking up to much space.**
- **JSON (JavaScript Object Notation)**
  - RFC4627: a lightweight, text-based, language-independent data interchange format.
- Web services providers nowadays offer their web services in JSON (e.g., Google APIs, Twitter API)
  - The objective of this lecture is to see how to store/query such data with a specialized document store, titled CouchDB (other: MongoDB (open), RavenDB (open))

JSON: Web 2.0 Data Interchange Format (Google Books)

Web1.0: The Unstructured Web

http://books.google.com/



(content in HTML only apprehensible to User)

JSON: Web 2.0 Data Interchange Format (Google Books API)

Web2.0: The Semi-structured Web!

https://www.googleapis.com/books/v1/volumes?q=databases

https://www.googleapis.com/books/v1/volumes?q=databases


```
{
 "kind": "books#volumes",
 "totalItems": 899,
 "items": [
  {
   "kind": "books#volume",
   "id": "4Z6tfpuBmmgC",
   "etag": "urgGiT9QlG4",
   "selfLink": "https://www.googleapis.com/books/v1/volumes/4Z6tfpuBmmgC",
   "volumeInfo": {
    "title": "Databases",
    "subtitle": "Organizing Information",
    "authors": [
     "Greg Roza"
    ],
    "publisher": "Rosen Central",
    "publishedDate": "2010-08-15",
    "description": "Describes how databases work and how to use tables, files, and relational databases.",
    "industryIdentifiers": [
     {
      "type": "ISBN_10",
      "identifier": "143589426X"
     },
     {
      "type": "ISBN_13",
      "identifier": "9781435894266"
     }
    ],
    "pageCount": 48,
    "printType": "BOOK",
    "contentVersion": "preview-1.0.0",
    "imageLinks": {
     "smallThumbnail": "http://bks3.books.google.com/books?id=4Z6tfpuBmmgC&printsec=frontcover&img=1&zoom=5&edge=cu
     "thumbnail": "http://bks3.books.google.com/books?id=4Z6tfpuBmmgC&printsec=frontcover&img=1&zoom=1&edge=curl&so
    },
    "language": "en",
    "previewLink": "http://books.google.com/books?id=4Z6tfpuBmmgC&printsec=frontcover&dq=databases&hl=&cd=1&source=
    "infoLink": "http://books.google.com/books?id=4Z6tfpuBmmgC&dq=databases&hl=&source=gbs_api",
```


content in XML/JSON apprehensible to Computer (data / format decoupled)

*https://www.googleapis.com/books/v1/volumes?q=flowers+inauthor:keyes&**key=yourAPIKey** => Provides additional details (e.g., purchase status)*

JSON: Web 2.0 Data Interchange Format (Twitter API)

https://twitter.com/users/dmslucy.json


```
-{
   "id":742558014,
   "follow_request_sent":null,
   "following":null,
   "screen_name":"DMSLUCY",
   "url":"http:\/\/dmsl.cs.ucy.ac.cy\/",
   "profile_use_background_image":true,
   "created_at":"Tue Aug 07 09:36:30 +0000 2012",
   "profile_text_color":"333333",
   "utc_offset":7200,
   "statuses_count":10,
   "default_profile_image":false,
   "verified":false,
   "name":"DMS Laboratory, UCY",
   "favourites_count":10,
   "profile_sidebar_border_color":"C0DEED",
   "friends_count":0,
   "profile_image_url_https":"https:\/\/si0.twimg.com\/profil
   e_images\/2728729106\/130bc7921970a06228d1ad0d352260de_nor
   mal.png",
   "description":"DMSL belongs to the Computer Science
   Department at the University of Cyprus. We focus on Data
   Engineering Systems and Knowledge Discovery Solutions. ",
   "profile_image_url":"http:\/\/a0.twimg.com\/profile_images
   \/2728729106\/130bc7921970a06228d1ad0d352260de_normal.png"
   ,
```

JSON: Web 2.0 Data Interchange Format (Google Geolocation API)

curl -d @**request.json** -H "Content-Type: application/json" -i "https://www.googleapis.com/geolocation/v1/geolocate?key=YOURKEY"

Request Format (request.json)

```
{
 "homeMobileCountryCode": 310,
 "homeMobileNetworkCode": 260,
 "radioType": "gsm",
 "carrier": "T-Mobile",
 "cellTowers": [
  {
   "cellId": 39627456,
   "locationAreaCode": 40495,
   "mobileCountryCode": 310,
   "mobileNetworkCode": 260,
   "age": 0,
   "signalStrength": -95
  }
 ],
 "wifiAccessPoints": [
  {
   "macAddress": "01:23:45:67:89:AB",
   "signalStrength": 8,
   "age": 0,
   "signalToNoiseRatio": -65,
   "channel": 8
  },
  {
   "macAddress": "01:23:45:67:89:AC",
   "signalStrength": 4,
   "age": 0
  }
 ]
}
```

Response Format

The response format is also JSON.

```
{
  "location": {
  "latitude": 51.0,
  "longitude": -0.1,
  },
  "accuracy": 1200.4,
}
```

JSON: Web 2.0 Data Interchange Format (Other Google APIs)

In fact, Web2.0 Services are omnipresent!

(Google, Twitter, Facebook, Youtube, Linkedin, …)

http://www.programmableweb.com/ - 7800 APIs!!! + 6800 Mashups!

https://code.google.com/apis

Google apis

API Project

Overview
Services
Team
API Access
Billing
Reports
Quotas

All (51) | Active (2) | Inactive (48) | Google Cloud Platform

All services

Select services for the project.

Service	Status	Notes
Ad Exchange Buyer API	OFF	Courtesy limit: 1,000 requests/day
AdSense Host API	Request access...	Courtesy limit: 100,000 requests/day
AdSense Management API	OFF	Courtesy limit: 10,000 requests/day
Analytics API	OFF	Courtesy limit: 50,000 requests/day
Audit API	OFF	Courtesy limit: 10,000 requests/day
BigQuery API	OFF	Courtesy limit: 10,000 requests/day • Pricing
Blogger API v3	Request access...	Courtesy limit: 10,000 requests/day
Books API	ON	Courtesy limit: 1,000 requests/day
Calendar API	OFF	Courtesy limit: 10,000 requests/day
Custom Search API	OFF	Courtesy limit: 100 requests/day • Pricing

The JSON Key-Value Data Model

At the core: key-value construct

Basic example:

```
"title": "The Social network"
```

Atomic data types: character strings, integers, floating-point number and Booleans (`true` or `false`). Non-string values need not be surrounded by '"'.

```
"year": 2010
```

The JSON Key-Value Data Model

Complex values: objects

An *object* is an unordered set of name/value pairs.

Json does not care about types (everything is essentially text)

The types can be distinct, and a key can only appear once.

```
{"last_name": "Fincher", "first_name": "David"}
```

A object can be used as the (complex) value component of a key-value construct:

```
"director": {
      "last_name": "Fincher",
      "first_name": "David",
      "birth_date": 1962
  }
```

The JSON Key-Value Data Model

Complex values: arrays

An array is an ordered collection of values that need not be of the same type.

```
"actors": ["Eisenberg", "Mara", "Garfield", "Timberlake"]
```

A *document* is an object. It can be represented with an unbounded nesting of array and object constructs

```
{
  "title": "The Social network",
  "year": "2010",
  "director": {"last_name": "Fincher",
               "first_name": "David"},
  "actors": [
       {"first_name": "Jesse", "last_name": "Eisenberg"},
       {"first_name": "Rooney", "last_name": "Mara"}
   ]
}
```

CouchDB: A JSON Database

What is CouchDB?

"a database that completely embraces the web"

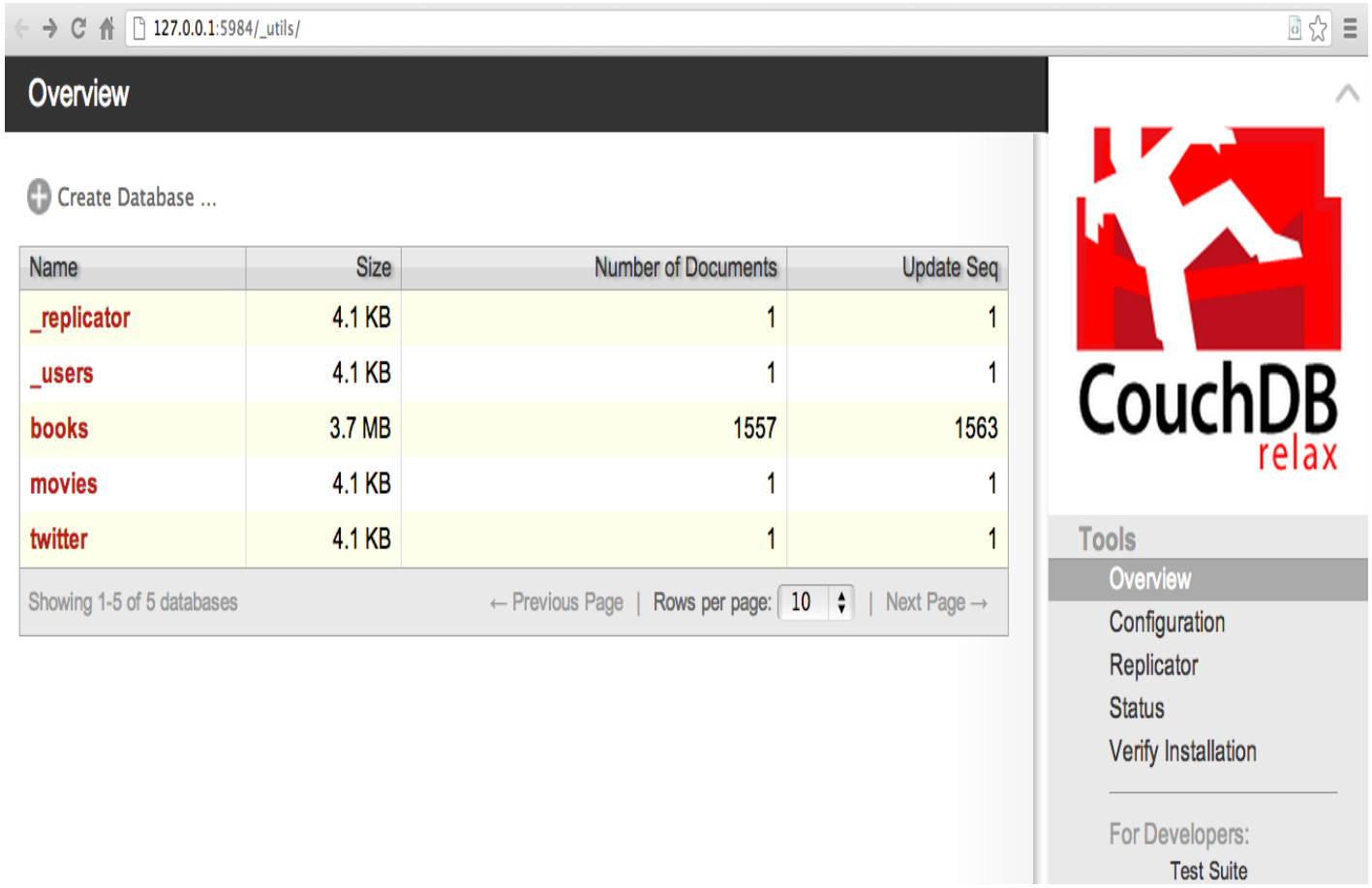
A system representative of the "NoSQL" trend.

1. a semi-structured data model, based on JSON;
2. no schema;
3. *structured materialized views* produced from document collections;
4. views defined with the MAPREDUCE paradigm, allowing both a parallel computation and incremental maintenance of their content;
5. distributed data management techniques: consistent hashing, support for data replication and reconciliation, horizontal scalability, parallel computing, etc.

→ conflict resolution

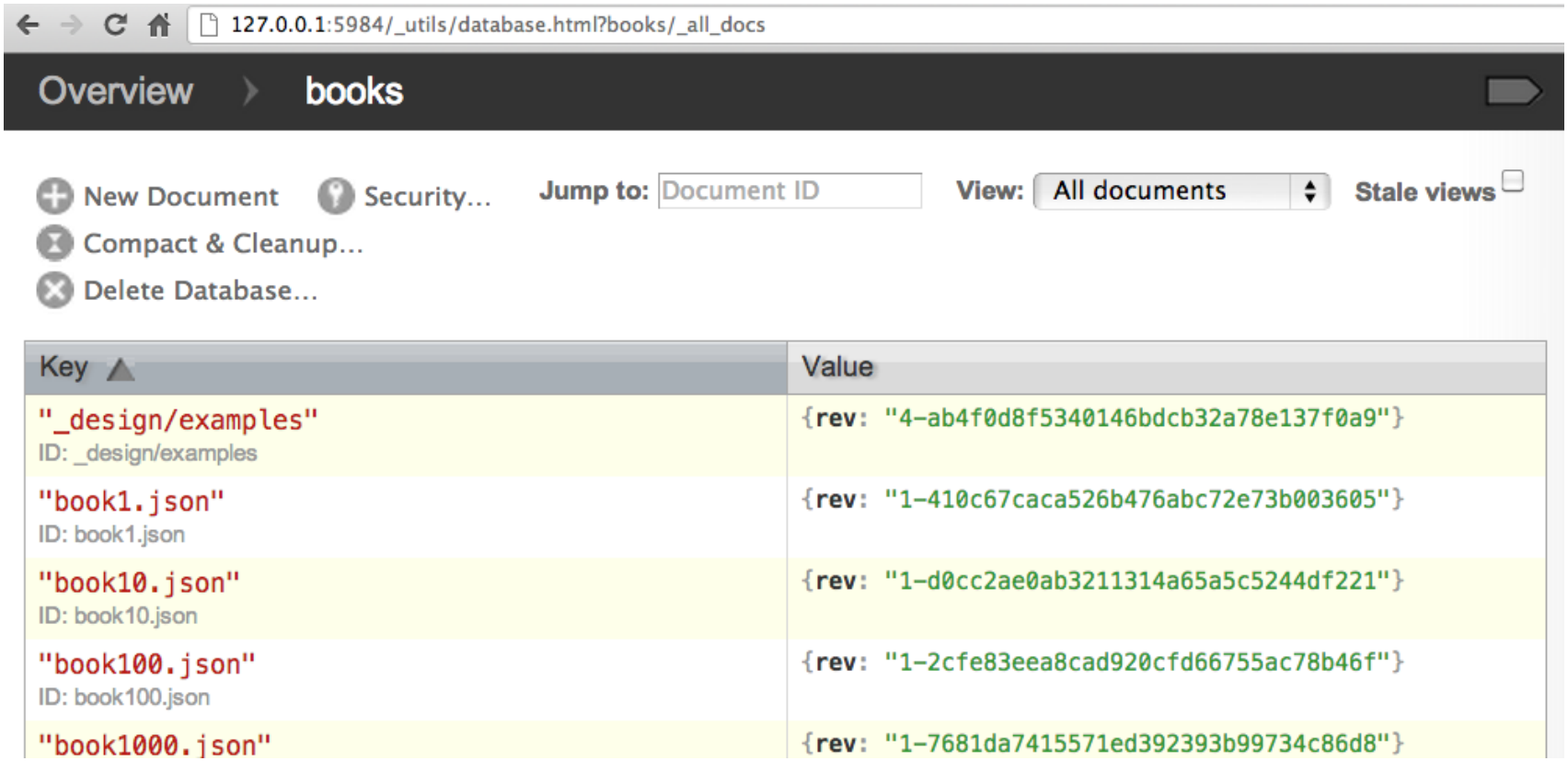
CouchDB: FUTON Web Admin GUI

Futon: A Web-based front-end for administering CouchDB



CouchDB: FUTON Web Admin GUI

Editing records (documents) with Futon



CouchDB in a Nutshell

CouchDB in a nutshell

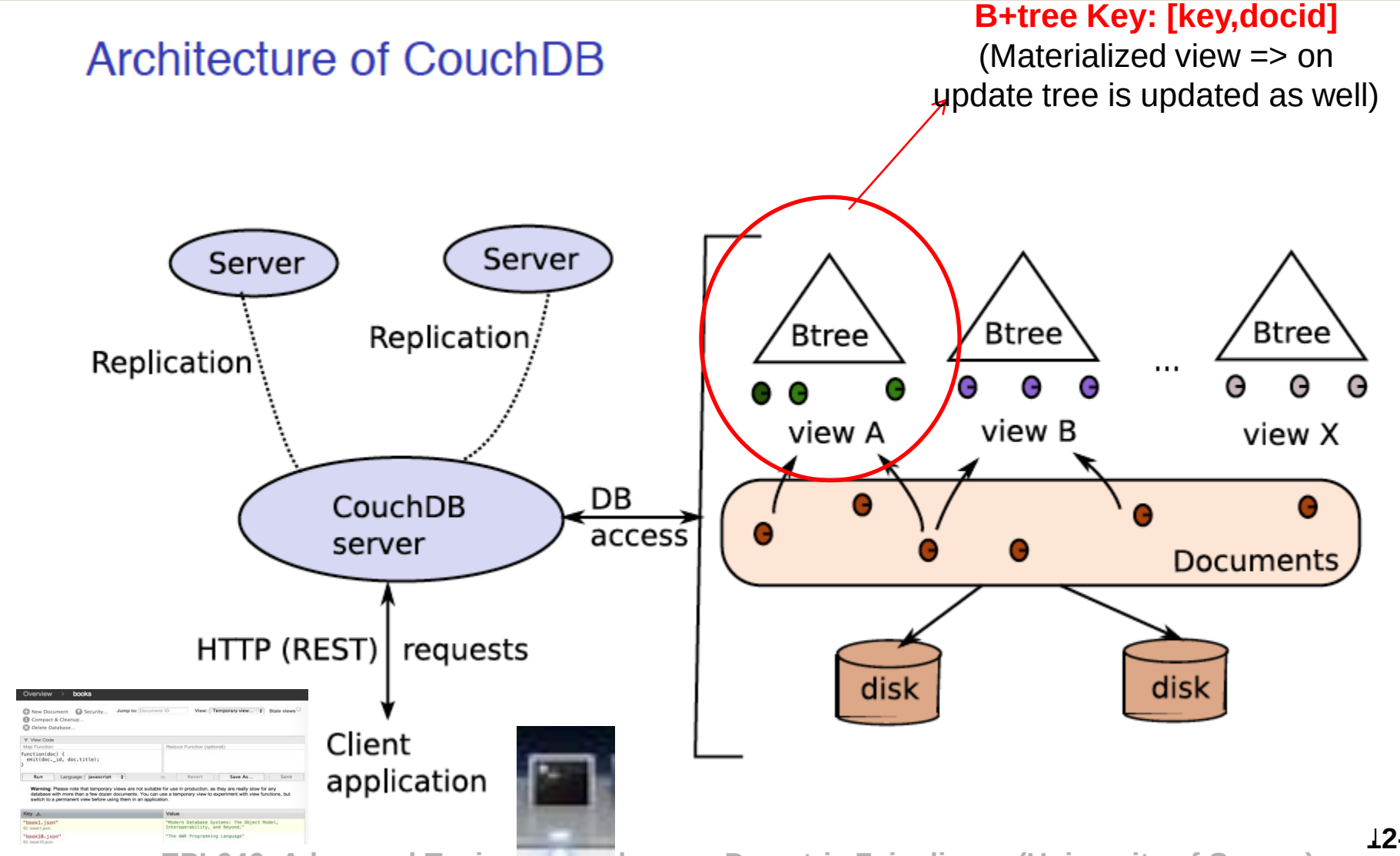
A document, web-oriented data system.

Document oriented. Document are complex and autonomous pieces of information. Can store files, functions, any type of media. But no references.

Typical functionalities of document application: versioning, replication, synchronization, restructuring.

Web-oriented. A document is a resource in the Web sense – it has a URI, and can be manipulated via HTTP (REST architecture).

CouchDB: A JSON Database (Architecture)



CouchDB: Filesystem Layout (Datastores and Materialized Views)