

Distributed CouchApps - Embracing eventual consistency

Dutch PHP Conference 2011

Kore Nordmann

May 20, 2011



About me

- ▶ Kore Nordmann (<kore@php.net>, <kore@apache.org>, <kore@qafoo.com>)
 - ▶ Twitter: @koredn
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiast
- ▶ Contributing to various FLOSS projects
- ▶ Founder of Qafoo GmbH
 - ▶ Provides training & consulting on PHP software quality tools & processes

Outline

Overview

API

Rewrites & VHosts

Replication & Eventual Consistency

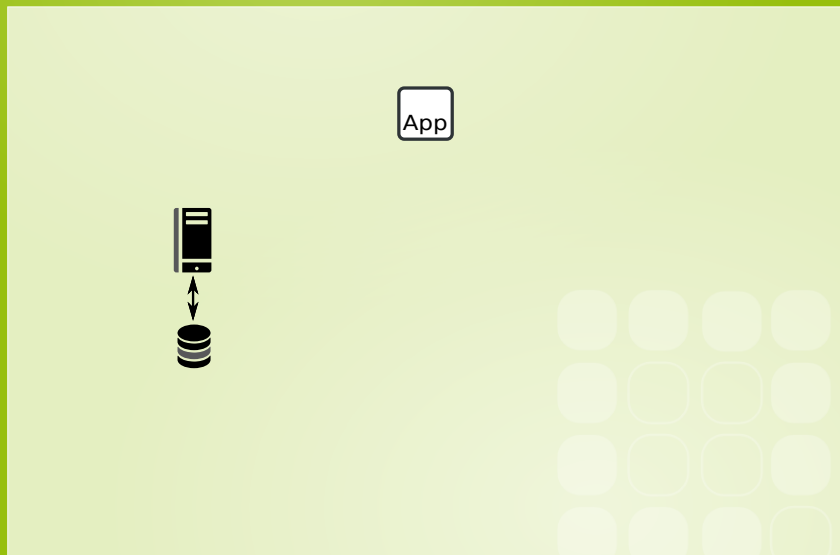
Conclusion



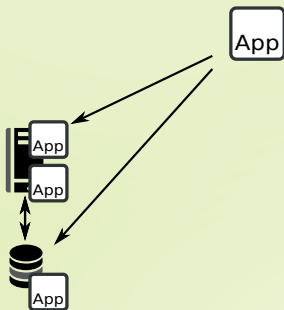
The idea



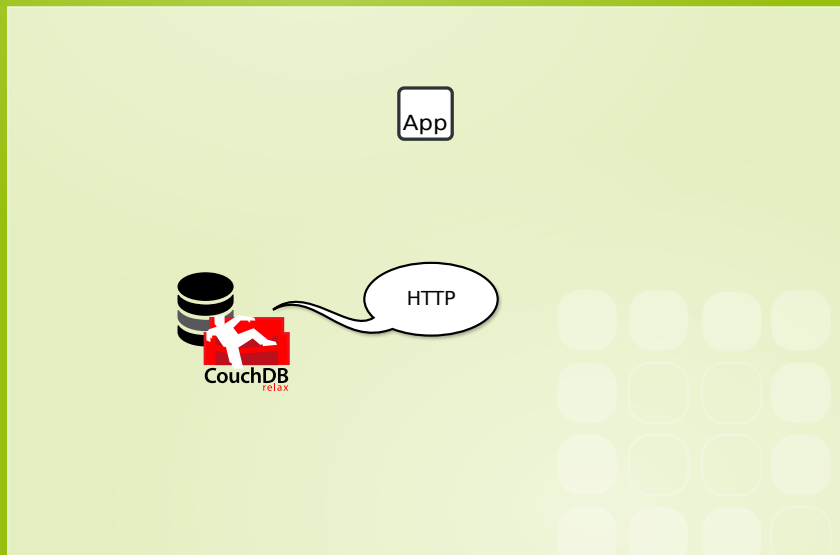
The idea



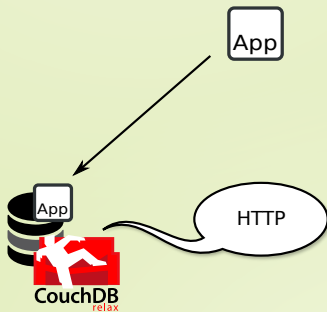
The idea



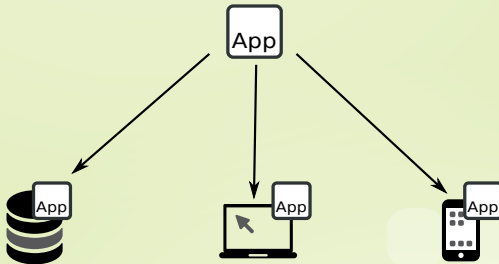
The idea



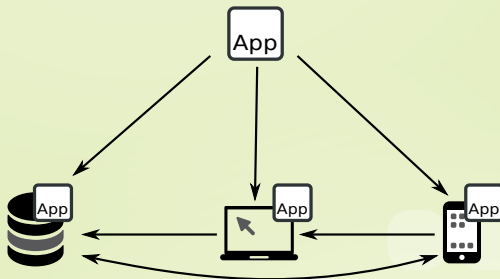
The idea



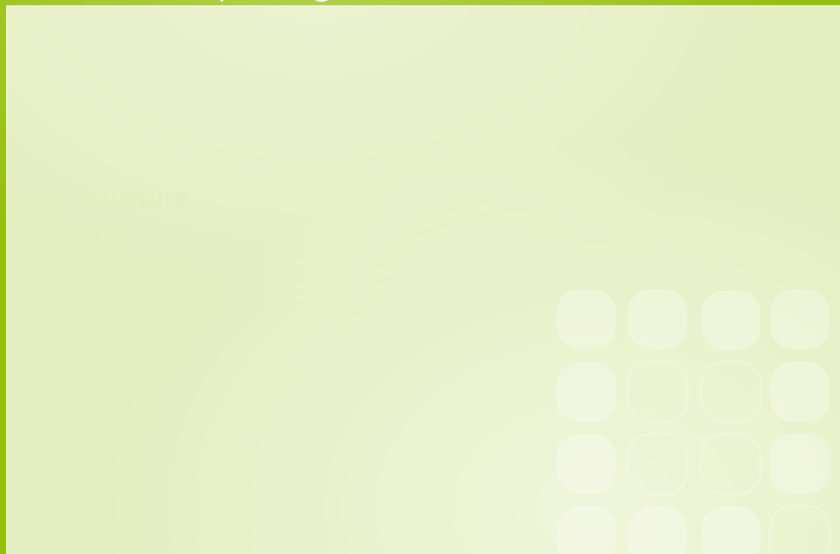
The idea



The idea



CouchDB is paradigm shift



CouchDB is paradigm shift

- ▶ Structure



CouchDB is paradigm shift

- ▶ Structure
- ▶ API



CouchDB is paradigm shift

- ▶ Structure
- ▶ API
- ▶ Consistency



CouchDB is paradigm shift

- ▶ Structure
- ▶ API
- ▶ **Consistency**



CouchDB



CouchDB
relax

top-level project

CouchDB



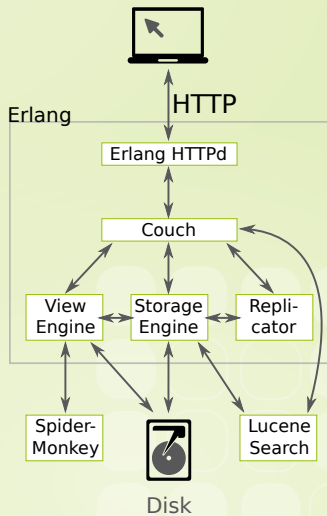
CouchDB

relax

- ▶ Apache top-level project

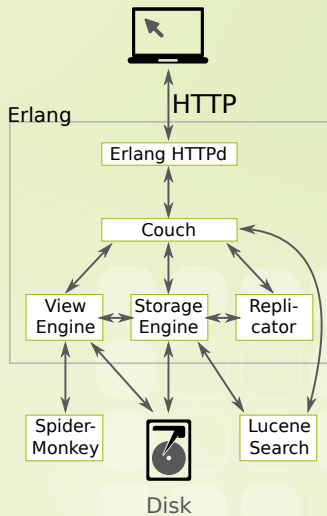
CouchDB

- ▶ Erlang/OTP virtual machine, developed by Ericsson



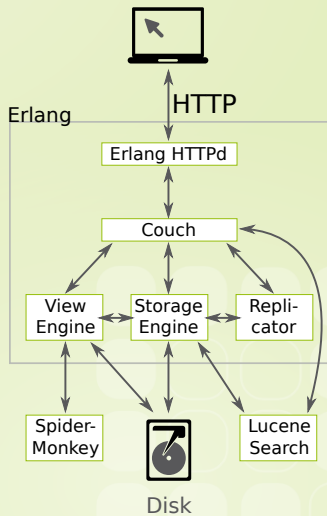
CouchDB

- ▶ Erlang/OTP virtual machine, developed by Ericsson
- ▶ Highly concurrent



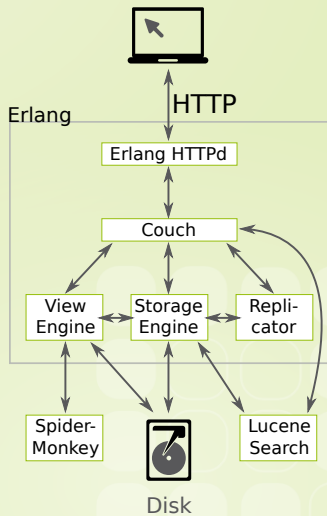
CouchDB

- ▶ Erlang/OTP virtual machine, developed by Ericsson
- ▶ Highly concurrent
- ▶ Scales nearly linearly with the amount of CPUs



CouchDB

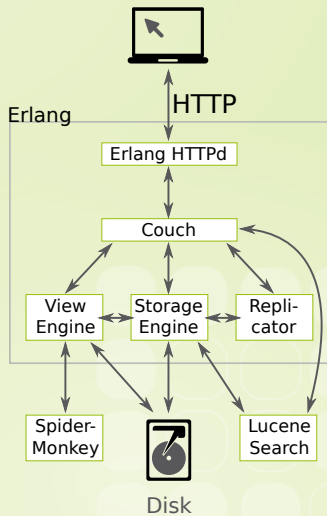
- ▶ Erlang/OTP virtual machine, developed by Ericsson
- ▶ Highly concurrent
- ▶ Scales nearly linearly with the amount of CPUs
- ▶ High reliability (nine nines)



is fast (enough)

CouchDB

- ▶ Erlang/OTP virtual machine, developed by Ericsson
- ▶ Highly concurrent
- ▶ Scales nearly linearly with the amount of CPUs
- ▶ High reliability (nine nines)
- ▶ CouchDB is fast (enough)



Install now!

- ▶ Available for:
 - ▶ Linux, MacOS
 - ▶ Windows
 - ▶ Android
 - ▶ iOS



Install now!

- ▶ Available for:
 - ▶ Linux, MacOS
 - ▶ Windows
 - ▶ Android
 - ▶ iOS



Install now!

- ▶ Available for:
 - ▶ Linux, MacOS
 - ▶ Windows
 - ▶ Android



Install now!

- ▶ Available for:
 - ▶ Linux, MacOS
 - ▶ Windows
 - ▶ Android
 - ▶ iOS



Outline

Overview

API

Rewrites & VHosts

Replication & Eventual Consistency

Conclusion



Create a database

```
1 $ curl -i -X PUT http://localhost:5984/dpc
2 HTTP/1.1 201 Created
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Location: http://localhost:5984/dpc
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 12
7
8 {"ok": true}
```

Create a database

```
1 $ curl -i -X PUT http://localhost:5984/dpc
2 HTTP/1.1 201 Created
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Location: http://localhost:5984/dpc
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 12
7
8 {"ok": true}
```

Create a database

```
1 $ curl -i -X PUT http://localhost:5984/dpc
2 HTTP/1.1 201 Created
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Location: http://localhost:5984/dpc
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 12
7
8 {"ok": true}
```

Re-Create a database

```
1 $ curl -i -X PUT http://localhost:5984/dpc
2 HTTP/1.1 412 Precondition Failed
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Content-Type: text/plain; charset=utf-8
5 Content-Length: 95
6
7 { "error": "file_exists",
8   "reason": "The database could not be created, the file already exists."
9 }
```

Re-Create a database

```
1 $ curl -i -X PUT http://localhost:5984/dpc
2 HTTP/1.1 412 Precondition Failed
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Content-Type: text/plain; charset=utf-8
5 Content-Length: 95
6
7 { "error": "file_exists",
8   "reason": "The database could not be created, the file already exists."
9 }
```


Re-Create a database

```
1 $ curl -i -X PUT http://localhost:5984/dpc
2 HTTP/1.1 412 Precondition Failed
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Content-Type: text/plain; charset=utf-8
5 Content-Length: 95
6
7 { "error": "file_exists",
8   "reason": "The database could not be created, the file already exists."
9 }
```

Get a database

```
1 $ curl -i -X GET http://localhost:5984/dpc
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Content-Type: text/plain; charset=utf-8
5 Content-Length: 211
6
7 { "db_name" :           "dpc" ,
8   "doc_count" :        0,
9   "doc_del_count" :    0,
10  "update_seq" :        0,
11  "purge_seq" :         0,
12  "compact_running" :  false ,
13  "disk_size" :         79,
14  "instance_start_time" : "1299141142936296" ,
15  "disk_format_version" : 5,
16  "committed_update_seq" : 0
17 }
```

Delete a database

```
1 $ curl -i -X DELETE http://localhost:5984/dpc
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Content-Type: text/plain; charset=utf-8
5 Content-Length: 12
6
7 {"ok": true}
```

Delete a database

```
1 $ curl -i -X DELETE http://localhost:5984/dpc
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Content-Type: text/plain; charset=utf-8
5 Content-Length: 12
6
7 {"ok": true}
```

Get a non-existing database

```
1 $ curl -i -X GET http://localhost:5984/dpc
2 HTTP/1.1 404 Object Not Found
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Content-Type: text/plain; charset=utf-8
5 Content-Length: 44
6
7 {"error": "not_found", "reason": "no_db_file"}
```

Outline

API

Documents



Structure

► Example wiki document

```
1 { "title": "Confoo_2011",
2   "text": "Welcome_to_Confoo...",
3   "creator": "user-bar",
4   "edited": 2935678239,
5   "revisions": [
6     { "title": "PHPUK_2010",
7       "text": "Welcome_to_Confoo...",
8       "creator": "user-foo",
9       "edited": 2935678183,
10      }
11   ],
12   ...
13 }
14 }
```

Structure

► Example wiki document

```
1 { "title": "Confoo_2011",
2   "text": "Welcome_to_Confoo...",
3   "creator": "user-bar",
4   "edited": 2935678239,
5   "revisions": [
6     { "title": "PHPUK_2010",
7       "text": "Welcome_to_Confoo...",
8       "creator": "user-foo",
9       "edited": 2935678183,
10      ]
11   },
12   ...
13 ]
14 }
```


Structure

► Example wiki document

```
1 { "title": "Confoo_2011",
2   "text": "Welcome_to_Confoo...",
3   "creator": "user-bar",
4   "edited": 2935678239,
5   "revisions": [
6     { "title": "PHPUK_2010",
7       "text": "Welcome_to_Confoo...",
8       "creator": "user-foo",
9       "edited": 2935678183,
10    }
11  ],
12  ...
13 }
14 }
```



Attachments

► Example wiki document

```
1 { "title":      "Confoo_2011",
2   "creator":   "user-bar",
3   "text":      "<h1>Welcome_to_the_Confoo</h1>
4
5   .....<img_src=\"dpc.2011/logo.png\"_alt=\"Confoo_logo\"/>
6   .....",
7   "_attachments": {
8     "logo.png": {
9       "content_type": "image/png",
10      "stub":          true,
11      "length":       42,
12    }
13  }
14 }
```

Attachments

► Example wiki document

```
1 { "title": "Confoo_2011",
2   "creator": "user-bar",
3   "text": "<h1>Welcome_to_the_Confoo</h1>
4
5   <img_src=\"dpc.2011/logo.png\"_alt=\"Confoo_logo\"/>
6   .....",
7   "_attachments": {
8     "logo.png": {
9       "content_type": "image/png",
10      "stub": true,
11      "length": 42,
12    }
13  }
14 }
```

Attachments

► Example wiki document

```
1 { "title": "Confoo_2011",
2   "creator": "user-bar",
3   "text": "<h1>Welcome_to_the_Confoo</h1>
4
5   ~~~~~<img_src=\"dpc.2011/logo.png\"_alt=\"Confoo_logo\"/>
6   ~~~~~",
7   "_attachments": {
8     "logo.png": {
9       "content_type": "image/png",
10      "stub": true,
11      "length": 42,
12    }
13  }
14 }
```

Create a new document

```
1 $ curl -i -X PUT http://localhost:5984/dpc/test
2   -d '{"title": "Hello World!"}'
3 HTTP/1.1 201 Created
4 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
5 Location: http://localhost:5984/dpc/test
6 Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
7 Content-Type: text/plain; charset=utf-8
8 Content-Length: 67
9
10 { "ok": true ,
11   "id": "test" ,
12   "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
13 }
```



Create a new document

```
1 $ curl -i -X PUT http://localhost:5984/dpc/test
2   -d '{"title": "Hello World!"}'
3 HTTP/1.1 201 Created
4 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
5 Location: http://localhost:5984/dpc/test
6 Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
7 Content-Type: text/plain; charset=utf-8
8 Content-Length: 67
9
10 { "ok": true ,
11   "id": "test" ,
12   "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
13 }
```



Create a new document

```
1 $ curl -i -X PUT http://localhost:5984/dpc/test
2   -d '{"title": "Hello World!"}'
3 HTTP/1.1 201 Created
4 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
5 Location: http://localhost:5984/dpc/test
6 Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
7 Content-Type: text/plain; charset=utf-8
8 Content-Length: 67
9
10 { "ok": true ,
11   "id": "test" ,
12   "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
13 }
```



Create a new document

```
1 $ curl -i -X PUT http://localhost:5984/dpc/test
2   -d '{"title": "Hello World!"}'
3 HTTP/1.1 201 Created
4 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
5 Location: http://localhost:5984/dpc/test
6 Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
7 Content-Type: text/plain; charset=utf-8
8 Content-Length: 67
9
10 { "ok": true ,
11   "id": "test" ,
12   "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
13 }
```


Create a new document

```
1 $ curl -i -X PUT http://localhost:5984/dpc/test
2   -d '{"title": "Hello World!"}'
3 HTTP/1.1 201 Created
4 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
5 Location: http://localhost:5984/dpc/test
6 Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
7 Content-Type: text/plain; charset=utf-8
8 Content-Length: 67
9
10 { "ok": true ,
11   "id": "test" ,
12   "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
13 }
```

Add an attachment

```
1 $ curl -i -X PUT 'http://localhost:5984/dpc/test/logo.png?rev=1-98
    da4da1fd7d7f90ceb752501d5b2321'
2   --data-binary '@graphics/couchdb.png'
3   -H 'Content-Type: image/png'
4 HTTP/1.1 201 Created
5 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
6 Location: http://localhost:5984/dpc/test/logo.png
7 Etag: "2-d9053f096f374284332575ab6956a658"
8 Content-Type: text/plain; charset=utf-8
9 Content-Length: 67
10
11 { "ok": true,
12   "id": "test",
13   "rev": "2-d9053f096f374284332575ab6956a658"
14 }
```

Add an attachment

```
1 $ curl -i -X PUT 'http://localhost:5984/dpc/test/logo.png?rev=1-98
    da4da1fd7d7f90ceb752501d5b2321'
2   --data-binary '@graphics/couchdb.png'
3   -H 'Content-Type: image/png'
4 HTTP/1.1 201 Created
5 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
6 Location: http://localhost:5984/dpc/test/logo.png
7 Etag: "2-d9053f096f374284332575ab6956a658"
8 Content-Type: text/plain; charset=utf-8
9 Content-Length: 67
10
11 { "ok": true,
12   "id": "test",
13   "rev": "2-d9053f096f374284332575ab6956a658"
14 }
```



Add an attachment

```
1 $ curl -i -X PUT 'http://localhost:5984/dpc/test/logo.png?rev=1-98
    da4da1fd7d7f90ceb752501d5b2321'
2   --data-binary '@graphics/couchdb.png'
3   -H 'Content-Type: image/png'
4 HTTP/1.1 201 Created
5 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
6 Location: http://localhost:5984/dpc/test/logo.png
7 Etag: "2-d9053f096f374284332575ab6956a658"
8 Content-Type: text/plain; charset=utf-8
9 Content-Length: 67
10
11 { "ok": true,
12   "id": "test",
13   "rev": "2-d9053f096f374284332575ab6956a658"
14 }
```



Add an attachment

```
1 $ curl -i -X PUT 'http://localhost:5984/dpc/test/logo.png?rev=1-98
2     da4da1fd7d7f90ceb752501d5b2321'
3     --data-binary '@graphics/couchdb.png'
4     -H 'Content-Type: image/png'
5 HTTP/1.1 201 Created
6 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
7 Location: http://localhost:5984/dpc/test/logo.png
8 Etag: "2-d9053f096f374284332575ab6956a658"
9 Content-Type: text/plain; charset=utf-8
10 Content-Length: 67
11 { "ok": true,
12   "id": "test",
13   "rev": "2-d9053f096f374284332575ab6956a658"
14 }
```

Get the document back

```
1 $ curl -i -X GET http://localhost:5984/dpc/test
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Etag: "2-d9053f096f374284332575ab6956a658"
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 177
7
8 { "_id": "test",
9   "_rev": "2-d9053f096f374284332575ab6956a658",
10  "title": "Hello_world!",
11  "_attachments": {
12    "logo.png": {
13      "content_type": "image/png",
14      "revpos":      2,
15      "length":      10785,
16      "stub":        true
17    }
18  }
19 }
```

Get the document back

```
1 $ curl -i -X GET http://localhost:5984/dpc/test
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Etag: "2-d9053f096f374284332575ab6956a658"
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 177
7
8 { "_id": "test",
9   "_rev": "2-d9053f096f374284332575ab6956a658",
10  "title": "Hello_world!",
11  "_attachments": {
12    "logo.png": {
13      "content_type": "image/png",
14      "revpos":      2,
15      "length":      10785,
16      "stub":        true
17    }
18  }
19 }
```

Get the document back

```
1 $ curl -i -X GET http://localhost:5984/dpc/test
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Etag: "2-d9053f096f374284332575ab6956a658"
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 177
7
8 { "_id": "test",
9   "_rev": "2-d9053f096f374284332575ab6956a658",
10  "title": "Hello_world!",
11  "_attachments": {
12    "logo.png": {
13      "content_type": "image/png",
14      "revpos":      2,
15      "length":      10785,
16      "stub":        true
17    }
18  }
19 }
```


Get the document back

```
1 $ curl -i -X GET http://localhost:5984/dpc/test
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 Etag: "2-d9053f096f374284332575ab6956a658"
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 177
7
8 { "_id": "test",
9   "_rev": "2-d9053f096f374284332575ab6956a658",
10  "title": "Hello_world!",
11  "_attachments": {
12    "logo.png": {
13      "content_type": "image/png",
14      "revpos": 2,
15      "length": 10785,
16      "stub": true
17    }
18  }
19 }
```

Get the attachment back

```
1 $ curl -i -X GET http://localhost:5984/dpc/test/logo.png
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 ETag: "2-d9053f096f374284332575ab6956a658"
5 Content-Type: image/png
6 Content-Length: 10785
7 Content-Encoding: identity
8
9 <PNG data>
```

Get the attachment back

```
1 $ curl -i -X GET http://localhost:5984/dpc/test/logo.png
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 ETag: "2-d9053f096f374284332575ab6956a658"
5 Content-Type: image/png
6 Content-Length: 10785
7 Content-Encoding: identity
8
9 <PNG data>
```

Get the attachment back

```
1 $ curl -i -X GET http://localhost:5984/dpc/test/logo.png
2 HTTP/1.1 200 OK
3 Server: CouchDB/1.0.1 (Erlang OTP/R13B)
4 ETag: "2-d9053f096f374284332575ab6956a658"
5 Content-Type: image/png
6 Content-Length: 10785
7 Content-Encoding: identity
8
9 <PNG data>
```

Attachments

- ▶ Attach to a document
 - ▶ HTML
 - ▶ JavaScript
 - ▶ Images

your CouchApp



Attachments

- ▶ Attach to a document
 - ▶ HTML
 - ▶ JavaScript
 - ▶ Images
- ▶ And you got your CouchApp



Design documents

► Defining the “logic” of a CouchApp

```
1 {  
2   _id      : "_design/app",  
3   _rev    : "1-0139ca4e37f873b846fd37714a191e1a",  
4   _attachments : { ... },  
5   views   : { ... },  
6   rewrites : [ ... ],  
7   filters  : { ... },  
8   validate_doc_update : "...",  
9 }
```

Design documents

► Defining the “logic” of a CouchApp

```
1 {  
2   _id           : "_design/app",  
3   _rev         : "1-0139ca4e37f873b846fd37714a191e1a",  
4   _attachments : { ... },  
5   views        : { ... },  
6   rewrites     : [ ... ],  
7   filters      : { ... },  
8   validate_doc_update : "...",  
9 }
```


Design documents

► Defining the “logic” of a CouchApp

```
1 {  
2   _id           : "_design/app",  
3   _rev         : "1-0139ca4e37f873b846fd37714a191e1a",  
4   _attachments : { ... },  
5   views        : { ... },  
6   rewrites     : [ ... ],  
7   filters      : { ... },  
8   validate_doc_update : "...",  
9 }
```

Design documents

► Defining the “logic” of a CouchApp

```
1 {  
2   _id           : "_design/app",  
3   _rev         : "1-0139ca4e37f873b846fd37714a191e1a",  
4   _attachments : { ... },  
5   views        : { ... },  
6   rewrites     : [ ... ],  
7   filters      : { ... },  
8   validate_doc_update : "...",  
9 }
```

Design documents

► Defining the “logic” of a CouchApp

```
1 {  
2   _id           : "_design/app",  
3   _rev         : "1-0139ca4e37f873b846fd37714a191e1a",  
4   _attachments : { ... },  
5   views        : { ... },  
6   rewrites     : [ ... ],  
7   filters      : { ... },  
8   validate_doc_update : "...",  
9 }
```

Design documents

► Defining the “logic” of a CouchApp

```
1 {  
2   _id           : "_design/app",  
3   _rev         : "1-0139ca4e37f873b846fd37714a191e1a",  
4   _attachments : { ... },  
5   views        : { ... },  
6   rewrites     : [ ... ],  
7   filters      : { ... },  
8   validate_doc_update : "...",  
9 }
```

Outline

Overview

API

Rewrites & VHosts

Replication & Eventual Consistency

Conclusion



Rewrites

- ▶ Allows to rewrite URLs
- ▶ Specified in design documents

```
1 { "_id": "_design/app",  
2   "rewrites": [  
3     { "from": "/",  
4       "to": "../../../blog/index.html",  
5     }, ...  
6   ],  
7 }  
8 }
```

Request to:

`http://localhost:5984/db/_design/app/_rewrite/`

Redirects to:

`http://localhost:5984/db/blog/index.html`

Rewrites

- ▶ Allows to rewrite URLs
- ▶ Specified in design documents

```
1 { "_id": "_design/app",  
2   "rewrites": [  
3     { "from": "/",  
4       "to": "../..//blog/index.html",  
5     },...  
6   ]  
7 }  
8
```

GET /_design/app

→ http://localhost:5984/db/_design/app/_rewrite/

→ http://localhost:5984/db/_design/app

→ http://localhost:5984/db/blog/index.html

Rewrites

- ▶ Allows to rewrite URLs
- ▶ Specified in design documents

```
1 { "_id": "_design/app",  
2   "rewrites": [  
3     { "from": "/",  
4       "to": "../..//blog/index.html",  
5     },...  
6   ],  
7 }  
8 }
```

Request as:

`http://localhost:5984/db/_design/app/_rewrite/`

Redirects to:

`http://localhost:5984/db/blog/index.html`

Rewrites

- ▶ Allows to rewrite URLs
- ▶ Specified in design documents

```
1 { "_id": "_design/app",  
2   "rewrites": [  
3     { "from": "/",  
4       "to": "../.. /blog/index.html",  
5     },...  
6   ]  
7 }  
8 }
```

- ▶ Requested as:
`http://localhost:5984/db/_design/app/_rewrite/`
- ▶ Rewrites to:
`http://localhost:5984/db/blog/index.html`

Rewrites

- ▶ Parameters are possible

```
1 { "_id":      "_design/app",  
2   "rewrites": [  
3     { "from":  "/images/:image",  
4       "to":   "../..//images/:image",  
5     },...  
6   ]  
7 }  
8
```

localhost:

`http://localhost:5984/db/_design/app/_rewrite/images/favicon.png`

rewrites to:

`http://localhost:5984/db/images/favicon.png`

Rewrites

- ▶ Parameters are possible

```
1 { "_id":      "_design/app",  
2   "rewrites": [  
3     { "from":  "/images/:image",  
4       "to":    "../..//images/:image",  
5     },...  
6   ]  
7 }  
8 }
```

- ▶ Requested as:

`http://localhost:`

`5984/db/_design/app/_rewrite/images/favicon.png`

- ▶ Rewrites to:

`http://localhost:5984/db/images/favicon.png`

Rewrites

- ▶ Match anything, including query parameters

```
1 { "_id": "_design/app",  
2   "rewrites": [  
3     { "from": "/*",  
4       "to": "../.. /blog/404.html",  
5     }, ...  
6   ]  
7 }  
8 }
```

vHosts

- ▶ Those are still pretty ugly URLs...

- ▶ vhosts to the rescue:

```
curl http://localhost:5984/db/blog
```

```
curl http://localhost:5984/db/blog/index.html
```

```
curl http://localhost:5984/
```

```
curl http://localhost:5984/db/blog/index.html
```

vHosts

- ▶ Those are still pretty ugly URLs...

- ▶ vHosts to the rescue:

```
1 $ cat /etc/couchdb/local.ini
2 [...]
3 myhost:5984 = /db/_design/app/_rewrite
4 [...]
```

myhost:5984/

localhost:5984/db/blog/index.html

vHosts

- ▶ Those are still pretty ugly URLs...

- ▶ vHosts to the rescue:

```
1 $ cat /etc/couchdb/local.ini
2 [...]
3 myhost:5984 = /db/_design/app/_rewrite
4 [...]
```

- ▶ Requested as:

<http://myhost:5984/>

- ▶ Rewrites to:

<http://localhost:5984/db/blog/index.html>

Setting up a CouchApp

► Create vHost

1. Create view document, containing:

2. Create roles

3. Add resources: HTML, JavaScript, Images

4. Add functions: View functions, show functions, list functions

5. Add validation functions

Setting up a CouchApp

- ▶ Create vHost
- ▶ Create view document, containing:
 - ▶ Rewrite rules

• HTML, JavaScript, Images
• View functions, show functions, list functions
• Validation functions

Setting up a CouchApp

- ▶ Create vHost
- ▶ Create view document, containing:
 - ▶ Rewrite rules
 - ▶ Attachments: HTML, JavaScript, Images

▶ Attachments: view functions, show functions, list functions, validation functions

Setting up a CouchApp

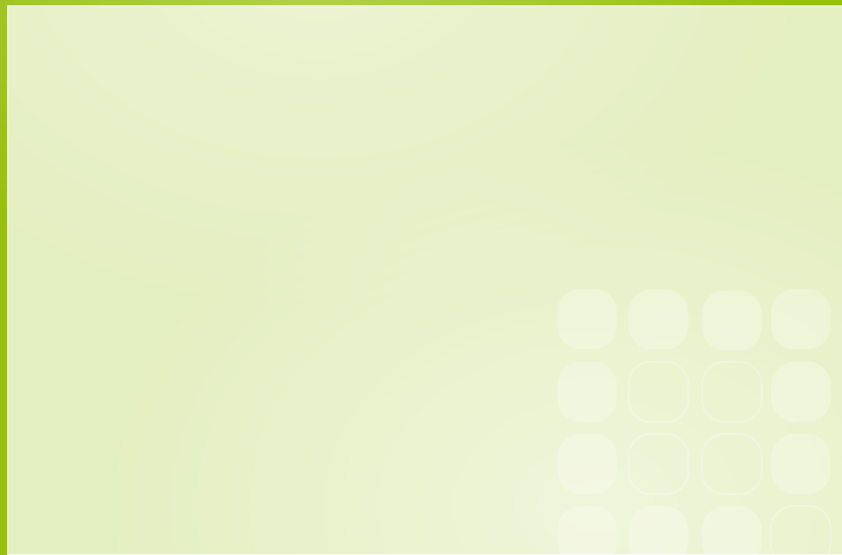
- ▶ Create vHost
- ▶ Create view document, containing:
 - ▶ Rewrite rules
 - ▶ Attachments: HTML, JavaScript, Images
 - ▶ Optionally: View functions, show functions, list functions

validation functions

Setting up a CouchApp

- ▶ Create vHost
- ▶ Create view document, containing:
 - ▶ Rewrite rules
 - ▶ Attachments: HTML, JavaScript, Images
 - ▶ Optionally: View functions, show functions, list functions
 - ▶ Document validation functions

Demo



Outline

Overview

API

Rewrites & VHosts

Replication & Eventual Consistency

Conclusion



Handling consistency

- ▶ Two possibilities for handling distributed data:

- ▶ **Strong consistency**
 - Try to keep all servers in sync
 - Not fault-tolerant and merge data, if required
 - Servers will not be consistent



Handling consistency

- ▶ Two possibilities for handling distributed data:
 - ▶ Force to keep all servers in sync

▶ Be more tolerant and merge data, if required
▶ Servers don't have to be consistent



Handling consistency

- ▶ Two possibilities for handling distributed data:
 - ▶ Force to keep all servers in sync
 - ▶ Be partition tolerant and merge data, if required

Eventually be consistent

Handling consistency

- ▶ Two possibilities for handling distributed data:
 - ▶ Force to keep all servers in sync
 - ▶ Be partition tolerant and merge data, if required
 - ▶ *Eventually* be consistent

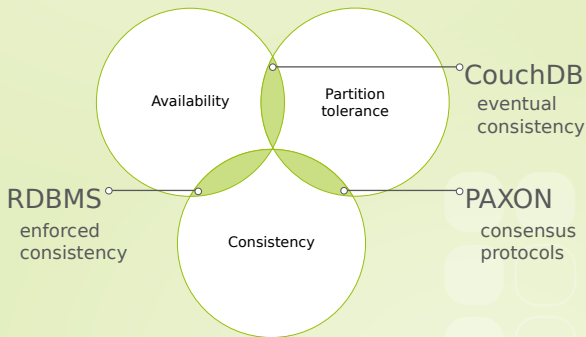


Remember:

- ▶ There is no ensured inter document consistency in CouchDB (relational integrity)

Scaling: The CAP theorem

- ▶ The CAP theorem, read more in “CouchDB: The Definitive Guide” [JCA09]



- ▶ CouchDB employs “Eventual Consistency” [Vog09]

Outline

Replication & Eventual Consistency

- Replication

- Filtered Replication

- Conflicts



Replication

► Replication is trivial

```
1 $ curl -X POST http://localhost:5984/_replicate \  
2   -H 'Content-Type: application/json' \  
3   -d '{"source": "dpc", "\\  
4   "target": "http://user:pass@192.168.1.3:5984/dpc"}'  
5  
6 { "ok": true ,  
7   "no_changes": true ,  
8   "session_id": "73d69e7b5cdaea059e55ed1db7802151" ,  
9   "source_last_seq": 141 ,  
10  "history": [ {  
11    ...  
12  } ]  
13 }
```

source and target can be any combination of remote and local

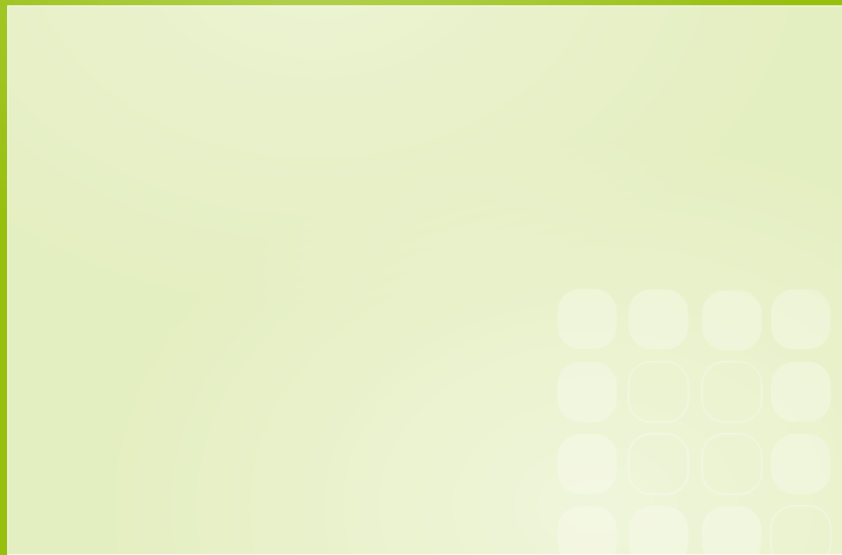
Replication

► Replication is trivial

```
1 $ curl -X POST http://localhost:5984/_replicate \  
2   -H 'Content-Type: application/json' \  
3   -d '{"source": "dpc", "\  
4   "target": "http://user:pass@192.168.1.3:5984/dpc"}'  
5  
6 { "ok": true ,  
7   "no_changes": true ,  
8   "session_id": "73d69e7b5cdaea059e55ed1db7802151" ,  
9   "source_last_seq": 141 ,  
10  "history": [ {  
11    ...  
12  } ]  
13 }
```

► Source and target can be any combination of remote and local URLs

Demo



Outline

Replication & Eventual Consistency

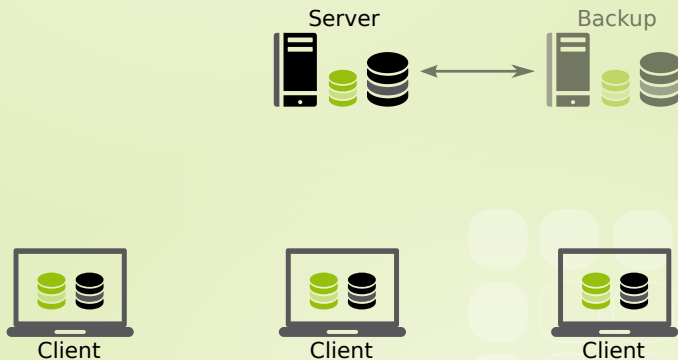
Replication

Filtered Replication

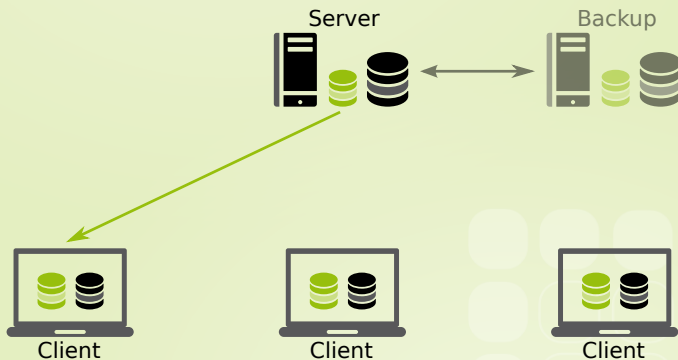
Conflicts



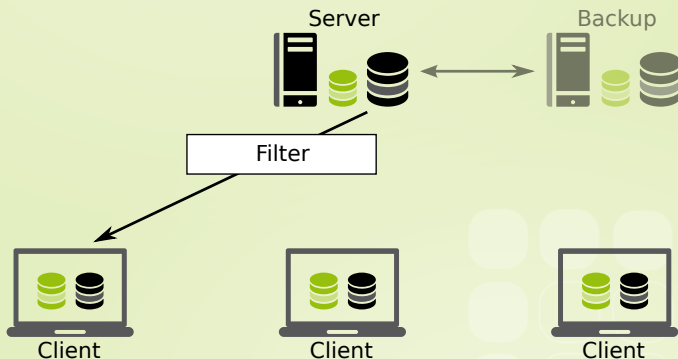
Filtered Replication



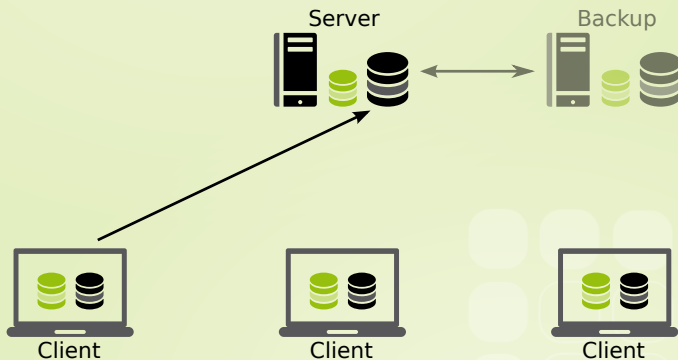
Filtered Replication



Filtered Replication



Filtered Replication



Filtered Replication

- ▶ Append a filter function to design document

```
1 { "_id": "_design/app",  
2   "language": "javascript",  
3   "filters": {  
4     "for_user": "function(_doc, _req){_return _false;}" ,...  
5  
6   }...  
7  
8 }
```

Filtered Replication

► Common filtering function

```
1 function( doc, req ) {  
2   if( !req.userCtx.name ) {  
3     throw( "Unauthorized!" );  
4   }  
5  
6   if( doc.recipients &&  
7     doc.recipients.indexOf( req.userCtx.name ) !== -1 )  
8     {  
9       return true;  
10    }  
11  
12   return false;  
13 }
```

Filtered Replication

► Usage during replication

```
1 $ curl -X POST http://localhost:5984/_replicate \  
2   -H 'Content-Type: application/json' \  
3   -d '{"source": "http://user:pass@192.168.1.3:5984/dpc", \  
4     "target": "dpc", \  
5     "filter": "app/for_user"}'...
```


Outline

Replication & Eventual Consistency

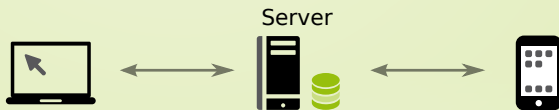
Replication

Filtered Replication

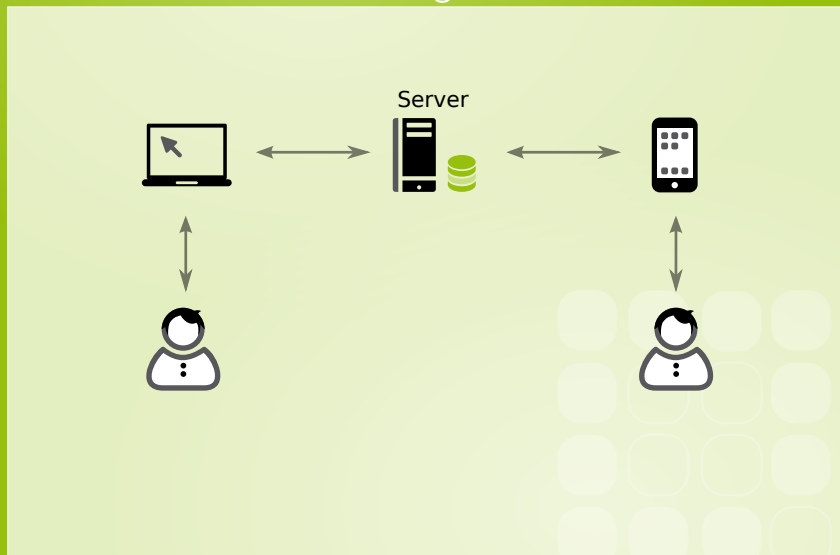
Conflicts



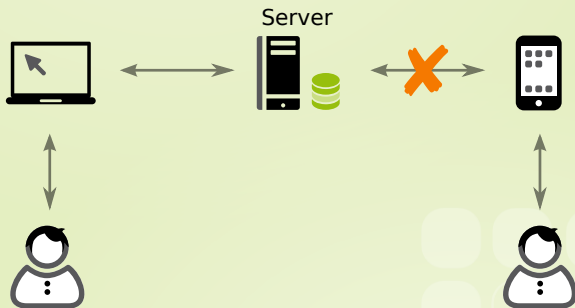
Document conflict handling



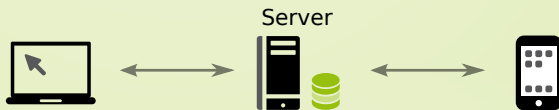
Document conflict handling



Document conflict handling



Document conflict handling



Guidelines

- ▶ Keep documents small and segregated
 - ▶ Aggregate data in views
- ▶ Check if you need manual conflict handling
 - ▶ Document merging is highly application specific

Examples

- ▶ Twitter App

Timeline of tweets & comments



Examples

- ▶ Twitter App
- ▶ Blog posts & comments



Examples

- ▶ Twitter App
- ▶ Blog posts & comments
- ▶ A wiki



Security

- ▶ Document validation functions
 - ▶ Limit changes based on users / groups
- ▶ Server / database admins
 - ▶ Limit read / write access to databases
- ▶ Filtered replication
 - ▶ Limit replicated user data based on users / groups

▶ You might want to use an HTTP proxy

Security

- ▶ Document validation functions
 - ▶ Limit changes based on users / groups
- ▶ Server / database admins
 - ▶ Limit read / write access to databases
- ▶ Filtered replication
 - ▶ Limit replicated user data based on users / groups
- ▶ You still might want to use an HTTP proxy

Problems

- ▶ VHosts and not replicatable
- ▶ Server side script includes are not replicable

Outline

Overview

API

Rewrites & VHosts

Replication & Eventual Consistency

Conclusion



Thanks for listening

- ▶ More information:
 - ▶ <http://couchdb.org>
 - ▶ <http://guide.couchdb.org>
 - ▶ <http://couchapp.org>
- ▶ More about us:
 - ▶ <http://qafoo.com>
- ▶ Please rate & comment:
 - ▶ <http://joind.in/3236>



Bibliography I

- [JCA09] Noah Slater J. Chris Anderson, Jan Lehnardt, *Couchdb: The definitive guide*, O'Reilly Media, Inc., 2009.
- [Vog09] Werner Vogels, *Eventually consistent - revisited*,
http://www.allthingsdistributed.com/2008/12/eventually_consistent.html, December 2009.