Distributed CouchApps - Embracing eventual consistency

Confoo

Kore Nordmann

March 9, 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

CouchDB is paradigmn shift

- Structure
- Consistency

CouchDB is paradigmn shift

- Structure

CouchDB is paradigmn shift

- Structure
- Consistency

CouchDB is paradigmn shift

- Structure
- Consistency
- API

CouchDB is paradigmn shift

- Structure
- Consistency
- API
- Applications

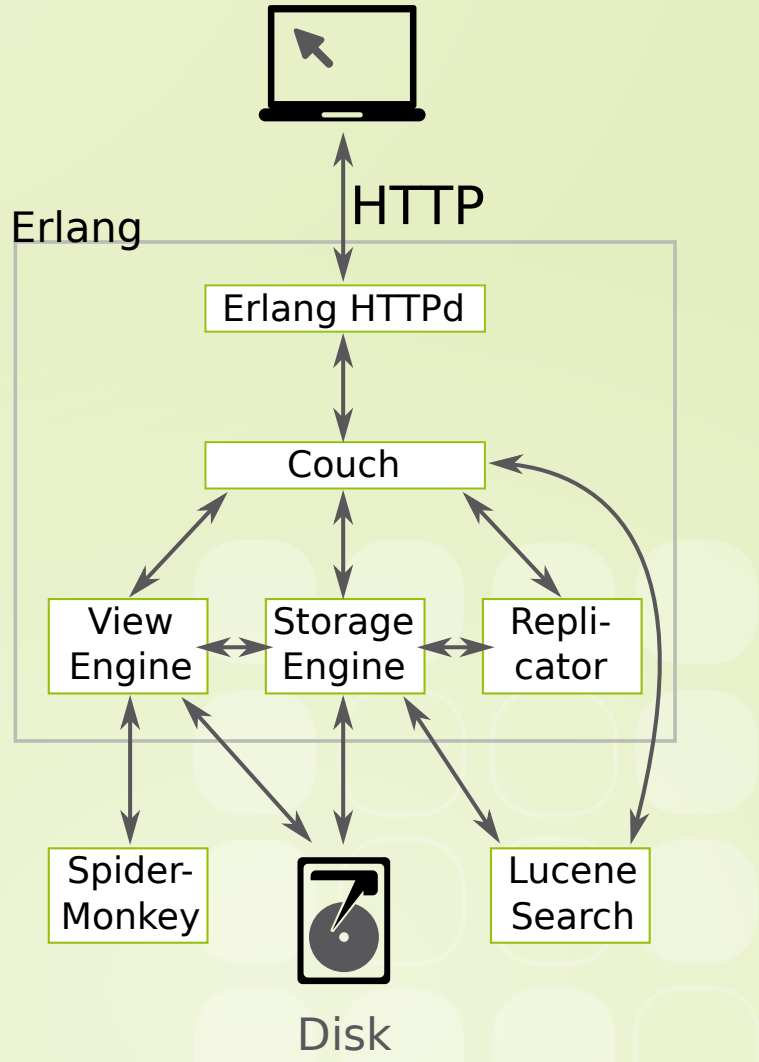
CouchDB

CouchDB

- Apache top-level project

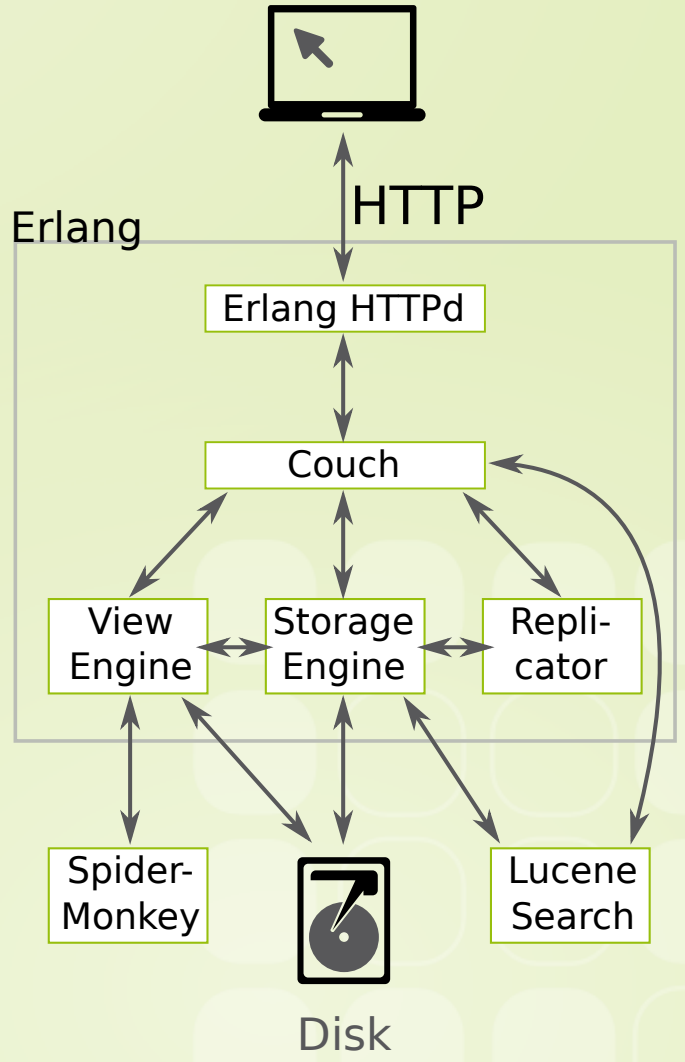
CouchDB

- Erlang/OTP virtual machine, developed by Ericsson



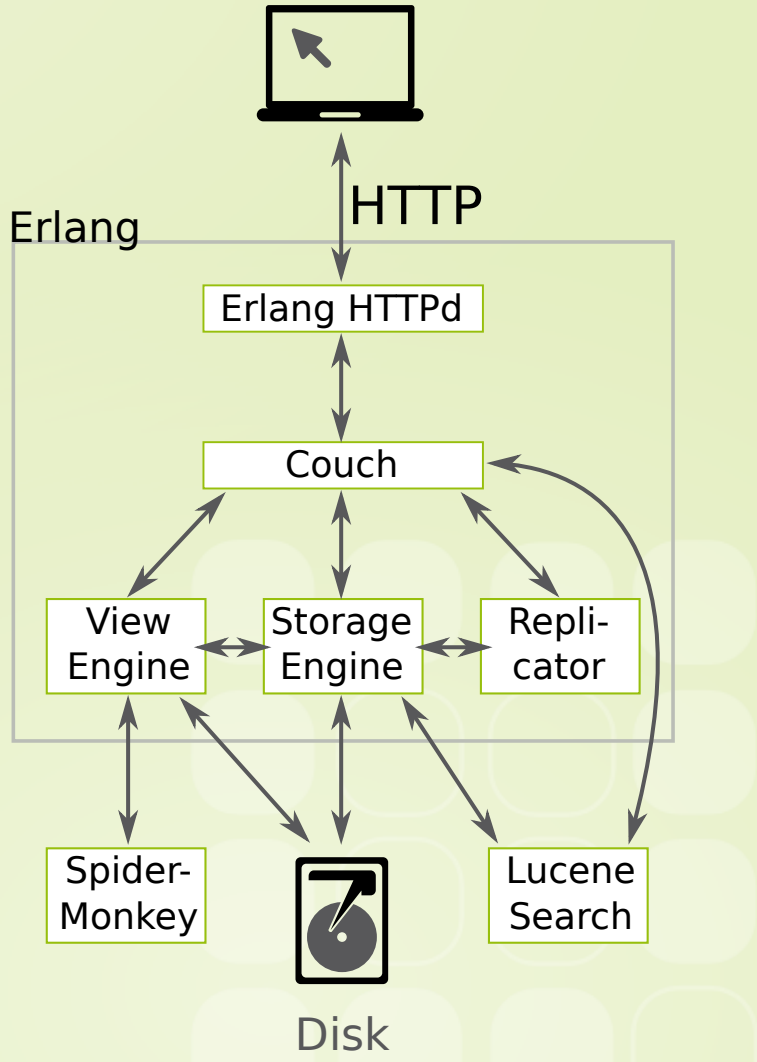
CouchDB

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



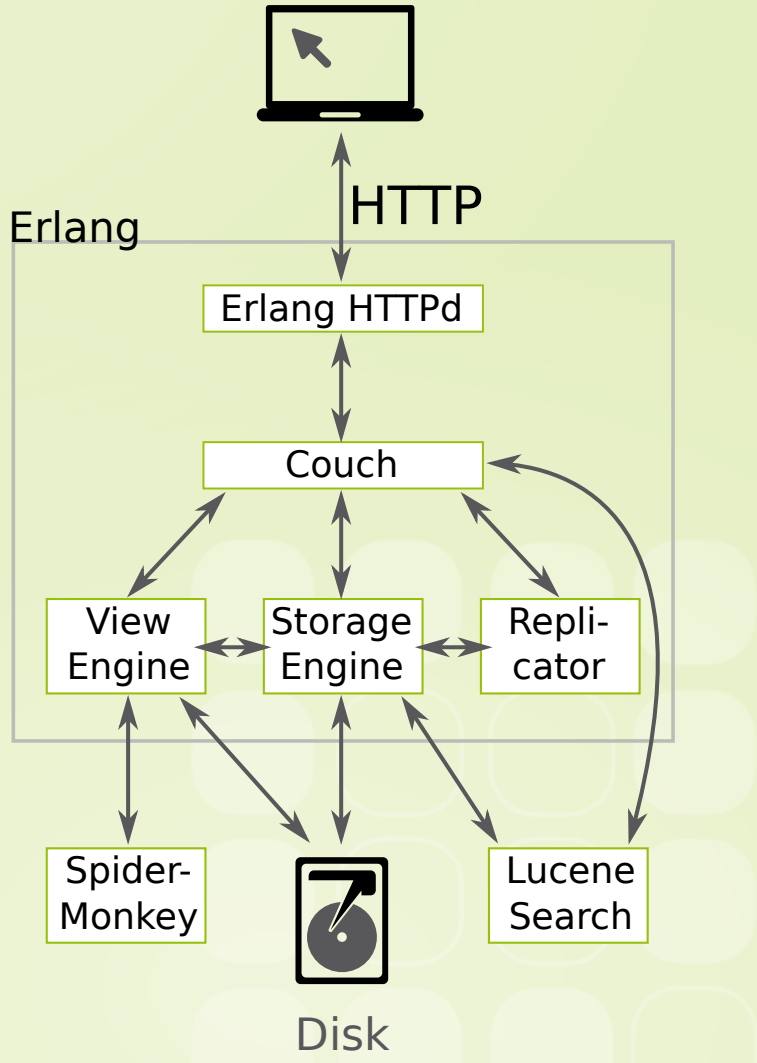
CouchDB

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



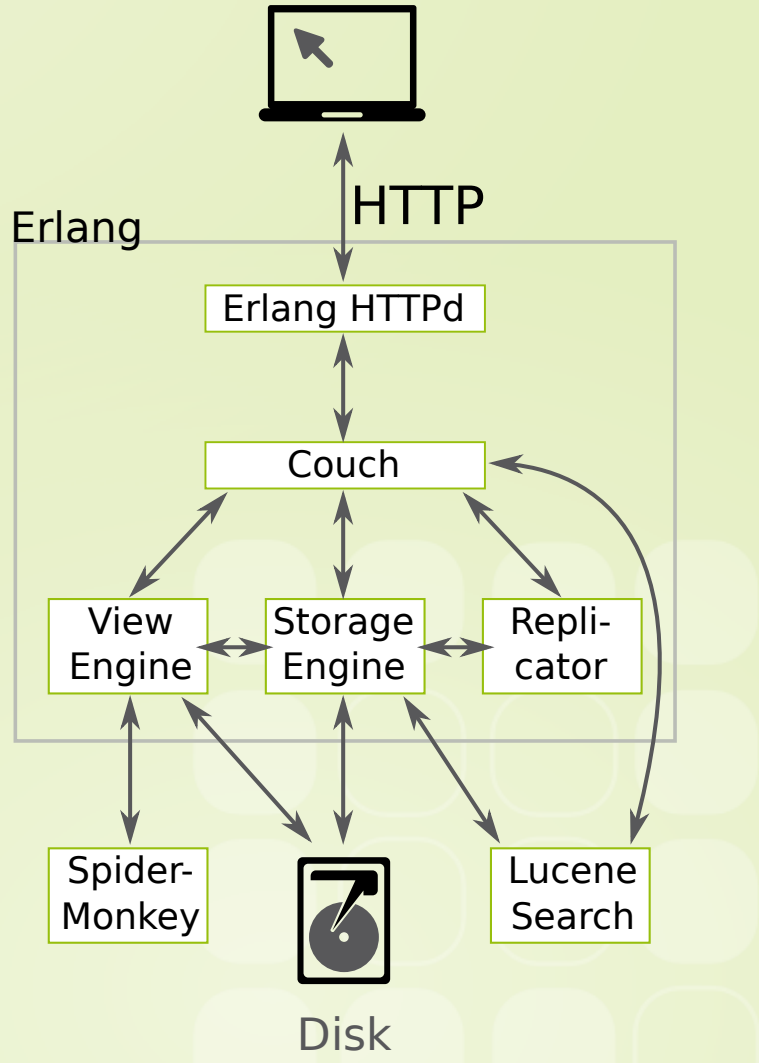
CouchDB

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



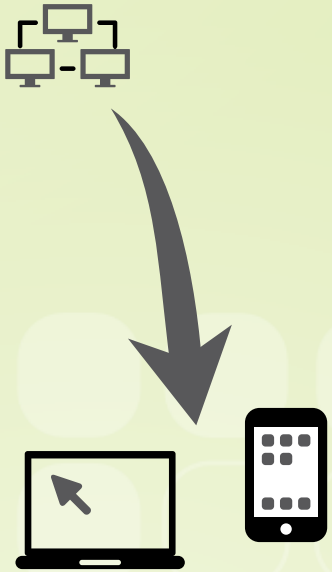
CouchDB

- 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)



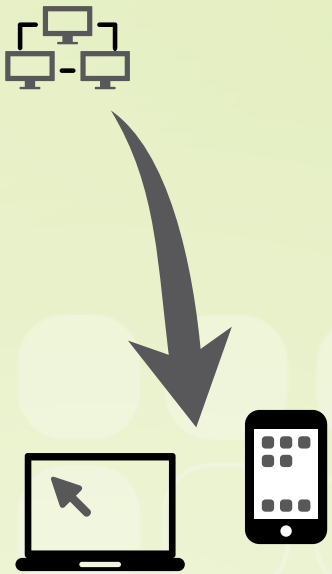
Install now!

- Availbale for:
  - Linux, MacOS
  - Windows
  - Android



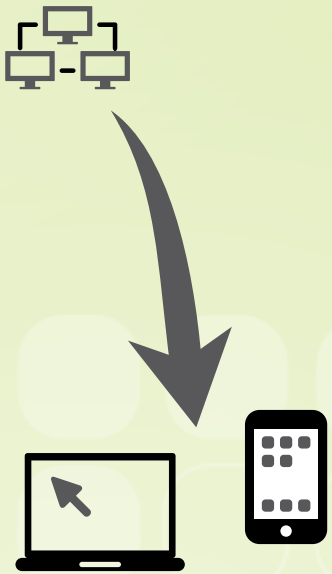
Install now!

- Availbale for:
  - Linux, MacOS
  - Windows
  - Android



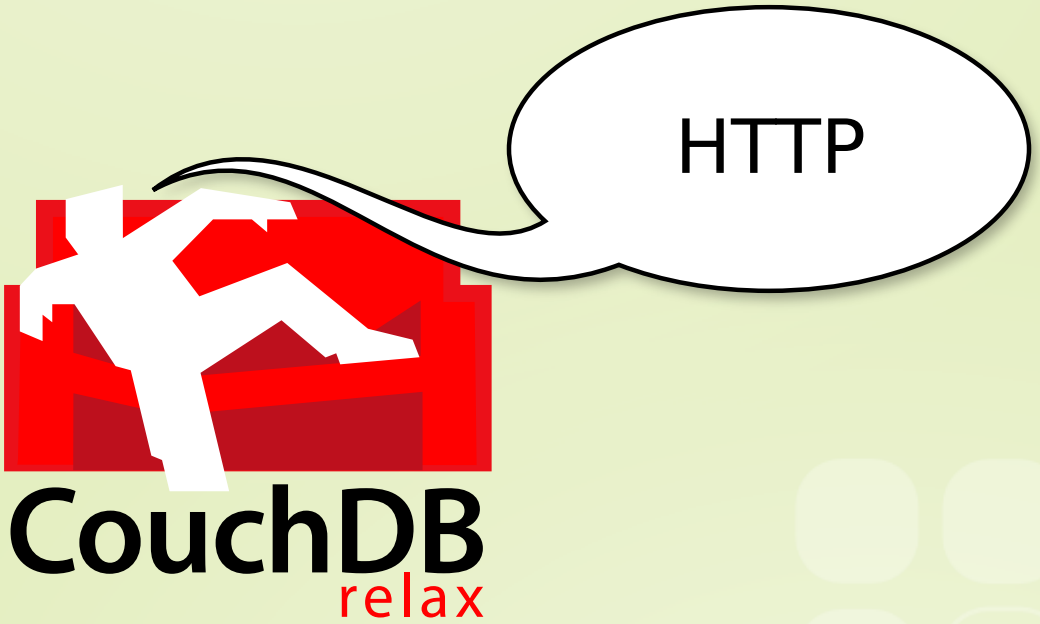
Install now!

- Availbale for:
  - Linux, MacOS
  - Windows
  - Android

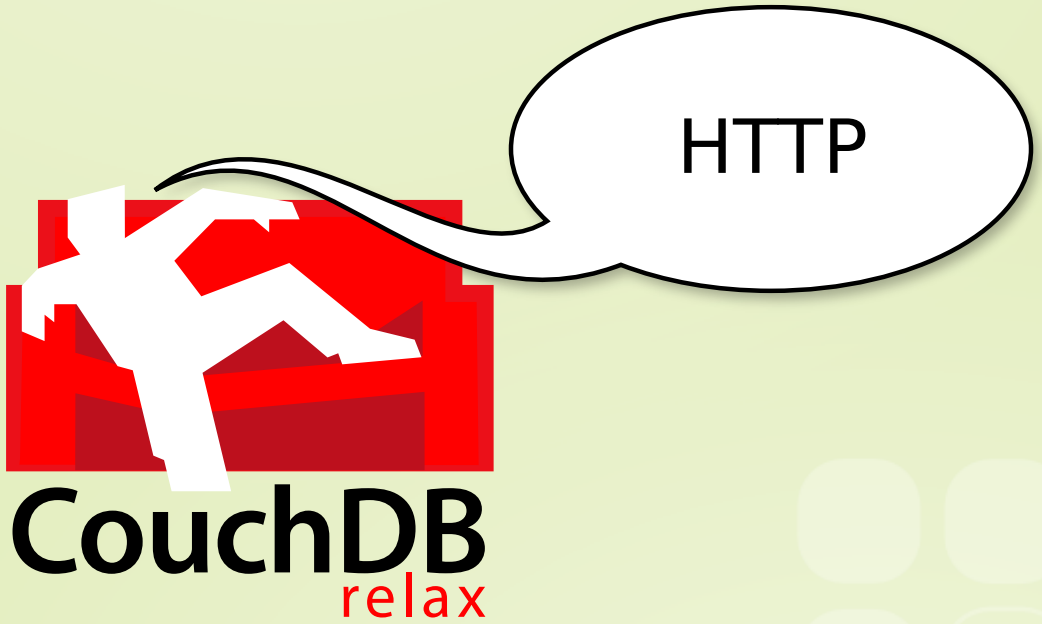


Outline

CouchDB speaks HTTP

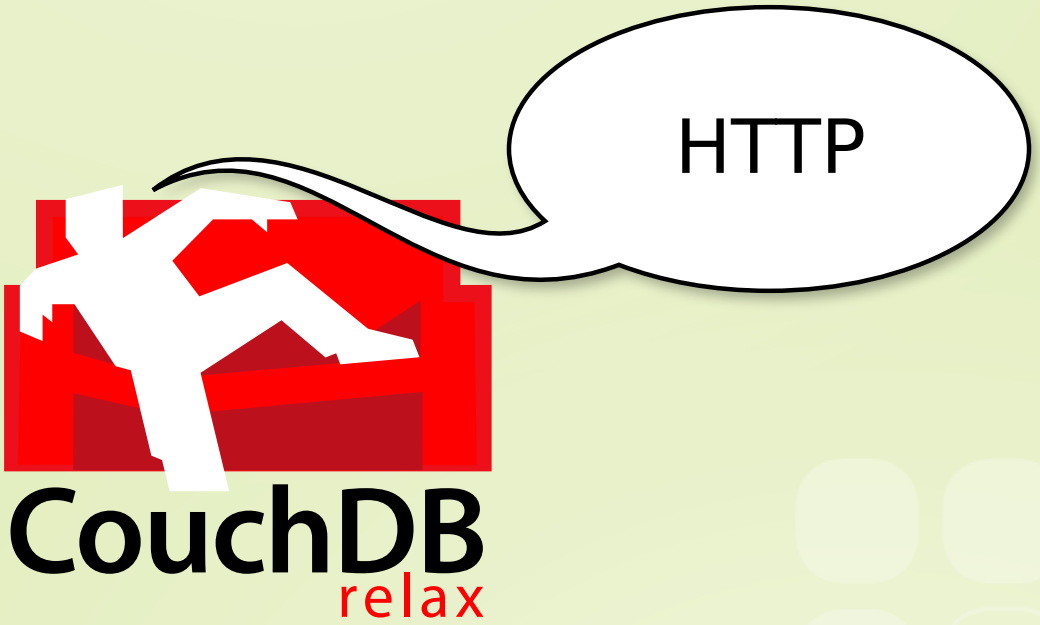


CouchDB speaks HTTP



- RESTful

CouchDB speaks HTTP



- RESTful
- JSON

Create a database

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

Create a database

```
$ curl -i -X PUT http://localhost:5984/confoo
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo
Content-Type: text/plain;charset=utf-8
Content-Length: 12

{"ok":true}
```

Re-Create a database

```
$ curl -i -X PUT http://localhost:5984/confoo
HTTP/1.1 412 Precondition Failed
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Content-Type: text/plain;charset=utf-8
Content-Length: 95

{"error":  "file_exists",
 "reason": "The database could not be created, the file already exists."
}
```

Re-Create a database

```
$ curl -i -X PUT http://localhost:5984/confoo
HTTP/1.1 412 Precondition Failed
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Content-Type: text/plain;charset=utf-8
Content-Length: 95

{"error":  "file_exists",
  "reason": "The database could not be created, the file already exists."
}
```

Re-Create a database

```
$ curl -i -X PUT http://localhost:5984/confoo
HTTP/1.1 412 Precondition Failed
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Content-Type: text/plain;charset=utf-8
Content-Length: 95

{"error":"file_exists",
"reason":"The database could not be created, the file already exists."
}
```

Get a database

```
$ curl -i -X GET http://localhost:5984/confoo
HTTP/1.1 200 OK
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Content-Type: text/plain;charset=utf-8
Content-Length: 211

{ "db_name":                 "confoo",
  "doc_count":               0,
  "doc_del_count":           0,
  "update_seq":              0,
  "purge_seq":               0,
  "compact_running":         false,
  "disk_size":               79,
  "instance_start_time":     "1299141142936296",
  "disk_format_version":     5,
  "committed_update_seq":    0
}
```

Delete a database

```
1  $ curl -i -X DELETE http://localhost:5984/confoo
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/confoo
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/confoo
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

CouchDB
- Documents

Structure

- Example wiki document

```
{ "title":      "Confoo 2011",
  "text":       "Welcome to Confoo ...",
  "creator":    "user-bar",
  "edited":     2935678239,
  "revisions": [
    { "title":    "PHPUK 2010",
      "text":     "Welcome to Confoo ...",
      "creator":  "user-foo",
      "edited":   2935678183,
      ]
    },
    ...
  ]
}
```

Structure

- Example wiki document

```
{ "title":      "Confoo 2011",
  "text":       "Welcome to Confoo ...",
  "creator":    "user-bar",
  "edited":     2935678239,
  "revisions": [
    { "title":    "PHPUK 2010",
      "text":     "Welcome to Confoo ...",
      "creator":  "user-foo",
      "edited":   2935678183,
      ]
    },
    ...
  ]
}
```

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 }
```

Attachements

- Example wiki document

```
{ "title":       "Confoo 2011",
  "creator":     "user-bar",
  "text":        "<h1>Welcome to the Confoo</h1>

                 <img src=\"confoo_2011/logo.png\" alt=\"Confoo logo\"/>
                 ...",
  "_attachments": {
    "logo.png": {
      "content_type": "image\/png",
      "stub":         true,
      "length":       42,
    }
  }
}
```

Attachements

- Example wiki document

```
1  { "title":       "Confoo 2011",
2    "creator":     "user-bar",
3    "text":        "<h1>Welcome to the Confoo</h1>
4
5  <img src=\"confoo_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 }
```

Attachements

- Example wiki document

```
{ "title":       "Confoo 2011",
  "creator":     "user-bar",
  "text":        "<h1>Welcome to the Confoo</h1>

                 <img src=\"confoo_2011/logo.png\" alt=\"Confoo logo\"/>
                 ...",
  "_attachments": {
    "logo.png": {
      "content_type": "image\/png",
      "stub":         true,
      "length":       42,
    }
  }
}
```

Create a new document

```
$ curl -i -X PUT http://localhost:5984/confoo/test
       -d '{"title": "Hello world!"}'
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo/test
Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
Content-Type: text/plain;charset=utf-8
Content-Length: 67

{ "ok":  true,
  "id":  "test",
  "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
}
```

Create a new document

```
$ curl -i -X PUT http://localhost:5984/confoo/test
       -d '{"title": "Hello world!"}'
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo/test
Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
Content-Type: text/plain;charset=utf-8
Content-Length: 67

{"ok": true,
 "id": "test",
 "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
}
```

Create a new document

```
$ curl -i -X PUT http://localhost:5984/confoo/test
       -d '{"title": "Hello world!"}'
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo/test
Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
Content-Type: text/plain;charset=utf-8
Content-Length: 67

{"ok": true,
 "id": "test",
 "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
}
```

Create a new document

```
$ curl -i -X PUT http://localhost:5984/confoo/test
       -d '{"title": "Hello world!"}'
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo/test
Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
Content-Type: text/plain;charset=utf-8
Content-Length: 67

{"ok": true,
 "id": "test",
 "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
}
```

Create a new document

```
$ curl -i -X PUT http://localhost:5984/confoo/test
       -d '{"title": "Hello world!"}'
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo/test
Etag: "1-98da4da1fd7d7f90ceb752501d5b2321"
Content-Type: text/plain;charset=utf-8
Content-Length: 67

{"ok": true,
 "id": "test",
 "rev": "1-98da4da1fd7d7f90ceb752501d5b2321"
}
```

Add an attachment

```
$ curl -i -X PUT 'http://localhost:5984/confoo/test/logo.png?rev=1-98
    da4da1fd7d7f90ceb752501d5b2321'
       --data-binary '@graphics/couchdb.png'
       -H 'Content-Type: image/png'
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo/test/logo.png
Etag: "2-d9053f096f374284332575ab6956a658"
Content-Type: text/plain;charset=utf-8
Content-Length: 67

{"ok": true,
 "id": "test",
 "rev": "2-d9053f096f374284332575ab6956a658"
}
```

Add an attachment

```
$ curl -i -X PUT 'http://localhost:5984/confoo/test/logo.png?rev=1-98
      da4da1fd7d7f90ceb752501d5b2321'
       --data-binary '@graphics/couchdb.png'
       -H 'Content-Type: image/png'
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo/test/logo.png
Etag: "2-d9053f096f374284332575ab6956a658"
Content-Type: text/plain;charset=utf-8
Content-Length: 67

{"ok":  true,
  "id":  "test",
  "rev": "2-d9053f096f374284332575ab6956a658"
}
```

Add an attachment

```
$ curl -i -X PUT 'http://localhost:5984/confoo/test/logo.png?rev=1-98
    da4da1fd7d7f90ceb752501d5b2321'
      --data-binary '@graphics/couchdb.png'
      -H 'Content-Type: image/png'
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo/test/logo.png
Etag: "2-d9053f096f374284332575ab6956a658"
Content-Type: text/plain;charset=utf-8
Content-Length: 67

{"ok": true,
 "id": "test",
 "rev": "2-d9053f096f374284332575ab6956a658"
}
```

Add an attachment

```
$ curl -i -X PUT 'http://localhost:5984/confoo/test/logo.png?rev=1-98
    da4da1fd7d7f90ceb752501d5b2321'
      --data-binary '@graphics/couchdb.png'
      -H 'Content-Type: image/png'
HTTP/1.1 201 Created
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Location: http://localhost:5984/confoo/test/logo.png
Etag: "2-d9053f096f374284332575ab6956a658"
Content-Type: text/plain;charset=utf-8
Content-Length: 67

{"ok":true,
 "id":"test",
 "rev":"2-d9053f096f374284332575ab6956a658"
}
```

Get the document back

```
$ curl -i -X GET http://localhost:5984/confoo/test
HTTP/1.1 200 OK
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Etag: "2-d9053f096f374284332575ab6956a658"
Content-Type: text/plain;charset=utf-8
Content-Length: 177

{ "_id":    "test",
  "_rev":   "2-d9053f096f374284332575ab6956a658",
  "title": "Hello world!",
  "_attachments": {
    "logo.png": {
      "content_type": "image/png",
      "revpos":       2,
      "length":       10785,
      "stub":         true
    }
  }
}
```

Get the document back

```
$ curl -i -X GET http://localhost:5984/confoo/test
HTTP/1.1 200 OK
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Etag: "2-d9053f096f374284332575ab6956a658"
Content-Type: text/plain;charset=utf-8
Content-Length: 177

{ "_id":    "test",
  "_rev":   "2-d9053f096f374284332575ab6956a658",
  "title": "Hello world!",
  "_attachments": {
    "logo.png": {
      "content_type": "image/png",
      "revpos":       2,
      "length":       10785,
      "stub":         true
    }
  }
}
```

Get the document back

```
$ curl -i -X GET http://localhost:5984/confoo/test
HTTP/1.1 200 OK
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Etag: "2-d9053f096f374284332575ab6956a658"
Content-Type: text/plain;charset=utf-8
Content-Length: 177

{ "_id":    "test",
  "_rev":   "2-d9053f096f374284332575ab6956a658",
  "title": "Hello world!",
  "_attachments": {
    "logo.png": {
      "content_type": "image/png",
      "revpos":       2,
      "length":       10785,
      "stub":         true
    }
  }
}
```

Get the document back

```
$ curl -i -X GET http://localhost:5984/confoo/test
HTTP/1.1 200 OK
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
Etag: "2-d9053f096f374284332575ab6956a658"
Content-Type: text/plain;charset=utf-8
Content-Length: 177

{ "_id":    "test",
  "_rev":   "2-d9053f096f374284332575ab6956a658",
  "title": "Hello world!",
  "_attachments": {
    "logo.png": {
      "content_type": "image/png",
      "revpos":       2,
      "length":       10785,
      "stub":         true
    }
  }
}
```

Get the attachment back

```
$ curl -i -X GET http://localhost:5984/confoo/test/logo.png
HTTP/1.1 200 OK
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
ETag: "2-d9053f096f374284332575ab6956a658"
Content-Type: image/png
Content-Length: 10785
Content-Encoding: identity

<PNG data>
```

Get the attachment back

```
$ curl -i -X GET http://localhost:5984/confoo/test/logo.png
HTTP/1.1 200 OK
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
ETag: "2-d9053f096f3742843325 75ab6956a658"
Content-Type: image/png
Content-Length: 10785
Content-Encoding: identity

<PNG data>
```

Get the attachment back

```
$ curl -i -X GET http://localhost:5984/confoo/test/logo.png
HTTP/1.1 200 OK
Server: CouchDB/1.0.1 (Erlang OTP/R13B)
ETag: "2-d9053f096f3742843325 75ab6956a658"
Content-Type: image/png
Content-Length: 10785
Content-Encoding: identity

<PNG data>
```

Attachments

- Attach to a document
  - HTML
  - JavaScript
  - Images

Attachments

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

Outline

Rewrites

- Allows to rewrite URLs
- Specified in design documents

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

```
curl -X
http://localhost:5984/db/_design/app/_rewrite/
-H "Accept: text/html"
http://localhost:5984/db/blog/index.html
```

Rewrites

- Allows to rewrite URLs
- Specified in design documents

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

Rewrites

- Allows to rewrite URLs
- Specified in design documents

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

Rewrites

- Allows to rewrite URLs
- Specified in design documents

```
1  { "_id":      "_design/app",
2    "rewrites": [
3      { "from": "/blog",
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  }
```

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
  - Especially useful for views, will be covered later

```
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:

```
$ cat /etc/couchdb/local.ini
[vhosts]
myhost:5984 = /db/_design/app/_rewrite
```

```
http://myhost:5984/blog
```

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

- A rewrite rule for "/" is also possible, of course...

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  [...]
```

vHosts

- Those are still pretty ugly URLs...
- vHosts to the rescue:

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

- Requested as:
  `http://myhost:5984/blog`
- Rewrites to:
  `http://localhost:5984/db/blog/index.html`

Rewrite rule for "/" is also possible, of course...

vHosts

- Those are still pretty ugly URLs...
- vHosts to the rescue:

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

- Requested as:
  `http://myhost:5984/blog`
- Rewrites to:
  `http://localhost:5984/db/blog/index.html`
  - A rewrite rule for `"/"` is also possible, of course...

Setting up a CouchApp

- Create vHost

Setting up a CouchApp

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

Setting up a CouchApp

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

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

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

Handling consistency

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

Handling consistency

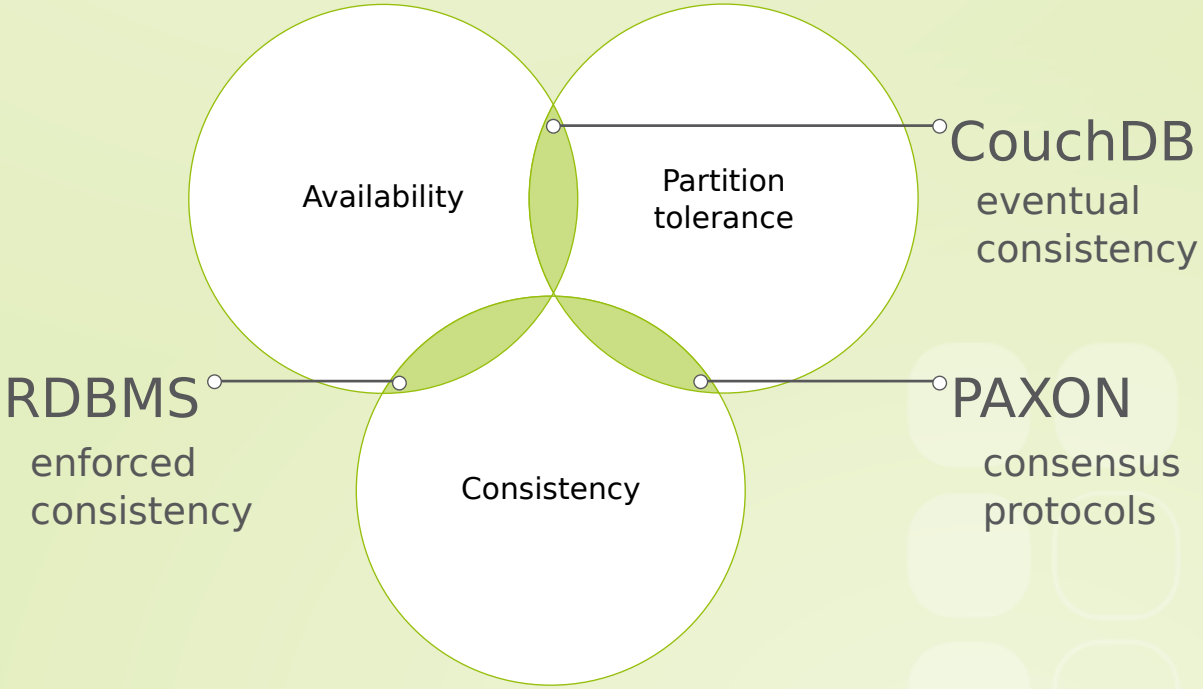
- Two possibilities for handling distributed data:
  - Force to keep all servers in sync
  - Be partition tolerant and merge data, if required
    - *Eventually* be consistent

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

- Replication is trivial

```
$ curl -X POST http://localhost:5984/_replicate \
    -H 'Content-Type: application/json' \
    -d '{"source":"confoo",\
          "target":"http://user:pass@192.168.1.3:5984/confoo"}'

{ "ok":                  true,
  "no_changes":          true,
  "session_id":          "73d69e7b5cdaea059e55ed1db7802151",
  "source_last_seq":     141,
  "history": [ {
    ...
  } ]
}
```

source and target can be any combination of remote and local

Replication

- Replication is trivial

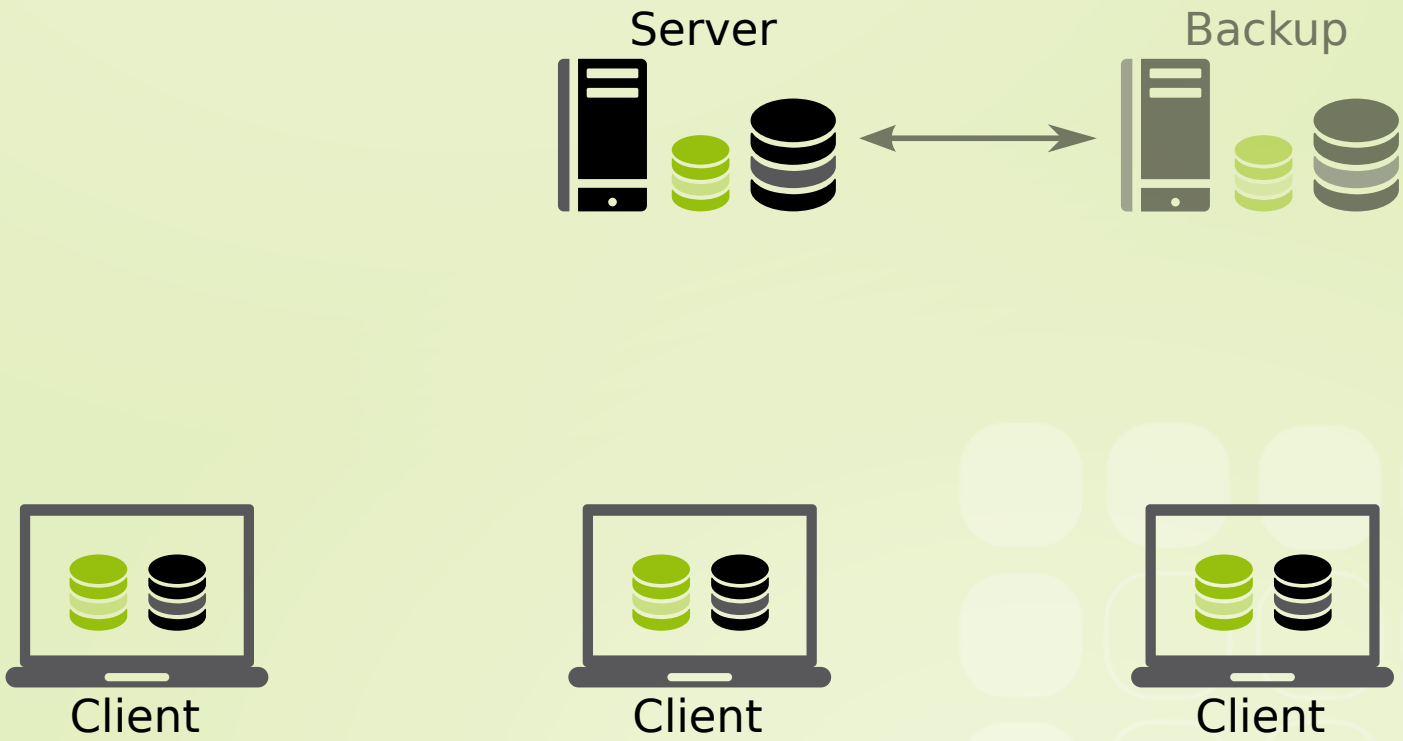
```
$ curl -X POST http://localhost:5984/_replicate \
    -H 'Content-Type: application/json' \
    -d '{"source":"confoo",\
          "target":"http://user:pass@192.168.1.3:5984/confoo"}'

{ "ok":                  true,
  "no_changes":          true,
  "session_id":          "73d69e7b5cdaea059e55ed1db7802151",
  "source_last_seq":     141,
  "history": [ {
    ...
  } ]
}
```

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

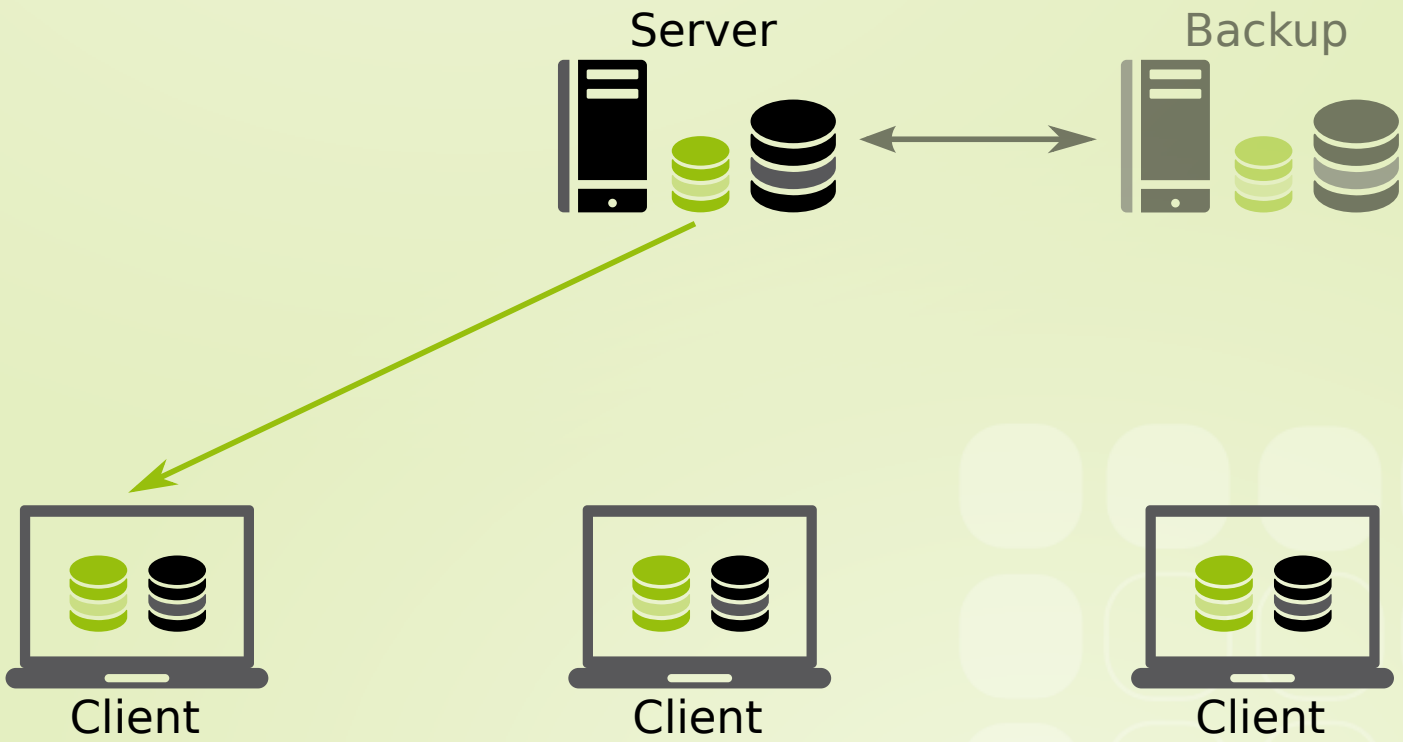
Demo

Outline

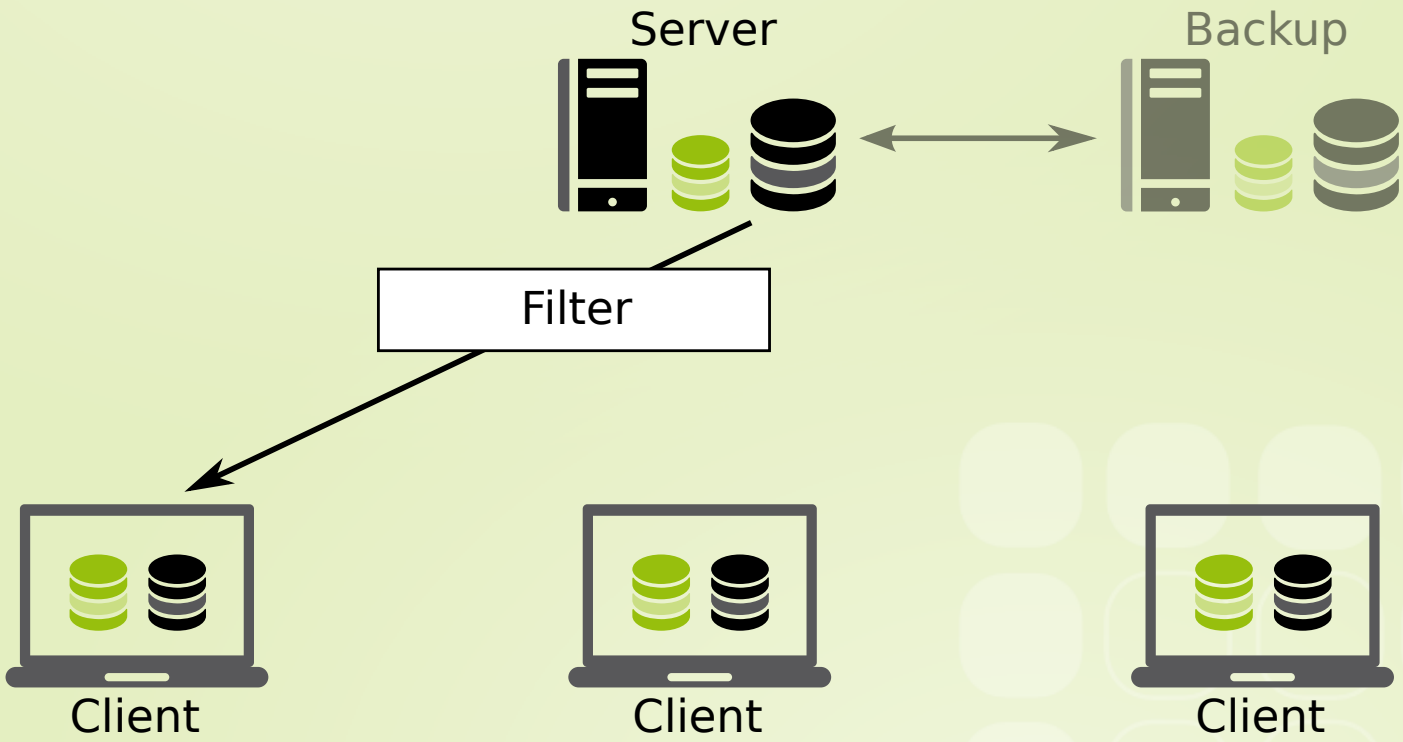
Filtered Replication



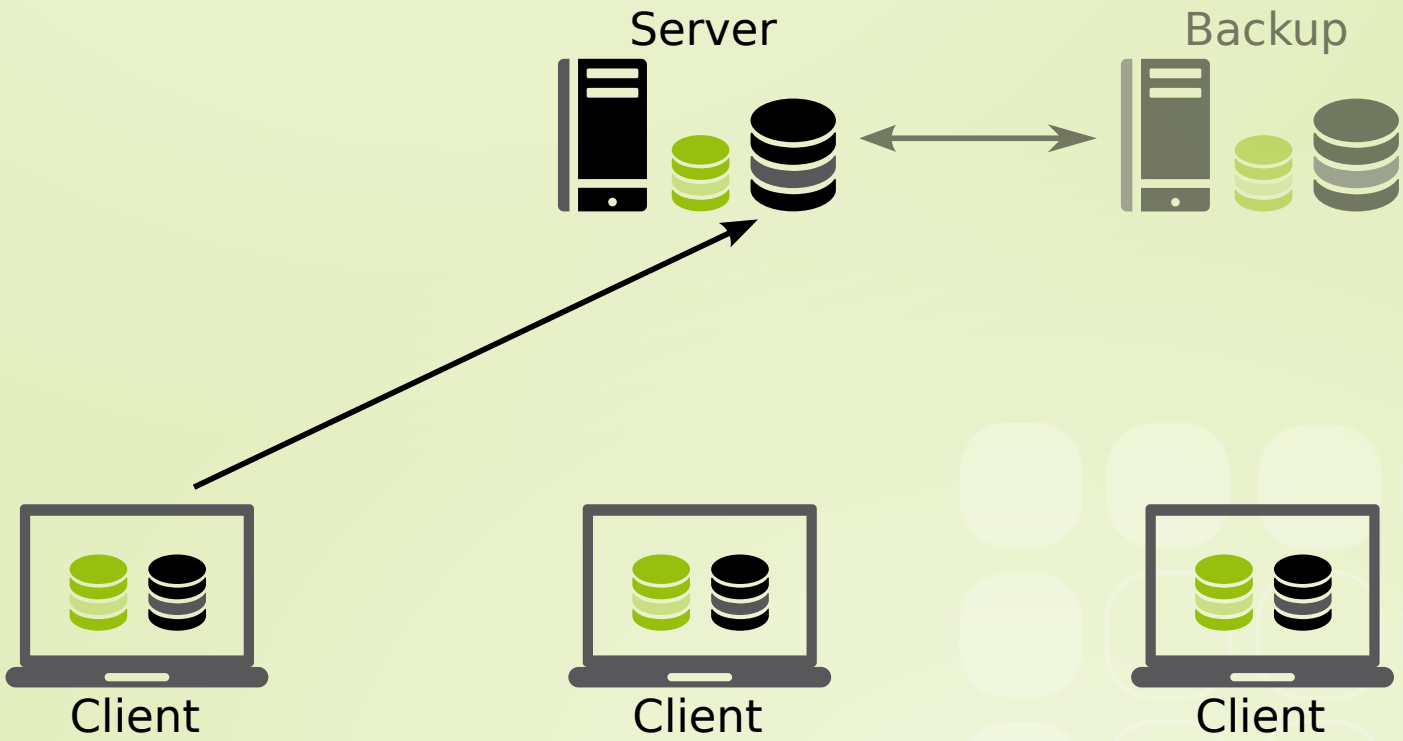
Filtered Replication



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 }
```

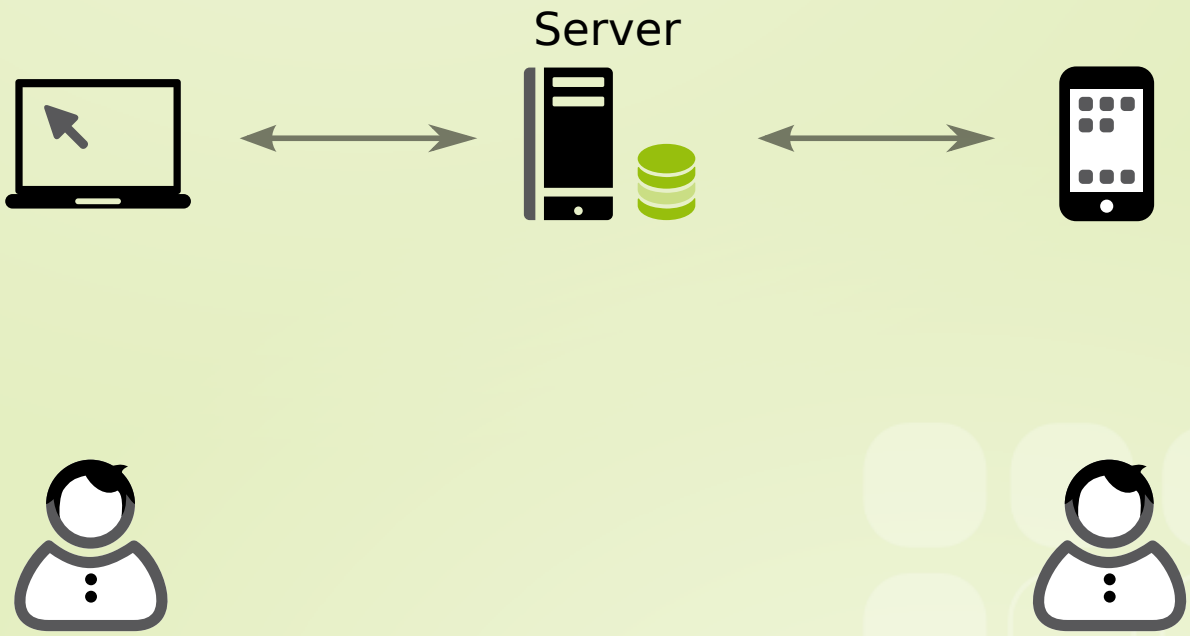
Filtered Replication

- Usage during replication

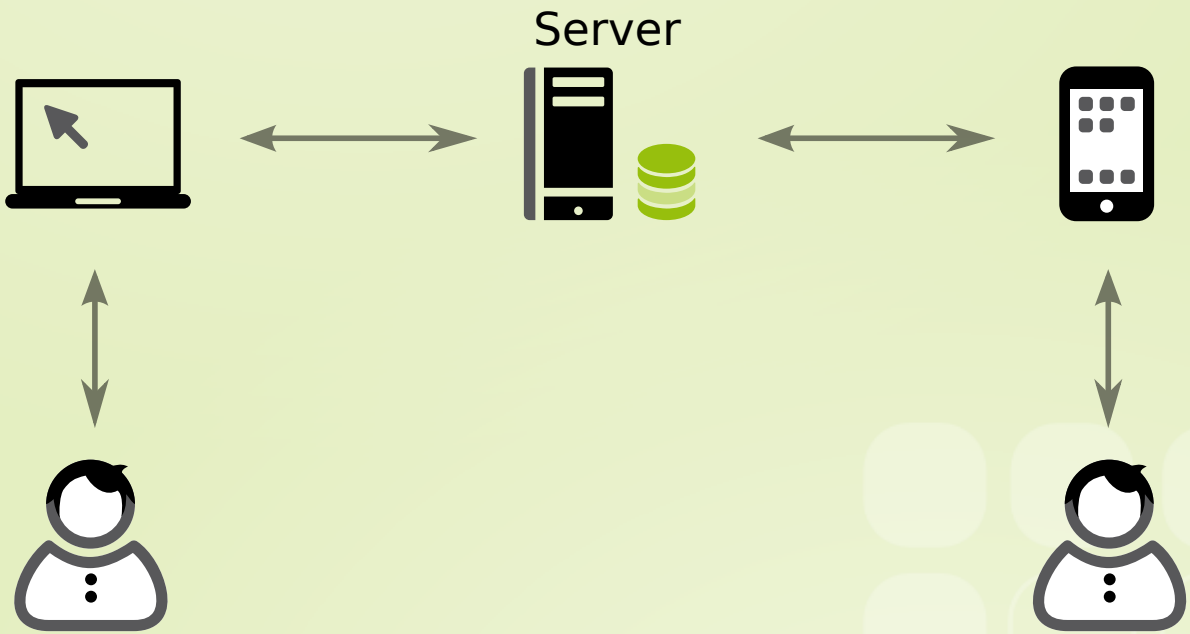
```
$ curl -X POST http://localhost:5984/_replicate \
    -H 'Content-Type: application/json' \
    -d '{"source":"http://user:pass@192.168.1.3:5984/confoo",\
          "target":"confoo",\
          "filter":"app/for_user"}'...
```

Outline

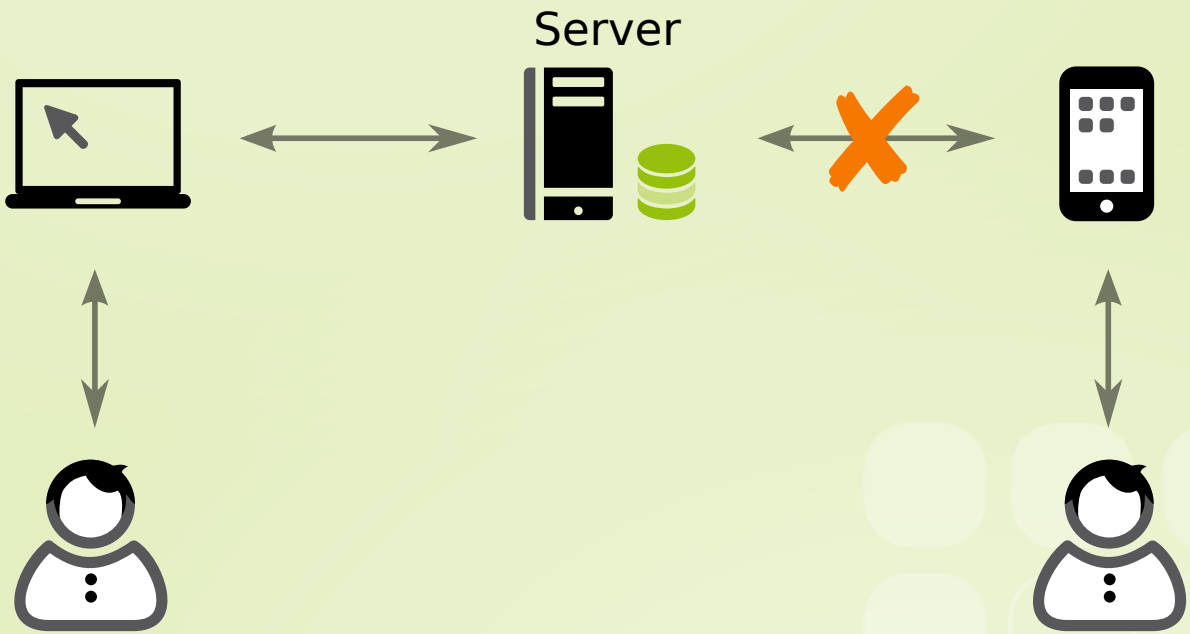
Document conflict handling



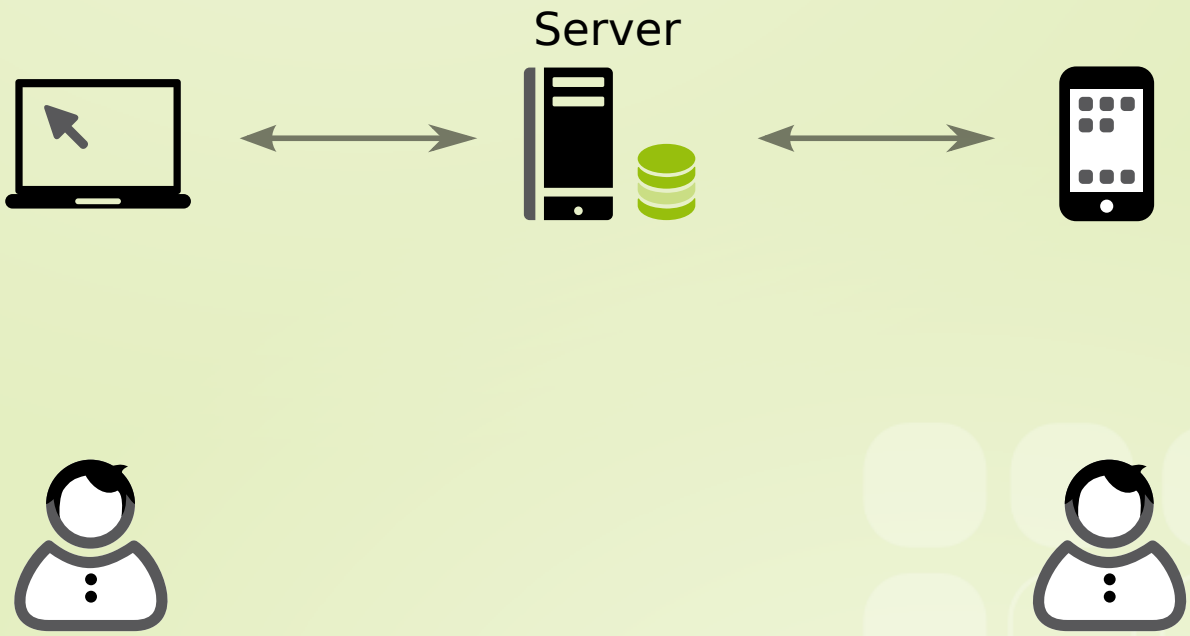
Document conflict handling



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

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

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

Outline

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/2778`

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.