

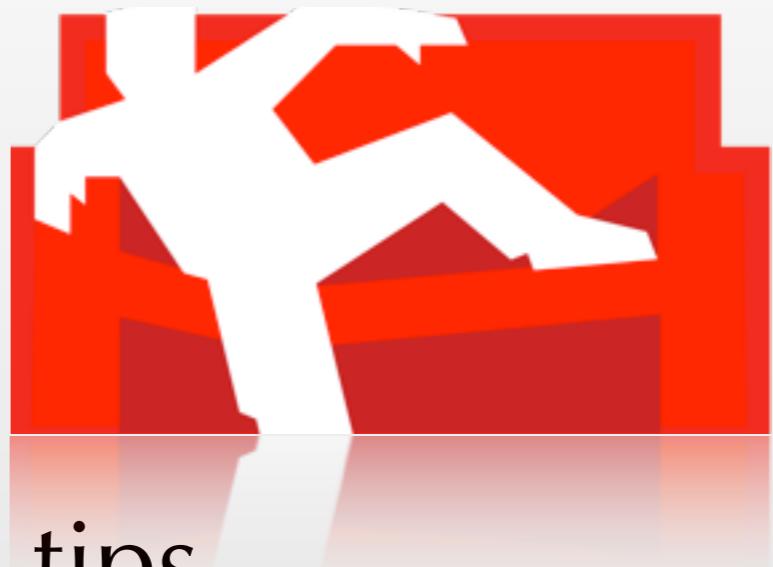


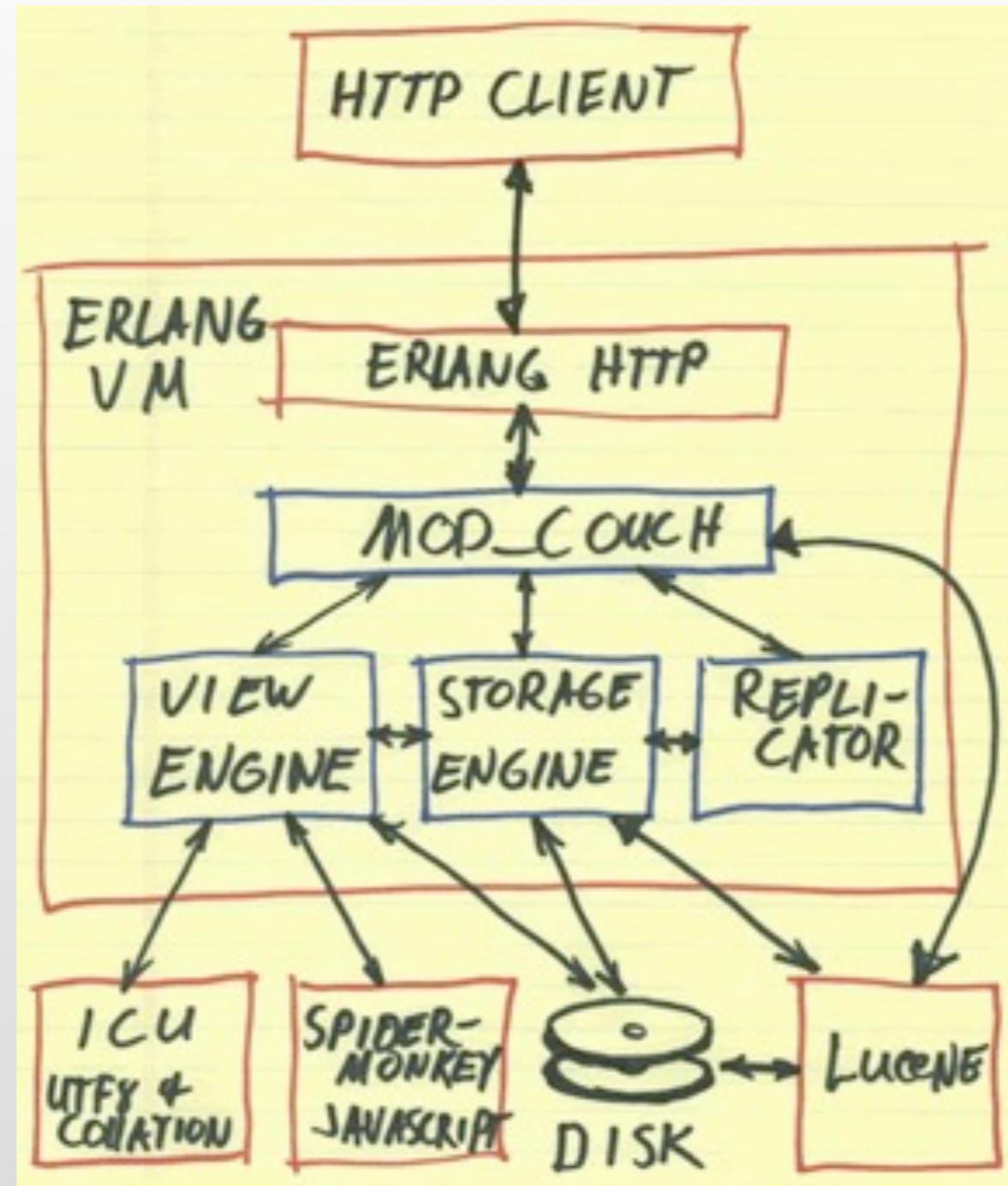
Apache  
**CouchDB**

# Relax

# Lightning Talks

- How you're using Couch
- How to use documents
- View algorithms and query tips
- How Couch is relaxing
- Other relaxing topics





# Hovercraft

- Direct Erlang access to CouchDB
- Simple API (wrapper around internal functions)
- Runs in the same node as CouchDB

# Wheeeee!



# Tour de source

```
-export([
    create_db/1,
    delete_db/1,
    db_info/1,
    open_doc/2,
    save_doc/2,
    delete_doc/2,
    start_attachment/3,
    next_attachment_bytes/1,
    query_view/5 % coming soon
]).
```

# http api

- couch\_httpd\_\*.erl
- (and) couch\_rep.\*\_erl
- couch\_httpd.erl
  - \*\_db.erl
  - \*\_view.erl
  - \*\_show.erl

# HTTP PUT document

handle all requests by looking up a URL handler

```
handle_request(MochiReq, UrlHandlers, DbUrlHandlers, DesignUrlHandlers) ->
    RawUri = MochiReq:get(raw_path),
    {"/" ++ Path, _, _} = mochiweb_util:urlsplit_path(RawUri),

    DefaultFun = fun couch_httpd_db:handle_request/1,
    HandlerFun = couch_util:dict_find(HandlerKey, UrlHandlers, DefaultFun),

    {ok, Resp} = try
        HandlerFun(HttpReq#httpd{user_ctx=AuthenticationFun(HttpReq)})
    catch
        send_error(HttpReq, Error)
    end,
    {ok, Resp}.
```

# HTTP PUT document

## handle db requests

```
% Database request handlers
handle_request(#httpd{path_parts=[DbName|RestParts],method=Method,
    db_url_handlers=DbUrlHandlers}=Req)->
    case {Method, RestParts} of
        {'PUT', []} ->
            create_db_req(Req, DbName);
        {'DELETE', []} ->
            delete_db_req(Req, DbName);
        {_, []} ->
            do_db_req(Req, fun db_req/2);
        {_, [SecondPart|_]}->
            Handler = couch_util:dict_find(SecondPart, DbUrlHandlers, fun db_req/2),
            do_db_req(Req, Handler)
    end
```

# HTTP PUT document

open the db and hand it to the db request function

```
do_db_req(#httpd{user_ctx=UserCtx,path_parts=[DbName|_]}=Req, Fun) ->
    case couch_db:open(DbName, [{user_ctx, UserCtx}]) of
        {ok, Db} ->
            try
                Fun(Req, Db)
            after
                couch_db:close(Db)
            end;
        Error ->
            throw(Error)
    end.
```

# HTTP PUT document

make a doc request when a the path ends with a DocId

```
db_req(#httpd{path_parts=[_, DocId]}=Req, Db) ->  
    db_doc_req(Req, Db, DocId);
```

the doc request on a PUT

```
db_doc_req(#httpd{method='PUT'}=Req, Db, DocId) ->  
    Location = absolute_uri(Req, "/" ++ ?b2l(Db#db.name) ++ "/" ++ ?b2l(DocId)),  
    update_doc(Req, Db, DocId, couch_httpd:json_body(Req),  
              [{"Location", Location}]);
```

# db & file module

- from the metal
  - couch\_file
  - term\_to\_binary
  - couch\_stream
  - couch\_btreet
  - attachments

# view module

- couch\_httpd\_view
- couch\_view
- couch\_view\_group
- couch\_view\_updater

# replication module

- concurrent bidirectional
- triggered
- resilient
- db\_update\_notification\*
  - (deprecated for)  
\_changes Comet API

# batch=ok

- Just a performance optimization
- I wanted an excuse to return the HTTP 202 Accepted response code
- Fun code, let's take a look

Apache CouchDB: Futon Utility Client

http://127.0.0.1:5984/\_utils/

Most Visited Smart Bookmarks Getting Started Latest Headlines Make it Play http://jchris.mfdz.co... Apache CouchDB Fu... ReCSS Share on Tumblr

Amazon... ElasticFox Apache... Apache... domU... View D... Apac... O Http...ent CouchD... JChris's... ▾

Overview Test Suite

Run All  Reload

Note: Each of the tests will block the browser. If the connection to your CouchDB server is slow, running the tests will take some time, and you'll not be able to do much with your browser while a test is being executed.

Name	Status	Elapsed Time	Details
basics	✓ success	1230ms	
conflicts	✓ success	162ms	
recreate_doc	✓ success	803ms	
copy_move_doc	✓ success	1574ms	
uuids	✓ success	114ms	
bulk_docs	✓ success	185ms	
lots_of_docs	✓ success	1728ms	
reduce	✓ success	6422ms	
reduce_false	✓ success	114ms	
multiple_rows	✓ success	170ms	
large_docs	✓ success	282ms	
utf8	✓ success	245ms	
attachments	✓ success	415ms	
content_negotiation	✓ success	71ms	
design_docs	✓ success	3008ms	
view_collation	@ running...		
view_conflicts	not run		
view_errors	not run		
view_include_docs	not run		

CouchDB relax

Tools Overview Replicator Configuration Test Suite Recent Databases

Futon on Apache CouchDB 0.9.0 Dev-including

## Recently...

### Peer Committee

2 days ago

I don't know the title at the moment, but I have an idea for a simple data structure across better nodes (not in a cluster). A replication peer, which tracks all known replicates, and which has successfully replicated evidence until it finds a new node. A can, for each update, provide to the user a list of remote nodes that have knowledge of the update. I think ...

sep, July, solar, drone, laptop,  
music, news, life, photo gallery,  
you, config, C++, code example,  
code

## Old News from a New Framework

2 days ago

String is a Scala API service for publishing Akka ConceptDB apps. The second part of this note is Old News: By taking advantage of these characteristics this module is able to publish data to a central store with traditional relational-persistent abstractions. Multiple exceptions include the ability to update dynamic text via JSON, or return results from reading ...

### Deployed So far

2 days ago

For more information about this cool software, visit the So far source code on GitHub I plan to run my blog from this software, and I've got a few other projects in the pipeline. Also, don't forget about the Twitter client. Same great software, better new address....

## Talk Announcements

4 weeks ago

My conference calendar is actually filling up with exciting ConceptDB talks. Here's the full list of them, just so you know: A new episode of the ConceptDB podcast is out. Download the app or subscribe to the RSS feed ConceptDB Talks Feed is in progress. Demos of ConceptDBs at PDX is (Lam) The Portland JavaScalite Application Group had its first meeting ...

## More I often

2 weeks ago

I often I often ...

```
function(doc, req) {
// !json templates.post
// !json blog
// !code helpers.template
// !code helpers.couchapp
// log(req.headers.Accept);

// we only show html
return template(templates.post, {
  title : doc.title,
  blogName : blog.title,
  post : doc.html,
  date : doc.created_at,
  author : doc.author,
  assets : assetPath(),
  editPostPath : showPath('edit', doc._id),
  index : listPath('index','recent-posts',{descending:true, limit:8})
});
}
```



A screenshot of a web browser displaying a blog post titled "Hello World For Real This Time". The browser's address bar shows the URL [http://jchris.net/dr/\\_show/sota/post>Hello-World-For-Real-This-Time](http://jchris.net/dr/_show/sota/post>Hello-World-For-Real-This-Time). The page content includes the post title, a placeholder text block, and a comment section with four entries:

- by J Chris A, 5 weeks ago  
With gravatar comments!
- by Jason Davies, 5 weeks ago  
SRSLY!
- by Jason Watkins, 5 weeks ago  
I AM INTRIGUED BY YOUR IDEAS AND WOULD LIKE TO SUBSCRIBE TO YOUR NEWSLETTER
- by J Chris A, 5 weeks ago  
@Jason feeds are on the way

```

1  function(head, row, req) {
2    // ljson templates.index
3    // ljson blog
4    // lcode helpers.couchapp
5    // lcode helpers.template
6    // log(req.headers.Accept);
7    var indexPath = listPath("index", "recent-posts", {descending:true, limit:8});
8    var feedPath = listPath("index", "recent-posts", {descending:true, limit:8, format:"atom"});
9    return respondWith(req, {
10      html : function() {
11        if (head) {
12          return template(templates.index.head, {
13            title : blog.title,
14            newPostPath : showPath("edit"),
15            index : indexPath,
16            assets : assetPath()
17          });
18        } else if (row) {
19          var post = row.value;
20          return template(templates.index.row, {
21            title : post.title,
22            summary : post.summary,
23            date : post.created_at,
24            link : showPath('post', row.id),
25            assets : assetPath()
26          });
27        } else {
28          return template(templates.index.tail, {
29            assets : assetPath()
30          });
31        }
32      },
33      atom : function() {
34        // with first row in head you can do updated.
35        if (head) {
36          var f = <feed xmlns="http://www.w3.org/2005/Atom"/>;
37          f.title = blog.title;
38          f.id = makeAbsolute(req, indexPath);
39          f.link.href = makeAbsolute(req, feedPath);
40          f.link.rel = "self";
41          f.generator = 'Sofa on CouchDB';
42          f.updated = new Date().rfc3339();
43          return {body:f.toXMLString().replace(/\<\!Feed\!\>/, '')};
44        } else if (row) {
45          var entry = <entry/>;
46          entry.id = makeAbsolute(req, '/' + encodeURIComponent(req.info.db_name) + '/' + encodeURIComponent(row.value._id));
47          entry.title = row.value.title;
48          entry.content = row.value.summary;
49          entry.content.type = 'html';
50          entry.updated = new Date(row.value.created_at).rfc3339();
51          entry.author = <author>-name-<row.value.author>-/name-</author>;
52          entry.link.href = makeAbsolute(req, showPath('post', row.id));
53          entry.link.rel = "alternate";
54          return {body:entry};
55        } else {
56          return {body : "</feed>"};
57        }
58      }
59    });
60  };

```

## Daytime Running Lights

web, vinyl, Scala, quote, photo, music, lorem, life, hello world, dev, couchdb, Couch, code couchapp, code

## Recently...

### Peer Commit

7 days ago

I don't know the literature, but I have an idea for a shared data-space across peer nodes (not in a cluster). A replication helper, which tracks all known replicas, and their last successfully replicated sequence num. In this way, Node A can, for each update, provide to the user a list of remote nodes that have knowledge of the update. I think t...

### Old News from a New Framework

12 days ago

Sling is a Scala app server for building Ajax CouchDB apps. The second half of this quote is Old News: By taking advantage of these characteristics pure-CouchDB applications can reach near parity with traditional server-heavy approaches. Notable exceptions include the ability to render dynamic text that isn't JSON, or return results that requi...

### Deployed Sofa

16 days ago

For more information about this blog software, visit the Sofa source code on Github I plan to run my blog from this software, and I've got a few other projects in the pipeline. Also, don't forget about the Twitter client. Same great software, great new address....

### Talk Announcements

4 weeks ago

My conference calendar is rapidly filling up with exciting CouchDB things. Here's the full list of them, but also: A new episode of the CouchDB podcast is out. Download the mp3 or subscribe to the RSS feed CouchDB Talks Feb 25 in Portland. Demoing CouchApps at PDX.js (7pm) The Portland JavaScript Admirers Group had its first meetin...

### More Lorem

5 weeks ago

Lorem Lorem Lorem ...

Apache CouchDB – Futon: View Document

[http://127.0.0.1:5984/\\_utils/document.html?sofa-hello/\\_design/sofa@tabular](http://127.0.0.1:5984/_utils/document.html?sofa-hello/_design/sofa@tabular)

Smart Bookmarks ReCSS Make it Play <http://jchris.mfdz.co...> Apache CouchDB: Fu... Share on Tumblr ChartForPage MemColor

Overview > sofa-hello > \_design/sofa

Save Document  Add Field  Upload Attachment  Delete Document

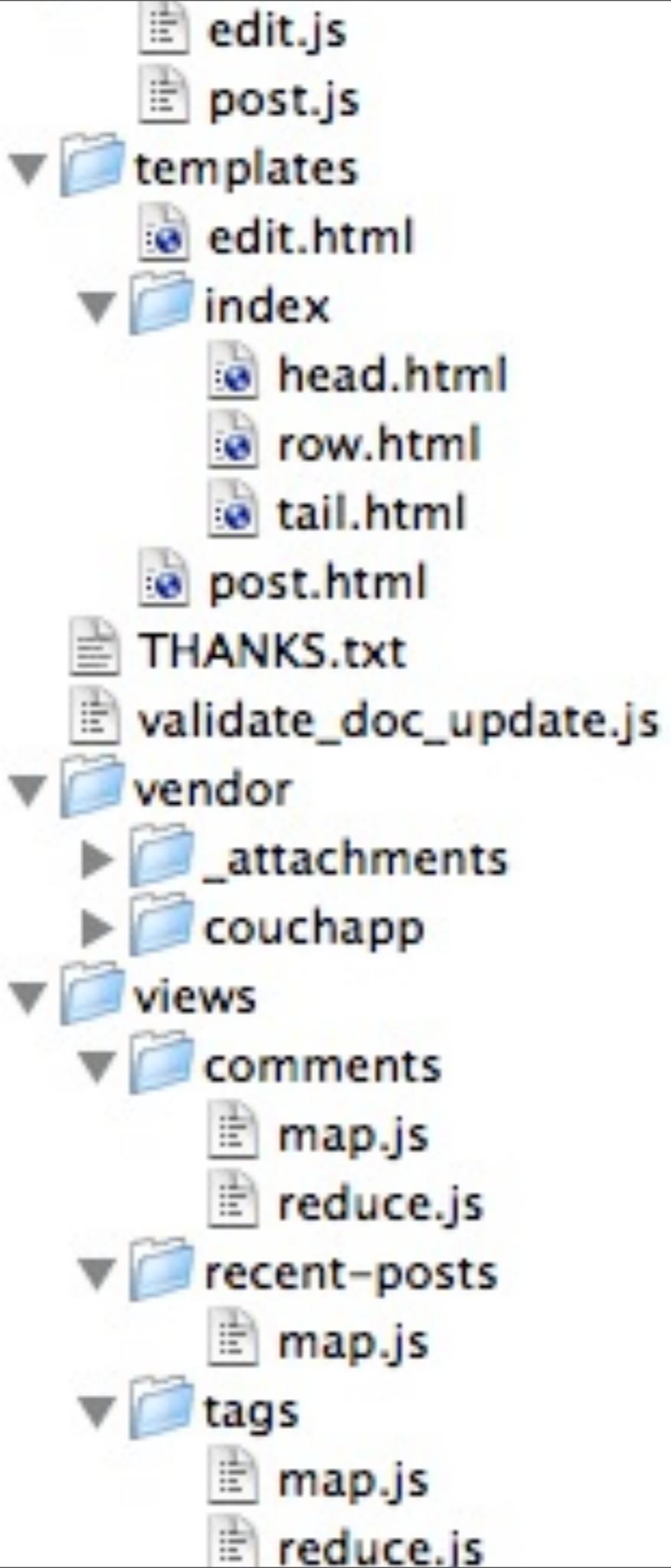
Fields  Source

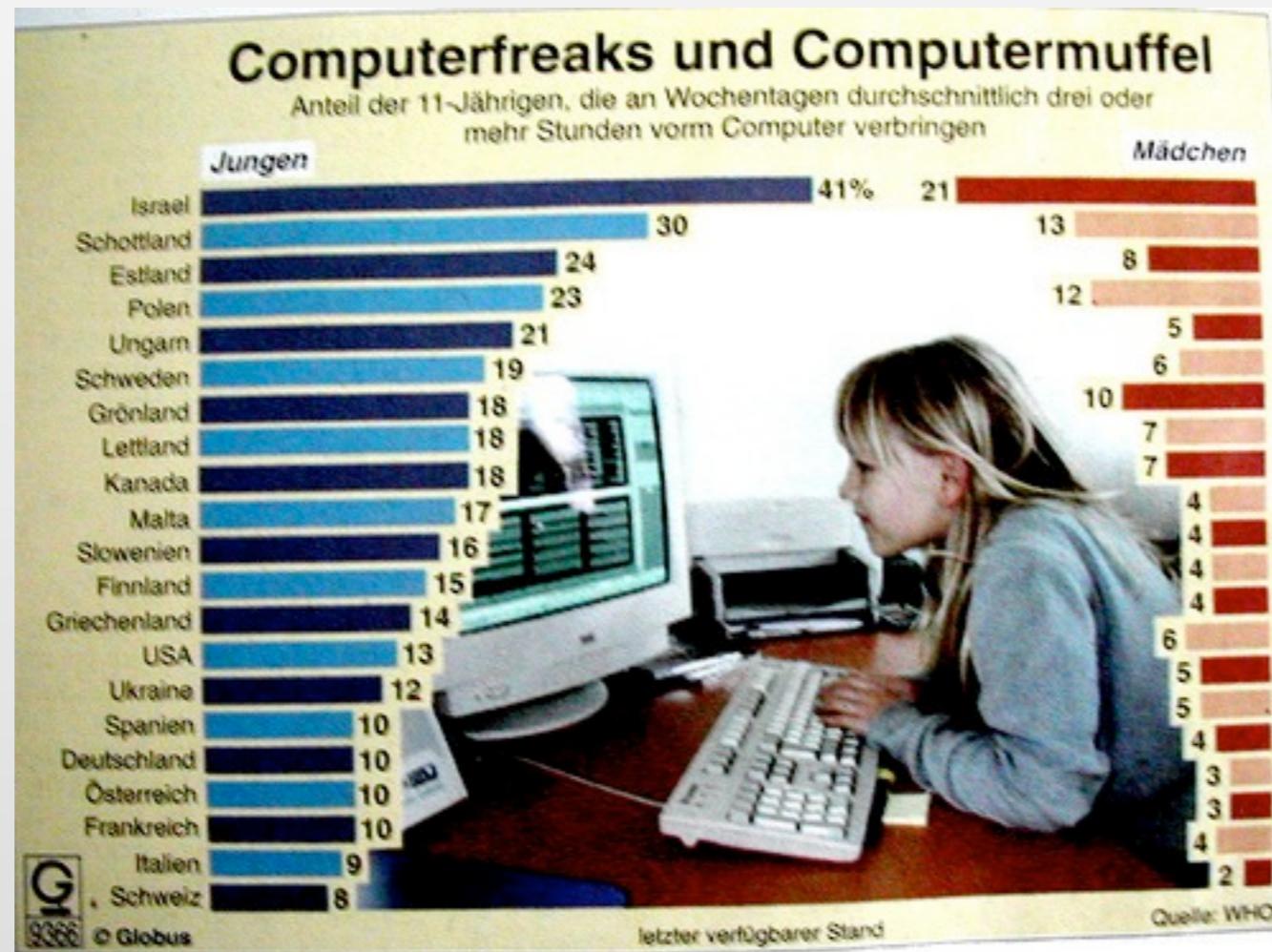
Field	Value
_id	sofa-hello
_rev	1
_attachments	<ul style="list-style-type: none"> <li><input type="checkbox"/> <a href="#">blog.js</a> 1.8 KB, application/javascript</li> <li><input type="checkbox"/> <a href="#">showdown-licensee.txt</a> 1.8 KB, text/plain</li> <li><input type="checkbox"/> <a href="#">showdown.js</a> 9.5 KB, application/javascript</li> <li><input type="checkbox"/> <a href="#">tests.js</a> 1.3 KB, application/javascript, <a href="#">open in test runner</a></li> <li><input type="checkbox"/> <a href="#">jquery.scrollTo.js</a> 2.0 KB, application/javascript</li> <li><input type="checkbox"/> <a href="#">account.html</a> 2.5 KB, text/html</li> <li><input type="checkbox"/> <a href="#">textile.js</a> 3.3 KB, application/javascript</li> <li><input type="checkbox"/> <a href="#">nd5.js</a> 8.4 KB, application/javascript</li> <li><input type="checkbox"/> <a href="#">THANKS.txt</a> 236 bytes, text/plain</li> <li><input type="checkbox"/> <a href="#">LICENSE.txt</a> 9.5 KB, text/plain</li> <li><input type="checkbox"/> <a href="#">vendor/cochapp/jquery.couchapp.js</a> 6.8 KB, application/javascript</li> <li><input type="checkbox"/> <a href="#">SCREEN.css</a> 1.7 KB, text/css</li> </ul>
README	<p><input type="checkbox"/> <a href="#">README</a></p> <p>File README contains instructions for getting up and running with couchapp and couchdb. It should provide an example for anyone to run their own couchapp, either through the command line or via the couchdb web interface. It also provides some basic information about couchapp and couchdb, and how they relate to each other. It also lists some common errors and how to fix them.</p>
THANKS	<p><input type="checkbox"/> <a href="#">THANKS</a></p> <p>File THANKS contains a list of people who have contributed to couchapp. It is intended to recognize people who have helped out, and to thank them for their work.</p>
blog	<p><input type="checkbox"/> <a href="#">blog</a></p> <p>File blog contains a list of posts from the couchapp blog.</p>
lists	<p><input type="checkbox"/> <a href="#">lists</a></p> <p>File lists contains a list of couchdb lists.</p>
shows	<p><input type="checkbox"/> <a href="#">shows</a></p> <p>File shows contains a list of couchdb shows.</p>
templates	<p><input type="checkbox"/> <a href="#">templates</a></p> <p>File templates contains a list of couchdb templates.</p>
validate_doc_update	<p><input type="checkbox"/> <a href="#">validate_doc_update</a></p> <p>File validate_doc_update contains a validate function for couchdb documents.</p>
vendor	<p><input type="checkbox"/> <a href="#">vendor</a></p> <p>File vendor contains a list of couchdb vendor files.</p>
views	<p><input type="checkbox"/> <a href="#">views</a></p> <p>File views contains a list of couchdb views.</p>

Showing revision 88 of 88

[← Previous Version](#) | [Next Version →](#)

```
$ couchapp clone http://jchrisa.net/drl/_design/sofa  
Cloning sofa to sofa...  
brindle 16:41:33 ~/code/tmp jchris  
$ cd sofa  
brindle 16:41:41 ~/code/tmp/sofa jchris  
$ couchapp push blog-database  
Pushing CouchApp in /Users/jchris/code/tmp/sofa to  
http://127.0.0.1:5984/blog-database/_design/sofa  
Visit your CouchApp here:  
http://127.0.0.1:5984/blog-database/_design/sofa/_  
brindle 16:41:47 ~/code/tmp/sofa jchris  
$ █
```





CC-BY-SA <http://www.flickr.com/photos/kelleys/492253912/>

## Crazy Train!

vinyl, Vidoo, upgrade, travel, talk, sysadmin, style, standalone, quote, photo, music, lorem, London, life, implementation, hello world, futurist, fitness, file-system, Distributed identity, CouchRest, CouchDB, CouchApp, code, bookmarklet, Apache, admin

## Recently...

### ApacheCon EU

yesterday

I'm sitting in the hotel lobby bar with Benoit and Jan, talking about CouchDB and CouchApp. The conference looks like it will be a lot of hallway conversations....

### Fun in London

4 days ago

Amy and I have been in London since Thursday morning. I think the first day I've been jet-lag free. And now we get on the plan to go to the Notable events (links to tweets): Meeting with friends and partners, Training and Consulting Meetings with top developers at the IBM UK office, Last FM, CouchDB meetup ...

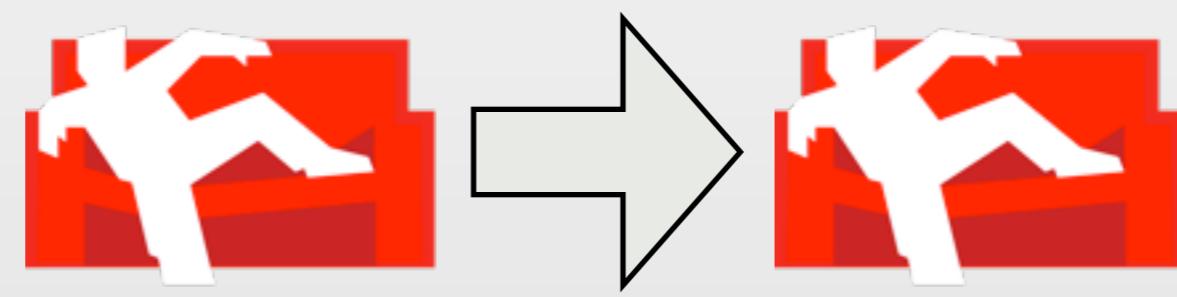
### Team Fitness

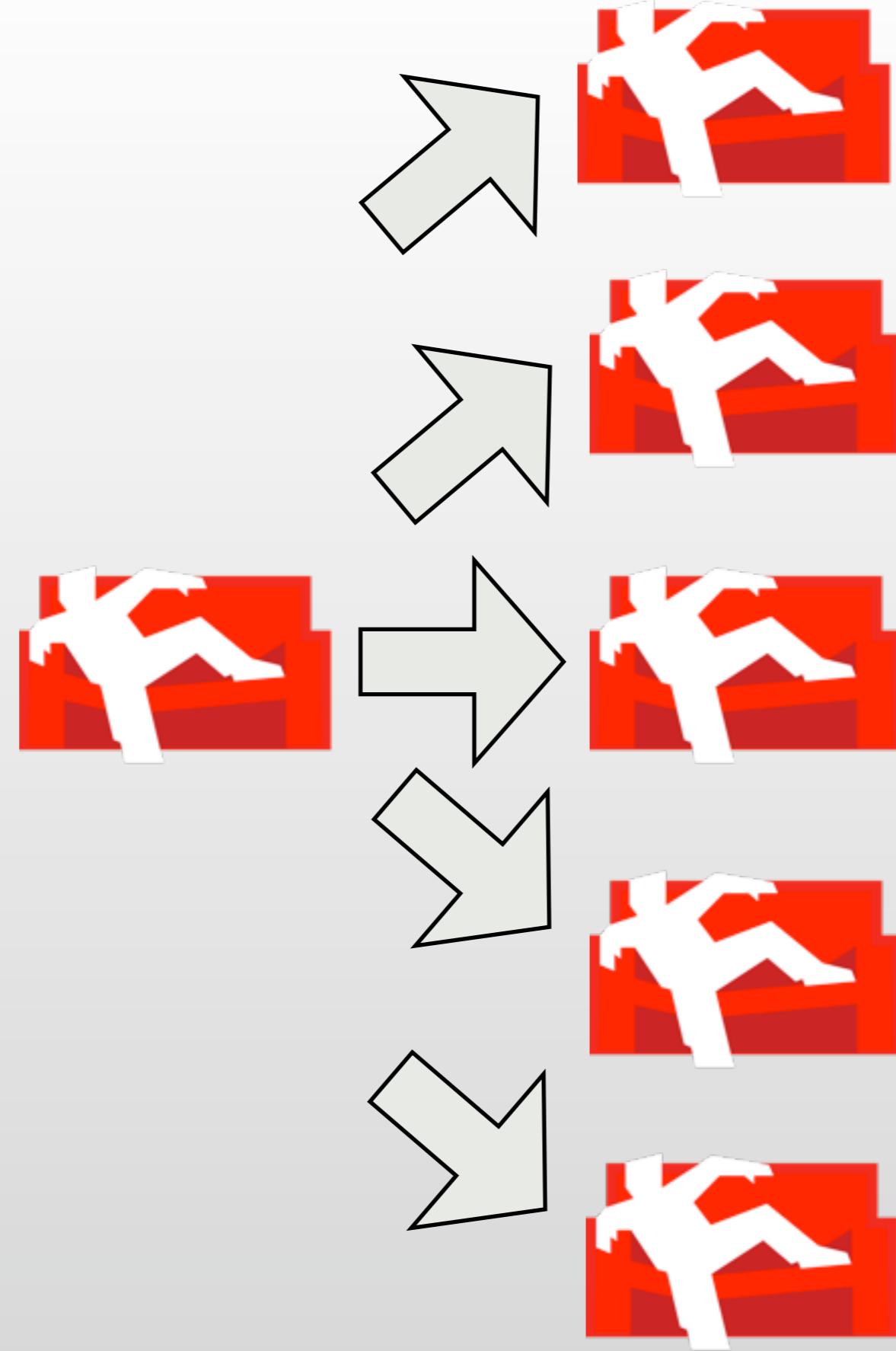
4 days ago

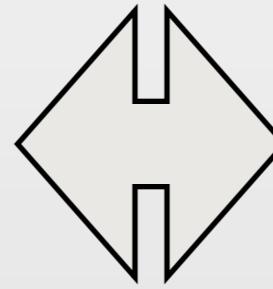
From my comment on Sam Ruby's blog: Iâ€™ve never been an athlete before, but in the last year my wife got me started with her â€œ

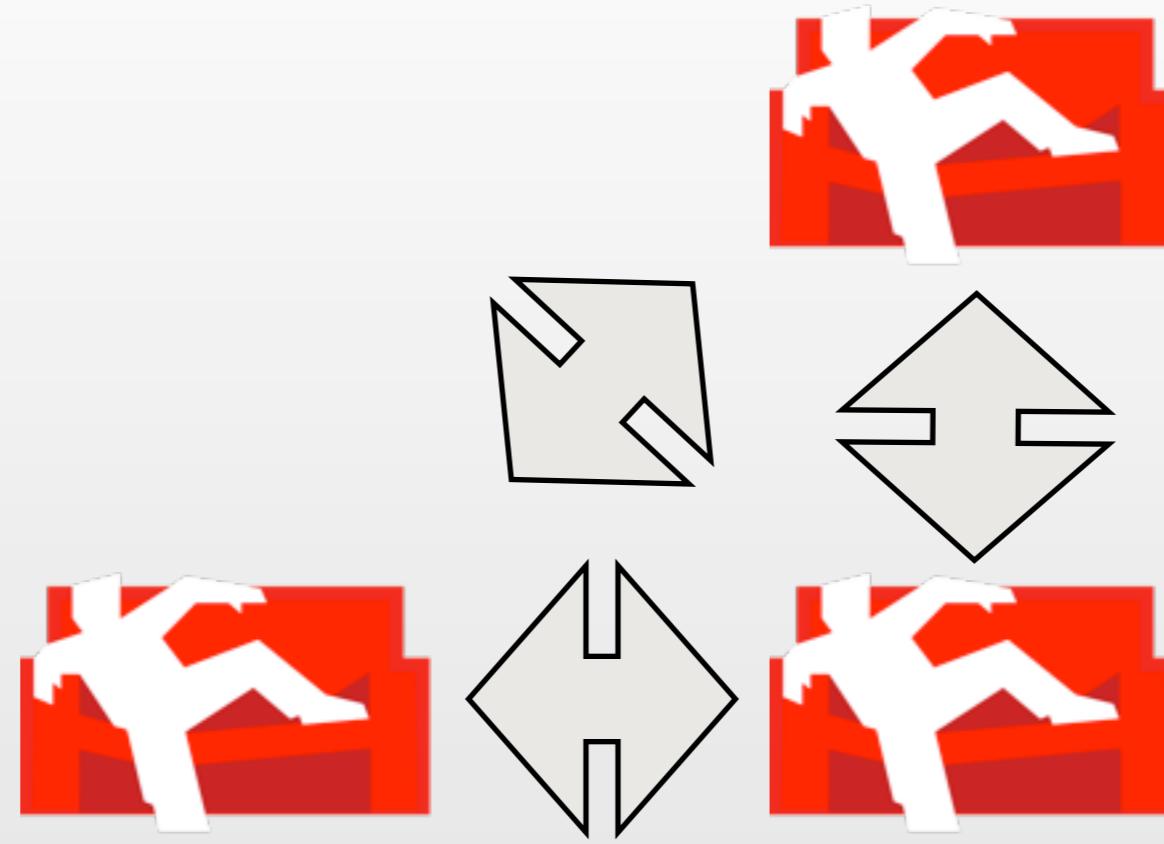


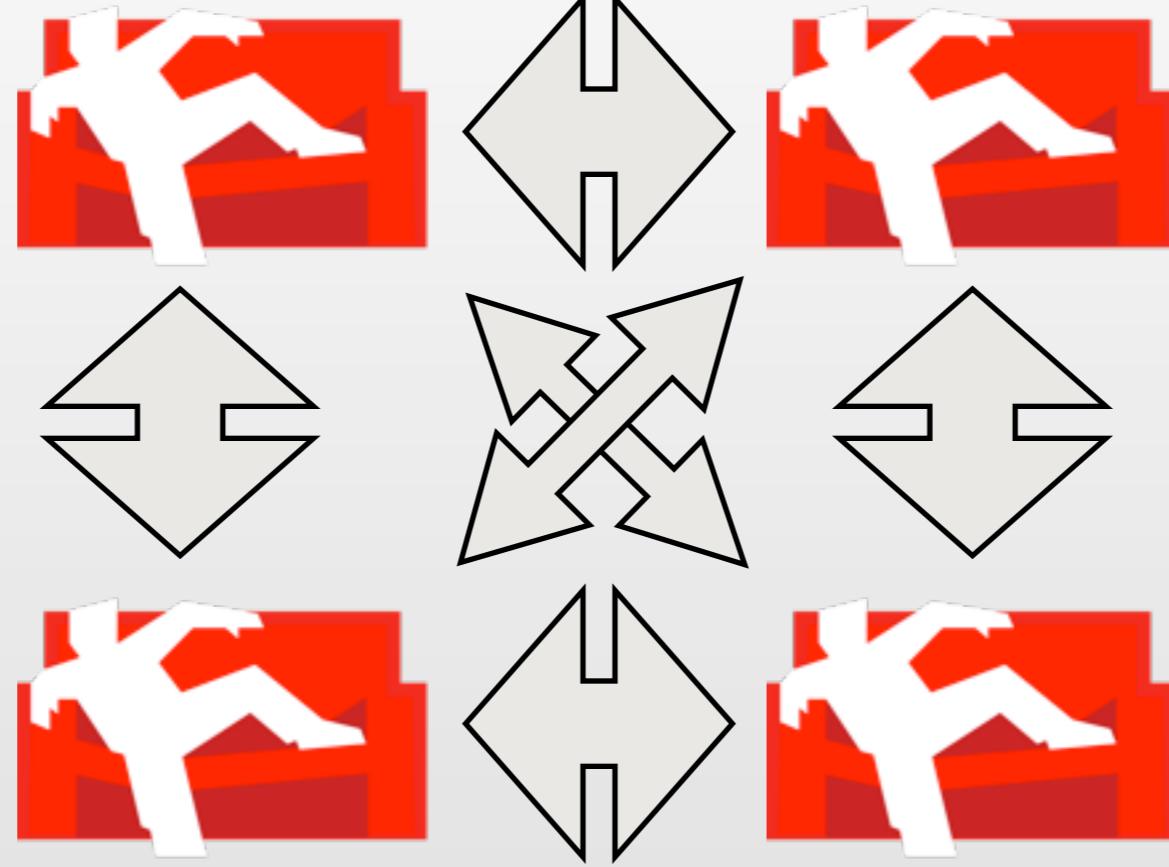


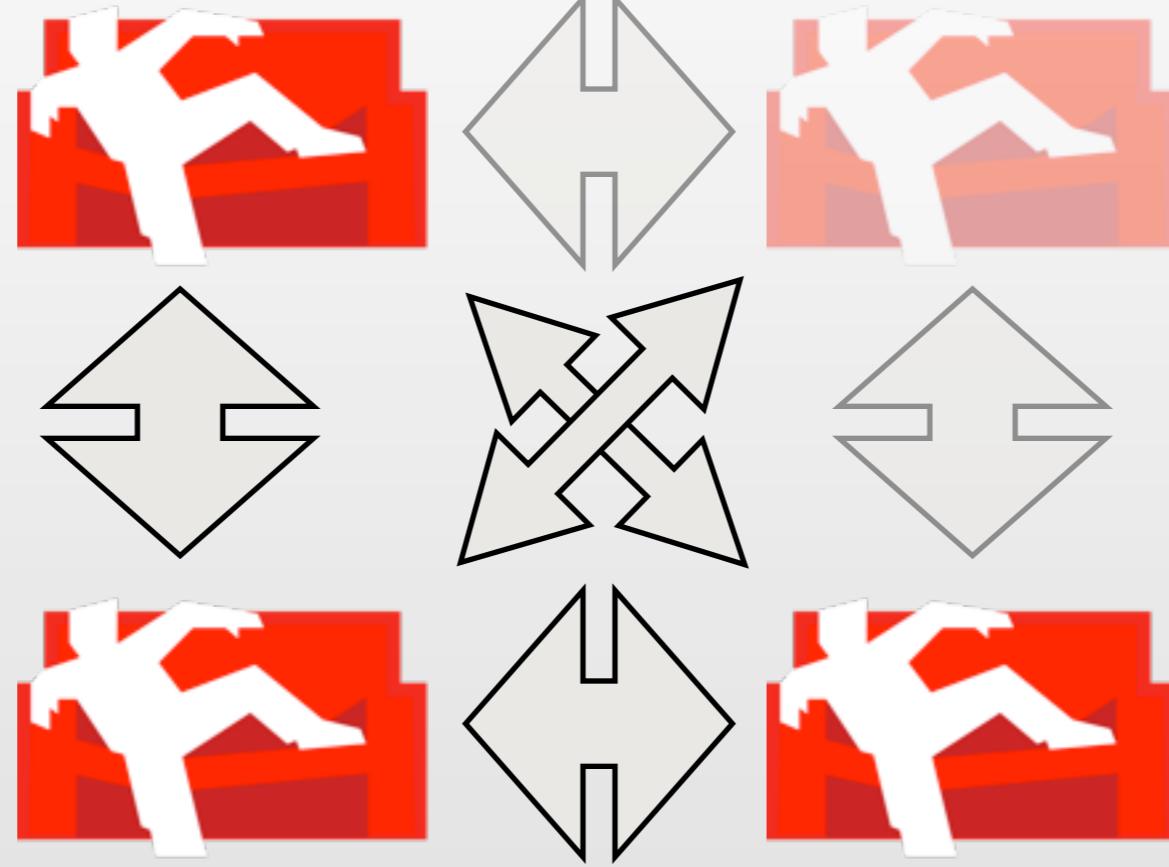


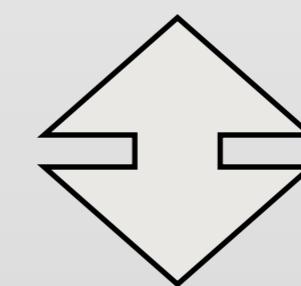
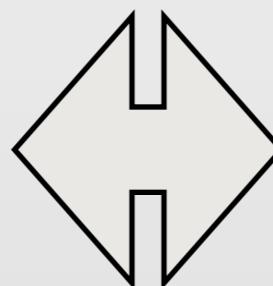
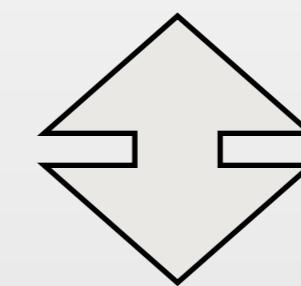
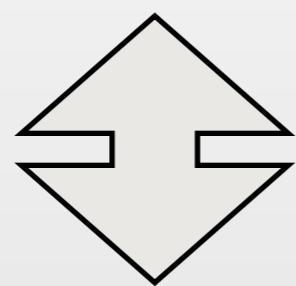
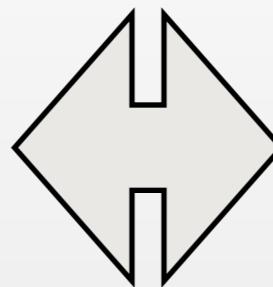
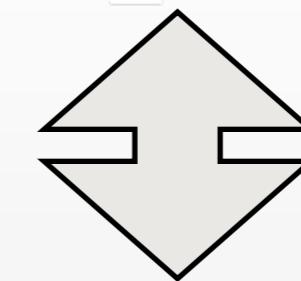


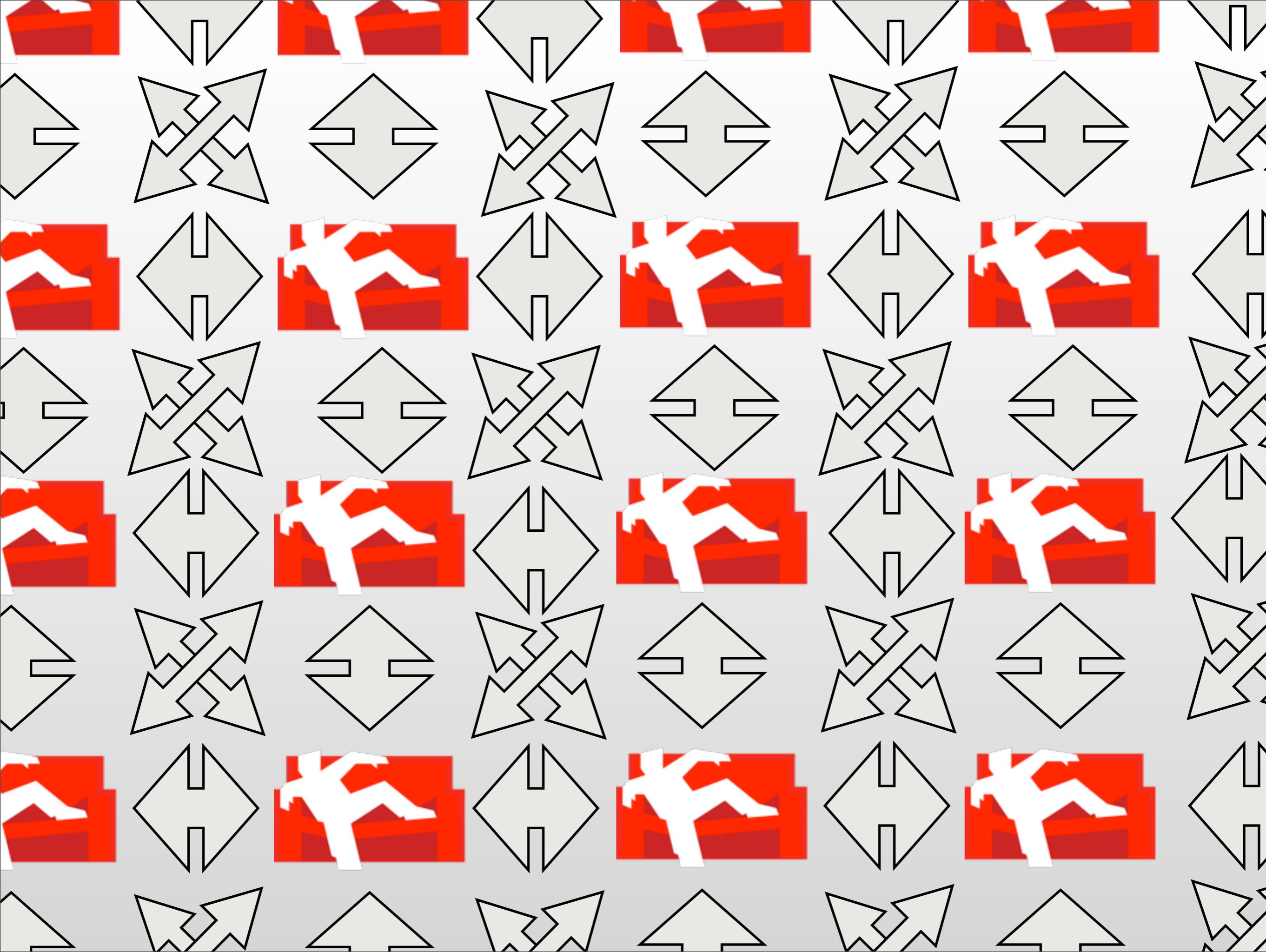












```

        function templateCache(data) {
            // Figure out if we're getting a template, or if we need to
            // load the template - and be sure to cache the result.
            var fn = cacheFn ||;

            // Generate a reusable function that will serve as a template
            // generator (and which will be cached).
            var functionCache = fn;
            var fnCacheFunction = function() {
                // Introduce the data as local variables using with()
                // (which is bad!).
                with(data);

                // Convert the template into pure JavaScript
                str =
                    .replace(/\{\{([^\}]+)\}\}/g, "'${$1}'")
                    .replace(/\{\{([^\}]+)\}\}/g, "'${$1}'")
                    .replace(/\{\{([^\}]+)\}\}/g, "'${$1}'")
                    .replace(/\{\{([^\}]+)\}\}/g, "'${$1}'")
                    .replace(/\{\{([^\}]+)\}\}/g, "'${$1}'")
                    .replace(/\{\{([^\}]+)\}\}/g, "'${$1}'")
                    .replace(/\{\{([^\}]+)\}\}/g, "'${$1}'");
                return str;
            };

            // Provide some basic currency to the user
            return data ? fnCacheFunction : fn;
        }

        // This stuff should be properly namespaced etc.
        // From stack 20
        function encodeOptions(options) {
            var buf = [];
            if (options === 'target') {
                for (var name in options) {
                    if (options.hasOwnProperty(name)) {
                        var value = options[name];
                        if (value === 'target') {
                            if (name === 'start' || name === 'end') {
                                value = '0x00000000';
                            } else {
                                value = hex2bin(value);
                            }
                        }
                        buf.push(encodeURIComponent(name) + '=' + encodeURIComponent(value));
                    }
                }
            } else if (buf.length) {
                return buf;
            }
            return '00' + buf.join('&');
        }

        function encodeArgsArray(args) {
            for (var i=0; i<args.length; i++) {
                args[i] = encodeURIComponent(args[i]);
            }
            return args;
        }

        function makePath(options) {
            var options, path;
            if (options.array.length > 1 && !options['string']) {
                // it's a parse hash
                options = options.array[0];
                options = options['string'];
                path = options.map(function(item) {
                    return encodeURI(item);
                }).join('/');
            } else {
                return path = encodeOptions(options);
            }
        }

        function assetPath() {
            var parts = ['req.path[0]', 'design', req.path[0]];
            return makePath(concatArgs(parts, arguments));
        }

        function shadowPath() {
            var parts = ['req.path[0]', 'shadow', req.path[0]];
            return makePath(concatArgs(parts, arguments));
        }

        function listPath() {
            var parts = ['req.path[0]', 'list', req.path[0]];
            return makePath(concatArgs(parts, arguments));
        }

        function makeAbsolute(path, url) {
            return 'http:///' + req.headers.host + path;
        }

        function fdd() {
            // Format integers to have at least two digits.
            return n + '0' + ('0' + n).slice(-2);
        }

        Data.prototype.toHexString = function() {
            return this.toJSON().toString();
        };
        FDD.prototype.toJSON = function() {
            return '0x' + this.id;
        };
        FDD.prototype.toHex = function() {
            return '0x' + this.id;
        };
        FDD.prototype.toBinary = function() {
            return '0b' + this.id;
        };
        FDD.prototype.toOctal = function() {
            return '0o' + this.id;
        };
        FDD.prototype.toDecimal = function() {
            return '0d' + this.id;
        };
    }

    // Log(req.headers.Accept);
    // We only show first
    return templateCache.templates.path, {
        title: doc.title,
        slugpath: blog.slug,
        post: doc.html,
        date: doc.createdAt,
        author: doc.user,
        assets: concatPaths(),
        authorPath: shadowPath('edit', doc._id),
        index: listPath('index', 'recent-posts', {descending: true, limit: 10})
    };
}

```

# Traditional Server

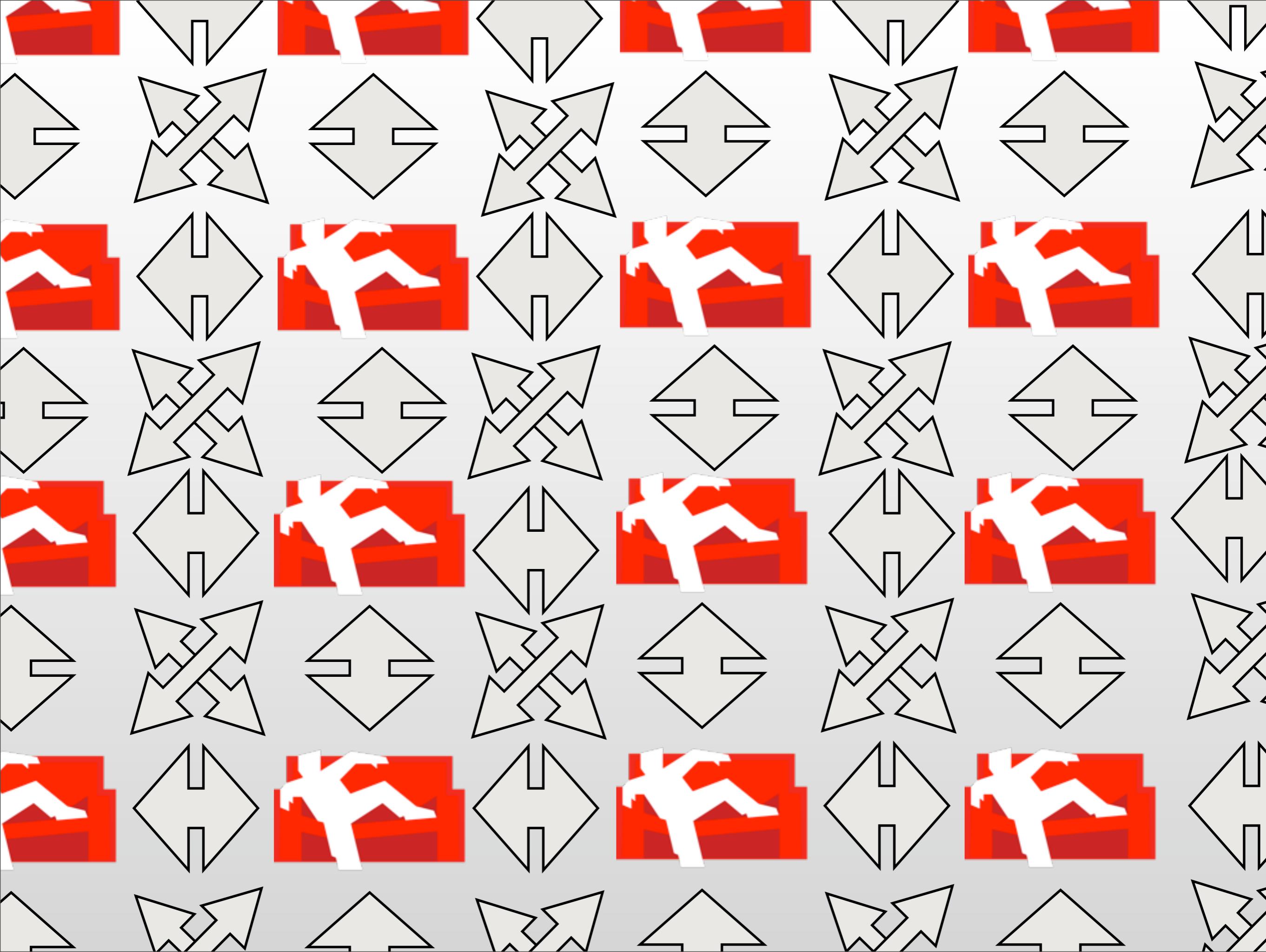
- We're familiar with the application model.
- We know we can scale CouchDB.
- Centralized servers are hard to run.
  - Users expect low latency responses.
  - Peak traffic events are interesting.

# Offline Mode

- Application model is more like a desktop.
- Code runs primarily at the client.
- Replication keeps the edge in sync with the cloud.
- Latency is less important than throughput.

# Peer to Peer

- Needs no primary service provider.
- Users can mashup apps and data.
- Generative because people own the code.
- We're still learning application models.









Apache  
**CouchDB**