Decentralized Document Delivery
Who am I?

Big Blue Hat

We're hiring!!
What I do @ hypothes.is

“an integration liaison and ambassador for the people of the Open Web at Hypothes.is; a platform ombudsperson to hold us accountable to our vision of a freely annotated Web.”
Why Decentralized
There is no center.

• Data “in the cloud”
  – only as good as my access to it.
• Small Data is usually enough for me.

• Cloud Powers (optional)
• Crowd Powers (optional)
• Core Powers (mine!)
Me in the middle.

EGOCENTRIC ARCHITECTURE
The Cloud is a Lie! The Cloud is a...
We just need

MOAR CLOUD!
Beyond the Silver Lining

- Cloud in your pocket?
- Cloud without connection?!
- Cloud in space?!!1!

There is no center.
Except your own.
Practical Deployments

- **Dimagi CommCareHQ**
  - Uses Apache CouchDB

- **MedicMobile.org**
  - Uses Apache CouchDB

- **eHealthAfrica**
  - Uses Apache CouchDB, Apache Cordova, & PouchDB
Why Documents
I :heart: Documents

• It’s how we think
• Data alone can be a bit too small
• Documents provide data + **context**
  – Identification
  – Provenance
  – Ownership
  – Licensing
  – Routing (yeah email!)
Why Delivery
Why Delivery

• Copies of copies of copies

• Single Source of Truth is...
  – Inefficient
  – Impossible
  – Not fault tolerant
  – A myth

• Eventually Consistent is...
  – Not a myth ;)}
APACHE COUCHDB DELIVERS!
Why Apache CouchDB

- Because Replication
- Stateless HTTP API
- JSON documents
- + binary attachments
- Map/Reduce-based index building

...without replication it’s just more NoSQL
Replication

• Multi-Version Concurrency Control

result after bi-directional replication

srsly
• Forgiving Conflict Model

CouchDB arbitrarily, but consistently picks a winner and keeps conflicts around...just in case.
Picking a different “winner”

• Ask for the conflicts
• GET the “losing” document @ 4b
• PUT it as a new revision
Eventually Things Match
Master-Master Replication

Cloud?

CouchDB

Laptop?

CouchDB

Desktop?
Delivering Documents...

BEYOND THE COUCH
CouchDB’s Little Cousin

pouchdb
CouchDB’s Little Cousin

pouchdb

Browser?

Hood.ie?

Node.js Server?

Cozy.io?

Node.js Laptop?

Cloud?
Meet PouchDB

• Implements CouchDB’s replication protocol
  – In the browser & node.js
• Web App becomes CouchDB-friendly replication endpoint
• Very active projects
• Lots of plugins & adapters
  – desktop & mobile browsers + node.js servers
PouchDB + CouchDB

- Data where you need it.
- Consistent Data Model on Server & Client
- Replication to tie them together
  - Master-Master replication (again)
- Consistent Conflict Model on both ends
Setup & Sync with PouchDB

```javascript
var db = new PouchDB('dbname');

db.put({
  _id: 'dave@gmail.com',
  name: 'David',
  age: 68
});

db.changes().on('change', function() {
  console.log('Ch-Ch-Changes');
});

db.replicate.to('http://example.com/mydb');
db.replicate.from('http://example.com/mydb');
// or
PouchDB.sync(db, 'http://example.com/mydb');
```
Markdown Editor built with PouchDB

PILLOW NOTES
Pillow Notes

• Yet Another Markdown Editor Thing
• JSON looks like:
  – “_id”: “...title of the note...”
  – “markdown”: “...the note...”
  – “created”: “...iso8601...”
  – “updated”: “...iso8601...”

• http://bigbluehat.github.io/pillow-notes
Pillow Notes

# 2015 ApacheCon: Big Data 0000001

Speaking on:
* Decentralized Document Delivery
* Upholstering Apache CouchDB
* Finding Your Way Around Apache

2015 ApacheCon: Big Data 0000001

Speaking on:
* Decentralized Document Delivery
* Upholstering Apache CouchDB
* Finding Your Way Around Apache
Pillow Notes Implementation

- HTML5, CSS, JS
- PouchDB
  - Persistence in browser
  - Replication out to CouchDB, Cloudant, etc
    - For backup, sharing, publication?
- Vue.js
  - Interaction
- HTML5 App Manifest (soon)
  - Fully offline (once added...)
Static Hosting Pillow Notes

- On GitHub Pages
  - `git push origin gh-pages`
  - http://bigbluehat.github.io/pillow-notes/

- On Cloudant
  - `couchapp push .`
  - http://{KEY}:{SECRET}@bigbluehat.cloudant.com/pillow-notes/
  - http://bigbluehat.cloudant.com/pillow-notes/_design/pillow-notes/_rewrite/

- On CouchDB locally
  - `couchapp push .`
  - http://{user}:{pass}@localhost:5984/pillow-notes/

- Apache server...of course ;)}
Username, Password, URL of Database
Click “Sync”

```javascript
PouchDB.sync(local, remote)
  .then(function(resp) {...});
```

Bi-directional Replication MAY create conflicts
CORS & Single Origin Pain

• Cross Origin Resource Sharing
  – Disables a core feature of the Web
  – Makes moving JSON with Browsers painful
• (re?)Enable CORS
  – npm install -g add-cors-to-couchdb
  – Cloudant has some UI, but only works over HTTPS
• Can’t share without CORS being **enabled**
• OK...it’s actually the Single Origin Policy...
getting from local to remote and back

WORK LOCALLY; SYNC GLOBALLY
Decentralized Cloud with Friends!

• Per user database
• Per share database
  – User to user
  – Group to group
• Client does most of the work
Federation for Alice, Bob, & Charlie

Federation

alice
private-user-space
filtered replication
share-with-bob
filtered replication
share-with-charlie

Cloudant or remote Apache CouchDB
private-user-space

bob
private-user-space
filtered
share-with-alice

charlie
private-user-space
filtered
share-with-alice

alice-bob
replicate
alice-charlie
replicate
Cloudant or remote Apache CouchDB

private-user-space

(alternative)

private-user-space

share-with-bob

share-with-charlie

filtered replication

filtered replication

replicate

replicate

groups

alice-bob

alice-charlie

filtered

share-with-alice

Extension / App

private-user-space

share-with-alice

Extension / App

private-user-space

share-with-alice

Extension / App

private-user-space

share-with-alice

Extension / App

private-user-space

share-with-alice

Extension / App

private-user-space

share-with-alice

Extension / App

private-user-space

share-with-alice
Federation for Alice, Bob, & Charlie

• Filtered replication on the client
• Peer-to-peer replication when cloudless
• Security centered around the database(s)
• (optional) Continuous replication to the cloud
Similar Projects

- **http://pouch.host/**
  - a service that lets your PouchDB applications easily provide login and online sync functionality
  - single user app scenarios (so far)

- **couch-per-user**
  - daemon that ensures that a private per-user database exists for each document in _users_

- Platforms: **hood.ie, cozy.io, ddoc.me**
Decentralized

DOCUMENT DESIGN
Design for Change

• Focus on change “vector”
  – Updated often?
  – Can I split this out?
  – Can I put it back together?
  – Can I build the index I want from this?

• Mind like Paper
Design for Change - _id

• Document ID
  – /{db}/{doc_id}

• Only source of uniqueness
  – UUID’s by default (via POST)

• Primary Index range
  – ?startkey=“contact”
  – &endkey=“contactZ”
Design for Change - keys

• Informative
• Can’t be underscore prefixed
  – The one thing CouchDB (& PouchDB) reserve
• JSON-LD?
  – Map Strings to Things
  – Bit tedious in JS doc[‘@type’] vs. doc.type
  – Still worth it
• Lazy (and large...) secondary index
Lazy (and large...) secondary index

```javascript
function (doc) {
    Object.keys(doc).forEach(function (key) {
        emit(key, doc[key].toString());
    });
}
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc_id</td>
<td>“company”</td>
<td>“[object Object]”</td>
</tr>
<tr>
<td>doc_id</td>
<td>“date_key”</td>
<td>“2015-09-29T...”</td>
</tr>
<tr>
<td>other</td>
<td>“age”</td>
<td>“34”</td>
</tr>
<tr>
<td>other</td>
<td>“name”</td>
<td>“Benjamin Young”</td>
</tr>
<tr>
<td>other</td>
<td>“array”</td>
<td>“list,of,keys”</td>
</tr>
</tbody>
</table>
Design for Change - values

• Values
  – How nested?
  – How legible?
  – What type?
    • String
    • Number
    • Object
    • Array

• Dates – use ISO 8601 vs. numeric Unix epoch
Other People’s JSON

• Postel’s Law > Sartre’s Plays?
  – conservative in what you send
  – liberal in what you accept

• Schemaless FTW!

• “normalize” at read time (not write time)
  – schema on the way out
conflict happens

DEALING WITH CONFLICT
Arbitrary but Awesome!

- CouchDB consistently picks **arbitrary** winner
- Winner is the current document
  - GET /{db}/{doc_id}
- Ask for conflicts to see non-winning revision(s)
  - GET /{db}/{doc_id}?conflicts=true
    - {…, “_rev”: “2-zzz”, “_conflicts”: [“2-1234”, …]}
- Pick a new winner by overwriting it
  - PUT /{db}/{doc_id}
    - {“_rev”: “2-1234”, …}
    - > {“_rev”: “3-qwerty”, …}
Map Reduce for Conflicts

```javascript
function (doc) {
    if (doc._conflicts) {
        emit(null,
             [doc._rev].concat(doc._conflicts));
    }
}
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc_id</td>
<td>null</td>
<td>“2-b91bb807b4685080c6a65111”</td>
</tr>
<tr>
<td>doc_id</td>
<td>null</td>
<td>“2-65db2a11b5172bf928e3bcf5”</td>
</tr>
<tr>
<td>other</td>
<td>null</td>
<td>“5-5bc3c6319edf62d4c624277f”</td>
</tr>
<tr>
<td>other</td>
<td>null</td>
<td>“5-4277fdd0ae191080c6a651oz”</td>
</tr>
<tr>
<td>other</td>
<td>null</td>
<td>“5-807b4685080c6a65111wer46”</td>
</tr>
</tbody>
</table>
Map **Reduce** for Conflicts

`_count` // built in reducer

<table>
<thead>
<tr>
<th>ID</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>doc_id</code></td>
<td>null</td>
<td>2</td>
</tr>
<tr>
<td><code>other</code></td>
<td>null</td>
<td>3</td>
</tr>
</tbody>
</table>

- Handy for UI-level conflict notifications
- GET `/{db}/{doc_id}?conflicts=true`
  - display them together & let the user pick
Human rights and all that.

BRINGING THIS TO ANNOTATION
Ask to Annotate?!  

• You bought the book
  – You can scribble in it.
  – You can share it.
  – You can write content about it.

• You should **not**
  – Have to ask.
  – Need a “middle man.”
Offline Annotator

- [http://github.com/bigbluehat/annotator-pouchdb](http://github.com/bigbluehat/annotator-pouchdb)
- offline-annotator.xpi (for Firefox)

- Uses PouchDB + [Annotator](http://github.com/bigbluehat/annotator-pouchdb)

- Soon:
  - Sync UI
  - W3C Web Annotation Data Model
  - Your help! ^_^
Thanks!

- **bigbluehat.com**
- **@bigbluehat**
- **github.com/BigBlueHat**
- bigbluehat on irc.freenode.net
  - #couchdb #pouchdb #hypothes.is
- **bigbluehat@apache.org**
- **byoung@bigbluehat.com**
- **bigbluehat@hypothes.is**