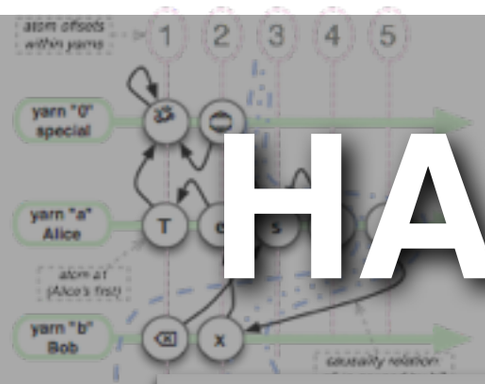
SWARM @ WEB
STANDARD DAYS

реактивная синхронизация данных
для веб- и мобильных приложений

Киев
6 декабря 2014

НАУКА И ТЕХНИКА



rooted subtree out of the whole tree; that subtree represents a revision. A causal tree grows in a very natural way, by forking and growing branches. To represent atom deletion in the same fork-and-grow way, deleted atoms are not removed from the tree, but followed by backspace atoms, designated as ⌫. Undeletion is similarly expressed by a special atom ♲, see [illegible] discussion of deletion/undeletion. [illegible] does not signal awareness. [illegible] was aware [illegible] causing atom [illegible] Bob's effect [illegible] logically merged [illegible] last [illegible] produced. [illegible] was the alphabetically ordered weave, which consists of all the symbols that ever existed, in order. Past or present revisions of a text are derived from its weave given a weft (see Fig. 2). All special symbols are included into the weave, but they are never shown in the resulting text, hence they

weft_a4

Figure 1: Yar
"Test", Bob co

user actions
Alice typed in "T
Bob: "Test"→"T
Alice saves Bob's

Figure 2: Weft-

form	atom1	at
value	x	

Figure 3: Diffe
b2 is the atom
the causing ato

Figure 4: Full

use, e.g. yarn3c
The second ba
fectively a revisi
id: last symbol i
those yarns were
across the yarns,
tor timestamps,
jected to transitiv
dependencies cau
format is more c
last symbol ids, s
ing of their yarn
phanumeric order
The causality
To give that tree
state of any text
root) and end, de
are between thos
typed, its causal
cept for the ⌫ sy
traversal of the t

Referencing within evolving hypertext

Victor Grishchenko, J. A. Pouwelse, and Henk Sips

Delft University of Technology
Mekelweg 4, 2628CD
Delft, The Netherlands
victor.grishchenko@gmail.com

Abstract. The classic hypertext model omits the process of text growth, evolution and synthesis. With hypertext creation becoming increasingly collaborative and change timescales becoming shorter, explicitly addressing text evolution is the key to the next stage of hypertext development. Uniform Resource Identifier (URI) is a proven general concept that enabled the Web. In application to versioned *deep* hypertext, expressive power of a classical hyperlink becomes insufficient.
Based on the Causal Trees model, we introduce a minimalistic but powerful query language of *specifiers* that provides us great flexibility of referencing within a changing hypertext. Specifiers capture the state of the text, point at changes, expose authorship or blend branches. Being a part of an URI, a specifier puts advanced distributed revision control techniques within reach of a regular web user.

1 Introduction

In the WWW/HTML model and, generally, in "chunked hypertext" systems the main addressable unit is a "page" which might optionally also have addressable "anchors" inside it. That is generally sufficient as long as we deal with static texts, albeit the requirement that a page author must pre-provision anchors is limiting. However, if we follow the general vision of a text as an evolving entity (the "wiki model"), then the expressive power of a classical hyperlink is insufficient. Since the addressed text is continuously changing, anchors might disappear, and the content that is actually addressed by the link might be re-edited or its surroundings may change. Similarly, there is no standard way of pointing at particular statements and passages in the text. For collaboratively created texts, such a possibility is desirable. Also, there is no semantics in place to address co-existing versions of a text (named "branches" in the version control parlance). Those might be drafts, reeditions, alternative versions. Thus, our mission is to explore possible approaches and conventions of referencing particular parts of text, its particular versions, or both. We want to measure, mark and cut text in breadth and depth!

This paper is structured as follows. First, we consider relevant existing models and their limitations in Section 2. Section 3 briefly describes the Causal Trees (*ct*) model of text versioning and the basic primitives available for text/ version

andex

Citrea and Swarm: partially ordered op logs in the browser

Victor Grishchenko
Citrea LLC
victor.grishchenko@gmail.com

25 March 2014

Abstract

Principles of eventual consistency are normally applied in large-scale distributed systems. I report experiences applying EC in Web app development. Citrea is a collaborative rich text editor employing the Causal Trees[4] technology of concurrency/version control (note: CT is *not* an OT flavor). CT employs symbol ids and trees and, generally, belongs to the same family as WOOT, Logoot or Treedoc [1, 3, 2]. CT makes the ids-and-trees approach production-practical by limiting itself to simple and lightweight algorithms and data structures. Swarm is a JavaScript object sync library that works in the browser, in real time. Swarm fully employs client-side storage and works well under intermittent connectivity. Swarm employs "pure" op-based model. Our top finding is a *specifier*, a serialized event description format that

ГЭДЖЕТЫ

СКОРОСТЬ

ВАЙ-ФАЙ ОТДЫХАЙ

ПРОШУ ВСЕХ ВСТАТЬ

ВЕБ ЕСТ

СИНХРОНИЗАЦИЯ

ЛИНЕАРИЗАЦИЯ

oplog, log

AP

CouchDB, LWW, CRD

LWWs, Meteor, Derby

Firebase

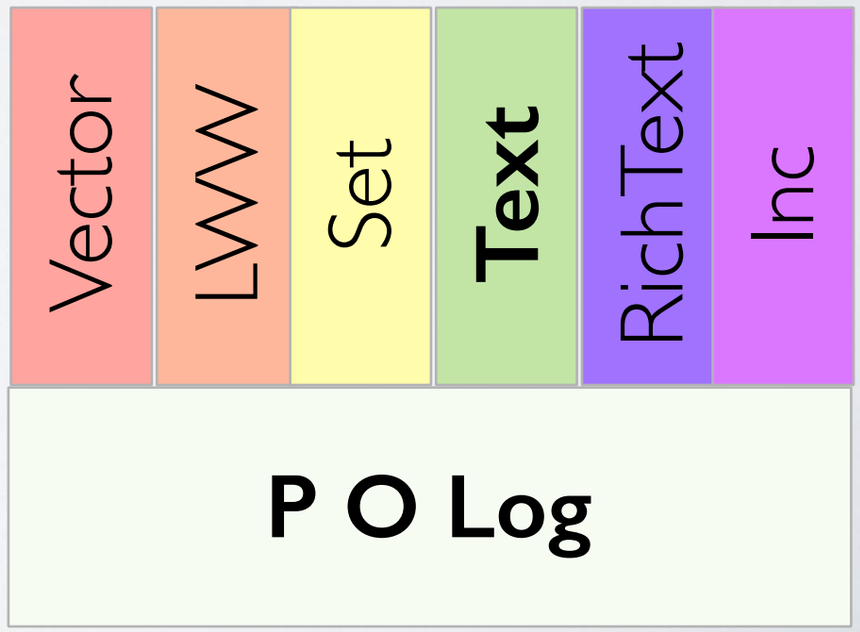
PubNub®

METEOR

CRDT

- Convergent…
- Commutative…
- Conflict-free…
- Cloud…

… Replicated Data Types



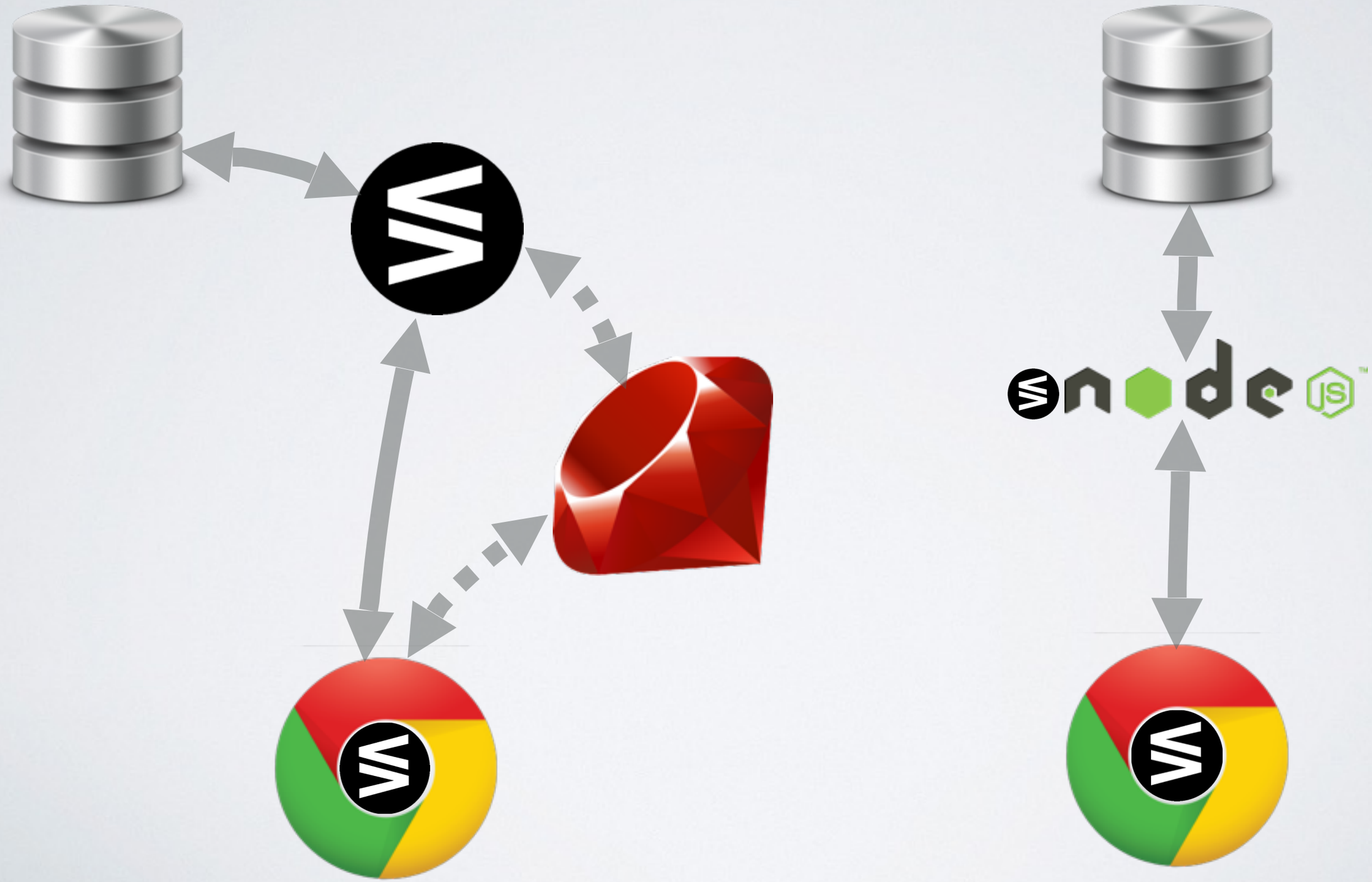
Cassandra

АБСТРАКЦИЯ

SWARM

Cup

АРХИТЕКТУРЫ



КОД

```
var host = new Swarm.Host('user~ssn');

host.connect('ws://localhost:8000');
```

```
var someMouse = new Mouse();

var mickey = host.get('/Mouse#Mickey',
   function(s, v, mickey){
      mickey.set({
            x:100,
            y:100
      });
   });
```

- [x] Мелкие баги
- [x] Инсталляция
- [x] Лендинг
- [x] Демонстрируемые фичи
- [x] Приятные фички
- [x] Совместимость
- [x] Чистота и порядок
- [x] Производительность и скорость
- [x] Тексты
- [x]
- [x] Потом

- [x] Скорость загрузки (bundle.js)
- [x] Префетч списков (через закладки?)
- [x] Скорость загрузки скрипта в FF (до prefet(
- [x] Убрать debug!!!

3 items left

ИЗОМОРФНО!

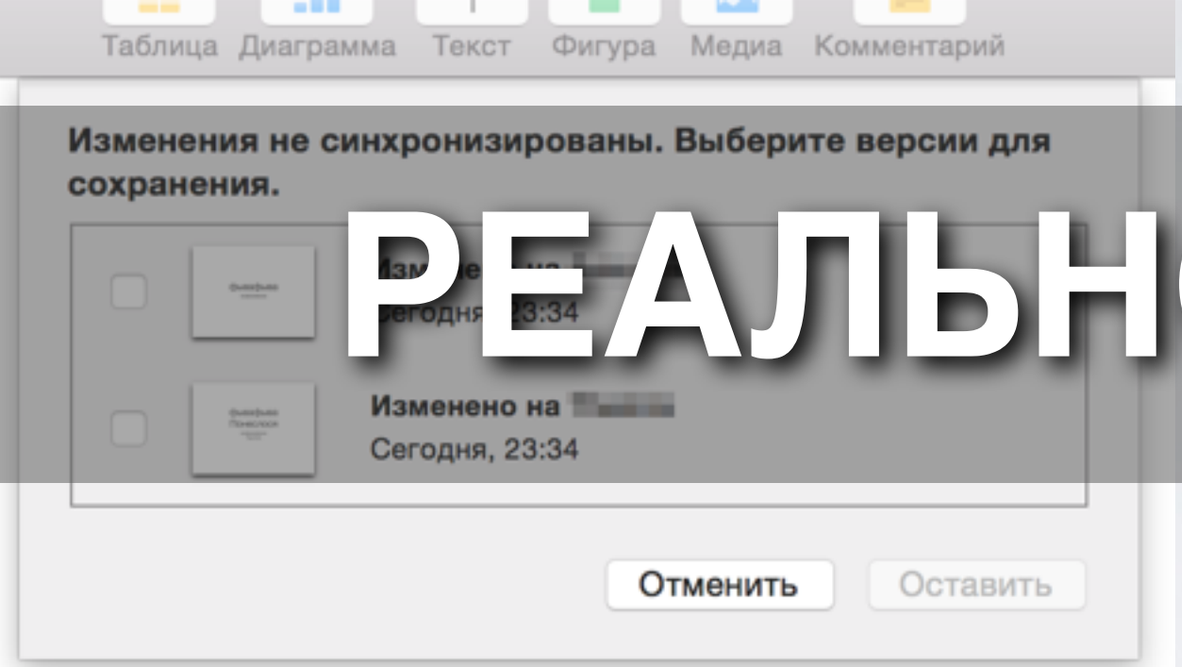
http://ppyr.us

Size Content	Time Latency	Timeline 400 ms 600 ms 800 ms
4.0 KB 12.5 KB	299 ms 152 ms	
34.4 KB 124 KB	623 ms 154 ms	
39.6 KB 146 KB	474 ms 152 ms	
637 B 324 B	288 ms 151 ms	
2.3 KB 2.1 KB	283 ms 146 ms	
0 B 0 B	Pending	

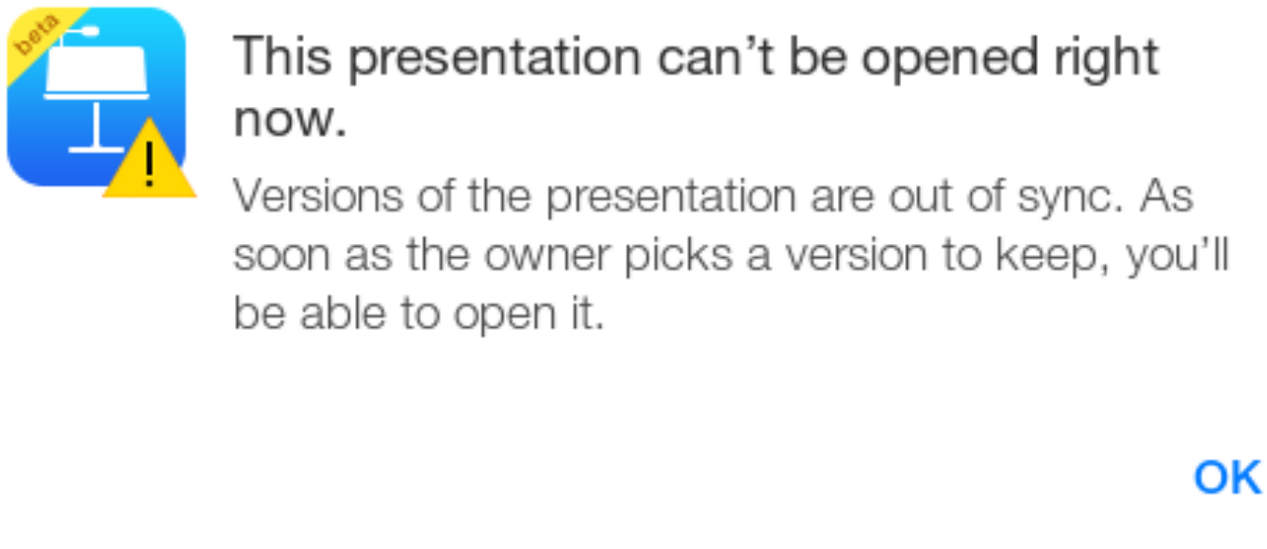
RTT
~140ms

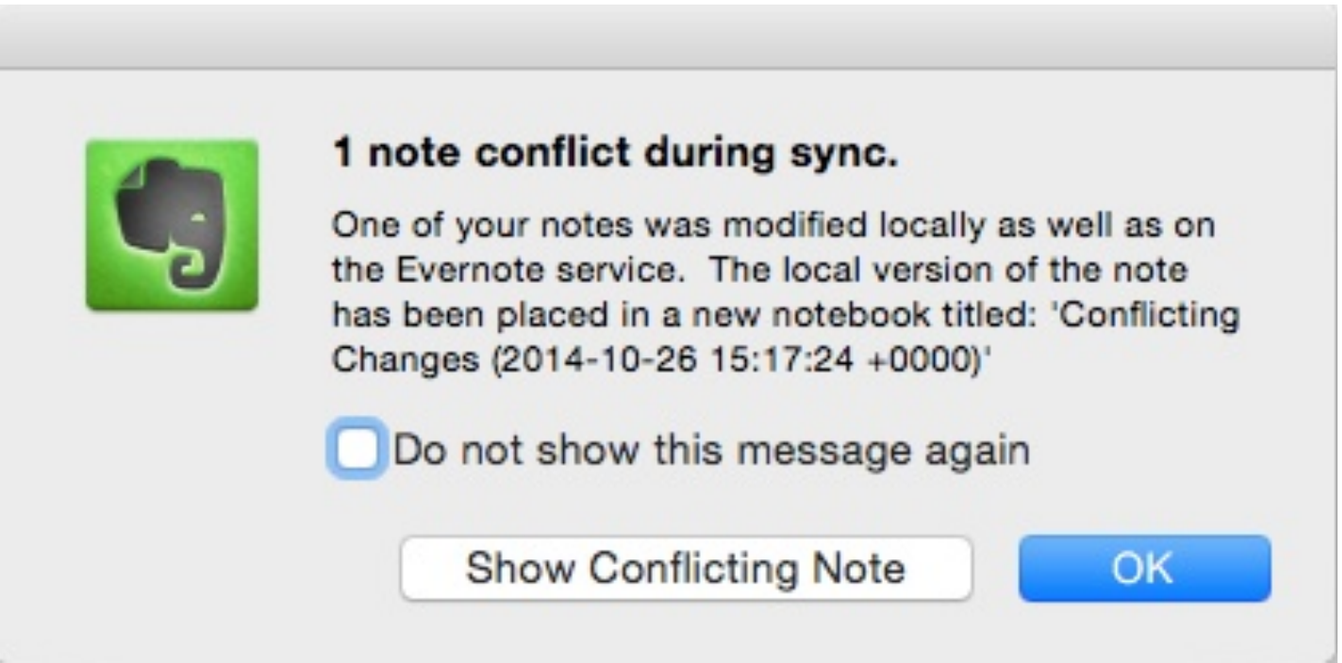
Size Content	Time Latency	Timeline
(from cache)	2 ms 2 ms	
(from cache)	9 ms 7 ms	
(from cache)	9 ms 9 ms	
(from cache)	9 ms 9 ms	
(from cache)	8 ms 8 ms	
0 B 0 B	Pending	

```
66  app.get(/[/+A-Za-z0-9_~]*/, function (req, res) {
67      var route = req.path;
68      var router = new TodoRouter();
69      res.header('Content-Type', 'text/html; charset=utf-8');
70      res.header('Cache-Control', 'private, max-age=0, no-cache');
71      res.write(htmlTemplate.head);
72      router.load(route, function (path) {
73          res.write(
74              React.renderComponentToString(
75                  TodoAppView({
76                      key: 'TodoApp',
77                      app: {path: path}
78                  })
79              )
80          );
81          res.write(htmlTemplate.tail);
82          res.end();
83      });
84  });
```



РЕАЛЬНОЕ ВРЕМЯ





КАК ЛОКАЛЬНОЕ

```
47  TodoApp.prototype.initSwarm = function () {
48      //this.storage = null;
49      this.storage = new Swarm.SharedWebStorage('webst',{persistent:true});
50      this.wsServerUri = 'ws://'+window.location.host;
51      this.host = Swarm.env.localhost = new Swarm.Host(this.ssnid,'',this.storage);
52      this.host.connect(this.wsServerUri, {delay: 50});
53  };
```

host.connect(uri);

ОФФЛАЙН

Oops! Keynote went offline.

Before editing, make sure you're connected so all your changes are saved.

OK

There was a problem

We're sorry. The server encountered an error. Please press "OK" to refresh the sheet.

OK

инвестор	Сергей Азатян	InVenture Partners	подтвердил	Маруся	нет ? Какое участие инв помогать / ?)

Connection Lost

search request

МАНИФЕСТ

```
1  CACHE MANIFEST
2  #2014-10-17 v0.0.7
3
4  CACHE:
5  /dist/react.min.js
6  /dist/TodoApp.app.js
7  /fake_auth_init.js
8  /css/base.css
9  /css/add.css
10 /css/bg.png
11
12 FALLBACK:
13 / /offline.html
```

ЧТО ДЕЛАТЬ?

https://github.com/gritzko/swarm

ССЫЛКИ

Demo: http://ppyr.us
Code: https://github.com/gritzko/todomvc-swarm
Swarm.js: https://github.com/gritzko/swarm
Blog: http://swarmjs.github.io
Twitter: http://twitter.com/swarm_js
Habr (ru): http://habrahabr.ru/company/swarm/
Slideshare: https://github.com/gritzko/swarm

Email: swarm.js@gmail.com

Twitter: @swarm_js