Hans-Christian Otto

@muhdiekuh

PouchDB

The perfect Offline-First Database for your Browser

PouchDB

suora

@muhdiekuh

Hans-Christian Otto

The perfect Offline-First Database for your Browser

PouchDB

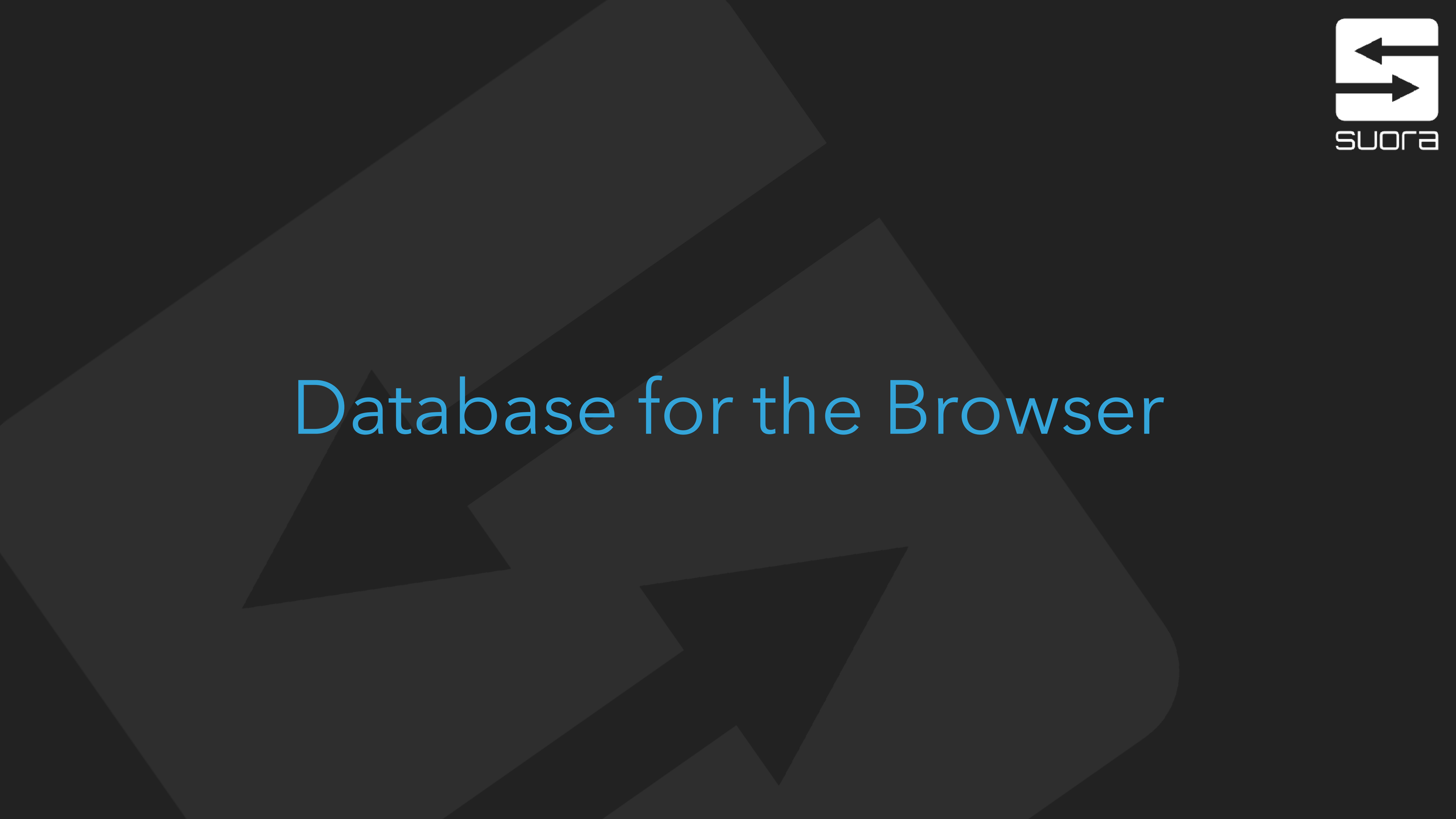
The perfect Database for PWAs?

PouchDB

So, you got your Service Worker...

But


suora
What about data?



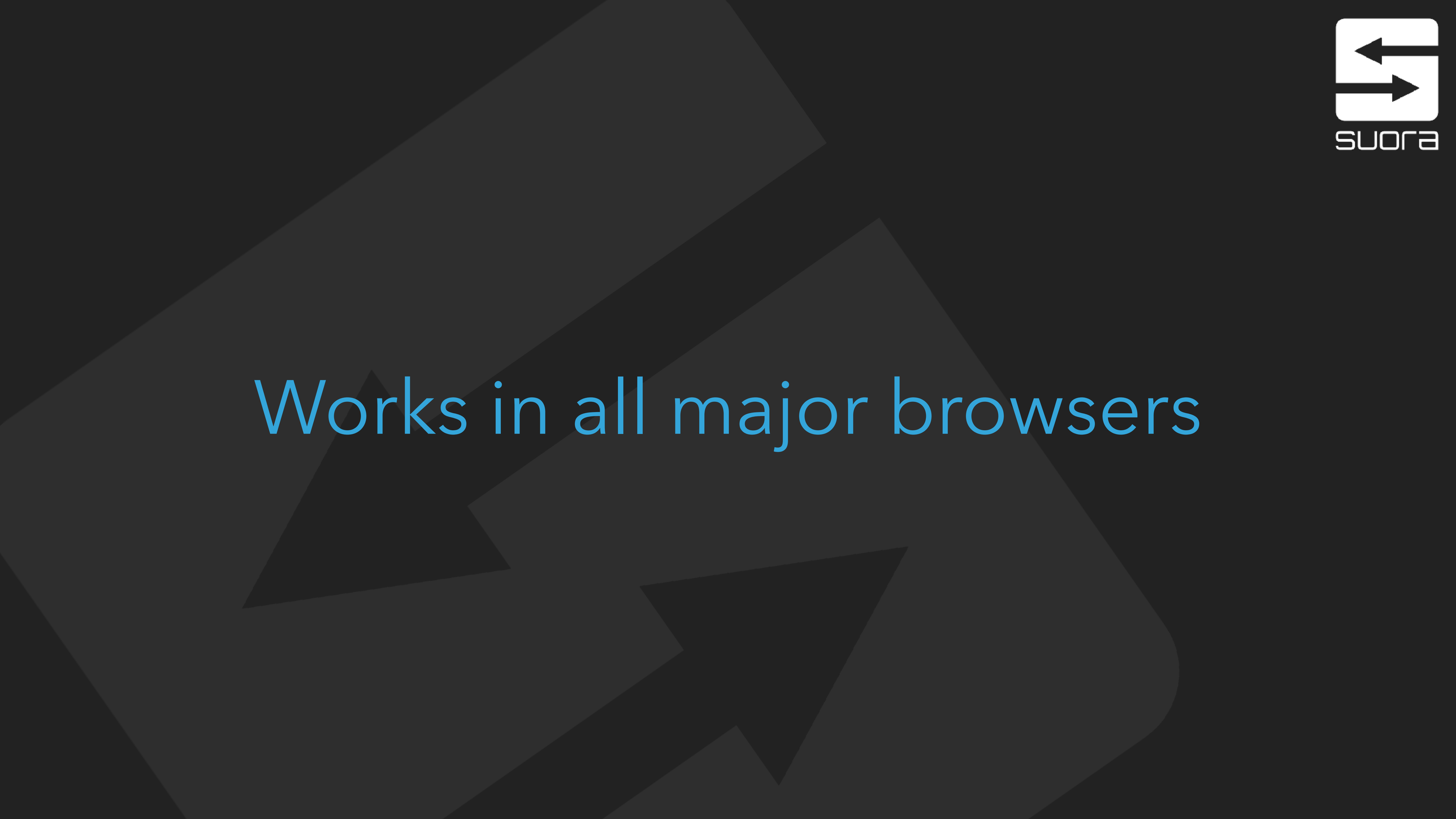
Database for the Browser

Stores data in IndexedDB, WebSQL, …

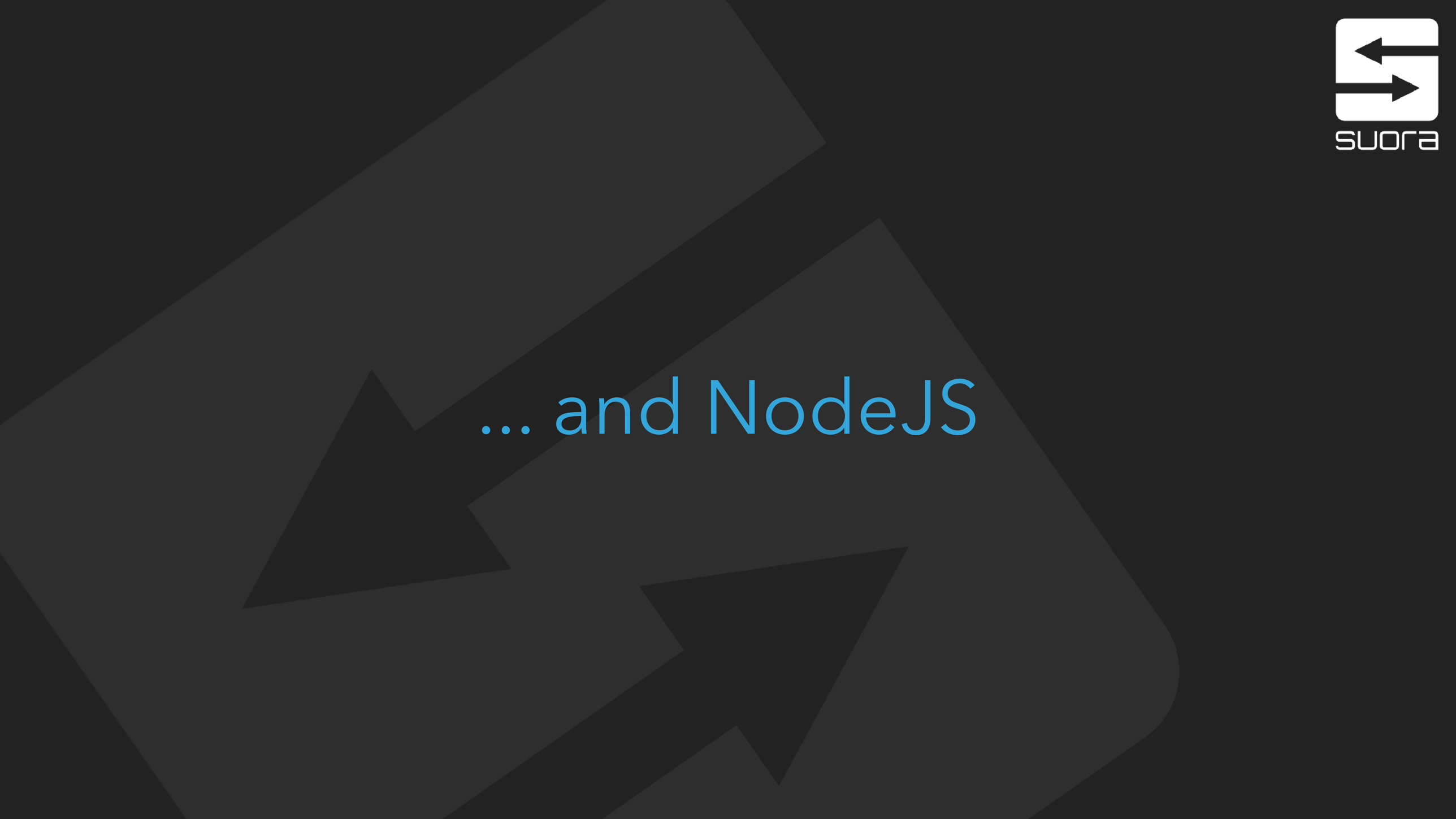
Lightweight (46kb)

Open Source
(Apache 2.0 licensed)

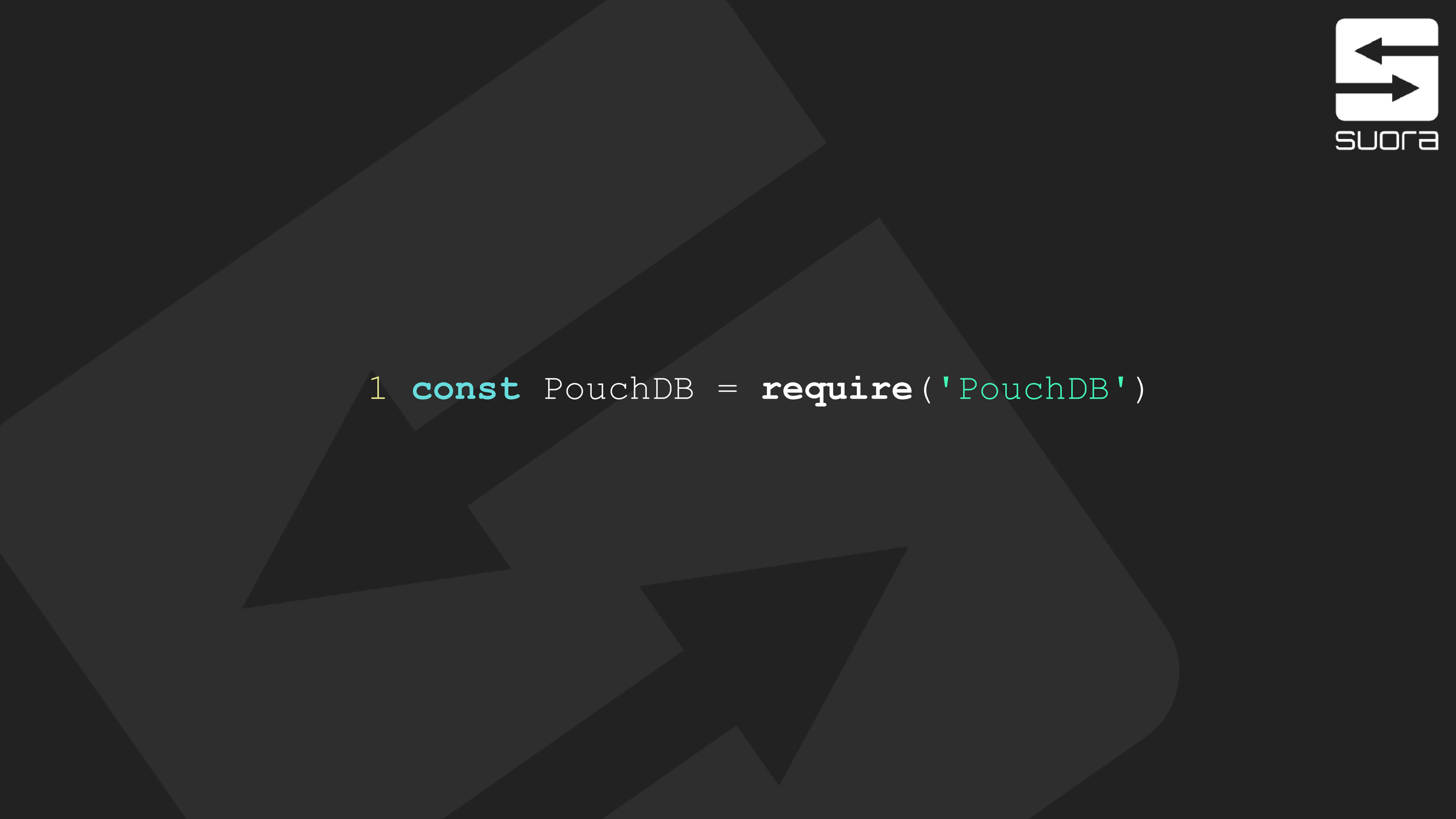
Nice JavaScript APIs



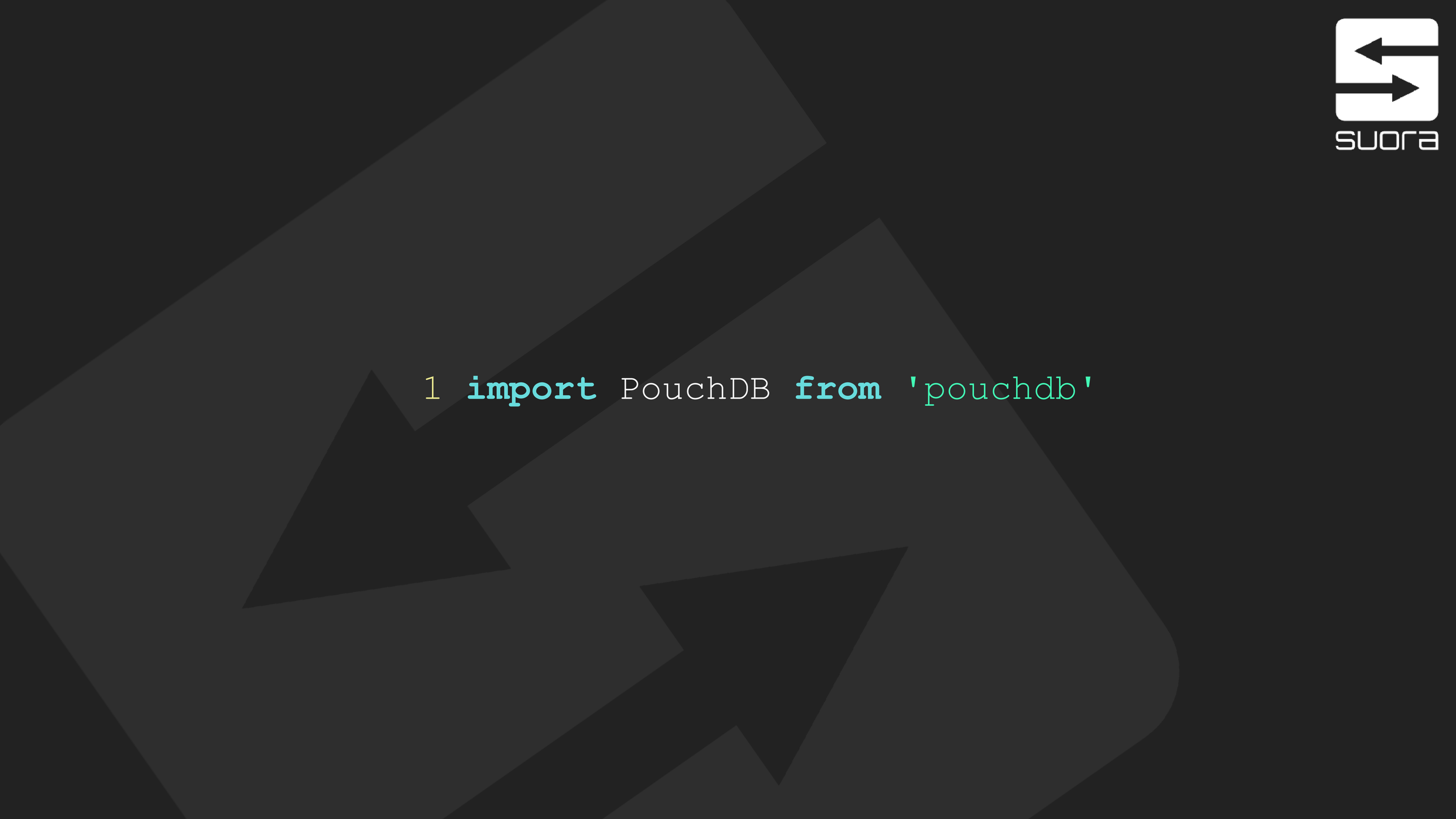
Works in all major browsers



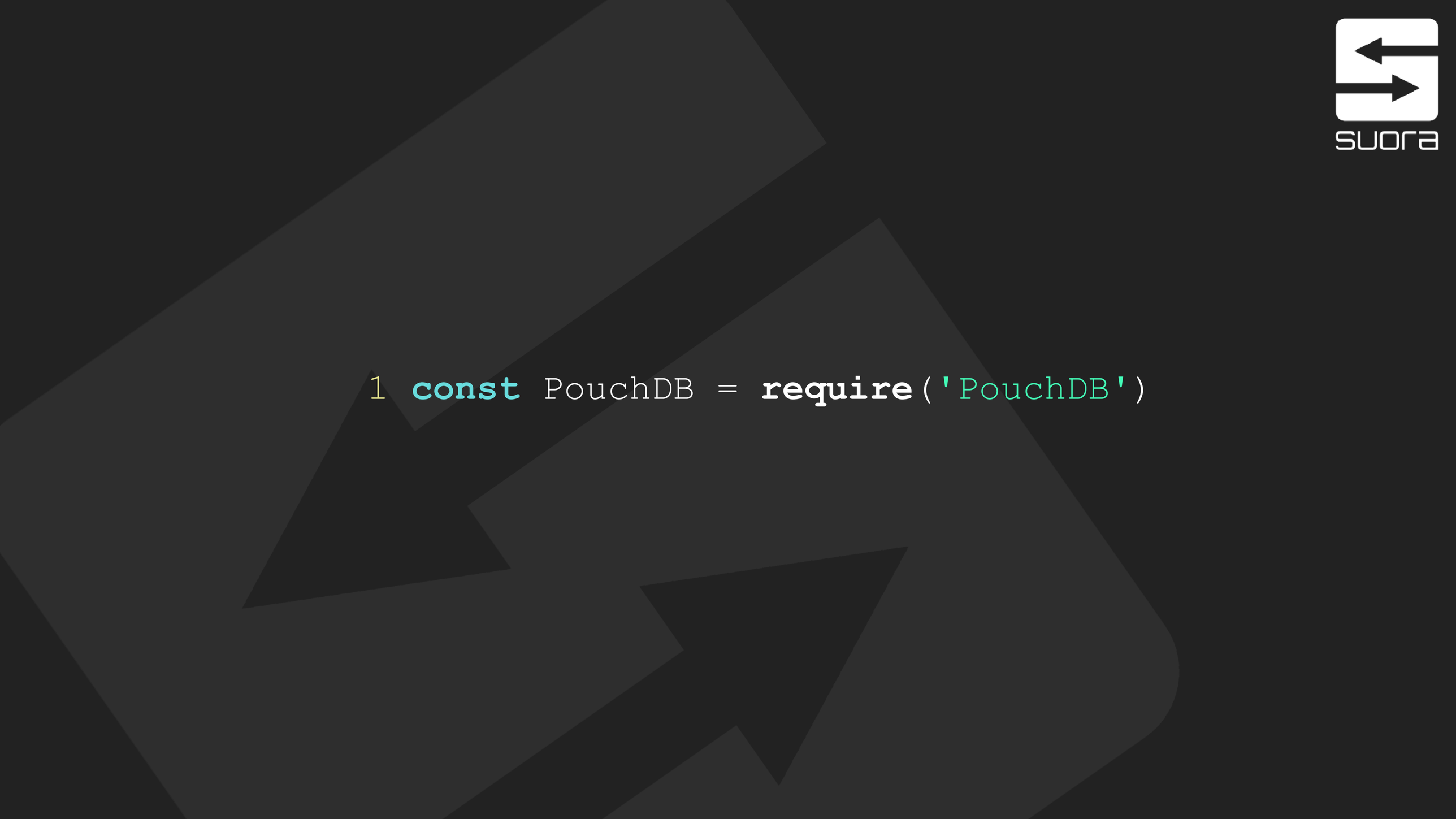
... and NodeJS




```
1 const PouchDB = require('PouchDB')
```




```
1 import PouchDB from 'pouchdb'
```




```
1 const PouchDB = require('PouchDB')
```


```
1 const PouchDB = require('PouchDB')
2
3 const db = new PouchDB('micro-status')
```


```
1 const PouchDB = require('PouchDB')
2
3 const db = new PouchDB('micro-status')
4 db.put({
5         _id: 'test',
6         type: 'status',
7         text: 'Hello World!',
8         time: new Date().toISOString()
9 })
```


```
1 const PouchDB = require('PouchDB')
2
3 const db = new PouchDB('micro-status')
4 db.get('test').then(function(doc) {
5   console.dir(doc)
6 });
```


```
1 const PouchDB = require('PouchDB')
2
3 const db = new PouchDB('micro-status')
4 db.get('test').then(function(doc) {
5   console.dir(doc)
6 });
```

```
1 { type: 'status',
2   text: 'Hello World!',
3   time: '2017-10-09T08:34:10.016Z',
4   _id: 'test',
5   _rev: '1-61c9cea4baca454b89c1e56e7f0f6ea6' }
```

List Functionality

„Querys“


suora
Map/Reduce

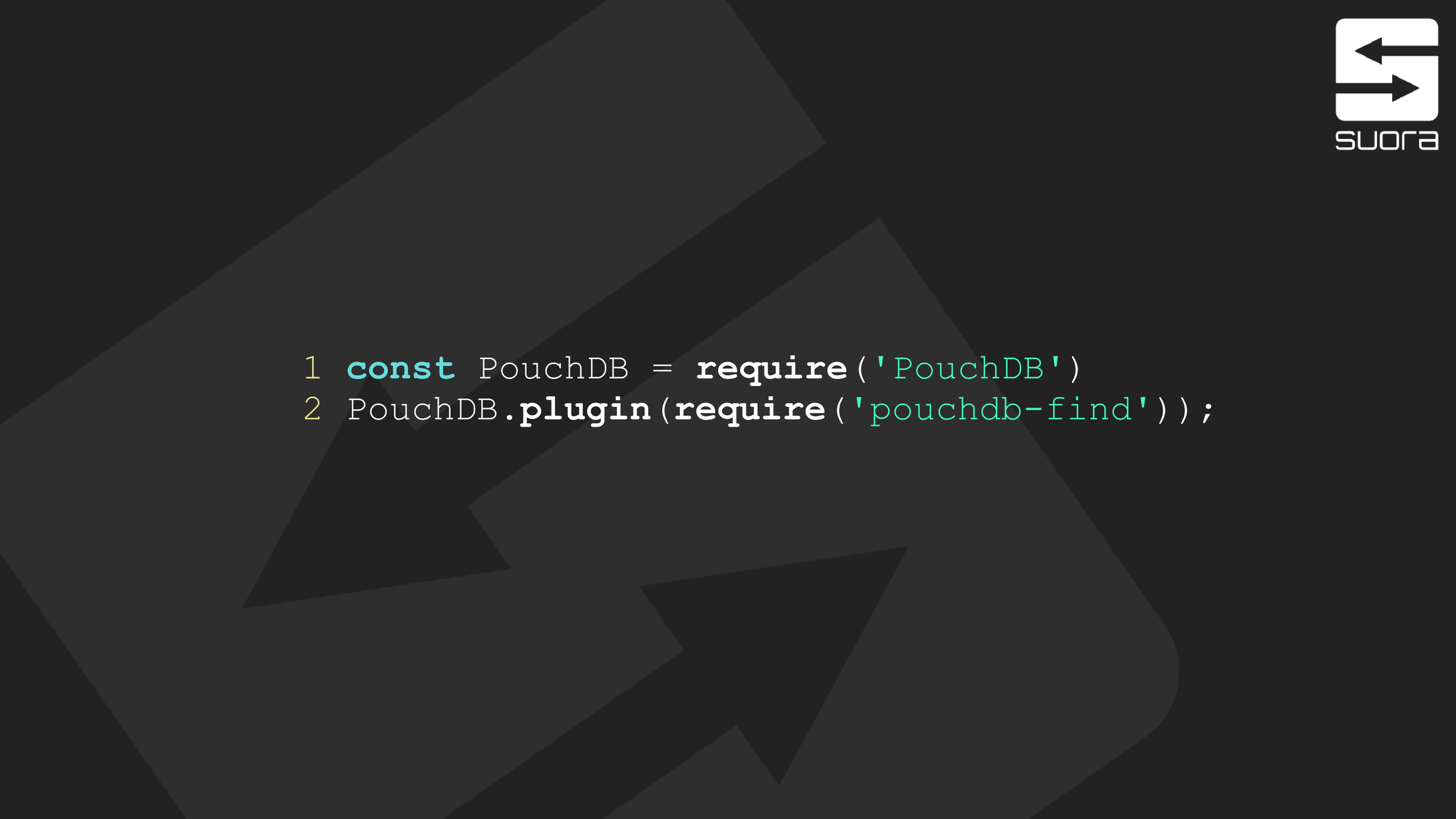
Mango Queries

aka: pouch-find

(aka: have you heard of mongodb?)

Pragmatic: allDocs

pouchdb-find




```
1 const PouchDB = require('PouchDB')
2 PouchDB.plugin(require('pouchdb-find'));
```


```
1  const PouchDB = require('PouchDB')
2  PouchDB.plugin(require('pouchdb-find'));
3
4  const db = new PouchDB('micro-status')
5
6  db.find({
7    selector: {type: 'status'},
8    fields: ['_id', 'text', 'time'],
9  }).then(function (result) {
10   console.dir(result)
11 })
```


```
{ docs:
   [ { _id: 'status-2017-10-09T09:49:30.479Z',
       text: 'Hello World!',
       time: '2017-10-09T09:49:30.479Z' },
     { _id: 'status-2017-10-09T09:54:06.840Z',
       text: 'Hello World!',
       time: '2017-10-09T09:54:06.840Z' },
     { _id: 'test',
       text: 'Hello World!',
       time: '2017-10-09T09:52:06.534Z' } ],
  warning: 'no matching index found, create an index to optimize query time' }
```


```
{ docs:
   [ { _id: 'status-2017-10-09T09:49:30.479Z',
       text: 'Hello World!',
       time: '2017-10-09T09:49:30.479Z' },
     { _id: 'status-2017-10-09T09:54:06.840Z',
       text: 'Hello World!',
       time: '2017-10-09T09:54:06.840Z' },
     { _id: 'test',
       text: 'Hello World!',
       time: '2017-10-09T09:52:06.534Z' } ],
  warning: 'no matching index found, create an index to optimize query time' }
```


```
1  const PouchDB = require('PouchDB')
2  PouchDB.plugin(require('pouchdb-find'))
3
4  const db = new PouchDB('micro-status')
5
6  db.createIndex({
7    index: {
8      fields: ['type', 'time']
9    }
10 })
```

PouchDB-find has more features!


```
1  const PouchDB = require('PouchDB')
2  PouchDB.plugin(require('pouchdb-find'));
3
4  const db = new PouchDB('micro-status')
5
6  db.find({
7    selector: {type: 'status', time: {'$gte':
'2017-10-09T09:54:06.840Z'}},
8    fields: ['_id', 'text', 'time'],
9    sort: ['time']
10 }).then(function (result) {
11   console.dir(result)
12 })
```

Changes Feed


```
1  const PouchDB = require('PouchDB')
2
3  const db = new PouchDB('micro-status')
4  var changes = db.changes({
5    since: 'now',
6    live: true,
7    include_docs: true
8  }).on('change', function(change) {
9    console.dir(change, {depth: null})
10 })
11
```

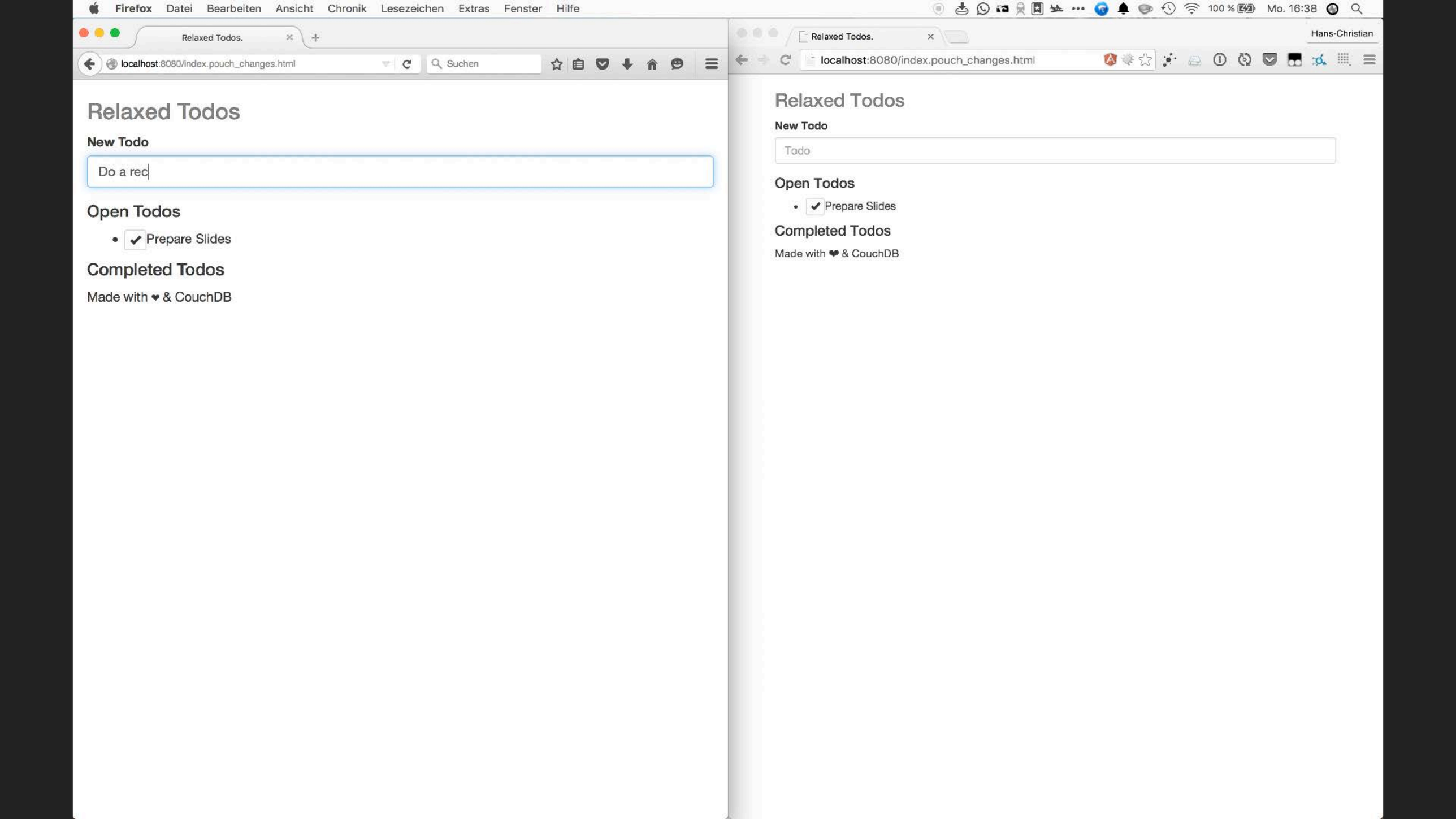

```
1 { id: 'status-2017-10-09T10:16:55.556Z',
2   changes: [ { rev: '1-
bb476e6b29b244c38e0acdce41a38168' } ],
3   doc:
4    { type: 'status',
5      text: 'Hello World!',
6      time: '2017-10-09T10:16:55.556Z',
7      _id: 'status-2017-10-09T10:16:55.556Z',
8      _rev: '1-bb476e6b29b244c38e0acdce41a38168' },
9   seq: 8 }
```

The Perfect database for PWAs?

me, a few minutes ago.


suora

Meet CouchDB


Firefox Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Fenster Hilfe
100 % Mo. 16:38
Relaxed Todos.
localhost:8080/index.pouch_changes.html
Suchen
Relaxed Todos
New Todo
Do a rec
Open Todos
Prepare Slides
Completed Todos
Made with ❤ & CouchDB
Relaxed Todos.
Hans-Christian
localhost:8080/index.pouch_changes.html
Relaxed Todos
New Todo
Todo
Open Todos
Prepare Slides
Completed Todos
Made with ❤ & CouchDB

Meet CouchDB

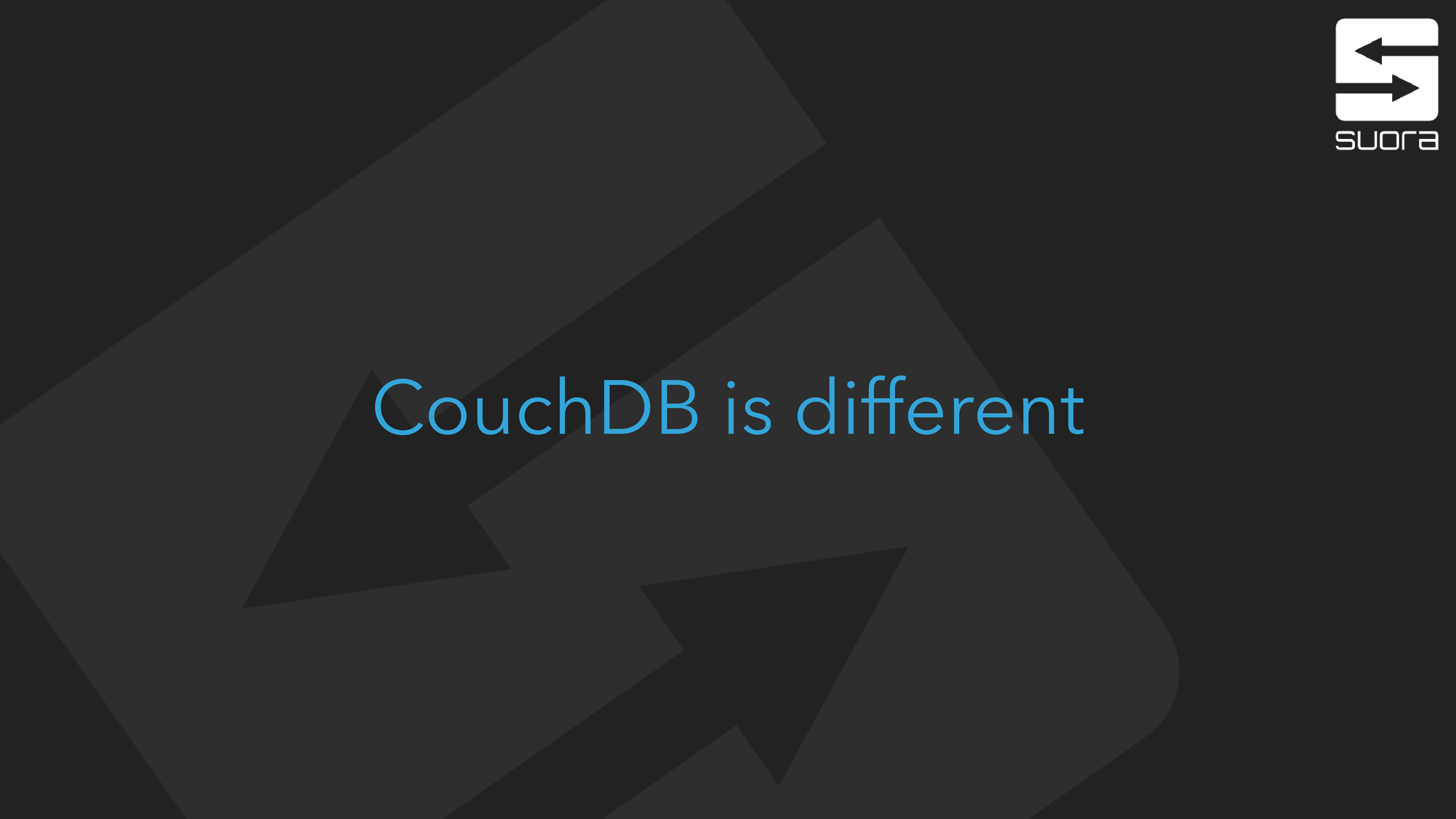
Database for the „cloud“

… or bare metal

NoSQL

Open Source
(Apache 2.0 licensed)

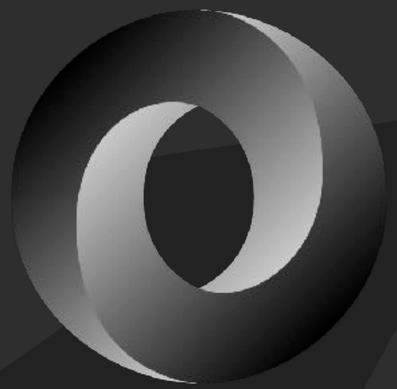
Written in Erlang



CouchDB is different

suora

There are no tables.

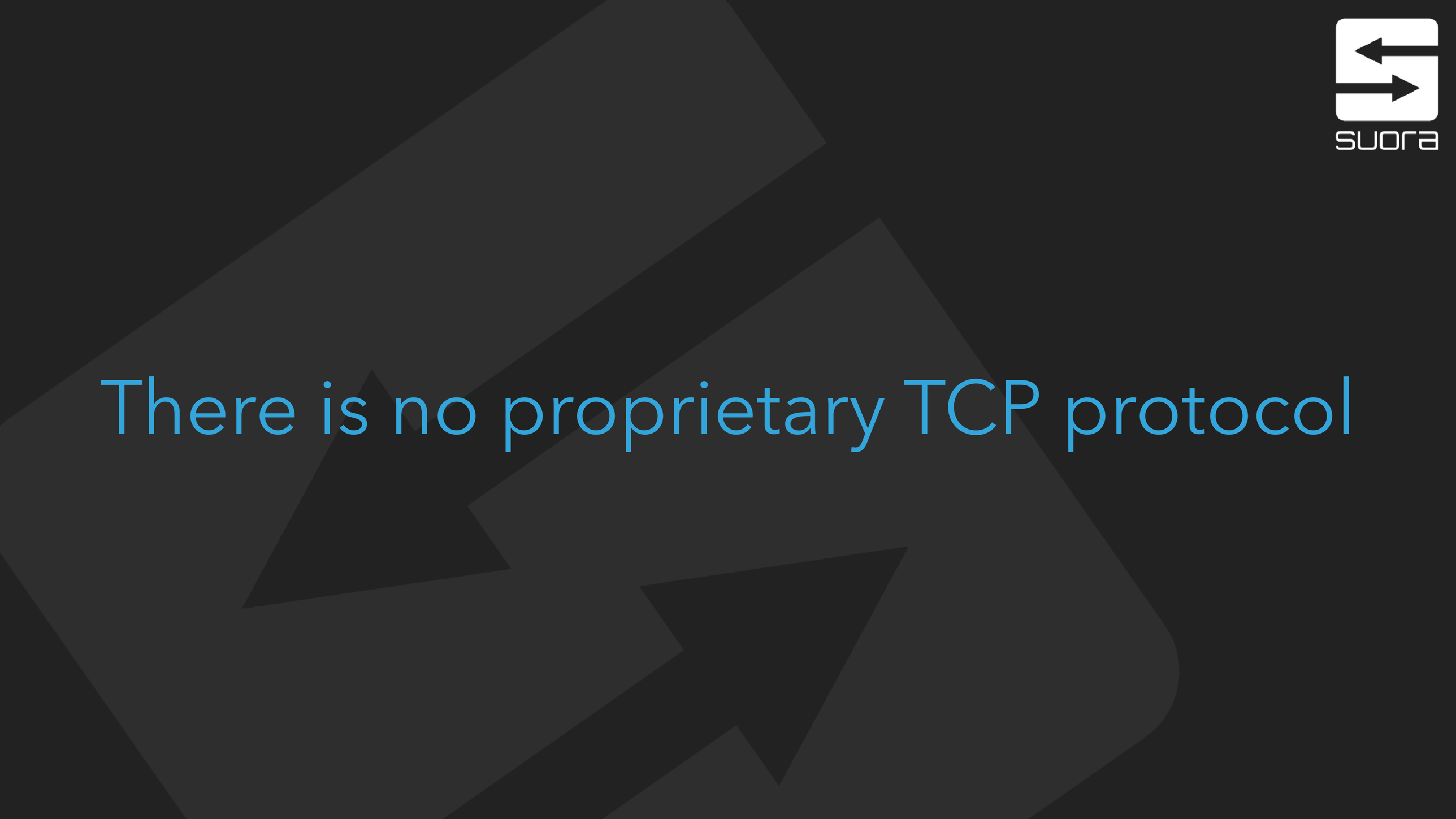


There is no „SQL“

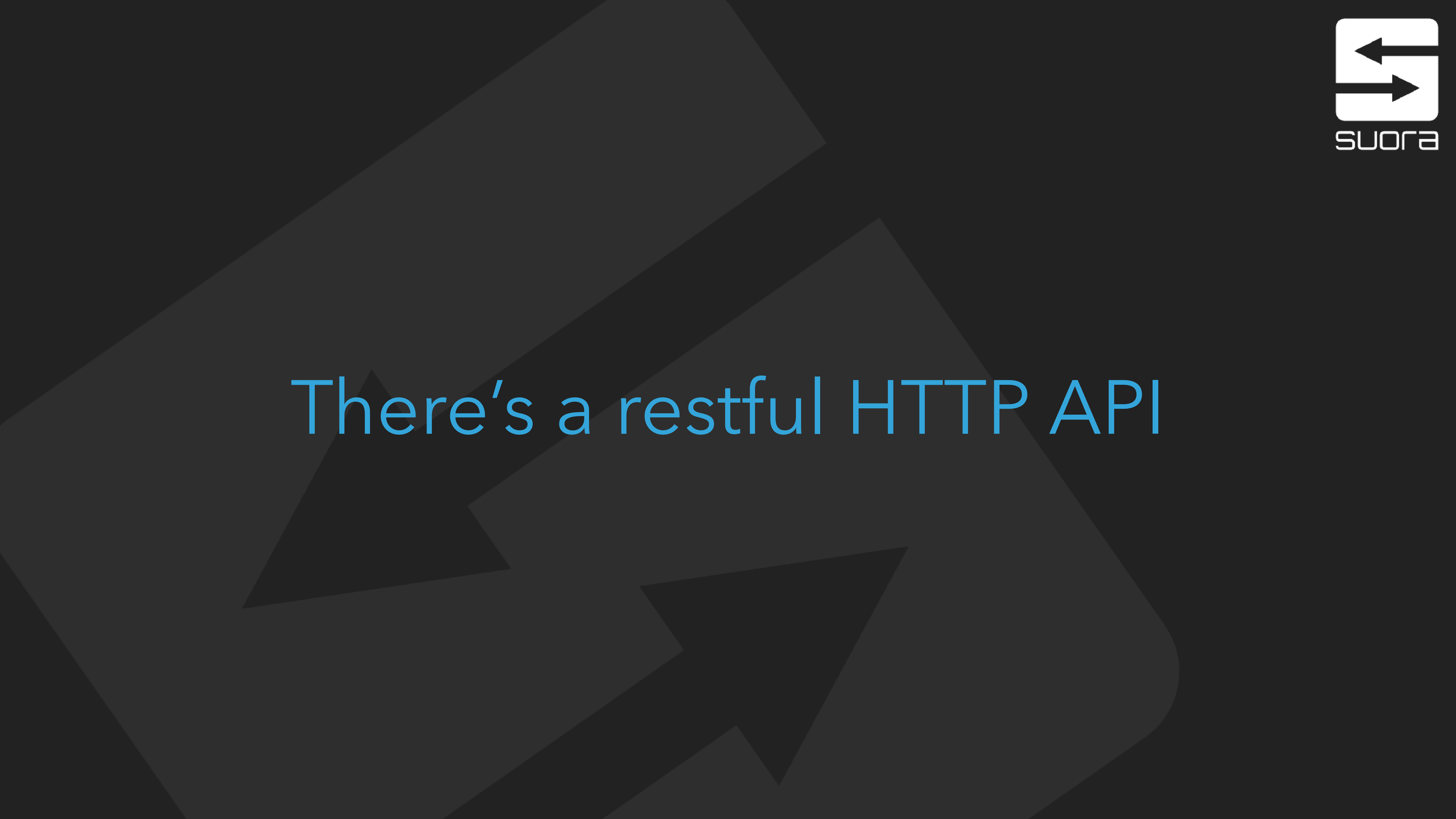
There is no „Consistency“

CouchDB is eventual consistent

There is no relational integrity.



There is no proprietary TCP protocol



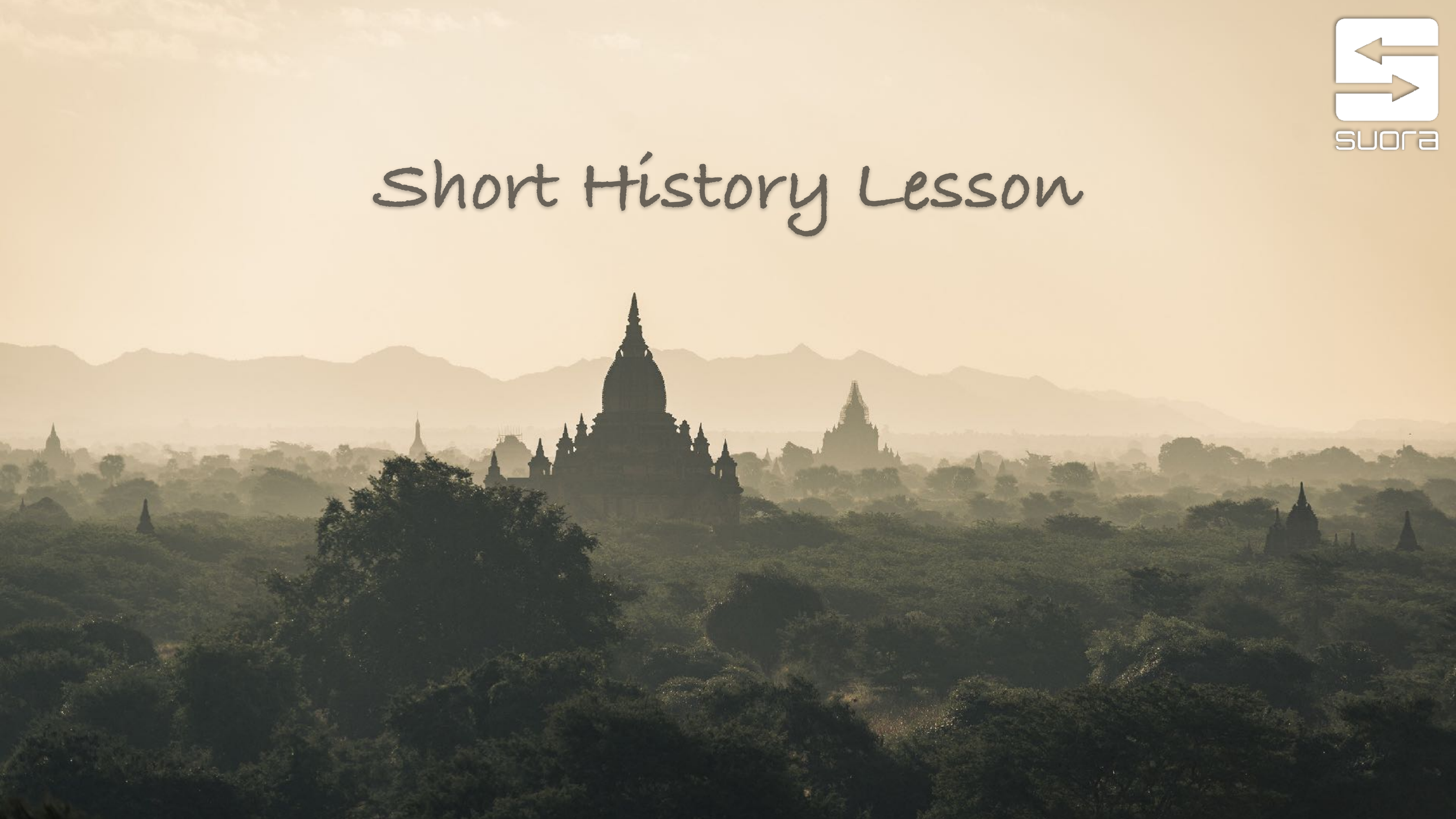
There’s a restful HTTP API

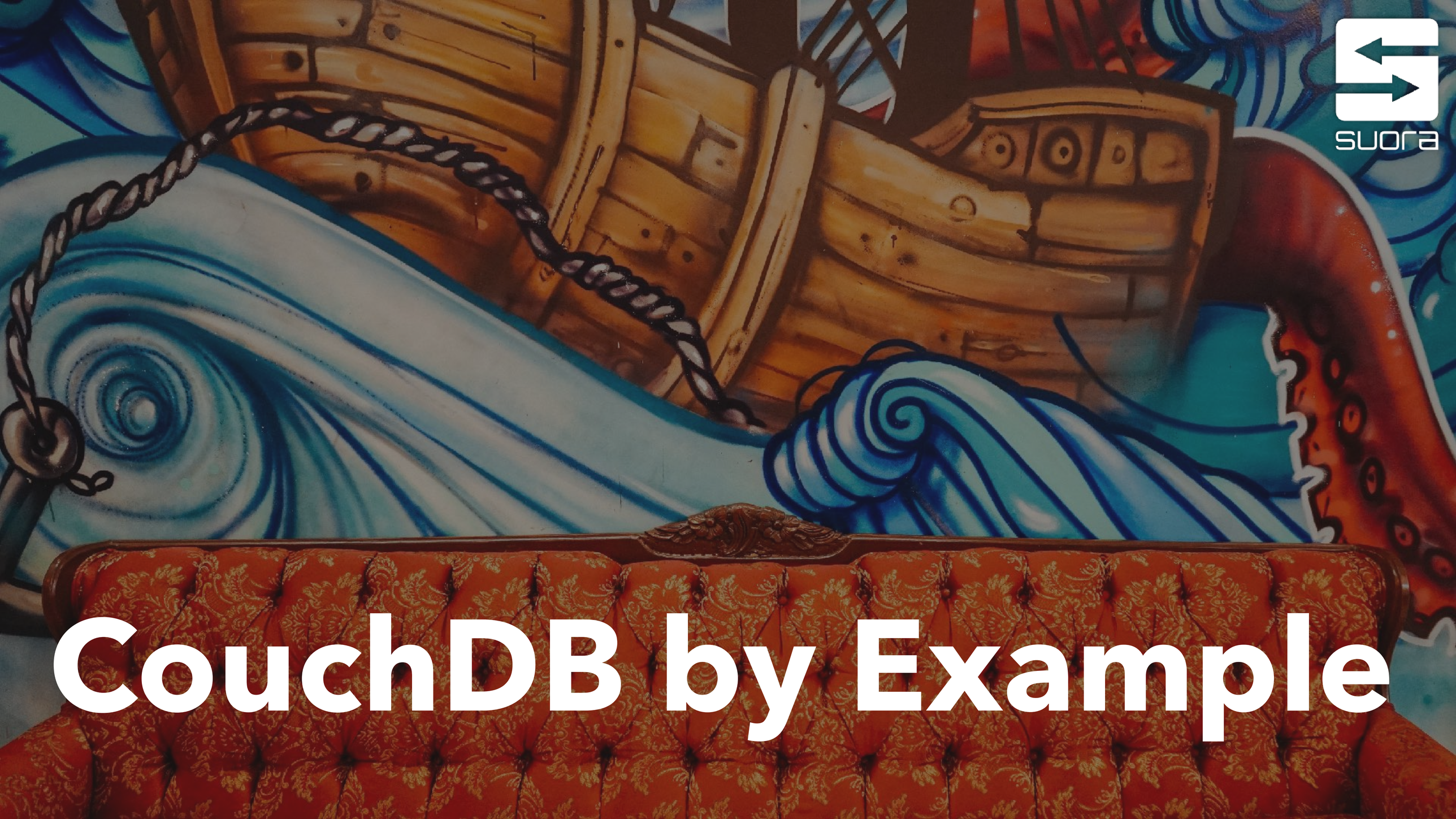
suora

CouchDB is always my favourite example of a great REST implementation 💖

Ole Michaelis
@OleMchls

Short History Lesson





CouchDB by Example

suora


```
hco@hco-mbp ~ $ http -j PUT http://localhost:5984/jsdays
HTTP/1.1 201 Created
Cache-Control: must-revalidate
Content-Length: 12
Content-Type: application/json
Date: Fri, 02 Oct 2015 18:52:43 GMT
Location: http://localhost:5984/jsdays
Server: CouchDB/1.6.1 (Erlang OTP/17)

{
    "ok": true
}
```

```
hco@hco-mbp ~ $ http -j -a […] PUT :5984/jsdays/test foo=bar
```

```
{
    "foo": "bar"
}
```

suora

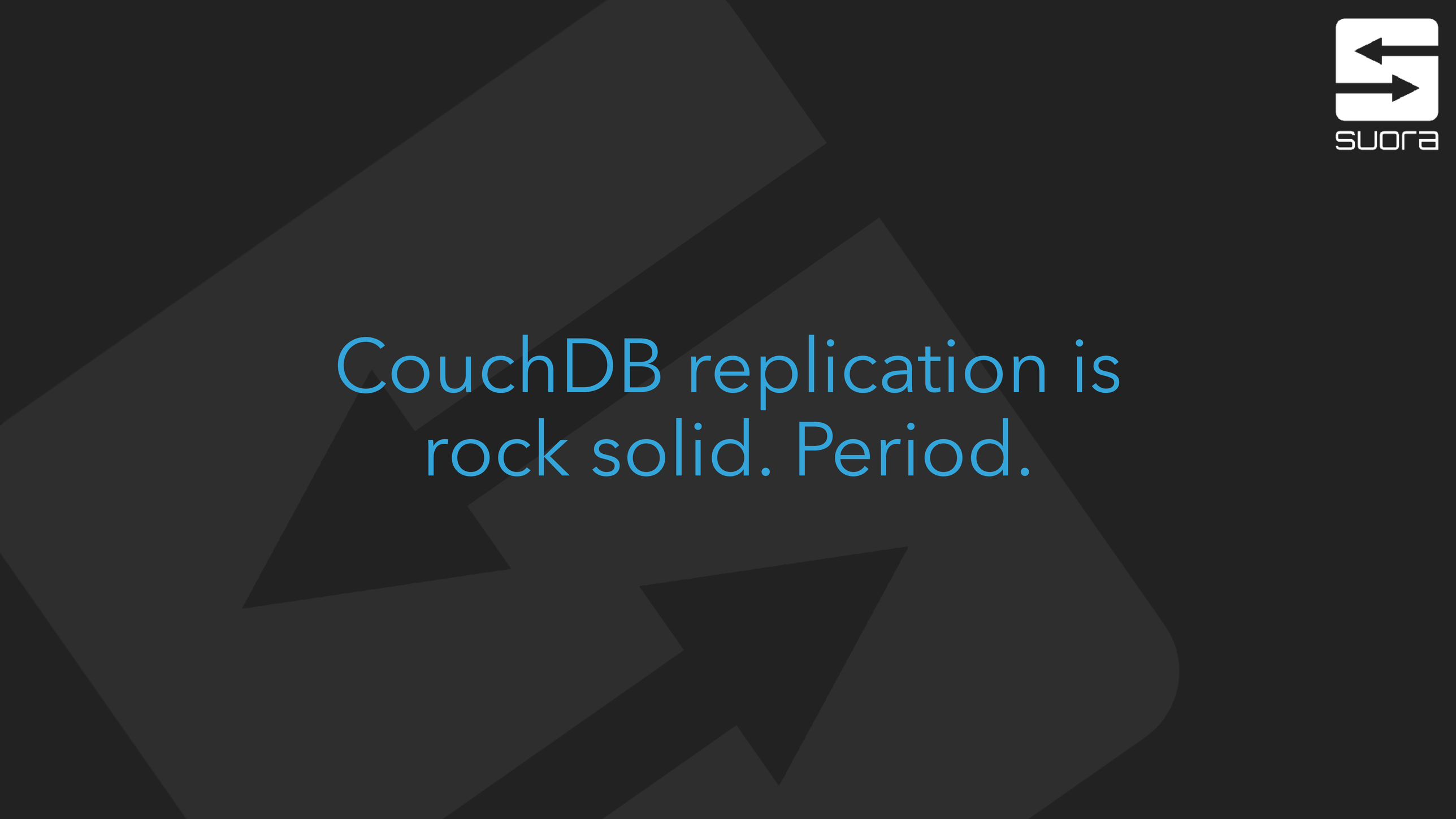

```
hco@hco-mbp ~ $ http -j -a […] PUT :5984/jsdays/test foo=bar
HTTP/1.1 201 Created
Cache-Control: must-revalidate
Content-Length: 67
Content-Type: application/json
Date: Fri, 02 Oct 2015 19:24:10 GMT
ETag: "1-4c6114c65e295552ab1019e2b046b10e"
Location: http://localhost:5984/jsdays/test
Server: CouchDB/1.6.1 (Erlang OTP/17)

{
    "id": "test",
    "ok": true,
    "rev": "1-4c6114c65e295552ab1019e2b046b10e"
}
```

```
hco@hco-mbp ~ $ http -j -a […] GET :5984/jsdays/test
```


```
hco@hco-mbp ~ $ http -j -a […] GET :5984/jsdays/test
HTTP/1.1 200 OK
Cache-Control: must-revalidate
Content-Length: 71
Content-Type: application/json
Date: Fri, 02 Oct 2015 19:27:52 GMT
ETag: "1-4c6114c65e295552ab1019e2b046b10e"
Server: CouchDB/1.6.1 (Erlang OTP/17)

{
    "_id": "test",
    "_rev": "1-4c6114c65e295552ab1019e2b046b10e",
    "foo": "bar"
}
```

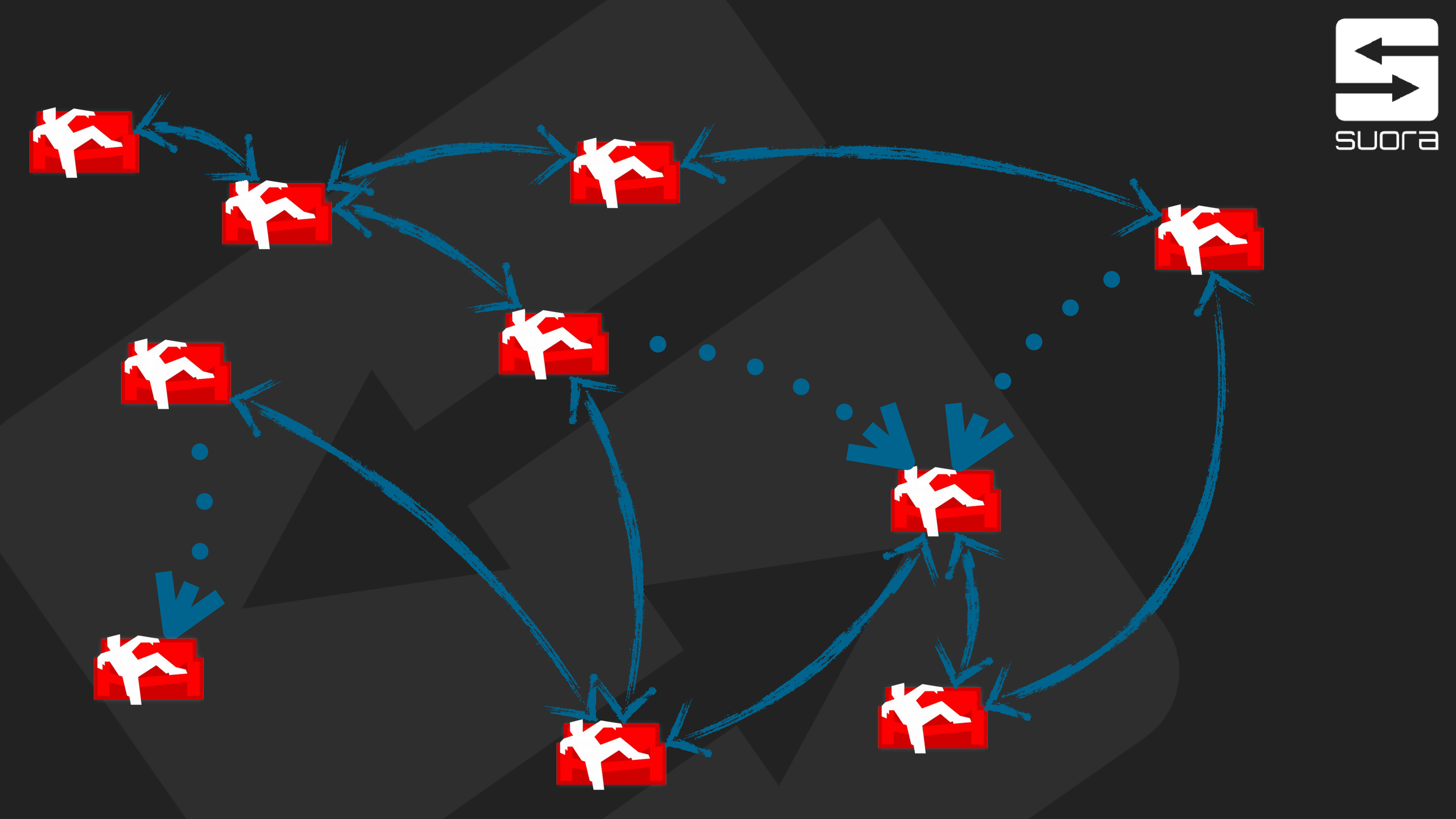
CouchDB sucks at everything
Except sync.
And incidentally, sync is the most important feature a developer cares about in the future.

Maybe by Jason Smith

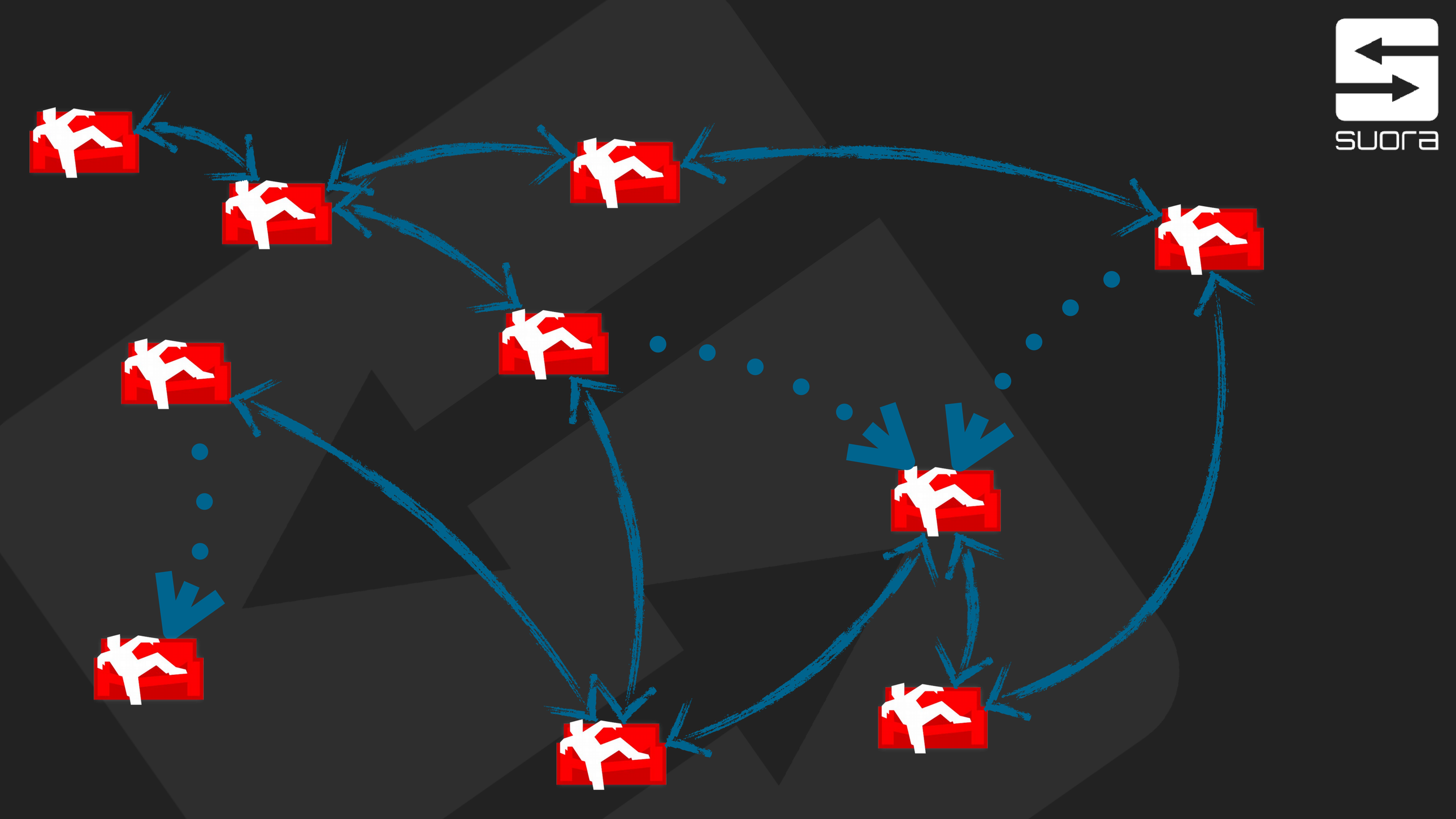


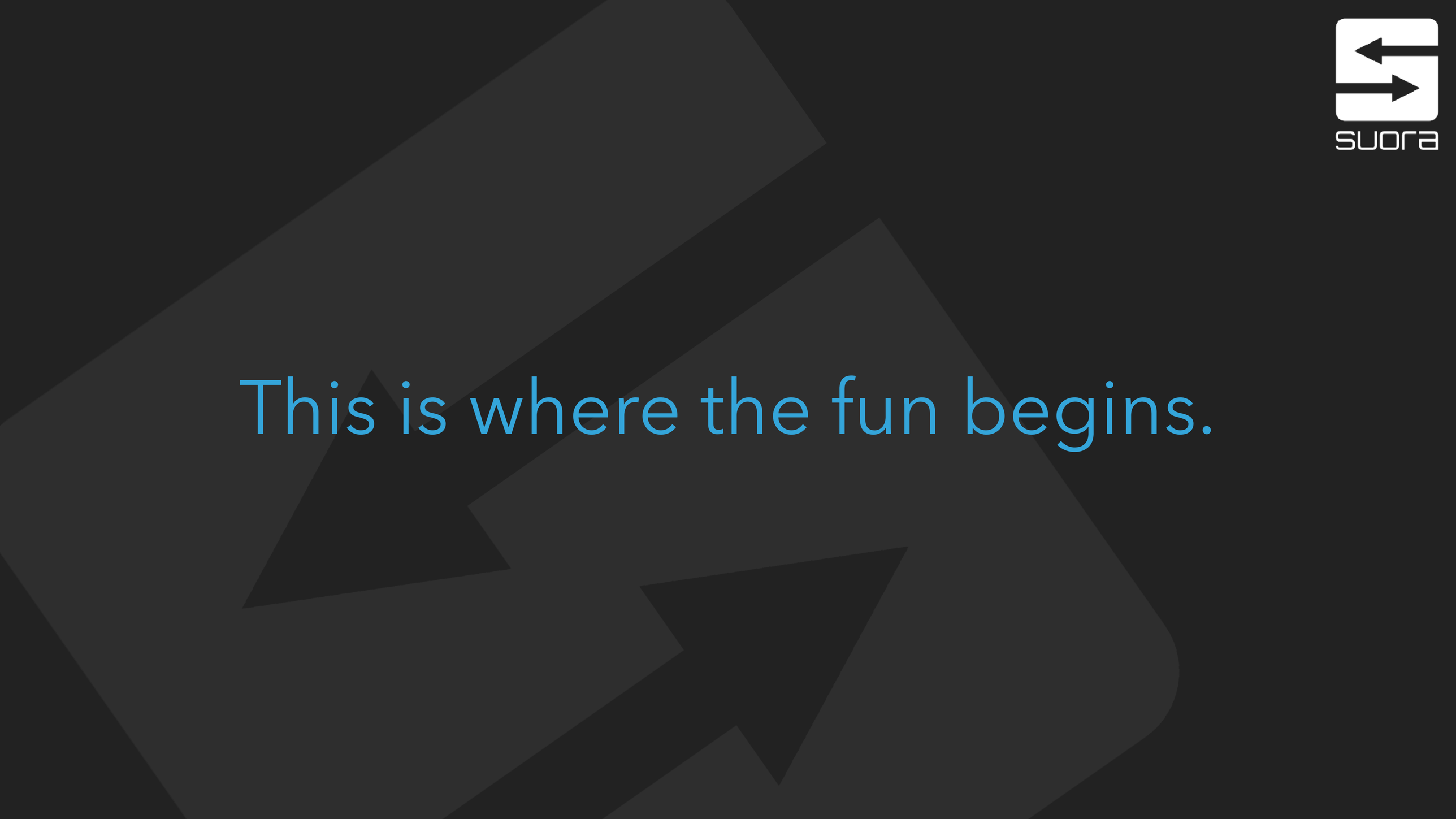
CouchDB replication is
rock solid. Period.


suora

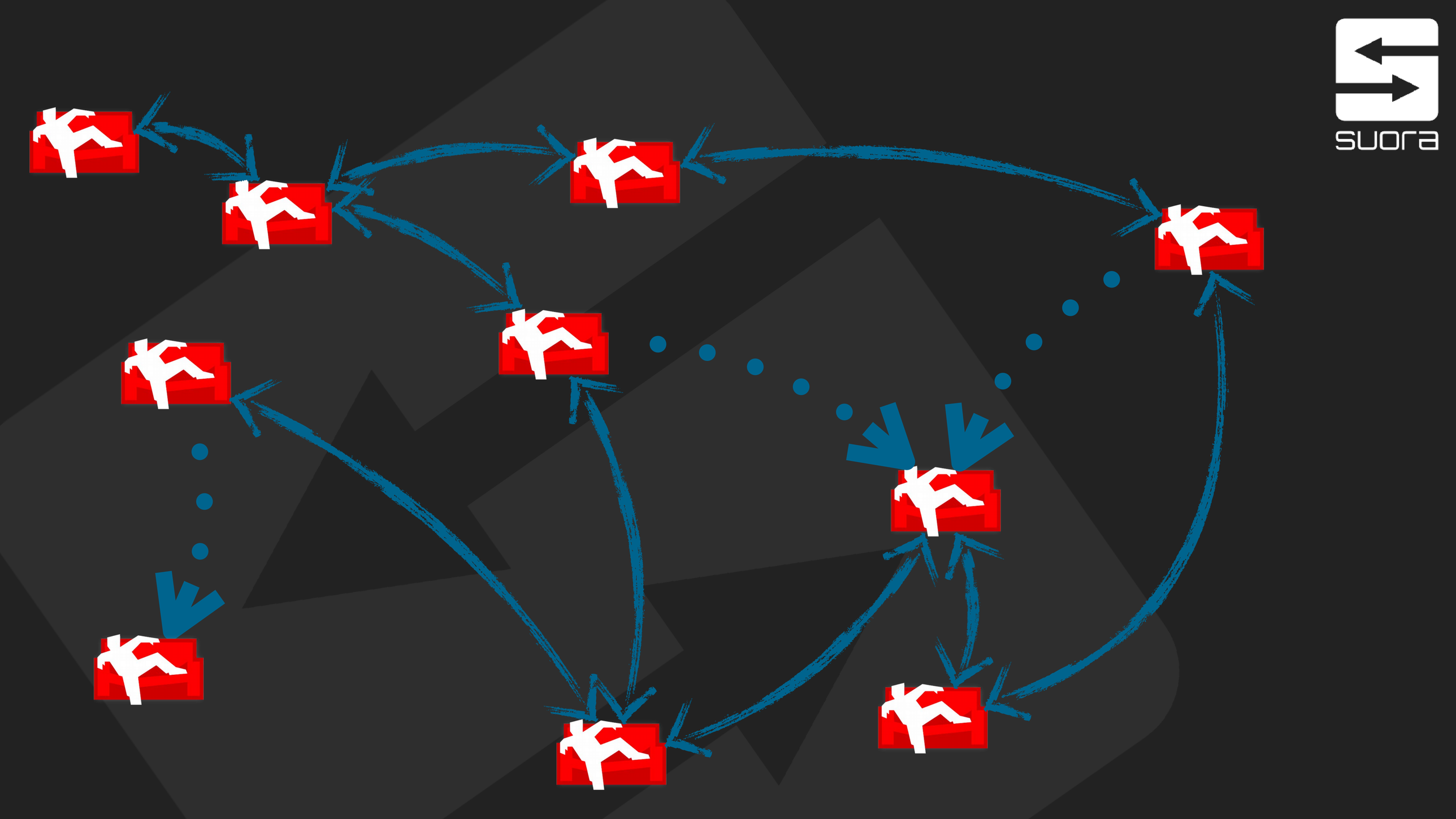

suora

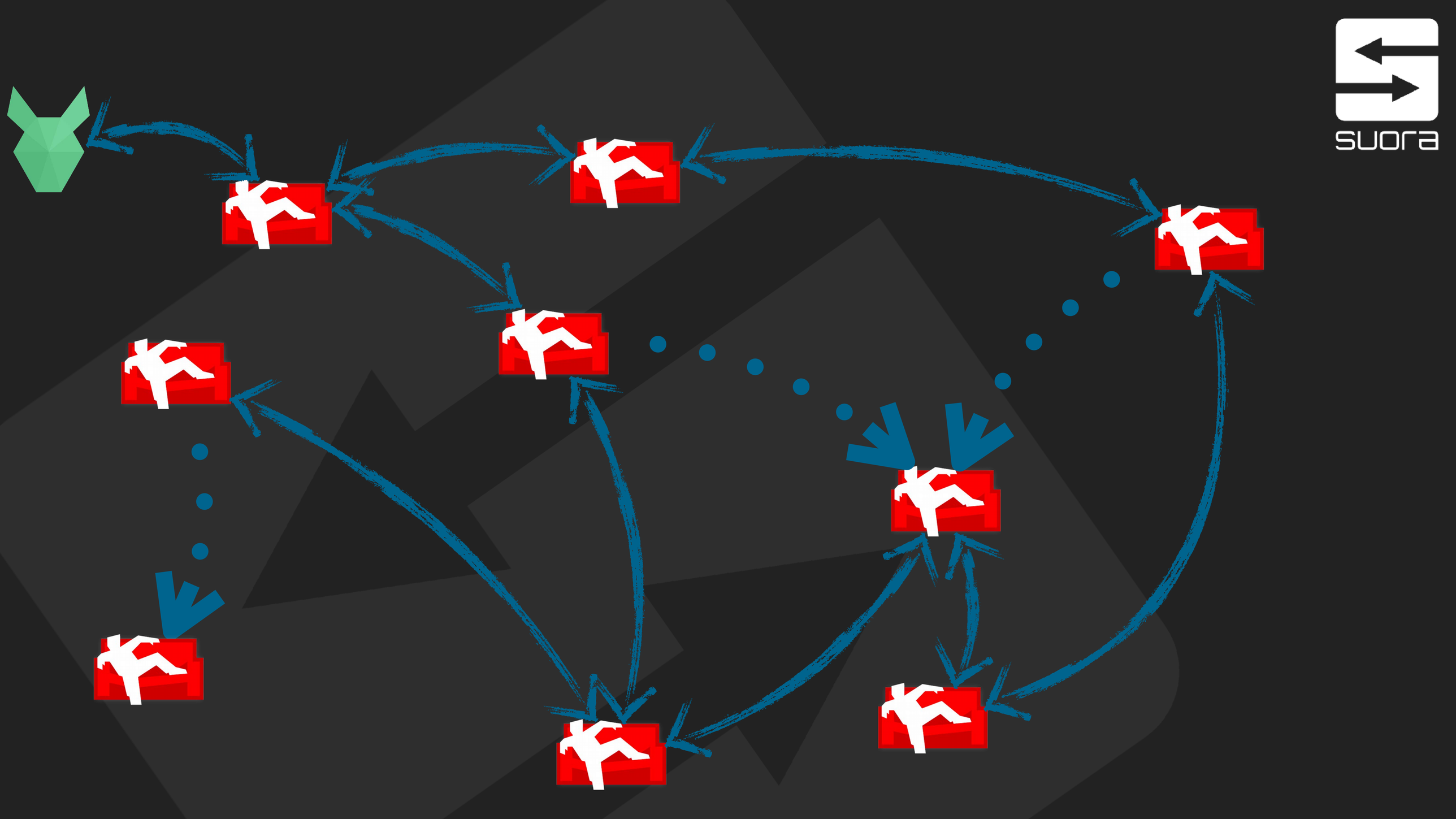

It’s like Blockchain!
Just cooler!
suora


suora

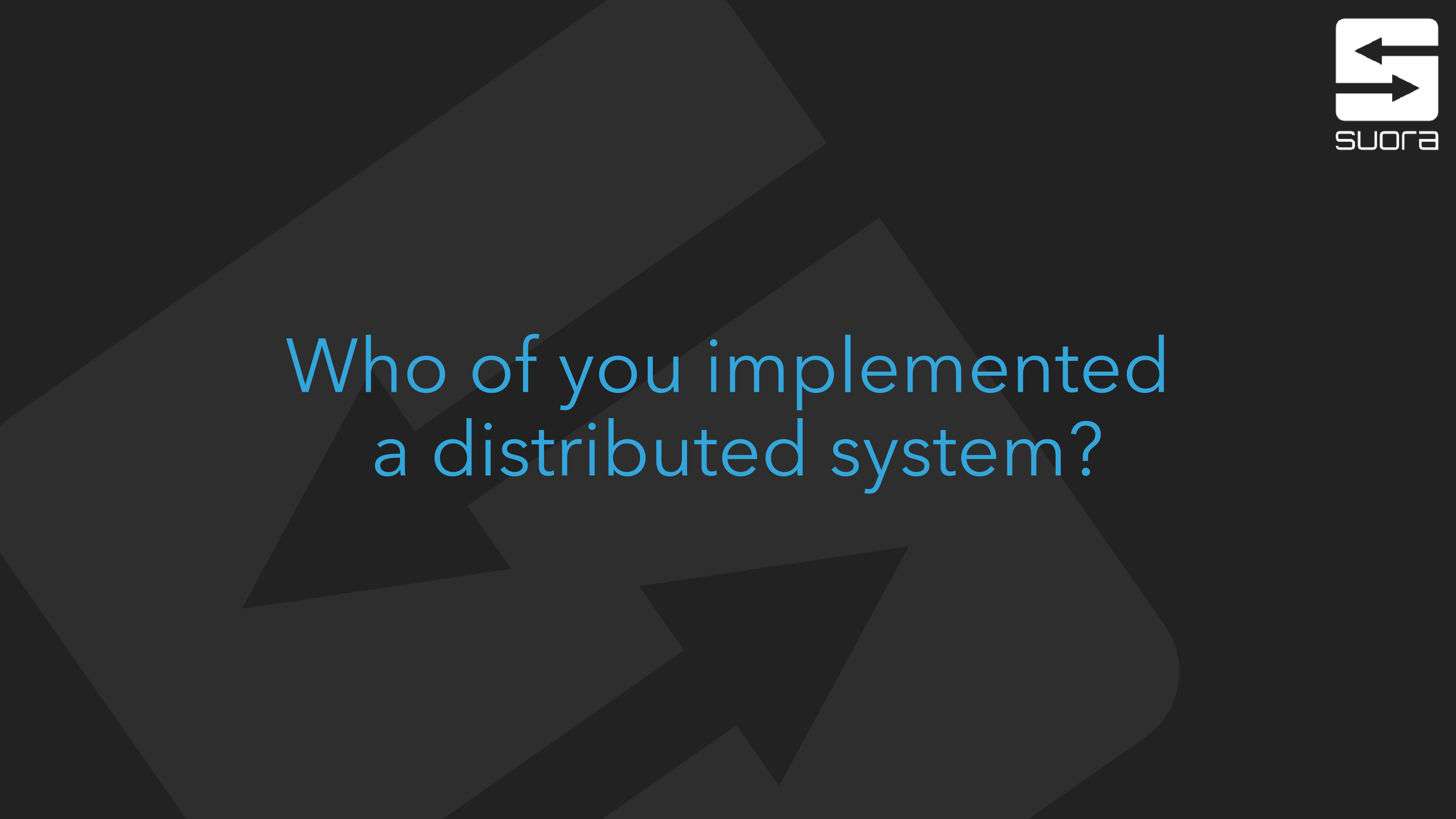


This is where the fun begins.

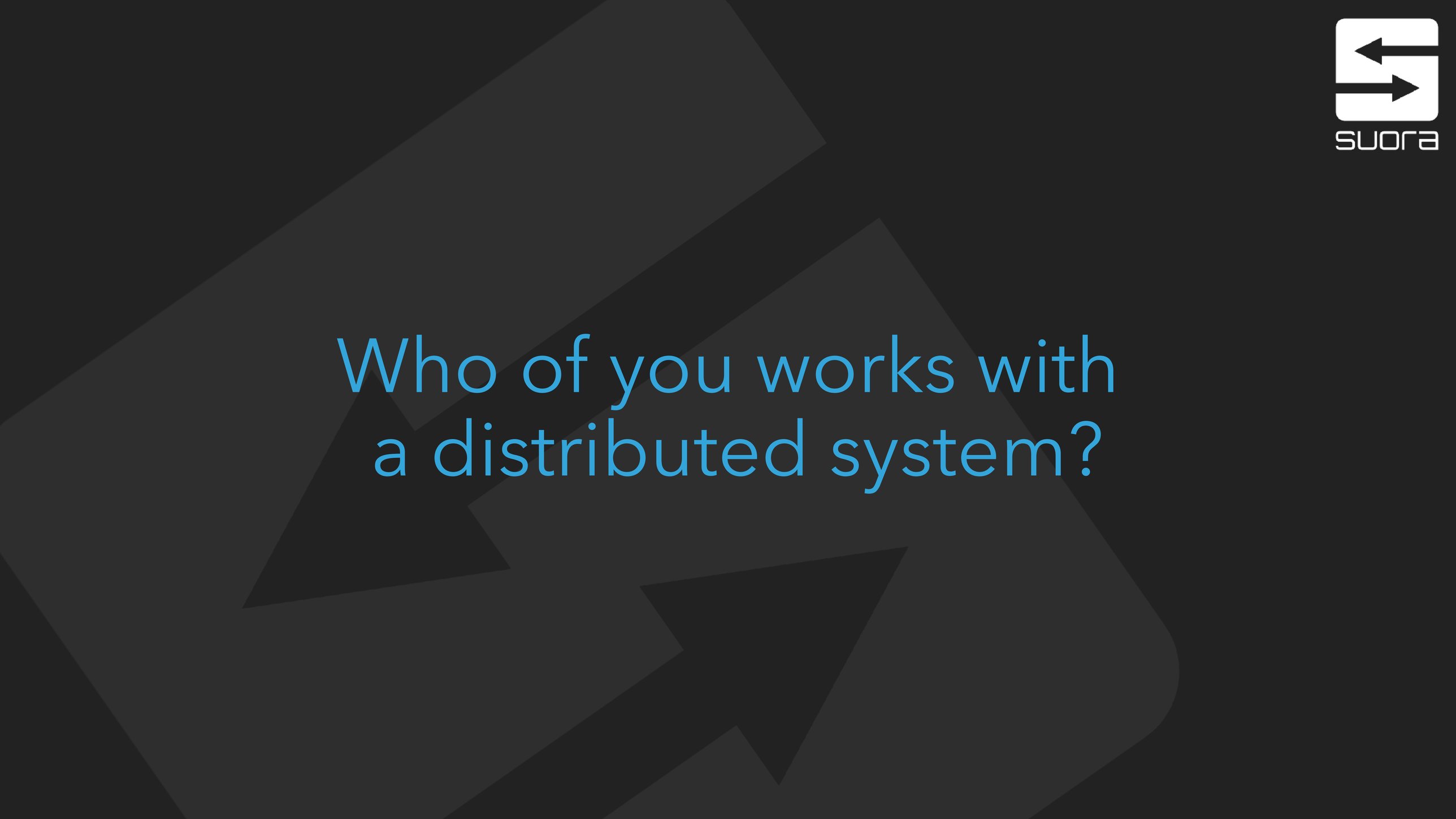

suora


suora

Multi-Master Replication \o/

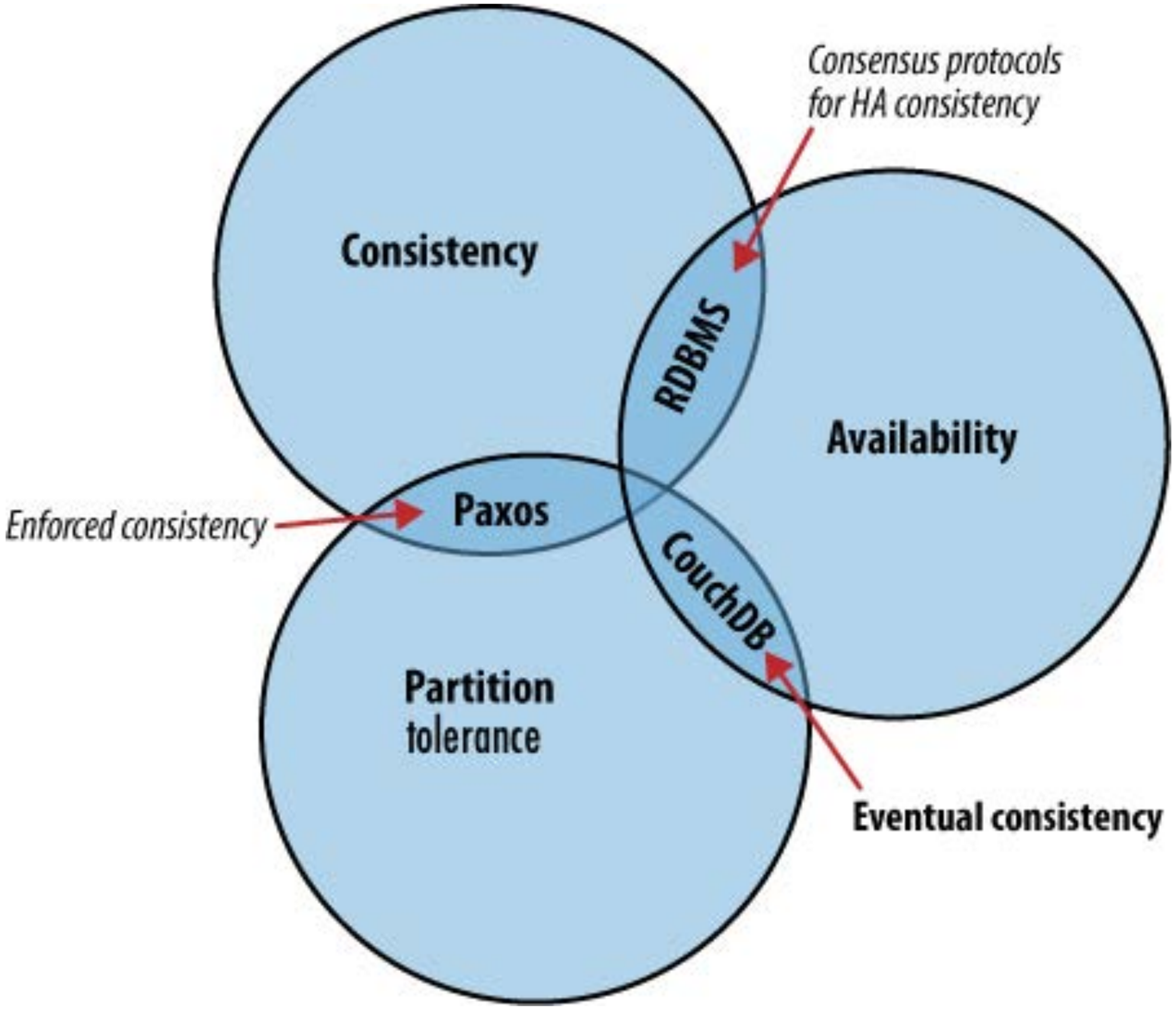


Who of you implemented
a distributed system?

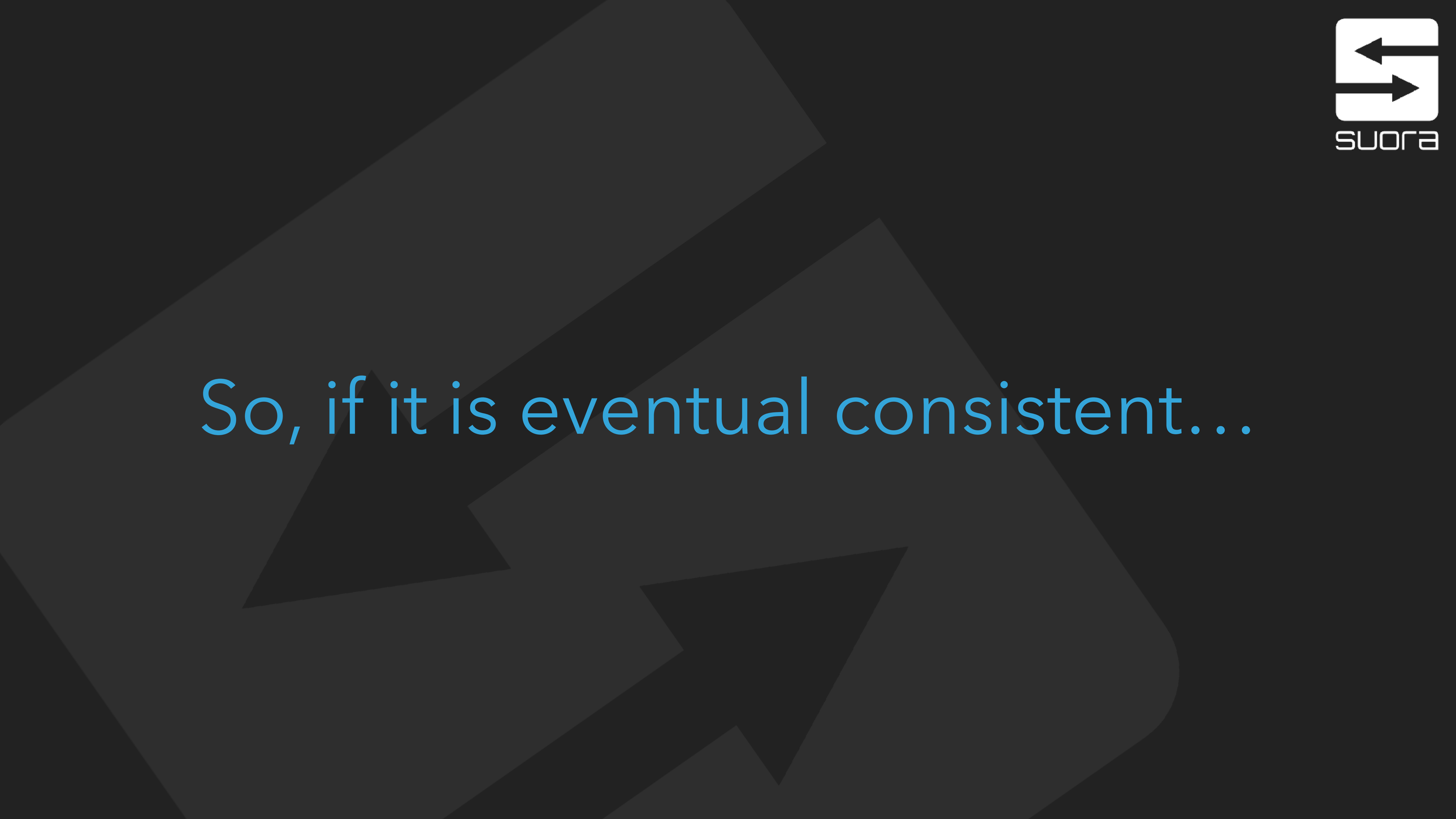


Who of you works with
a distributed system?

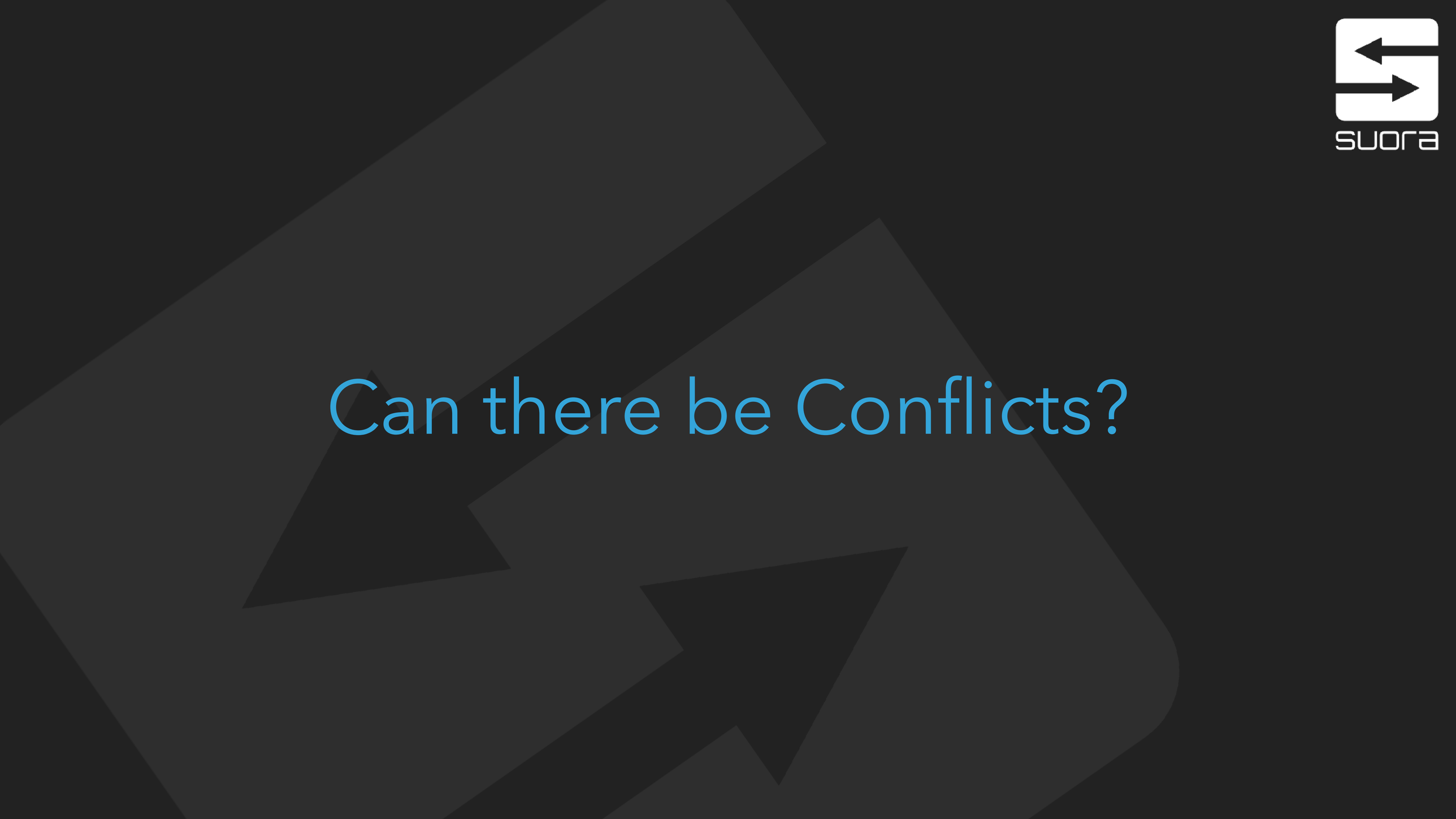
The CAP-Theorem



http://guide.couchdb.org/draft/consistency.html



So, if it is eventual consistent…



Can there be Conflicts?

Yes.

But they don’t hurt.

(unless they do)

Conflicts in CouchDB


```
1 ServerBootstrap bootstrap = new ServerBootstrap(
2     new NioServerSocketChannelFactory(
3         Executors.newCachedThreadPool(),
4         Executors.newCachedThreadPool()));
5
6 // Set up the pipeline factory.
7 bootstrap.setPipelineFactory(new ChannelPipelineFactory() {
8   public ChannelPipeline getPipeline() throws Exception {
9     return Channels.pipeline(new ObjectEncoder(),
10        new ObjectDecoder(), new ServerHandler());
11  }
12 });
13 bootstrap.bind(new InetSocketAddress(1111));
```


```
1 const PouchDB = require('PouchDB')
2
3 const db = new PouchDB('micro-status')
4 db.sync("http://localhost:5984/micro-status")
```

suora

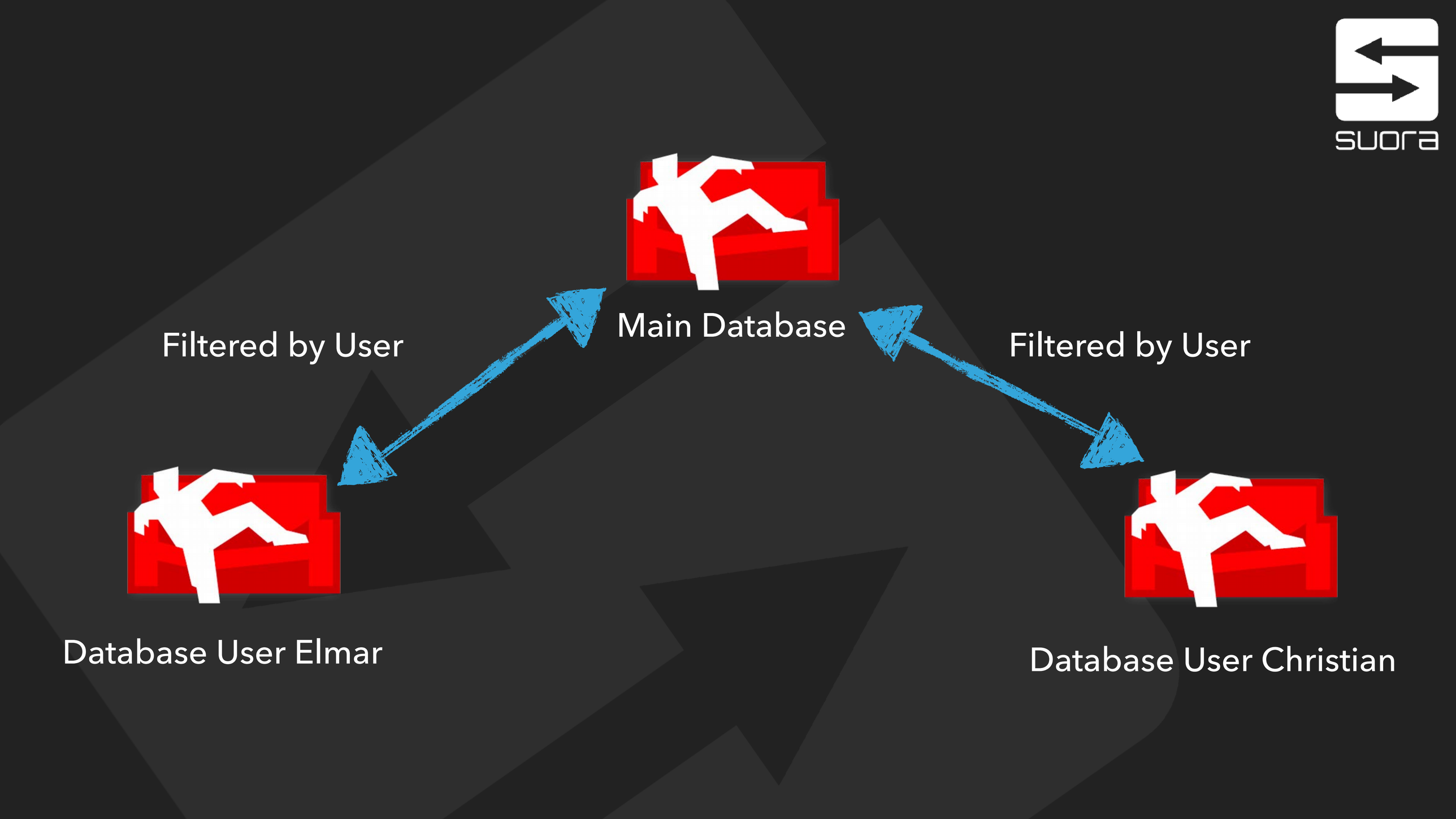
But should I expose my CouchDB to the Internet?

Maybe, but…

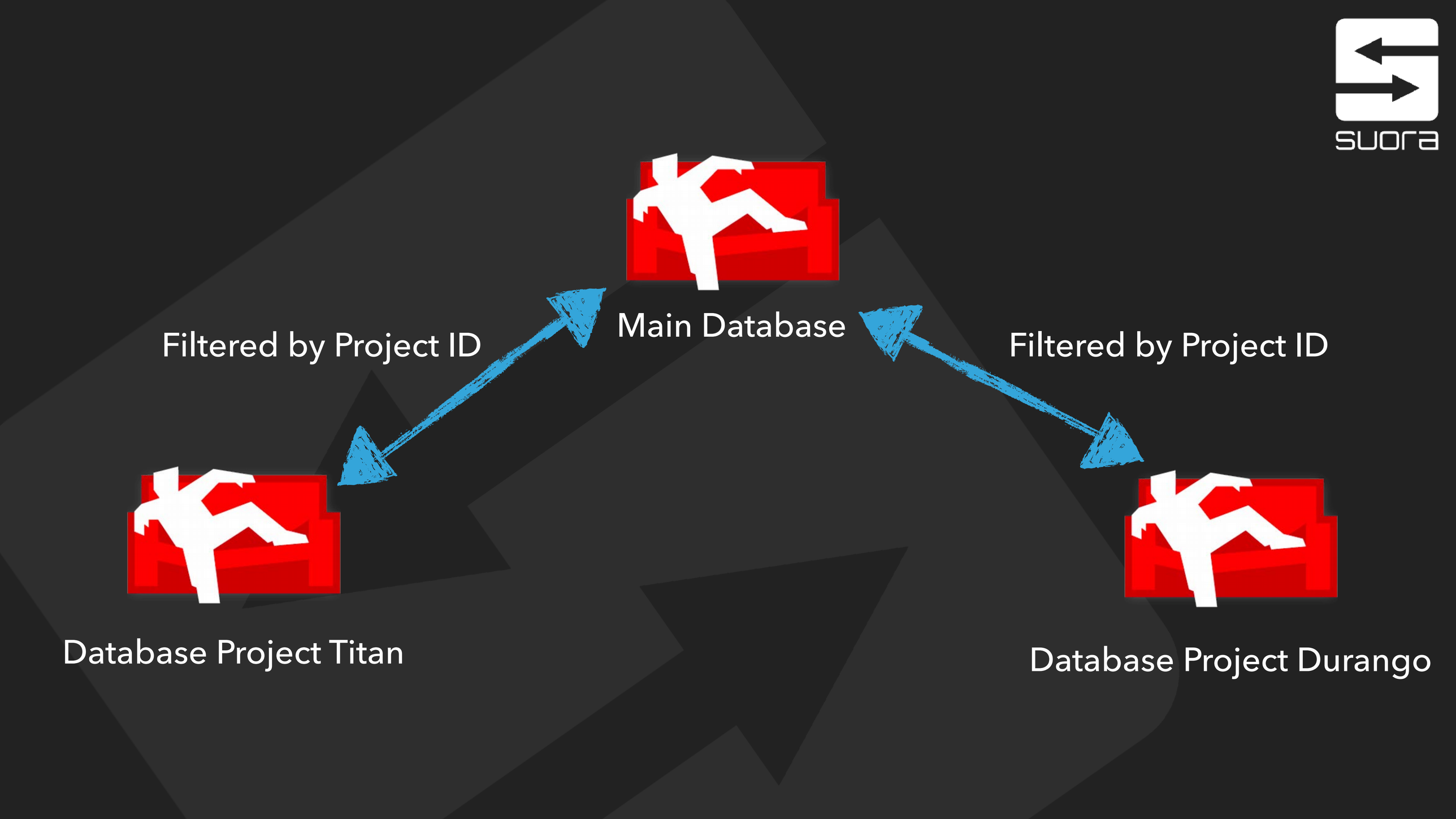
Read-Permissions
can only be granted per database

Write-Permissions
can be controlled by a JS function

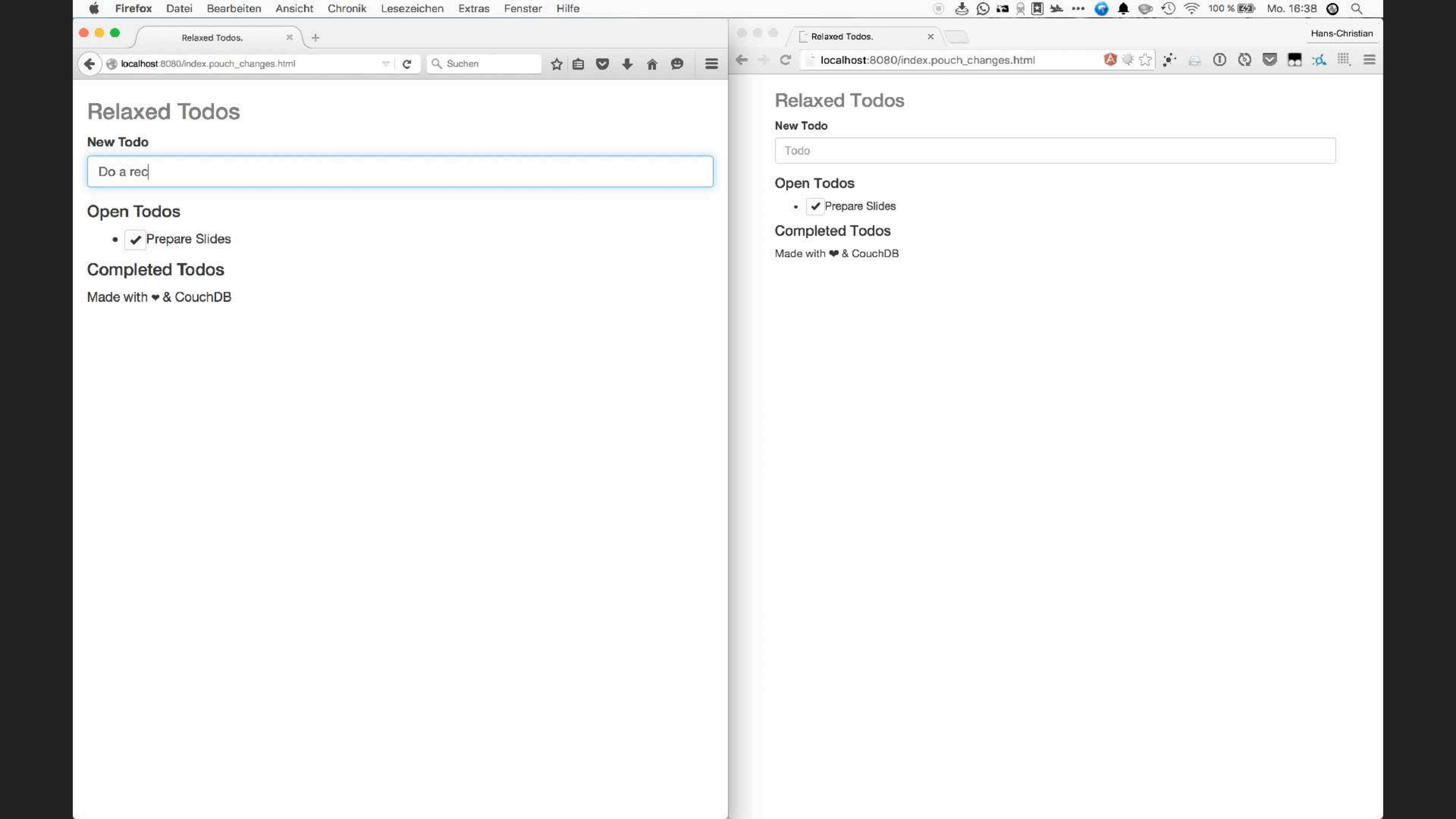
„Database per User“-Pattern


Main Database
Filtered by User
Filtered by User
Database User Elmar
Database User Christian

„Database per Project“-Pattern

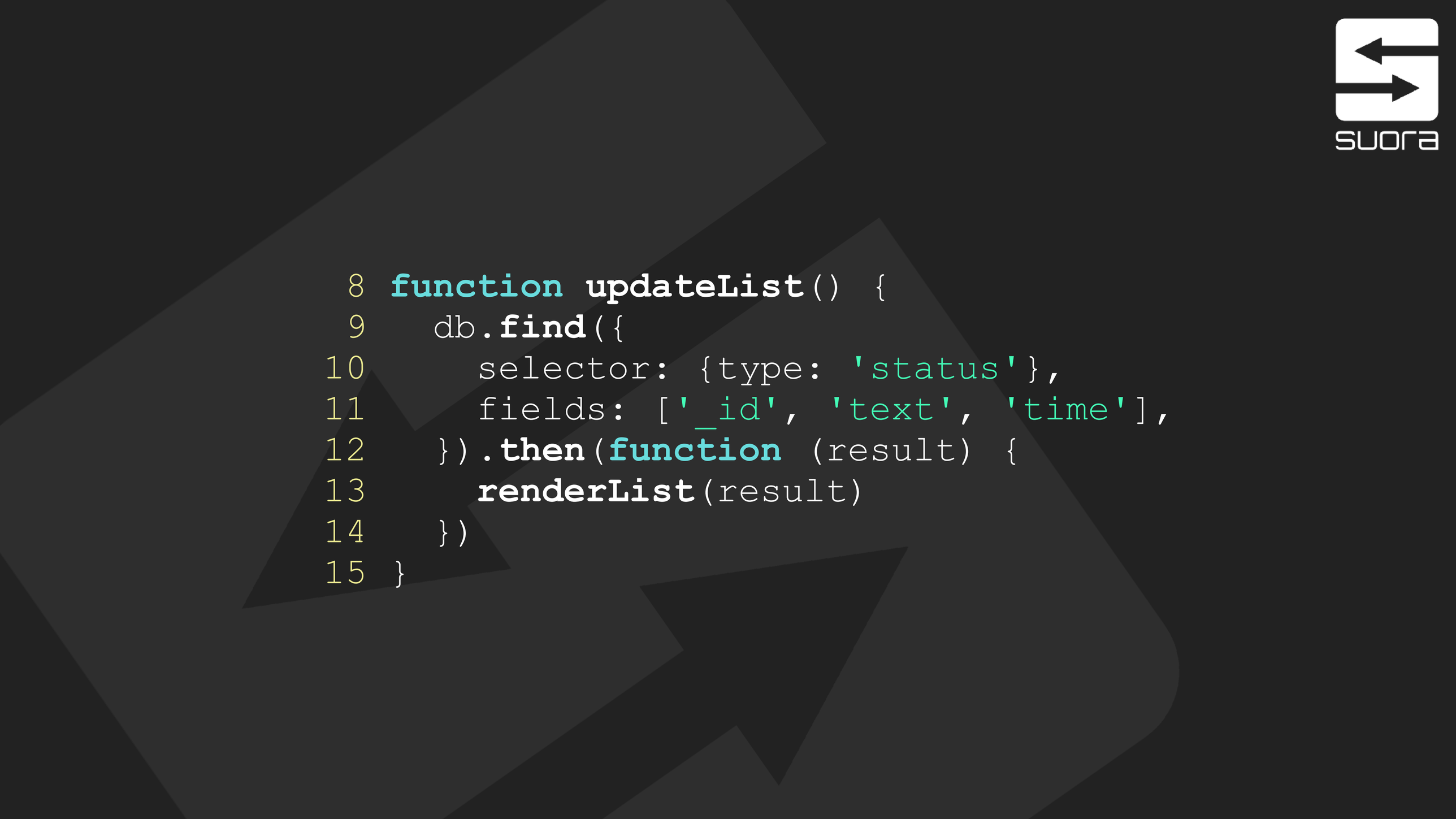

suora
Main Database
Filtered by Project ID
Filtered by Project ID
Database Project Titan
Database Project Durango

Now let’s get back to the PWA


Firefox Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Fenster Hilfe
100 % Mo. 16:38
Relaxed Todos.
localhost:8080/index.pouch_changes.html
Suchen
Relaxed Todos
New Todo
Do a rec
Open Todos
Prepare Slides
Completed Todos
Made with ❤ & CouchDB
Relaxed Todos.
Hans-Christian
localhost:8080/index.pouch_changes.html
Relaxed Todos
New Todo
Todo
Open Todos
Prepare Slides
Completed Todos
Made with ❤ & CouchDB

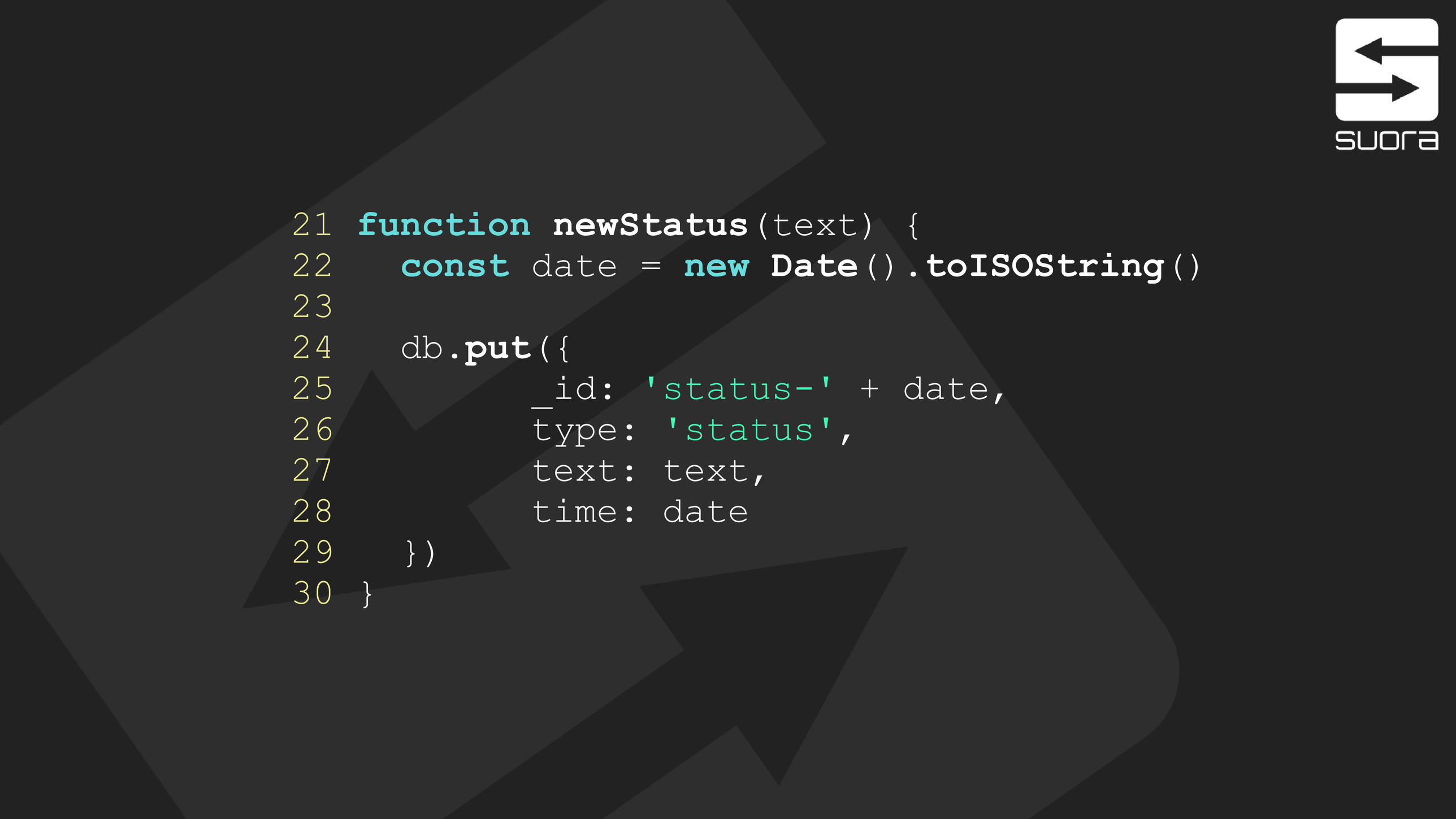
```
const PouchDB = require('PouchDB')
PouchDB.plugin(require('pouchdb-find'));

const db = new PouchDB('micro-status')

db.sync("http://localhost:5984/micro-status")

function updateList() {
  db.find({
    selector: {type: 'status'},
    fields: ['_id', 'text', 'time'],
  }).then(function (result) {
    renderList(result)
  })
}

var changes = db.changes({since: 'now', live: true})
  .on('change', updateList);


function newStatus(text) {
  const date = new Date().toISOString()

  db.put({
        _id: 'status-' + date,
        type: 'status',
        text: text,
        time: date
  })
}
```


```
1 const PouchDB = require('PouchDB')
2 PouchDB.plugin(require('pouchdb-find'));
3
4 const db = new PouchDB('micro-status')
5
6 db.sync("http://localhost:5984/micro-status")
```



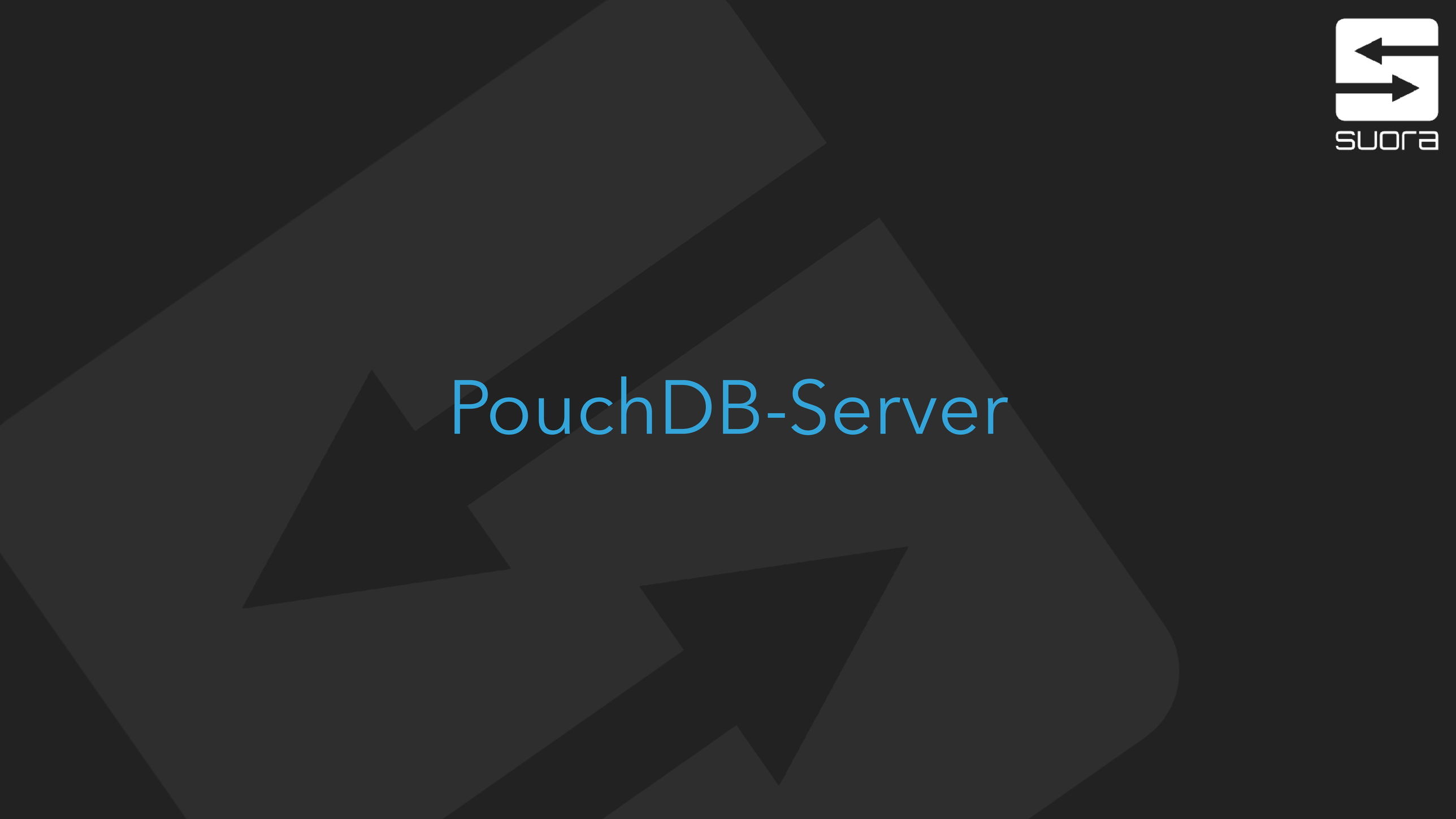

```
 8 function updateList() {
 9   db.find({
10     selector: {type: 'status'},
11     fields: ['_id', 'text', 'time'],
12   }).then(function (result) {
13     renderList(result)
14   })
15 }
```


```
17 var changes = db.changes({since: 'now', live: true})
18    .on('change', updateList);
```




```
21 function newStatus(text) {
22   const date = new Date().toISOString()
23
24   db.put({
25           _id: 'status-' + date,
26           type: 'status',
27           text: text,
28           time: date
29   })
30 }
```

```
const PouchDB = require('PouchDB')
PouchDB.plugin(require('pouchdb-find'));

const db = new PouchDB('micro-status')

db.sync("http://localhost:5984/micro-status")

function updateList() {
  db.find({
    selector: {type: 'status'},
    fields: ['_id', 'text', 'time'],
  }).then(function (result) {
    renderList(result)
  })
}

var changes = db.changes({since: 'now', live: true})
  .on('change', updateList);


function newStatus(text) {
  const date = new Date().toISOString()

  db.put({
        _id: 'status-' + date,
        type: 'status',
        text: text,
        time: date
  })
}
```



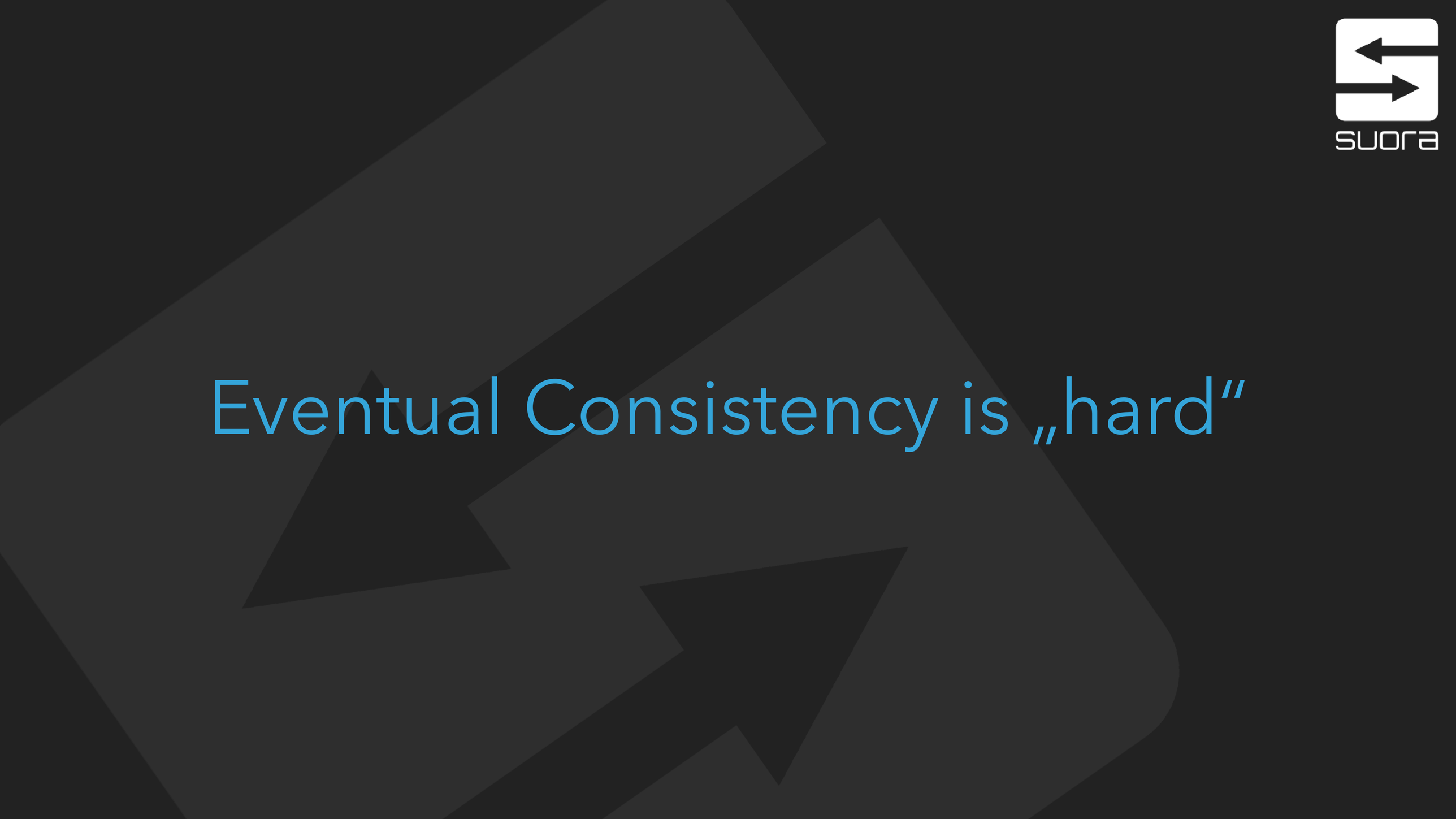
PouchDB-Server

Pouch as a Client for Couch

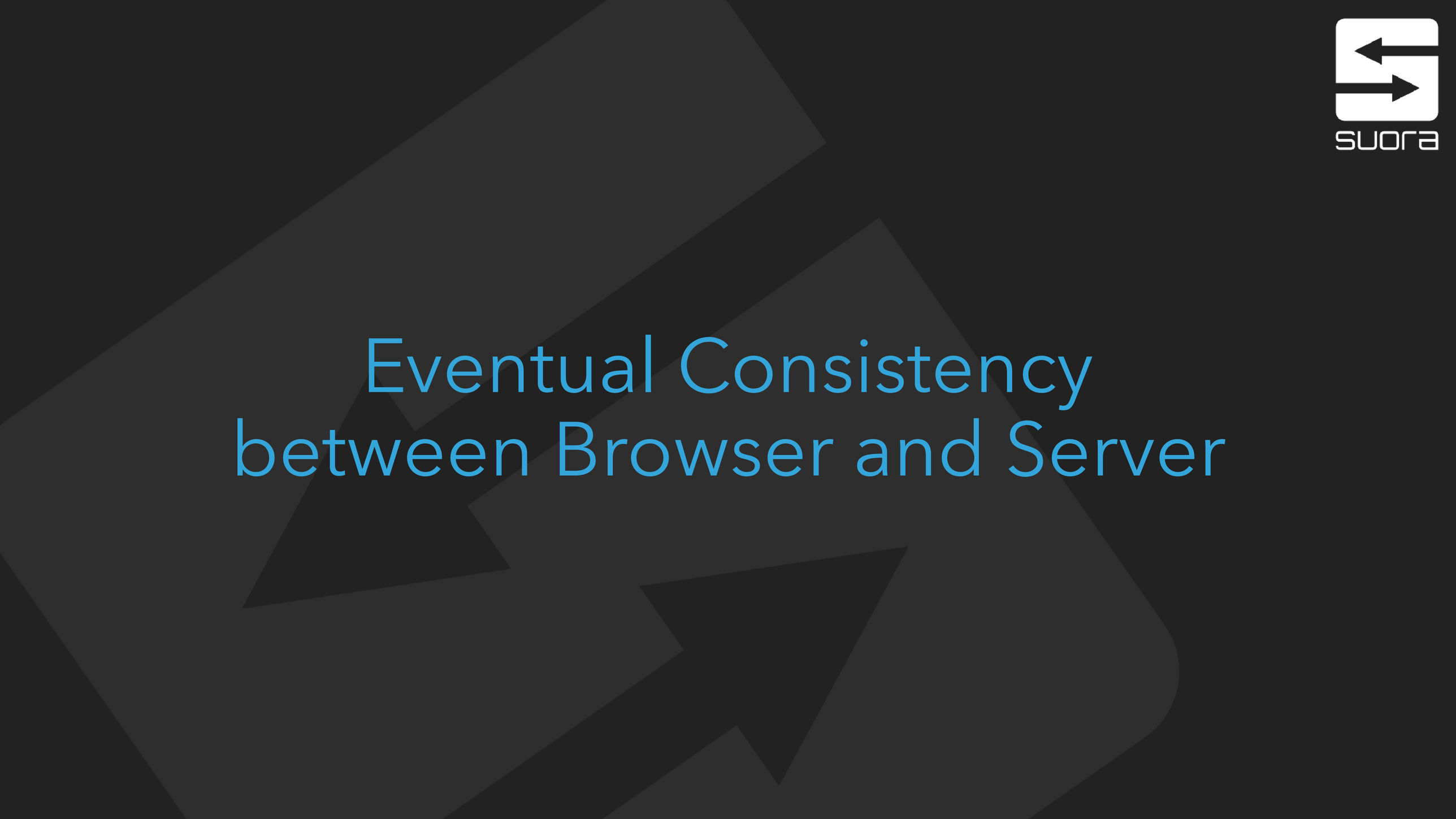

```
1 const PouchDB = require('PouchDB')
2
3 const db = new PouchDB('micro-status')
4 db.get('test').then(function(doc) {
5   console.dir(doc)
6 });
```


```
1 const PouchDB = require('PouchDB')
2
3 const db = new PouchDB('http://localhost:5984/micro-status')
4 db.get('test').then(function(doc) {
5   console.dir(doc)
6 });
```

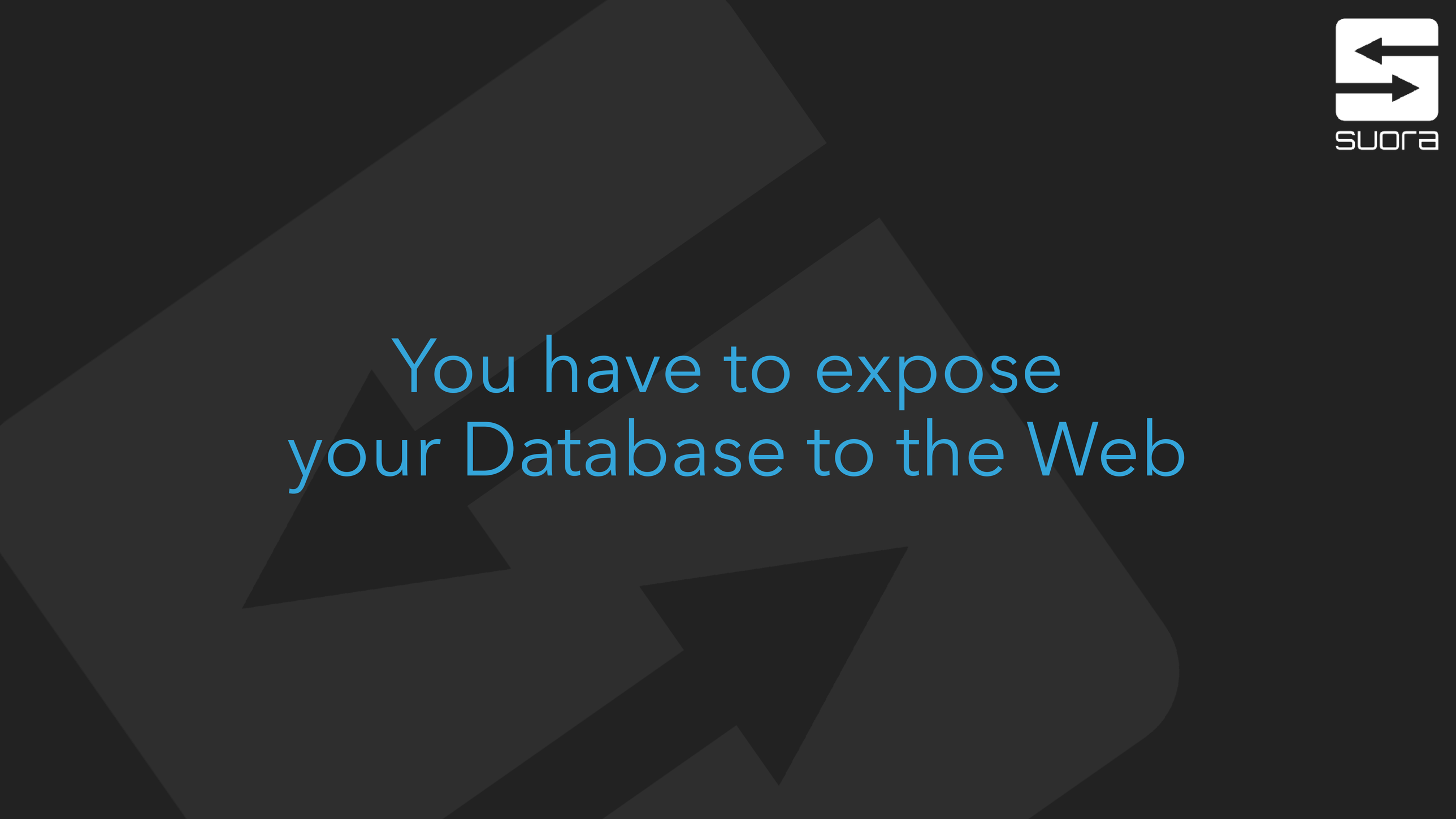
What are the Problems?



Eventual Consistency is „hard“

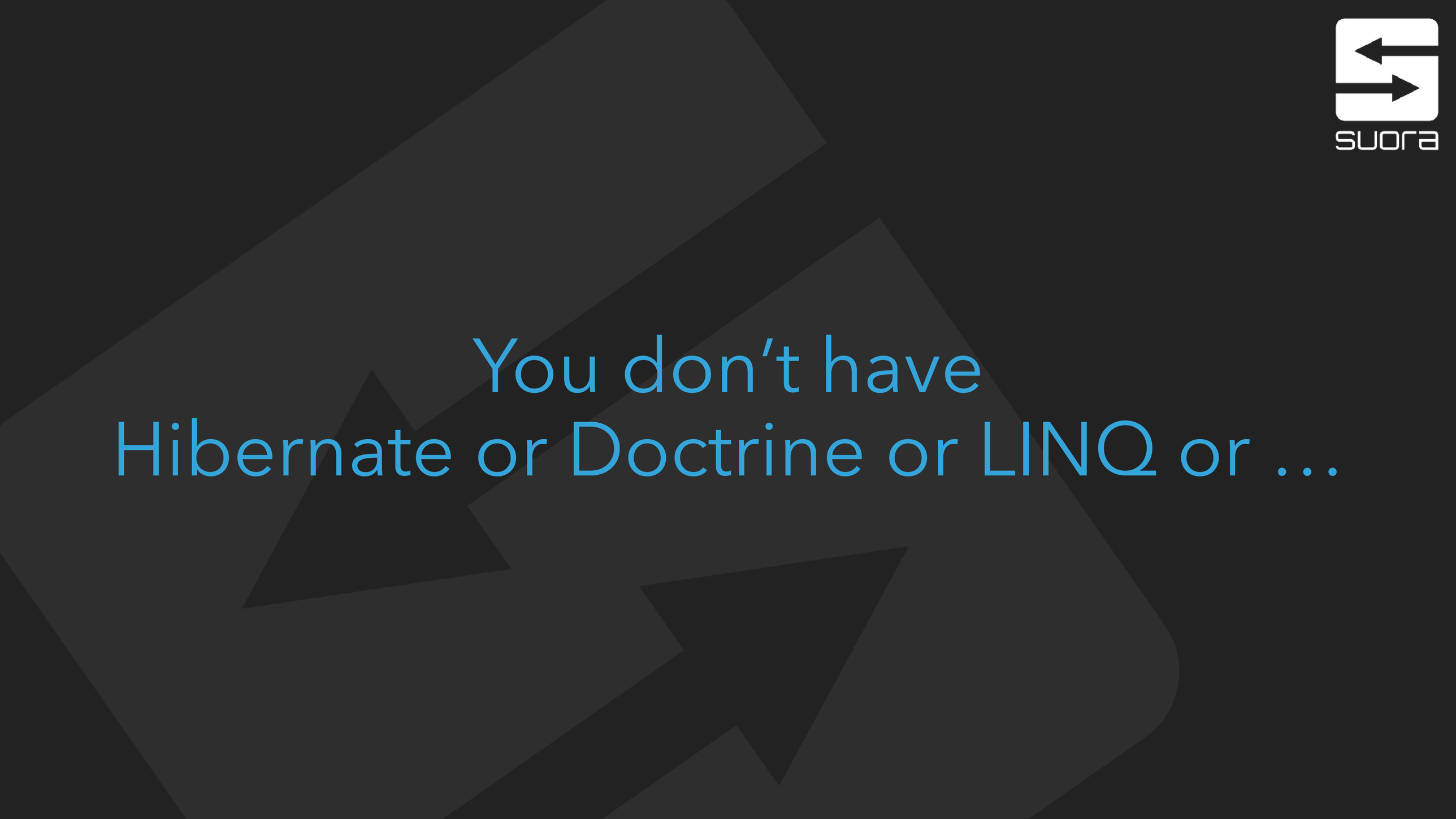


Eventual Consistency between Browser and Server

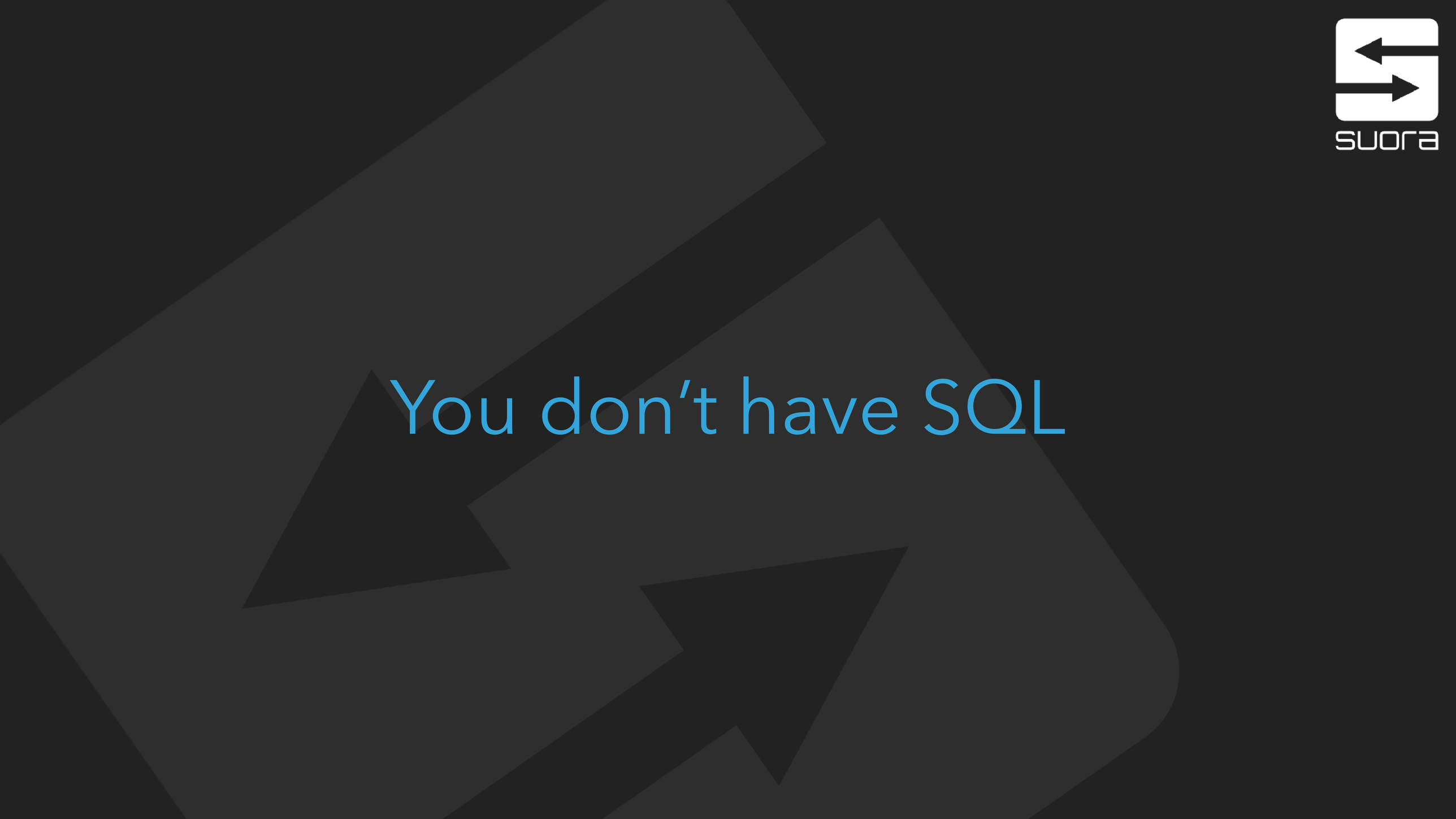


You have to expose
your Database to the Web

You don’t have a „middleware“ between Database and Browser



You don’t have
Hibernate or Doctrine or LINQ or …



You don’t have SQL

Can I use it in Production?

Yes, absolutely!

Thanks!

Hans-Christian Otto
Suora GmbH
c.otto@suora.com

@muhdiekuh
@SuoraGmbH
suora.com