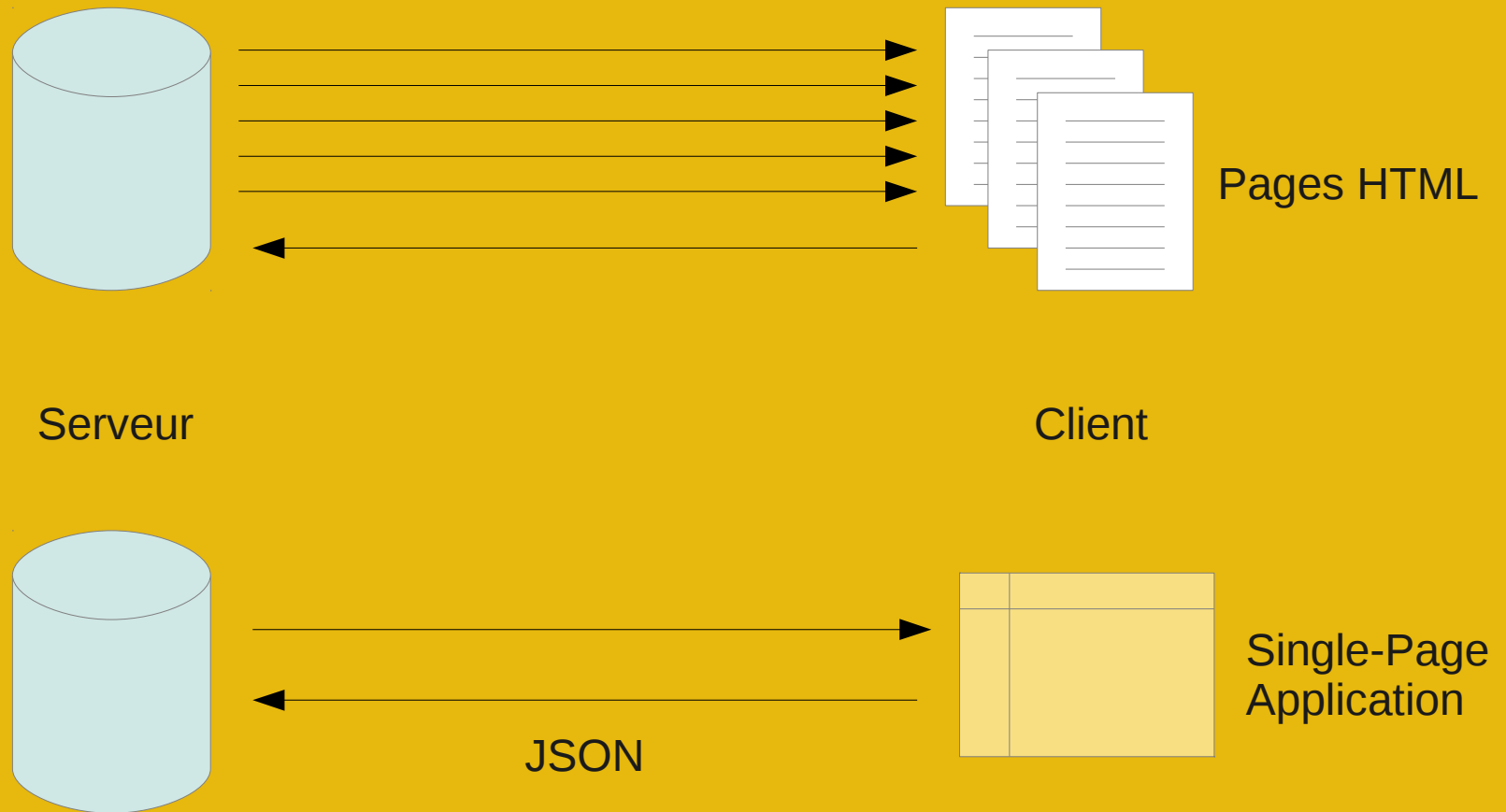


HTML 5

stockage local & synchronisation

Une mutation est en cours



HTML



Form validation

Geolocation

Drag n' Drop API

Web Notifications

Canvas

WebGL

Video

Audio

File API

Web Workers

Web Sockets

HTML



Form validation

Geolocation

Drag n' Drop API

Web Notifications

Stockage local

Canvas

Audio

File API

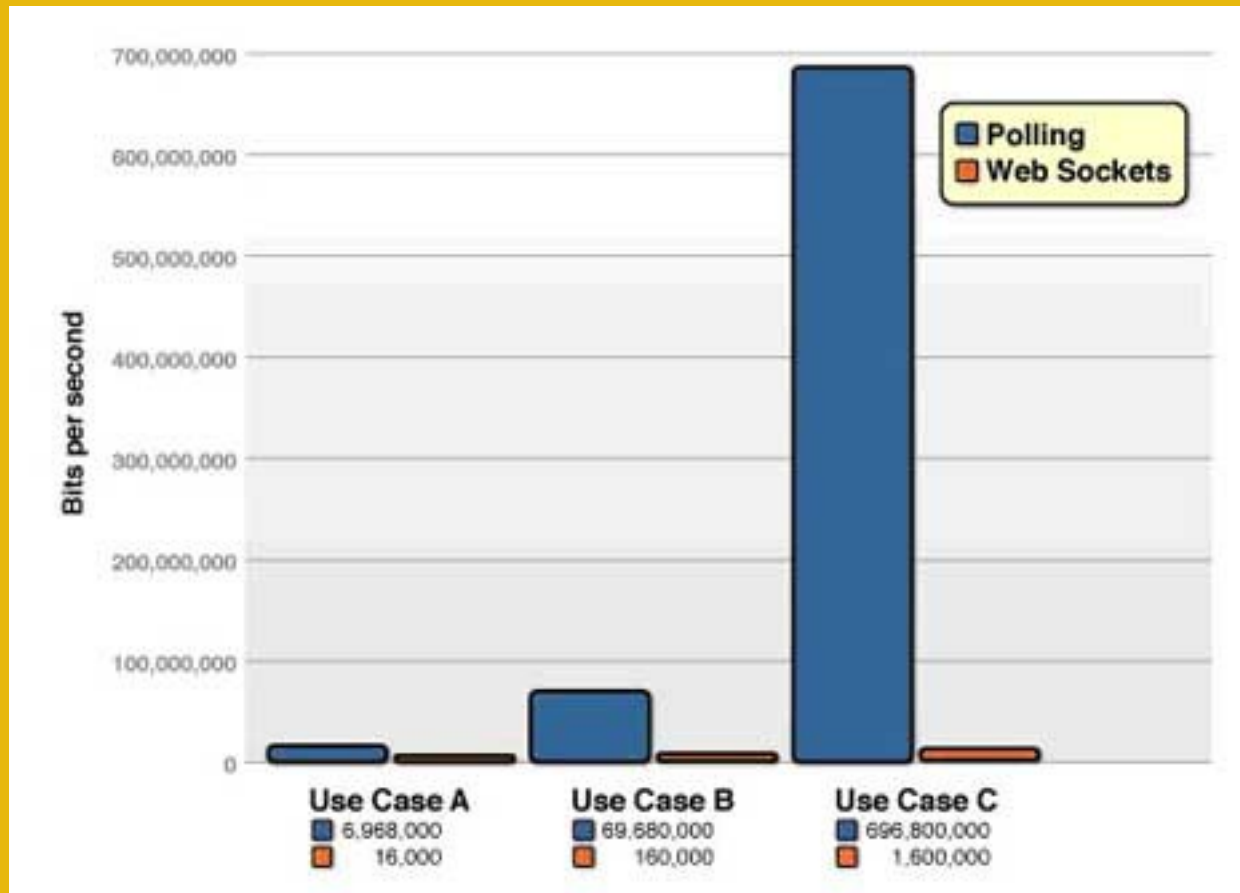
Web Workers

Web Sockets

1er bénéfice : le offline ?

The image shows a screenshot of the BlueMind Agenda interface. The top navigation bar includes 'Messagerie', 'Contacts', and 'Agenda'. The user is identified as 'Raphaël Rougeron' and is currently 'En ligne' (online). The main area displays a calendar for April 2012, with a specific event titled 'test' scheduled for Saturday, April 7th, from 08:30 to 14:00. A dialog box titled 'Mode de connexion' is overlaid on the calendar, asking the user: 'Vous êtes actuellement en ligne. Voulez-vous passer hors ligne?' (You are currently online. Do you want to go offline?). A red warning box below the question states: 'Vous allez passer hors ligne. Toutes vos modifications seront synchronisées lors de votre prochaine connexion.' (You will go offline. All your modifications will be synchronized during your next connection). The dialog has 'Oui' (Yes) and 'Non' (No) buttons. Below the dialog, a zoomed-in view of the calendar interface is shown, where the user's status has changed to 'Hors ligne' (Offline), indicated by a grey circle icon and a dashed line. The event 'test' is still visible on the calendar grid.

1er bénéfice : économiser le serveur !



Source : HTML5 Web Sockets: A Quantum Leap in Scalability for the Web
<http://soa.sys-con.com/node/1315473>

4 étapes

Etape 1

**Rendre les ressources
disponibles hors connexion**

Application Cache

```
<html manifest="cache.manifest">
```



CACHE MANIFEST

CACHE:

index.html

css/style.css

img/logo.png

js/main.js

text/cache-manifest



**Avoir un manifeste
change complètement
les règles d'accès aux ressources**

CACHE MANIFEST

v.1.2-build1234

CACHE:

index.html

css/style.css

img/logo.png

js/main.js

CACHE MANIFEST

v.1.2-build1234

CACHE:

index.html

css/style.css

img/logo.png

js/main.js

NETWORK

/sync

/api/*

<http://api.twitter.com>

CACHE MANIFEST

v.1.2-build1234

CACHE:

index.html

css/style.css

img/logo.png

js/main.js

NETWORK

/sync

/api/*

http://api.twitter.com

FALLBACK

images/large/ images/offline.jpg

*.html /offline.html

window.applicationCache

status

UNCACHED

IDLE

CHECKING






DOWNLOADING




UPDATEREADY

OBSOLETE

```
var appCache = window.applicationCache;

appCache.addEventListener('updateready', function(e) {
    if (appCache.status == appCache.UPDATEREADY) {
        appCache.swapCache();
        if (confirm('Une nouvelle version est dispo')) {
            window.location.reload();
        }
    }
}, false);
```

				
$\geq 10 ?$	≥ 3.5	≥ 4.0	≥ 4.0	≥ 10.6

		
≥ 3.2	≥ 2.1	≥ 11.0

Etape 2

Découpler l'application du réseau



BACKBONE.JS

```
var Todo = Backbone.Model.extend({

  toggle: function() {
    this.save({done: !this.get("done")});
  },

  clear: function() {
    this.destroy();
  }

});

var TodoList = Backbone.Collection.extend({

  model: Todo,

  done: function() {
    return this.filter(function(todo) { return todo.get('done'); });
  },

  remaining: function() {
    return this.without.apply(this, this.done());
  }

});
```

Backbone.Sync

- `create()` → **POST** `/collection`
- `read()` → **GET** `/collection[/id]`
- `update()` → **PUT** `/collection/id`
- `delete()` → **DELETE** `/collection/id`

Etape 3

Stocker localement les données

3 options

FileSystem API



seulement

```
window.requestFileSystem = window.requestFileSystem
    || window.webkitRequestFileSystem;

window.requestFileSystem(window.TEMPORARY,
    5*1024*1024,
    onInitFs, errorHandler);

window.webkitStorageInfo.requestQuota(window.PERSISTENT,
    5*1024*1024,
    initFS, errorHandler);

function initFS( grantedBytes ) {
    window.requestFileSystem(PERSISTENT, grantedBytes,
        onInitFs, errorHandler);
}
```

FileError

code

QUOTA_EXCEEDED_ERR

NOT_FOUND_ERR

SECURITY_ERR

INVALID_MODIFICATION_ERR

INVALID_STATE_ERR


```
function onInitFs(fs) {
  fs.root.getFile('data.txt', {create: true}, function(entry) {
    entry.file(function(file) {

      var reader = new FileReader();

      reader.onloadend = function(e) {
        repository.init(JSON.parse(e.target.result));
      };

      reader.readAsText(file);

    }, errorHandler);
  }, errorHandler);
}
```

FileEntry

name

fullpath

isFile

isDirectory

...

file()

createWriter()

moveTo()

copyTo()

remove()

...

```
function persistData(fs, data) {
  fs.root.getFile('log.txt', {create: true}, function(entry) {
    entry.createWriter(function(writer) {

      writer.onwriteend = function(e) {
        console.log('Write completed.');
```

```
      };

      writer.onerror = function(e) {
        console.log('Write failed: ' + e.toString());
      };

      var bb = new BlobBuilder();
      bb.append(JSON.stringify(data));
      writer.write(bb.getBlob('text/plain'));

    }, errorHandler);
  }, errorHandler);
}
```

Web Storage

localStorage et sessionStorage

Un simple dépôt clé-valeur

```
localStorage.setItem("foo", "bar");
```

```
localStorage.setItem("tweets", JSON.stringify(tweets));
```

```
var tweets = JSON.parse(localStorage.getItem("tweets"));
```



Quota de 5 Mo
Pas de transactions
Pas d'indexation

```
localStorage["tweets:1234"] = "Lorem ipsum...";  
localStorage["tweets:1235"] = "Nam mauris lorem...";  
  
localStorage["tags"] = JSON.stringify(["Java", "Python",  
                                       "Ruby", "PHP"]);  
localStorage["tags:Java"] = JSON.stringify([1234, 1235]);
```



- Très bien supporté
- API simple
- Mis en avant par de nb compagnies/projets



- Performances
 - API synchrone
 - Sériailisation
 - Requêtes
- Complexité
 - "indexation" manuelle
 - Maintien intégrité





≥ 8



≥ 3.5



≥ 4.0



≥ 4.0



≥ 10.5



≥ 3.2



≥ 2.1



≥ 11.0



localStorage adapter

```
Backbone.sync = function(method, model, options) {

  var resp;
  var store = model.localStorage || model.collection.localStorage;

  switch (method) {
    case "read":      resp = model.id ? store.find(model) :
                      store.findAll();
                      break;
    case "create":   resp = store.create(model);
                      break;
    case "update":   resp = store.update(model);
                      break;
    case "delete":   resp = store.destroy(model);
                      break;
  }

  if (resp) {
    options.success(resp);
  } else {
    options.error("Record not found");
  }
};
```

Petit intermède

Vous connaissez Redis ?

BankersBox

```
var bb = new BankersBox(1);

bb.set("foo", "bar");
bb.get("foo"); // returns "bar"

bb.set("count", 10);
bb.incr("count"); // sets "count" to 11, returns 11

bb.incr("newcount"); // sets "newcount" to 1, returns 1

bb.lpush("mylist", "hello");
bb.lrange("mylist", 0, -1); // returns ["hello"]

bb.rpush("mylist", "world");
bb.lrange("mylist", 0, -1); // returns ["hello", "world"]

bb.sadd("myset", "apple");
bb.sadd("myset", "orange");
```

IndexedDB

Préfixes, préfixes...

```
window.indexedDB = window.indexedDB || window.mozIndexedDB ||  
    window.webkitIndexedDB;  
window.IDBKeyRange = window.IDBKeyRange ||  
    window.webkitIDBKeyRange;  
window.IDBTransaction = window.IDBTransaction ||  
    window.webkitIDBTransaction;
```

API asynchrone

```
function IndexedDBAdapter() {
  this.db = null;

  var request = indexedDB.open("contactApp");

  request.onsuccess = function(e) {
    this.db = e.target.result;
  }.bind(this);

  request.onfailure = function(e) {
    console.log("Could not connect to the database");
  }
};
```

Création d'un dépôt

```
IndexedDBAdapter.prototype.create = function(storeName) {
  var v = "1.0";
  var request = this.db.setVersion(v);

  request.onsuccess = function(e) {
    var store = this.db.createObjectStore(storeName, {
      "keyPath": "id",
      "autoIncrement": true
    });
  };

  request.onblocked = function(e) {
    console.log("The database is open in another tab.");
  };
};
```

Persistence d'un objet

```
IndexedDBAdapter.prototype.save(storeName, object, callback)
{
  var trans = db.transaction([storeName],
                             IDBTransaction.READ_WRITE, 0);
  var store = trans.objectStore(storeName);
  var request = store.put(object);

  request.onsuccess = function(e) {
    callback(object);
  };
};
```


Requête simple

```
IndexedDBAdapter.prototype.all(storeName, callback) {
  var trans = db.transaction([storeName],
                             IDBTransaction.READ_WRITE, 0);
  var store = trans.objectStore(storeName);

  var keyRange = IDBKeyRange.lowerBound(0);
  var request = store.openCursor(keyRange);

  request.onsuccess = function(e) {
    var cursor = e.target.result;
    if (cursor) {
      callback(cursor.value);
      cursor.continue();
    }
  };
};
```

Key Ranges

```
IDBKeyRange.upperBound(x);           // <= x
IDBKeyRange.upperBound(x, true);     // < x
IDBKeyRange.lowerBound(y);          // >= y
IDBKeyRange.lowerBound(y, true);     // > y
IDBKeyRange.bound(x, y);             // >= x && <= y
IDBKeyRange.bound(x, y, true, false); // > x && <= y
```

Index






```
var contacts = [  
  { firstname: "John", lastname: "Doe", email: "jdoe@zz.com" },  
  { firstname: "Jane", lastname: "Doe", email: "jane@zz.com" },  
  { firstname: "Johnny", lastname: "Carson", email: "jca@zz.com" }  
];  
  
objectStore.createIndex("firstname", "firstname", { unique: false });  
objectStore.createIndex("lastname", "lastname", { unique: false });  
objectStore.createIndex("email", "email", { unique: true });
```




Index

```
var contacts = [  
  { firstname: "John", lastname: "Doe", email: "jdoe@zz.com" },  
  { firstname: "Jane", lastname: "Doe", email: "jane@zz.com" },  
  { firstname: "Johnny", lastname: "Carson", email: "jca@zz.com" }  
];
```






```
var index = objectStore.index("lastname");  
index.openCursor(IDBKeyRange.only("Doe")).onsuccess = function(e) {  
  var cursor = e.target.result;  
  if (cursor) {  
    callback(cursor.value);  
    cursor.continue();  
  }  
};
```




```
var index = objectStore.index("firstname");  
index.openCursor(IDBKeyRange.lowerbound("John")).onsuccess =  
function(e) {  
  ...  
};
```

				
$\geq 10 ?$	≥ 8.0	≥ 16.0	?	?

		
?	?	?

**Et pourquoi pas
Web SQL Database ?**

				
X	X	≥ 4.0	≥ 3.1	≥ 10.5

		
≥ 3.2	≥ 2.1	≥ 11.0

Le polyfill ultime ?

Lawnchair

- localStorage
- indexedDB
- webkit-sqlite
- gears-sqlite
- ie-userdata
- backberry-persistent-store
- window-name



Etape 4

**Développer une stratégie de
synchronisation**

L'exemple d'ActiveSync

```
POST /Microsoft-Server-ActiveSync?User=jdoe@zz.com&...  
&Cmd=<Commande>
```

```
Cmd=FolderSync&SyncKey=123456789
```

```
{  
  folders: [  
    contacts: {  
      id: 1234,  
      created: 0,  
      updated: 2,  
      deleted: 1  
    },  
    todos: {  
      id: 5678,  
      created: 5,  
      updated: 0,  
      deleted: 2  
    }  
  ]  
}
```

Cmd=**Sync**&SyncKey=123456789&FolderId=5678

```
{
  syncKey: 987654321,
  created: [
    { id: 4321, label: "Acheter le pain" },
    ...
  ],
  updated: [
    ...
  ],
  deleted: [
    5678, 7890
  ]
}
```

sync_state

•user_id	integer
•client_id	integer
•sync_key	varchar
•last_sync	timestamp

contacts

•id	integer
◦firstname	varchar
◦lastname	varchar
◦created_at	timestamp
◦updated_at	timestamp
◦deleted_at	timestamp
•deleted	boolean

Autres pistes

Mozilla Services

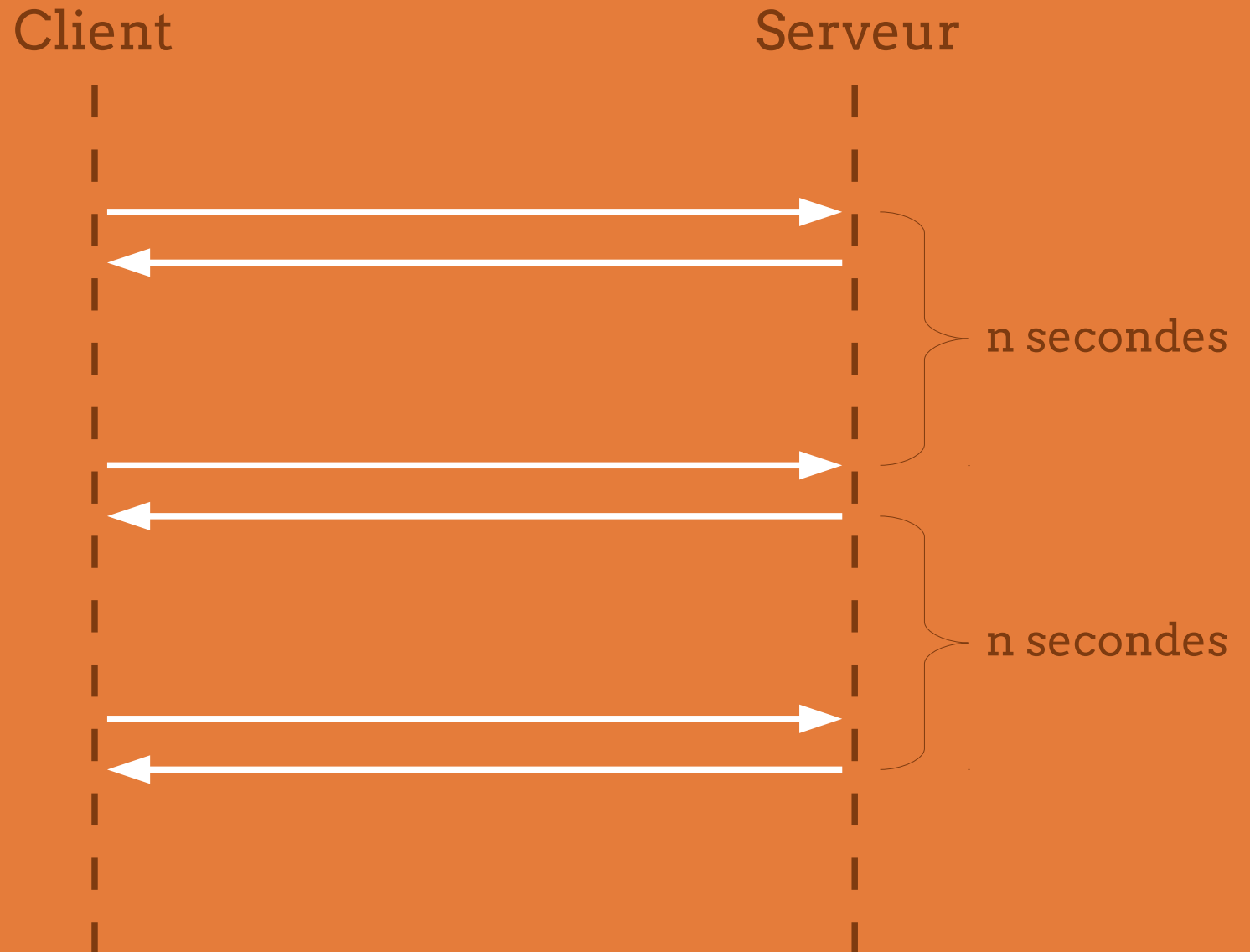
SyncML

Operational Transformation

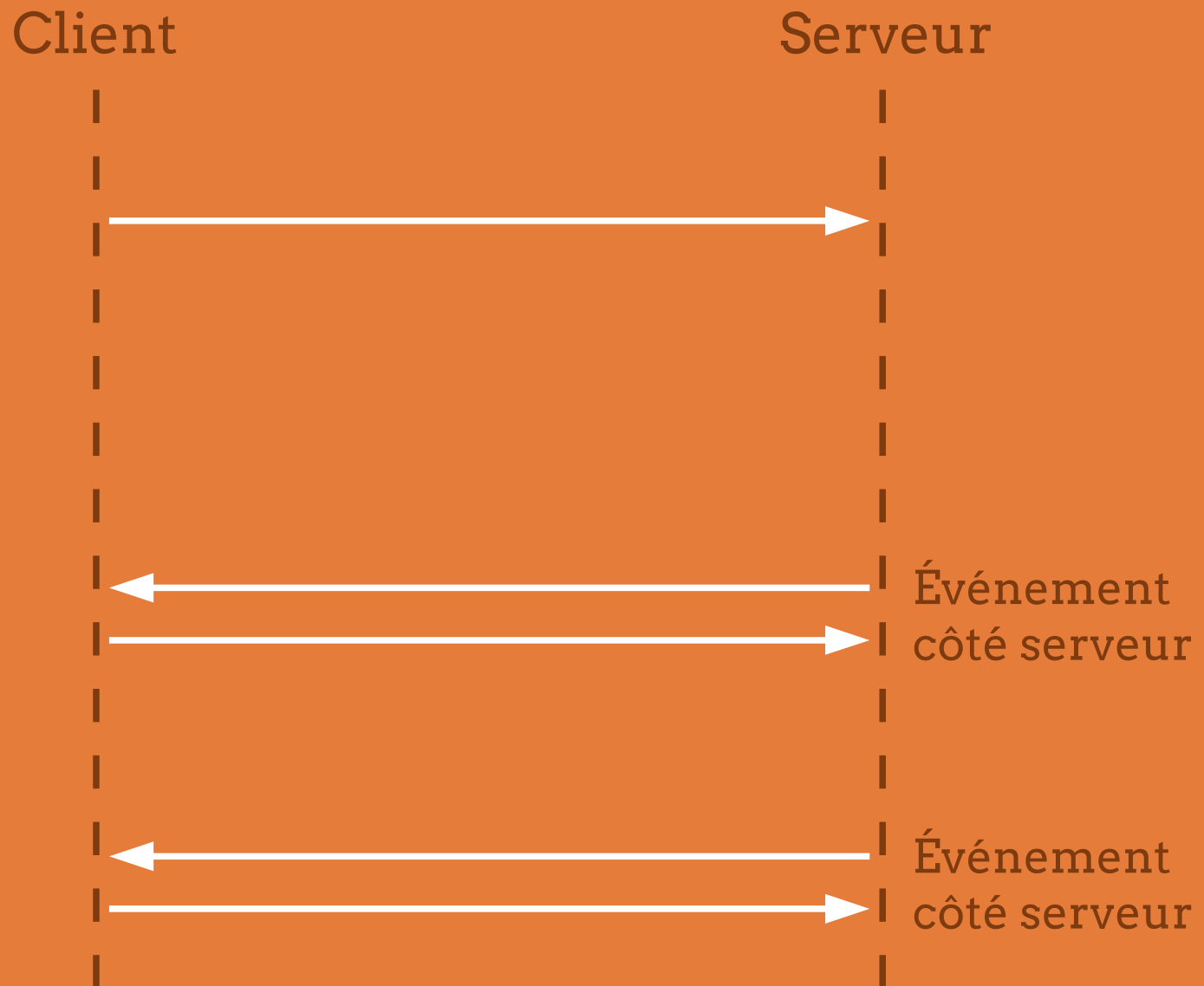
→ ShareJS

Etape subsidiaire
Le "push"

Polling



Long polling (COMET)



WebSockets

Handshake

```
GET /resource name/ HTTP/1.1
Upgrade: WebSocket
Connection: Upgrade
Host: /server/
Origin: /origin/
WebSocket-Protocol: /protocol/
```

```
HTTP/1.1 101 Web Socket Protocol Handshake
Upgrade: WebSocket
Connection: Upgrade
WebSocket-Origin: /origin/
WebSocket-Location: /url/
WebSocket-Protocol: /subprotocol/
```






Serveur (avec node.js)




```
var io = require('socket.io').listen(80);

io.sockets.on('connection', function (socket) {
  socket.emit('news', { hello: 'world' });
  socket.on('my other event', function (data) {
    console.log(data);
  });
});
```

Client




```
var socket = io.connect('http://localhost');
socket.on('news', function (data) {
  console.log(data);
  socket.emit('my other event', { my: 'data' });
});
```

				
$\geq 10 ?$	≥ 6.0	≥ 14.0	≥ 5.0	≥ 11.0

		
≥ 4.2	?	≥ 11.0

Avec Socket.IO

				
≥ 5.5	≥ 3.0	≥ 4.0	≥ 3.0	≥ 10.61

		
≥ 3.2	≥ 2.1	≥ 11.0

Server-Sent Events

HTTP traditionnel

```
GET /myAppStream
```

```
Content-Type: text/event-stream
```



```
if (window.EventSource) {
    var source = new EventSource('myAppStream');
} else {
    // fallback to something...
}

source.addEventListener('message', function(e) {
    console.log(e.data);
}, false);

source.addEventListener('open', function(e) {

}, false);

source.addEventListener('error', function(e) {
    if (e.readyState == EventSource.CLOSED) {

    }
}, false);
```

Format des messages

```
data: Hello world\n\n
```

Multi-ligne

```
data: Hello\n
```

```
data: world\n\n
```

JSON

```
data: {\n
```

```
data: "msg": "Hello world",\n
```

```
data: "id": 12345\n
```

```
data: }\n\n
```

Format des messages

Utilisation des IDs

```
id: 12345\n
```

```
data: {"user": "goldo", "msg": "Hello !"}\n\n
```

Utilisation des events

```
data: {"msg": "Hello !"}\n\n
```

```
event: login\n
```

```
data: {"user": "goldo"}\n\n
```

```
event: update\n
```

```
data: {"user": "goldo", "status": "away"}\n\n
```

Avantages

Reconnexion automatique

Ids

Events

Questions / réponses