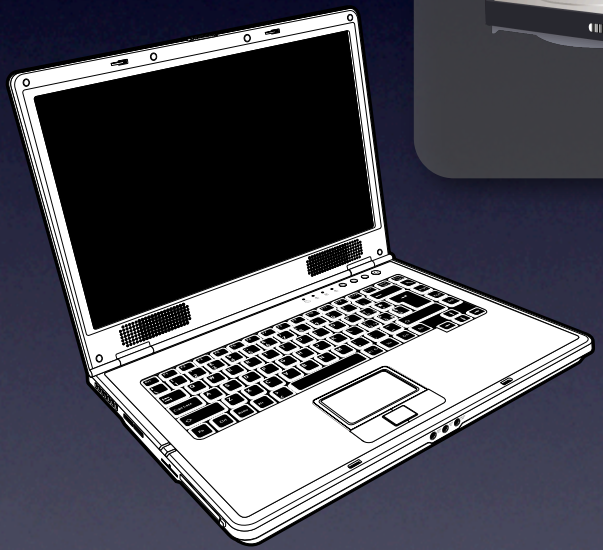
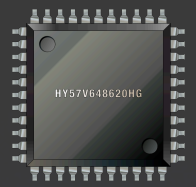
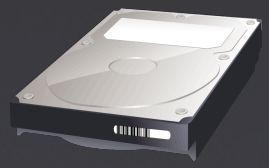
Introduction to Gears

Brad Neuberg
Google





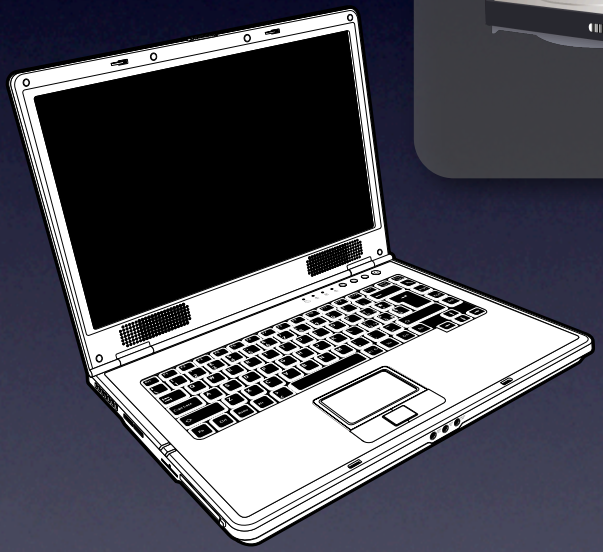


JavaScript

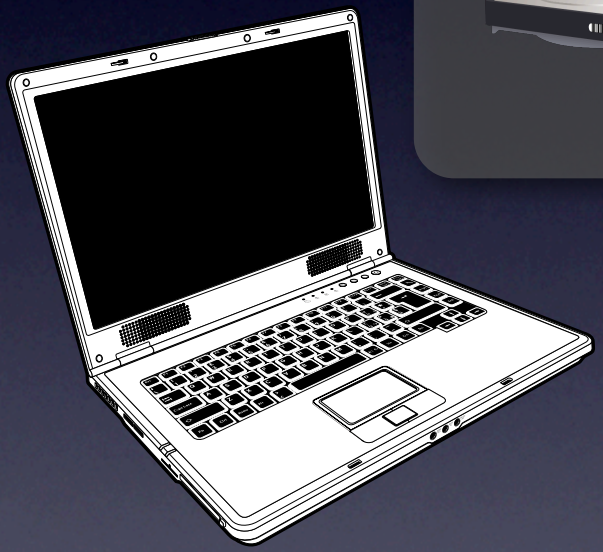
CSS

HTML

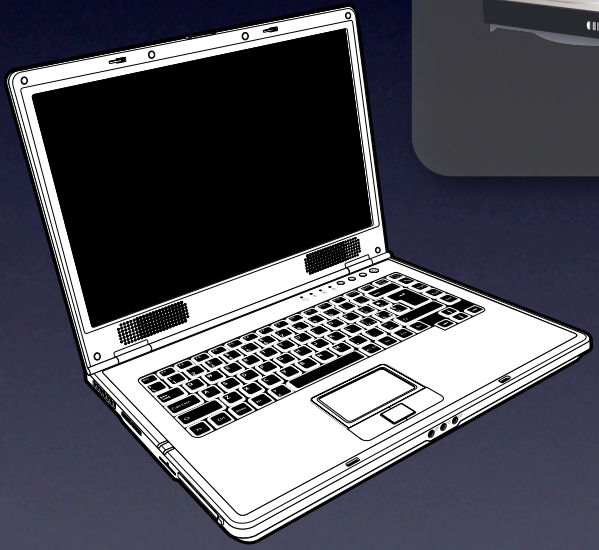
What is Gears?



What is Gears?

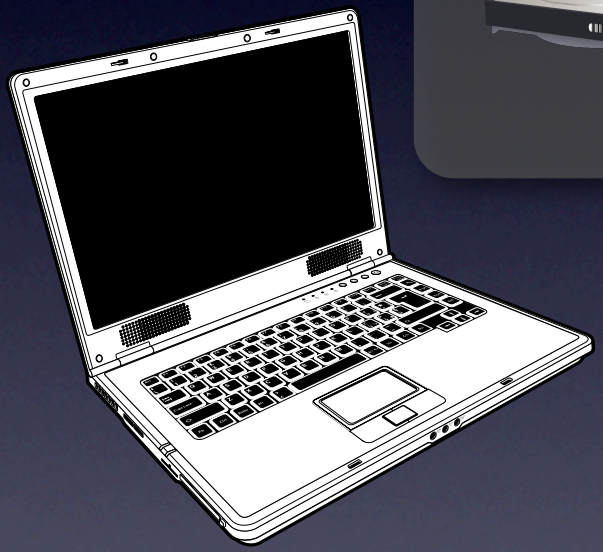


What is Gears?



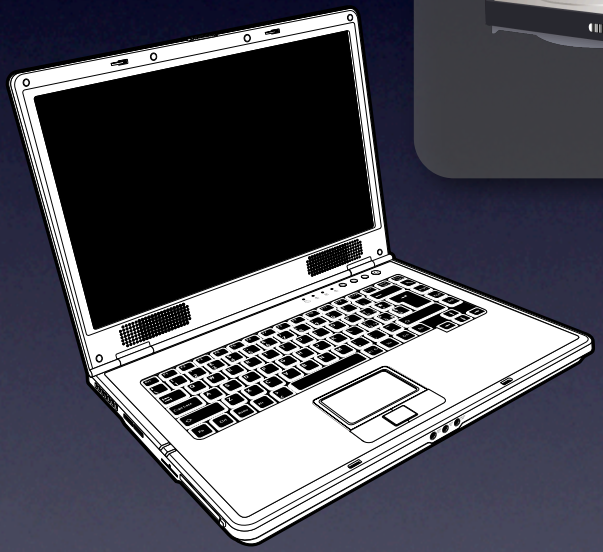
What is Gears?

HTML 5



What is Gears?

HTML 5



Open Source!

What is Gears?

Ajax++

Database

JavaScript

CSS

HTML

Ajax++

Database

Client-Side Search

JavaScript

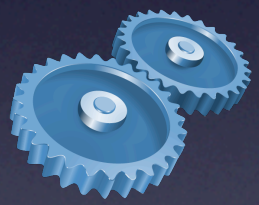
CSS

HTML

Ajax++

Database

Client-Side Search



Worker Pool

JavaScript

CSS

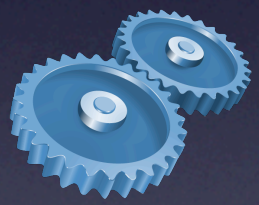
HTML

Ajax++

Database

Desktop API

Client-Side Search



Worker Pool

JavaScript

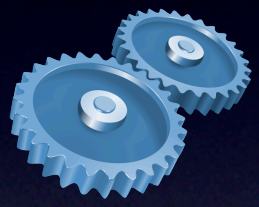
CSS

HTML

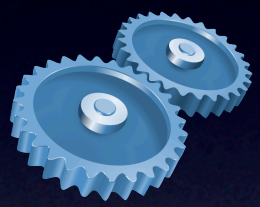
Ajax++

Database

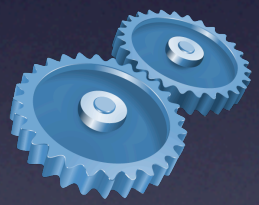
Desktop API



Client-Side Search



Local Server



Worker Pool

JavaScript

CSS

HTML

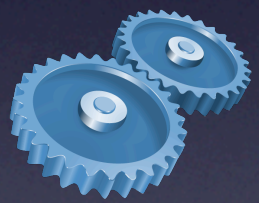
Ajax++

Database

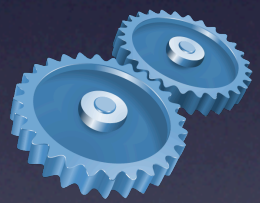
Desktop API

Client-Side Search

Local Server



Worker Pool



Blobs

JavaScript

CSS

HTML

Ajax++

Ajax++

File System API

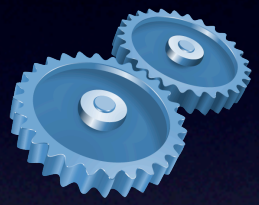
JavaScript

CSS

HTML

Ajax++

File System API



Geolocation

JavaScript

CSS

HTML

Ajax++


Database

Database

- Local SQL storage
- SQLite: Open source, mature, small (343K), fast
- Full-featured relational database
- Gigabytes of storage capacity
- Strict same-origin security model

Database Code

Database Code

```
var db = google.gears.factory.create('beta.database');
```

Database Code

```
var db = google.gears.factory.create('beta.database');

db.open('database-test');
```

Database Code

```
var db = google.gears.factory.create('beta.database');

db.open('database-test');

db.execute('CREATE TABLE IF NOT EXISTS Test' +
```

Database Code

```
var db = google.gears.factory.create('beta.database');

db.open('database-test');

db.execute('CREATE TABLE IF NOT EXISTS Test' +
           ' (Phrase TEXT, Timestamp INT)');
```

Database Code

```
var db = google.gears.factory.create('beta.database');

db.open('database-test');

db.execute('CREATE TABLE IF NOT EXISTS Test' +
           ' (Phrase TEXT, Timestamp INT)');

db.execute('INSERT INTO Test VALUES (?, ?)', ['Monkey!',
```

Database Code

```
var db = google.gears.factory.create('beta.database');

db.open('database-test');

db.execute('CREATE TABLE IF NOT EXISTS Test' +
           ' (Phrase TEXT, Timestamp INT)');

db.execute('INSERT INTO Test VALUES (?, ?)', ['Monkey!',
            new Date().getTime()]);
```

Database Code

Database Code

```
var rs;
```

Database Code

```
var rs;

try {
```

Database Code

```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');
```

Database Code

```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');

  while (rs.isValidRow()) {
```

Database Code

```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');

  while (rs.isValidRow()) {
    console.log(rs.fieldByName(‘Phrase’) + '@'
```

Database Code

```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');

  while (rs.isValidRow()) {
    console.log(rs.fieldByName(‘Phrase’) + '@'
                + rs.fieldByName(‘Timestamp’));
```

Database Code

```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');

  while (rs.isValidRow()) {
    console.log(rs.fieldByName(‘Phrase’) + '@'
                + rs.fieldByName(‘Timestamp’));
    rs.next();
```

Database Code

```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');

  while (rs.isValidRow()) {
    console.log(rs.fieldByName(‘Phrase’) + '@'
                + rs.fieldByName(‘Timestamp’));
    rs.next();
  }
```

Database Code

```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');

  while (rs.isValidRow()) {
    console.log(rs.fieldByName(‘Phrase’) + '@'
                + rs.fieldByName(‘Timestamp’));
    rs.next();
  }
} finally {
```

Database Code

```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');

  while (rs.isValidRow()) {
    console.log(rs.fieldByName(‘Phrase’) + '@'
                + rs.fieldByName(‘Timestamp’));
    rs.next();
  }
} finally {
  rs.close();
```

Database Code

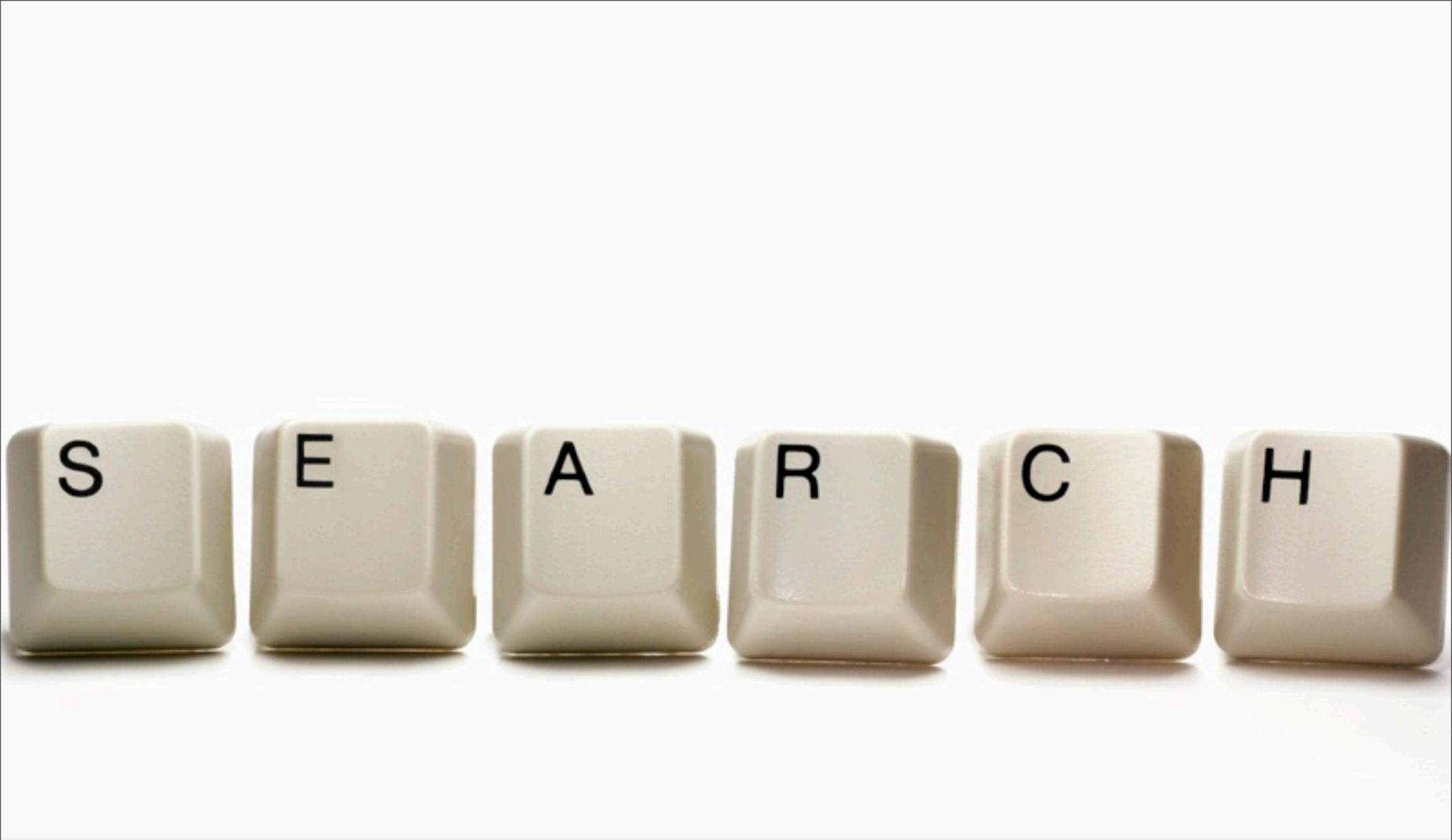
```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');

  while (rs.isValidRow()) {
    console.log(rs.fieldByName(‘Phrase’) + '@'
                + rs.fieldByName(‘Timestamp’));
    rs.next();
  }
} finally {
  rs.close();
  db.close();
```

Database Code

```
var rs;

try {
  rs = db.execute('SELECT * FROM Test ORDER BY Timestamp DESC');

  while (rs.isValidRow()) {
    console.log(rs.fieldByName(‘Phrase’) + '@'
                + rs.fieldByName(‘Timestamp’));
    rs.next();
  }
} finally {
  rs.close();
  db.close();
}
```

Demo:

MySpace Messaging + Gears



Full-Text Search

Full Text Search

- Gears added FTS2 to SQLite
- Create the database

```
db.execute('CREATE VIRTUAL TABLE recipe USING
fts2(dish, ingredients)');
```

- Search the database

```
db.execute('SELECT dish FROM recipe WHERE recipe
MATCH ?', ['tomatoes']);
```

Fun queries: *dish:stew tomatoes*

Find rows with 'stew' in the dish field, and 'tomatoes' in any field.

Demo:

PubTools Search

Local Server

Local Server

- Run web applications offline
- Capture UI: HTML, JavaScript, CSS
- Serves locally even when connected

Local Server

- ResourceStore
  - Capture individual URLs
- ManagedResourceStore
  - Capture manifest of resources

Local Server Code

```
// site-manifest.js
{
  "betaManifestVersion": 1,
  "version": "1.1",
  "entries": [
    { "url": "site.html" },
    { "url": "gears_init.js" }
  ]
}
```

Local Server Code

Local Server Code

```
var localServer = google.gears.factory.create
```

Local Server Code

```
var localServer = google.gears.factory.create
                  ('beta.localserver');
```

Local Server Code

```
var localServer = google.gears.factory.create
                  ('beta.localserver');

var store = localServer.openManagedStore(‘test-store’)
```

Local Server Code

```
var localServer = google.gears.factory.create
                  ('beta.localserver');

var store = localServer.openManagedStore(‘test-store’)
          | localServer.createManagedStore('test-store');
```

Local Server Code

```
var localServer = google.gears.factory.create
                  ('beta.localserver');

var store = localServer.openManagedStore(‘test-store’)
          | localServer.createManagedStore('test-store');

store.manifestUrl = 'site-manifest.js';
```

Local Server Code

```
var localServer = google.gears.factory.create
                  ('beta.localserver');

var store = localServer.openManagedStore(‘test-store’)
          | localServer.createManagedStore('test-store');

store.manifestUrl = 'site-manifest.js';

store.checkForUpdate();
```

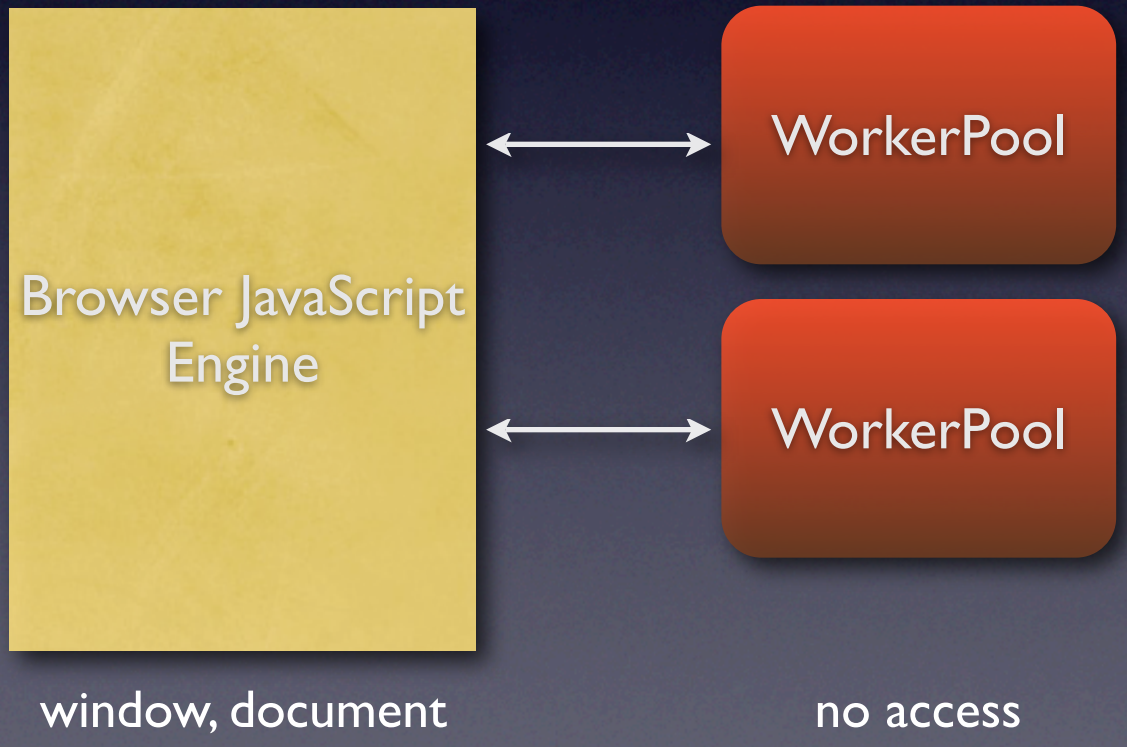
Demo:

WordPress

Worker Pool

Worker Pool

JavaScript needs threads after all? Brendan!

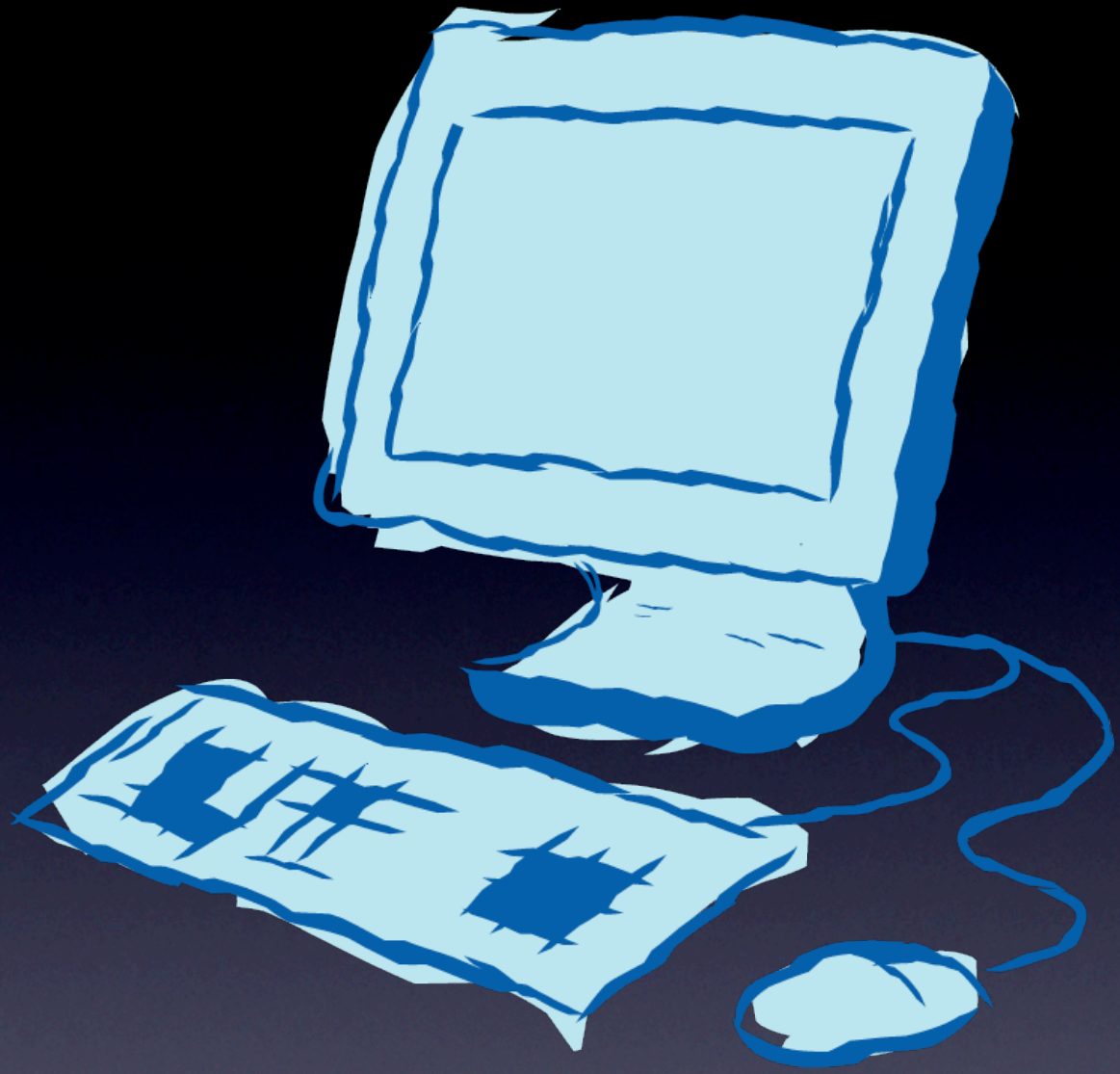


Worker Pool

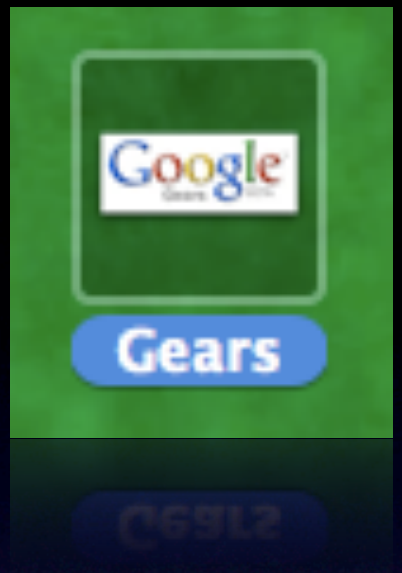
- Demo

Worker Pool Code

```
var pool = google.gears.factory.create('beta.workerpool');

pool.onmessage = function(message) {
  console.log('next prime is: ' + message);
}

function nextPrime(n) {
  // TODO: Prime-finding algorithm goes here.
  google.gears.workerPool.sendMessage(result);
}

var runMe = String(nextPrime) + '; nextPrime()';

var worker = pool.createWorker(runMe);
```



Desktop API



Desktop

Shortcuts

```
var desktop = google.gears.factory.create('beta.desktop');
desktop.createShortcut("Test Application",
                       "http://www.test.com/index.html",
                       {"16x16": "http://www.test.com/icon16x16.png",
                        "32x32": "http://www.test.com/icon32x32.png",
                        "48x48": "http://www.test.com/icon48x48.png",
                        "128x128": "http://www.test.com/icon128x128.png"});
```

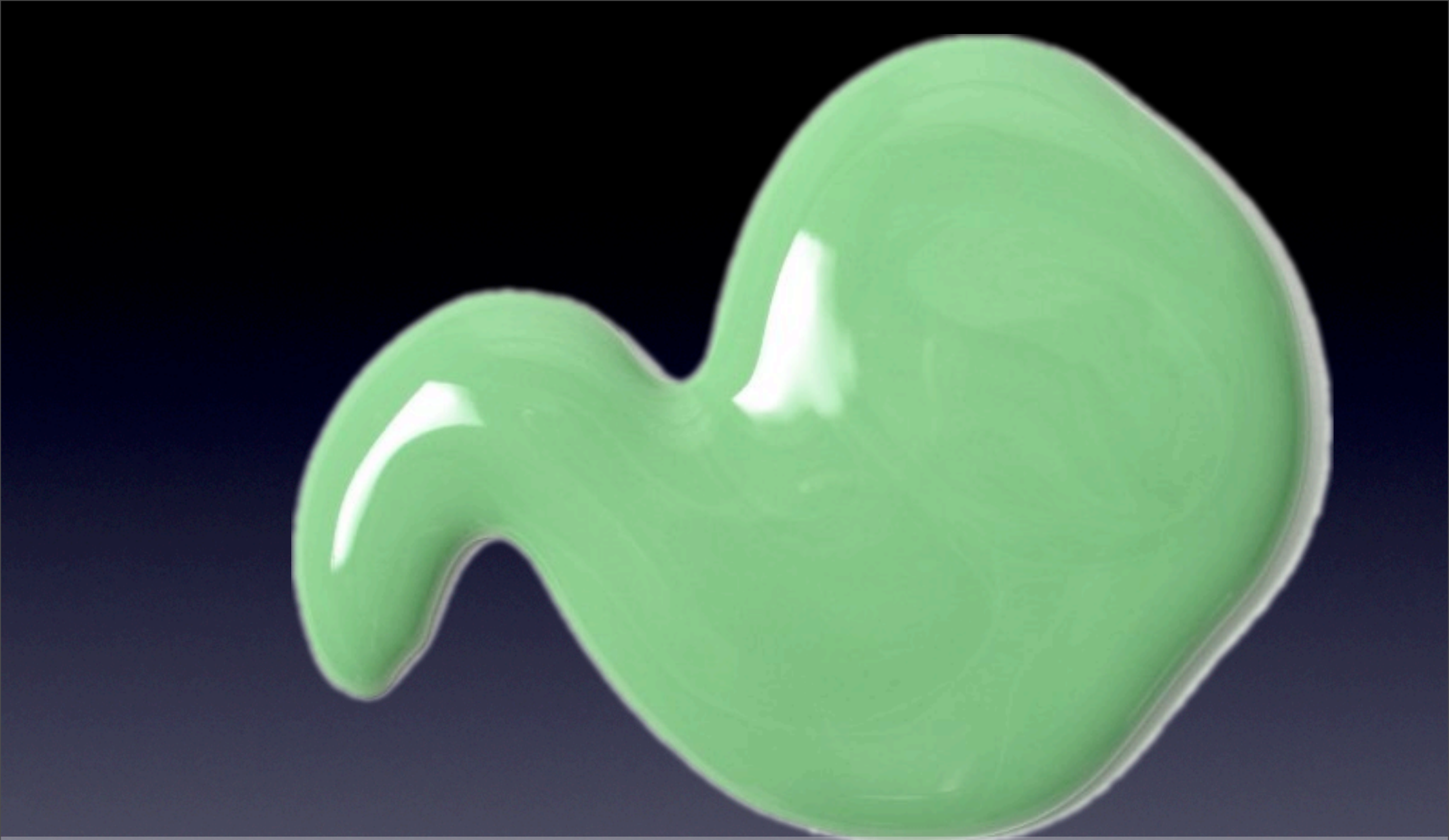
File System

File System

- Uploading multiple files is incredibly tedious!
- Native OS look-and-feel
- Access does not persist
- User has full control

File System

```
interface FileSystem {
  File[] loadFiles(FileLoadOptions)
}


interface FileLoadOptions {
  string accept     // MIME types
  bool   multiFile  // default is true
}


interface File {
  string name  // no path, for privacy
  Blob   content
}
```



Blob

What is a Blob?!

- Generic way to pass around binary data
  - Just a handle
- Gears APIs can accept Blobs
  - HttpRequest
  - WorkerPool

Blobs

```
interface Blob {
  integer length
  Blob slice(start, length)
}
```

- Semantics:
  - Blobs are immutable
  - APIs can return a new Blob

Fixing large uploads

("Connection aborted" after 500 MBs... not fun!)

Resumable Upload “API”

- No new API necessary!
  - Blob.slice() + HttpRequest.send(blob)

Resumable Upload “API”

```
var CHUNK_BYTES = 1000000;  // 1 MB (arbitrary)

function ResumableUpload(blob, httpRequest) {
  var bytesDone = 0;
  while (bytes_done <= blob.length) {
    httpRequest.send(blob.slice(bytesDone, CHUNK_BYTES));
    // On success, increment bytesDone.
    // On failure, resume at current offset.
  }
}
```

Demo:

uploadmovietool.appspot.com

Geolocation API

Geolocation API

- Hands back lat, long, accuracy
- Will use best provider
  - GPS, Wifi IDs, Cell IDs, IP address
- Pluggable service providers

Geolocation API

```
interface Geolocation {
  Position lastPosition  // zero delay
  void getCurrentPosition(PositionCallback,
                          optional PositionOptions)
  integer watchPosition(PositionCallback,   // returns id
                        optional PositionOptions)
  void clearWatch(id)
}

interface Position {
  readonly double latitude            // degrees
  readonly double longitude
  readonly double altitude            // meters
  readonly double horizontalAccuracy
  readonly double verticalAccuracy
  readonly Date timestamp  // when location established
}
```

Geolocation API

```
var geo = google.gears.factory.create('beta.geolocation');

// Get the latest position.
geo.getCurrentPosition(function(position) {
  UpdateMap(position.latitude, position.longitude);
});

// Watch the position over time.
var watchId = geo.watchPosition(function(position) {
  UpdateMap(position.latitude, position.longitude);
});
geo.clearWatch(watchId);
```

Demo:

lastminute.com/radar

Questions?

- Google Gears is an open source plugin that aims to push the Web forward
- The components are simple to use

  - http://code.google.com/apis/gears/
  - http://gears.google.com/

  - Thanks for your time!

Download Slides

Introduction to Gears

Brad Neuberg
Google