

Doing the Write Thing

Louis Suárez-Potts & Peter Kelly



APACHE CON
EUROPE

CORINTHIA HOTEL
BUDAPEST, HUNGARY
— NOVEMBER 17-21, 2014 —



It's 2014 now...

Desktop

Commercial office suites

4

MS Office, iWork, ...

Open source office suites

4

OpenOffice, LibreOffice

Web

4

Google Docs, MS Office, iWork, ...

6

Nothing

Mobile

4

MS Office, iWork, ...

6

Nothing



Let's talk about portability

HTML & CSS

Extremely flexible and powerful today. **Let the browser handle layout** – no point in writing our own layout engine. Yes, really.

- If you want to work on layout/rendering, go and contribute to WebKit

JavaScript

Rely on modern, high-performance JS engines: JavascriptCore, V8, SpiderMonkey

Minimize dependencies

Keep building and porting **simple**

Don't impose a specific UI or toolkit

Let people do what they want – native apps with UI suitable for platform, cross-platform toolkits, command-line tools, server-side components

Re-usable libraries

Make them easy to understand and integrate. Anyone can build **open source** or **commercial** apps on top.



HTML/CSS as a native file format

HTML



Ubiquitous

Billions of documents.
Used by everyone, every day.

CSS



Well-implemented

Mature, open source runtimes.
e.g. WebKit/Gecko

Understood

by millions of developers



HTML/CSS as a native file format

Pros

- Simple, easy to learn
- Responsive design
- Powerful layouts with CSS
- Well-supported in browser engines
- Great tooling ecosystem
- Used universally on the web
- Not ODF

Cons

- Poor support for print layout
 - Pagination
 - Headers/footers
 - Footnotes and endnotes
- Not traditionally thought of as a word processing file format
- For text-based documents only, not spreadsheets or presentations
- Not ODF



Incubator proposal

The Corinthia Project

Portable libraries for building office apps + a set of related open source apps

Existing components:

- **Converter** (C library) – supports .docx ↔ HTML translation
- **Editor** (JavaScript library) – currently word processing only

Origin: UX Write (iOS app store), > 2 years on market, many thousands of users

New components:

- Editing front-ends for desktop, mobile, and web
- Spreadsheet, presentation, drawing facilities in editor
- File conversion filters (ODF, RTF, .docx, Markdown, LaTeX et. al.)
- Batch conversion, indexing, version control, content management...
- etc.

The only limit is your imagination



Using the libraries to build an application

Your App



Editor

Run inside embedded web view (e.g. WebKit) in native app

Run in IFRAME within browser

App → Editor: ~140 API functions
(not all need to be used)

Editor → App: ~10 Event messages
(informs app about changes)



Converter

Bidirectional transformation

get(Source) → View

update(Source, View') → Source'

Only updates – does not overwrite.
Unsupported elements &
formatting are fully retained



Discuss

<https://github.com/uxproductivity/Corinthia>