

Mobile Productivity

Louis Suárez-Potts & Peter Kelly



APACHECON
EUROPE

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



The ODF has failed on mobiles

The OpenDocument Format (ODF) has failed on mobiles—tablets—leaving the field to the usual suspects.



Why?

There is no significant vendor of devices or software making and distributing an ODF implementation for general use.

All major mobile vendors tightly link device to OS to apps, excluding casual inquiry and limiting sophisticated intervention.

None has seen fit to include ODF support in its mobile app.

That is not likely to change. There is a desire and even a market for ODF. But it can be satisfied by less open formats and their implementations and this provides compelling reason to continue with the enclosing we see.



Why?

The problem of ODF goes beyond format. There are no major open source productivity apps on mobile.

Efforts to change that have not succeeded.

There are reasons for this lack. It's a new field, an uncertain market, a transitional space—from entertainment to productivity—and so on.

We could be hopeful. We could hold our breath. With excitement. Wait for next year.



Solution?

We argue that rather than build an open and dedicated implementation of ODF for mobiles—which can be done—it is more effective and efficient to build on popular code and coding practices.

This means HTML5, WebKit (same) and JavaScript.

The result would be a converter able to work with the major formats. There would be no vendor lock in.



Solution?

Our argument is pragmatic.

- * JavaScript and HTML work on all mobile systems—and on desktops, too
- * JS in particular is increasingly popular
- * Forming and maintaining productive communities building JS and HTML apps is likely easier—and more affordable
- * A converter able to read and write to a variety of formats nullifies the threat to open source already detailed



Standards

How important is it to maintain the ODF as an open standard?

It's a strategic issue, and mobile has radically changed the campaign. Did anyone notice?



BYOD Standards

We won the desktop. But the desktop is less relevant to the future use. Further, mobile adds new interfaces—the touchscreen—whose inclusion in the desktop has proven uncertain.

The new possibilities offered by mobile's form factor and capabilities are not liabilities. They are to be exploited as advantages. We propose to do this.

The desktop will remain. But the excitement lies in mobile. And it is not clear that an open standard carries the same force in the BYOD world.



BYOD Standards

We won the desktop. But the desktop is less relevant to the future use. Further, mobile adds new interfaces—the touchscreen—whose inclusion in the desktop has proven uncertain.

The new possibilities offered by mobile's form factor and capabilities are not liabilities. They are to be exploited as advantages. We propose to do this.

The desktop will remain. But the excitement lies in mobile. And it is not clear that an open standard carries the same force in the BYOD world.



Summary

To recapitulate. We are not opposed to open standards or to the bodies that maintain them.

But it would be foolish to rely on any open standard to preserve coding freedom—or any freedom.

(A loud voice, a soapbox and a park is not a throne with an army.)

We also recognize that the development of anything of the hoarding ambition of last century's office suite is madness.



Complicated?

We therefore propose channelling our efforts into projects that use popular languages and architectures and whose size keeps the effort feasible, intimate, innovative.

To succeed globally, on mobile as well as on desktops (via a browser, say), any resulting code will need to be able to work with existing open standards.

Otherwise, this would only be an effort of futility, a work of bad conceptual art.



We are...

Louis Suárez-Potts, Age of Peers

@luispo

luispo@gmail.com

Peter Kelly, UX Productivity

kellypmk@gmail.com

<http://www.uxwrite.com>