On-Line and In your Pocket

A glimpse into some futures ...

Michael Meeks
michael.meeks@suse.com

“Stand at the crossroads and look; ask for the ancient paths, ask where the good way is, and walk in it, and you will find rest for your souls…” - Jeremiah 6:16
The message in a nutshell ...

- The challenge
  - Things you already know
- Some ways to get there ...
- Two Prototypes
  - Web Office
  - Android Office
- LibreOffice 3.5
  - how we're executing ...
- Conclusions
Let's think about the Future!
A glimpse into the future:

“In the future, Cloud computing **will be** the **only** choice”

“Successful businesses may soon have no chief executive, no headquarters, and no IT infrastructure”

*The Telegraph *inverviewing Nick Martindale
Another glimpse into the future:

“App Stores: the future of desktop apps?”

“Over 300,000 iPhone and iPad apps and 10 billion downloads later, Apple has taken the concept to the desktop”

APC Mag
Mix and match client apps ...

- Web / SaaS / cloud-on-line everythings
- solve the deployment problems nicely
- often add **cool** collaborative features wiki-style
- Major **regressions**; WYSIWYG, off-line access eg.

- App Stores
  - **wildly** popular: Apple, Intel, Amazon, Microsoft etc. ...
  - Apple: 5 million downloads per day (~2bn per year?)
  - We should do that on Linux! - oh ...

- So just **'mash'** the two together!
  - Web App – App-Stores ...
  - I'm just 'installing' my Web-App etc. etc.
Searching for the magic bullet!
(how we keep finding fakes)
Have we found the magic bullet?

“you can easily realize a ten-fold productivity improvement just by using language: [insert: Lisp, Smalltalk, Java, C#, …]”

“it will all be so simple that feral children and trained monkeys will write software for free!”

I also have some prime waterfront real-estate at an unbelievable price for you!
CODE RE-USE …
(you know, the thing no-one wants to do)

- The best line of code, is the one we didn't have to write.
- Free Software makes this easy
  - cf. Proprietary 'Components', version mgmt. etc. etc.
  - we actually get quite a lot of this right

**But** re-writing is such fun!
Collaboration & re-use is boring!
A quick history of (failed) Office re-writes ...

  - MicroPro – once biggest S/W company in the US!
  - WordStar - #1 Office suite in 1980's
  - killed the company by re-writing / duplicating their product
- Corel Office (in Java)
  - beautiful Java technology demonstration in 1997
  - never shipped final version, squandered cash & position
- Microsoft: Pyramid … - wikipedia

Word 6.0 was the second attempt to develop a common code base version of Word. The first, code-named Pyramid, had been an attempt to completely rewrite the existing product. It was abandoned when it was determined that it would take the development team too long to rewrite and then catch up with all the new capabilities that could have been added in the same time without a rewrite.
Not a terribly new idea:

“Every kingdom divided against itself will be ruined, and every city or household divided against itself will not stand” -Matthew 12 ~30AD
A different approach ...

- Re-write in Javascript message
- Corel heard ~this before
- ~10 million lines of type-unsafe JavaScript not a win.
  - perhaps you disagree, if so please go for it … elsewhere.
- Telephone companies don't believe in themselves:
  - Where is your address book ?
  - How about your photos ?
  - WHAT ? You have a big whack of flash & a monster CPU in your (increasingly fat) phone ? Why ?
    - Your phone GPU has that many transistors !
  - Why is it not all 'in the cloud' ? - with your
A design that re-uses code ...

Cut one feature to share the rest: the off-line web app

- Help
- User Mgmt.
- Sandbox
- Web UI

- Templates
- Clipart
- Traditional UI: Linux/Mac/Win

- Translations
- Database
- Exotic filters

- Impress
- Calc
- Writer
- Android / Handset UI

95% of code shared, most featureful is the web app!
What a crazy plan !?

- It is possible eg. Linux Kernel ...
  - Substantially the same code on your:
    - Phone
    - Super-Computers
    - Cluster of doom ...
- The joys of sharing a common code-base ...
  - Rendering bug fix → fixes everyone
  - Interoperability / import fix → every platform benefits
  - Performance wins for small devices:
    - Notice that 15%+ of CPU on startup is config parsing
    - And 20% of that config is 'Label descriptions' fix it!
The joys of sharing a code-base

- Build / Debug 95%+ on the fastest hardware...
- Interoperability / import fix → every platform benefits
- UI improvements:
  - Touch usable / pretty style selection eg.
  - Mouse-wheel to zoom vs. pinch
- Device blurring:
  - Intel / Ultrabooks → now with touch screens …
    - Sexy ultra-thin-ness etc.
  - Android tablets with detachable keyboards.
    - looking very much like netbooks …
More joys: Performance ...

- Android – can only cope with 96 shared libraries ...
  - This 'impress' - ~100 libO specific shared libraries.
  - Android has ~45 system libs we need to link.
  - Matus Kukan's – GSOC project …
  - Faster linking for all, better, smaller code via LTO …

Performance wins for small CPU / devices:
- Notice that 15%+ of CPU on startup is config parsing
- And 20% of that config is 'Label descriptions' fix it!

- Full Page rendering for accelerated panning / Web Office
LibreOffice on-line ...
(getting pixels to users)
**Cloud Office / HTML5 prototype** ...

- Rendering using the **HTML5 Canvas** ...
- Communication using Web-Sockets (**now with v7**)
  - very simple plain-text event / control protocol
  - **compressed png** for screen updates ...
- Bandwidth usage **low** for simple tasks:
  - We can see what is re-rendered as we type:
  - But really as we type a character: only **this** is sent:
  - And that tiny rectangle is png compressed …
- CopyArea on the client for scrolling.

Hello World this is me typing
Uses gtk+ 3's / broadway backend to render (via some javascript shims) in the browser:
- All intelligence and rendering on the 'server' app.
- `broadway.js` - < 3k LOC ...

Still a prototype
- Rendering performance improving, but lumpy in places.

This approach allows a full-feature experience
- ie. complex text-layout, large, complex spreadsheets, WYSIWYG (eg. edit your footnotes on the page), VBA macros, Pivot Tables etc.

Also a number of challenges:
- Good APIs for printing etc.
Example:
Custom contoured picture, with text fill

Gnome/OpenOffice 1.1

A small table with an embedded imager

This is a picture of our wedding with some nice text flow going on around it, of course you can edit a custom contour as well as having a crude ellipse, which makes for a very nice look. Below it is an alpha blended free-hand, filled spline with drop shadow and gradient.

And chart of the week:

Persistant Sins

LibreOffice
Example:
VBA macro driven spreadsheet button workflow that proves Novell costs you less :)
It can't possibly perform!

- LibreOffice in-use in multi-user remote/X environments
- Load avg 0.05/0.3/0.34 for 73 live users in writer, and 10 in calc.
- 4 CPU Quad-Core estimated 200-300 concurrent users.
- If rented servers in the cloud are truly cheaper it should be cheap.
Show me the code!

- Shipping out of the box in LibreOffice 3.5
- Work continues on master; -plenty- to fix.
- To use a modern browser ie. v7 websockets
  - you need master gtk3+
  - or an update to openSUSE 12.1
- Otherwise – that's it:
  - `git clone git://anongit.freedesktop.org/libreoffice/core`
- Hacking help **very much** appreciated
- Plenty of VCL/gtk3/rendering bugs
  - no browser required.
- Funky – tunnelling Browser Javascript ↔ server UNO
- Improved performance & page render / pushing
Android ...
(something in your pocket)
A pre-proto-type ...

- I'm a fraud! The real work & glory should be addressed to: Tor Lillqvist's hours of toil
- Cross compiling work originally for using MINGW → windows
- Also useful for cross-compiling to Android, iOS etc. from Linux/Mac
- A staggering learning curve to climb
- But we're getting there slowly …
- Latest code in feature/android branch …
- Dozens of calc unit tests running flawlessly, and …
Size matters, but less than you might think.

- Big?
- But small compared to modern, popular Android games
- Fast is key ...
- 200Mb small – and tons of scope for optimisation.

Prototype...
Notice the poor sizing and general hackery

But – more encouragingly
Tor has scads of unit tests running ...
This is a pre-prototype hack, with no ship date and is clearly highly sub-optimal.

Obviously the menu/toolbar/UI framework will need replacing for a new simple native experience.

Hackers most welcome ...
Clearly this is not near the end point!

A native UI / wrapper is required

Potential plan:

- Target a high-fidelity viewer first
- Adding limited in-document editing
- Improved shared UI controls for styles?
- Your vision / hackery here … ?

There are more mobile / tablet platforms than Android: iOS, Tizen, WP7? etc. …
Conclusions ...
LibreOffice Conclusions

- Code Sharing is nearly the silver bullet
- Re-writing a huge Office suite is a pointless waste
- LibreOffice is improving rapidly
- It's a fun place to hack and make a real difference ...
- strike a dis-proportionate blow for freedom here
- On-line & Android ports are coming along nicely.
- Thanks for all of your help and support!
  - we rely on you, to prove the power of Freedom with us.
  - to get people excited about Free Software they use ...
- Please do get involved and help us ...

Oh, that my words were recorded, that they were written on a scroll, that they were inscribed with an iron tool on lead, or engraved in rock for ever! I know that my Redeemer lives, and that in the end he will stand upon the earth. And though this body has been destroyed yet in my flesh I will see God, I myself will see him, with my own eyes - I and not another. How my heart yearns within me. - Job 19: 23-27