Application Development for Mobile and Ubiquitous Computing

Whiteboard

Final Presentation

Group 14: Wolfgang Hönig, Frank Tetzel

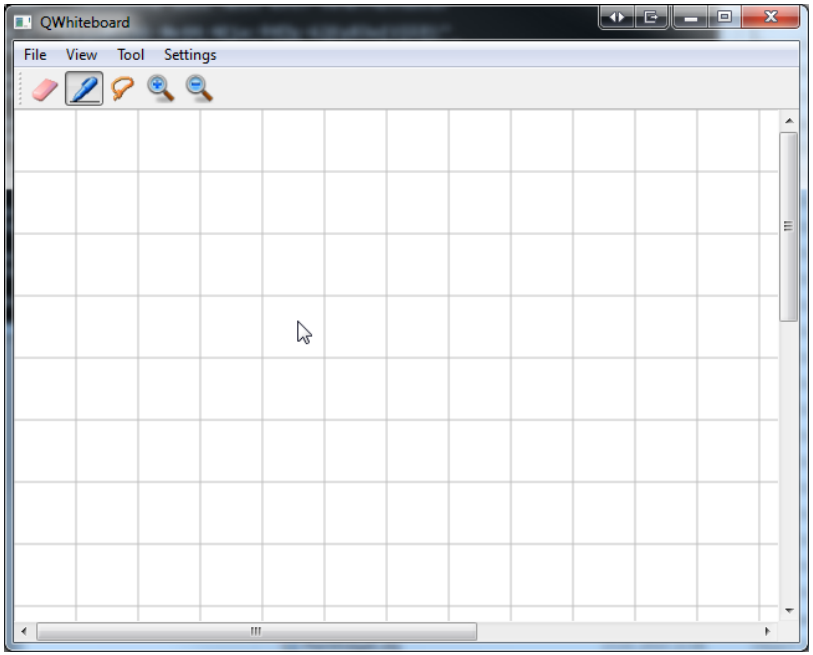
01/29/2010

Application Tour

Distributed and locally usable cross-platform whiteboard with vector graphics support.

Application Tour

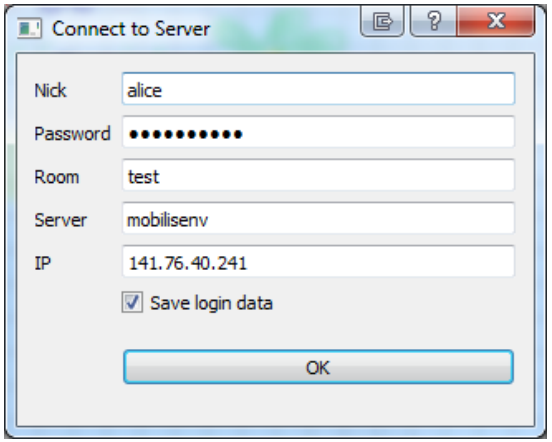
Alice uses Windows and starts QWhiteboard



Alice / Windows

Application Tour

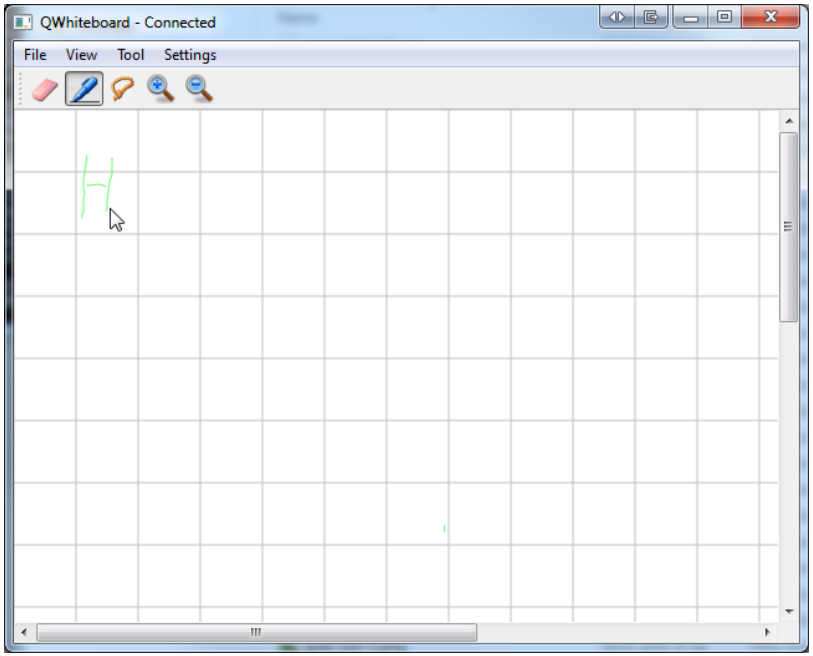
Alice connects to an OpenFire (XMPP/Jabber) server



Alice / Windows

Application Tour

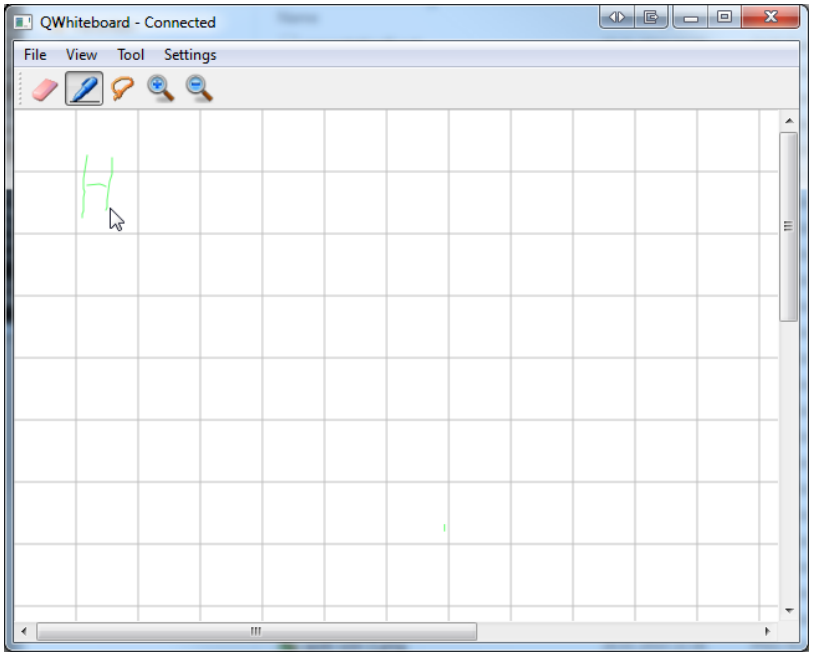
Alice starts drawing using the pen



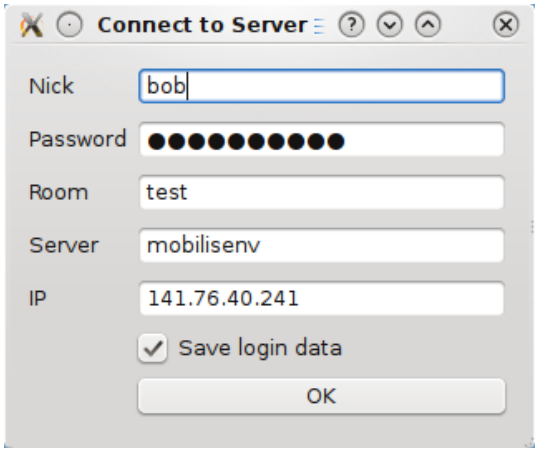
Alice / Windows

Application Tour

Bob (using a Tablet PC running Linux) joins the session



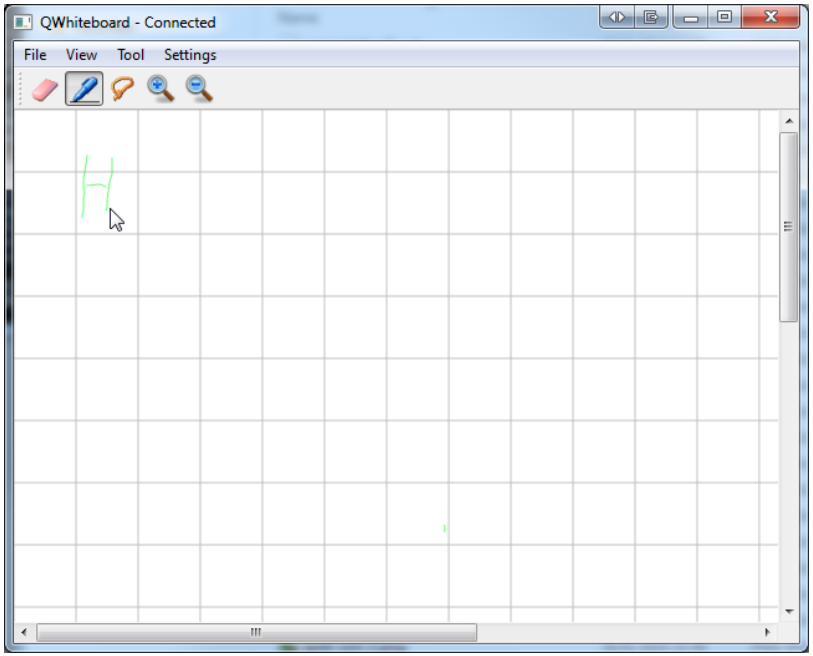
Alice / Windows



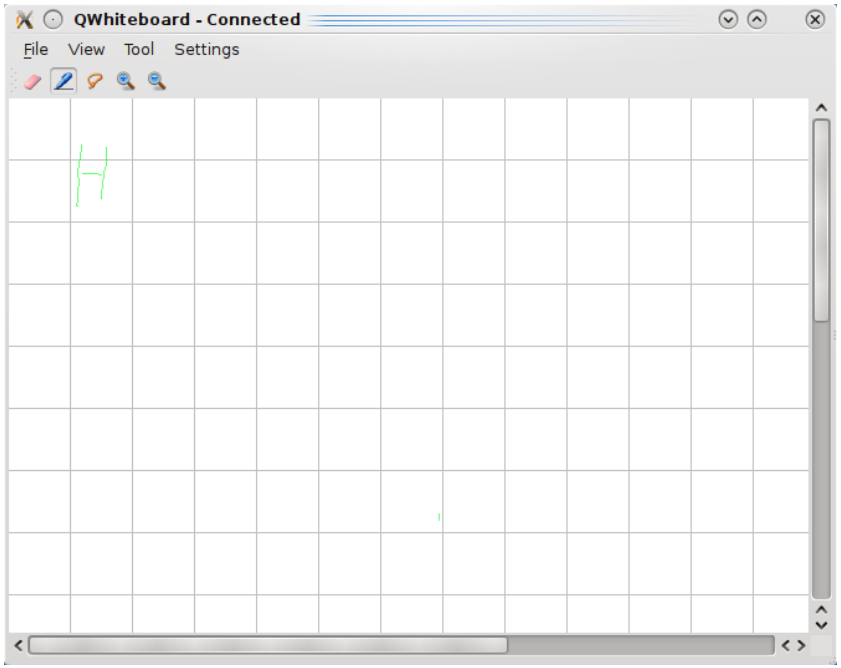
Bob / Linux

Application Tour

The screens will be automatically synchronized



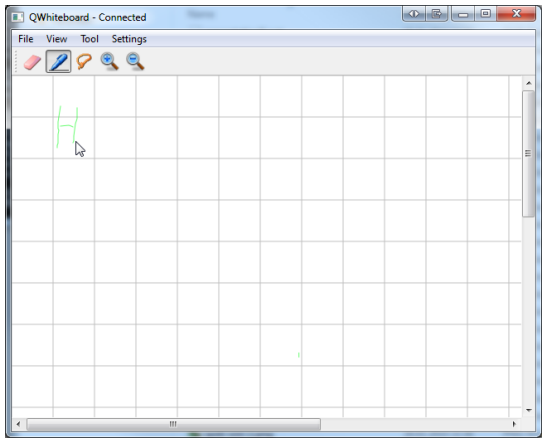
Alice / Windows



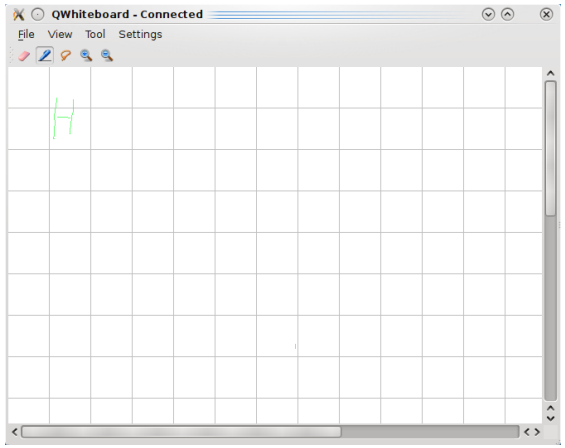
Bob / Linux

Application Tour

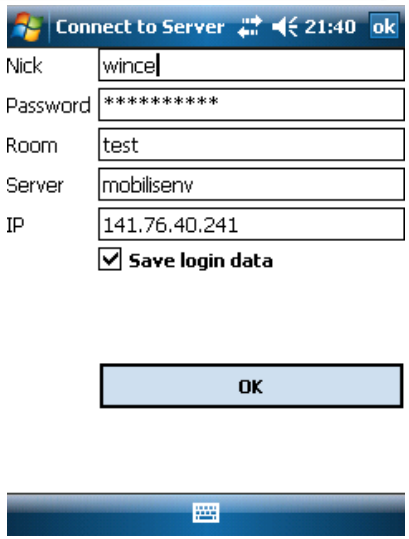
A Windows Mobile user wants to join the meeting



Alice / Windows



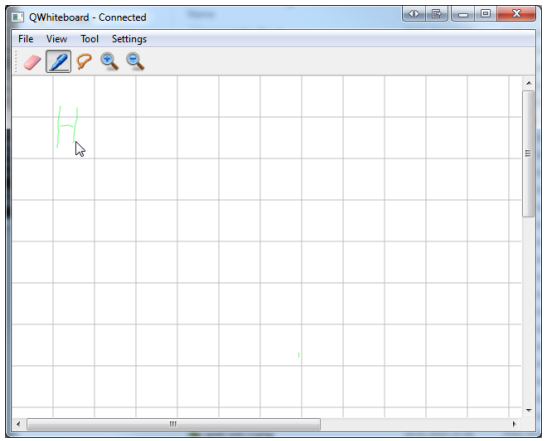
Bob / Linux



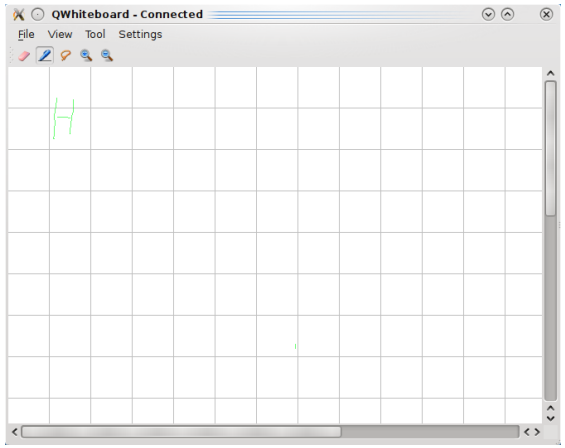
Windows Mobile

Application Tour

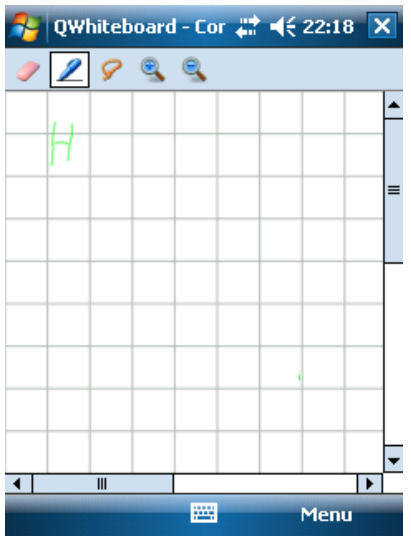
A Windows Mobile user wants to join the meeting



Alice / Windows



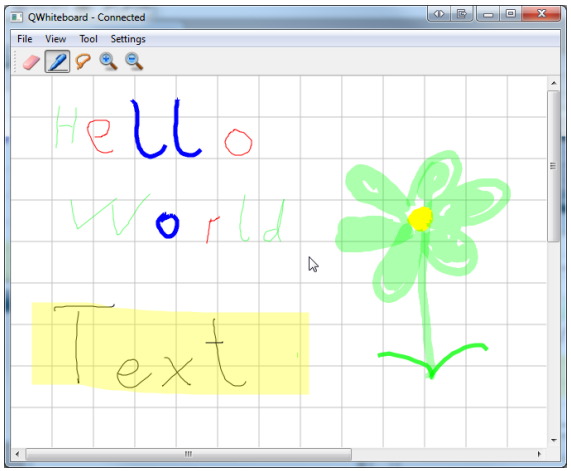
Bob / Linux



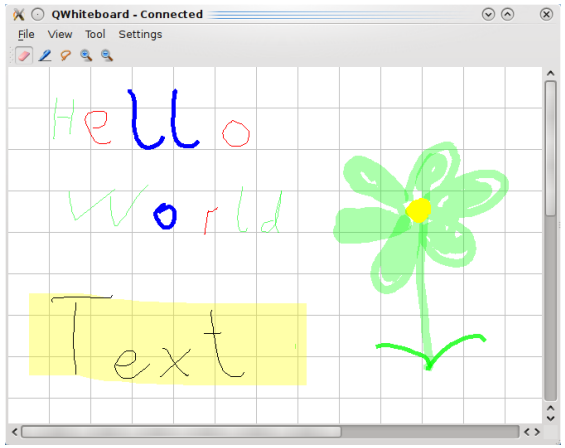
Windows Mobile

Application Tour

During the meeting there is a lot hard work to do:



Alice / Windows



Bob / Linux

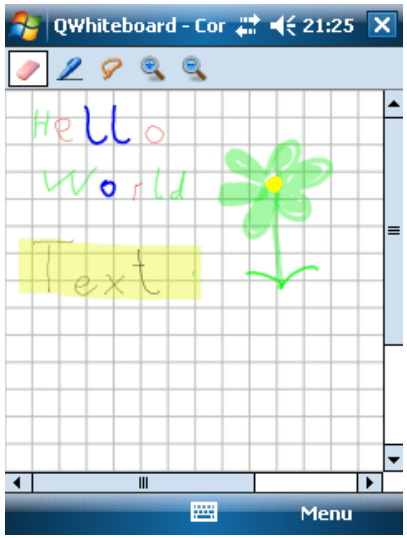
Windows Mobile

QWhiteboard takes care about synchronizing the screen contents.

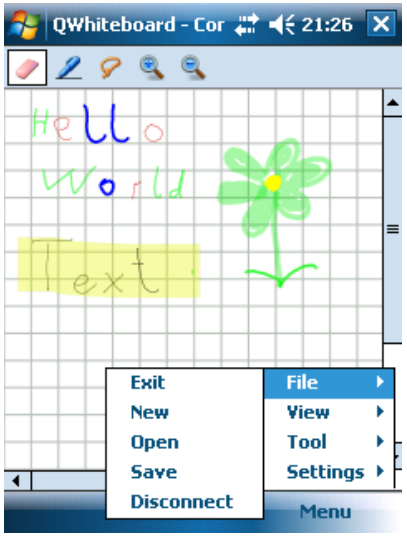
Application Tour

While drawing the WLAN connection of the Windows Mobile user frequently disconnects. QWhiteboard detects that automatically and caches all data and tries to synchronize it later, if the connection is back again. The user will only notice that by looking on the title bar.

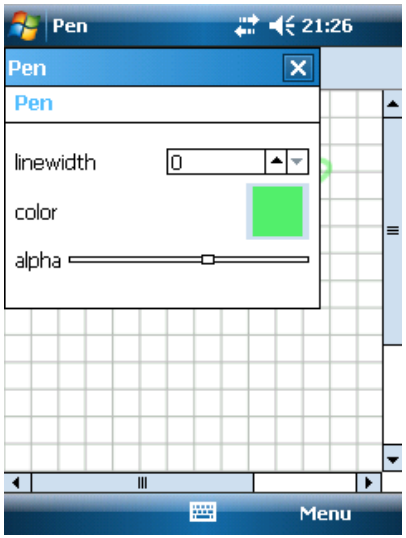
Other nice features



zooming

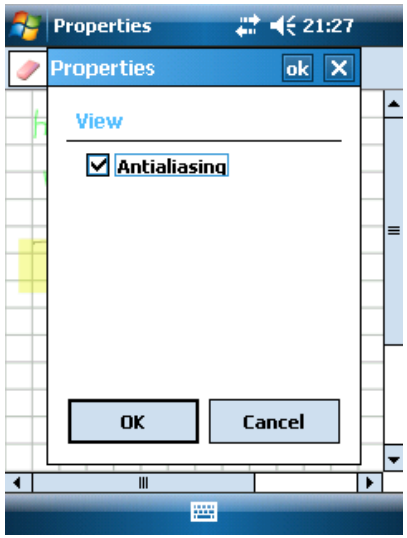


open / save



change pen

Other nice features

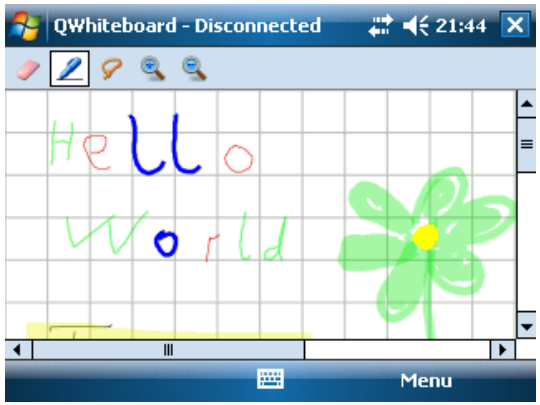


antialiasing

select group

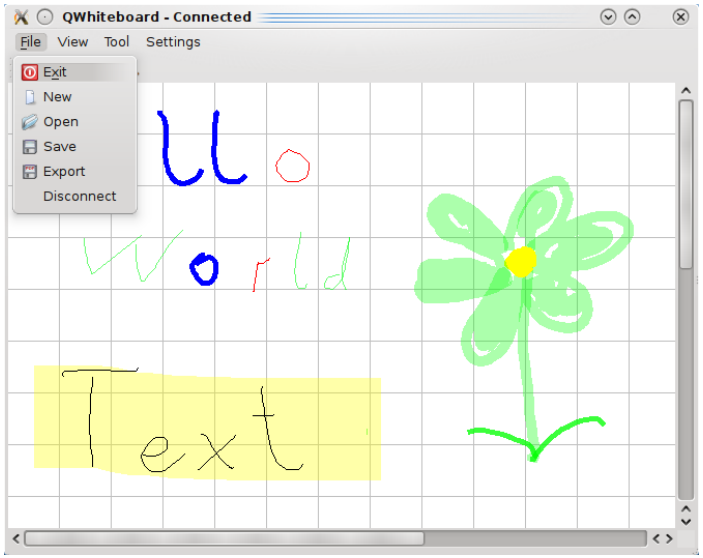
move group

Other nice features

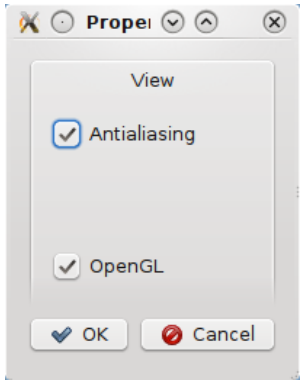


other resolutions or rotated

Other nice features - only some platforms



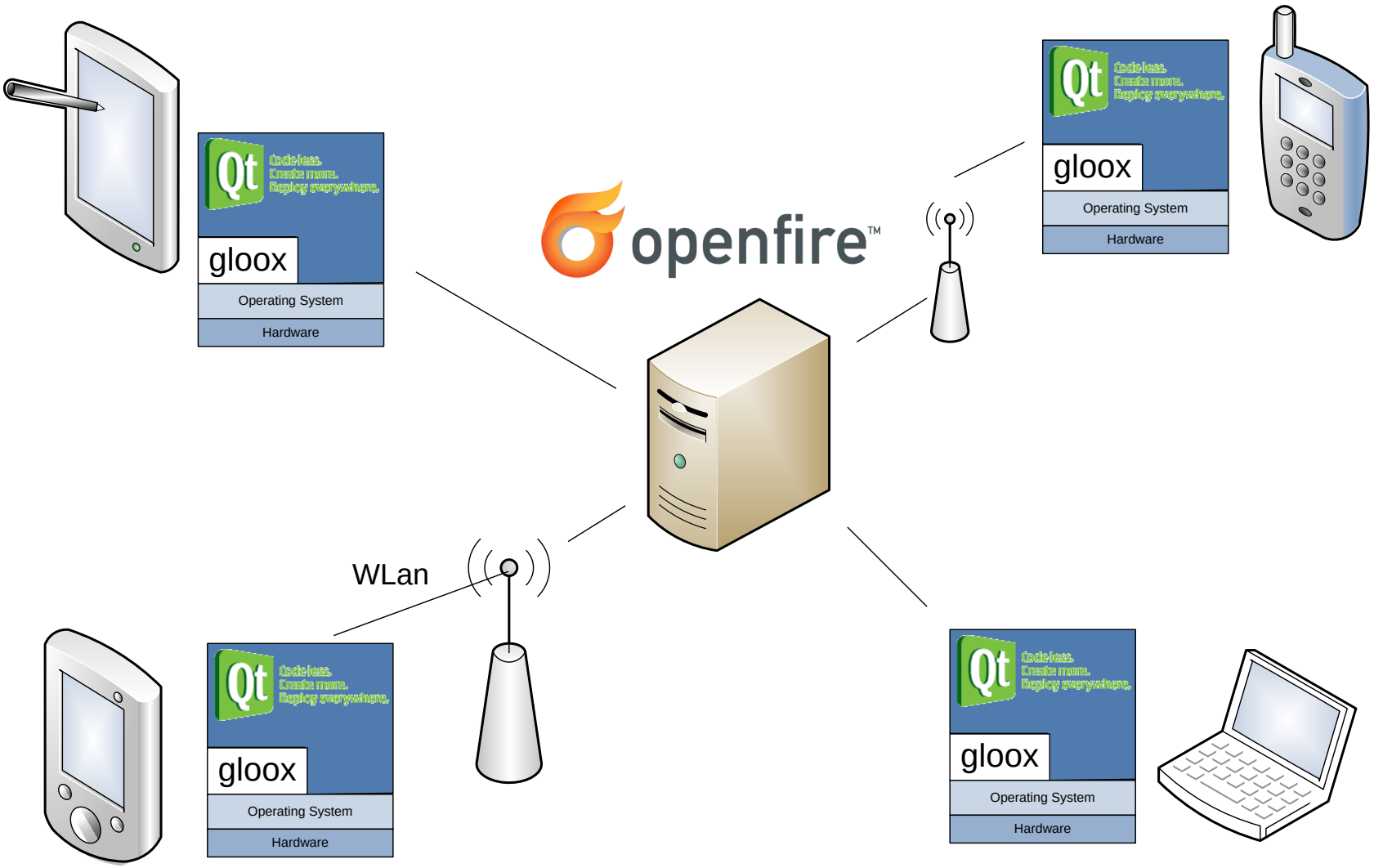
PDF export



OpenGL acceleration

special support for tablet pc eraser

Technologies



Challenges

- different screen sizes
  - vector graphics in QGraphicsView
- Heterogeneity
  - only crossplatform libraries, i.e. gloox, Qt
- Bandwidth, Latency
  - binary code (QDataStream)
  - compression of gloox
- Disconnections, Consistency
  - OpenFire saves history, reconnect gets current state
  - cache on client site

Pitfalls

- JXTA-C is not usable
- disconnection events do not work properly in gloox: needs polling hack
- some odd disconnections $\Rightarrow$ need whitespace pings
- Qt has some issues on WinCE:
  - complex ui-layouts have bugs
  - no printer support
  - minor menu and drawing bugs
  - shared library too big for some devices (especially Qt 4.6)
- dynamic_cast expensive on mobile devices

What we've learned

- Qt is very nice and easy to use for cross-platform development
- issues for mobile devices have to take into account at first stage of planning
- crosscompiling (libs etc.) can be hard due Windows Mobile limitations
- development tools (Visual Studio / Device Emulator) very useful (no surprises after deployment on real device due binary compatibility)

Thanks for your Attention!

Questions?

Download at

`http://sourceforge.net/projects/qwhiteboard/`