

MAEMO FOR A SYMBIAN DEVELOPER

Ville Voutilainen



INTRODUCTION AND AGENDA

INTRODUCTION AND AGENDA

- Differences between Maemo and Symbian
- Similarities between Maemo and Symbian
 - Example of manual resource management on both
- What's better in Symbian
- What's better in Maemo
- Examples of things you can do better in Maemo
- Future convergence towards Qt

DIFFERENCES AND SIMILARITIES BETWEEN MAEMO AND SYMBIAN



HOW MAEMO IS DIFFERENT FROM SYMBIAN

- Default programming language is C (although bindings to other languages are available, their use is not actively encouraged).
- Maemo is based on open source libraries and tools.
- The amount of existing libraries for Maemo is thus larger.
- Porting existing code to Maemo is easier

HOW MAEMO IS SIMILAR TO SYMBIAN

- Both use a vendor-specific UI framework (Hildon vs. S60/UIQ).
- Both are going to use Qt in the future (but vendor-specific UI may still be needed).
- Both require careful management of eg. network connections and power/memory consumption (like all mobile platforms do).

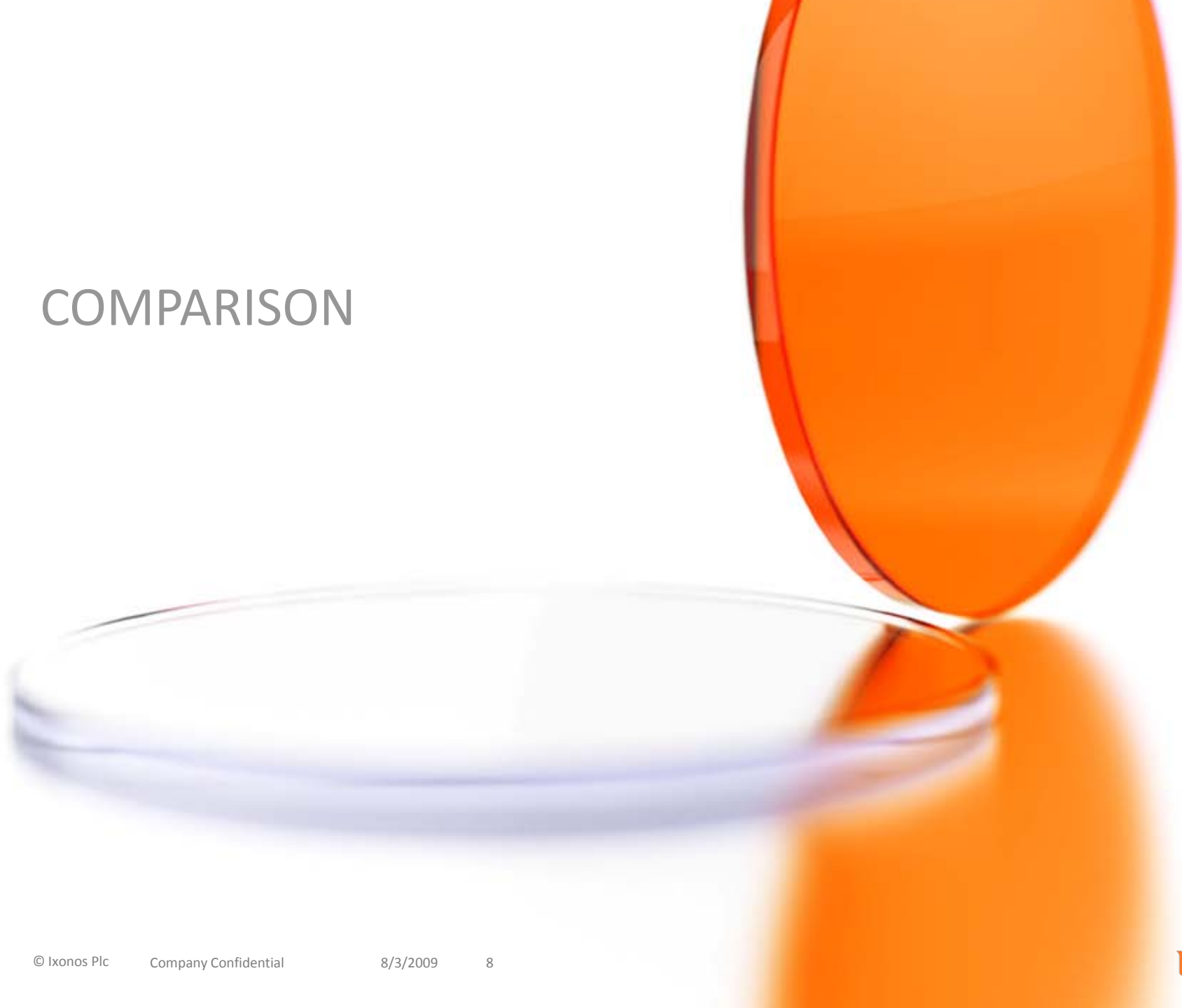
MEMORY MANAGEMENT EXAMPLE

```
GString* my_string =  
    g_string_new(NULL);  
// use my_string  
g_string_free(my_string, TRUE);
```

```
HBufC* myString =  
HBufC::NewLC(1024);  
// ..use myString  
CleanupStack::PopAndDestroy();
```

- Left side is a Maemo glib example, right side is a Symbian example
- Both require manual cleanup of any allocated resources (in this case, strings)

COMPARISON



WHAT'S BETTER IN SYMBIAN

- Using menus and dialogs is very easy (if you can write a correct resource file) and well abstracted. Maemo has declarative tools available for menus/dialogs/UI, but their usage is not actively encouraged.
- In general, the application framework does a lot of work abstracting away the details (this is both a pro and a con).
- The platform is mature and the APIs have been stable for a long time.
- The conventions are, if not universally liked, mature and relatively well known and widely used.

WHAT'S BETTER IN MAEMO

- No application signing required
- The imperative APIs are much more powerful, fewer things are hidden behind abstractions (this is also both a pro and a con).
- The libraries and APIs are available on multiple platforms (except the UI).
- Writing portable programs and porting applications to Maemo is easier.
- The availability of debugging and tracing tools is better.
- If you choose to use C++, you can use the real deal, not a vendor-specific dialect.
 - Using standard code in Maemo is straightforward, that's not the case for Symbian

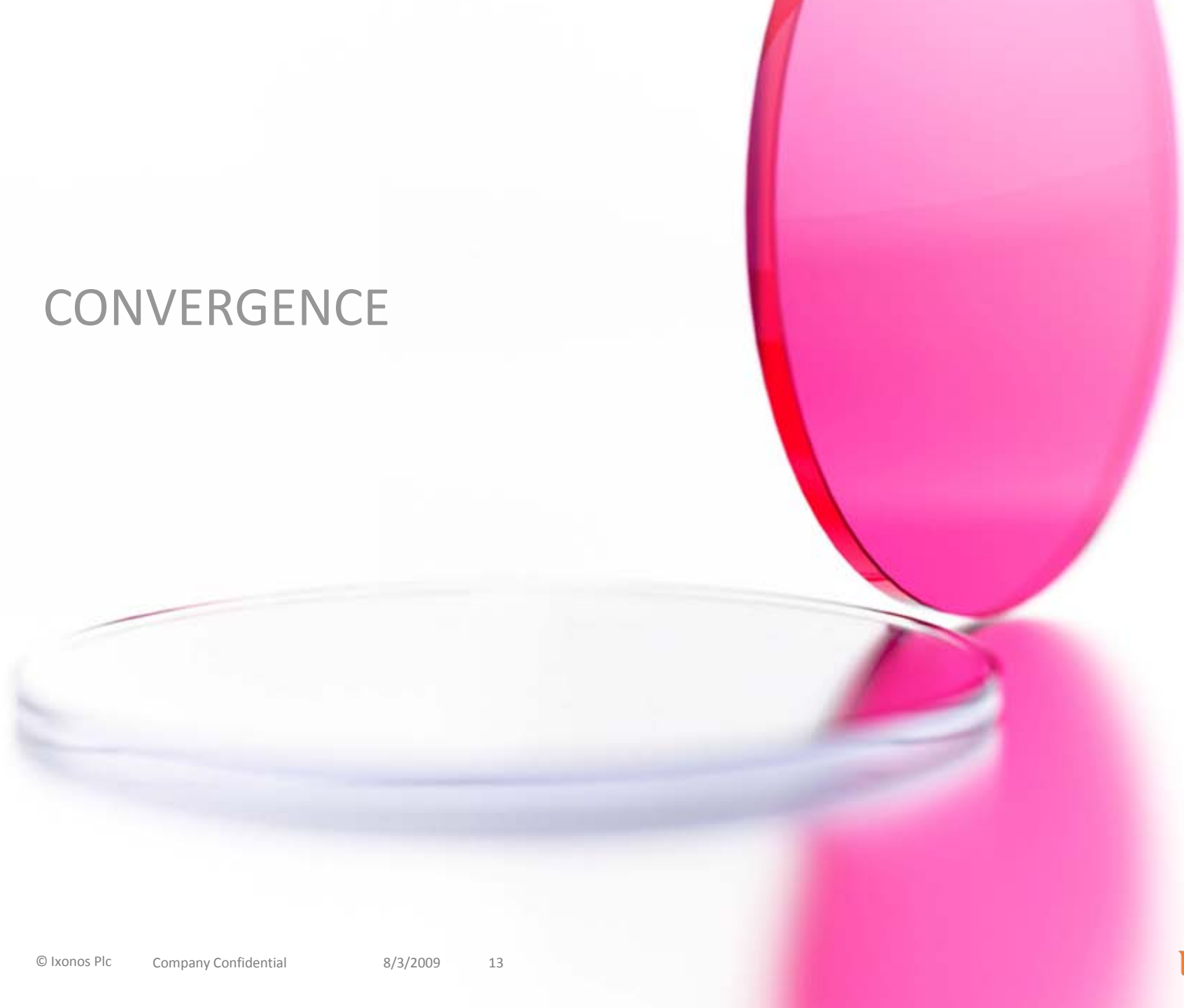
MAEMO OR SYMBIAN?



EXAMPLES OF WHAT A DEVELOPER CAN EASILY DO ON MAEMO BUT NOT ON SYMBIAN

- You can ssh onto a device and examine things, run a debugger on a device, manage processes, examine logs, etc.
- You can run a debugger or other tools directly on the device, without any network connection from the desktop to the device.
- You can run applications on the device and display them on the desktop, using a remote X connection.
- You can run applications on the desktop and display them on the device, using a remote X connection.

CONVERGENCE



FUTURE CONVERGENCE

- New Symbian platforms (like S60 5th edition) already support standard libraries like POSIX, STL and Boost, making writing portable code and porting existing code easier on Symbian.
- Both Symbian and Maemo are adopting Qt, decreasing the differences and helping with portability.
- It's unknown if UI frameworks will converge, even with Qt on both platforms.

MEMORY MANAGEMENT IN QT

```
QString myString;  
// use myString  
QSharedPointer<Foo> x(new Foo());  
// use x
```

- Note the lack of manual cleanup!
- Smart pointers provide a way to handle any free-store resources automatically, facilitating use of RAII techniques.
- Memory deallocation is automatic and exception-safe.

CONCLUSIONS

CONCLUSIONS

- Current versions of both platforms are difficult to program for regarding resource management.
- The programming APIs for either aren't very easy/pleasant to use and aren't compatible with the other platform.
- The current user interfaces for both platforms are on their way out.
- Qt will be available on future devices for both, and provides easier and safer APIs, especially considering resource allocation/deallocation.

THANK YOU!