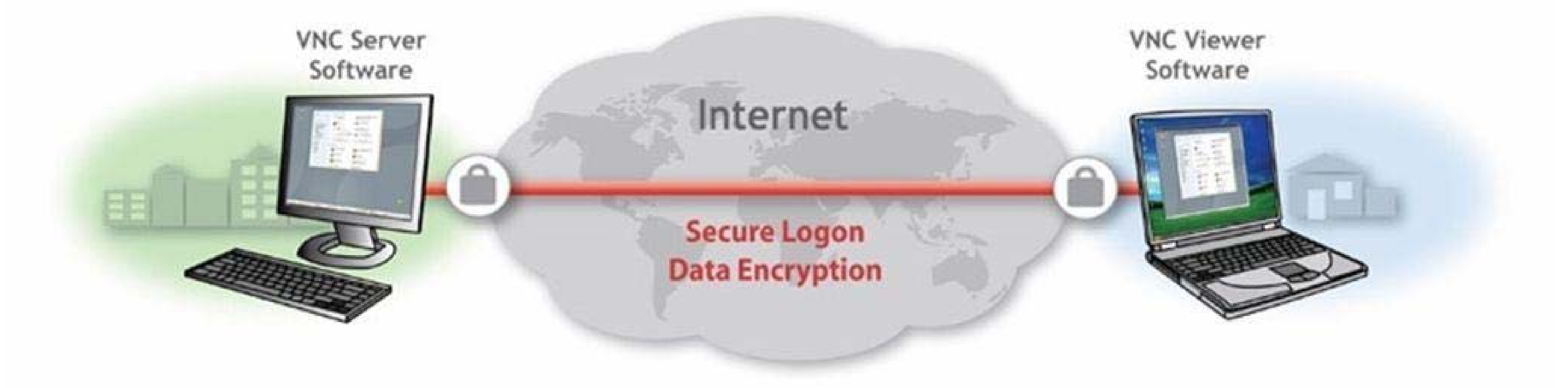
Porting VNC to Mobile Platforms

Technical and Political Challenges

Dr Andy Harter

What is VNC?



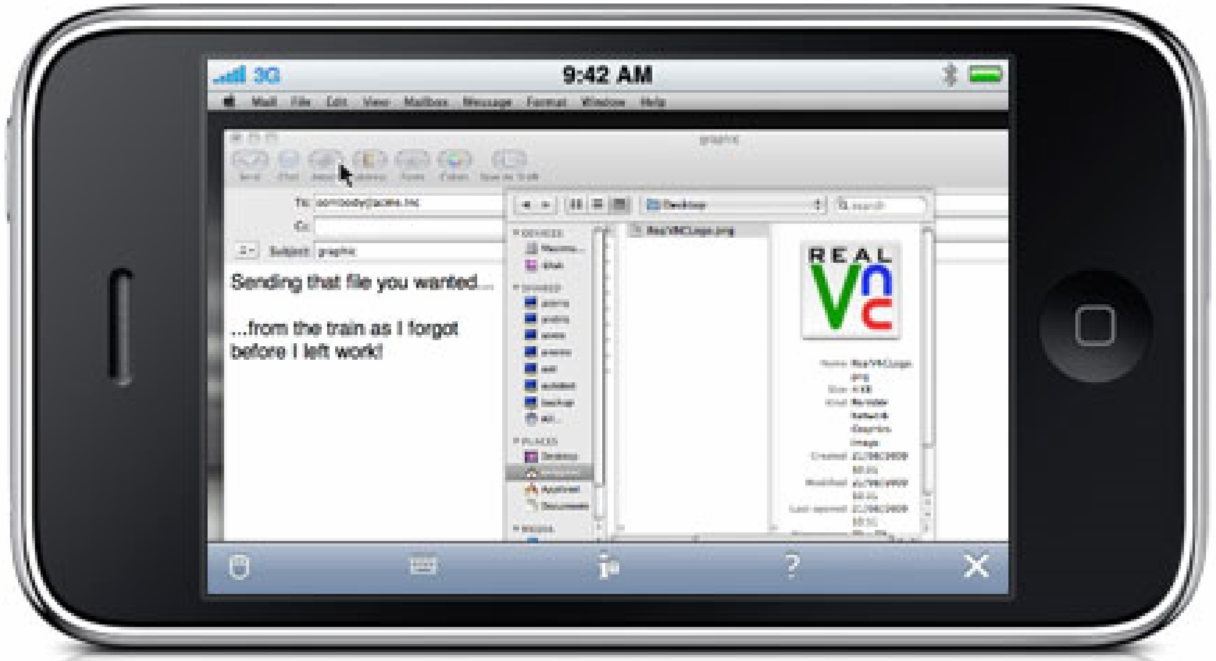
- The VNC **viewer** application takes remote control of a device using the VNC **server** application on the device
  - Views the screen
  - Controls the keyboard and mouse
- **RFB** is the protocol used by VNC to communicate between the VNC viewer and VNC server

VNC – a remote access standard

- RFB v3 protocol specification freely available
- RFC published
- Source code and executables available from RealVNC web site
  - Over 100 million downloads
  - Range of derived open source implementations available
- Embedded in a range of platforms
  - Apple remote desktop
  - Linux distributions
  - Intel 2010 i5 & i7 chipsets

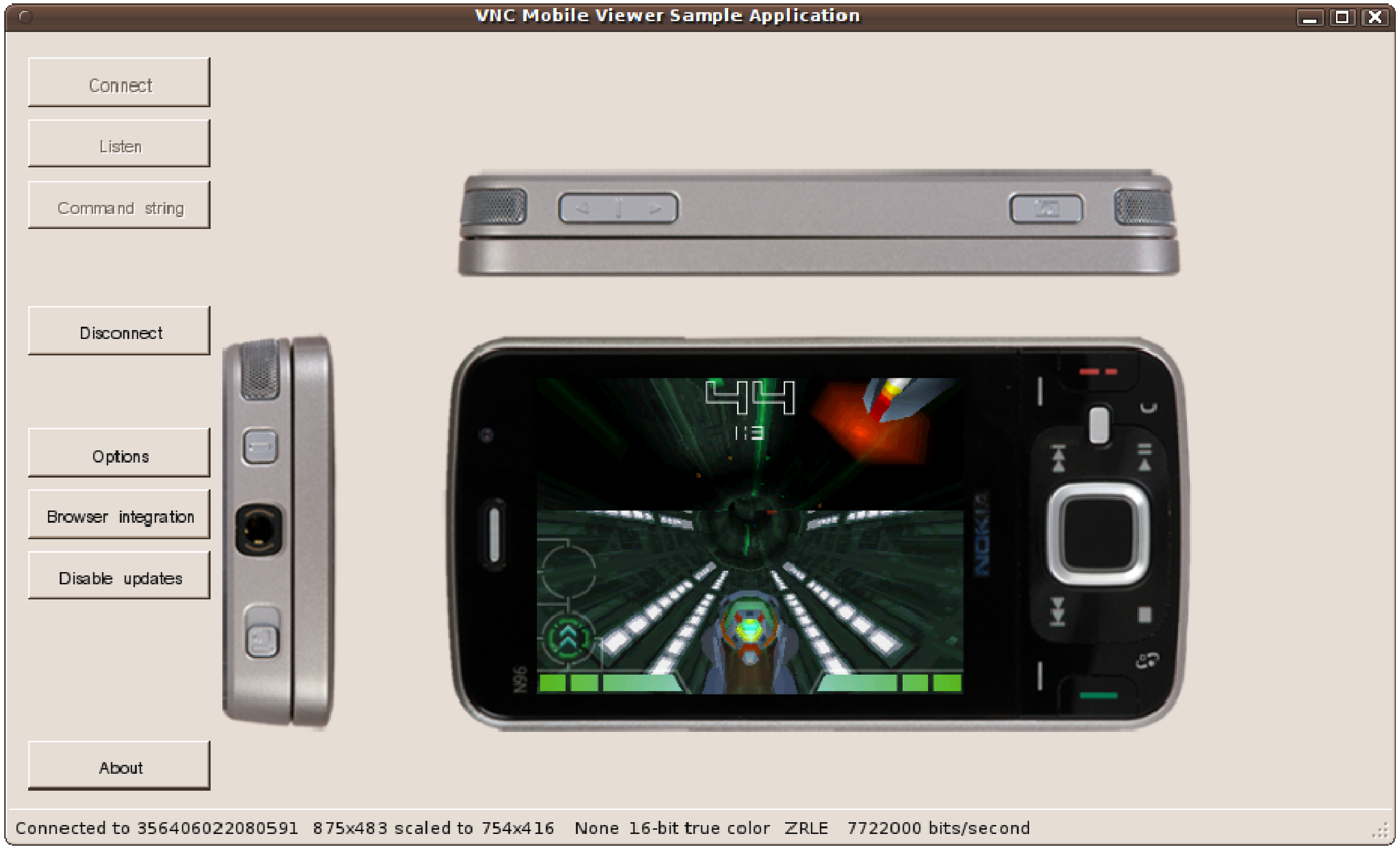
What is VNC Mobile?

- Controlling PCs from phones...


3G
9:42 AM
Mail File Edit View Mailbox Message Format Window Help
Sending that file you wanted...
...from the train as I forgot before I left work!
REAL VNC

What is VNC Mobile?

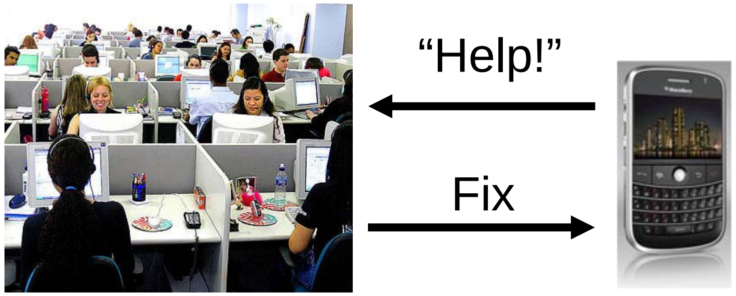
- But also controlling phones from PCs!



Connected to 356406022080591 875x483 scaled to 754x416 None 16-bit true color ZRLE 7722000 bits/second

Why remote access a mobile?

– Helpdesks



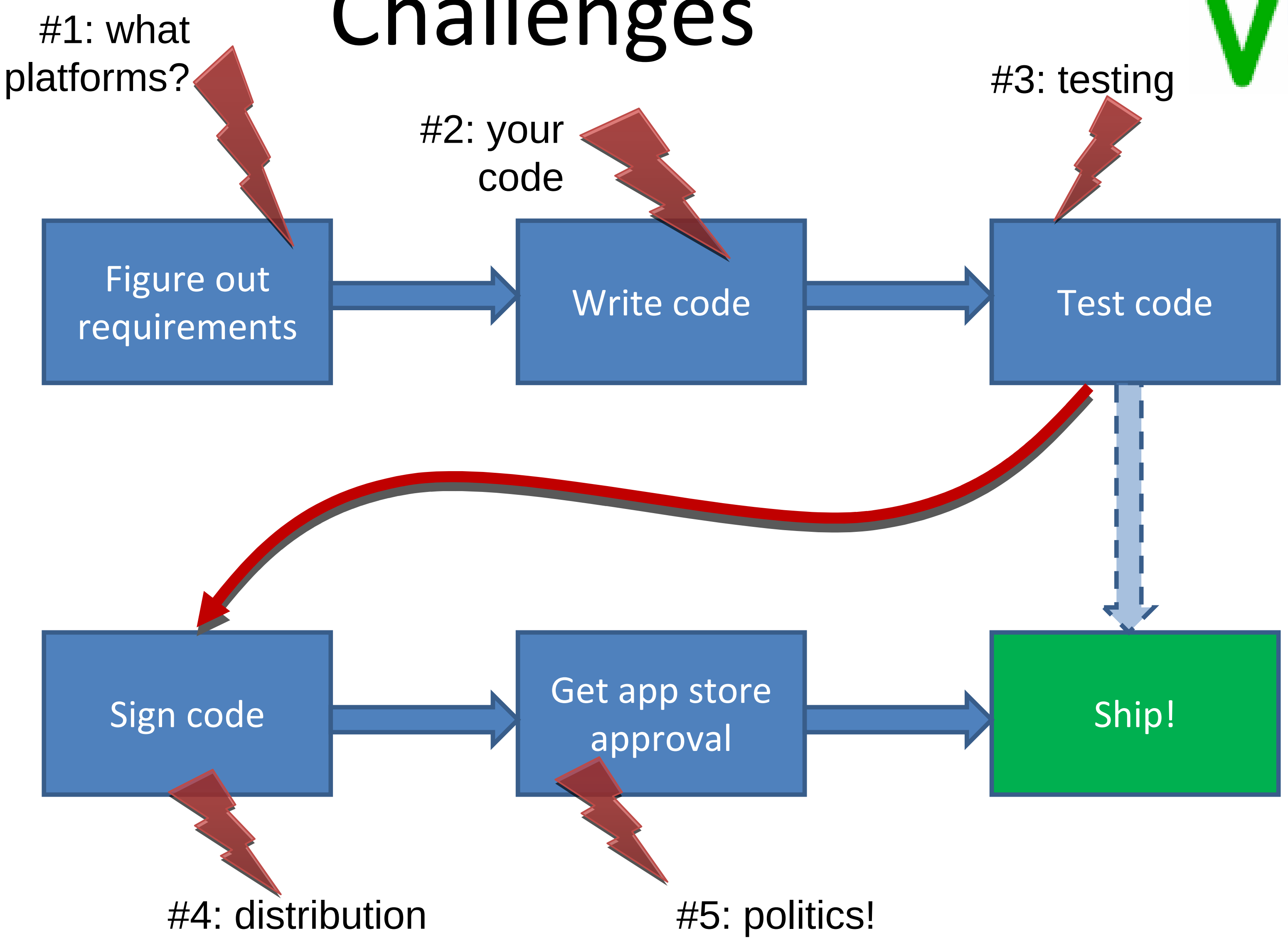
– Automotive

– Testing

What are phones?

- Phones usually run a modern multi-tasking operating system with memory protection and often even demand paging
- Some are actually based on popular desktop operating systems e.g. Linux or versions of Windows
- But they usually expose a smaller set of APIs to application developers: stronger security
- **Challenge:** How do you get from having a good solid codebase, to having a good solid codebase which users are running on phones?
- 20% of the effort is writing the code; 80% of the effort is getting it **tested** and **deployed** on phones.

Challenges


#1: what platforms?
#2: your code
#3: testing
Figure out requirements
Write code
Test code
Sign code
Get app store approval
Ship!
#4: distribution
#5: politics!

REAL VNC

Challenge #1: which platform?

- What platforms do you choose?

Windows Mobile

symbian

BlackBerry

Windows Phone 7 Series

iPhone

android

palm webOS

J2ME Java

bada

MeeGo™

VNC viewer platforms

- Requirements: multi-touch and a big screen

Windows Mobile

symbian

BlackBerry

Windows Phone 7 Series

iPhone

android

palm webOS

J2ME Java

bada

MeeGo

REAL VNC

VNC server platforms

- Requirements: multi-tasking and flexible APIs

Windows Mobile

symbian

BlackBerry

Windows Phone 7 Series

iPhone

android

palm webOS

J2ME Java

bada

MeeGo™

Thorns in our side

Everybody wants VNC servers for iPhone and Android – but they don't have the APIs we need

Everybody wants viewers for BlackBerry – but they have small screens and no multi-touch

Next version of Windows Mobile doesn't have multi-tasking: we can't write a VNC server.

Challenge #2: your code

- RealVNC has the most cross-platform codebase in the world. Runs on virtually every desktop OS.
- So you'd think it would be easy to port to mobiles...?

Which language?

Platform	UI	Back-end
android	Java	Java or C/C++
BlackBerry	Java	Java
Windows Mobile	C/C++	C/C++
symbian	C/C++	C/C++
iPhone	Objective-C	C/C++/Objective-C
Windows Phone 7 Series	CLR (e.g. C#)	CLR (e.g. C#)

At least three code-bases needed

etc.

Java Code Issues

- Java?
  - Fast these days on PCs. Still not on phones!
- Android
  - Still interprets each instruction one at a time
  - Replace with native code or wait until they fix Android!
- BlackBerry
  - Also slow but no chance to use native code
  - Abandon doing our own (a) image compression and (b) encryption; find a way to use whatever OS facilities are provided since they call into native code

“Native” (C/C++) code issues

- Faster
- But limited C/C++ libraries
- Symbian
  - No global variables
  - Weird APIs and memory management techniques
- Symbian & Android
  - No C++ exceptions
- Windows Mobile
  - Subset of normal Windows APIs
- All of them
  - Endianness issues
  - Different ways of handling asynchronous events
  - Limited set of APIs available, for greater security

Porting your existing codebase is probably going to be hard, especially if you use C++

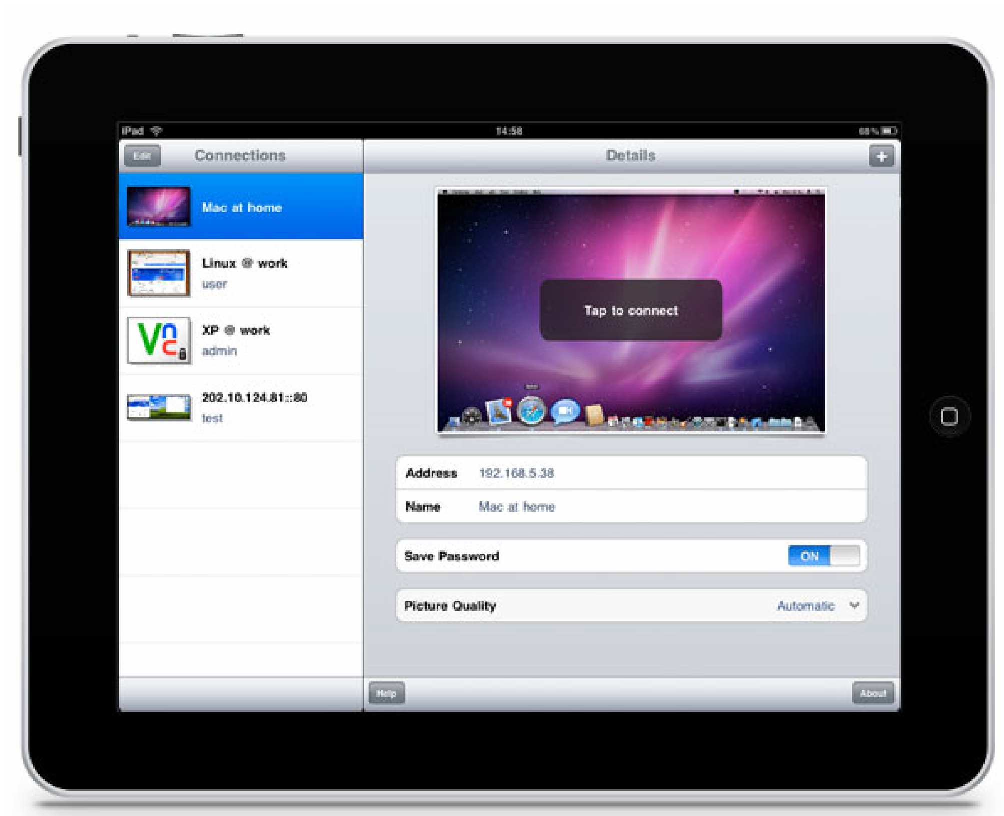
Performance & RAM

- There isn't much RAM. Deal with it. How?

	symbian Symbian "Cleanup stack"	android Android "Memory over-commit"
How you allocate...	Be ready for each allocation to fail. Keep track of all objects at all times. If anything fails at any point, your program must remain in a consistent state. Or else.	Dude, don't worry. All your allocations will succeed.
Memory leaks?	Programs must not leak. Ever.	Don't be silly, of course they do.
What if the system runs out of RAM?	An allocation will fail. The perfect program will gracefully degrade and all will be well. No need to kill anything.	The system may suddenly kill a process. Not at allocation-time, mind you: pages of memory are only actually allocated when they are first *used.* So some process will get killed at some arbitrary point when somebody first uses a variable.
Implication?	You-the-programmer must be perfect (unrealistic!) Code bloat to track each allocation (~10 lines of code for each)	You can't build a perfectly predictable system, so don't pretend. See what happens, it's probably fine most of the time.

User interface

- You need a new one. Porting desktop user interfaces just won't work.
- Completely different user interface APIs on each platform
- Really high expectations from the users on some platforms – especially iPhone – months of work required for iPhone viewer user interface.



Address 192.168.5.38
Name Mac at home
Save Password ON
Picture Quality Automatic

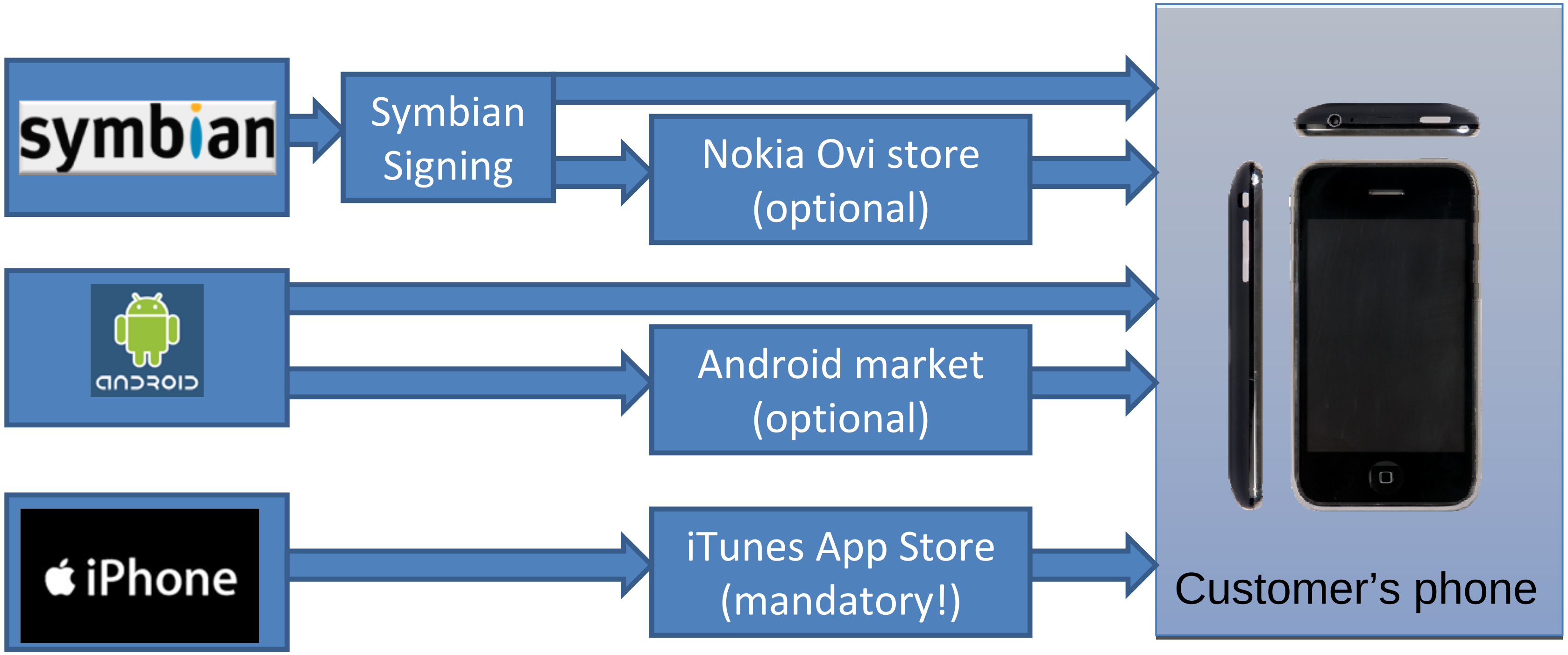
Challenge #3: testing

- There are simply lots, and lots, and lots, of phones. And they're all annoyingly different.

- We test each release on 50+ models
- Each one takes several days

Challenge #4: distribution

- Getting your application into the user's hands.
- Varies across the different platforms. Examples:


symbian
Symbian Signing
Nokia Ovi store (optional)
android
Android market (optional)
iPhone
iTunes App Store (mandatory!)
Customer's phone

Who decides what applications are OK?

- Who buys phones from the phone manufacturers?
  - End-users? Wrong!
  - Majority bought by network operators, then sold to end-users
  - Network operators specify how the phones should behave
- How does this affect mobile phone apps?
  - Is British Gas exciting? (Market capitalisation £16b)
  - Is Vodafone exciting? (Market capitalisation £78b)
  - Network operators don’t like to think of themselves as mere utilities, but instead as exciting content-delivery providers, to keep their shares price up
  - So they *really* care about apps: they ultimately wield control

Distribution: Symbian

- Still the leading mobile operating system in Europe
- All applications must go through “Symbian Signed” procedure
  - All Symbian APIs divided into sets – e.g. for network communication, GPS tracking etc.
  - Specify what sets you need and why
  - Some sets of APIs only available with manufacturer/operator permission
  - You submit your application binaries to Symbian Signed
  - They check your application works, and that your justifications stack up, and cryptographically sign your binaries
  - Without that signature, your application can’t be installed on a phone.
- A problem for VNC servers:
  - We need to capture the screen to send to a VNC viewer
  - Operators want to show DRM locked-down video without anyone being able to capture it
  - We have workarounds... for now

Distribution: Android

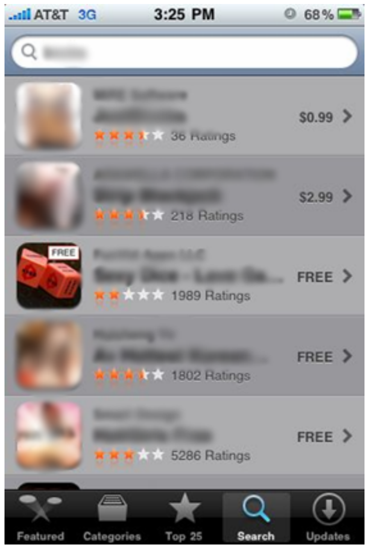
- Very open
- No signing/approval: you can publish your code however you like
- But for that reason there are no risky APIs: network operators wouldn't allow developers total unrestricted access
- Our problem: no way to grab a screenshot of an Android phone

Distribution: iPhone

- Apple had a big battle with the network operators and gained control. Go Apple!
- However, Apple are pretty opinionated themselves:
  - They can (and do) reject apps submitted to the app store
  - **Example 1**: two weeks ago they forbade anything written in cross-platform toolkits like Adobe Flash
  - **Example 2**: no porn. Oh, unless it's from big lucrative pornographers
  - **Example 3:** background apps....

Apple's iPad war on Adobe and Flash

The Observer, 18th March



iPhone Background Apps

- **iPhone OS 3**: no background apps.
- **iPhone OS 4**: allowed but only if you fit into one of Apple's seven approved reasons to multi-task:
  - Music
  - Voice-over-IP
  - Location
  - Finishing off downloads
  - Notifying of e-mails etc.
  - Err, that's about it.
- RealVNC may be able to offer a server for the iPhone?
- RealVNC iPhone viewers lose their connections when they go to the background

Challenge #5: politics!

- Creating a VNC server for a mobile phone is really tough: We need to grab the screen and we need to control the phone user interface.
- These APIs are on all desktop platforms but not reliably available on *any* mobile platform.
- We rely on contacts and customers.
  - Network operators want to sell to enterprises. Enterprises want to manage their devices from central helpdesks. So we sell to helpdesk operators, and they help us get access to the device APIs.
  - We're going to be producing some cool exciting apps to persuade phone vendors that all this is worth doing. Watch this space!

Politics, how to play the game…

- Android: submit lots of patches into open-source project
- BlackBerry: pay lots of money to be part of their Alliance programmes
- S60: hire lots of ex-Nokia engineers
- Windows Phone 7: no good plan yet. Their decision not to allow multi-tasking seems crazy and we hope they realise that!

VNC Mobile Viewers

- But there's good news!
  - Simple applications like a VNC Viewer are possible on any of these platforms.
  - So long as you don't need multitasking or special APIs, it's quite easy to produce and distribute such things

Summary

- If you plan to produce mobile applications, don’t underestimate the time & effort needed to get them tested and published
- Don’t rely on multitasking.
- Don’t rely on having any specific OS APIs.
- And... You *could* make money

Android Market
Developer keeps 70%
Development easy
Fewer customers?

iPhone App Store
Developer keeps 70%
Development harder
More customers

Summer internships available!

Graduate jobs available!

careers@realvnc.com