What on Earth is... Android?

The not-at-all paranoid **Graham Morrison** puts Google's Linux-based mobile phone platform though his personal Voight-Kampff machine.

When someone mentions Android, why do I feel as though they're all out to get me?

You must have read *Hitchhiker's Guide to the Galaxy* at a formative age. That heady mixture of hormones and puberty means that the word 'android' will forever be associated with anxiety and depression. Many of us feel the same.

However, the Android we're talking about should fill you with joy. It's a new operating system for mobile phones, designed by the good people at Google. And it uses Linux. This means that unlike nearly every other mobile phone operating system we've endured for 10 years, we finally get to enjoy a little freedom on our mobile devices.

So the rumours were true. Google is working on the mystical gPhone?

Actually, no. Those rumours stemmed from leaked details about Android. When Android was officially announced in November last year, Andy Rubin, Google's director of mobile platforms, made it very clear that Google had no intention of releasing a gPhone. "Android is more significant and ambitious than a single phone," he said, "We hope Android will be the foundation for many new phones and will create an entirely new mobile experience for users, with new applications and new capabilities we can't imagine today."

> **"Unlike Apple, Google is making Android's API a level playing field."**

That doesn't mean to say that a gPhone won't appear in the future, but with several mobile phone manufacturers involved in Android's development, it looks unlikely. HTC, the Taiwanese Windows Mobile phone manufacturer, has promised an Android-based phone before the end of the year – Google won't want to dampen similar initiatives.

Why would mobile phone manufacturers dump their existing OS for a new one?

The idea of an open platform is obviously appealing, but early prototypes of Android show that the user interface looks rather good (some might say a little like the iPhone). Thanks to Android's use of OpenGL, windows smoothly slide into and out of view, and scrolling around the display is smooth and seamless. This is a different experience from the one that most of us have with mobile phones. There's no doubt that some manufacturers are looking for something to compete with the iPhone interface, and Google might have just come along at the right time to make it happen – but it's still pushing 'open' as Android's main feature.

Didn't Google acquire a company called Android not so long ago?

Yes they did. That's where Andy Rubin worked before the takeover. He was the co-founder of Android, along with Nick Sears. They didn't like the time and effort it took to develop mobile phone software, and reasoned that the missing link was an open platform. "Open is good because it's about choice. And it allows people to do what they want," is the mantra according to Steve Horowitz, Android's engineering director. And freedom seems to be what Android is about, by creating a software stack that gives control back to the customer, and the companies involved in mobile phone technology, from the chip manufacturers to the telecoms providers.

Does that mean Android is only going to be of use to developers?

Initially, yes. At least until we can get our hands on a working device. Until then, the only people who can get something from Google's hyperbole are developers. They can download an early version of the API and start developing applications for the Android platform right now. Many have done just that, and the general consensus is that Google has done A Good Thing.

But Google isn't working alone. It has the support of the Open Handset Alliance, a group with a single ambition – open standards for mobile devices. HTC is one such member, but there are over 33 others, including Intel, Motorola, Samsung, T-Mobile and even Nvidia.

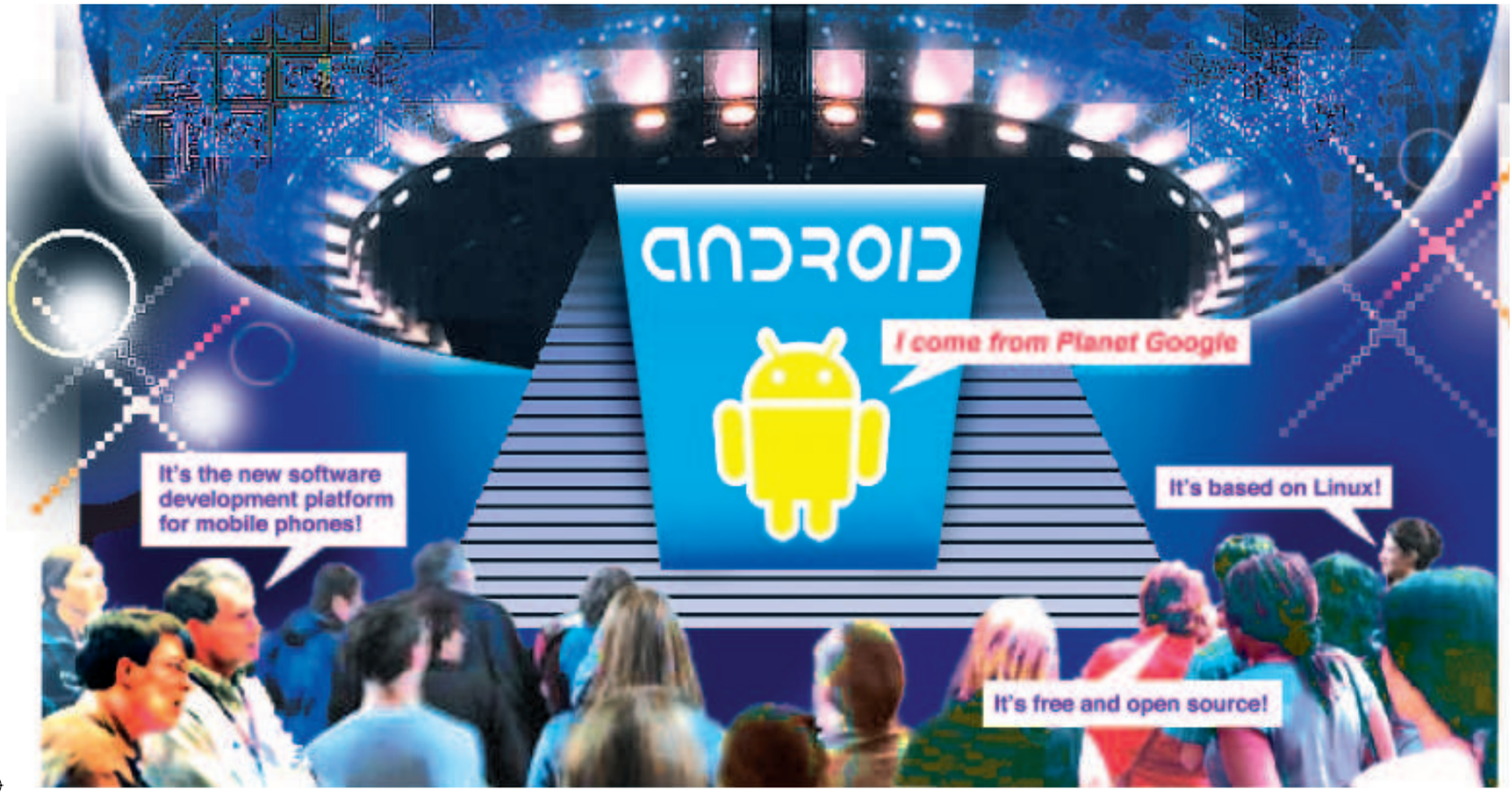
Will the alliance keep Google on the path of righteousness ?

It's difficult to say, because the formation of the alliance was instigated by Google and announced at the same time as Android. But with so many important partners on board, it looks like the ideal of creating an open standard for mobile phones might actually work.

The only slight sticking point is licensing. Google has promised to use an open source licence for 'most' of its software stack, and the licence it has chosen is *Apache v2*. This is commonly used by companies that want to keep some parts of their work proprietary, but it's too early to tell what Google will want to keep for itself. So far Google has released only the kernel it's using, which it had to do because the kernel is licensed under the GPL.

OK, so there's the Linux kernel; what other software does Android have?

It includes everything a mobile phone needs to offer basic functionality. At the bottom of the stack is a Linux 2.6 kernel, doing exactly the same kind of things that the kernel does so well on our PCs. On top of the kernel is something known as Android Runtime. This is the paranoid part that safeguards the kernel from the rest of the stack, protecting it against anything from badly written code to malicious viruses. Above the Android Runtime layer are the C/C++ libraries that add all the neat



functionality that Google hopes will make Android such a great development platform. These libraries include media playback of MPEG4, H.264, MP3, AAC, AMR, JPG and PNG files; 2D and 3D transformations used by the GUI; and the *SQLite* database engine and integrated *WebKit* page rendering. Developers can access all these features using the application framework.

What's to stop people developing programs to break Android and your mobile phone?
That's the job of Android Runtime. It protects the kernel by running third-party software inside a virtual machine, in the same way that the Java Virtual Machine protects the operating system from wayward Java applications. Android's VM is designed to be compatible with Java Class files, and after a compile-time conversion process, can run Java Class files directly. The core Java libraries are used by Android to provide essential functionality. This means that Android is Java-based, supporting a large subset of Java Standard Edition 5. This also means that Java will be the principle development environment for Android. And unlike Apple, which created one version of the iPhone API for internal development and another for third-party developers, Google is making Android's API a level playing field, so Google's own applications won't have secret access to Android's hidden features.

Isn't Java going to put a lot of people off?
Because of its perceived sluggishness, Java has long been a source of contention among developers. But it's also a tried and tested technology. By creating its own VM, Google is trying to improve Java's performance on its own platform. The MSM chipset specifically supported by Android includes hardware Java acceleration, for instance, so speed shouldn't be an issue.

Using Java also sidesteps the difficulty of hardware compatibility. On the Symbian OS, for example, programs need to be compiled specifically for the platform running the application. You can't simply download a Symbian 3rd Edition application and expect it to run. You need to have the same version of Symbian, and in most cases, exactly the right phone. Running software in a virtual machine obviates the need to worry about the real hardware – the virtual machine does that for you. This should mean that an Android application will run on any Android phone, and the specification states that the hardware should provide basic common functionally to help this to happen.

Does that mean they'll be something to play with when the first phone is released?
The API is easy enough to use, and you don't even need a phone to develop for Android. The developer's kit includes an Android emulator – a fully functional phone on your screen, complete with clickable buttons – and developers can use this to see how their applications look, as well as debugging their behaviour. Prototype phones show music players, Google Mail integration and a web browser, along with many other applications. Google has also sponsored the Android Developer Challenge to encourage development. Developers had until the middle of April to submit ideas for applications, with 50 of the most promising entries being awarded $25,000 to fund further development. The best of those could then go on to be awarded up to $275,000 in additional funding.

The list of possible areas for funding give us a good idea of how Google sees Android. It wants to see social networking, news, media consumption, gaming, mash-ups, location based services and apps with humanitarian benefits.

How is this likely to affect rival mobile phone projects, like OpenMoko?
The OpenMoko project has very similar aims, but doesn't have anything like the same resources. But there is another difference between the two. OpenMoko is committed to releasing the entire project using GPL/LGPL, whereas Google's position isn't clear. This is likely to lead to a split, with many geeks and open source advocates plumping for OpenMoko, while those who simply want a smooth-running Linux mobile with a sleek interface are likely to go for Android. Hopefully, both projects will be able to coexist.

Further information?
Check out **www.openhandsetalliance.com** and **http://code.google.com/android/index.html** for the latest on the Open Handset Alliance and Android. LXF