

LiPS's planned deliverables include a set of base specifications (telephony, messaging, presence, PIM, graphics library, multimedia, reference hardware requirements, security and application framework), software development kits and a set of usage profile specifications for Linux devices.

In June 2007, LiPS announced the publication of its specifications for the address book, voice call enabler and user interface services based on GTK. The LiPS Forum plans to finalise the remaining elements of its Release 1.0 by adding specifications relating to telephony, messaging, calendar, IM, presence and additional UI services' API specifications.

Figure 4.15: Specification release schedule for the LiPS Forum

H1 2007	H2 2007	2008
LiPS 1.0 Part1	LiPS 1.0 Part2	
<ul style="list-style-type: none"> • Address Book • Voice Call Enabler • UI Services • Widget setsKey • Navigation 	<ul style="list-style-type: none"> • Telephony services • Messaging services • Calendar • IM & Presence services • UI Services APIs 	<ul style="list-style-type: none"> • Additional Application FW • Services APIs (e.g. IMS, SIP apps) • Device Management APIs (based on OMA DM) • Additional Enabler APIs (e.g. multimedia apps)

Source: Informa Telecoms & Media, LiPS Forum

In early 2007, the LiPS Forum enlisted the help of a newly-launched project, the GPE Phone Edition (GPE2), producing a community-led implementation of a mobile Linux OS; GPE2 aims to conform to future LiPS specifications.

LiPS has a direct route to market through Orange/France Telecom, which has publicly committed to supporting ACCESS's ALP platform. Indeed, Orange is strongly committed to contributing to LiPS in terms of day-to-day management and technical work. Informa Telecoms & Media believes that Orange's involvement alone will help bring LiPS-compliant devices to market from ODMs and perhaps some OEM models, too. However, due to the early stage of the LiPS specifications, commercial products are not expected before the second half of 2008.

GNOME Mobile and Embedded (GMAE) Initiative

Formed in late 2006 and formally announced in April 2007, GNOME Mobile and Embedded (GMAE) is an initiative aiming to bridge community- and commercially-led software development around GNOME Linux components. GMAE is backed by primarily Nokia (Maemo) and ACCESS (ALP), both of which use the GTK+ graphics library and related components. The founding organisations also include Imendio, Intel, OLPC, OpenedHand and Red Hat, plus GMAE contributors CodeThink, Collabora, FIC, Fluendo, Igalia, Kernel Concepts, Movial, Nomovok, Openismus, Vernier, Waugh Partners and Wolfson

Sun Java: what open source means for Java ME

Java refers to three distinct concepts:

- Java is a software environment for developing and deploying third party applications on mobile, desktop and enterprise environments.
- Java is a programming language for developing such applications.
- Java is a trademark owned by Sun Microsystems, which controls the terms of use of the trademark and manages the Java Community Process that oversees the evolution of Java platforms.

For the past six years, Java has been nurtured in the mobile industry as a software platform that allows applications to be written once, but downloaded and run on any mobile phone. Reality has been much more complicated than theory, however, with Java for handsets being fondly likened to ‘write once, debug anywhere’.

On one hand Java is indeed ubiquitous; just over 50% of handset sales are forecast to be Java-enabled in 2007. According to Tira Wireless and Sun Microsystems at the 2006 JavaOne Conference, there are now over 500 different handset models in the market using Java and over 20 new Java-enabled handset models are brought to the market each month.

On the other hand, there are over 500 variants of Java ME (Java Platform Mobile Edition) platform implementations (including regional variations) that application providers have to cater for. To address 80% of the market of Java devices, developers typically have to port their Java applications to over 200 phones, each of which is a variant of the Java ME platform. This well-known phenomenon of Java fragmentation is down to a number of factors, such as the differences in handsets (screen colour, screen resolution, key layout, etc), the differences in Java ME APIs embedded with each device and the differences in interpretation of the Java ME specifications.

In practice, Java is widely used today by application developers as a development and deployment platform, but at a considerable cost of application porting, testing and stock unit management. In Java’s defence, it is still substantially less complex and less expensive to build and deploy applications in Java compared with the native OS. In addition, fragmentation is expected to decrease with the adoption of the Mobile Services Architecture (MSA), also known as JSR248, which bundles several other APIs and introduces clarifications to bridge variances in Java ME implementations. The first device to support MSA, the Sony Ericsson Z750, was announced in March 2007.

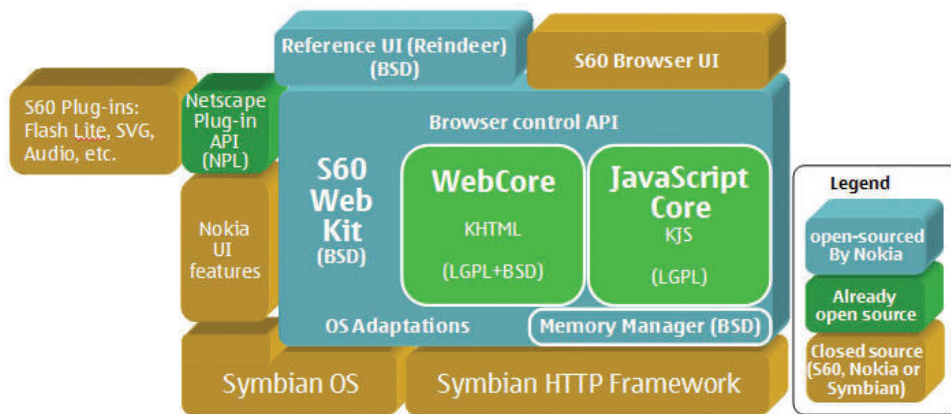
out to be too resource-intensive for mobile devices. Nokia's browser team decide to adopt the WebCore and JavaScriptCore components based on Apple's Safari browser Web Kit; these were chosen because of their small RAM/ROM requirements, an easy learning curve for developers, a fast start-up time and good overall performance of the browser.

To date, the Web Kit source code has been used by over 50 diverse applications, including lightweight or 'kids-friendly' web browsers, Nokia's OSB browser, Apple's Safari, S60 Web Kit, instant messaging applications, news browsers and help viewers. Nokia has also developed a port of Apple's Web Kit for the widely used GTK+ graphic libraries running on top of Linux.

Browser architecture

As the engine behind the S60 web browser, S60 Web Kit was developed from the foundations of Apple's Safari browser Web Kit – specifically the WebCore and JavaScriptCore components. These components have in turn been based on KHTML and KJS components from KDE's Konqueror open source project. Nokia claims that the use of open source has enabled the browser team to reuse a proven desktop rendering engine that has been developed and optimised by a large open source community over many years.

Figure 5.4: Nokia S60 web browser architecture



Source: Nokia

The S60 Web Kit includes several feature additions on top of the WebCore and JavaScriptCore components of Apple's Web Kit, such as a memory manager for memory-constrained devices, a mouse pointer for a desktop-like navigation experience, layout scaling, text search capabilities and support for dynamic HTML. Nokia's open source code includes a reference UI codenamed Reindeer, but does not include its own S60 UI implementation.