Symbian OS Version 9.1

Product description

Sander Siezen, Product Manager, Symbian Ltd
Revision 1.1, February 2005

Summary

Symbian OS v9 enables Symbian OS licensees to accelerate and lower the cost of their development of smaller, less expensive and even more capable Symbian OS smartphones. Phones based on Symbian OS v9 will provide network operators with a robust and secure, open standards-based platform for the cost-effective deployment of mass market, revenue-generating services, content and applications on 2.5G and 3G networks around the world.

This paper gives a detailed overview of features and functionality available in Symbian OS Version 9.1.

1 Symbian OS Overview

Symbian OS is the operating system licensed by the world's leading mobile phone manufacturers. Symbian OS is designed for the specific requirements of open, data-enabled 2G, 2.5G and 3G mobile phones.

Symbian OS is characterized by:

- **integrated multimode mobile telephony** – Symbian OS integrates the power of computing with mobile telephony, bringing advanced data services to the mass market
- **open application environment** – Symbian OS enables mobile phones to be a platform for deployment of applications and services (programs and content) developed in a wide range of languages and content formats
- **open standards and interoperability** – with a flexible and modular implementation, Symbian OS provides a core set of application programming interfaces (APIs) and technologies that is shared by all Symbian OS phones. Key industry standards are supported
- **multi-tasking** – Symbian OS is based on a micro kernel architecture and implements full multi-tasking and threading. System services such as telephony, networking middleware and application engines all run in their own processes
- **fully Object-oriented and component based** – the operating system has been designed from the ground up with mobile devices in mind, using advanced OO techniques, leading to a flexible component based architecture
- **flexible user interface design** – by enabling flexible graphical user interface design on Symbian OS, Symbian is fostering innovation and is able to offer choice to manufacturers, carriers, enterprises and end-users. Using the same core operating system in different designs also eases application porting for third party developers
- **robustness** – Symbian OS maintains instant access to user data. It ensures the integrity of data, even in the presence of unreliable communication and shortage of resources such as memory, storage and power.

1.1 Key Features of Symbian OS v9.1

- **Rich suite of application services** – the suite includes services for contacts, scheduling, and messaging, OBEX for exchanging appointments (vCalendar) and business cards (vCard); integrated APIs for data management, text, clipboard and graphics
- **Java support** – supports the latest wireless Java standards, including MIDP 2.0, CLDC 1.1, JTWI (JSR185), Mobile Media API (JSR135), Java API for Bluetooth (JSR082), Wireless Messaging (JSR120), Mobile 3D Graphics API (JSR184) and Personal Information Management and FileCF APIs (JSR075)
- **Device Management/OTA provisioning** – OMA DM 1.1.2 compliant, OMA Client provisioning v1.1
- **Messaging** – enhanced messaging (EMS) and SMS; internet mail using POP3, IMAP4, SMTP and MHTML; attachments
- **Multimedia** – audio and video support for recording, playback and streaming; image conversion
- **Graphics** – direct access to screen and keyboard for high performance; graphics accelerator API; increased UI flexibility (support for multiple simultaneous display, multiple display sizes and multiple display orientation)

- **Communications protocols**– wide area networking stacks including TCP/IP (dual mode IPv4/v6) and WAP 2.0 (Connectionless WSP and WAP Push), personal area networking support including infrared (IrDA), Bluetooth and USB; support is also provided for multihoming and link layer Quality-of-Service (QoS) on GPRS and UMTS networks
- **Mobile telephony** – Symbian OS v9.1 is ready for the 3G market with support for WCDMA (3GPP R4); GSM circuit switched voice and data (CSD and EDGE CSD) and packet-based data (GPRS and EDGE GPRS); CDMA circuit switched voice, data and packet-based data (IS-95 and 1xRTT); SIM, RUIM, UICC Toolkit; other standards can be implemented by licensees through extensible APIs of the telephony subsystem
- **Platform security** – proactive system defence mechanism based on granting and monitoring application capabilities. Infrastructure to allow applications to have private protected data stores. In addition, full encryption and certificate management, secure protocols (HTTPS, SSL and TLS) and WIM framework
- **Realtime** – a realtime, multithreaded kernel provides the basis for a robust, power-efficient and responsive phone
- **Hardware support** – supports latest CPU architectures, peripherals and internal and external memory types
- **CDMA specific features** including CDMA network roaming, third party OTA API, NAM programming mode, CDMA SMS stack, NAI handset identification, interfaces to enable Mobile IP and bridge and router gateway modes of operation
- **International support** – supports the Unicode Standard version 3.0
- **Data synchronization** – Over-The-Air (OTA) synchronization support using OMA standards; PC-based synchronization over serial, Bluetooth, infrared and USB; a PC Connectivity framework providing the ability to transfer files and synchronize PIM data
- **Developing for Symbian OS** – content development options include: C++, Java (J2ME) MIDP 2.0, and WAP; tools are available for building C++ and Java applications; reference telephony abstraction layer for 2G, 2.5G and 3G provided.

1.2 Key new features of Symbian OS v9.1

RTP – Symbian OS v9.1 provides a native RTP (Realtime Transfer Protocol) stack. This stack can be used by licensee and 3rd party applications without the need for a separate RTP stack.

Device Management – Symbian OS provides features which give network operators and enterprises new capabilities to manage phones in the field. This includes OMA Device Management 1.1.2 support and OMA Client provisioning 1.1.

Bluetooth – Symbian OS v9.1 continues to add Bluetooth innovations to the operating system. In this release support for Bluetooth eSCO and Bluetooth Stereo headset profiles are implemented.

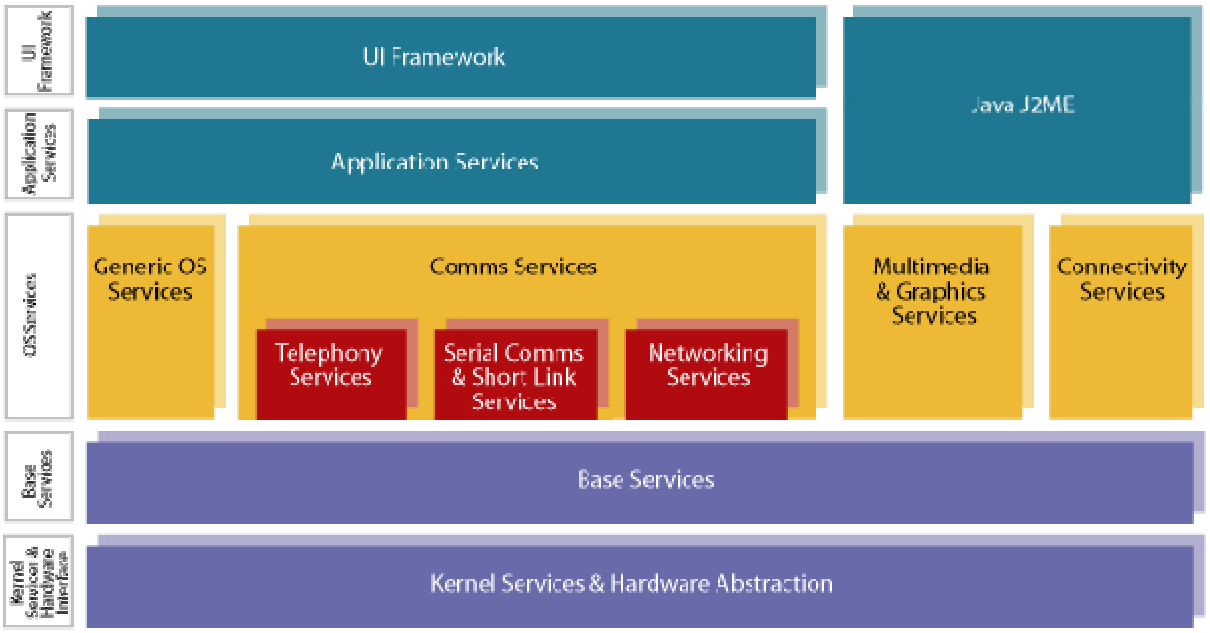
EABI tooling – Symbian OS v9.1 is built using the ARM RVCT 2.1 compiler. This compiler is compliant with the ARM EABI standard. This allows compatibility with the latest ARM compliers and reduces the Symbian OS footprint while enhancing performance.

Platform Security – Symbian OS v9.1 provides a proactive defence mechanism against malware. The platform security infrastructure uses a capability based model which ensures that sensitive operations (for example modifying user data, making calls, using network connections) can only be accessed by applications which have been certified by an appropriate signing authority.

Data Caging – this feature allows applications to have their own private data partition. This allows for applications to guarantee a secure data store. This can be used for e-commerce, location applications and others.

1.3 Symbian OS Architecture Overview

Pictured below is an abstract rendering of the software architecture of Symbian OS. The rest of this product description will follow the structure of the diagram below, starting with UI framework and moving downwards in the diagram.



2 Symbian OS Product Description

2.1 UI Framework

Symbian OS is designed to support a variety of user interfaces, enabling differentiated product designs. The application framework subsystem provides a powerful environment for licensees and partners to create differentiated user interfaces, while retaining maximum compatibility for application developers.

2.1.1 UI Application Framework

A principal objective of the graphical user interface (GUI) framework is to define as little policy as possible and therefore minimize the constraints placed on a product's UI designer. This eases the porting of the user interface of applications between different Symbian OS phones.

Main features of the GUI framework:

- an event-driven GUI and widget architecture
- a windowing system for sharing screen, keyboard and pointer between applications; clocks and animated bitmaps, and a control framework for sharing an application window between application components
- a direct navigation link (DNL) system enables close task-based integration between applications
- a mechanism for the licensee to customize the look and feel (LAF) of the GUI
- a plug-in mechanism for the user to input non-standard data (e.g., for ideogram input or voice recognition for phones that may not have a keyboard)
- control factory structure allows the framework to be extended by multiple applications
- a notifier framework allowing system events and alarms to be handled more flexibly by the GUI
- a flexible screen indicator and status bar framework
- runtime changeable color schemes
- bitmap animation performed in the window server thread
- semi-transparent windows
- support for multiple displays
- application and window scaling.

Other application framework features and utilities:

- rich text rendering for various locales, providing a text model with character and paragraph formatting, embedded graphics, and a text view which supports efficient formatting, display and interaction
- two multi-level undo/redo capabilities: a plain text undo system that can undo text insertion, deletion and clipboard operations, and a rich text undo system that can cope with anything a rich text object can do, including embedded objects
- generic support for plug-in parsers that recognize certain strings, eg URLs, email addresses, phone numbers. This enables to run services or applications from any application in the system
- support for PC-style changeable color schemes in editable text and for auto-sizing text editors

- background images: arbitrary graphics can be drawn behind text, with control of parameters like transparency and background scrolling
- XML Framework with support for XML and WBXML parsing.

2.1.2 Internationalization support

The main internationalization features are:

- conforms to the Unicode Standard version 3.0
- supports rendering and editing of all European languages, simplified Chinese, traditional Chinese, Japanese, Korean, Arabic, Hebrew and Thai
- a front-end processor (FEP) framework for text input using handwriting recognition or keyboard to enable input of ideographic characters. FEPs can take the form of a floating window, or a “transparent window”, or be invisible. They can also interact closely with their target controls, e.g. to do “inline editing”
- conversion between Unicode and other character sets, via a plug-in mechanism.

2.2 Application Services

Symbian OS provides the application engines for the core mobile phone applications; contacts, calendar, messaging and device synchronization.

This provides Symbian licensees with core application functionality and ensures compatibility between Symbian devices.

2.2.1 PIM services

The core PIM services include agenda (schedule), to-do and contacts.

Main features are:

- Agenda engine: provides client-server shared access; vCalendar support with send and receive functionality; provides DST and UTC support; synchronization with PIM applications using Symbian Connectivity Services, also includes an API for group scheduling functionality for popular PC PIM applications
- To-do engine: integrated with the agenda engine for storing agenda type entries
- Contacts model: provides a generic client-server shared access contacts database engine, integrates with messaging application for emails, faxes and SMSs, caller number matching, receive a vCard (supports Versit vCard 2.1 standard), group support, support for multiple templates, support for extensible field types, current item support, connectivity requirements, searching and filtering by contact item type.

2.2.2 Messaging services

The messaging framework provides messaging protocols which support sending and receiving of text messages (SMS), enhanced messages (EMS), and email. The framework uses polymorphic Message Type Modules (MTMs) to handle specific types of message. MTMs can be added at run-time to expand the messaging capabilities of Symbian OS phones after market.

A “Send-As” API enables the creation of messages (email, SMS, EMS) straight from any other application. For example, messages containing application-specific data such as vCards can be created directly from the contacts applications.

Main features of Messaging are:

Bearer Independent Object messaging
Bearer Independent Object (BIO) messaging allows application communication of arbitrary data types between devices. BIO messaging uses an SMS watcher framework to support messages sent over-the-air to the operating system or application rather than to the end-user.

BIO message types currently supported include compact business card, vCard, vCalendar, email notification, operator logo, ring tone, and settings for internet access, mail settings, MMS and WAP. The framework is open, allowing third parties to add further capabilities and value at run-time.

The vCard and vCalendar BIO message file types are also supported over infrared and Bluetooth links.

Short Message Service (SMS)
SMS support consists of an SMS stack with a messaging API to send and receive SMS and provides the following features:

- the SMS stack is implemented as a plug-in protocol. The GSM (03.40) SMS protocol is provided
- the GSM SMS stack can be used as a bearer for the WAP protocol module
- transmission and reception of GPRS SMS
- SMS: send and receive streamed SMS messages. Enumerate, read, write and delete access to the SMS storage areas of the phone and SIM. Receive messages that match a specified text
- 7-bit SMS alphabet, 8-bit SMS alphabet and UCS2 data coding schemes are supported
- supports sending and receiving concatenated SMS messages
- scheduled sending: on a specific date/time, “now” or upon request. Specify and review scheduled actions.

Enhanced Messaging Service (EMS)
EMS support in Symbian OS is compliant with 3GPP release 4 (TS 23.040) and supports the following features:

- mobile originated pictures: variable picture, pre-defined picture: 16 x 16, pre-defined picture: 32 x 32
- mobile terminated pictures: variable pictures (1024x1 to 8x128), small pictures 16 x 16 and large pictures 32 x 32
- animations: pre-defined animations (multiple separate animations), black and white animations and mobile terminated user-defined animation
- sounds: iMelody
- formatting: both mobile originated and terminated formatting, text size (small, medium, large), text style (bold, underline, strikethrough, italic) and message alignment (left, center, right)
- Object Distribution Indicator (ODI): as specified in 3GPP release 5 (TS 23.040).

Email
Email has the following main features:

- Internet mail: supports disconnected mode, cache management, single operation get-new-mail for both POP3 and IMAP4, SMTP client

enhancements (copy-to-self, separate emails for Bcc: send email for specific SMTP servers via specific ISP accounts, multiple SMTP connections, e.g. for home and work accounts, with simultaneous connections to separate accounts), UUE and MIME encoding, MHTML, automatic receipt notification, automatic MIME character set conversion, automatic email signature (or vCard). Character set conversion takes place during sending or receiving messages

- email filtering is enabled over IMAP and POP3 including synchronization filtering by number, size and age and download filtering by size
- Internet access points (IAP): connection over GPRS is supported as well as over GSM CSD. Multiple IAPs, both GPRS and GSM CSD, can be associated with each email MTM to specify whether the default preferred connection or a specific connection should be used
- secure socket connections: facility to establish a TLS socket connection to email servers using secure ports or standard ports via STARTTLS.

2.2.3 Content Handling services

The Content Handling Framework (CHF) provides a generic framework that allows various content handlers to be included at run time, providing services across the platform. An example of this would be an HTML plug-in that could offer its services to e.g. Email.
Key benefits of CHF include potential savings in code and memory, as well as speed and ease of development.

2.2.4 Internet and Web Application Support

HTTP Transport Framework

Symbian OS v9.1 provides a generalized framework for Internet applications to use HTTP like protocols. The framework presents a unified, high level API that is independent of particular header representations, specific protocol details or the underlying transport layer. This framework is used as the interface to HTTP and WSP protocols.

The framework allows for extension and customization at an application or platform level by the use of filter plug-ins.

An HTTP 1.0 and 1.1 Client stack supports persistent connections, pipelining, and chunked transfer encoding. Filters provide support HTTP Redirection and HTTP Basic and Digest Authentication. It enables applications such as SyncML, OCSP, Web Based Application Installation. This stack can also be used by third parties for applications such as web browsing.

WAP Stack

The WAP stack subsystem includes support for a subset of WAP 1.2.1 (WAP June 2000), push functionality and GPRS as a bearer. The WAP stack supports protocol specifications version 1.1 and 1.2.1 in connection-less mode only. The WAP stack supports the following bearers: GSM CSD, GPRS UDP, GSM SMS and GPRS SMS for connectionless push. The WAP stack, with interfaces to each of those layers of the protocol stack, has the following layers:

- CL WSP, connection-less session protocol for WAP
- WDP, datagram protocol for WAP, client and server
- a WAP push watcher which listens for secure and non-secure push messages received using connectionless mode over all supported bearers (both GSM and GPRS are supported). Incoming push messages are identified by an application-id in the message header and handled by plug-in DLLs (a handler for the WML

User Agent is provided). The WML User Agent handler identifies the content of the message by the MIME type specified in the header.

2.2.5 Data synchronization services (OMA)

The OMA (SyncML) data synchronization client is compliant to the OMA specifications. The data synchronization client has the following features:

- OMA Data Sync 1.1.2 compliant
- one way, two way and refresh sync with server and client initiation
- supports HTTP, WSP and OBEX over infrared, Bluetooth and USB
- contacts and calendar synchronization
- database Adapter (DBA): provides a generic interface which de-couples the SyncML data synchronization engine from the data being synchronized. The engine can extract and exchange data with the database without having to know what the data is
- transport API (TPA): this provides a common generic API to the Sync Client Engine so that it is able to initiate connections, send and receive data etc. in a manner that is independent of the underlying transport protocol and transport media used. An HTTP TPA is provided
- a contacts DBA and an agenda DBA are provided to allow contacts and schedule synchronization respectively
- database and transport adapters are plug-ins so that additional ones can easily be written and installed.

2.2.6 Provisioning services (OMA)

Nokia Ericsson OTA Specification and Nokia Smart Messaging Specification allow the network operator to deliver settings to the device. This allows the network operator to securely manage the device via sessions which can read and write device settings. It allows for unambiguous identification of the device, including a unique device identifier, the manufacturer's name and the device model, the client revision version and the current language in use on the device.

The extensible framework allows for licensees and third parties to also have their own software managed remotely.

Symbian OS v9.1 provides the following device management and OTA (Over-The-Air) provisioning functionality:

- Nokia Ericsson OTA Specification v6
- Nokia Smart Messaging v2
- OMA Device Management v1.1.2
- Supports remote management of messaging, networking and browsing settings
- OMA Client Provisioning 1.1.

2.3 Java

Symbian OS provides a leading Java application execution environment optimized for mobile devices and mobile applications. This enables compatibility with wireless Java applications written for mobile devices and advanced Java applications that can make use of a Symbian device’s capabilities.

2.3.1 MIDP 2.0 and CLDC

The Symbian OS v9.1 release provides developers and operators with the functionality necessary to develop compelling games, services, and utilities. These can be client-server, peer to peer, or stand alone.

Symbian OS v9.1 implements J2ME MIDP 2.0 and CLDC 1.1, using Sun's CLDC HI 1.1 Java VM, and is JTWI (JSR185) compliant. The following additional JSRs are also provided:

- CLDC HI 1.1 and MIDP 2.0
- JTWI (JSR185)
- Java API for Bluetooth 1.0 (JSR082), excluding OBEX
- PIM and FileCF (JSR075)
- Wireless Messaging 1.1 (JSR120)
- Mobile media 1.1 (JSR 135)
- Mobile 3D graphics API for J2ME 1.0 (JSR184)

Tools include support for the Universal Emulator Interface (UEI), enabling integration with industry standard IDEs. This allows on target and emulator debugging (the emulator also uses the CLDC HI 1.1 Java VM, rather than KVM as in 8.0).

.

MIDP 2.0 implementation

J2ME CLDC/MIDP2.0 provides a compact Java environment for mobile phones, with a particular emphasis on games support. Symbian's implementation will run the numerous MIDlets being developed for mobile phones, while at the same time allowing MIDlets specifically purposed for Symbian OS to benefit from many features of a Symbian OS phone. Symbian's implementation is fast, has a tight footprint, and provides an environment that scales smoothly from mid tier to more capable smartphones. This is achieved by tight integration with the native platform and use of an optimized implementation of Sun's CLDC HI 1.1 VM.

Features:

- supports the OTA recommended practice document for MIDlet installation
- supports the JTWI untrusted security domain and enables a full RP security model to be implemented
- heap size, code size, and persistent storage are unconstrained, allowing larger, more compelling MIDlets to be run
- MIDlets look and behave very much as native applications, for example they use the native application installer and launcher, and native UI components
- supports native color depth (4096 colors)
- Generic Connection Framework implementation includes sockets and datagram support
- provides debugging support
- implements Bluetooth (excluding OBEX)
- implements wireless messaging.

2.4 OS Services

The heart of Symbian OS, this layer provides vital OS infrastructure components. These elements include multimedia and graphics subsystems, networking, telephony, shortlink protocols, cryptography libraries and PC connectivity infrastructure.

2.4.1 Platform security and Security Services

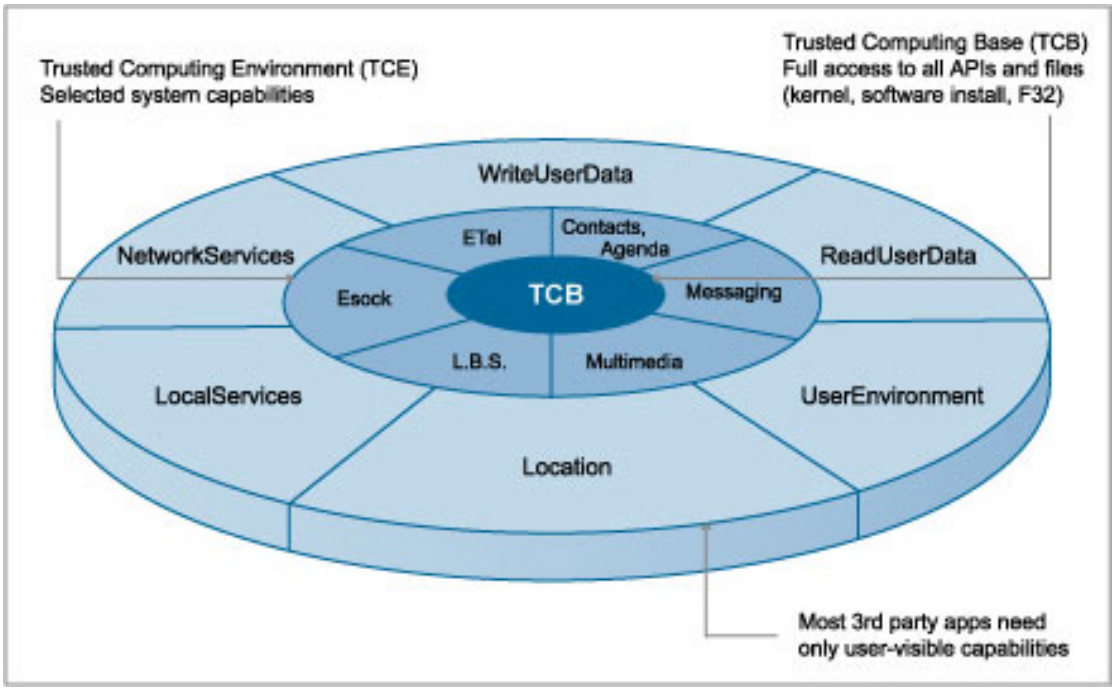
Platform security is extended in Symbian OS v9, providing control over the capabilities of applications installed on the devices. This is done to ensure the integrity of the phones and the network, while still enabling an open environment for third party applications.

The security subsystem also enables data confidentiality, integrity and authentication by providing underlying support for secure communications protocols such as TLS/SSL and IPSec. It also supports the authentication of installable software using digital signatures.

Platform Security architecture

The kernel and file server, and the software installer, are part of the Trusted Computing Base (TCB) and have unrestricted access to the device resources. They are responsible for maintaining the *integrity* of the device, and applying the fundamental rules of platform security. The rest of the operating system must trust them to behave correctly, and their code has consequently been very strictly reviewed.

Beyond this core, other system components require access to some, but not all, sensitive system resources. For instance, the window server requires direct access to keyboard events, but not to the phone stack. The Trusted Computing Environment (TCE) is composed of these key Symbian OS components that protect the device's resources from *misuse*. In Symbian OS, server programs are used to control access to shared resources, so the TCE is made up of the important system servers.



Capability Management

A capability is an access token that corresponds to permission to access sensitive system resources. As implied by the description of the TCE, the security architecture provides a number of different capabilities, such as access to the phone stack or to the complete file system. To access a system resource, a client program must hold the appropriate capability. In Symbian OS, the 'unit' or base level of protection between any two entities is the process. Thus, under platform security, each process is assigned a set of capabilities. When a process

makes a request of another process, the servicing process is able to examine the capabilities of the requestors' process and determine whether the request should proceed.

Data Caging
Data caging allows applications on a Symbian OS device to have private data which is not accessible by other applications. Data caging is enforced using the Platform Security infrastructure.

Cryptography module
The cryptography module includes the following significant components:

- cryptography algorithms allowing data to be encrypted and decrypted and supporting symmetric ciphers: DES, 3DES, AES, RC2 and ARC4, and asymmetric ciphers: RSA, DSA and DH
- hash functions: MD5, SHA1 and HMAC
- pseudo-random number generator for generating cryptographic keys
- PKCS #7 cryptographic message syntax

Cryptographic token framework
The cryptographic token framework enables licensees to integrate support for removable hardware devices, such as WIM modules, in a flexible manner. It consists of two parts:

- a framework which enables application code to query the system for the availability of implementations of specific cryptographic interfaces and their attributes (e.g., whether they are implemented in hardware, whether they are removable, whether they implement their own access control mechanism)
- the definition of a set of certificate- and key-related interfaces. Licensees may supply their own implementations of any of the defined interfaces and these will be picked up by applications using the framework (so for example they may provide a WIM implementation which implements the certificate storage interface, and then certificates stored on the WIM will be visible in the certificate management application and available to the certificate validation module).

The following interfaces are defined:

- a read-only certificate storage interface, supporting the retrieval of X.509 certificates, and certificate authority (CA) and user certificates
- a certificate storage interface supporting the retrieval of X.509 format certificates, and CA and user certificates
- a read-only private key storage interface
- an interface to authentication objects (e.g., PINs)
- a standard interface for performing security-critical user interface operations (e.g., PIN entry)
- A key management API.

The following implementations of these interfaces are provided:

- an implementation of the writeable certificate storage interface providing access to certificates used by all other software (e.g., TLS and Software Install)
- an implementation of an application to manage stored certificates

- a reference implementation of the interface for performing security-critical user interface operations
- a software-only key store reference implementation (supports for the WAP certificate profile according to the WAP-278-WAPCert specification).

Certificate management module
The certificate management module is used for authentication of other entities (e.g. third-party developers, web servers) to the user of the phone, and for authentication of the user of the phone. It uses X.509 certificates according to the PKIX Certificate Profile (RFC 2459). This module provides the following services:

- storage and retrieval of certificates using the cryptographic token framework
- assignment of trust status to a certificate on an application-by-application basis
- certificate chain construction and validation
- verification of trust of a certificate
- certificate revocation checking using the Online Certificate Status Protocol (OCSP)
- application update notification for installed applications.

Digital rights management
Licensees and technology partners are able to provide an implementation of a DRM agent such that protected content is accessible by authorized content consumers without modification of those consuming applications. Content consumers comprise those system components responsible for processing the following file types: GIF, BMP, JPEG, PNG, JAR, SIS, MIDI, MMF, AMR, and WAV.

Software installation
The software installation system provides a secure and fast installation process. The installation tool supports:

- installation of C++ executables, including authentication of software components using digital signatures to provide a measure of confidence that applications being installed onto a Symbian OS phone are from a known reputable vendor
- installation of Java MIDP 2.0 MIDlets, including authentication of MIDlets using digital signatures
- installation of Java MIDP 1.0 MIDlets. The MIDP OTA recommended practice document is fully supported
- compression of install packages to reduce disk space and download times. The compression library is a generic shareable DLL which can be called by other applications
- different varieties of phones, allowing the installation package creator to ensure the correct software is installed onto an appropriate phone.

2.5 Communications services

The Communications infrastructure subsystem provides the key frameworks and system services for communications and networking.

The Comms Infrastructure subsystem provides the key frameworks and system services for communications and networking. This includes:

- a communications database manager which controls the system-wide communications configuration
- a socket server and client-side API which provides a framework for implementing various communications protocols through a socket interface. Plug-in protocols are dynamically loaded
- multi-homing support: multiple simultaneous OSI Level datalink interfaces, each with its own unique IP address. This is not limited to Primary Packet Data Protocol (PDP) contexts but enables multiple access technologies to be provided such as WLAN, Bluetooth Network Encapsulation Protocol and Ethernet along with multiple PDP contexts
- a network interface manager which provides a framework for connection to other computers or networks. The manager provides a mechanism for the client to monitor progress over e.g., a PPP connection
- a serial communications server provides a serial port (RS232C) abstraction to allow Symbian OS phones to function as a DCE and a DTE as required. Dynamically loadable plug-in communications modules are used to actually communicate with device drivers and other protocol stacks.

2.5.1 Telephony services

The Telephony subsystem provides a multimode API to its clients. The abstract cellular make it easier for handset manufacturers to port Symbian OS from one mobile phone standard to another.

The Telephony subsystem provides a multimode API to its clients. The abstraction layer for cellular networks include support for GSM, GPRS, EDGE, CDMA (IS-95), 3GPP2 cdma2000 1x RTT and 3GPP W-CDMA making it easier for handset manufacturers to port Symbian OS from one mobile phone standard to another. The multimode telephony abstraction is key in Symbian OS to providing integration with the rest of the operating system to enable creation of advanced data services.

Functionality common to all networks includes:

- phone and network information: retrieve signal and battery strengths, provide access to the network names detected by the phone, information about the current network, receive notifications when there are network registration changes and retrieve the phone identity information
- phonebook: read, write, search and delete access to the phonebook storage areas of the phone and SIM (GSM 11.11) or R-UIM (cdma2000 1xRTT) or UICC(W-CDMA).
- phonebook synchronizer: mechanism to synchronize phonebook entries stored on a SIM or R-UIM card to the contact database so that clients can access all contact data via the contacts model API
- ISV Telephony API: interface providing support for reading network and phone settings, and making and receiving voice and circuit-switched data calls.

GSM/EDGE telephony

Support for GSM, GPRS and EDGE conforms to the 3GPP GSM Phase 2+ (releases R97/98).

GSM

The GSM telephony framework provides an abstract telephony interface for GSM voice, data and fax, and for landline modems for data and fax as well as phone number resolution and SIM Application Toolkit. Main features are:

- voice calls: initiate, terminate and answer voice calls
- circuit-switched data calls: initiate, terminate and answer data calls including HSCSD. Pass the control of serial port to communication protocols to stream data
- the abstraction supports a wide variety of ETSI GSM phase 2+ functionality
- GSM phase 2+ SIM Application Toolkit, Class 3 (ETSI 11.14 R98), with the exception of class 'a' and class 'b'
- supplementary services supported include: Alternative Line Service (ALS), Alternating Call Services (between voice and data, and voice and fax), retrieve NITZ time information, call forwarding, call waiting, call barring, Called/Calling Party Identity Presentation (CLIP) and Restriction (CLIR), setting up Closed UserGroup (CUG) call, User-User signalling (UUS), conference call, charging information, message waiting identification, network service requests (USSD).

GPRS

The General Packet Radio Service (GPRS) framework provides an abstract telephony interface for GPRS class B functionality. GPRS Release 97/98, Release 99 (GPRS and UMTS) and Release 4 (UMTS) packet services as well as CDMA/CDMA2000 are the specifications implemented. With class B functionality, phones are able to make and receive GSM calls while simultaneously remaining registered with GPRS. If a Packet Data Protocol context is active, GPRS services are automatically suspended and resumed. The main features are:

- attachment and detachment from the GPRS network
- activation and deactivation of a Packet Data Protocol (PDP) context for data transfer
- ability to activate and deactivate the PDP context automatically with no explicit client intervention
- ability to automatically suspend a GPRS data connection when an incoming or outgoing GSM voice call is made, and to resume a suspended GPRS data connection on notification from the GPRS network
- information and notification service to the client software of network information such as GPRS capabilities, current GPRS network availability, change in the current state of a GPRS connection and general PDP contexts parameters.

EDGE

The Enhanced Data-rates for Global Evolution (EDGE) framework provides an abstract telephony interface for 3GPP GSM/EDGE. In addition to supporting the GSM and GPRS functionality described above, main features are:

- supports EDGE enhanced CSD (ECSD)
- supports EDGE enhanced GPRS (EGPRS).

CDMA

CDMA (IS-95)
The CDMA telephony framework provides an abstract telephony interface for CDMA (IS-95) voice, data (circuit- and packet-switched) and fax. Main features are:

- voice calls: initiate, terminate and answer voice calls
- circuit-switched data: support for service options: asynchronous data and fax for both rate Set 1 and rate Set 2
- CDMA network roaming support
- packet-switched data: support for service options: CDPD for both rate Set 1 and rate Set 2
- text messaging
- sending, receiving and storing SMS messages, acknowledge receipt of SMS messages, 7-bit SMS alphabet, 8-bit SMS alphabet and UCS2 format SMS
- SMS teleservices:
- wireless paging (WPT-95), wireless messaging (WMT-95), voice mail notification (VMN-95), broadcast SMS, service category programming (SCPT), wireless enhanced, messaging (WEMT), card application toolkit protocol (CATPT)
- operation in AMPS (Voice only) networks
- single stack Quick Net Connect (QNC): packet data service that runs on top of a circuit switched connection, typically at a rate of 14.4 Kbps. This service enables fast set-up of a direct connection to the Internet
- enable third party Over-The-Air (OTA) provisioning applications and transparent setting of IP based Over-The-Air (IOTA) settings
- NAM Programming support
- support for CDMA specifications for PPP and functionality enabling Mobile IP
- supplementary services supported include: set preferred band class operation (band class A only/preferred, or band class B only/preferred), set preferred network operation (digital only/preferred or analog only/preferred), call forwarding, call waiting, Called/Calling Party Identity Presentation (CLIP) and Restriction (CLIR), conference call, message waiting identification, network service requests, enable handset identification using NAI.

cdma2000 1xRTT
The 3GPP2 cdma2000 1xRTT telephony framework provides an abstract telephony interface for 3GPP2 cdma2000 1xRTT voice, data (circuit- and packet-switched) and fax. In addition to the functionality of CDMA (IS-95) described above, main features are:

- circuit-switched data: support for IS-95B services
- packet-switched data: support for IS-95B services plus service options 22-29, 33, 34 for high speed packet data
- Removable-User Identity Module (R-UIM): support access to R-UIM files such as phonebook entries and stored SMS messages
- phonebook synchronizer: mechanism to synchronize phonebook entries stored on a SIM or R-UIM card to the contact database so that clients can access all contact data via the contacts model API

W-CDMA 3GPP R99/R4/R5
The WCDMA telephony framework provides an abstract telephony interface for voice, data (circuit- and packet-switched) and fax.

Main features are:

- UICC (USIM Integrated Circuit Card): support access to UICC phonebook and UICC security. Supports the USIM phonebook entry fields: index up to two name tags per entry multiple phone numbers multiple email addresses user defined grouping hidden entry flag
- supports USIM Application functionality
- QoS framework and QoS API
- the Toolkit API provides support for the R99 and R4 USAT commands
- support for Multimedia calls
- WMA/SMS Adapter
- support for Multicall Service
- 3GPP R5 address discovery over DHCP
- 3GPP R5 SBLP
- 3GPP R5 Telephony support for IMS.

2.5.2 Shortlink services

These services enable point to point communications with other devices and peripherals. Shortlink services in Symbian OS are available through support of Bluetooth, serial, USB and infrared.

Bluetooth
Bluetooth support is provided with a core Bluetooth 1.2 protocol stack supporting L2CAP (flow and error control) and Adaptive Frequency Hopping. The Host Controller Interface (HCI) provides a hardware interface via a UART based reference implementation. The OBEX implementation provides an OBEX v1.2 client and server that operate over Bluetooth and IrDA. The OBEX server also provides USB support. Security is provided at the application level (i.e. security level 2).

The following profiles are supported: full Generic Access Profile (GAP), Generic Object Exchange Profile (GOEP), Serial Port Profile implementations, eSCO v1.2 and PAN (PAN U and PAN GN roles).

The development of DUN, LAN, fax, headset, hands free, FTP, basic imaging and object push profiles by third parties is enabled.

In addition Symbian supports the Bluetooth Stereo Headset Profile.

USB
USB class support is provided for the WMCDC WHCM and OBEX class, and for multiple CDC ACM classes. All classes are controlled by a USB Manager component. All USB support is provided ultimately by a USB v2.0client driver.

Symbian also provides support for the USB Mass storage functionality.

IrDA
The infrared IrDA stack is contained in a socket server protocol module that implements the following IrDA layers: IrLAP v1.1, IrLMP v1.1 and IrTinyTP v1.1. The following features are supported:

- infrared (SIR) supporting throughputs of 9.6 Kbps to 115.2 Kbps
- IrOBEX v1.2 (IrDA object exchange)
- IrTRANP v1.0 digital camera picture infrared transfer protocol
- IrCOMM v1.0 supports fax/modem functionality and is implemented in a serial communications server module.

The infrared message type module integrates IrOBEX handling into the messaging framework.

2.5.3 Networking services

This contains the key frameworks and system services for wide area network communications. It provides a framework for implementing various communications protocols through a socket interface.

A dual stack is provided that supports both IPv4 and IPv6. The IP stack provides a plug-in architecture allowing licensees or ISVs to implement extensions. An important plug-in delivered is IPSec, for secure communications. See Appendix A for the list of supported RFCs.

Networking support in Symbian OS includes:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- IPv4/v6 stack. The TCP/IP stack provides a plug-in architecture. Plug-ins can interact with OSI level 2, 3 and 4 components and can be installed, loaded and unloaded at runtime. IP-based Symbian OS clients such as email, HTTP, SSL, Java MIDP, SyncML over HTTP and web can use IPv6 addressing as well as IPv4 addressing, DHCP for IPv6
- Internet Control Message Protocol (ICMP)
- Point to Point Protocol (PPP)
- Domain Name System (DNS)
- dial up networking support
- security protocols for secure electronic commerce: Transport Layer Security (TLS) and Secure Sockets Layer (SSL)
- IPSec: IP layer protocol used to secure host-to-host or firewall-to-firewall communication. IPSec is a plug-in module to the IP stack providing tunnelling, authentication and encryption for both IPv4 and IPv6. VPN clients based on IPSec will be commercially available from Symbian partners
- Telnet Protocol engine
- File Transfer Protocol (FTP) engine
- Ethernet support: wired interface (PCMCIA cards for WINS and on-board Ethernet chip for development board) supports 10BaseT and 100BaseTX in full or half duplex; Wireless interface (IrLAN)
- RTP – Realtime protocol support.

2.6 Multimedia and Graphics Services

2.6.1 Multimedia

Multimedia framework

The Multimedia Framework (MMF) provides a lightweight, multi-threaded framework for handling multimedia data. The framework provides audio recording and playback and audio streaming functionality. Support is provided for video recording, playback and streaming. The framework allows developers to write efficient and powerful plug-ins.

The main features are:

- generic multimedia plug-in system: plug-ins can be written using abstract and concrete classes that represent actual resources and abstract components. Concrete classes include files, descriptors, sockets, audio i/o, and video i/o
- the audio framework provides commonly used functionality including audio playback, audio recording, audio streaming and audio conversion, formats supported include WAV, AU, RAW (in various formats), PCM, uLaw, aLaw, GSM6.10 etc.; a codec API is provided and the framework supports codec plug-ins
- audio controller plug-in; file, descriptor and microphone source plug-ins, and file, descriptor and speaker sync plug-ins
- video playback, recording and telephony framework
- MIDI client API
- Speaker dependent speech recognition API
- concurrent processing of multiple multimedia data streams.

Media Support Library (MSL)

The Media Support Library is a collection of Hardware Abstraction Layers for interfacing to Multimedia Hardware acceleration. It includes support for audio, video, MIDI and speech recognition engines.

Image Conversion Library (ICL)

The image conversion library is a lightweight, optionally multithreaded, client-side framework for still image codecs and conversion; Plug-ins supplied by Symbian include JPEG, GIF, BMP, MBM, SMS, WBMP, PNG, TIFF, WMF and ICO. APIs are provided for arbitrarily scale images.

Third party plug-ins are enabled through the extensible nature of the framework.

Camera support

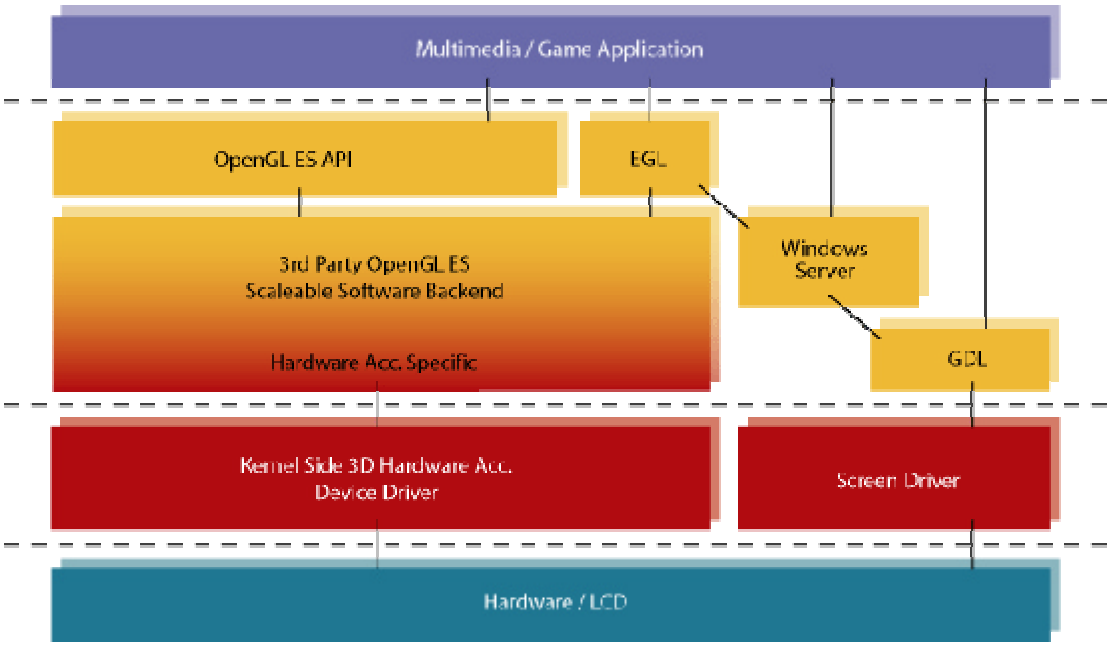
An onboard camera API providing a consistent interface to basic camera functions.

2.6.2 OpenGL ES

OpenGL ES API

A Symbian OS release of the OpenGL(R) ES 1.0 header files to ensure compatibility across Symbian OS platforms implementing OpenGL(R) ES. A reference implementation of OpenGL ES 1.0 is supplied with the platform for testing purposes. This must be licensed separately for use in shipping product.

2.6.3 Graphics



The graphics subsystem provides Symbian OS applications with shared access to the screen, keyboard and pointing devices input, bitmap fonts and scalable fonts (provided through the Open Font System), and bitmaps by using a shared heap. It also implements the Graphics Device Interface (GDI), providing a generic framework for drawing to any graphics device, and supplies concrete implementations for drawing to windows, bitmaps or to a printer. Main features are:

- direct screen access: the window server allows video (as in any visual content or graphics) to be safely rendered from hardware or a device driver
- anti-aliasing support for text on all types of displays (monochrome, color, color using palette, etc) to improve readability
- 2D Hardware Abstraction Layer (HAL) that provides hardware accelerator support for common 2D graphics operations including BitBlit operations, rectangle fill, polygon drawing, rectangle invert and rectangle fade operations.

2.7 Connectivity Services

2.7.1 Device Connection

The Connection Manager manages connections between a PC and a Symbian OS phone using standard TCP/IP protocols for data transfer. It includes both PC-side and phone-side components. Key features of the Connection manager are:

- abstraction of the bearer from the protocol layer. The Connection manager works over a range of peer to peer bearers. Supported bearers include serial links, infrared and Bluetooth
- ability to support multiple client applications on the PC. This is achieved by ensuring that all data transfers are atomic operations. The Connection manager can multiplex/de-multiplex data to/from a Symbian OS-side custom server component. In addition, the Connection manager has the ability to broadcast to all its clients changes of connection state
- supports for PC-side “unify” operation. This permits a selection of tasks to be run at a single click/cradle button press
- support for PC or Symbian OS phone based connection initiation
- detection of unexpected disconnection of a phone and broadcast of disconnection to all clients on the PC. This ensures that all custom servers open on the Symbian OS phone are shutdown
- provision of APIs for the following functionality: engine interface, remote link and device-side custom servers.

2.7.2 Connectivity Framework

The PC-Connectivity toolkit has many features, some to offer functionality out of the box, and some to ease the job of developers wanting to plug into the synching architecture:

- framework architecture including the following services:
- Synchronization
- Software Install from PC
- Backup and Restore
- Remote file management
- Time of Day Setting
- framework UI view plug-ins: an engine and a UI component, which plug into the framework UI
- archive application: lists and restores files archived using the backup facility
- task scheduler: carries out a number of regularly scheduled tasks including sync and backup. A 'unify' facility allows a selection of tasks to be run at a single click/cradle button press
- control panel: gives the user access to all the connectivity settings from one place. Individual control panel items are applets that plug into the control panel. These are: connection, log settings, machine manager, and file types
- task drivers: they consist of an engine and property pages, and plug into the task scheduler component
- backup task driver: provides the ability to back up a connected Symbian OS phone to the PC. Backed up files can later be restored using the archives view plug-in
- framework to enable sync components to be integrated.

2.8 Base Services

The base subsystem provides the programming framework for all other components of Symbian OS. The main user visible elements are the file system and the common user libraries.

2.8.1 Low level libraries and frameworks

Base also contains middleware widely used across Symbian OS. Here is the main functionality:

- the C standard library
- a relational database access API. Two DBMS implementations are provided: a small and relatively lightweight client-side implementation; and, a client-server implementation for when multiple clients must have write access to a database. Databases can be manipulated either through a subset of SQL or through a Symbian OS proprietary C++ API
- a stream store that defines two major abstractions: streams (an abstract interface to convert between an object's internal and external representations) and stores (an abstract interface to manipulate a network of streams). Stores allow externalizing and internalizing data structures as complex as whole application documents or databases. Several implementations are provided for both streams and stores including memory-based and permanent file stores. It is possible to define stackable streams doing pre-processing, for example encryption and decryption streams are provided.

2.8.2 User library and fileserver

The File server provides shared access to the filing systems, a client-side interface that hides the client-server architecture and a framework for dynamically mounting plug-in file systems, with physical storage of files associated with each filing system. File systems and reference media drivers are provided for the following types of media:

- internal RAM drive
- internal NOR Flash
- internal NAND Flash
- ATA/CF
- MultiMediaCard (MMC)
- Secure Digital (SD) memory card (including both the user and protected areas of these devices).

Main features:

- file system drivers can be added when required without having to reboot
- clients can register for notification of file-server events, for example, entries changing in given directory, changing disk or disk space crossing a specified threshold
- interoperability with other systems, the VFAT filing system (in both FAT16 and FAT32 formats) is used for removable media
- all filing systems guarantee data integrity in the face of unexpected power loss.

2.9 Kernel Services and Hardware Abstraction Interface

This layer provides an abstraction to facilitate design across multiple platforms and resources, making it easier to port Symbian OS to new types of hardware. The kernel services and hardware abstraction layer ensures Symbian OS robustness, performance and efficient power management - all essential in a mobile phone.

2.9.1 Kernel services

The kernel runs in privileged mode, owns device drivers, implements the scheduling policy, does power management and allocates memory to itself and user-mode (that is, unprivileged) processes. It runs natively on ARM cores. The kernel implements a message-passing framework for the benefit of user-side servers (such as the networking and telephony stacks and the file system). The user library is the lowest-level user-mode code, which offers library functions to user-mode code, and controlled access to the kernel.

Here are the main features:

- process, thread, program and memory management
- error handling and cleanup framework
- descriptors: strings of characters and buffers of binary data
- container classes: arrays and lists
- active objects, for event-driven multi-tasking without requiring the overheads of multi-threading
- client-server architecture, for simple and efficient inter-process communication. The client-server architecture supports both thread-relative and process-relative client resource ownership. The latter is to ease porting of code written for other platforms to Symbian OS, and delivers considerably enhanced Java performance
- a hardware abstraction layer (HAL) presenting a consistent interface to hardware across all devices
- a kernel-side power model, to allow fine-grained power management
- silent running mode: device can operate with screen switched off
- locale support including currency, time and date formatting
- internal and tightly-coupled RAM support
- the kernel can be extended by the use of DLLs (such as device drivers and kernel extensions) that can link dynamically against the kernel
- Kernel: offers hard real time guarantees to kernel and user mode threads. The kernel is optimized for ARMv5 ISA and is built using ARM's RVCT 2.1 compiler.

2.9.2 Logical Device drivers

The Base subsystem provides device drivers and/or software controllers for the following devices:

- DTE serial port
- DCE serial port
- Infrared (SIR)
- USB client 1.1
- SDIO Cards

- LCD
- Keyboard
- Digitizer
- Ethernet
- the Base subsystem also provides the following reference Multimedia reference device drivers:
- MIDI
- Audio
- Speech
- Video

The majority of these are split into a logical layer component implementing the higher layer functionality common to all devices of that device type together with a physical layer component implementing the hardware specific functionality.

2.10 Tools

The Symbian OS device creation community builds Symbian OS, device drivers, middleware components, GUI frameworks and applications into Symbian OS phones. This activity is supported with Symbian OS kits, together with hardware and software tools focused on C++ development.

2.10.1 Symbian OS Kits

Symbian delivers Symbian OS to its licensees and development partners in two products:

- Symbian OS Customization Kit, to enable licensees to quickly integrate their code bases into Symbian OS and support continued development
- Symbian OS Development Kit, a super-SDK supporting all forms of device creation development activity.

The Symbian OS kits include:

- virtually all Symbian OS source code
- extensive documentation and examples
- TechView, a GUI framework for testing OS and middleware components
- the Symbian OS emulator, supporting quick development and debugging of all Symbian OS based code (except kernel or device drivers) on Windows-hosted PCs
- ROM building tools to build ROMs for hardware development boards, for prototype or for final phone hardware.

2.10.2 C++ development tools for device creation

IDEs

Symbian OS kits support the following IDE:

- CodeWarrior for Symbian OS, an evolving product line being developed by Nokia in partnership with Symbian, supporting software development on the Symbian OS emulator or target device, and including integrated support for on-target debugging

Announcements in relation to additional IDEs are expected during the lifetime of Symbian OS v9.

Target compiler

Symbian OS v9 supports ARM's RVCT 2.2. RVCT's options for ARMv5T ISA and ARM9 architectures are supported.

Additional partner tools

Check www.symbian.com/developer and the partner solutions directory for the latest tools available from Symbian partners.

Platform

Symbian OS has been tested and verified on the following hardware reference Platform:

- TI OMAP H2 Development Board

Hardware integration boards

No additional hardware integration board have been defined for this release

Telephony stack integration and testing

A GSM telephony stack integration module is provided consisting of a TSY, CSY and NIF which communicates with the radio modem using standard AT commands. In Symbian OS v9.0 the generic TSY has been updated to support GPRS, UMTS and multihoming providing product quality reference plug-ins.

The integration module supports all functionality required by products and has been developed to work in conjunction with Symbian's hardware reference platform. It has been designed to be easily ported to other hardware (communicating either using AT command or a proprietary layer 3 interface) and may be used by licensees as the basis of their integration component.

2.10.3 Application development

Symbian OS licensee SDKs
Symbian's licensees develop phones and support software development on these devices with SDKs for the ISV community. The Symbian OS Customization Kit provides tools required by licensees to build SDKs. SDKs may be delivered to the ISV market either directly by licensees or, along with tools offerings, by tools companies.

C++
Application development is supported in C++, for high performance, access to Symbian OS native APIs, and native application look-and-feel.

Java
Symbian OS v9.1 offers a strong MIDP 2.0 implementation with many additional APIs: see above.

- the Universal Emulator Interface (UEI) is implemented, simplifying tools integration.

MIDP
J2ME MIDP provides a Java environment for even the most memory-constrained mobile phone. It provides an installation and execution environment for the many MIDlets being developed.

- Symbian's MIDP implementation is compliant with V2.0 of the MIDP specification and v1.1 of the CLDC specification
- supports installation of JAR and JAD files
- fast implementation with a small footprint
- uses light-weight threading with non-blocking IO support to ensure that waiting for IO on one thread will not block all other threads
- heap size and persistent storage are unconstrained, allowing larger, more powerful MIDlets to be run
- MIDlets look and behave very much as native applications, they use the native application installer and launcher, and native UI components
- supports native color depth
- Generic Connection Framework implementation including sockets, server sockets, datagram sockets, secure sockets, HTTP and HTTPS
- conforms to Over-The-Air Recommended Practice document for MIDlet provisioning as mandated by the MIDP 2.0 specification.