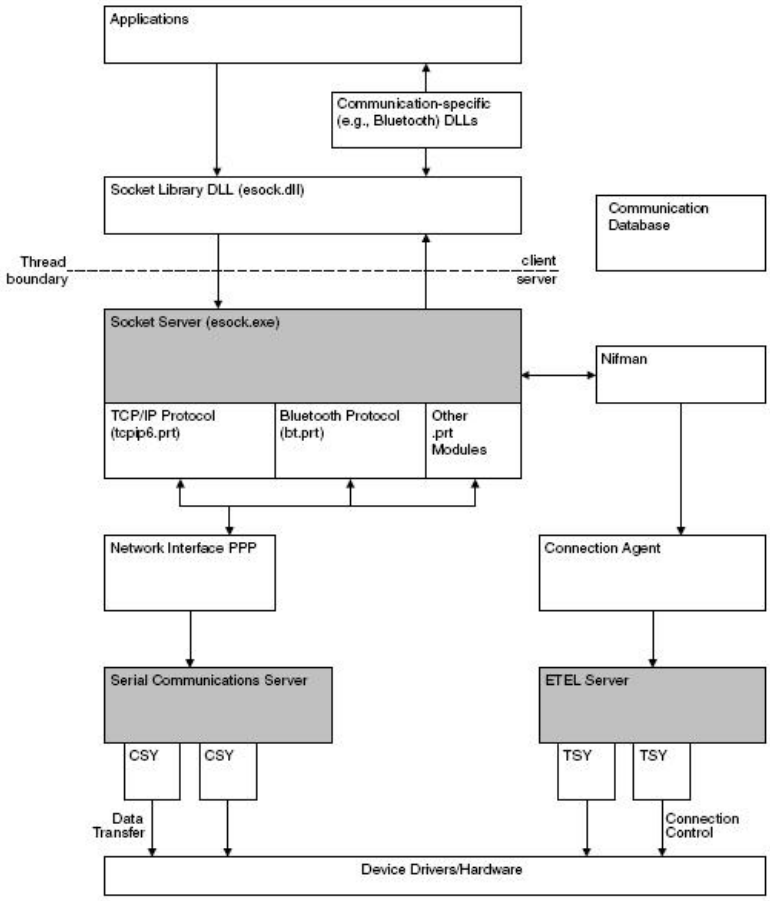
Symbian OS Communication Architecture

Main components of the Symbian OS communication structure



Communication architecture

Components of the Symbian communication architecture

Following list describes the components of the Symbian communication architecture :

- **Applications and DLLs**

Applications use networking API classes in DLLs to access communications features. As with other application-level DLLs, the communication DLLs hide the details of the underlying architecture. Symbian OS provides a socket-based API that operates in a similar way to the BSD socket API.

- **Communication-Specific Functionality**

In addition to the socket API, there are also APIs specific to certain types of communication, such as Bluetooth. A Bluetooth program would call functions from the Bluetooth DLL for device discovery, for example, then use the socket API for the bulk of the data communications.

- **Socket Server**

The socket server is a process that implements and manages communication sockets. Applications act as clients to this server through the application-level communication DLLs. As with the GUI DLLs, these functions hide the actual client/server communications from the socket server.

- **Protocol Modules**

The socket server uses protocol modules for handling the network data protocols. These are polymorphic DLLs (prt files) that implement different communication protocols, while providing a consistent interface to the socket server. Examples of protocol modules are TCP/IP, Bluetooth and IR. New protocol modules can be created and used. Protocol modules are independent of the data-link layer – bringing up the connection and exchanging data with the device is done through an abstracted interface. This

(C) Copyright Nokia 2014. All rights reserved.

interface is accomplished with two other plug-in modules that attach to the socket server: a network interface (which is usually the PPP module) and a connection agent.

- **Network Interface Manager (Nifman)**

The socket server with protocol modules use Nifman (network interface manager) to establish the connection and set up the data path to the data link level. In order to start a particular physical connection, Nifman will load a connection agent.

- **Connection Agent**

A connection agent is a polymorphic DLL that is responsible for starting and stopping the communication connection. Not only is it responsible for establishing the connection itself (e.g. dialing a number for GSM or starting GPRS), but it provides information to set up the data communication path between the physical device and the network protocol module. The connection agent will normally use ETEL to start the connection.

- **Communication Database**

A connection agent will consult the communication database to determine how to establish the network connection. This database contains all the settings applicable to communication connections. Depending on database settings, the connection agent can choose to start a preferred connection or it may prompt the user to select a connection. Once the connection is chosen, the agent will extract all the applicable connection parameters from the database to start the connection.

A connection to a network on a Symbian OS smartphone is known as an Internet Access Point (IAP). An example is a GSM CSD connection using a specific ISP phone number and login information – all stored in the communication database entries for that IAP.

- **ETEL Server**

ETEL is a low-level server used to establish a connection with a communication device. It provides an abstracted telephony API to its clients, with functions for tasks such as establishing the connection, terminating the connection, and retrieving line status and device capabilities. Modules, called TSY modules, are installed to contain the implementation for the target device. An ETEL client will load the appropriate TSY, then use the ETEL-abstracted API to control the device. Symbian OS has many built-in TSYs for devices such as GPRS and GSM (files end in .tsy).

In addition to loading a Connection Agent, Nifman will load a network interface module (DLLs suffixed with .nif). This is usually PPP.nif which implements the PPP data link protocol. This module uses the abstracted API of the Communications Server to transfer data to the device.

- **Serial Communications Server**

The Serial Communications Server provides an abstraction for serial communication across multiple devices. Reading and writing serial data and managing data flow control are example functions of this abstracted API. The details of the low-level protocols for handling a specific device are implemented in DLL CSY modules (suffixed by .csy). Example CSY modules include IR, GPRS and UART.

- **CSY Modules**

CSY modules communicate with the hardware through device drivers. The device drivers handle the actual control of the communications hardware. Symbian OS v7.0 and previous versions could have only one active IAP connection at a time. Symbian OS v7.0s introduced a multi-homing capability: the ability to have multiple IAP connections – each with its own IP address – active at once. This is useful, for example, if you want multiple functions active that use different GPRS contexts (such as MMS and web browsing). Another example is having interfaces such as WLAN and GPRS up at the same time.

This feature opens up many possibilities for devices that support multiple ways of accessing the Internet and will become more important for future smartphones.

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.