SqueezeOS Architecture

From SqueezeboxWiki

Contents

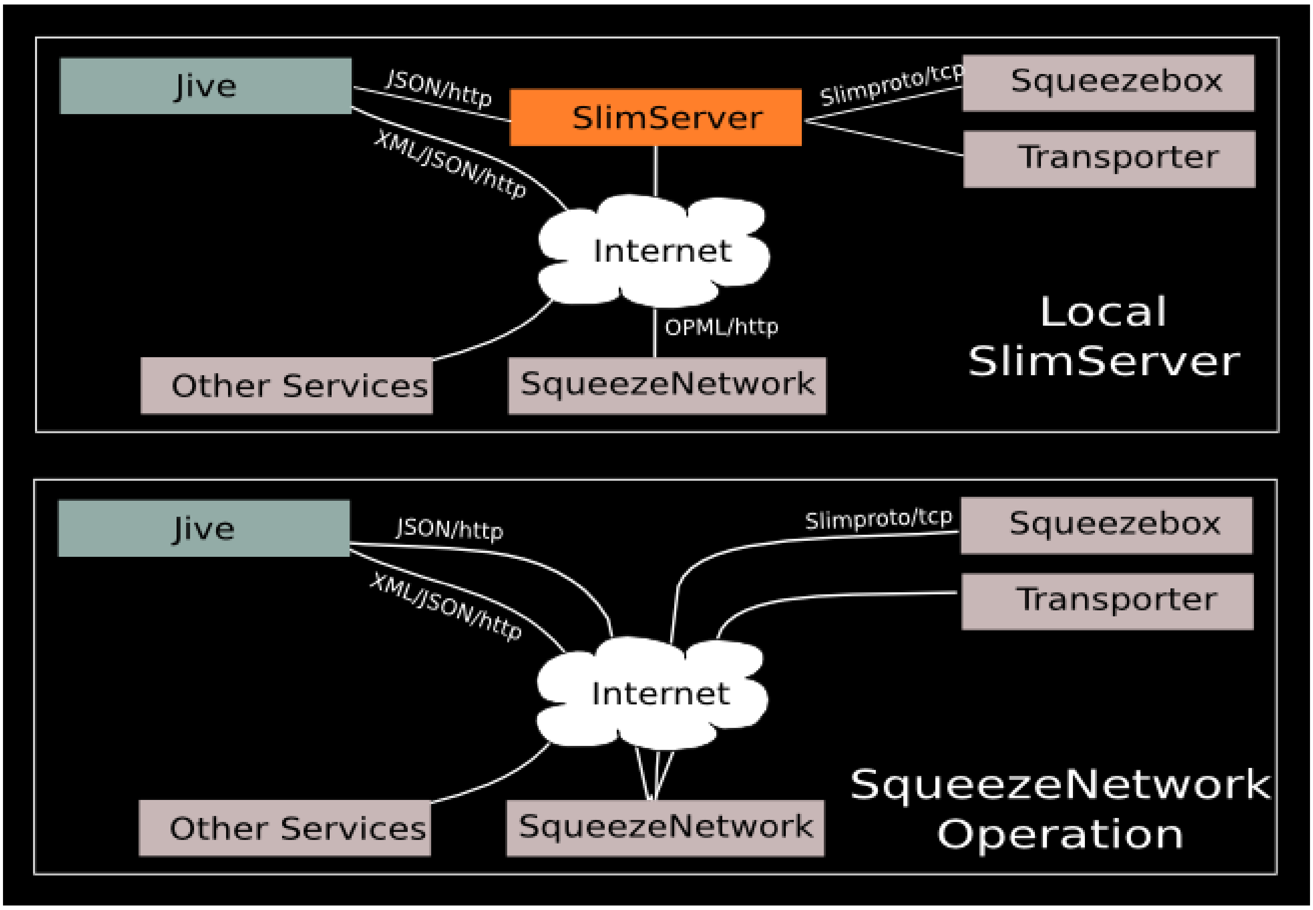
SqueezeOS Software Architecture

This document describes the SqueezeOS Software Architecture. The SqueezeOS components are described, along with all package dependencies and software licensing requirements. The communication protocols used between the SqueezeOS components are identified.

NOTE: Some of the diagrams below use "Jive" which was the the old name for the SqueezePlay application.

The design of the architecture is intended to keep the SqueezeOS software simple and leverage the capabilities of the SqueezeCenter and SqueezeNetwork. This allows rapid development of new SqueezeOS services, and service enhancements without a SqueezeOS firmware update. This is in keeping with the Squeezebox/SqueezeCenter design philosophy.

The SqueezeOS system diagram is shown below:



The following system components are used in the Logitech SMS product family:

SqueezeOS – A hardware platform or PC emulator running the SqueezePlay application software.

Squeezebox/Transporter – The Slim Devices music players, including Transporter, Squeezebox v3, Squeezebox 2, Squeezebox, slimp3 and SoftSqueeze.

SqueezeCenter – The server software that powers the Slim Devices audio players.

SqueezeNetwork – An always-on service for your Squeezebox v3, Transporter or Squeezebox2. This service allows users to listen to Internet Radio, Pandora, Rhapsody, mp3tunes and more.

Other Services – Other non-Logitech services can be accessed by 3rd party SqueezeOS applets.

The following network protocols are used to communicate between the system components.

Slimproto/tcp – The message protocol used between SqueezeCenter and the Squeezebox range of audio players (including SoftSqueeze). This protocol is used to control the audio, display and IR.

Audio/http – Audio streams are sent from SqueezeCenter or Internet Radio Stations to the audio players. Many compressed (MP3, WMA, OGG) and uncompressed (FLAC, WAV, AIFF) formats are supported.

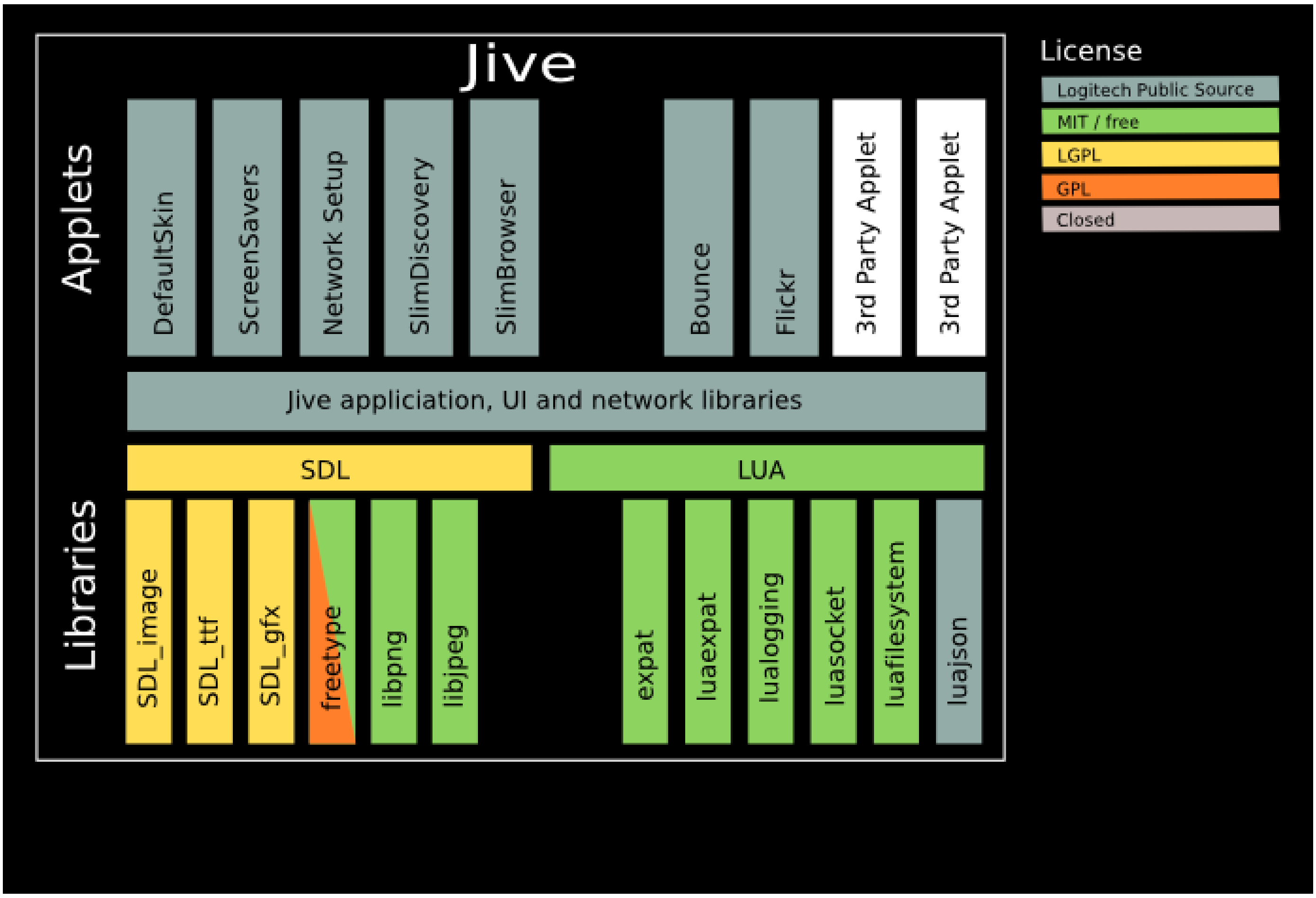
JSON/http – A serialization of the SqueezeCenter Command Line Interface using JSON over a asynchronous HTTP protocol (based on Comet). This interface allows control of the players connected to the SqueezeCenter. A subset of this protocol will be supported by SqueezeNetwork.

OPML/http – An extended OPML feed will be used between SqueezeCenter and SqueezeNetwork enabling the SqueezeNetwork Internet Services to be used with players connected to SqueezeCenter.

Two modes of operation are supported. Firstly when the user has one or more SqueezeCenters located on their local network. In this scenario all the audio players and SqueezeOS-based controllers are connected to the SqueezeCenter. The SqueezeCenter controls the state and actions of all connected devices. Internet services are available via the OPML/http connection from SqueezeNetwork (this is planned for SqueezeCenter 7).

The second scenario is when the user is not using SqueezeCenter (either they don’t have it installed or the PC is turned off). In this case all the audio players and SqueezeOS-based controllers are connected directly to SqueezeNetwork.

A block diagram of the SqueezeOS software packages and software licensing is shown below.



The application software can be running on many types of system included embedded devices and desktop PCs. All the SqueezeOS libraries are supported on a number of operating systems. A SqueezeOS Emulator, (also known as SqueezePlay) and SDK for Plugin development are available on Windows, OS X and Linux.

The feature sets supported by a specific SqueezeOS hardware platform are described in separate documentation for the hardware platform. We will now describe the software packages used by the SqueezePlay application software.

SqueezeOS Software Framework

The SqueezePlay application software provides a framework for one or more applets to provide services to the user. The applets are implemented in Lua (http://www.lua.org) . Libraries for UI elements (using SDL) and network primitives enable rapid applet development. The applet manager will allow downloading new applets from SqueezeCenter or SqueezeNetwork to SqueezePlay.

LUA

"Lua is a powerful light-weight programming language designed for extending applications.

Lua combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics. Lua is dynamically typed, runs by interpreting bytecode for a register-based virtual machine, and has automatic memory management with incremental garbage collection, making it ideal for configuration, scripting, and rapid prototyping.

The implementation goals are simplicity, efficiency, portability, and low embedding cost. The result is a fast language engine with small footprint, making it ideal in embedded systems too." - http://www.lua.org/about.html

A lightweight framework enables management of SqueezeOS Applets. Applets are loaded on demand to make the optimum use of the available memory. Some Lua extensions are included in SqueezeOS enabling threads, Unicode, bitwise operators, integer optimizations and hexadecimal to be supported.

Lua provides a suitable scripting language for development of additional services by community members and 3rd party companies for the SqueezeOS system. Lua is already used in many commercial games, of note “World of Warcraft” has an active user community developing add-ons.

User Interface

The SqueezeOS User Interface is implemented in Lua and C using the SDL graphics library.

"Simple DirectMedia Layer is a cross-platform multimedia library designed to provide low level access to audio, keyboard, mouse, joystick, 3D hardware via OpenGL, and 2D video framebuffer.

SDL supports Linux, Windows, Windows CE, BeOS, MacOS, Mac OS X, FreeBSD, NetBSD, OpenBSD, !BSD/OS, Solaris, IRIX, and QNX. SDL is written in C, but works with C++ natively, and has bindings to several other languages." - http://www.libsdl.org/

Several system widgets are implemented in Lua (with C optimizations where required). The library includes support for localization. They are:

Checkbox – a checkbox.

Choice – a list of options for selection. E.g. "On" or "Off".

Icon – an image, possibility animated. It can be included in a Label.

Label – a text label with an optional icon (or other widget). This widget supports multi-line text and horizontal text scrolling.

Menu – a list of Labels (or other widgets) for selection, with vertical scrolling.

RadioButton – a radio button.

Textarea – a text area with vertical scrolling.

Textinput – text input, including WEP key and IP address input modes.

Window – a container for other widgets.

Additional widgets will be developed as required.

Networking Library

The SqueezeOS platform provides a networking library based on luasocket. The library support asynchronous HTTP (similar to Comet, using HTTP chunked-transfer-encoding) and includes an HTTP priority queue (e.g. for downloading images).

Networking Configuration Files and Scripts

Configuration files and script used to control wired and wireless interface. SqueezeOSNetworking

SqueezeOS Applets

The following show examples of SqueezeOS applets. This section will be extended to include details of all the standard SqueezeOS applets.

Default Skin

A device skin is implemented as an applet. This includes the images and the layout rules for the interface. As the skin is scripted in lua it allows interactive skin features to be developed. For example to allow selection of the skin's background image.

New skins will be required for devices with different size displays. All the layout rules are included in the skin allowing easy customization for new devices.

Screen Savers

The Screen Saver applet allows the user to select the current screen saver and modify screen saver options (e.g. timeout). The screen savers are implemented as applets (see Bounce and Flickr).

Slim Discovery

The Slim Discovery applet finds SqueezeCenters and players on the local network.

Slim Browser

The Slim Browser applet provides control of one or more players. This includes Now Playing, Browse, Search and Internet Radio.

Bounce Screensaver

The Bounce applet provides an example of how an animated screensaver can be developed.

Flickr Screensaver

The Flickr applet provides an example of how another Internet Service can be integrated on SqueezeOS. This applet is a screensaver using photographs from the Flickr site.

SqueezeOS Libraries

SDL_image

SDL_image is an image loading library that is used with the SDL library. SqueezeOS supports PNG, GIF and JPEG images.

SDL_ttf

SDL_ttf is a TrueType font rendering library that is used with the SDL library. It depends on freetype2 to handle the TrueType font data. SqueezeOS includes the freetype fonts, other font libraries can be supported.

SDL_gfx

The SDL_gfx library offers several components including graphic primitives. The Primitives component provides basic drawing routines: pixels, hlines, vlines, lines, aa-lines, rectangles, circles, ellipses, trigons, polygons, Bezier curves, and an 8x8 pixmap font for drawing onto any SDL Surface.

LOOP

LOOP is a set of packages for supporting object-oriented programming in Lua. This is used to provide an object framework for the SqueezeOS core classes and applets.

Luajson

LUA json implements JSON encoding and decoding for Lua. This library has been developed by the Logitech SMS team, based on the json-c library and is released under the MIT license.

Luafilesystem

LuaFileSystem is a Lua library developed to complement the set of functions related to file systems offered by the standard Lua distribution. LuaFileSystem offers a portable way to access the underlying directory structure and file attributes.

Luasocket

LuaSocket is the most comprehensive networking support library for the Lua language. It provides easy access to TCP, UDP, DNS, SMTP, FTP, HTTP, MIME and much more.

Lualogging

LuaLogging provides a simple API to use logging features in Lua. Its design was based on log4j. LuaLogging currently supports, through the use of appenders, console, file, email, socket and sql outputs.

Luaexpat

LuaExpat is a SAX XML parser based on the Expat library.

Retrieved from "http://wiki.slimdevices.com/index.php?title=SqueezeOS_Architecture&oldid=8359"
Categories: SqueezeOS To Edit | SqueezeOS