Sentinel HASP Licensing Components

UML Component Diagram Example

An example of UML **component diagram** with some simplified view of provided and implemented components utilizing SafeNet **Sentinel HASP** Software Licensing Security Solution and Licensing API.

On the top of the diagram we have some software implemented using Sentinel HASP - **License Status** .Net application and **License Services Java component**. License Status application is intended to show license status and is **manifested** (implemented) by **license_status.exe artifact**. License Services Java component implements License Service **interface** and could be used by other Java applications or services.

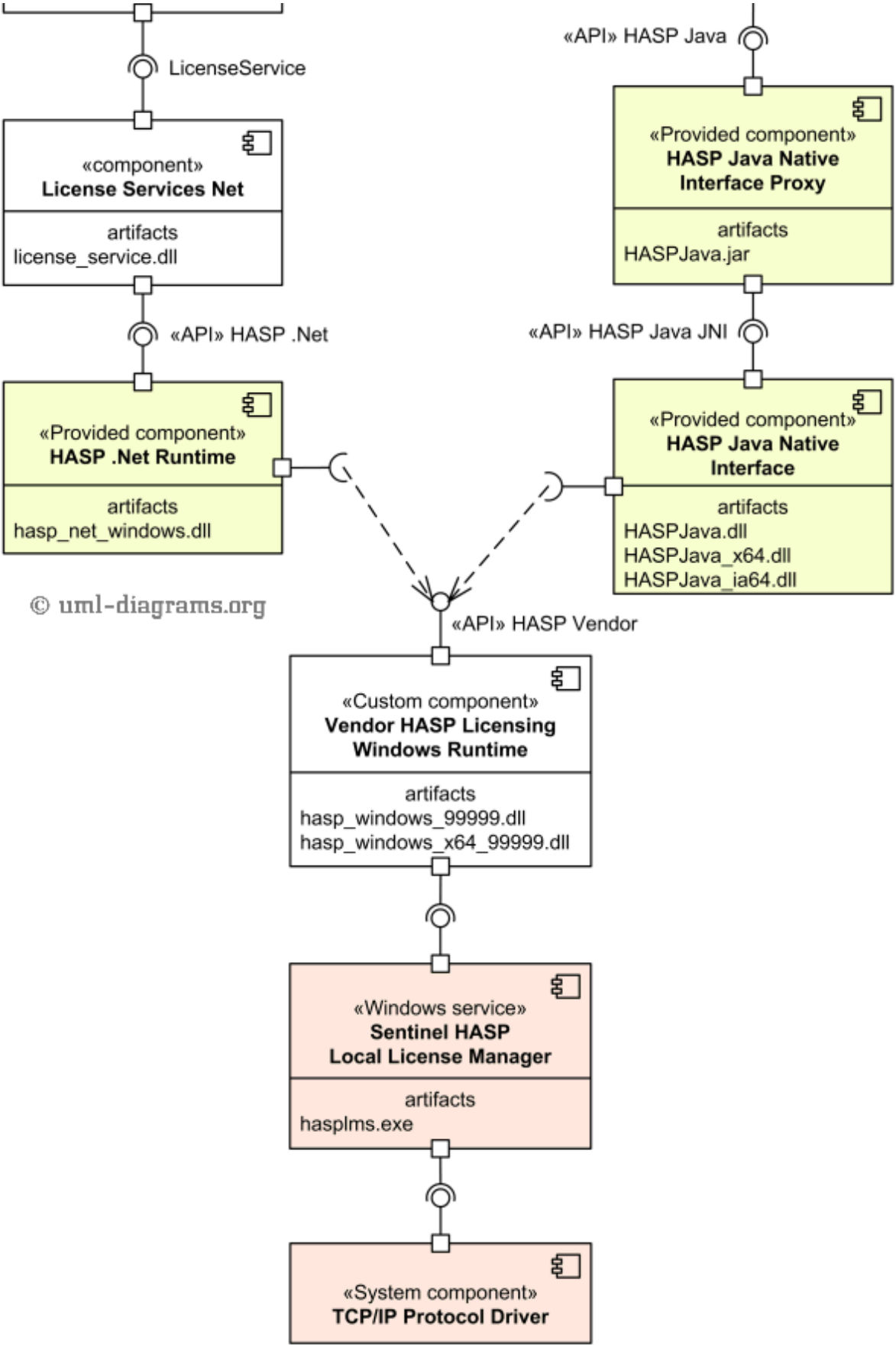
License Status application uses **License Services Net** component through the License Service interface implemented by this component. The License Services Net component uses **HASP .Net API** provided by **HASP .Net Runtime** component which is part of Sentinel HASP product.

License Services Java component uses **HASP Java Native Interface Proxy** to communicate with **HASP Java Native Interface** component, both components provided by Sentinel. When product is used in Microsoft Windows, the HASP Java Native Interface could be manifested by either **HASPJava.dll** (32 bit OS), **HASPJava_x64.dll**, or **HASPJava_ia64.dll** (64 bit OS).

The SafeNet **Sentinel LDK 6.1 Licensing API** includes:

- structural declarations and information on individual Sentinel Licensing API functions,
- description of XML tags that can be used to define scope and output format of various API functions,
- description of all API return codes.

Sentinel LDK installation includes API samples for various programming languages including C, C#, Java. For example, **HASP Java API** includes 4 Java classes: Hasp, HaspApiVersion, HaspTime, and HaspStatus in Aladdin package. Those classes are bundled into **HASPJava.jar** artifact. The HASP Java API classes load and link native methods from a platform-specific native library using the System.loadLibrary() method. As we said, for Microsoft Windows the DLL library loaded is one of the HASPJava.dll, HASPJava_x64.dll, or HASPJava_ia64.dll.



An example of component diagram for a product using Sentinel HASP Software Licensing Security Solution.

Each software vendor which uses Sentinel HASP is assigned a unique **batch code** and corresponding **vendor key**. The vendor generates and uses own customized dynamic libraries implementing HASP Vendor API (Sentinel Licensing API). Those dynamic libraries are named with vendor key as a part of the file name. The format of API library names for Windows looks like:

hasp_windows_[language/bits]_[vendorcode].[library extension]

For example,

hasp_windows_x64_99999.dll

is a 64 bit DLL API library associated with the software vendor key 99999.

When Sentinel HASP Runtime is installed in Microsoft Windows, it also installs **Sentinel HASP Local License Manager** manifested as **hasplms.exe** and running as a local **Service**. This service has dependency on **TCP/IP Protocol Driver** system component.

by Kirill Fakhroutdinov

This document describes **UML 2.5** and is based on **OMG™ Unified Modeling Language™ (OMG UML®) 2.5** specification ***[UML 2.5 FTF - Beta 1]***.
All UML diagrams were created in **Microsoft Visio** 2007-2016 using ***UML 2.2 stencils***. You can send your comments and suggestions to webmaster at **webmaster@uml-diagrams.org**.