License Management Solution

Proposal Letter

Dear Valued Customer,

If you are looking for a license management system, I would like to introduce a fast, simple and secure solution to you. Please spend few minutes to read below brief information. You will find the latest version of all demo video, software in https://github.com/vuquangtrong/PoC/tree/master/LicenseManagement.

Contact me at any time: +84 973-750-337, *skype:* vuquangtrong or *mail:* vuquangtrong@gmail.com

1. Introduction

This solution aims to provide a **fast, secured and simple** license management system which includes:

- Single / Volume License
- Offline / Online Activation
- Hardware-based ID
- Selectable Features
- Traditional License Key
- Modem License Key with signature verification
- Automatic licensing server
- Retained license on device

The solution is written in C# .Net 4.0 framework, provided in DLL with predefine UI controls, License Management APIs to easily embed into your application.

1.1.   Single / Volume license

Single license will tight to only one device. Volume license will count the number of activated devices. Both of two methods can have *Time Limitation* to automatically set the license expired. Therefor you can use license as a *subscription*.

1.2.   Offline / Online Activation

If the target device has the Internet connection, license can be acquired automatically by sending a request to *Activation Server*. The activation process is fast and simple based on REST API and JSON. Key is sent directly to your application if customer already purchased. This also helps customer if they re-install software.

If there is no Internet, customer can activate application by offline ***.lic** file which is sent via emails when they purchase a license.

1.3.   Hardware-based ID

Each time your application runs on a computer, it generates a unique ID based on Hardware ID which includes *Processor ID*, *Hard Disk Serial Number*, *Motherboard ID* and some secret keys.

1.4.   Selectable Features

License file includes direction to enable/disable a feature of application. You won't need to build different versions, just include license checking steps before calling to your features in application.

1.5.   Traditional License Key

Traditional Key consists of short phases which allow to print on CD Label, Software Box. Users can easily type the key into software. However, this type of license key is not secure because the algorithm which generate key is embedded on your application.

1.6.   Modem License Key

Using RSA algorithm, the license is encrypted and signed with your private key in your server only. Application will use your public key to verify the license file to make sure that the license is not tamped by any outside incident. This method is quite secure, but use should use a license importer.

1.7.   Automatic licensing server

Serving online activations, there is a server that responses to activation requests from applications. The license is automatically generated after a purchase is made and then the license will be sent to customer directly. This method is fast, secure, and can be integrated to your selling system.

1.8.   Retained License on device

The license can be saved into a secret location which will be remained even user re-install OS. When user re-install your application again, the license is automatically checked and activated.

2. Brief of Architecture and Solution

2.1.   Purchasing and Activation
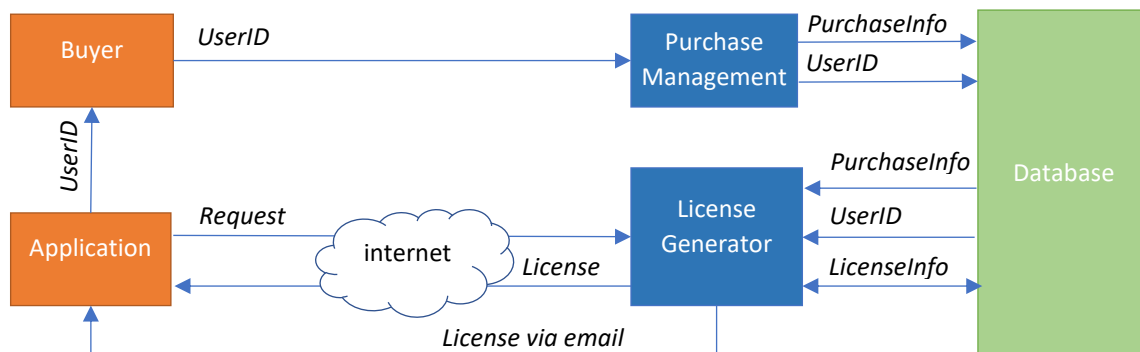
1. Buyers install your application and get the unique *UserID*
2. Buyers will send their information with *UserID* when purchasing your product. Buyer choose the options of license, system will make *PurchaseInfo* in pair of *UserID*
3. Database saves *PurchaseInfo* and *UserID* which will be used by License Generator
4. License is generated with buyer's *UserID*, *PurchaseInfo*, historical *LicenseInfo*, and sent to buyers via email or to application via online activation
5. Online Activation is performed with internet connection automatically
6. Offline Activation needs user to import license file which is received in email
7. Volume License will have a *VolumeCode* which is shared between users/ devic

2.2.   License Generation
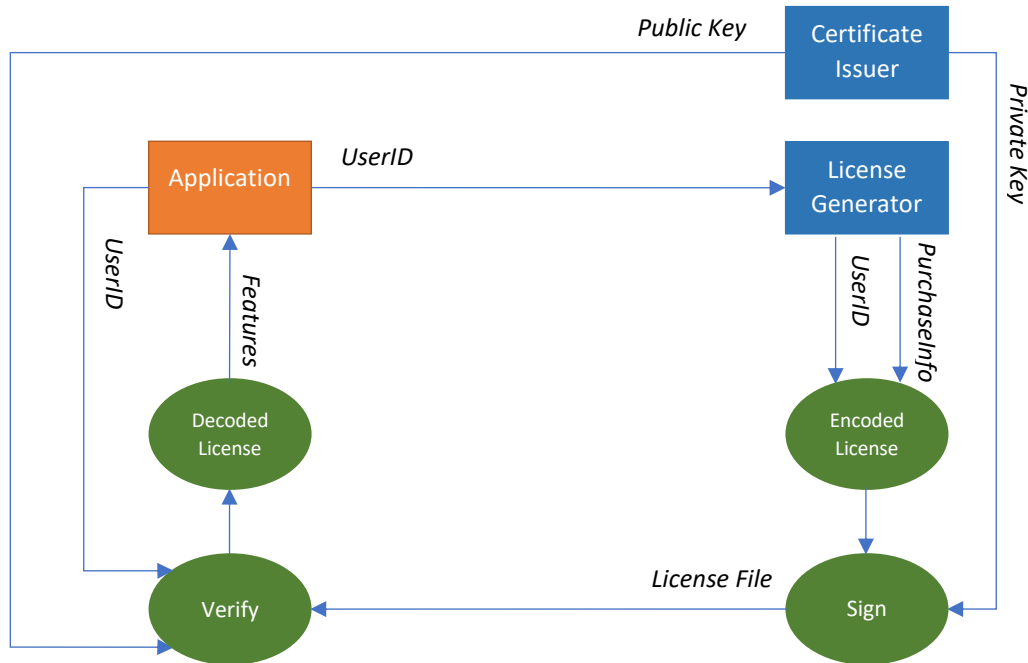
1. Your system will generate a certificate which will give out a PrivateKey and a PublicKey
2. PrivateKey is used in your License Generator only, to sign and encrypt the LicenseObject
3. PublicKey is used in your application to verify and decrypt the LicenseObject
4. Your application only has decrypting code, and the pair of keys are unique

3. Easy to be Integrated

The License Management Solution is provided in DLL package.

3.1.   License Entity

You only need to extend *LicenseBase* class and add your additional properties into your license.

```
namespace MyLicense
{
    12 references
    public class DemoLicense : LicenseEntity.LicenseBase
    {
        // Set target app of this license
        public const string TargetAppName = "DemoClient";

        // customized properties
        [DisplayName("Enable Feature A")]
        [Category("License Options")]
        [XmlElement("EnableFeatureA")]
        [LicenseInfoAttribute(true, "Enable Feature A", LicenseInfoAttribute.FormatType.String)]
        3 references
        public bool EnableFeature_A { get; set; }
```

To process your license, *LicenseHandler* will take care of Generate UID, Validate UID, Generate License, Load License, Sign and Verify License

3

3.2.   License User Controls

DLL package contains pre-defined User Controls. You only need to drag-then-drop UI to your application. After that, you can call methods of controls to set/ get data.
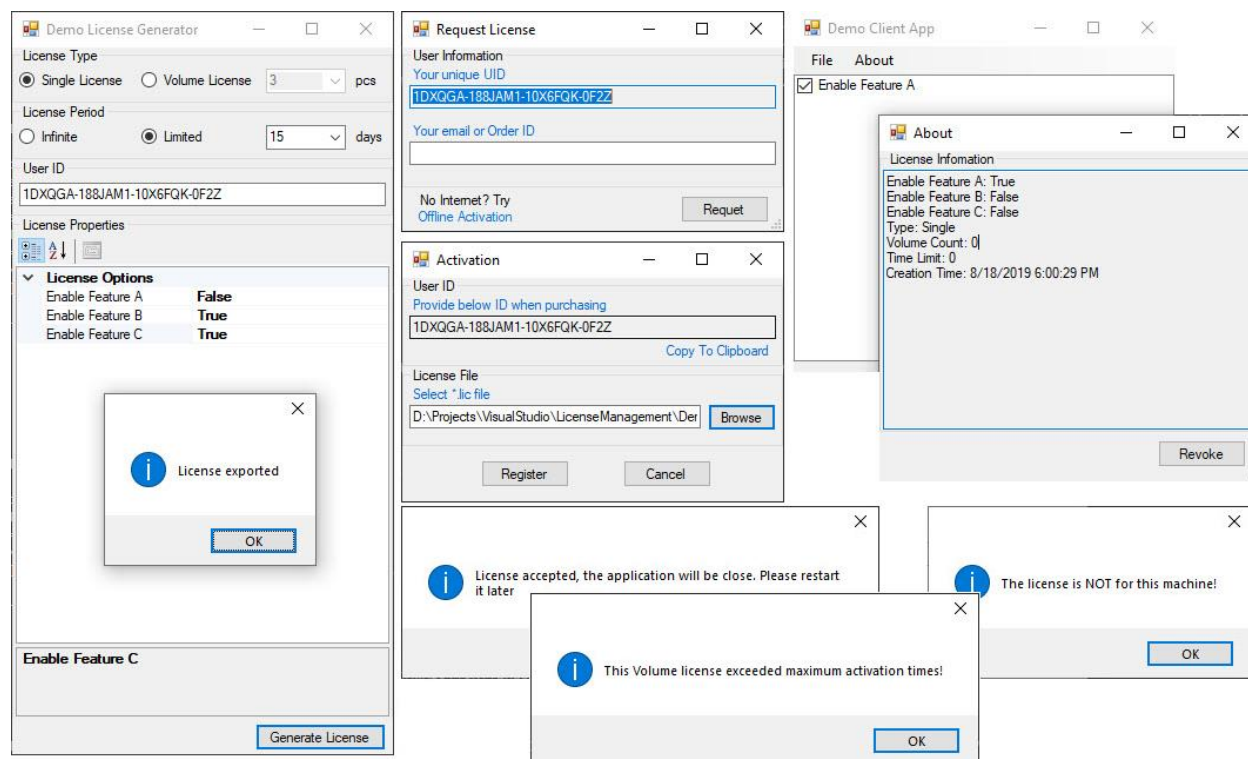
3.3.   Console version

The solution includes GUI and Console version. Console version is used for automated licensing server. In demo version, it is integrated with webserver to generate license upon user requests.

4. Development and Integration Plan

The solution is developed completely as DLL format with necessary components. If your system needs more than pre-included options, I will develop those additional features. I will support you to integrate this License Management Solution into your application and system.

Depending on your current application status and how you want to deploy the solution, the integration may take a week or a month.

If you are interesting to this solution, please feel free to contact me. If you are in Hanoi, we can meet each other to have a detail plan for our cooperation.

5. Technical Background

I graduated from Hanoi University of Science and Technology in 2013. My major is Electronics and Telecommunication. Currently, I am working as a Part Manager, a Technical Leader of Humax Automotive Part – a company makes devices and software for Car Manufactures in Korea, China, US and EU. I have been writing a lot of software which are used for my personal projects as well as for company and for sale.