Database schema for the licensing and provisioning related stuff

1. Every package will have a license.
2. APL will load the license for the package which is launched.
3. APL will then validate the license and check for its expiry.
4. All information related to license is in encrypted format. (??) will it be a symmetric key algorithm since data is read write in this db.
5. The license files will be stored in some location on the machine which is not uninstalled. So, if a user installs the same package with the same serial number, it will pick the older license and then run the product with that license.
6. It will be a licensing and verification based on the input file in which the validation logic is written in an encrypted format.

Send the public key along with the media. It may happen that in future, the private-public key pair may need to be changed and hence it should be self-upgradeable and manageable.

Method to avoid the infringement by changing the executable and dll.

The hacker can patch the exe and binary dlls. Either sign it and then check the signing before exe or dll. use handshaking in every api call.

The validating string from the server will consist of the time for which the license would work. This is a configurable entity on the server side. It will also have a default value and a maximum limit.

The library format:

1. APL: a dll which will contain most of the licensing and serial number related apis. It is a signed dll and it will be checked against its hash for any patching done in it. This dll will be used by installer at the time of installation.

Use cases where the hacker can misuse the application and the solution to it.

1. User changes the time of the system and keep using the application. He can also slow down the system time.
2. User formats the machine and reinstalls the application with the same entry code which was given to him earlier and he gets a grace period.

**Solution I**: in the entry code, return the time also from the server and licensing should start from that time. Also, if user machine time is very different, warn him to set the proper time.

**Solution II**: every time put some random number also in the challenge key. Server will encrypt this challenge key and then activation will go on.

1. User can copy the whole product files from one computer to another.

It should take into account the scenario where the APL versions would change and hence the validating logic. There can be co-existence issue with that.