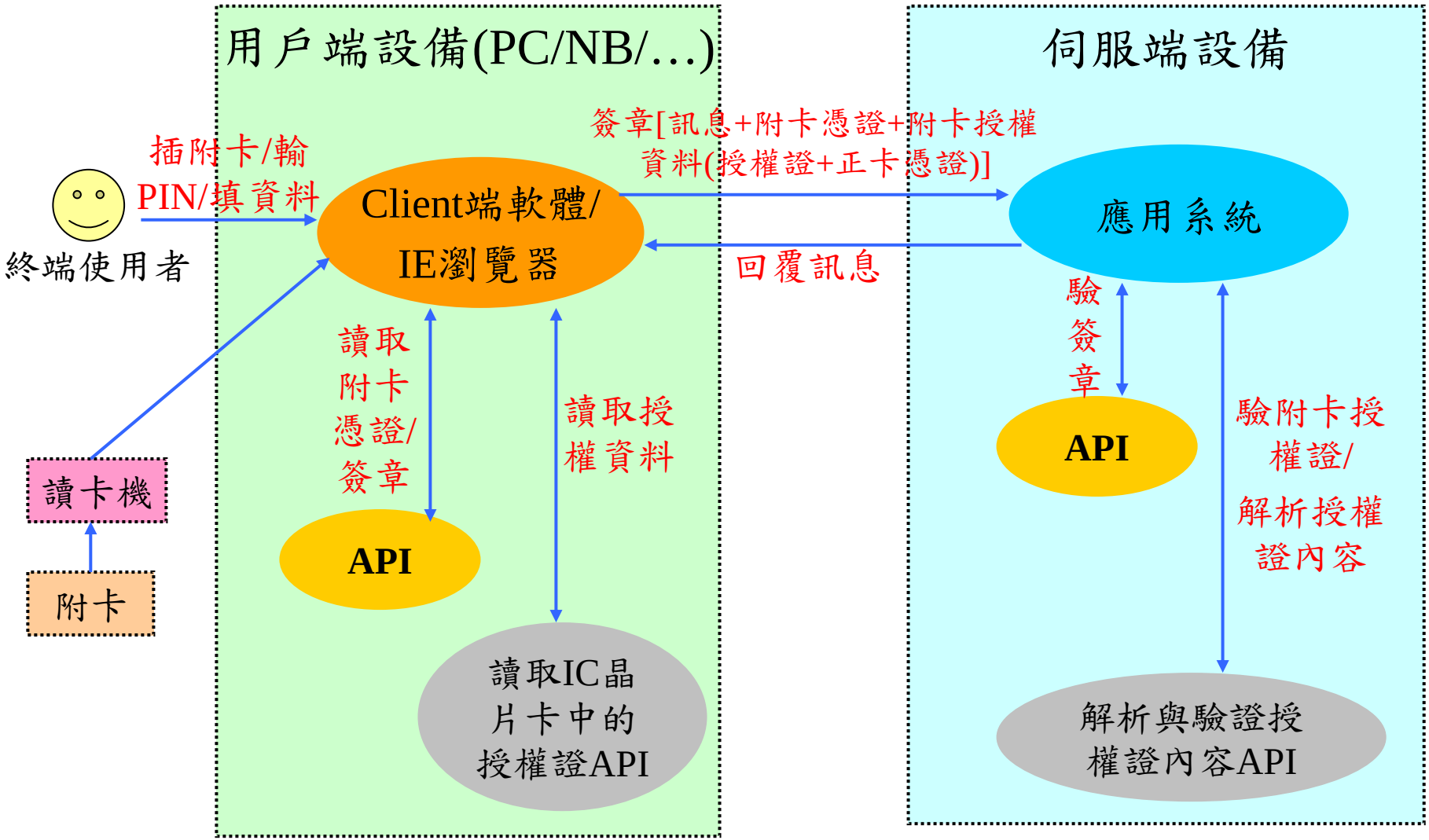
經濟部

工商憑證附卡授權應用API使用簡介

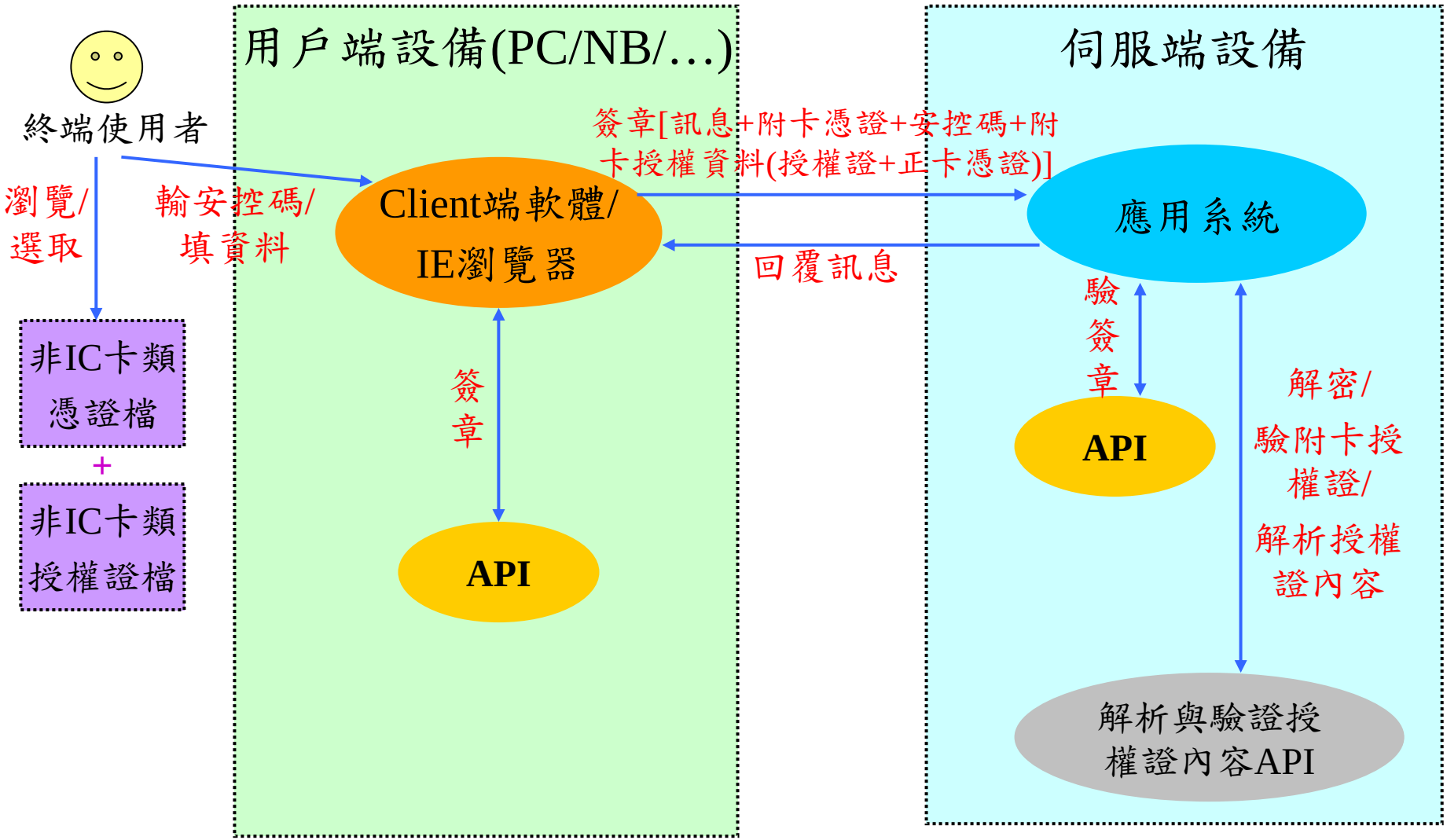
中華電信股份有限公司
電信研究所資通安全研究室

99年11月

API應用情境示意(IC卡類)

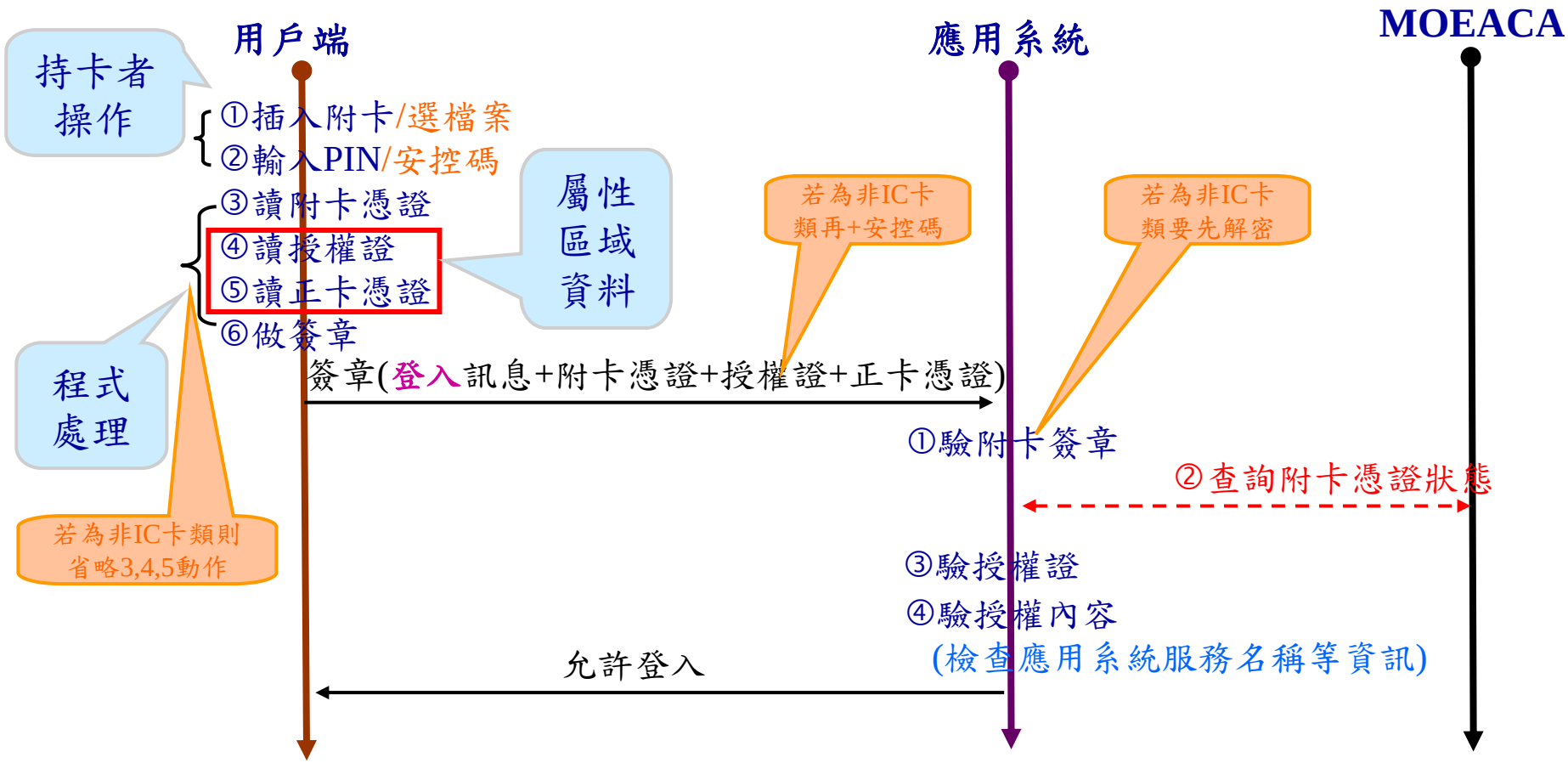


API應用情境示意(非IC卡類)



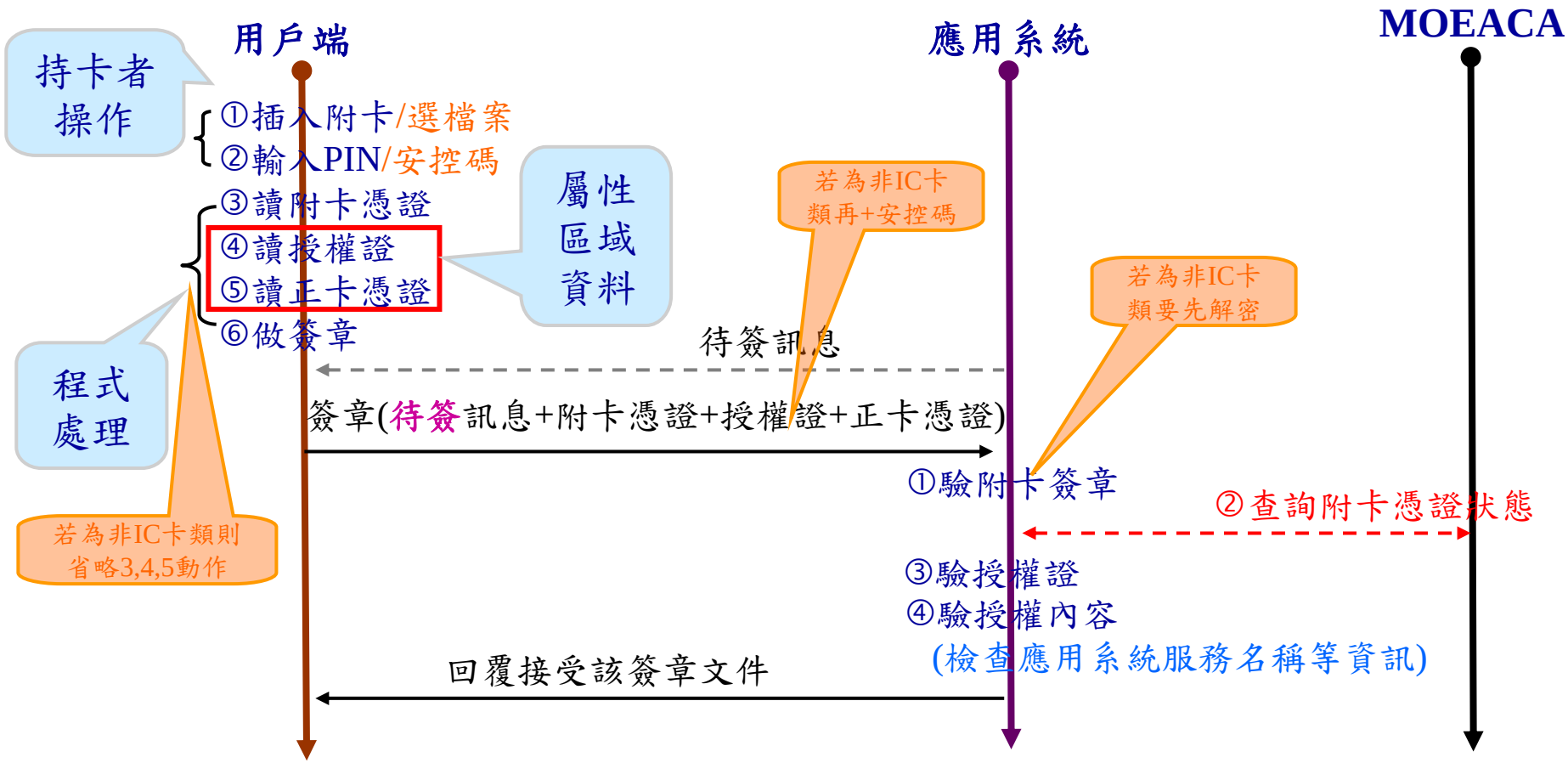
API應用情境時序圖－系統登入

- 用戶端由IC卡讀資料(正附卡憑證、授權證)
- 應用系統本身驗授權證與授權內容



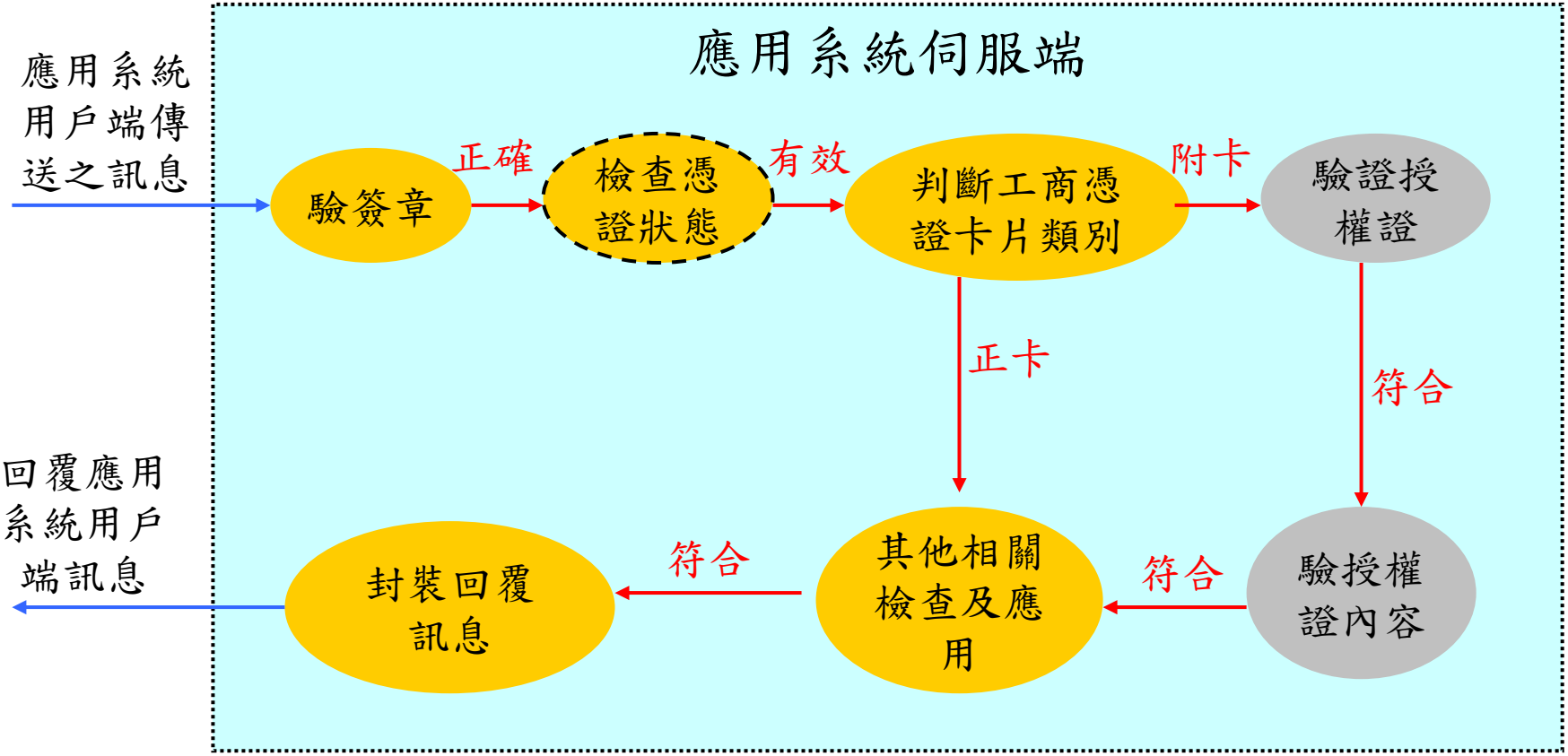
API應用情境時序圖－線上文件簽核與驗證

- 待簽訊息包含統一發票、專利證書...等電子化文件。
- 查詢附卡憑證狀態可透過CRL或OCSP



應用系統檢查步驟流程圖

- 應用系統至少應有以下檢查步驟，可依應用系統所需自行增加檢查項目。



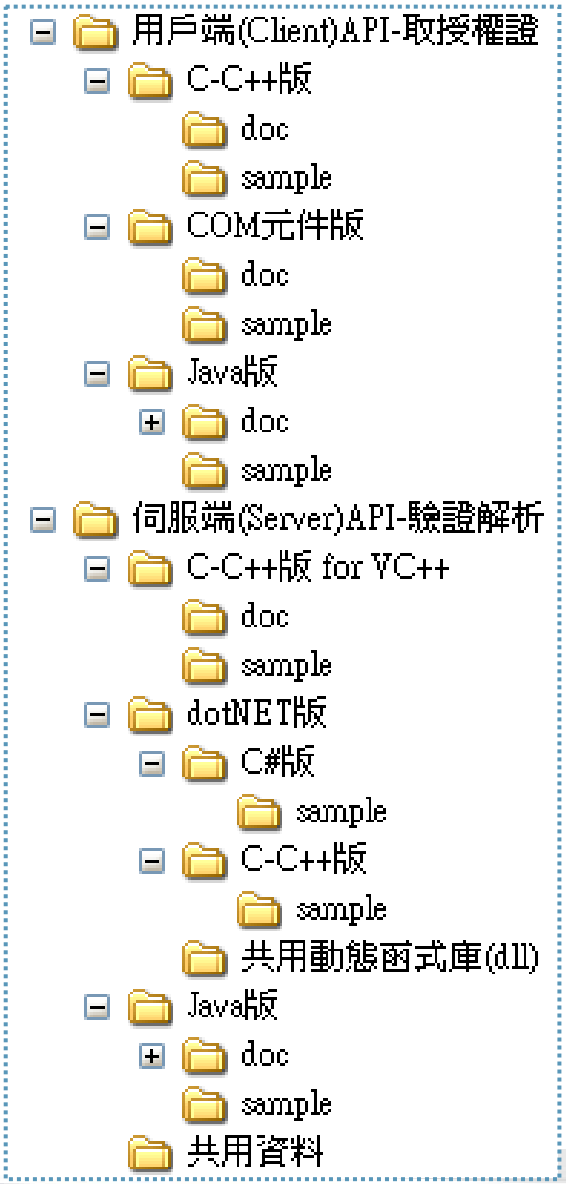
測試套件

- 光碟一片
  - 應用程式介面(API)檔案
  - API使用相關文件
- GTestCA測試卡兩張

附卡授權API的支援性

	AP的程式語言版本	是否提供API
用戶端API (讀授權證)	Java (*.jar)	■是
	COM元件 (*.cab)	■是
	C/C++ for .NET (*.dll)	■是
	C/C++ for VC++ (*.dll)	■是
伺服端API (驗證及解析授權證)	Java (*.jar)	■是
	C# (*.dll)	■是
	C/C++ for .NET (*.dll)	■ 是
	C/C++ for VC++ (*.jar)	■ 是

API目錄結購



- 各版本目錄：各程式語言版本API存放地方
- doc：說明文件
- sample：各程式語言版本範例程式
- 共用資料：伺服端各程式語言共用的測試檔案資料

讀取IC晶片卡中的授權證API (Client端)

讀取IC晶片卡中的授權證API -大綱

- 支援與清單說明
  - Java
  - C語言
  - COM元件
- 使用方法
- 提供函式
- 使用範例

讀取IC晶片卡中的授權證API -支援與清單說明

目前支援：

- Java
  - 動態函式庫： ICCDFCLIENT.dll
  - 執行壓縮檔(.jar)：ICCDFCLIENTJProj.jar
- C語言
  - 動態函式庫：ICCDFCDLL.dll
  - 靜態函式庫：ICCDFCDLL.lib
  - 表頭檔(.h)：ICCDFC.H
- COM元件
  - ActiveX：ICCDFCOM.cab

讀取IC晶片卡中的授權證API使用方法 – Java(1/2)

- 設定連結與函式庫置放
  - 在Project中加入ICCDFCLIENTJProj.jar
  - 將ICCDFCLIENT.dll放置於可載入的路徑上
- 開發程式注意事項
  - .java原始程式中加入import com.chttl.icc.ICCDFCLIENT ;
  - 讀取授權證資料函式呼叫的順序，一定要先ICcard.readUserData( )再ICcard.getUserData( )

讀取IC晶片卡中的授權證API提供函式 – Java(2/2)

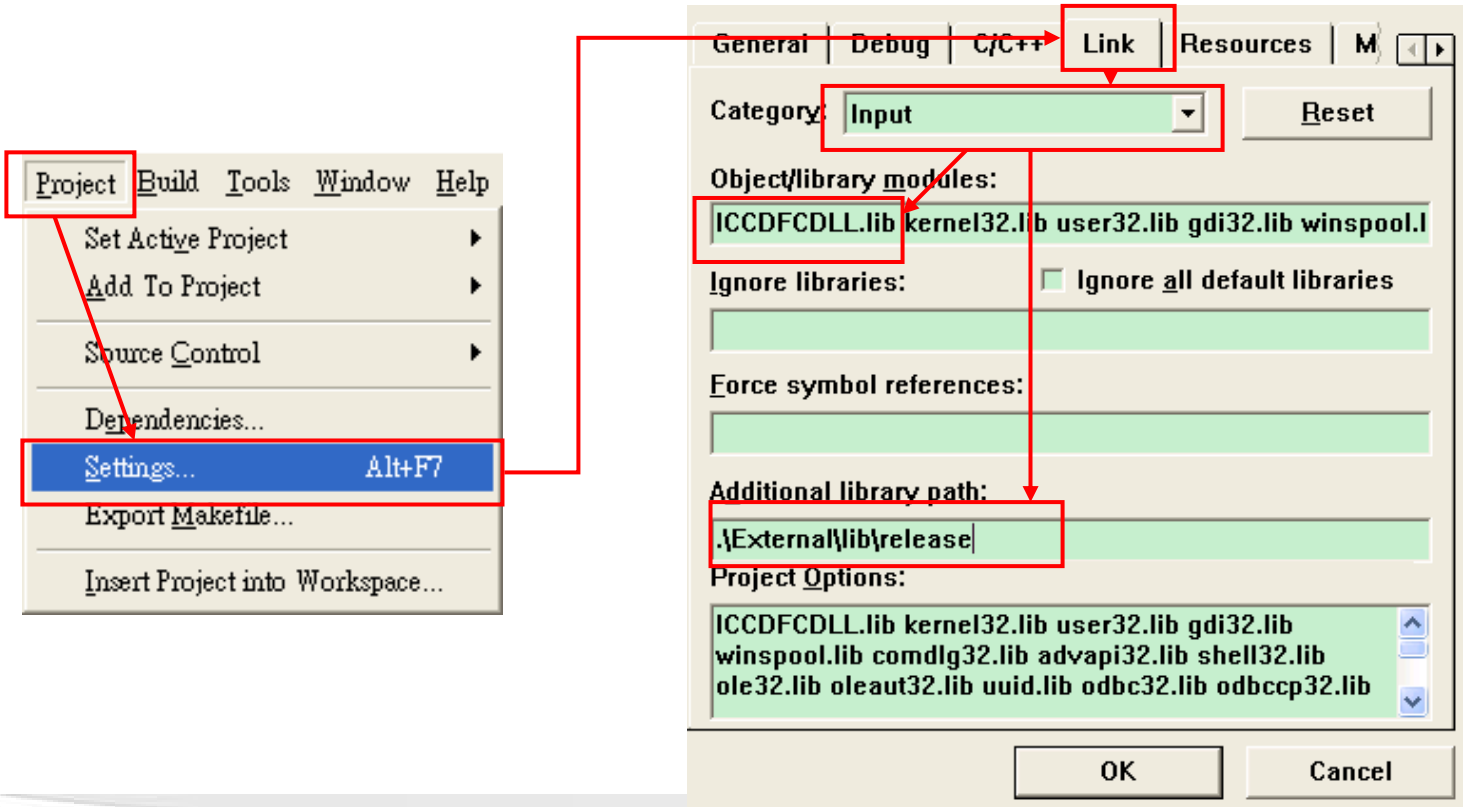
- Class ICCDFCLIENT
- Constructor Summary：ICCDFCLIENT()
- 提供函式
  - readUserData：讀取工商憑證附卡IC卡中授權證空間資料命令
  - getUserData：取得工商憑證附卡IC卡中授權證空間資料
  - getUserDataLen：取得工商憑證附卡IC卡中授權證空間資料長度
- 使用範例

```
ICCDFCLIENT ICcard = new Iccdf();
ICcard.readUserData(null, “12345678”, “0001”);
String S_UserData = new String(ICcard.getUserData());
```

讀取IC晶片卡中的授權證API 使用方法－C語言(1/4)

設定連結函式庫路徑

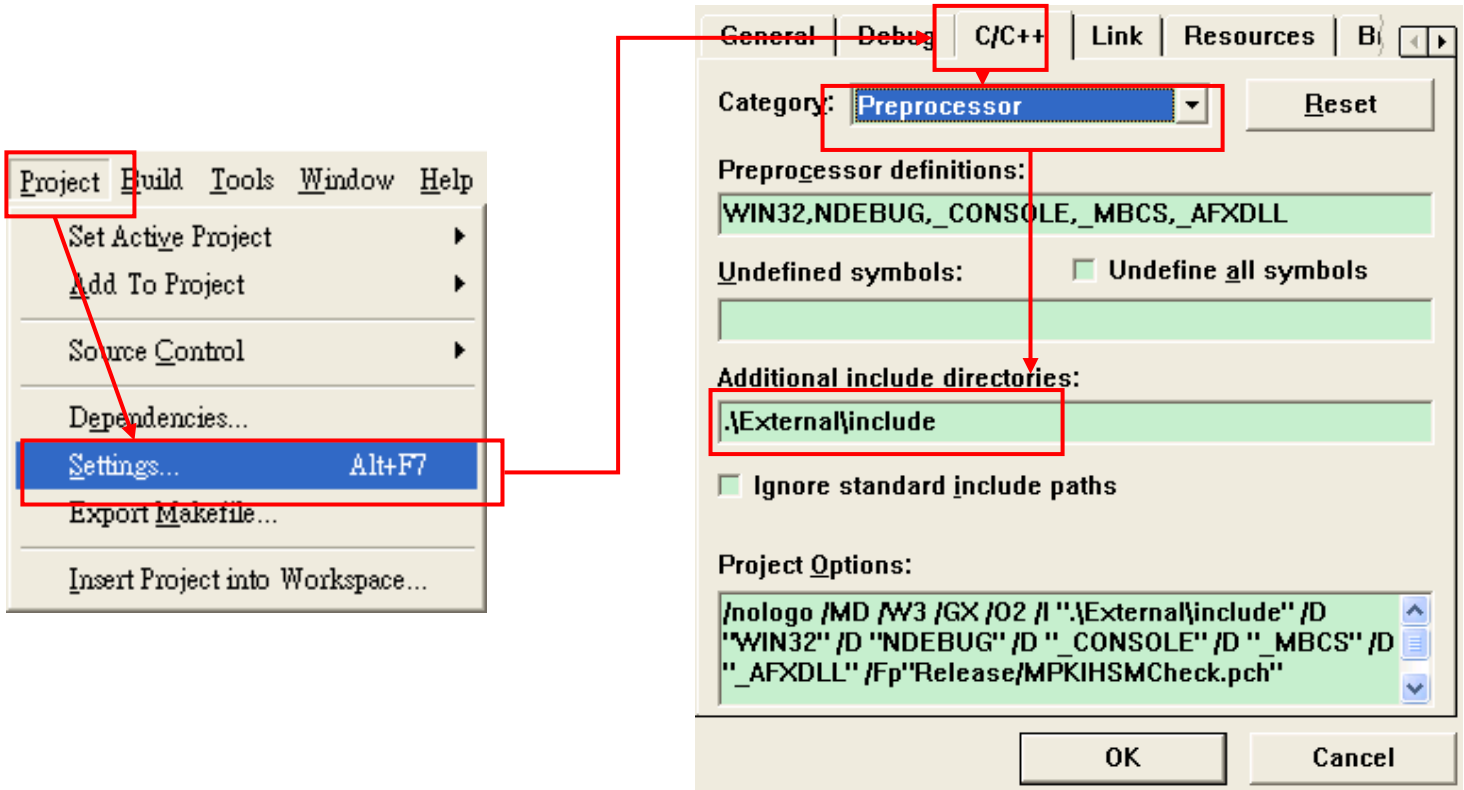
- Project→Settings…→Link頁籤→Category欄位選擇Input
  - Object/library modules欄位新增ICCDFCDLL.lib
  - Additional library path欄位填寫正確路徑



讀取IC晶片卡中的授權證API 使用方法－C語言(2/4)

設定連結表頭檔路徑

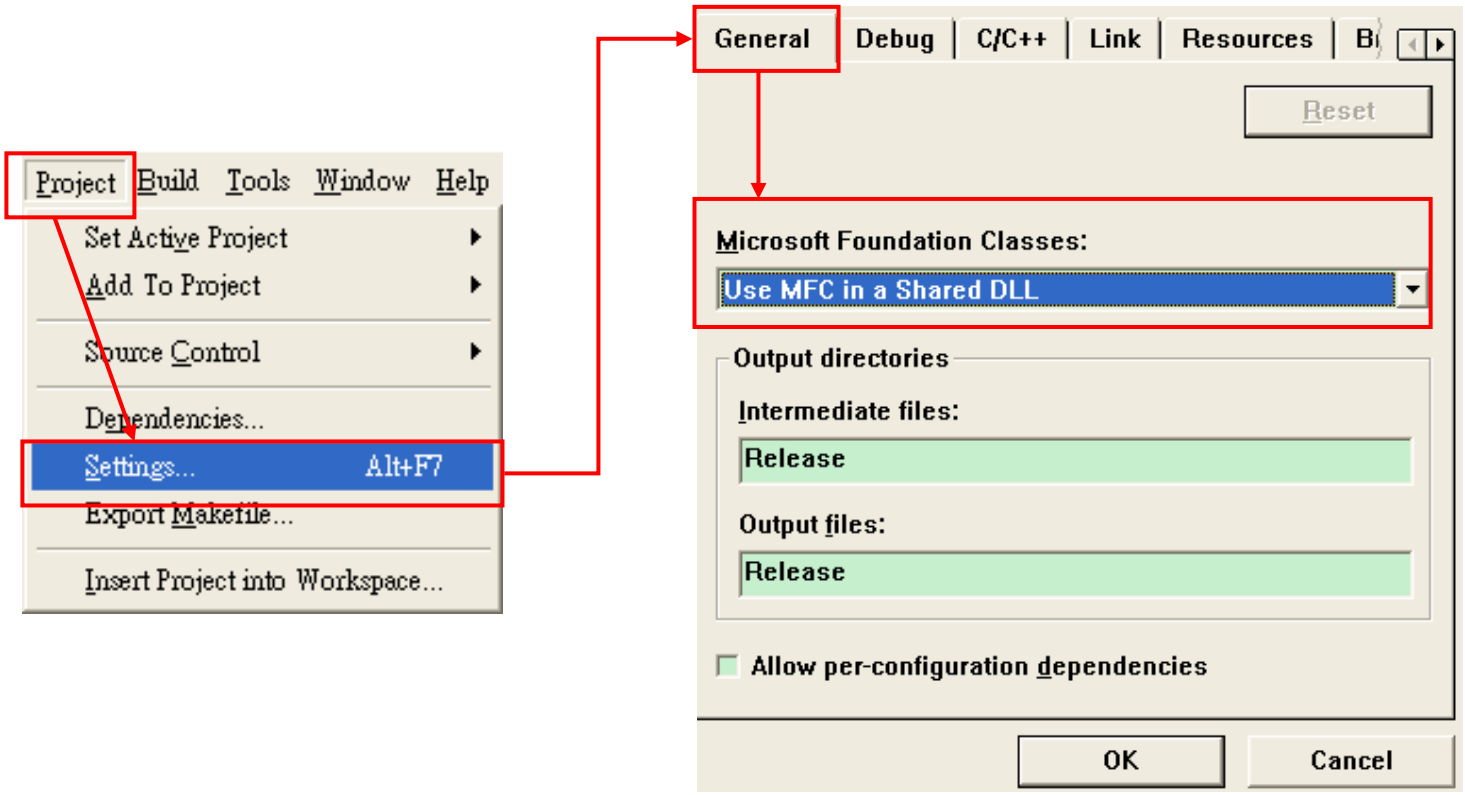
- Project→Settings…→C/C++頁籤→Category欄位選擇Preprocessor
  - Additional include directories欄位填寫正確路徑



讀取IC晶片卡中的授權證API 使用方法－C語言(3/4)

設定程式類別

- Project→Settings…→General頁籤→Microsoft Foundation Classes欄位選擇Use MFC in a Shared DLL



讀取IC晶片卡中的授權證API 使用方法&提供函式－C語言(4/4)

- 開發程式注意事項
  - .cpp原始程式中加入#include “ICCDFC.H”
  - 產出的執行檔(.exe)同目錄要放入ICCDFCDLL.dll動態函式庫
- 提供函式
  - readUserData：由工商憑證附卡IC卡中取出授權證資料
- 使用範例

```
unsigned char       data[3000];
int                 dataLen=3000;
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~(省略)~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
readUserData(NULL, (unsigned char *)"12345678", 8, (unsigned char *)"0001", data, &dataLen);
```

讀取IC晶片卡中的授權證API使用方法 – COM元件(1/2)

- 使用在網頁中的ActiveX元件供使用者下載
- 使用說明
  - 介於<head>與</head>之間，增加<OBJECT>

```
<OBJECT id=ICCDF style="LEFT: 0px; TOP: 0px"
codeBase=".\ICCDFCOM.cab#Version=1,0,0,1" classid="clsid:B487BEA3-404B-4158-
950C-DF64352CB7DF" VIEWASTEXT>
        <PARAM NAME="_Version" VALUE="65536">
        <PARAM NAME="_ExtentX" VALUE="2117">
        <PARAM NAME="_ExtentY" VALUE="1058">
        <PARAM NAME="_StockProps" VALUE="0">
</OBJECT>
```

讀取IC晶片卡中的授權證API 提供函式 – COM元件(2/2)

提供函式

- readUserData：由工商憑證附卡IC卡中取出授權證資料

使用範例

```
<SCRIPT language=vbscript>
   sub ReadUserData()
      rcode = ICCDF.readUserData(NULL, "12345678", "0001", data)
      if(rcode<>0) then
         msgbox "readUserData fail:" & rcode
         exit sub
      end if
   end sub
</SCRIPT>
```

解析與驗證授權證內容API
(Server端)

解析與驗證授權證內容API — 大綱

- 支援與清單說明
  - Java
  - C# for ASP.NET
  - C/C++ for .NET
  - C/C++ for Visual C++
- 使用方法
- 提供函式
- 使用範例

解析與驗證授權證內容API支援與清單說明(1/3)

- 目前支援：
  - Java
    - 範例測試檔：TestAc.java
    - 執行壓縮檔(.jar)：CHTAttributeCertificate.jar
  - C# for ASP.NET
    - 範例測試檔：TestAC.cs
    - 動態函式庫： CHTAttributeCertificate.dll、IKVM.OpenJDK.Core.dll、IKVM.OpenJDK.Security.dll、IKVM.OpenJDK.Text.dll、IKVM.OpenJDK.Util.dll、IKVM.OpenJDK.XML.API.dll、IKVM.OpenJDK.XML.Parse.dll、IKVM.Runtime.dll

解析與驗證授權證內容API 支援與清單說明(2/3)

- C/C++ for .NET
  - 範例測試檔：TestAC.cpp
  - 動態函式庫： CHTAttributeCertificate.dll、IKVM.OpenJDK.Core.dll、IKVM.OpenJDK.Security.dll、IKVM.OpenJDK.Text.dll、IKVM.OpenJDK.Util.dll、IKVM.OpenJDK.XML.API.dll、IKVM.OpenJDK.XML.Parse.dll、IKVM.Runtime.dll
- C/C++ for Visual C++
  - 範例測試檔：TestAC.cpp
  - 執行壓縮檔(.jar)：CHTAttributeCertificate.jar
  - 動態函式庫(視系統可有可無)：MFC42D.DLL、MSVCP60D.DLL、MSVCRT.DLL、MSVCRTD.DLL

解析與驗證授權證內容API 支援與清單說明(3/3)

- 共用資料檔
  - 範例測試檔： MOEACA_附卡授權證_??????已加密.sec、MOEACA_附卡授權證_??????未加密.p7b、Test4SCAP01For??????.cer、QueryNonICCardStatusPath.ini(for非IC卡類)

解析與驗證授權證內容API 使用方法 – Java(1/8)

- 設定連結與函式庫置放
  - 在Project中加入CHTAttributeCertificate.jar
  - QueryNonICCardStatusPath.ini與CHTAttributeCertificate.jar放置於同目錄
- 開發程式注意事項
  - .java原始程式中加入
    - import tw.com.cht.ac.CHTAttributeCertificate;
    - import java.security.cert.*;
- 參考測試檔TestAC.java

解析與驗證授權證內容API 提供函式 – Java(2/8)

- Class CHTAttributeCertificate
- Constructor Summary
  - CHTAttributeCertificate(java.io.InputStream is)
    - 輸入參數is為由IC卡中取得之P7B格式的授權證資料，可能由檔案中讀取
  - CHTAttributeCertificate(java.lang.String hexP7b)
    - 輸入參數hexP7b為經過16進位編碼後之P7B格式的授權證資料字串
  - CHTAttributeCertificate(byte[] ICcardGetACCert)
    - 輸入參數 ICcardGetACCert為直接讀取IC卡的授權證資料
  - CHTAttributeCertificate()
    - 不需任何參數，專門為非IC卡類所建立之建構子，若為IC卡中取出之授權證資料，請勿使用
    - 必需與函式decoderP7b搭配使用

解析與驗證授權證內容API 提供函式 – Java(3/8)

提供函式種類

- 驗證授權證(共3個函式，函式名稱verify)
  - 檢查授權證IssuerDN與正卡憑證SubjectDN是否一致
  - 檢查授權證HolderIssuer與附卡憑證IssuerDN是否一致
  - 檢查授權證HolderSN與附卡憑證序號是否一致
  - 正附卡憑證的SubjectDN是否一致
  - 正附卡憑證的IssuerDN是否一致
  - 檢查授權證的有效期限
  - 以正卡憑證對授權證中的簽章做驗證

解析與驗證授權證內容API 提供函式 – Java(4/8)

提供函式種類

- 非IC卡類專用(共2個函式，函式名稱decoderP7b、chknonIccardAuthCertStatus)
  - 解密非IC卡類所簽發的授權證
  - 檢查該非IC卡類所簽發的授權證有效狀態
- 其他(共1個函式，函式名稱getErrorMsg)
  - 取得錯誤原因
- 解析授權證(共17個函式)
  - 取得授權證各欄位資料
  - 欄位資料型態包含String/byte[ ]/BigInteger/Date

解析與驗證授權證內容API 提供函式 – Java(5/8)

解析授權證各函式

- java.lang.String getAccessIdentiferValueAt(int index)
  - 取得應用系統服務名稱，index為1-5
- java.lang.String getAffiliatedGroupNameValueAt(int index)
  - 取得使用單位名稱，index為1-5
- byte[] getAuthorityKeyIdentifier()
  - 取得授權金鑰識別元
- java.lang.String getAuthorizedIdentity()
  - 取得持卡者證號
- java.lang.String getGroupValueAt(int index)
  - 取得使用單位代碼，index為1-5
- java.lang.String getHolderIssuer()
  - 取得附卡的發行單位

解析與驗證授權證內容API 提供函式 – Java(6/8)

解析授權證各函式

- java.math.BigInteger getHolderSerialNumber()
  - 取得附卡的憑證序號
- java.lang.String getIssuerDn()
  - 取得附卡的授權者識別名稱
- java.lang.String getIssuerIssuer()
  - 取得附卡的授權者之發行單位
- java.math.BigInteger getIssuerSerial()
  - 取得附卡的授權者之憑證序號
- java.util.Date getNotAfter()
  - 取得有效期限之結束日期
- java.util.Date getNotBefore()
  - 取得有效期限之開始日期

解析與驗證授權證內容API 提供函式 – Java(7/8)

解析授權證各函式

- java.lang.String getRoleValueAt(int index)
  - 取得使用者職稱，index為1-5
- java.math.BigInteger getSerialNumber()
  - 取得授權證的憑證序號
- java.lang.String getSignatureAlgorithmID()
  - 取得授權證的數位簽章演算法之OID，例如：1.2.840.113549.1.1.5
- java.lang.String getSignatureID()
  - 取得授權證的數位簽章演算法之識別名稱，例如：SHA1withRSA
- byte[] getSignatureValue()
  - 取得授權證的數位簽章內容

解析與驗證授權證內容API 使用範例 – Java(8/8)

```
InputStream is = new BufferedInputStream(new ileInputStream("Test4SCAP02For??????.cer"));
X509Certificate SecCert = CertUtil.generateCert(is);取得附卡憑證
java.io.FileInputStream fis= new java.io.FileInputStream("MOEACA_附卡授權證_??????
    未加密.p7b");取得授權證測試資料
```

宣告屬性憑證格式

```
CHTAttributeCertificate ac = new CHTAttributeCertificate(fis);
```

```
ac.verify(SecCert); 驗證授權證(true or false)
```

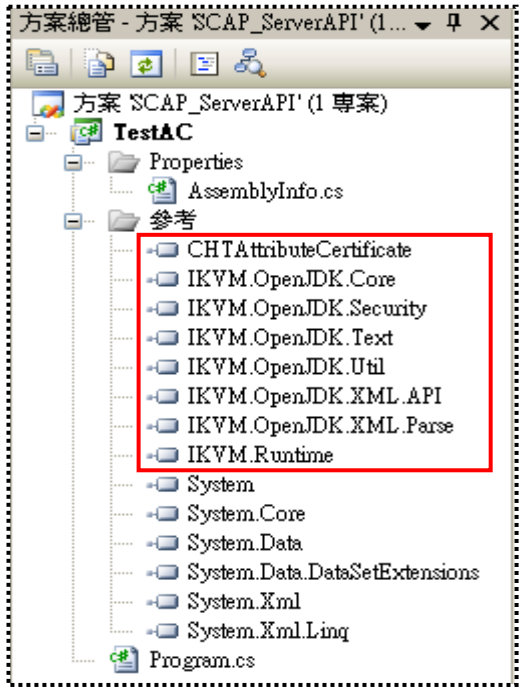
驗證授權證函式

```
ac.getAuthorizedIdentity(); 取得持卡者證號
for(int i=1;i<=5;i++){
        ac.getAccessIdentiferValueAt(i); 取得應用系統服務名稱
        ac.getAffiliatedGroupNameValueAt(i); 取得使用單位名稱
        ac.getRoleValueAt(i); 取得使用者角色
        ac.getGroupValueAt(i); 取得使用單位代碼
}
ac.getHolderIssuer(); 取得附卡的發行單位
ac.getHolderSerialNumber().toString(); 取得附卡的憑證序號
ac.getSerialNumber().toString(); 取得授權證的憑證序號
ac.getNotBefore(); 取得有效期限之開始日期
ac.getNotAfter(); 取得有效期限之結束日期
new java.math.BigInteger(1,ac.getAuthorityKeyIdentifier()).toString(16)); 取得授權金鑰識別元
ac.getIssuerDn(); 取得附卡的授權者識別名稱
```

解析授權證函式

解析與驗證授權證內容API 使用方法 – C# for ASP.NET

- 在參考中加入
  - CHTAttributeCertificate.dll
  - IKVM.OpenJDK.Core.dll
  - IKVM.OpenJDK.Security.dll
  - IKVM.OpenJDK.Text.dll
  - IKVM.OpenJDK.Util.dll
  - IKVM.OpenJDK.XML.API.dll
  - IKVM.OpenJDK.XML.Parse.dll
  - IKVM.Runtime.dll



- 將上列dll檔、QueryNonICCardStatusPath.ini檔與執行檔放置相同目錄
- 參考測試檔TestAC.cs

解析與驗證授權證內容API 使用範例 – C# for ASP.NET

```
//取得IC卡附卡憑證
java.io.InputStream isData = new java.io.BufferedInputStream(new java.io.FileInputStream("Test4SCAP01For000038.cer"));
java.security.cert.X509Certificate SecCert = tw.com.chttl.CertUtil.generateCert(isData);

//未加密的IC卡類P7B檔案
java.io.FileInputStream fis = new java.io.FileInputStream("MOEACA_附卡授權證_000038未加密.p7b");
tw.com.cht.ac.CHTAttributeCertificate acP7B = new tw.com.cht.ac.CHTAttributeCertificate(fis);

//驗簽章
if (acP7B.verify(SecCert) == false)
{
    Console.WriteLine("Verify ICCard = false" + "\r\n" + "ErrorMsg=" + acP7B.getErrorMsg());
}
else
{
    Console.WriteLine("Verify ICCard = true");
}

//解析IC卡類授權證各欄位
Console.WriteLine("identity=" + acP7B.getAuthorizedIdentity());
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine("AccessIdentifer" + i + "=" + acP7B.getAccessIdentiferValueAt(i));
    Console.WriteLine("AffiliatedGroupName" + i + "=" + acP7B.getAffiliatedGroupNameValueAt(i));
    Console.WriteLine("Role" + i + "=" + acP7B.getRoleValueAt(i));
    Console.WriteLine("Group" + i + "=" + acP7B.getGroupValueAt(i));
}
Console.WriteLine("Holder=" + acP7B.getHolderIssuer() + "," + acP7B.getHolderSerialNumber().toString());
Console.WriteLine("Serial=" + acP7B.getSerialNumber().toString());
Console.WriteLine("NotBefore=" + acP7B.getNotBefore());
Console.WriteLine("NotAfter=" + acP7B.getNotAfter());
Console.WriteLine("Authority key identifier=" + new java.math.BigInteger(1, acP7B.getAuthorityKeyIdentifier()).toString(16));
if (acP7B.getIssuerDn() != null)
    Console.WriteLine("IssuerDn=" + acP7B.getIssuerDn());
if (acP7B.getIssuerIssuer() != null)
    Console.WriteLine("Issuer's issuer=" + acP7B.getIssuerIssuer());
if (acP7B.getIssuerSerial() != null)
    Console.WriteLine("Issuer's SerialNumber=" + acP7B.getIssuerSerial().toString());
```

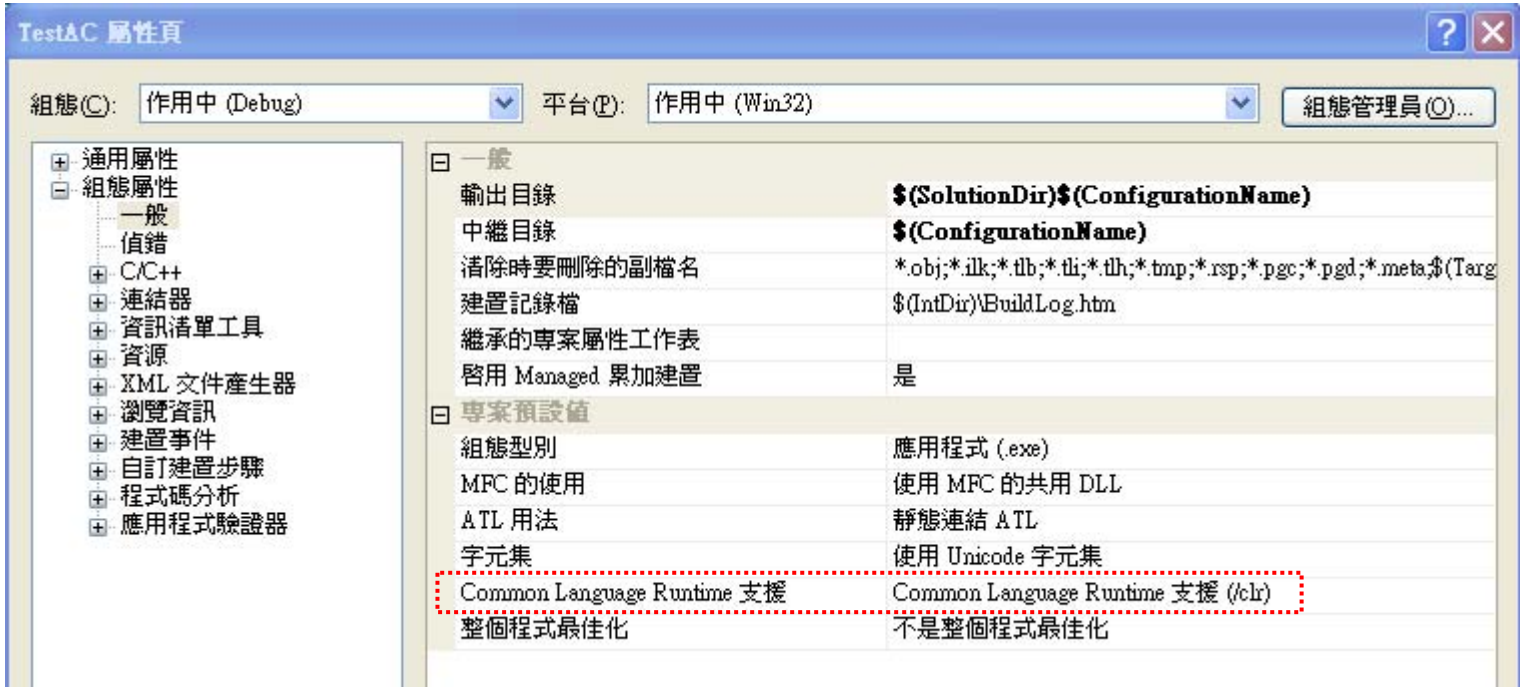
宣告屬性憑證格式

驗證授權證函式

解析授權證函式

解析與驗證授權證內容API 使用方法 – C/C++ for .NET(1/2)

- 在屬性→組態屬性→一般→Common Language Runtime欄位中，選擇『Common Language Runtime 支援 (/clr)』



解析與驗證授權證內容API 使用方法 – C/C++ for .NET(2/2)

- 使用動態函式庫

```
#using   ".\External\lib\CHTAttributeCertificate.dll"
#using   ".\External\lib\IKVM.OpenJDK.Core.dll"
#using   ".\External\lib\IKVM.OpenJDK.Security.dll"
#using   ".\External\lib\IKVM.OpenJDK.Text.dll"
#using   ".\External\lib\IKVM.OpenJDK.Util.dll"
#using   ".\External\lib\IKVM.OpenJDK.XML.API.dll"
#using   ".\External\lib\IKVM.OpenJDK.XML.Parse.dll"
#using   ".\External\lib\IKVM.Runtime.dll"
```

- 將上列dll檔、QueryNonICCardStatusPath.ini檔與執行檔放置相同目錄
- 參考測試檔TestAC.cpp

解析與驗證授權證內容API 使用範例 – C/C++ for .NET

```
//取得IC卡附卡憑證
java::io::InputStream ^isData = gcnew java::io::BufferedInputStream(gcnew java::io::FileInputStream("..\\Debug\\Test4SCAP01For000038.cer"));
java::security::cert::X509Certificate ^SecCert = tw::com::chttl::CertUtil::generateCert(isData);

//未加密的IC卡類P7B檔案
java::io::FileInputStream ^fis = gcnew java::io::FileInputStream("..\\Debug\\MOEACA_附卡授權證_000038未加密.p7b");
tw::com::cht::ac::CHTAttributeCertificate ^acP7B = gcnew tw::com::cht::ac::CHTAttributeCertificate(fis);

//驗簽章
if (acP7B->verify(SecCert) == false){
    printf("Verify ICCard = false\r\nErrorMsg=%s\n", acP7B->getErrorMsg());
}else{
    printf("Verify ICCard = true\n");
}

//解析IC卡類授權證各欄位
printf("identity=%s\n", acP7B->getAuthorizedIdentity());
for (int i = 1; i <= 5; i++){
    printf("AccessIdentifer%d=%s\n", i, acP7B->getAccessIdentiferValueAt(i));
    printf("AffiliatedGroupName%d=%s\n", i, acP7B->getAffiliatedGroupNameValueAt(i));
    printf("Role%d=%s\n", i, acP7B->getRoleValueAt(i));
    printf("Group%d=%s\n", i, acP7B->getGroupValueAt(i));
}

java::math::BigInteger ^HolderSerialNumber = acP7B->getHolderSerialNumber();
java::math::BigInteger ^SerialNumber = acP7B->getSerialNumber();
java::math::BigInteger ^AuthorityKeyIdentifier = gcnew java::math::BigInteger(1, acP7B->getAuthorityKeyIdentifier());
java::util::Date ^NotBefore = acP7B->getNotBefore();
java::util::Date ^NotAfter = acP7B->getNotAfter();
printf("Holder=%s,%s\n", acP7B->getHolderIssuer(), HolderSerialNumber->toString());
printf("Serial=%s\n", SerialNumber->toString());
printf("NotBefore=%s\n", NotBefore->toString());
printf("NotAfter=%s\n", NotAfter->toString());
printf("Authority key identifier=%s\n", AuthorityKeyIdentifier->toString(16));

sprintf(IssuerDn, "%s", acP7B->getIssuerDn());
sprintf(IssuerIssuer, "%s", acP7B->getIssuerIssuer());
sprintf(IssuerSerial, "%d", acP7B->getIssuerSerial());
if(strcmp(IssuerDn, "(null)") != 0)
    printf("IssuerDn=%s\n", acP7B->getIssuerDn());
if(strcmp(IssuerIssuer, "(null)") != 0)
    printf("Issuer's issuer=%s\n", acP7B->getIssuerIssuer());
if(strcmp(IssuerSerial, "0") != 0){
    java::math::BigInteger ^l_biIssuerSerial = acP7B->getSerialNumber();
    printf("Issuer's SerialNumber=%s\n", l_biIssuerSerial->toString());
}
```

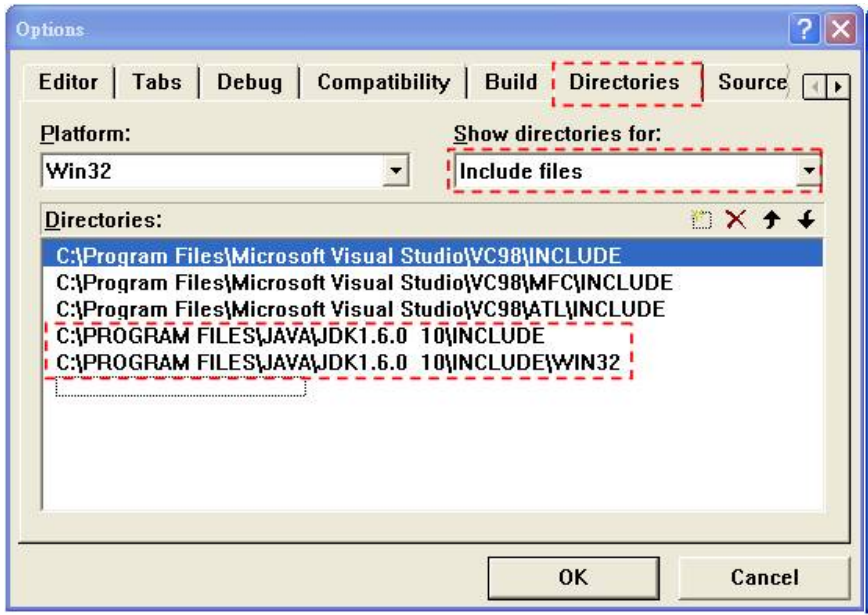
宣告屬性憑證格式

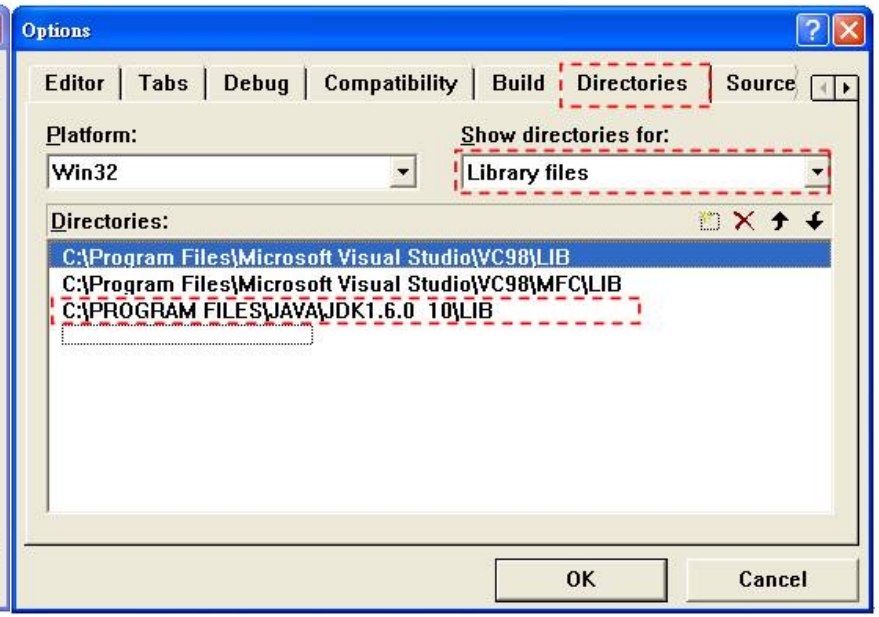
驗證授權證函式

解析授權證函式

解析與驗證授權證內容API 使用方法 – C/C++ for Visual C++(1/6)

- 開發環境要先安裝JDK，執行環境僅需安裝JRE，1.4版以上較佳
- Visual C++ 6.0於Tools→Options選項中添加JDK相關的目錄





解析與驗證授權證內容API 使用方法 – C/C++ for Visual C++(2/6)

- 將Java的函式庫(*.jar或*.class)放置於系統可載入的位置，這裡CHTAttributeCertificate.jar放置於與執行檔(*.exe)相同目錄
- 將下列檔案與執行檔放置相同目錄
  - MFC42D.DLL、MSVCP60D.DLL、MSVCRT.DLL、MSVCRTD.DLL、QueryNonICCardStatusPath.ini
- 表頭定義

```
#include <jni.h>
#include <string>
#include <iostream>
```

- 參考測試檔TestAC.cpp

解析與驗證授權證內容API 使用方法 – C/C++ for Visual C++(3/6)

宣告JVM相關變數並設置初始化

```
//宣告啓動JVM相關變數
typedef jint (WINAPI *PFunCreateJavaVM)(JavaVM **, void **, void *);  //定義一個函數指針，下面用來指向JVM中的JNI_CreateJavaVM函數
int res;
JavaVMInitArgs vm_args;
JavaVMOption options[3];
JavaVM *jvm;
JNIEnv *env;

/*設置初始化參數*/
options[0].optionString = "-Djava.compiler=NONE";  //不知道幹麻的
options[1].optionString = "-Djava.class.path=.;c:\\;.\\CHTAttributeCertificate.jar";  //設置classpath，如果程序用到了第三方的JAR包，也可以在這
options[2].optionString = "-verbose:NONE";  //該參數可以用來觀察C++調用JAVA的過程，設置該參數后，程序會在標准輸出設備上打印調用的相關信息

//設置版本號，版本號有JNI_VERSION_1_1，JNI_VERSION_1_2和JNI_VERSION_1_4...
//選擇一個跟你安裝的JRE版本最近的版本號即可，不過你的JRE版本一定要等於或者高于指定的版本號
vm_args.version = JNI_VERSION_1_4;
vm_args.nOptions = 3;
vm_args.options = options;
vm_args.ignoreUnrecognized = JNI_TRUE;  //該參數指定是否忽略非標准的參數，如果填JNI_FLASE，當遇到非標准參數時，JNI_CreateJavaVM會返回JNI_ERR
```

用來指定*.jar檔的位置

從註冊表中讀出JRE安裝路徑並找出jvm.dll位置

```
//從註冊表中讀出JRE安裝路徑並找出jvm.dll位置
RegResult = RegOpenKey(HKEY_LOCAL_MACHINE, "SOFTWARE\\JavaSoft\\Java Development Kit\\1.6", &hKey);
RegResult = RegQueryValueEx(hKey, "JavaHome", NULL, &dwType, (LPBYTE)sz, &sl);
l_csJavaHome.Format("%s\\jre\\bin\\client\\jvm.dll", sz);
```

解析與驗證授權證內容API 使用方法 – C/C++ for Visual C++(4/6)

- 載入JVM.DLL動態庫並創建Java虛擬機

```
//加載JVM.DLL動態庫
HINSTANCE hInstance = ::LoadLibrary(l_csJavaHome.GetBuffer(0));

if (hInstance == NULL)
{
    printf("加載JVM.DLL動態庫失敗!!\n");
    system("pause");
    return false;
}
printf("加載JVM.DLL動態庫成功!!\n");

//取得里面的JNI_CreateJavaVM函數指針
PFunCreateJavaVM funCreateJavaVM = (PFunCreateJavaVM)::GetProcAddress(hInstance, "JNI_CreateJavaVM");

//調用JNI_CreateJavaVM創建虛擬機
res = (*funCreateJavaVM)(&jvm, (void**)&env, &vm_args);
```

- 查找類別，Java若為java.math.BigInteger這裡要改為java/math/BigInteger

```
jclass cls_BigInteger = env->FindClass("java/math/BigInteger");
```

解析與驗證授權證內容API 使用方法 – C/C++ for Visual C++(5/6)

- 獲取類別中的方法，最后一個參數是方法的簽名，通過javap -s -p 文件名可以獲得

```
jmethodID mid_BigInteger = env->GetMethodID(cls_BigInteger, "<init>", "(I[B)V");
jmethodID mid_BigIntegerToString = env->GetMethodID(cls_BigInteger, "toString", "()Ljava/lang/String;");
jmethodID mid_BigIntegerToStringInt = env->GetMethodID(cls_BigInteger, "toString", "(I)Ljava/lang/String;");
```

- 使用範例：

```
//==================================ICCard類P7B檔案=====================================//
printf("\nIC卡類：\n");
//取得IC卡附卡憑證(從檔案)
const char szTest[] = "Test4SCAP01For000038.cer";
jstring arg = NewJString(env, szTest);
jobject FIS = env->NewObject(cls_FileInputStream, mid_FileInputStream, arg);  //建立物件
jobject BIS = env->NewObject(cls_BufferedInputStream, mid_BufferedInputStream, FIS);
jobject SecCert = env->CallObjectMethod(cls_CertUtil, mid_CertUtil, BIS);

//未加密的IC卡類P7B檔案(從檔案)
const char szTest1[] = "MOEACA_附卡授權證_000038未加密.p7b";
jstring arg1 = NewJString(env, szTest1);
jobject FIS1 = env->NewObject(cls_FileInputStream, mid_FileInputStream, arg1);
jobject acP7B = env->NewObject(cls_CHTAttributeCertificate, mid_CHTAttributeCertificate, FIS1);

//驗簽章
jstring msg = (jstring)env->CallObjectMethod(acP7B, mid_Verify, SecCert);
printf("Verify=");cout<<env->CallObjectMethod(acP7B, mid_Verify, SecCert);printf(" (true:00000001,false:00000000)\n");
msg = (jstring)env->CallObjectMethod(acP7B, mid_getErrorMsg);
printf("(Verify ICCard)ErrorMsg=");cout<<JStringToCString(env, msg);printf("\n");
```

宣告屬性憑證格式

驗證授權證函式

解析與驗證授權證內容API 使用方法 – C/C++ for Visual C++(6/6)

```
//解析IC卡類授權證各欄位
msg = (jstring)env->CallObjectMethod(acP7B, mid_getAuthorizedIdentity);
printf("identity=");cout<<JStringToCString(env, msg);printf("\n");
for(int i=1;i<=5;i++){
    msg = (jstring)env->CallObjectMethod(acP7B, mid_getAccessIdentifer, i);
    printf("AccessIdentifer%d=", i);cout<<JStringToCString(env, msg);printf("\n");
    msg = (jstring)env->CallObjectMethod(acP7B, mid_getAffiliatedGroupName, i);
    printf("AffiliatedGroupName%d=", i);cout<<JStringToCString(env, msg);printf("\n");
    msg = (jstring)env->CallObjectMethod(acP7B, mid_getRole, i);
    printf("Role%d=", i);cout<<JStringToCString(env, msg);printf("\n");
    msg = (jstring)env->CallObjectMethod(acP7B, mid_getGroup, i);
    printf("Group%d=", i);cout<<JStringToCString(env, msg);printf("\n");
}
msg = (jstring)env->CallObjectMethod(acP7B, mid_getHolderIssuer);
printf("Holder=");cout<<JStringToCString(env, msg);
jobject obj_getHolderSerialNumber = env->CallObjectMethod(acP7B, mid_getHolderSerialNumber);
msg = (jstring)env->CallObjectMethod(obj_getHolderSerialNumber, mid_BigIntegerToString);
printf(",");cout<<JStringToCString(env, msg);printf("\n");
jobject obj_getSerialNumber = env->CallObjectMethod(acP7B, mid_getSerialNumber);
msg = (jstring)env->CallObjectMethod(obj_getSerialNumber, mid_BigIntegerToString);
printf("Serial=");cout<<JStringToCString(env, msg);printf("\n");
jobject obj_getNotBefore = env->CallObjectMethod(acP7B, mid_getNotBefore);
msg = (jstring)env->CallObjectMethod(obj_getNotBefore, mid_DateToString);
printf("NotBefore=");cout<<JStringToCString(env, msg);printf("\n");
jobject obj_getNotAfter = env->CallObjectMethod(acP7B, mid_getNotAfter);
msg = (jstring)env->CallObjectMethod(obj_getNotAfter, mid_DateToString);
printf("NotAfter=");cout<<JStringToCString(env, msg);printf("\n");
jobject obj_mid_getAuthorityKeyIdentifier = env->CallObjectMethod(acP7B, mid_getAuthorityKeyIdentifier);
jobject AKI = env->NewObject(cls_BigInteger, mid_BigInteger, 1, obj_mid_getAuthorityKeyIdentifier);
msg = (jstring)env->CallObjectMethod(AKI, mid_BigIntegerToStringInt, 16);
printf("Authority key identifier=");cout<<JStringToCString(env, msg);printf("\n");
msg = (jstring)env->CallObjectMethod(acP7B, mid_getIssuerDn);
printf("IssuerDn=");cout<<JStringToCString(env, msg);printf("\n");
msg = (jstring)env->CallObjectMethod(acP7B, mid_getIssuerIssuer);
printf("Issuer's issuer=");cout<<JStringToCString(env, msg);printf("\n");
jobject obj_getIssuerSerial = env->CallObjectMethod(acP7B, mid_getIssuerSerial);
msg = (jstring)env->CallObjectMethod(obj_getIssuerSerial, mid_BigIntegerToString);
printf("Issuer's SerialNumber=");cout<<JStringToCString(env, msg);printf("\n");
```

解析授權證函式

聯絡方式：

- API、測試套件取得及其他問題—
  - 黃可婷　tiffany0@cht.com.tw
  - 鄭惇方　tfcheng@cht.com.tw
- API使用上問題—
  - 服務時間：週一至週五（國定列假日除外）上班時間上午9點至下午6點
  - 服務電話：02-2192-2918
  - e-mail信箱：hisecure6@cht.com.tw

附錄

讀取IC晶片卡中的授權證API提供函式 – Java(1/5)

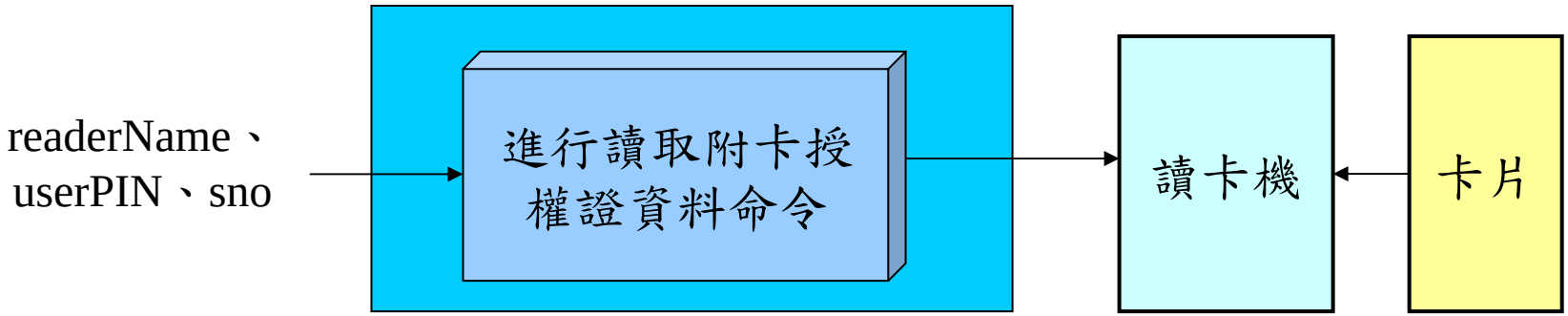
使用時機：進行讀取工商憑證附卡IC卡中授權證空間資料命令

```
public int readUserData(
        java.lang.String readerName,
        java.lang.String userPIN,
        java.lang.String sno)
```

讀卡機名稱，若為null則自動取得

IC卡PIN碼

應用服務代碼：0001



注意：要先ICcard.readUserData( )再ICcard.getUserData( )

讀取IC晶片卡中的授權證API提供函式 – Java(2/5)

readUserData回傳值說明

錯誤碼	錯誤訊息	錯誤訊息中文	錯誤解釋
0	OK	正確執行	可順利讀出授權證
-1		參數設定錯誤	請確認使用的函式參數型態資料是否與函式定義所需輸入的參數型態資料相同
29441	E_NOT_LOAD_DLL	沒有載入DLL	Client端API所需要的DLL找不到，可能是沒有將DLL放在可載入的路徑上(Java)，或是DLL沒有和執行檔放在相同目錄(C語言)，使用COM元件應不會出現此錯誤
29442	E_NOT_SUPPORT_FUNCTION	沒有支援的函式	可能在開發過程中將支援的函式名稱寫錯，造成找不到函式
29443	E_SLOT	讀取不到卡片	需請用戶將檢查是否有將卡片插好，若仍然讀不到，先試試將IC卡晶片用橡皮擦擦一擦，若還是不行有可能是卡片損壞
28929	READER_NOT_SELECT_ERROR	找不到讀卡機錯誤	請用戶檢查讀卡機是否有正確連接，並有安裝驅動程式，另外，並建議一台電腦只連接一台讀卡機，連接太多台容易造成錯誤

讀取IC晶片卡中的授權證API提供函式 – Java(3/5)

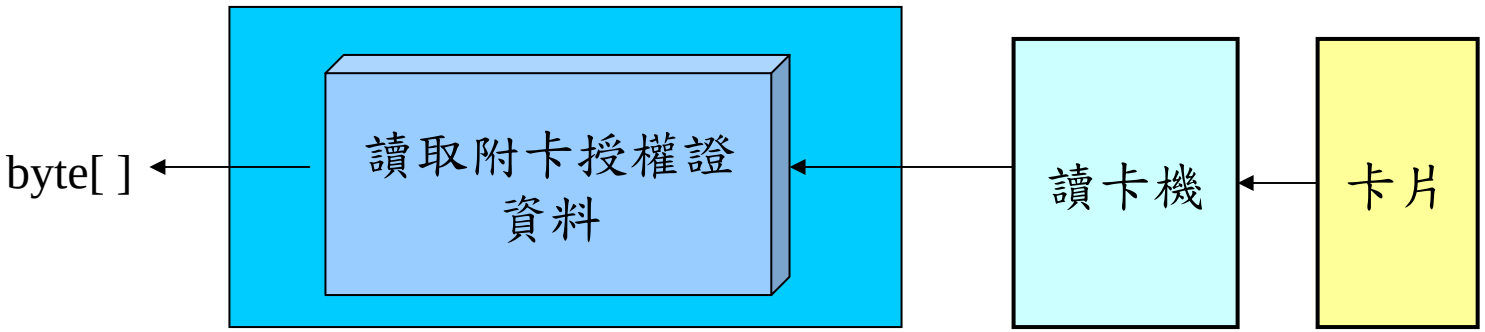
readUserData回傳值說明

錯誤碼	錯誤訊息	錯誤訊息中文	錯誤解釋
29697	BUFFER_TO_SMALL	宣告的buffer太小	程式開發過程中，要取出授權證資料時所宣告的buffer太小或不爲陣列，則會出現此錯誤
29713	SNO_EXIST	應用服務代碼存在	用以判斷服務代碼是否存在，有些情形需要存在，有些時候需不存在，該錯誤碼一般不會出現(使用讀取授權證函式時不會出現，可以忽略)
29714	SNO_NO_EXIST	應用服務代碼不存在	表示該卡片未開檔，或已開檔但應用服務代碼錯誤(程式中的SNO寫錯，SNO在程式中都是固定的，授權證爲服務代碼爲0001，發生這種情形的機率很低)，因此一般發生此問題都是尚未開檔
29715	INDEX_FORMAT_ERROR	動態區索引格式錯誤	表示該卡片屬性憑證區已損毀，無法再授權，需送回卡廠檢查，此錯誤一般不會出現
29716	CARDSPACE_NOT_ENOUGH	卡片動態區空間不足	屬性憑證區的空間大小是有限的，若要寫入的資料超過可容許之大小，則會出現此錯誤訊息，不過因爲提供給各AP所使用之API並無寫卡權限，故此錯誤碼並不會出現(使用讀取授權證函式時不會出現，可以忽略)
29717	DATA_TO_LARGE	資料太大	寫卡時要寫入的資料太大，但因提供給各AP所使用之API並無寫卡權限，故此錯誤碼並不會出現(使用讀取授權證函式時不會出現，可以忽略)

讀取IC晶片卡中的授權證API提供函式 – Java(4/5)

使用時機：取得工商憑證附卡IC卡中授權證空間資料

```
public byte[ ] getUserData( )
```



注意：要先ICcard.readUserData( )再ICcard.getUserData( )

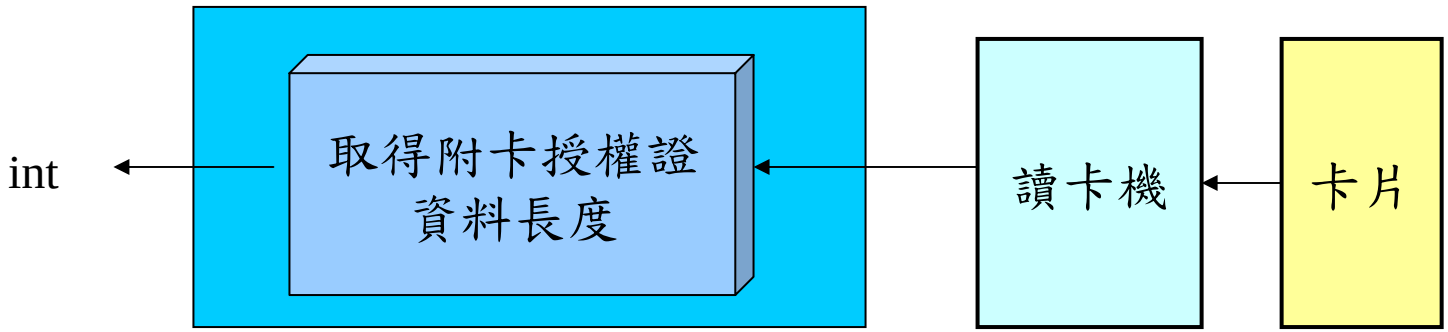
<回傳值>

回傳IC卡內動態空間內的資料

讀取IC晶片卡中的授權證API提供函式 – Java(5/5)

- 使用時機：取得工商憑證附卡IC卡中授權證空間資料長度

```
public int getUserDataLen( )
```



<回傳值>

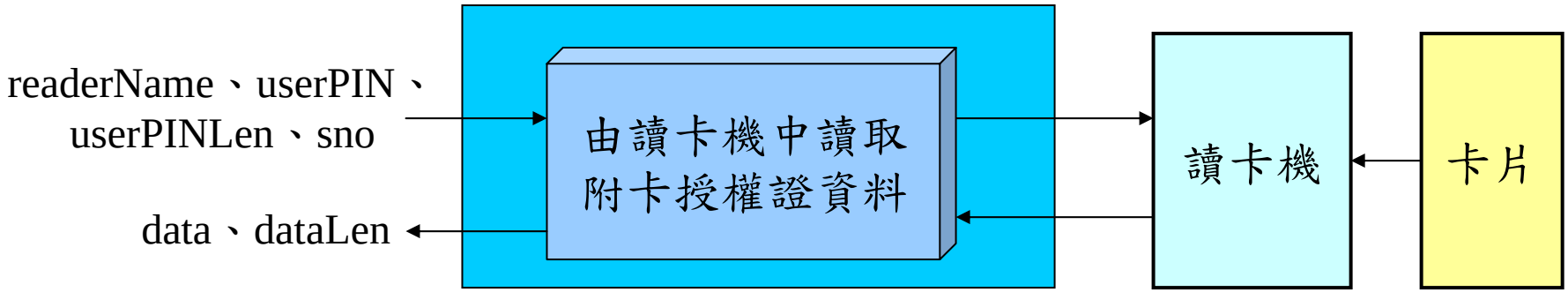
回傳IC卡內動態空間內的資料長度

讀取IC晶片卡中的授權證API提供函式－C語言(1/3)

- 使用時機：欲從工商憑證附卡IC卡中取出授權證資料時使用

```
int readUserData( unsigned char   *readerName,
                  unsigned char   *userPIN,
                  int             userPINLen,
                  unsigned char   *sno,
                  unsigned char   *data,
                  int             *dataLen);
```

- *readerName：讀卡機名稱，若為null則自動取得
- *userPIN：IC卡PIN碼
- userPINLen：IC卡PIN碼長度
- *sno：應用服務代碼：0001
- *data：回傳的資料
- *dataLen：回傳的資料長度



讀取IC晶片卡中的授權證API提供函式－C語言(2/3)

readUserData回傳值說明

錯誤碼	錯誤訊息	錯誤訊息中文	錯誤解釋
0	OK	正確執行	可順利讀出授權證
-1		參數設定錯誤	請確認使用的函式參數型態資料是否與函式定義所需輸入的參數型態資料相同
29441	E_NOT_LOAD_DLL	沒有載入DLL	Client端API所需要的DLL找不到，可能是沒有將DLL放在可載入的路徑上(Java)，或是DLL沒有和執行檔放在相同目錄(C語言)，使用COM元件應不會出現此錯誤
29442	E_NOT_SUPPORT_FUNCTION	沒有支援的函式	可能在開發過程中將支援的函式名稱寫錯，造成找不到函式
29443	E_SLOT	讀取不到卡片	需請用戶將檢查是否有將卡片插好，若仍然讀不到，先試試將IC卡晶片用橡皮擦擦一擦，若還是不行有可能是卡片損壞
28929	READER_NOT_SELECT_ERROR	找不到讀卡機錯誤	請用戶檢查讀卡機是否有正確連接，並有安裝驅動程式，另外，並建議一台電腦只連接一台讀卡機，連接太多台容易造成錯誤

讀取IC晶片卡中的授權證API提供函式－C語言(3/3)

readUserData回傳值說明

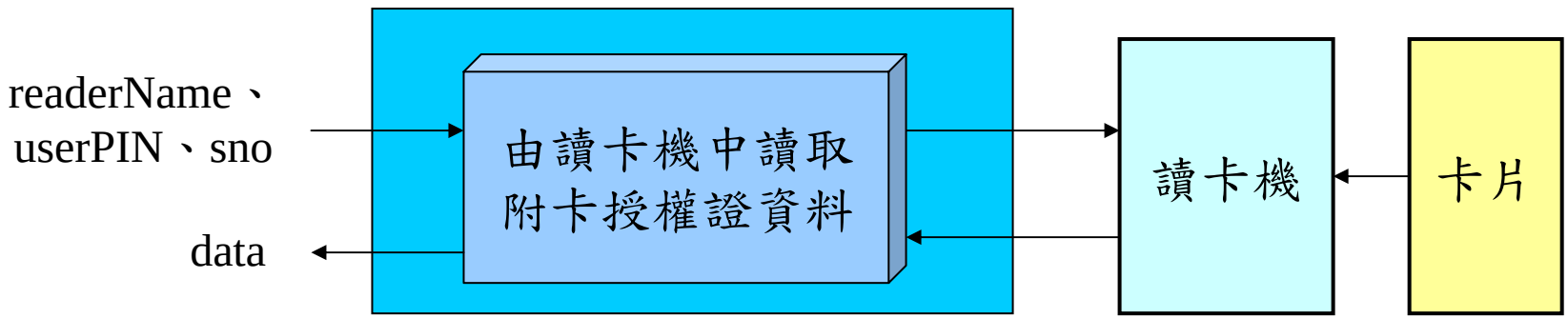
錯誤碼	錯誤訊息	錯誤訊息中文	錯誤解釋
29697	BUFFER_TO_SMALL	宣告的buffer太小	程式開發過程中，要取出授權證資料時所宣告的buffer太小或不爲陣列，則會出現此錯誤
29713	SNO_EXIST	應用服務代碼存在	用以判斷服務代碼是否存在，有些情形需要存在，有些時候需不存在，該錯誤碼一般不會出現(使用讀取授權證函式時不會出現，可以忽略)
29714	SNO_NO_EXIST	應用服務代碼不存在	表示該卡片未開檔，或已開檔但應用服務代碼錯誤(程式中的SNO寫錯，SNO在程式中都是固定的，授權證爲服務代碼爲0001，發生這種情形的機率很低)，因此一般發生此問題都是尚未開檔
29715	INDEX_FORMAT_ERROR	動態區索引格式錯誤	表示該卡片屬性憑證區已損毀，無法再授權，需送回卡廠檢查，此錯誤一般不會出現
29716	CARDSPACE_NOT_ENOUGH	卡片動態區空間不足	屬性憑證區的空間大小是有限的，若要寫入的資料超過可容許之大小，則會出現此錯誤訊息，不過因爲提供給各AP所使用之API並無寫卡權限，故此錯誤碼並不會出現(使用讀取授權證函式時不會出現，可以忽略)
29717	DATA_TO_LARGE	資料太大	寫卡時要寫入的資料太大，但因提供給各AP所使用之API並無寫卡權限，故此錯誤碼並不會出現(使用讀取授權證函式時不會出現，可以忽略)

讀取IC晶片卡中的授權證API提供函式 – COM元件(1/3)

- 使用時機：欲從工商憑證附卡IC卡中取出授權證資料時使用

```
int readUserData( VARIANT   readerName,
                  VARIANT   userPIN,
                  VARIANT   sno,
                  VARIANT   *data);
```

- readerName：讀卡機名稱，若為null則自動取得
- userPIN：IC卡PIN碼
- sno：應用服務代碼：0001
- data：回傳的資料



讀取IC晶片卡中的授權證API提供函式 – COM元件(2/3)

readUserData回傳值說明

錯誤碼	錯誤訊息	錯誤訊息中文	錯誤解釋
0	OK	正確執行	可順利讀出授權證
-1		參數設定錯誤	請確認使用的函式參數型態資料是否與函式定義所需輸入的參數型態資料相同
29441	E_NOT_LOAD_DLL	沒有載入DLL	Client端API所需要的DLL找不到，可能是沒有將DLL放在可載入的路徑上(Java)，或是DLL沒有和執行檔放在相同目錄(C語言)，使用COM元件應不會出現此錯誤
29442	E_NOT_SUPPORT_FUNCTION	沒有支援的函式	可能在開發過程中將支援的函式名稱寫錯，造成找不到函式
29443	E_SLOT	讀取不到卡片	需請用戶將檢查是否有將卡片插好，若仍然讀不到，先試試將IC卡晶片用橡皮擦擦一擦，若還是不行有可能是卡片損壞
28929	READER_NOT_SELECT_ERROR	找不到讀卡機錯誤	請用戶檢查讀卡機是否有正確連接，並有安裝驅動程式，另外，並建議一台電腦只連接一台讀卡機，連接太多台容易造成錯誤

讀取IC晶片卡中的授權證API提供函式－COM元件(3/3)

readUserData回傳值說明

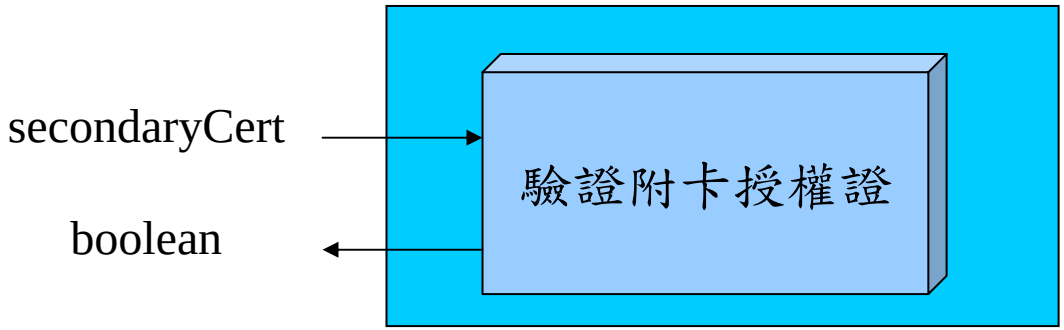
錯誤碼	錯誤訊息	錯誤訊息中文	錯誤解釋
29697	BUFFER_TO_SMALL	宣告的buffer太小	程式開發過程中，要取出授權證資料時所宣告的buffer太小或不爲陣列，則會出現此錯誤
29713	SNO_EXIST	應用服務代碼存在	用以判斷服務代碼是否存在，有些情形需要存在，有些時候需不存在，該錯誤碼一般不會出現(使用讀取授權證函式時不會出現，可以忽略)
29714	SNO_NO_EXIST	應用服務代碼不存在	表示該卡片未開檔，或已開檔但應用服務代碼錯誤(程式中的SNO寫錯，SNO在程式中都是固定的，授權證爲服務代碼爲0001，發生這種情形的機率很低)，因此一般發生此問題都是尚未開檔
29715	INDEX_FORMAT_ERROR	動態區索引格式錯誤	表示該卡片屬性憑證區已損毀，無法再授權，需送回卡廠檢查，此錯誤一般不會出現
29716	CARDSPACE_NOT_ENOUGH	卡片動態區空間不足	屬性憑證區的空間大小是有限的，若要寫入的資料超過可容許之大小，則會出現此錯誤訊息，不過因爲提供給各AP所使用之API並無寫卡權限，故此錯誤碼並不會出現(使用讀取授權證函式時不會出現，可以忽略)
29717	DATA_TO_LARGE	資料太大	寫卡時要寫入的資料太大，但因提供給各AP所使用之API並無寫卡權限，故此錯誤碼並不會出現(使用讀取授權證函式時不會出現，可以忽略)

解析與驗證授權證內容API 提供函式 – Java(1/10)

- 種類：驗證授權證
- 使用時機：驗證工商憑證附卡授權證資料

```
public boolean verify(
                java.lang.String secondaryCert)
```

hex String格式的附卡憑證



<回傳值>

True 驗證成功

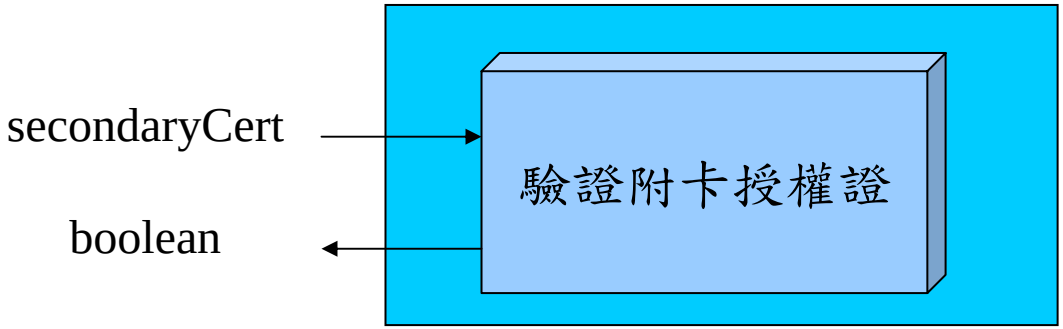
False 驗證失敗

解析與驗證授權證內容API提供函式 – Java(2/10)

- 種類：驗證授權證
- 使用時機：驗證工商憑證附卡授權證資料

```
public boolean verify(
                java.security.cert.X509Certificate secondaryCert)
```

X.509格式的附卡憑證



<回傳值>

True 驗證成功

False 驗證失敗

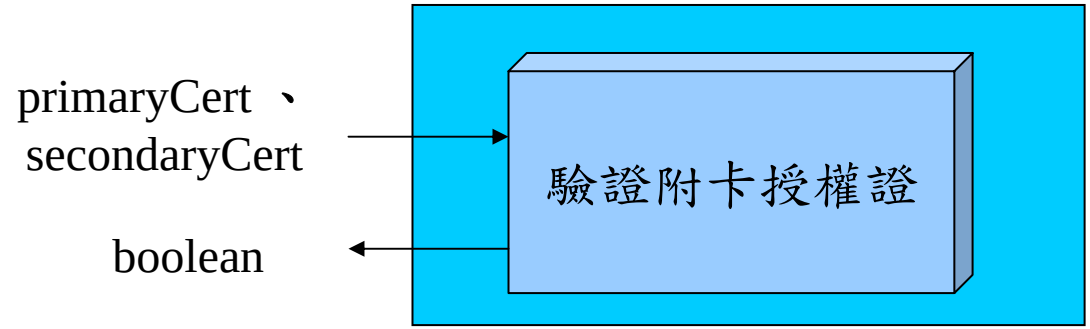
解析與驗證授權證內容API 提供函式 – Java(3/10)

- 種類：驗證授權證
- 使用時機：驗證工商憑證附卡授權證資料

```
public boolean verify(
        java.security.cert.X509Certificate primaryCert,
        java.security.cert.X509Certificate secondaryCert)
```

X.509格式的正卡憑證

X.509格式的附卡憑證



<回傳值>

True 驗證成功

False 驗證失敗

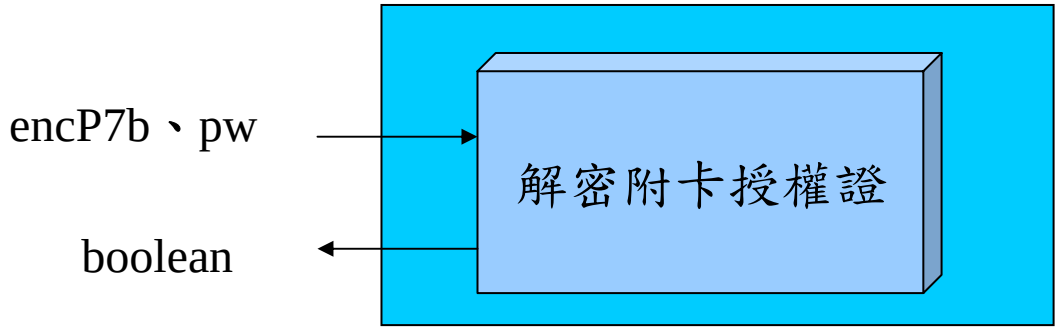
解析與驗證授權證內容API
提供函式 – Java(4/10)

- 種類：非IC卡類專用
- 使用時機：解密非IC卡類工商憑證附卡授權證資料

```
public boolean decoderP7b (
        java.io.InputStream encP7b,
        java.lang.String    pw)
```

待解密授權證資料

授權證用戶安控密碼



<回傳值>

True 解密成功

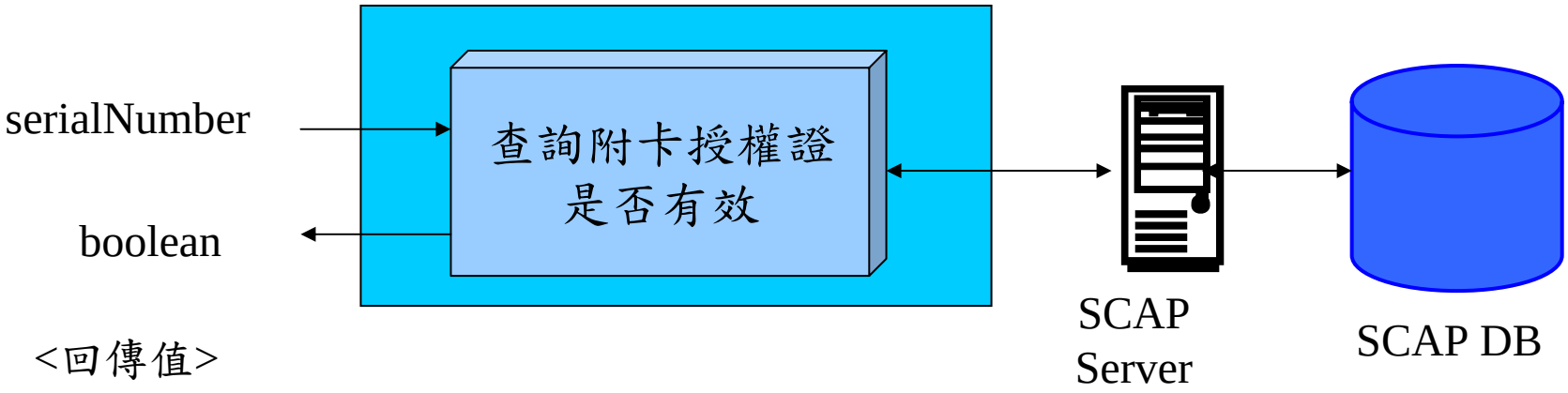
False 解密失敗

解析與驗證授權證內容API
提供函式 – Java(5/10)

- 種類：非IC卡類專用
- 使用時機：查詢非IC卡類工商憑證附卡授權證有效狀態

```
public boolean chknonIccardAuthCertStatus (
                    java.math.BigInteger serialNumber)
```

要查詢的授權證序號



<回傳值>

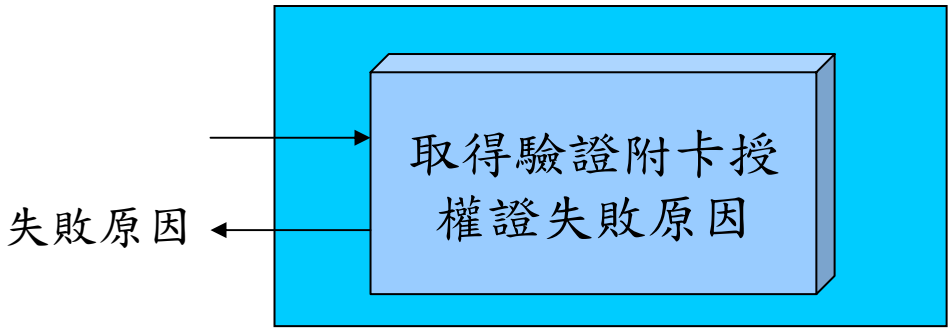
True 查詢成功且授權證有效

False 查詢失敗或授權證已無效

解析與驗證授權證內容API 提供函式 – Java(6/10)

- 種類：其他
- 使用時機：工商憑證附卡授權證資料驗證失敗時取得失敗原因，或非IC卡類附卡簽發的授權證解密或檢查狀態失敗時取得失敗原因

```
public String getErrorMsg( )
```



<回傳值>

Null 沒有失敗原因

各種失敗原因

解析與驗證授權證內容API提供函式 – Java(7/10)

驗證授權證失敗時取得錯誤訊息

<失敗原因>

- 授權證IssuerSerial與正卡憑證SubjectDN不一致
- 授權證HolderIssuer與附卡憑證IssuerDN不一致
- 授權證HolderSN與附卡憑證序號不一致
- 正附卡憑證的SubjectDN不一致
- 正附卡憑證的IssuerDN不一致
- 授權證已過期
- 以正卡憑證對授權證中的簽章做驗證不通過

解析與驗證授權證內容API提供函式 – Java(8/10)

❑ 解密非IC卡類授權證失敗錯誤訊息

<失敗原因>

- ◆密碼轉換格式錯誤
- ◆不存在此種演算法
- ◆解密密碼格式錯誤
- ◆解密參數設定錯誤
- ◆無法取得待解密資料
- ◆取得待解密資料錯誤
- ◆解密資料錯誤
- ◆解密資料格式錯誤

解析與驗證授權證內容API
提供函式 – Java(9/10)

❑ 檢查非IC卡類授權證失敗錯誤訊息

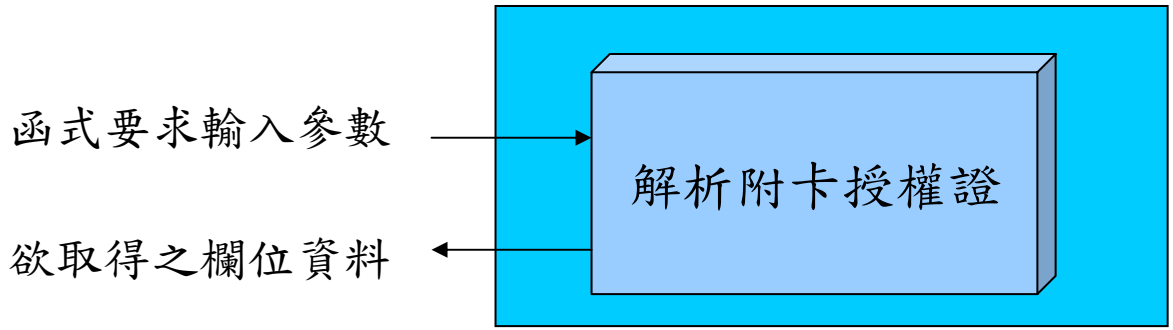
<失敗原因>

- ◆讀取ini檔案錯誤
- ◆找不到ini檔案
- ◆網路連線錯誤
- ◆URL錯誤
- ◆授權證簽屬成功但未存檔
- ◆授權證存檔失敗
- ◆授權證已自動無效
- ◆授權證已停用
- ◆查詢非IC卡類授權證狀態錯誤:無收到來源上傳資料
- ◆查詢非IC卡類授權證狀態錯誤:無收到Web端憑證資料
- ◆查詢非IC卡類授權證狀態錯誤:無法連接資料庫
- ◆查詢非IC卡類授權證狀態錯誤:於Server端依授權證序號取對應欄位資料失敗
- ◆查詢非IC卡類授權證狀態錯誤:記憶體配置錯誤查詢非IC卡類授權證狀態錯誤

解析與驗證授權證內容API提供函式 – Java(10/10)

- 種類：解析授權證
- 使用時機：解析工商憑證附卡授權證各欄位資料

```
public String/byte[ ]/BigInteger/Date 函式名稱(
        函式要求輸入參數)
```



<回傳值>

成功取回各欄位資料或取回Null

解析與驗證授權證內容API 使用範例 – Java

取得附卡憑證(X509Certificate **SecCert**)

附卡憑證由檔案讀取

```
InputStream is = new BufferedInputStream(new
    FileInputStream("Test4SCAP02For??????.cer"));
X509Certificate SecCert = CertUtil.generateCert(is);
```

讀檔案

附卡憑證由IC卡取得(使用HiSecure)

```
X509Certificate SecCert=null ;
Module.initialize();
Module module = Module.getInstance();
try {
        Token tok = module.getToken(module.getTokens()[0]);
        SecCert = tok.getCert(Token.ID_SIGN);
        tok.logout();
}catch (Exception ex) {
}
```

讀IC卡

解析與驗證授權證內容API 使用範例 – Java

取得授權證測試資料與宣告(CHTAttributeCertificate **ac**)

- 授權證測試資料為一P7B格式檔案

讀檔案

```
java.io.FileInputStream fis= new java.io.FileInputStream("MOEACA_附卡授權證_??????未加密.p7b");
CHTAttributeCertificate ac = new CHTAttributeCertificate(fis);
```

- 授權證測試資料由讀取IC卡取得

讀IC卡

```
ICCDFCLIENT ICcard = new ICCDFCLIENT();
ICcard.readUserData(null, PINCode, sno);
CHTAttributeCertificate ac = new  CHTAttributeCertificate(ICcard.getUserData());
```

- 授權證測試資料為非IC卡類簽發之加密檔案

非IC卡類

```
java.io.FileInputStream fis= new java.io.FileInputStream("MOEACA_附卡授權證_??????已加密.sec");
CHTAttributeCertificate ac = new CHTAttributeCertificate();
ac.decoderP7b(fis, pw);解密授權證(true or false)
ac.chknonIccardAuthCertStatus(ac.getSerialNumber());查詢有效性(true or false)
```

解析與驗證授權證內容API
使用範例 – Java

驗證與解析範例

```
ac.verify(SecCert); 驗證授權證(true or false)
```

→ 驗證授權證函式

```
ac.getAuthorizedIdentity(); 取得持卡者證號
for(int i=1;i<=5;i++){
        ac.getAccessIdentiferValueAt(i); 取得應用系統服務名稱
        ac.getAffiliatedGroupNameValueAt(i); 取得使用單位名稱
        ac.getRoleValueAt(i); 取得使用者角色
        ac.getGroupValueAt(i); 取得使用單位代碼
}
ac.getHolderIssuer(); 取得附卡的發行單位
ac.getHolderSerialNumber().toString(); 取得附卡的憑證序號
ac.getSerialNumber().toString(); 取得授權證的憑證序號
ac.getNotBefore(); 取得有效期限之開始日期
ac.getNotAfter(); 取得有效期限之結束日期
new java.math.BigInteger(1,ac.getAuthorityKeyIdentifier()).toString(16)); 取得授權
    金鑰識別元
ac.getIssuerDn(); 取得附卡的授權者識別名稱
ac.getIssuerIssuer(); 取得附卡的授權者之發行單位
ac.getIssuerSerial().toString(); 取得附卡的授權者之憑證序號
```

→ 解析授權證函式