

發明專利說明書

(本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※ 申請案號：93126908

※ 申請日期：2004 年 9 月 6 日

※IPC 分類：

G06F 17/00 (2006.01)

G06F 19/00 (2011.01)

G06F 21/00 (2013.01)

一、發明名稱：(中文/英文)

用於軟體授權的系統、方法及電腦可讀取媒體

SYSTEM, METHOD, AND COMPUTER-REDABLE MEDIUM FOR  
SOFTWARE LICENSING

二、申請人：(共 1 人)

姓名或名稱：(中文/英文)

美商·微軟公司

Microsoft Corporation

代表人：(中文/英文)

艾華那諾爾 D 巴特萊

EPPENAUER, D. BARTLEY

住居所或營業所地址：(中文/英文)

美國華盛頓州列德蒙微軟路 1 號

One Microsoft Way, Building 8, Redmond, WA 98052-6399, U.S.A.

國籍：(中文/英文)

美國/USA

三、發明人：(共 3 人)

姓名：(中文/英文)

1. 古亞克堤卡格拉/GUNYAKTI, CAGLAR

2. 張寧/ZHANG, NING

3. 徐文平(史考特)/HSU, WEN-PIN SCOTT

國籍：(中文/英文)

1. 土耳其/Turkey
2. 美國/USA
3. 中華民國/ROC

#### 四、聲明事項：

主張專利法第二十二條第二項  第一款或  第二款規定之事實，其事實發生日期為： 年 月 日。

申請前已向下列國家(地區)申請專利：

【格式請依：受理國家(地區)、申請日、申請案號 順序註記】

有主張專利法第二十七條第一項國際優先權：  
美國；2003年10月24日；10/692,868

無主張專利法第二十七條第一項國際優先權：

主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號 順序註記】

主張專利法第三十條生物材料：

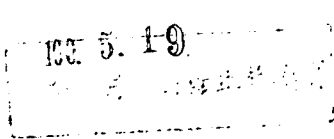
須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼 順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼 順序註記】

不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。



## 五、中文發明摘要：

一種軟體授權「應用程式設計介面(API)」，其可讓軟體產品能夠運用一公用服務之授權管理功能。一授權可標定一軟體產品內的權利。該軟體產品可呼叫該 API 上之一消費方法以消費一權利。若該權利存在，則該服務會將該權利連接於其中發現有該權利之授權。該軟體產品會根據是否發現該權利之一有效實例而藉由准允，或拒絕，存取至部分或所有的特性，來啟動該授權之各條項。可將任意資料關聯於一權利。該 API 包含一用以自一先前經該消費方法所連接之權利擷取資料的方法。

## 六、英文發明摘要：

A software licensing Application Programming Interface (API) that allows software products to use the license management functionality of a common service. A license specifies rights in a software product. The software product calls a consume method on the API in order to consume a right. If the right exists, the service binds the right to the license in which the right is found. The software product enforces the terms of the license by granting, or denying, access to some or all features depending on whether a valid instance of the right is found. Arbitrary data can be associated with a right. The API includes a method to retrieve data from a right that has been previously bound by the consume method.

## 七、指定代表圖：

(一)、本案指定代表圖為：第 3 圖。

(二)、本代表圖之元件代表符號簡單說明：

302 應用程式發出 API 呼叫

306 服務送回結果至應用程式

304 服務處理 API 呼叫

308 應用程式根據該結果決定

行為

八、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無

## 九、發明說明：

### 【發明所屬之技術領域】

本發明係關於電腦軟體領域，且更明確而言係關於一可支援電子授權啟動作業之程式設計介面。

### 【先前技術】

商業生產的軟體傳統上係需在定義使用該軟體之可允許項目的授權下方可使用。當開始實施軟體授權時，授權一般會採取定義該使用者對於該軟體之權利的法律文件形式。這種文件仰賴於執行法律的法律系統。而近來授權已變成需要依電子方式來執行 - 亦即會需要一種含有能夠主動地阻擋或防止按違反該授權方式使用軟體之程式碼的電腦程式。

多數提供電子授權執行的軟體會提供其本身的基礎架構，以管理該軟體的授權作業及授權使用。因此，一典型商業軟體產品可不僅包含以執行該產品之核心功能的程式碼，同時亦可載荷有用以獲得、評估、保護及管理對於軟體之授權的程式碼。對於各軟體販售廠商，發展及併入此基礎架構於該軟體內通常會是重複耗費人力。從而會希望能夠提供一種系統，可執行有關於軟體授權的基本功能，而其中可由各式各樣的軟體應用程式按一均勻且經定義之方式使用該系統。

有鑑於前文所述，目前業界對於可克服習知缺失之機制存有著需求。

## 【發明內容】

本發明提供一種軟體授權「應用程式設計介面(API)」，此者可提供一些由軟體產品使用的授權功能。一授權服務可執行有關於授權運用的功能，且透過該API將該等功能暴出予該軟體產品。該服務可執行像是取得授權、儲存及管理授權、保護授權而不受破壞、評估一授權的有效性，以及評估一授權是否經正確地綁接於其上運用此授權之機器及/或軟體產品的功能。該軟體能夠藉由呼叫該API之各項方法以運用此項功能。

該API的典型使用裡，一軟體產品呼叫一「開放」API方法，藉以取得一由該授權服務所使用俾識別該應用程式之特別控制碼(handle)。然後該軟體產品呼叫一「消費權利」API方法。在此語意內容下，「消費」意思是執行一標定權利。呼叫該「消費權利」方法可由該軟體產品之控制碼，以及藉待予消費之權利的名稱所參數化。然後該授權服務會試圖定位出一或更多含有所指定之權利的有效、經正確綁付授權。如並不存在該等授權，則會將此失敗訊息通知該軟體產品。如該等授權確實存在，則會將該權利連接於各授權其一者，且會將此項連接處理通知該呼叫軟體產品。在此時，該軟體產品會知悉該權利存在，且可執行與此項權利相關的所有功能。

在一較佳具體實施例裡，該授權服務並未定義在該權利下該軟體能夠或無法作到的項目，或是在運用該軟體時

應執行的重要限項。相反地，該授權服務是按照一軟體產品藉由呼叫該 API 來決定一權利是否確實，或並非，存在而讓該軟體能夠因此正確運作之方式來管理該授權。例如，可稱一權利為「執行(run)」，表示該使用者有權利執行該軟體產品。該軟體產品可使用該 API 以決定是否確有一有效(且經正確地連接及未逾期)權利以執行該軟體。不過，若該 API 呼叫送回一表示沒有權利執行該軟體之指示，則會根據並不存在此項權利，而由該軟體停止操作或是採取一些動作。

可將一權利關聯於一資訊，而此者在成功地呼叫該「消費權利」方法之後變成可用。例如，一給定軟體產品可具有關於何時可允許以編輯、列印、存檔等之個別規則，且可將這些規則儲存在含有該權利之授權內。該 API 提供一能夠從該授權中擷取出此項資訊之「取得資訊」方法。

本發明之其他特性可如後述。

## 【實施方式】

### 概論

商業軟體的使用通常是由一授權所掌控，且愈來愈多是按可由該軟體本身加以執行之電子形式來實作此授權。產創一電子授權系統的一項挑戰就是會需要一基礎架構以管理授權使用。對每一軟體產品複製此基礎架構會極為繁瑣及耗費資源。本發明提供一種讓不同軟體產品能夠使用一執行各式授權功能之共同基礎架構的 API。

### 示範性計算環境

第 1 圖顯示一其中可實作本發明特點之示範性計算環境。該計算系統環境 100 僅為一適當計算環境之範例，且非為意指對本發明運用及功能性之範圍進行任何限制。該計算環境 100 亦不應被詮釋為具有對任何關於該示範性操作環境 100 內各所述元件之任者或組合的依賴性或要求。

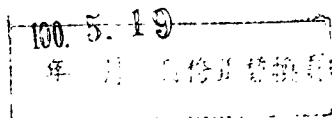
本發明可運作於各式其他一般目的或特殊目的計算系統環境或組態。適合運用於本發明之眾知計算系統、環境及/或組態之範例包含，但不限於此，個人電腦、伺服器電腦、手持式或膝上型裝置、多重處理器系統、微處理器基礎式系統、機頂盒、可程式化消費性電子裝置、網路 PC、迷你電腦、大型主機電腦、嵌入式系統，含有前述各系統或裝置之任者的分散式計算環境等等。

本發明可按一像是由一電腦所執行之程式模組的電腦可執行指令之一般語意內容所描述。一般說來，程式模組包含副程式、程式、物件、元件、資料結構等，可執行特定任務或是實作特定的抽象資料型態。亦可在一分散式計算環境內實作本發明，其中是由經一通訊網路或其他資料傳送媒體所鏈結之遠端處理裝置來執行各項任務。在一分散式計算環境裡，各項程式模組及其他資料可位於含記憶體儲存裝置之本地及遠端電腦儲存媒體兩者處。

參照第 1 圖，一用以實作本發明之示範性系統包含一按一電腦 110 之形式的一般目的計算裝置。該電腦 110 的

組成元件可包含(但不限於)一處理單元 120、一系統記憶體 130，以及一可將包含該系統記憶體在內之各式系統元件耦接於該處理單元 120 的系統匯流排 121。該處理單元 120 可代表多個邏輯處理單元，像是在一多重執行緒處理上所支援者。該系統匯流排 121 可為多款含有一記憶體匯流排或記憶體控制器之匯流排結構任一者、一週邊匯流排，以及一利用各匯流排架構任一者之區域匯流排。僅為範例，且不限於此，這些架構包含「工業標準架構(ISA)」匯流排、「微通道架構(MCA)」匯流排、改良 ISA (EISA) 匯流排、「視訊電子標準協會(VESA)」區域匯流排，及「週邊元件互連(PCI)」匯流排(又稱為 Mezzanine 匯流排)。該系統匯流排 121 亦可經實作為在各通訊裝置之間的點對點互連，交換構物等。

電腦 110 通常含有各種的電腦可讀取媒體。電腦可讀取媒體可為任何可由該電腦 110 存取之可用媒體，且含有揮發性及非揮發性媒體、可移除及非可移除媒體兩者。僅為範例，且非為限制，電腦可讀取媒體可包含電腦儲存媒體及通訊媒體。電腦儲存媒體包括按任何方法或技術所實作，以儲存資訊像是電腦可讀取指令、資料結構、程式模組或其他資料之揮發性及非揮發性、可移除及非可移除媒體兩者。電腦儲存媒體包含，但不限於此，RAM、ROM、EEPROM、快閃記憶體或其他記憶體技術、CD-ROM、DVD 光碟或其他光學碟片儲存物、磁匣、磁帶、磁碟儲存物或其他磁性儲存裝置，或是任何其他可用以儲存所欲資訊且



可由該電腦 110 存取之媒體。通訊媒體通常可於一像是一載波或其他傳送機制之經調變資料信號內具體實作電腦可讀取指令、資料結構、程式模組或其他資料，並且包含任何資訊遞送媒體。該名詞「經調變資料信號」意思是具有一或更多其特徵集合，或是按如將資訊編碼於該信號內之方式而改變的信號。僅以範例，但非限於此，通訊媒體包含像是有線網路或直接有線連線之有線媒體，以及像是音響、RF、紅外線及其他無線媒體之無線媒體。任何上述各者之組合亦應被納入該電腦可讀取媒體的範圍內。

該系統記憶體 130 包含按揮發性及/或非揮發性記憶體形式之電腦儲存媒體，像是唯讀記憶體 (ROM) 131 及隨機存取記憶體 (RAM) 132。一基本輸入/輸出系統 133 (BIOS) 包含有助於在該電腦 110 之各元件間傳送資訊的基本副程式，該者通常會含有立即可存取至及/或現由該處理單元 20 所操作之資料及/或程式模組。僅以範例，但非限於此，第 1 圖列出作業系統 134、應用程式 135、其他程式模組 136 及程式資料 137。

電腦 110 亦可含有其他的可移除及非可移除、揮發性及非揮發性電腦儲存媒體。僅以範例，第 1 圖繪出一可對非可移除、非揮發性磁性媒體進行讀寫之硬碟機 141、一可對可移除、非揮發性磁性媒體 152 進行讀寫之磁碟機 151，以及一可對像是 CD-ROM 或其他光學媒體之可移除、非揮發性光學碟片 156 進行讀寫的光碟機 155。其他可適用於本示範性作業環境之可移除/非可移除、揮發性/非揮

發性電腦儲存媒體包含，但不限於此，磁帶匣、快閃記憶卡、數位光碟 DVD、數位影帶、全晶體 RAM、全晶體 ROM 等等。該硬碟機 141 通常是透過像是介面 140 之非可移除記憶體介面而連接於該系統匯流排 121，且磁碟機 151 及光碟機 155 通常是透過像是介面 150 之可移除記憶體介面而連接於該系統匯流排 121。

如前所述且第 1 圖所示之各項裝置及該等相關電腦儲存裝置可提供儲存該電腦 110 的電腦可讀取指令、資料結構、程式模組及其他資料。例如，在第 1 圖裡硬碟機 141 經圖繪為儲存作業系統 144、應用程式 145、其他程式模組 146 及程式資料 147。注意這些元件可與該作業系統 134、應用程式 135、其他程式模組 136 及程式資料 137 互為相同或是相異。在此將該等作業系統 144、應用程式 145、其他程式模組 146 及程式資料 147 給定不同編號，以為說明該等至少可為不同拷貝。一使用者可透過像是一鍵盤 162 及點指裝置 161 之輸入裝置，在此共稱為滑鼠、軌跡球或觸控板，將命令及資訊輸入該電腦 20 內。其他輸入裝置(未以圖示)可包含一麥克風、搖桿、遊戲板、衛星碟、掃描器等。這些及其他輸入裝置通常是透過一耦接於該系統匯流排之使用者輸入介面 160 而連接到該處理單元 120，但亦可藉由其他介面及匯流排結構所連接，像是平行埠、遊戲埠或一通用序列匯流排(USB)。一監視器 191 或其他種類的顯示裝置亦可經一像是視訊介面 190 之介面而連接於該系統匯流排 121。除顯示器外，電腦亦可包含其他像是喇

以 197 及印表機 196 的週邊輸出裝置，這些可透過一輸出週邊介面 195 而連接。

該電腦 110 可利用接至像是一遠端電腦 180 之一或更多遠端電腦邏輯的連接而運作於一網接環境內。該遠端電腦 180 可為一個人電腦、一伺服器、一端點裝置或其他常見網路節點，且通常包含前文對於該電腦 110 所敘述之許多或所有元件，然第 1 圖中僅描繪一記憶體儲存裝置 181。如第 1 圖所繪之各邏輯連接包含一區域網路 (LAN) 171 及一廣域網路 (WAN) 173，但亦可包含其他網路。這些網路環境屬辦公室、企業領域電腦網路、企業內網路及網際網路中所常見者。

當運用於一 LAN 網接環境內時，該電腦 110 可透過一網路介面或配接器 170 而連接於該 LAN 171。當運用於一 WAN 網接環境內時，該電腦 110 通常會包含一數據機 172 或其他裝置以於像是網際網路之 WAN 173 上建立通訊作業。該數據機 172 可為內插式或外接式，且可透過使用者輸入介面 160，或是其他適當機制，連接於該系統匯流排 121。在一網接環境下，前文對於該電腦 110，或其局部，所述之程式模組可儲存在該遠端記憶體儲存裝置內。僅以範例，而非其限制，第 1 圖係按常駐於記憶體裝置 181 內之方式來描述遠端應用程式 185。應瞭解所繪之各網路連接僅為示範性，且確可採用其他用以在各電腦之間建立一通訊鏈路的方式。

## 軟體授權服務

第 2 圖顯示一種可提供一軟體授權服務 202 之示範性系統。該軟體授權服務 202 運作於電腦 110 (如第 1 圖所示) 之內。在一範例裡，該軟體授權服務 202 係執行於電腦 110 上之作業系統的一部分。軟體授權服務可維護一授權儲存庫 204，其中儲存有對於各軟體之授權檔案。各授權檔案可為例如標定對軟體之權利，且亦可標定實施該等權利之各項條件的「可延伸權利標號語言(XrML)」檔案。軟體 204 亦維護一信賴儲存庫 206。該信賴儲存庫 206 可按一防竄改之方式來儲存未可辨核、動態資料；該信賴儲存庫 206 儲存用於授權核可處理之資料。例如，有些授權或具有逾期日期，且為防止藉時鐘回轉而蓋掩過該逾期日期，故可將目前時間(及經過時間)週期性地儲存在該信賴儲存庫 206 內，以確保時鐘總是會向前行走。

該軟體授權服務 202 管理該授權儲存庫 204 及該信賴儲存庫 206，並亦可執行關於軟體授權處理之各種功能。例如該軟體授權服務 202 可含有用以剖析授權檔案之模組、用以將一授權連接於一特定機器及/或至一軟體產品之特定實例之模組，以及一安全計時器/計數器(此係按前述方式運用該信賴儲存庫 206)。

該軟體授權服務 202 可呈現一應用程式設計介面(API) 208，讓應用程式軟體(像是應用程式 135)能夠利用軟體授權服務 202。該應用程式 135 可藉由對該 API 208 之方法提出邏輯程序呼叫(LPC)，行使該軟體授權服務 202 的各項

特性。一可由該軟體授權服務 202 所呈現之 API 方法的示範性集合可如下述。

由一應用程式利用 API 208 的方式可參照第 3 圖所述。首先，該應用程式提出一 API 呼叫(302)。然後，服務 302 處理該 API 呼叫(304)，並將該 API 呼叫的結果送回給應用程式 306。例如，一 API 呼叫可請求執行(「消費」)在一授權所准允之權利，或是以自一授權中擷取資訊。然後該應用程式接收該 API 呼叫結果，並根據該結果決定該應用程式的行為應為如何(308)。換言之，在一較佳具體實施例裡，該軟體授權服務 202 並不直接地執行一授權，而是透過可管理及運用哪些授權來提供該基礎架構。例如，若一應用程式提出一 API 呼叫以消費一權利，而該服務 202 決定並無有效授權准允此項權利，則在一較佳具體實施例裡該服務 202 並不阻止該應用程式執行，而是通知該應用程式該權利非屬可用者。如此，該應用程式會利用其本身的機制以回應於非可用權利而決定應如何處理。該 API 的這種方式可提供軟體販售廠商一種彈性，來決定應如何運用該服務 202 所提供之授權基礎架構。在其他實作裡，可將應用程式連接於授權服務功能性。

#### 示範性軟體授權 API

底下係一可由一軟體授權服務所呈現之 API 方法的示範性集合：

## SLOpen

此 SLOpen 函式開啟一必須對所有後續 SL API 呼叫加以運用之 SL 客戶端控制碼(在整個示範性 API 說明裡，「SL」是指軟體授權服務。如此，該 SL 客戶端語意內容控制碼會是由一與該軟體授權服務相通訊之客戶端所使用的控制碼)。

## HRESULT

```
SLOpen(  
    CONST GUID* pguidApp,  
    HSLC* phSLC  
);
```

## 參數

### pguidApp

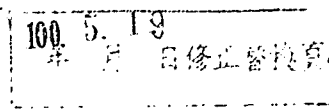
可為一識別一應用程式之應用程式 GUID 的指標。如此引數為無效值(NULL)，則會送回一 E\_INVALIDARG 錯誤。

### phSLC

[輸出] SL 客戶端語意內容控制碼，或是如失敗則為 INVALID\_HANDLE\_VALUE。

## 附註說明

利用該 SLClose 函式以關閉一由 SLOpen 所送回之



語意內容控制碼。

應用程式 GUID：應用程式的唯一 ID。對於微軟公司 Office 應用程式套裝的 Windows 版本，WinWord 具有一不同於 Excel 之應用程式 GUID 的應用程式 GUID。對於 Windows 來說，Windows 雖係由許多程式所組成，然其本身係一應用程式。

即以 Office 的情況而言，一使用者可於該機器上安裝 Office 套裝及 WinWord 自立式產品兩者。從 SL 的觀點來看，在兩者產品內的 WinWord 具有相同的應用程式 GUID。WinWord 具有關聯於兩個產品 GUID 的應用程式 GUID。換言之，WinWord 可使用 Office 套裝的產品授權或 WinWord 的產品授權。

當 SLOpen 成功後：

既已建妥一 RPC (遠端程序呼叫) 連接處理。

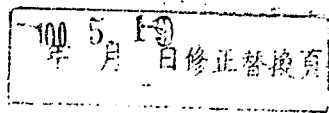
於 SL 服務上產生一語意內容記憶體。可利用該語意內容以保持對該客戶端之呼叫方的狀態資訊。

該 SLC 控制碼就像是一檔案控制碼。一處理可開啟多個管理語意內容控制碼，但是控制碼是在該呼叫方處理之內為有效。

送回

成功或失敗。

SLClose



此 SLClose 函式關閉一經開啟之 SL 客戶端語意內容控制碼。任何在此語意內容中的資訊會被自動釋放。

HRESULT

SLClose (

HSLC hSLC

);

參數

hSLC

[輸入]至目前 SL 客戶端語意內容控制碼的控制碼。

附註說明

當完成 SLClose 後，會釋放該 RPC 連接處理且摧毀此語意內容。

送回

成功或失敗。

SLInstall

此 SLInstall 函式可安裝應用程式的授權並登註應用程式的資訊。

HRESULT SLInstall(

HSLC

hSLC;

```

CONST SL_PRODKEY*   pAppPrdKey,
DWORD               dwNumOfApps,
CONST GUID*         pguidApps,
DWORD               dwNumOfLicFiles,
PCWSTR              ppsLicFiles[],
BOOL                bVerify
);

```

參數

hSLC

[輸入] 至目前 SL 客戶端語意內容控制碼的控制碼。

pApp

[輸入] 應用程式產品鍵值結構。該產品鍵值可為微軟產品鍵值格式或應用程式的產品鍵值格式。

```

typedef struct_tagSL_PRODKEY
{
    DWORD        cbSize; //SL_PRODKEY 結構的大小
    DWORD        dwVersion //SL_PRODKEY 結構的版本
    WCDAR        szProdKey[MAX_PRODKEY_SIZE+1];
    SL_PRODKEY_TYPE eProdKeyType;//產品鍵值型態
    SL_CUSTOM_PRODKEY_INFO CustomPrdKeyInfo;
    //客戶產品鍵值資訊

```

```
} SL_PRODKEY;
```

該 eProdKeyType 可為如下數值其中一者：

```
SL_PRODKEY_CUSTOM
```

```
SL_PRODKEY_MS2002
```

```
SL_PRODKEY_MS2003
```

如果該產品鍵值型態為非微軟產品鍵值（亦即 eProdKeyType==SL\_PRODKEY\_CUSTOM），則呼叫方必須填入其自訂產品鍵值資訊。如該產品利用微軟產品鍵值，則可忽略 CustomPrdKeyInfo。

```
typedef struct_tagSL_CUSTOM_PRODKEY_INFO
    DWORD dwSKUID; //特定 SKU 之唯一 ID，例如 MS
PID 內的群組 ID
    DWORD dwSerialNumber; //唯一序號，即如 MS PID
內的通道+序號
} SL_CUSTOM_PRODKEY_INFO;
```

SL\_PRODKEY 的目前版本編號為 1。該呼叫方可利用 dwVersion 欄位內的 SL\_CURRENT\_PRODKEY\_VERSION。

```
dwNumOfApps
```

[輸入] pguidApps 內的應用程式 GUID 編號

`pguidApps`

[輸入]— GUID 應用程式列表。該應用程式 GUID 代表經安裝有授權之應用程式。例如，Office 安裝程式可呼叫此函式，以藉由標定 `pguidApps` 內的各應用程式 GUID 來安裝對於 Word、Excel 之該(等)授權。在此，`pguidApp` 不可為無效值。

`dwNumOfApps`

[輸入]授權檔案數量。

`ppszLicFile`

[輸入]字串陣列內的檔案名稱。

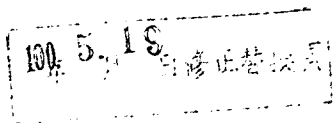
送回

成功或失敗。

`SLUninstall`

此 `SLUninstall` 函式可自一應用程式卸除一產品授權。

```
HRESULT SLUninstall (
    HSLC hSLC,
    CONST SL_PRODKEY* pAppPrdKey
);
```



參數

hSLC

[輸入]至目前 SL 客戶端語意內容控制碼的控制碼。

pAppPrdKey

[輸入]參見前文有關 SLInstall 對此之定義。

一應用程式可具有一個以上的產品授權。例如，當該使用者卸除 Office 套裝時，應移除於 Office 套裝授權與 WinWord 之間的關聯性，但是不應移除來自 WinWord 自立式產品的授權。

當 SLUninstall 成功時：

會移除有關於此產品鍵值之資訊(參見前 SLInstall 其相關資訊)。

會移除與該應用程式 GUID 有關的產品鍵值。

會最好保留與該產品 GUID 有關的授權檔案。

送回

成功或失敗。

SLConsumeRight

此 SLConsumeRight 函式可讓一應用程式對於一本地儲存授權進行檢查或執行權利。呼叫此函式會將一授權連接

於在 pszRightName 中所說明之權利。如無法由該目呼叫方執行此權利，則該應用程式失敗。如該函式成功，則可進行與此權利相關的動作(像是減少使用計數、減少時間配額或無)。

```
HRESULT SLConsumeRight (
    HSLC hSLC,
    PCWSTR pszRightName,
    SL_ASYNC_CONTEXT* pAsyncContext
);
```

#### 參數

**hSLC**

[輸入]至目前 SL 客戶端語意內容控制碼的控制碼。

**pszRightName**

[輸入]需要對權利名稱加以評估。在目前設計裡，該權利名稱是由應用程式所定義。SL 開啟該授權，並根據該權利名稱來評估各條件。

**pAsyncContext**

[輸入/輸出]如 pAsyncContext 為無效值，則此函式會按同步模式運作，否則該函式會按非同步模式運作。SL\_ASYNC\_CONTEXT 對於呼叫方為不透明，且是由 SLC

加以管理。

#### 附註說明

所有與該應用程式 GUID (於 SLOpen 中所標定) 相關的授權會概念上被合併為單一邏輯授權。

如該等為該權利的多個可消費准允，則會先消費具有較高優先權的授權。

#### 送回

成功或失敗。

#### SLInitializeAsyncContext

此 SLInitializeAsyncContext 函式會將對 SLC 函式之非同步語意內容加以初始化，俾進行非同步呼叫。

```
HRESULT SLInitializeAsyncContext(  
    SL_ASYNC_CONTEXT* pAsyncContext,  
        //非同步語意內容  
    HANDLE    hEvent,    //事件控制碼  
    PVOID     pvReserved //保留，無效值  
);
```

#### 參數

pSyncContext

[輸入/輸出] 朝向含非同步呼叫資訊之非同步語意內容的指標。

hEvent

[輸入] 用以同步作業之事件物件。

pvReserved

[輸入] 保留或延伸。

送回

成功或失敗。

SLCancelAsyncCall

此 SLCancelAsyncCall 函式係用於取消一非同步呼叫。

```
HRESULT SLCancelAsyncCall(  
    SL_ASYNC_CONTEXT* pAsyncContext,  
    //非同步語意內容  
    BOOL fAbortCall //立即取消  
);
```

參數

pAsyncContext

[輸入] SL 非同步呼叫的非同步語意內容。

### fAbortCall

[輸入]如為「真」，則立即取消該呼叫。如為「偽」，則等待該 SL 完成此呼叫。

呼叫方有兩種方式來請求取消一非同步呼叫 - 中止性及非中止性。在中止性取消裡 (fAbortCall 為「真」)，該 SLCancelAsyncCall 函式會送出一取消通知給該 SLC，且立即取消該呼叫，而不必等待來自該 SLC 的回應。在非中止性取消裡 (fAbortCall 為「偽」)，該 SLCancelAsyncCall 函式會通知該 SLC 進行取消，並且呼叫方會等待 SLC 完成該呼叫。

### 送回

成功或失敗。

### SLCompleteAsyncCall

此 SLCompleteAsyncCall 函式係用於完成一 SLC 非同步呼叫。

```
HRESULT SLCompleteAsyncCall(  
    SL_ASYNC_CONTEXT* pAsyncContext,  
    //非同步語意內容  
    HRESULT* phrAsyncCall
```

100.5.19  
//所遞呈之非同步呼叫的錯誤碼

);

### 參數

pAsyncContext

[輸入]SL 非同步呼叫之非同步語意內容。

phrAsyncCall

[輸出]所遞呈之非同步呼叫的錯誤碼。

### 附註說明

如該呼叫方在回覆抵達之前呼叫此函式，該呼叫會送回 E\_SLC\_ASYNC\_CALL\_PENDING。緩衝器必須足夠大，以接收該送回數值。如呼叫並未送回 E\_SLC\_ASYNC\_CALL\_PENDING，則對於該非同步呼叫，此 SLCompleteAsyncCall 行使會為最終。在此函式呼叫之後，無論成功或失敗，都會釋放所有經配置給此非同步呼叫的資源(對於該 SLCompleteAsyncCall 或 SLCompleteAsyncCancel 的後續呼叫會具有未定義結果，一直到啟動對 SL\_ASYNC\_CONTEXT 結構上的新呼叫為止)。

### 送回

數值

S\_OK

意義

成功完成該呼叫

E_SLC_INVALID_ASYNC_CONTEXT	此非同步呼叫語意內容非有效
E_SLC_ASYNC_CALL_PENDING	尚未完成該呼叫
E_SLC_CALL_CANCELLED	取消該呼叫

### SLGetInformation

此 SLGetInformation 函式係用來取得各種資訊。

```

HRESULT SLGetInformation(
    HSLC      hSLC,          //SL 客戶端語意內容控制碼
    DWORD     dwCategory,   //待擷取之資訊範疇
    PCWSTR    pszKeyName,  //鍵值名稱
    DWORD*    pdwType,      //數值型態
    SIZE_T*   pcbValue,    //數值大小
    PBYTE*    ppbValue     //指向數值緩衝器之指標
);

```

#### 參數

**hSLC**

[輸入] 至目前 SL 客戶端語意內容控制碼的控制碼。

**dwCategory**

[輸入] 資訊範疇。

範疇	意義
SL_CAT_RIGHTD ATA	從經連接權利取得資訊。在取得這些權利資料之前，必須成功地消費該授權。
SL_CAT_DYNAMI CPROPERTY	取得不在授權內，但會在執行時間內計算的資訊。例如 Remaining GracePeriodDays。必須在呼叫之前消費該權利。

名稱	意義
RemainingG race PeriodDays: DWORD	惠緩時段係經定義於開盒授權內。一旦安裝該應用程式後就會開始倒數計時。應用程式可消費授權後，檢查剩餘的惠緩時段。
ActivationS tatus: DWORD	在應用程式消費授權之後，就會取得消費授權型態。所送回之數值可為如下：
SL_LIC_OO B	所消費之授權 為開盒授權
SL_LIC_AC QUIRED	所消費之授權 為取得授權
SL_LIC_NO NE	無可用授權
SL_CAT_SERVICE INFO	取得不與授權相關的資訊。該呼叫方可取得此範疇之資訊，而無需消費授權。

	名稱	意義
	SLVersion:	Sl.1.2.3.4 格式之版本
	DWORD	
	HWID:	目前 HWID
	BINARY	
SL_CAT_WINDOW INFO		取得在 Windows 授權裡所連接之權利性質的資訊。這是為元件化作業。SL 服務既已消費 Windows 授權，且 SL 服務會保留所連接之權利性質。
SL_CAT_ENUMCL ICINFO		當呼叫 SLEnumLicense 且成功時，該呼叫方可藉由利用此範疇來詢查所列舉之授權的資訊。

pszKeyName

[輸入]該鍵值之名稱，即如 BuildNumber。

pdwType

[輸出]資料型態

數值	意義
SL_DATATYPE_SZ	Unicode 字串
SL_DATATYPE_DWORD	DWORD
SL_DATATYPE_BINARY	二進位

pcbValue

[輸出]所配置之緩衝器的大小(按位元組)。

### ppbValue

[輸出]如為成功，會由 SLC 將該資料送回於經配置之緩衝器內。該呼叫方必須呼叫 SLFreeMemory 以釋放記憶體。

### 送回

成功或失敗。

### SLAcquireLicense

此 SLAcquireLicense 函式係用於取得對該使用者的線上授權。SLC 列舉出與該應用程式相關的產品鍵值，並檢拾出具有最高產品優先權的產品鍵值(參見 SLInstall，登註資訊)。然後 SL 會從該開盒授權取得結算中心(Clearing House) URL，並連接至此結算中心以取得一授權。

SLAcquireLicense 或為一冗長程序。應用程式可藉由標定 pAsyncContext (「無效值」表示為同步模式)，按非同步模式來呼叫此函式。

### HRESULT SLAcquireLicense(

```

HSLC          hSLC          //SL 客戶端語意內容控制碼
PCWSTR        pszProdKeyHash, //產品鍵值雜湊

```

```
PCWSTR pszPublishLicense, //刊行授權字串
SL_ASYNC_CONTEXT* pAsyncContext
);
```

參數

hSLC

[輸入]至目前 SL 客戶端語意內容控制碼的控制碼。

pszProdKeyHash

[輸入]產品鍵值雜湊之字串。該產品鍵值雜湊係在當呼叫 SLInstall 時所產生，且由該授權服務所維護。

pszPublishingLicense

[輸入]刊行授權字串。

pAsyncContext

[輸入]SL 非同步呼叫的非同步語意內容。

附註說明

會據此而將所取得之授權儲存在授權儲存庫內，並且亦會登註該授權資訊(參見 SLInstall)。

應用程式或需要增加更多的客戶端資訊以供授權。該應用程式可將資訊放在該呼叫的 pbAppData 內，並且會將此資料送往結算中心。

當此函式成功時：

這會將必要的連接資訊送出給所標定的授權伺服器。

這會從該授權伺服器接收授權。

這會將授權儲存在授權儲存庫內。參見前文中有關於 SLInstall 如何儲存授權檔案之說明。

送回

成功或失敗。

### SLGenerateTextChalleng

可產生及安裝待路由傳送到一按一開盒方式(電話、電子郵件、檔案分享等)之授權發出者的挑戰文字。

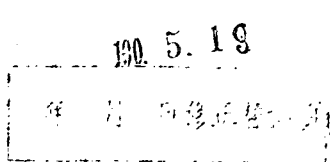
```

HRESULT SLGenerateTextChallenge(
    HSLC      hSLC,          // SL 客戶端語意內容控制碼
    PCWSTR   pszProdKeyHash, // 產品鍵值雜湊之字串
    BOOL     fSingleSession, // 僅單一會期?
    PWSTR    *ppszChallenge // 指向為握持挑戰文字之
緩衝器的指標
);

```

參數

hSLC



[輸入]至目前 SL 客戶端語意內容控制碼的控制碼。

#### pszProdKeyHash

[輸入]產品鍵值雜湊之字串。該產品鍵值雜湊係在當呼叫 SLInstall 時所產生，且由該授權服務所維護。

#### bSingleSession

[輸入]標定來自於授權發訊之相對應文字回應是否僅對此 SLC 會期控制碼之生命時段為有效。

#### ppszChallenge

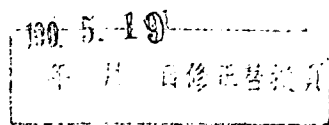
[輸出]如為成功，會送回該文字挑戰於由 SLC 所配置之該緩衝器內。呼叫方會需要呼叫 SLFreeMemory 以釋放經配置之記憶體。

#### 送回

成功或失敗。

#### SLDepositTextResponse

將對一安裝挑戰之回應存積於該授權系統內。用以藉由一條件存取碼來啟動一授權。僅有在確有一既已對一授權而發訊之顯著文字挑戰時方為有效。如原始授權標定該挑戰僅係對一單一會期，則必須在關閉該 SLC 控制碼之前先藉由該回應來呼叫此 API，否則存積該回應將會失敗。



```
HRESULT SLDepositTextResponse(  
    HSLC    hSLC,           //SL 客戶端語意內容控制碼  
    PCWSTR  pszProdKeyHash,  
    PWSTR   pszResponse //含回應文字之緩衝器  
);
```

參數

hSLC

[輸入]至目前 SL 客戶端語意內容控制碼的控制碼。

pszProdKeyHash

[輸入]產品鍵值雜湊之字串。

pszChallenge

[輸入]回應文字。

送回

成功或失敗。

SLEnumLicense

此 SLEnumLicense 函式係用於列舉所安裝之授權，並從該授權取得資訊。

```

HRESULT SLEnumLicense(
    HSLC      hSLC,          //SL 客戶端語意內容控制碼
    CONSTGUID*  pguidApp,   //應用程式 GUID
    DWORD      dwIndex      //索引編號
);

```

**hSLC**

[輸入]至目前 SL 客戶端語意內容控制碼的控制碼。

**pguidApp**

[輸入]參見 SLInstall。如 pguidApp 非為無效值，則會列舉出與此 GUID 相關的授權。若該 GUID 為無效值，則會列舉出所有的授權。

**dwIndex**

[輸入]待予擷取之授權的索引編號。對於第一次呼叫 SLEnumLicense 函式，此數值應為零，且按後續呼叫予以增量。

該函式可按任意順序送回授權。

**送回**

**E\_SL\_NO\_MORE\_DATA** - 在所標定之索引處並無授權。

## 附註說明

若 SLEumLicense 為成功，則可藉由呼叫 SLGetInformation 來存取所選定之授權資訊，且範疇為 SL\_CAT\_ENUMLICINFO。

範例：

```
DWORD i=0;
PBYTE pbProductPriority=NULL;
PBYTE pbRemainingGracePeriodDays=NULL;
for(i=0;;i++)
{
    EXIT_ON_ERROR(SLEnumLicense(hSLC,NULL,dwIndex));
    if(E_SL_NO_MORE_DATE==SCODE(hr))
    {
        hr=S_OK;
        break;
    }

    EXIT_ON_ERROR(SLGetInformation(hSLC,SL_CAT_ENUM
LIC,"ProductPriority",&dwType,&cbProductPriority,&pbPro
ductPrioirty));

    EXIT_ON_ERROR(SLGetInformation(hSLC,SL_CAT_ENUM
LIC,"RemainingGracePeriodDays",&dwType,&cbRemainingG
```

```
racePeriodDays,&pbRemainingGracePeriodDays));
```

```
Exit:
```

```
SLFreeMemory(pbProductPriority);
```

```
SLFreeMemory(pbRemainingGracePeriodDays);
```

```
}
```

### SLFreeMemory

此 SLFreeMemory 函式係用於釋放由 SLC 所配置之記憶體。

```
VOID SLFreeMemory(
```

```
    PVOID          pvMemblock, //指向記憶體之指標
```

```
);
```

#### 參數

pvMemBlock

[輸入]待予釋放之先前配置記憶體區塊。

#### 送回

無。

利用軟體授權以控制軟體使用

一種軟體產品，此者可利用本發明 API 以供各種關於授權處理之目的，包含消費一授權內之各權利，以及從該

100.5.19

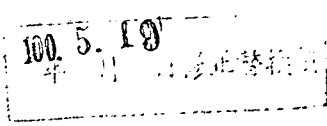
授權中擷取資料。即如前述，該 API 可供軟體決定在該授權中出現哪些權利，但是最好是由軟體來決定如何處置該資訊 - 即如准允或拒絕存取至一特性、完全停止操作等等。後述第 4 及 5 圖之說明顯示一軟體產品如何利用本發明。

第 4 圖顯示一示範性處理，藉此一應用程式可「消費」一權利。該應用程式呼叫該 SLConsumeRight 方法(402)。即如前述，對 SLConsumeRight 函式的引數包含由該授權服務所指派之客戶端管理，以及該權利的名稱(這是由該權利所歸屬之軟體販售廠商所指派)。該授權服務(如第 2 圖所繪之服務 202)接收呼叫(404)。然後該服務定位出含有該項權利的各個授權，並檢查授權連接及有效性。即如前述，該授權位於授權儲存庫內；如有屬於該 SLConsumeRight 呼叫所歸屬之應用程式軟體的一個以上授權，則可利用一優先權規則以選擇其中一項可適用授權。檢查連接處理的意思是為決定：(1)該授權係經連接於由該客戶端管理所識別之應用程式的產品鍵值；以及(2)該授權係經連接於其上執行該軟體之機器(或是該目前機器所歸屬為成員之一的機器群組)。檢查有效性可包含決定該權利尚未逾期(在標定有一逾期日期之授權的情況裡)，以及並未超過該權利的最高使用量(在授權標定可使用該權利(亦即「消費」)之最高次數的情況裡)。

如發現該授權及/或權利為經正確連接且有效(408)，則該授權會被連接於該 API 呼叫中所請求之權利(412)。(應

注意將一授權「連接」於一機器、一環境及一產品鍵值意思是該授權標定哪一(些)機器及可運用之產品鍵值；而將一授權「連接」於一權利意思是該消費函式為成功，且該權利正從一特定授權值中所消費。在本說明全篇中，可自語意內容而清晰看出適用哪一「連接處理」意義)。然後，該 API 呼叫送回至該呼叫應用程式，並表示該呼叫為成功(414)。如發現該授權及/或權利為無效，或是未經正確連接於該機器、環境或產品 ID，則該 SLConsumeRight 呼叫會送回該呼叫應用程式，並表示該操作失敗(410)。

如該 SLConsumeRight 呼叫送回一失敗，則無法從一授權中消費在該呼叫內所標定的權利，且不會有關於該權利之資訊會可用於該呼叫應用程式。然而，如成功地消費該權利，則該應用程式可利用對該授權之權利連接，以從該授權取得關於該權利的資訊。例如，一授權可含有一稱為「執行(run)」的一般權利，這表示可執行該應用程式。然而，對於「run」權利，該授權含有更多關於使用該應用程式的特定參數 - 即如該授權可標定一應用程式之特定特性(即如列印、編輯、存檔等)是否應為開啟或關閉，且可給定特定參數以運用這些特性(即如只能將文件儲存在一特定領域內執行之機器上，或只能使用列印特性 30 天等)。該 SL API 並不要求將任何特定型態的資訊關聯於一權利，而卻是提供一機制，一應用程式販售廠商可藉此將各種型態的資訊關聯到一權利，然後可由該應用程式加以擷取及解譯。



假定既已如第 4 圖所示成功地消費一權利，然後該應用程式可擷取關聯於該權利之資訊。擷取該資訊之處理可如第 5 圖所示。

首先該應用程式呼叫連接權利上之 `SLGetInformation` 方法(502)。可加以擷取之各種資訊可如前文關於 `SLGetInformation` 方法之說明所敘述。然後，該授權服務接收該呼叫(504)。該服務從含有所連接權利之授權檔案裡擷取所請求的資訊(506)。然後該授權服務將資訊放在一緩衝器內(508)，並送回給呼叫應用程式(510)。然後該呼叫應用程式讀取該緩衝器裡的內容，並執行該者根據所擷取資訊而認為屬必要的任何動作。

應注意該授權服務或並不知悉該者正在處理而作為 `SLGetInformation` 呼叫之一部分之資訊的意義。即如前述，該授權框架提供一種機制，藉此一軟體販售廠商可產生各項權利，且可將資訊關聯於各權利。本發明亦並不受任何可經關聯於該權利之特定的資訊型態所限制。當從授權中擷取出該資訊時，會僅通過該授權服務而傳到一緩衝器內的應用程式。然後該應用程式解譯該擷取資訊，根據該資訊決定應進行哪些動作，以及利用其本身的安全特性來執行該應用程式的決策。(即如，根據所擷取資訊，若該應用程式決定關閉列印功能，則該應用程式含有真正會關閉此項特性的程式碼，或可能以及為防止駭客入侵破壞列印功能關閉處理的程式碼)。

應注意既已提供各前揭範例僅係為解釋目的，而不應

被詮釋為限制本發明。雖既已按參照於各式具體實施例之方式說明本發明，然應瞭解在此所運用之字詞係為描述及說明性質之詞句，而非限制性質之詞句。此外，本發明雖既已參照於各特定裝置、材料及具體實施例而描述，然本發明不受限於本揭之各特定項目；相反地，本發明可延伸至所有功能性等同結構、方法及使用方式，像是如隨附請求項之範圍者。熟諳本項技藝之人士經閱覽本規格書教示後即可對其進行無數修改，且可加以改變而無虞悖離本發明各項特點之精神及原理。

#### 【圖式簡單說明】

經併同於各隨附圖式後即可更佳瞭解前揭概述，以及後纂之較佳具體實施例詳細說明。為說明本發明，各附圖中繪示有本發明各項示範性指令；然本發明並不限於本揭之各特定方法及機器。在各圖式裡：

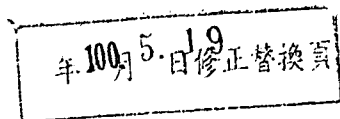
第 1 圖係一示範性計算環境區塊圖，其中可實作本發明。

第 2 圖係一架構區塊圖，其中一系統可執行授權功能，並呈現一 API 以供軟體產品使用。

第 3 圖係一經此一軟體產品可使用一授權 API 之方法流程圖。

第 4 圖係一藉此一軟體產品可消費一權利之方法流程圖。

第 5 圖係一藉此一軟體產品可擷取出有關於一經消費



權利之資訊的方法流程圖。

【主要元件符號說明】

- 100 計算系統環境
- 110 電腦
- 120 處理單元
- 121 系統匯流排
- 130 系統記憶體
- 131 唯讀記憶體 (ROM)
- 132 隨機存取記憶體 (RAM)
- 133 基本輸入/輸出系統 133 (BIOS)
- 134 作業系統
- 135 應用程式
- 136 其他程式模組
- 137 程式資料
- 140 非可移除非揮發性記憶體介面
- 141 硬碟機
- 144 作業系統
- 145 應用程式
- 146 其他程式模組
- 147 程式資料
- 150 可移除非揮發性記憶體介面
- 151 磁碟機
- 152 可移除非揮發性磁性媒體

- 155 光碟機
- 156 可移除非揮發性光學碟片
- 160 使用者輸入介面
- 161 點指裝置
- 162 鍵盤
- 170 網路介面
- 171 區域網路(LAN)
- 173 廣域網路(WAN)
- 180 遠端電腦
- 181 記憶體儲存裝置
- 185 遠端應用程式
- 191 監視器
- 190 視訊介面
- 195 輸出週邊介面
- 196 印表機
- 197 喇叭
- 202 軟體授權服務
- 204 授權儲存庫
- 206 信賴儲存庫
- 208 SL API
- 302 應用程式發出 API 呼叫
- 304 服務處理 API 呼叫
- 306 服務送回結果至應用程式
- 308 應用程式根據結果決定行為

- 402 應用程式藉由標定權利來呼叫消費-權利 API
- 404 服務接收呼叫
- 406 服務檢查授權連接及有效性
- 408 該授權(或權利)是否經連接且有效?
- 410 消費-權利將失敗送回給應用程式
- 412 授權經連接至所標定之權利
- 414 將成功送回至應用程式
- 502 應用程式呼叫於連接權利上的取得-資訊 API
- 504 服務接收該呼叫
- 506 服務從授權檔案擷取資訊；亦可取得或計算其他資訊
- 508 服務將所請求之資訊置放入緩衝器內
- 510 呼叫從服務送回至呼叫應用程式

102年7月5日修(更)正本  
P.45-53

## 十、申請專利範圍：

1. 一種執行軟體授權的系統，該系統包含：

一處理器，該處理器用以執行電腦可執行指令；及

一電腦可讀取儲存媒體，該電腦可讀取儲存媒體具有編碼於其上的電腦可執行指令以支援執行一電腦程式的授權，該電腦程式包含：在複數個授權下使用的一組指令，在該電腦程式中每一授權允許不同的權利，該等電腦可執行指令包含：

一授權元件，該授權元件為該電腦程式的該等複數個授權所共有，該授權元件維護其中儲存有該等授權之一授權儲存庫，該授權元件更維護以一防竄改的方式儲存動態資料的一信賴儲存庫，其中該動態資料用以驗證該等授權，每一授權包含：在該電腦程式中之至少一權利，以及一組與該至少一權利有關之資料，該授權元件將一可呼叫介面呈現於該電腦程式，該可呼叫介面包含：

一權利消費元件，該權利消費元件可從該電腦程式接收該權利之一識別碼，並決定是否可執行該權利，該決定是否可執行該權利之步驟係藉由檢查：

(1) 該授權對於一電腦程式的一產品鍵值之連接；(2) 該授權對於該系統的連接，其中該電腦程式在該系統上執行；(3) 該權利尚未逾期的一有效性，其中該電腦程式被通知決定該權利是否可被執行的結果，及基於該結果來決定該電腦程式的行為；

；以及

一 資訊擷取元件，該資訊擷取元件可從該電腦程式接收該權利之該識別碼，並提供該資料組、或是將以該資料組為基礎之資訊提供給該電腦程式。

2. 如申請專利範圍第 1 項之系統，其中該授權元件可由複數個不同的電腦程式共有和使用，及該授權儲存庫儲存該等複數個不同的電腦程式之授權，該電腦程式被納入於該等複數個不同的電腦程式內，其中該可呼叫介面更包含：

一 控制碼(handle)-開啟元件，該控制碼-開啟元件提供一控制碼給該電腦程式；

其中該權利消費元件從該電腦程式接收該控制碼，並利用該控制碼以識別出自該者而接收到對該權利消費元件之呼叫的電腦程式，及其中該電腦程式和授權元件位於一單一計算裝置。

3. 如申請專利範圍第 1 項之系統，其中該權利消費元件使得該授權元件依據一或更多如下所列之因素來選擇該授權：

該授權儲存庫是否與該電腦程式相關連；以及

一 衝突規則，該衝突規則決定從複數個關聯於該電腦程式之授權中選出何者授權。

4. 如申請專利範圍第 1 項之系統，其中該授權元件並未執行對該電腦程式的授權限制，及該可呼叫介面更包含：

一非同步語意內容啟動元件，該非同步語意內容啟動元件建立用於非同步處理之一語意內容，及將該語意內容之一識別碼提供給該電腦程式；

其中該權利消費元件從該電腦程式接收到該語意內容之該識別碼，並回應於接收該語意內容之該識別碼以非同步方式處理一權利消費請求。

5. 如申請專利範圍第 1 項之系統，其中該權利消費元件係根據該權利是否經識別為在該授權內，來決定是否可執行該權利。

6. 如申請專利範圍第 1 項之系統，其中該電腦程式及該授權元件執行於一機器上，且其中該權利消費元件係根據該授權是否經連接於該機器來決定是否可執行該權利。

7. 如申請專利範圍第 6 項之系統，其中該電腦程式係關聯於一產品識別碼，且其中該權利消費元件根據該授權是否經連接於該機器、或是經連接於其中該機器歸屬為其成員之一的機器類別來決定是否可執行該權利。

8. 一種由至少一機器實施的限制使用一電腦程式的方法，該電腦程式包含：在複數個授權下使用的一組指令，每一授權標定出在該電腦程式內之不同的權利，該方法包含：

透過該機器藉由從該電腦程式發出一第一呼叫至該授權服務之一可呼叫介面之第一方法，召喚該等複數個授權共有的一授權服務，而該第一呼叫係由與該權利相關聯之一識別碼所參數化；

藉由該授權服務，維護以一防竄改的方式儲存的動態資料之一信賴儲存庫，其中該動態資料用以驗證該等複數個授權；

回應於該第一呼叫，藉由該電腦程式接收一指示，該指示係關於：根據該等複數個授權的至少一者而是否執行該權利，其中是否可執行該權利之決定係藉由檢查：(1) 該授權對於一電腦程式的一產品鍵值之連接；(2) 該授權對於該系統的連接，其中該電腦程式在該系統上執行；(3) 該權利尚未逾期的一有效性，其中該電腦程式被通知決定該權利是否可被執行的結果，及基於該結果來決定該電腦程式的行為；以及

藉由該電腦程式，若該指示係肯定的，以一第一行為來進行，及若該指示係否定的，以一第二行為來進行。

9. 如申請專利範圍第 8 項之方法，其中該第一行為包含：允許該電腦程式執行，且其中該第二行為包含：不再繼續執行該電腦程式。

10. 如申請專利範圍第 8 項之方法，其中該第一行為包含：允許該電腦程式執行一第一組函數，且其中該第二行為包

含：允許該電腦程式執行與該第一組函數不同的一第二組函數。

11. 如申請專利範圍第 8 項之方法，其中該權利係關聯於一組資料，其中該方法更包含：

發出一第二呼叫給該界面之一第二方法，該第二方法係由該權利之一指示所參數化；以及

回應於該第二呼叫，接收該組資料。

12. 如申請專利範圍第 11 項之方法，更包含：

根據該組資料來導引該電腦程式之作業。

13. 如申請專利範圍第 8 項之方法，更包含：

發出一第二呼叫給該介面之一第二方法；以及

回應於該第二呼叫，接收一控制碼；

其中該第二呼叫係於該第一呼叫之前即發出，且其中該第一呼叫更進一步由該控制碼所參數化。

14. 如申請專利範圍第 8 項之方法，進一步包含：

發出一第二呼叫給該介面之一第二方法；以及

回應於該第二呼叫，接收一非同步語意內容；

其中該第二呼叫係於該第一呼叫之前即發出，且其中該第一呼叫係進一步由該非同步語意內容所參數化，且其中當非同步地處理該第一呼叫時，該電腦程式會執行至

少一個動作。

15. 如申請專利範圍第 8 項之方法，其中該第一方法根據包含如下所列之一或更多因素來決定是否為執行該權利：

該授權是否連接於其上正執行該電腦程式之一機器或環境；

該授權或權利是否連接於與該電腦程式相關連之一產品識別碼；

該授權或權利是否既已逾期；以及

該權利是否已消費超過在該授權中所標定權利的次數。

16. 一種於其上具有經編碼之電腦可執行指令以執行促進對一電腦程式執行一授權的方法之電腦可讀取媒體，該電腦程式包含：在複數個授權下使用的一組指令，每一授權在該電腦程式中允許不同的權利，該方法包含：

藉由該等複數個授權所共有的一授權服務，接收來自該電腦程式之一第一方法呼叫，該第一方法呼叫可識別該電腦程式內之一權利；

藉由該授權服務，決定：該權利係包含於該等複數個授權間的一特定授權中，且該權利藉由該授權服務，透過使用一授權儲存庫和於其中具有儲存的動態資料的一信賴儲存庫而執行，該動態資料係以一防竄改的方式儲存和經使用以驗證該等複數個授權；以及

藉由該授權服務，將執行該權利之一指示送回給該電腦程式，其中是否可執行該權利之決定係藉由檢查：(1) 該授權對於一電腦程式的一產品鍵值之連接；(2) 該授權對於該系統的連接，其中該電腦程式在該系統上執行；(3) 該權利尚未逾期的一有效性，及其中該電腦程式被通知決定該權利是否可被執行的結果，及基於該結果來決定該電腦程式的行為。

17. 如申請專利範圍第 16 項之電腦可讀取媒體，其中該指示包含：連接該權利至該特定授權。

18. 如申請專利範圍第 16 項之電腦可讀取媒體，其中該決定步驟係基於該權利是否經標定於該特定授權內。

19. 如申請專利範圍第 16 項之電腦可讀取媒體，其中該決定步驟係基於該特定授權是否經連接於其上正執行該電腦程式之一機器。

20. 如申請專利範圍第 16 項之電腦可讀取媒體，其中該決定步驟係基於該特定授權或該權利是否連接於該電腦程式。

21. 如申請專利範圍第 16 項之電腦可讀取媒體，其中該決定步驟係基於該特定授權或該權利是否未逾期。

22. 如申請專利範圍第 16 項之電腦可讀取媒體，其中該決定步驟係基於該特定授權是否經消費超過一限制之次數。

23. 如申請專利範圍第 16 項之電腦可讀取媒體，其中該方法更包含：

接收來自該電腦程式之一第二方法呼叫；以及

回應於該第二方法呼叫，將一可識別該電腦程式之一控制碼送回給該電腦程式；

其中該第一方法呼叫係接續於該第二方法呼叫而執行，且其中該第一方法呼叫可進一步識別該控制碼。

24. 如申請專利範圍第 16 項之電腦可讀取媒體，其中該方法進一步包含：

接收來自該電腦程式之一第二方法呼叫；以及

回應於該第二方法呼叫，將一非同步語意內容送回給該電腦程式，其中該第一方法呼叫係接續於該第二方法呼叫而執行，且識別該非同步語意內容；以及

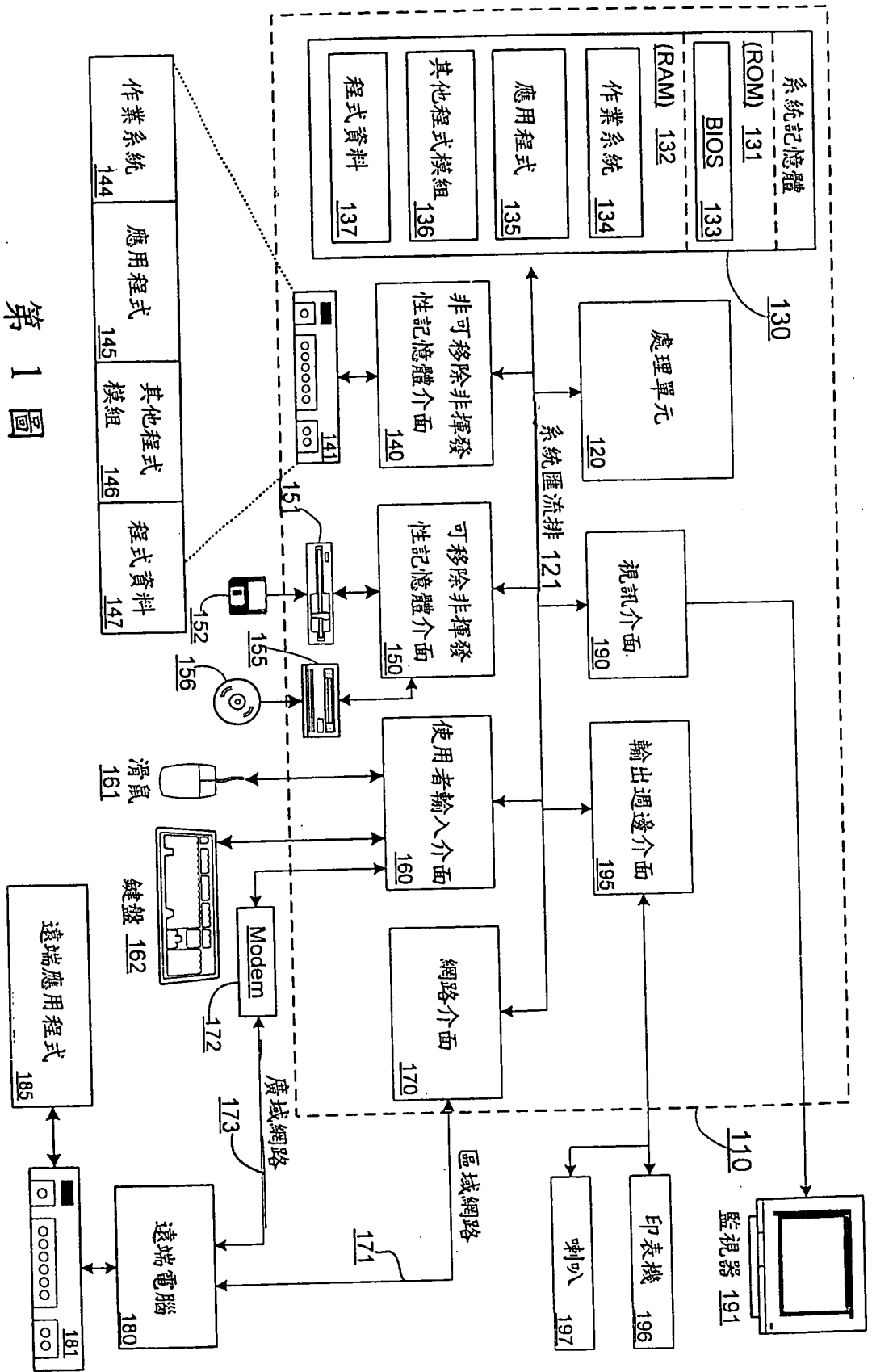
當該電腦程式執行一動作時，以非同步方式執行該第一方法呼叫。

25. 如申請專利範圍第 16 項之電腦可讀取媒體，其中該權利關聯於一組資料，且其中該方法進一步包含：

接收指示該權利之一第二方法呼叫；以及

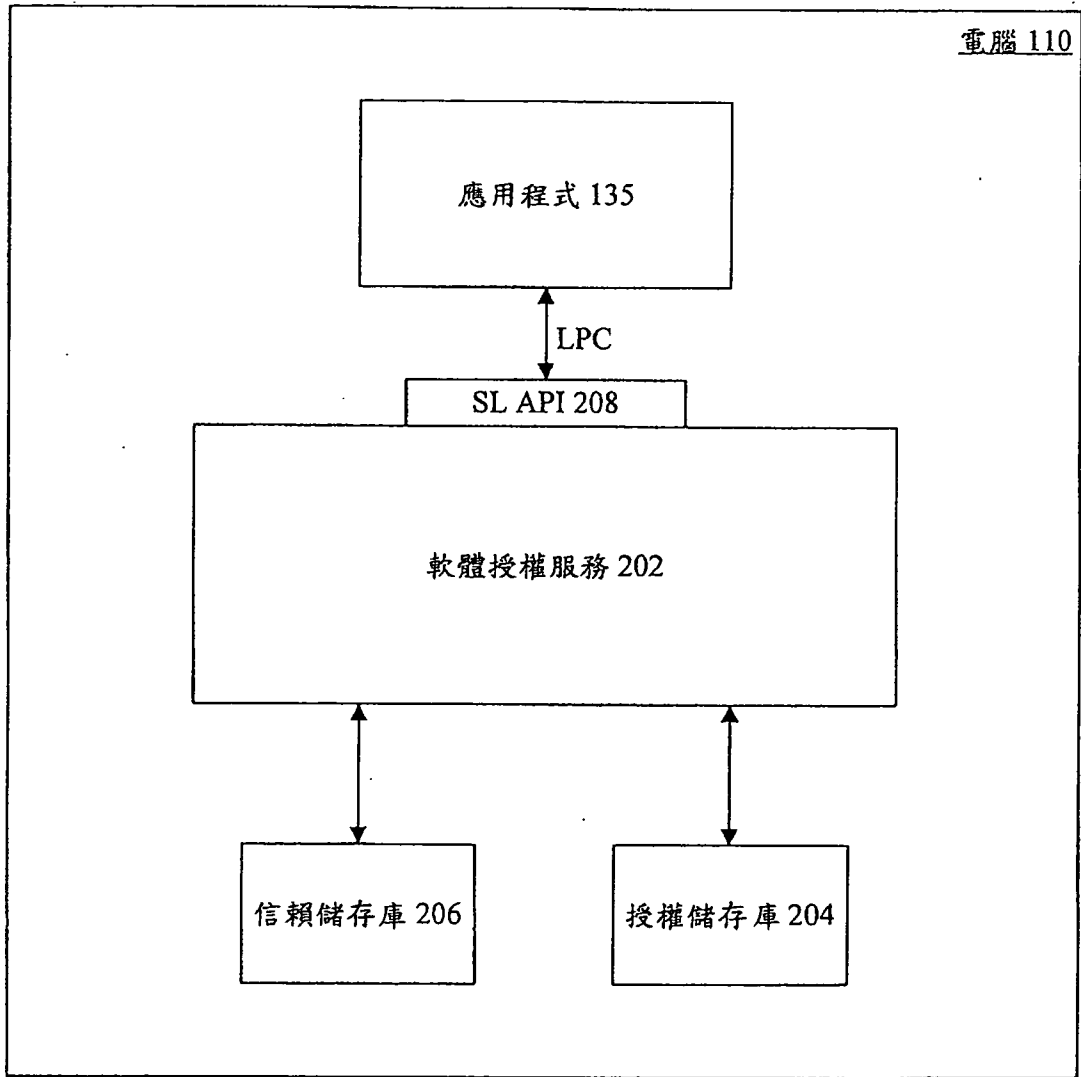
回應於該第二方法呼叫，提供該組資料給該電腦程式。

計算系統環境 100

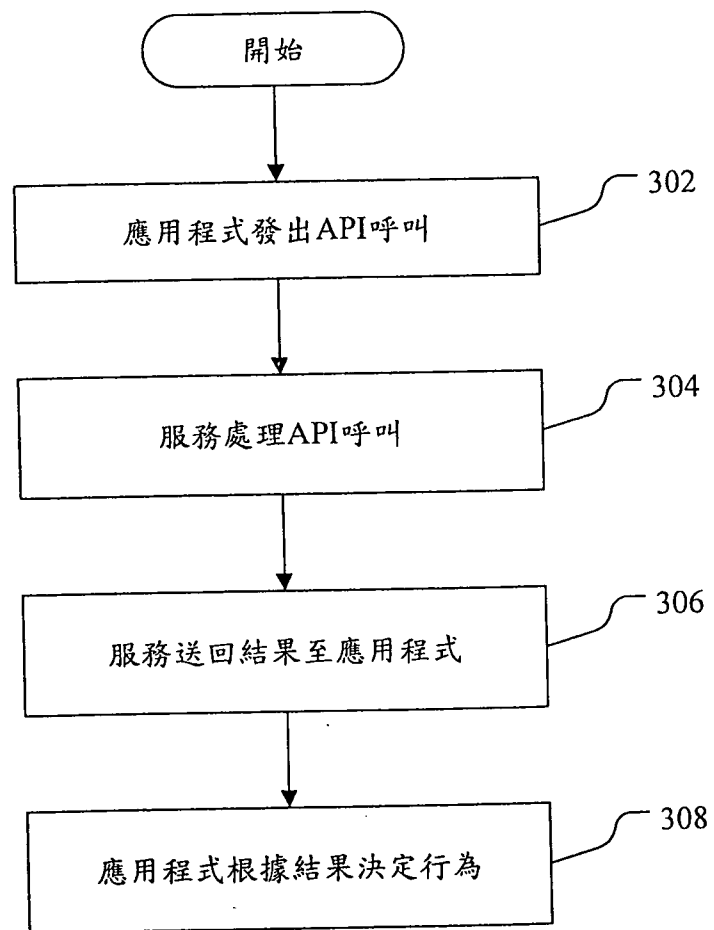


第 1 圖

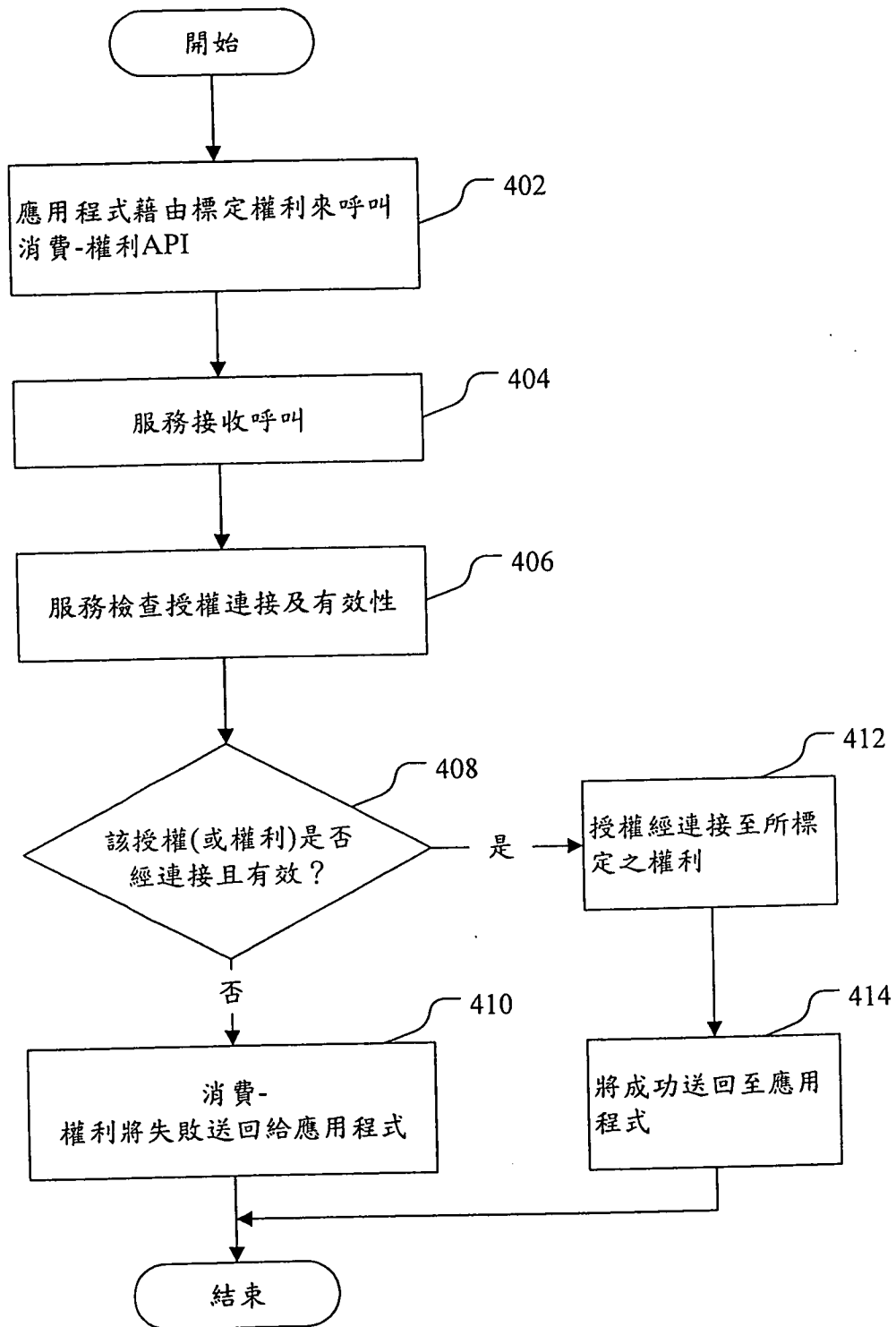
100.5.19  
年 月 日修正登錄頁



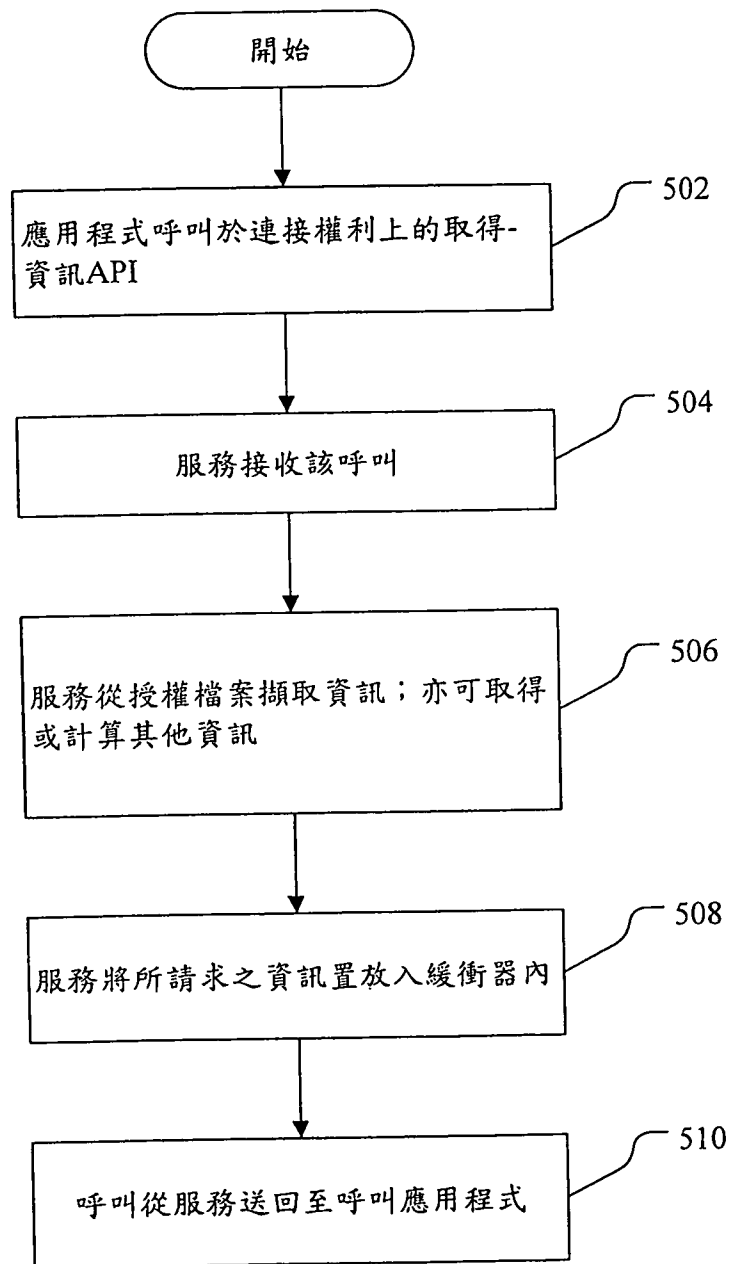
第 2 圖



第 3 圖



第 4 圖



第 5 圖