



(21)申請案號：105118399

(22)申請日：中華民國 105 (2016) 年 06 月 13 日

(51)Int. Cl.：G06F9/44 (2018.01)G06F3/0481 (2013.01)

(30)優先權：2015/12/02 中國大陸201510873960.3

(71)申請人：香港商阿里巴巴集團服務有限公司 (香港地區) ALIBABA GROUP SERVICES LIMITED (HK)
香港

(72)發明人：陳舒婷 (CN)；謝繼彬 (CN)；陸陽 (CN)；甘桂敏 (CN)；常敏 (CN)

(74)代理人：林志剛

(56)參考文獻：
CN 101025738A CN 101114226A
CN 102779034A

審查人員：朱明宗

申請專利範圍項數：9 項 圖式數：5 共 28 頁

(54)名稱
應用程式的處理方法和裝置

(57)摘要

本申請提出一種應用程式的實現方法和裝置，該應用程式的實現方法包括：將預先生成的元件保存在元件庫中；在要啟動應用程式的功能項時，根據所述應用程式的資訊和所述功能項的資訊，從服務端獲取特徵資料；根據所述特徵資料確定所述元件庫中需要使用的元件；根據所述需要使用的元件，啟動所述功能項。該方法能夠提高應用程式的實現效率並降低資源浪費。

指定代表圖：

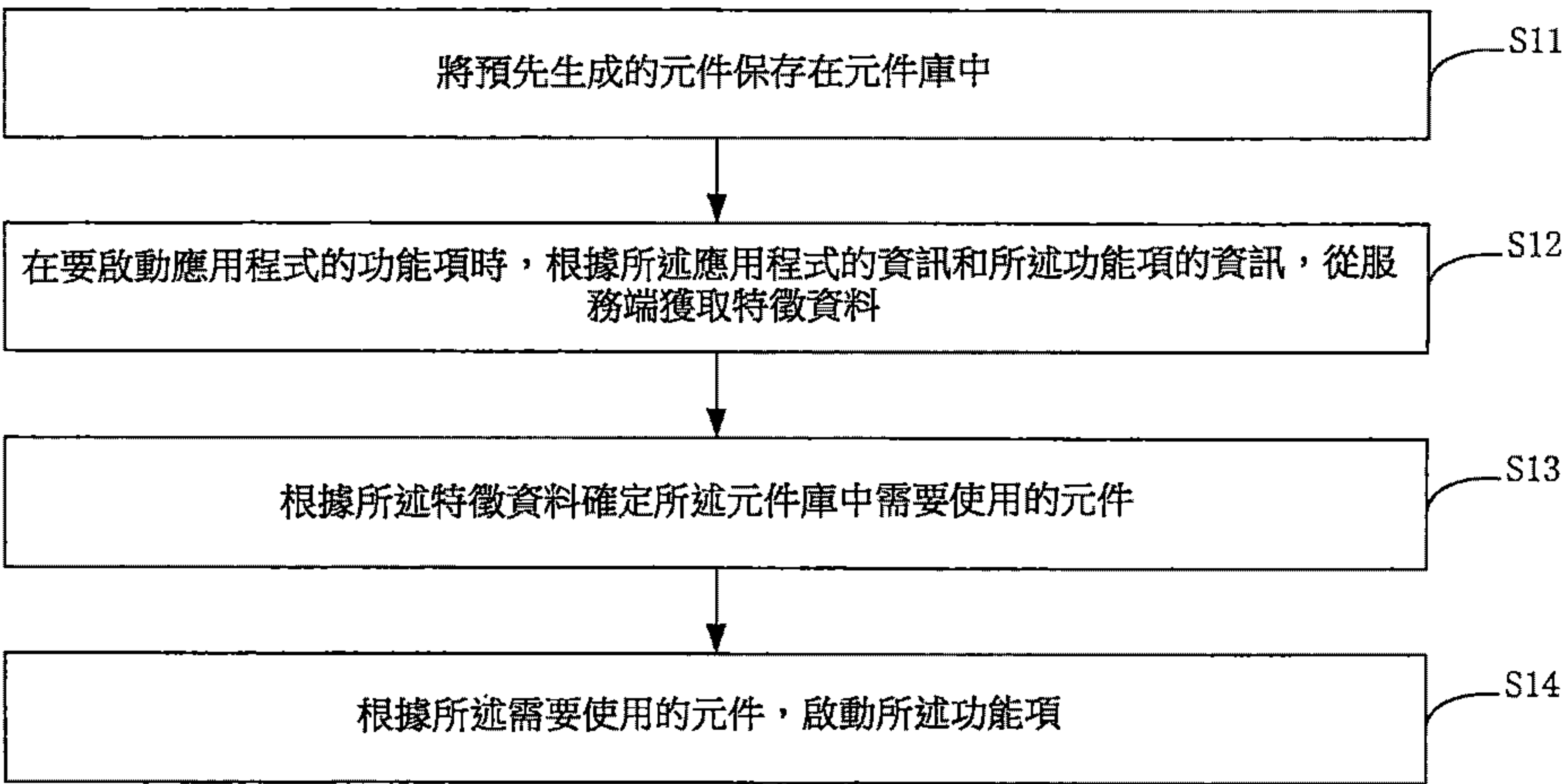


圖 1

I703497

發明摘要

※申請案號：105118399

※申請日：105 年 06 月 13 日

※IPC 分類：G06F 9/44 (2018.01)
G06F 3/0481 (2013.01)

【發明名稱】(中文/英文)

應用程式的處理方法和裝置

【中文】

本申請提出一種應用程式的實現方法和裝置，該應用程式的實現方法包括：將預先生成的元件保存在元件庫中；在要啟動應用程式的功能項時，根據所述應用程式的資訊和所述功能項的資訊，從服務端獲取特徵資料；根據所述特徵資料確定所述元件庫中需要使用的元件；根據所述需要使用的元件，啟動所述功能項。該方法能夠提高應用程式的實現效率並降低資源浪費。

【英文】

【代表圖】

【本案指定代表圖】：第(1)圖。

【本代表圖之符號簡單說明】：無

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：無

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

應用程式的處理方法和裝置

【技術領域】

本申請關於互聯網技術領域，尤其關於一種應用程式的處理方法和裝置。

【先前技術】

隨著資訊化的發展，出現了大量的應用程式（APP），其中，同一個廠商可以提供多個 APP。即使是不同的 APP，尤其是同一個廠商提供的多個 APP，可能存在相似的功能。但是，目前每個 APP 都是各自開發，這就會造成效率低下和資源浪費。

【發明內容】

本申請旨在至少在一定程度上解決相關技術中的技術問題之一。

為此，本申請的一個目的在於提出一種應用程式的處理方法，該方法可以提高應用程式的實現效率並降低資源浪費。

本申請的另一個目的在於提出一種應用程式的處理裝置。

為達到上述目的，本申請第一方面實施例提出的應用程式的處理方法，包括：將預先生成的元件保存在元件庫中；在要啟動應用程式的功能項時，根據所述應用程式的資訊和所述功能項的資訊，從服務端獲取特徵資料；根據所述特徵資料確定所述元件庫中需要使用的元件；根據所述需要使用的元件，啟動所述功能項。

本申請第一方面實施例提出的應用程式的處理方法，透過在元件庫中保存元件，可以對應不同應用程式需要使用的相同元件開發一次，不需要對應每個應用程式分別開發一次，從而可以提高應用程式的實現效率以及降低資源浪費。

為達到上述目的，本申請第二方面實施例提出的應用程式的處理裝置，包括：保存模組，用於將預先生成的元件保存在元件庫中；獲取模組，用於在要啟動應用程式的功能項時，根據所述應用程式的資訊和所述功能項的資訊，從服務端獲取特徵資料；確定模組，用於根據所述特徵資料確定所述元件庫中需要使用的元件；啟動模組，用於根據所述需要使用的元件，啟動所述功能項。

本申請第二方面實施例提出的應用程式的處理裝置，透過在元件庫中保存元件，可以對應不同應用程式需要使用的相同元件開發一次，不需要對應每個應用程式分別開發一次，從而可以提高應用程式的實現效率以及降低資源浪費。

本申請附加的方面和優點將在下面的描述中部分給

出，部分將從下面的描述中變得明顯，或透過本申請的實踐瞭解到。

【圖式簡單說明】

本申請上述的和/或附加的方面和優點從下面結合附圖對實施例的描述中將變得明顯和容易理解，其中：

圖 1 是本申請一實施例提出的應用程式的處理方法的流程示意圖；

圖 2 是本申請實施例對應的應用程式系統示意圖；

圖 3 是本申請另一實施例提出的應用程式的處理方法的流程示意圖；

圖 4 是本申請另一實施例提出的應用程式的處理裝置的結構示意圖；

圖 5 是本申請另一實施例提出的應用程式的處理裝置的結構示意圖。

【實施方式】

下面詳細描述本申請的實施例，所述實施例的示例在附圖中示出，其中自始至終相同或類似的標號表示相同或類似的模組或具有相同或類似功能的模組。下面透過參考附圖描述的實施例是示例性的，僅用於解釋本申請，而不能理解為對本申請的限制。相反，本申請的實施例包括落入所附加權利要求書的精神和內涵範圍內的所有變化、修改和等同物。

圖 1 是本申請一實施例提出的應用程式的處理方法的流程示意圖，該方法包括：

S11：將預先生成的元件保存在元件庫中。

參見圖 2，本實施例對應的應用程式系統包括用戶端程式（APP）和服務端程式。用戶端程式可以包括業務包、元件庫、核心庫、資料層、適配層和應用環境。

業務包是 APP 的私有包，包含運行在 APP 上的對應該 APP 業務特性的邏輯處理常式。不同的 APP 可以具有對應不同業務特性的業務包。透過不同的業務包，可以將相同的元件透過不同的樣式、佈局展示出來。

元件庫、核心庫和資料層能夠被不同的 APP 公用。其中，元件庫包含了展示在各個 APP 上的萬用群組件。核心庫包含一些基礎庫，例如：事件中心、主題庫、範本快取等。資料層主要包含介面資料的定義，是資料通信基礎。可以理解的是，雖然圖 2 中未示出，但是，不同 APP 可以公用的模組還可以包括其他模組，如流程管理模組和生命週期管理模組。

適配層是保證元件庫、核心庫等代碼能夠成功運行在各個 APP 環境的核心。其主要功能是在業務層（包括業務包、組件庫、核心庫和資料層等）與 APP 層（即應用環境）之間搭建一個橋樑，讓元件庫、核心庫能夠成功運行在不同的 APP 應用環境之上，無需破壞原有 APP 的設計與特性。適配層一般在業務層初始化的時候透過依賴註冊的方式，將協定與不同 APP 特殊實現進行綁定。業務

層透過調用適配層定義的介面，間接使用 APP 的特殊實現。

應用環境包括一個或多個 APP，多個 APP 可以位於相同或不同的移動終端上，APP 不同的 APP 具有不同的網路出口、使用者行為檢測等。

服務端用於為用戶端提供資料。為了實現不同的 APP 透過統一介面從服務端獲取資料，在服務端設置有適配層。服務端的適配層回應 APP 的請求並返回相應於該 APP 的資料。服務端的資料可以分為靜態資料和動態資料，靜態資料例如是固定不變的資料，動態資料例如是根據不同情況變化的資料。

本實施例中，以元件是使用者介面（User Interface，UI）元件為例，相應的，元件可以保存在用戶端的元件庫中。

以一個公共元件為例，假設該公共元件的鍵值設置為 A，則可以對應 A 設置相應的元件實現類，假設該公共元件的元件實現類用 B 表示，則可以在元件庫中保存鍵值 A 的具體內容和元件實現類 B 的具體內容，並建立鍵值 A 和元件實現類 B 之間的關聯關係。具體的，鍵值 A 與元件實現類 B 之間的關聯關係可以記錄在元件庫內的元件映射配置表中。

與現有技術中各 APP 各自開發不同的是，公共元件 A 可以被不同的 APP 使用，從而可以避免重複開發和資源浪費。

S12：在要啟動應用程式的功能項時，根據所述應用程式的資訊和所述功能項的資訊，從服務端獲取特徵資料。

其中，特徵資料可以包括靜態資料和/或動態資料，不同的 APP 可以具有不同的特徵資料。

根據要啟動的功能項的不同，特徵資料也相應是不同的，例如，在要進行頁面內容展示時，特徵資料可以包括：與待展示的頁面內容相關聯的介面資料，以及頁面展示的佈局範本資料。以待展示的頁面內容是商品詳情為例，介面資料例如包括：商品圖片、商品標題、商品價格、商品所在地等商品詳情相關的資訊。佈局範本資料用於描述頁面展示的佈局，例如包括需要使用哪些元件以及這些需要使用的元件的展示順序等。

一些實施例中，所述根據所述應用程式的資訊和所述功能項的資訊，從服務端獲取特徵資料，包括：

向服務端的適配層發送請求消息，所述請求消息中包含所述應用程式的資訊和所述功能項的資訊；

接收服務端發送的特徵資料，所述特徵資料是所述服務端根據所述應用程式的資訊和所述功能項的資訊獲取的；

其中，所述服務端的適配層能夠接收不同應用程式發送的請求消息。

例如，參見圖 2，在服務端設置有適配層，不同 APP 可以透過用戶端的資料層向服務端的統一的適配層發送請

求消息，服務端中可以預先配置應用程式的資訊、功能項的資訊及特徵資料之間的關聯關係，以根據應用程式的資訊和功能項的資訊獲取相應的特徵資料。應用程式的資訊例如包括：應用程式的名稱和版本號，在商品詳情頁展示時，功能項的資訊例如是要展示的商品的 ID。

S13：根據所述特徵資料確定所述元件庫中需要使用的元件。

例如，特徵資料中可以包括需要使用的元件的資訊，從而確定出需要使用的元件。

S14：根據所述需要使用的元件，啟動所述功能項。

在確定出需要使用的元件後，可以採用元件庫中保存的具體的元件實現類實現元件的功能，以及，在需要使用的元件是多個時，組合多個需要使用的元件的功能，從而啟動應用程式的功能項。

本實施例中，透過在元件庫中保存元件，可以對應不同應用程式需要使用的相同元件開發一次，不需要對應每個應用程式分別開發一次，從而可以提高應用程式的實現效率以及降低資源浪費。

圖 3 是本申請另一實施例提出的應用程式的處理方法的流程示意圖，該方法以要啟動的應用程式的功能項是展示應用程式內的商品詳情頁為例。該方法包括：

S31：將預先生成的元件保存在元件庫中，所述元件能被不同的應用程式使用。

例如，預先生成的元件的鍵值是 A，元件的實現類是

B，則可以在元件庫中保存鍵值 A 和元件實現類 B，並建立鍵值 A 和元件實現類 B 之間的關聯關係。

S32：建立元件與視圖資料模型之間的關聯關係。

視圖資料模型（view model）用於視圖渲染，包含了 UI 組件的視圖渲染所需的全部資訊。

具體的，可以在資料層中建立元件與視圖資料模型之間的關聯關係，例如，鍵值 A 對應視圖資料模型 C。

S33：資料層接收應用程式的頁面載入器發送的請求消息，所述請求消息中包含應用程式的資訊和商品 ID。

應用程式的資訊例如包括應用程式的名稱和版本號。

商品 ID 是待展示詳情頁的商品的 ID。

S34：向服務端的適配層發送請求消息，所述請求消息中包含所述應用程式的資訊和商品 ID。

當資料層接收到頁面載入器發送的請求消息後，可以確定要啟動應用程式的商品詳情頁，並向服務端的適配層發送請求消息，以獲取特徵資料。

S35：接收服務端發送的介面資料和佈局範本資料，所述介面資料和佈局範本資料是所述服務端根據所述應用程式的資訊和商品 ID 獲取的。

其中，服務端可以預先關聯保存商品 ID 與介面資料，從而可以根據商品 ID 獲取到相應的介面資料，介面資料例如包括：商品標題、價格、圖片等商品詳情相關資訊。

另外，服務端還可以預先關聯保存應用程式的資訊與

佈局範本資料，從而可以根據應用程式的資訊獲取到相應的佈局範本資料。

服務端在預先保存的資料中獲取介面資料和佈局範本資料後，可以將其發送給用戶端的資料層。

S36：用戶端根據佈局範本資料確定需要使用的元件。

例如，佈局範本資料中包含需要使用的元件的資訊，如需要使用哪些元件以元件的排列順序等，從而可以確定出需要使用的元件。

S37：根據所述需要使用的元件，以及所述元件與視圖資料模型之間的關聯關係，確定與所述需要使用的元件對應的初始的視圖資料模型。

例如，需要使用的元件的鍵值是 A，則由於預先建立的關聯關係中與 A 對應的視圖資料模型是 C，則可以確定初始的視圖資料模型是 C。

S38：根據所述佈局範本資料和所述介面資料更新所述初始的視圖資料模型，得到更新後的視圖資料模型。

其中，初始的視圖資料模型中具體的數值可以是空，經過佈局範本資料和介面資料的處理，可以填充具體的數值，如填充商品價格、商品標題和商品圖片等，從而得到更新後的視圖資料模型。

S39：採用更新後的視圖資料模型進行頁面渲染，展示渲染後的頁面。

由於更新後的視圖資料模型中包含了渲染所需的全部

資訊，因此，根據更新後的視圖資料模型在渲染後，可以得到商品詳情頁的具體內容，並展示給具體內容。

一些實施例中，當網路環境比較差時，可能會獲取不到介面資料和佈局範本資料，此時，為了方便使用者看到基本資訊，則可以將渲染分為首次渲染和二次渲染，上述的根據介面資料和佈局範本資料生成更新後的視圖資料模型，並根據更新後的視圖資料模型進行的渲染為二次渲染，在此之前可以先根據初始資料進行首次渲染。

例如，在 S34 和 S35 之間還可以包括：

S35'：根據初始的佈局範本資料和基本資訊進行頁面的初始渲染，並展示初始渲染後的頁面。

其中，初始的佈局範本資料和基本資訊可以是預先配置的，以便使用者看到基本的頁面內容，如展示商品標題和價格等。

本實施例中，透過不同的應用程式公用元件，可以實現透過跨 APP 的元件化解決方案，透過一次開發，就能夠將業務快速應用到移動終端不同的 APP 上，大大的節約了研發成本，同時能夠讓業務快速同步上線。

圖 4 是本申請另一實施例提出的應用程式的處理裝置的結構示意圖，該裝置 40 包括：保存模組 41、獲取模組 42、確定模組 43 和啟動模組 44。

保存模組 41，用於將預先生成的元件保存在元件庫中，所述元件能被不同的應用程式使用；

參見圖 2，本實施例對應的應用程式系統可以包括：

用戶端程式（APP）和服務端程式，用戶端程式可以包括：業務包、元件庫、核心庫、資料層、適配層和應用環境。

業務包是 APP 的私有包，包含運行在 APP 上的對應該 APP 業務特性的邏輯處理常式。不同的 APP 可以具有對應不同業務特性的業務包。透過不同的業務包，可以將相同的元件透過不同的樣式、佈局展示出來。

元件庫、核心庫和資料層能夠被不同的 APP 公用。其中，元件庫包含了展示在各個 APP 上的萬用群組件。核心庫包含一些基礎庫，例如：事件中心、主題庫、範本緩存等。資料層主要包含介面資料的定義，是資料通信基礎。可以理解的是，雖然圖 2 中未示出，但是，不同 APP 可以公用的模組還可以包括其他模組，如流程管理模組和生命週期管理模組。

適配層是保證元件庫、核心庫等代碼能夠成功運行在各個 APP 環境的核心。其主要功能是在業務層（包括業務包、組件庫、核心庫和資料層等）與 APP 層（即應用環境）之間搭建一個橋樑，讓元件庫、核心庫能夠成功運行在不同的 APP 應用環境之上，無需破壞原有 APP 的設計與特性。適配層一般在業務層初始化的時候透過依賴註冊的方式，將協定與不同 APP 特殊實現進行綁定。業務層透過調用適配層定義的介面，間接使用 APP 的特殊實現。

應用環境包括一個或多個 APP，多個 APP 可以位於

相同或不同的移動終端上，不同的 APP 具有不同的網路出口、使用者行為檢測等。

服務端用於為用戶端提供資料。為了實現不同的 APP 透過統一介面從服務端獲取資料，在服務端設置有適配層。服務端的適配層能夠回應 APP 的請求並返回相應於該 APP 的資料。服務端的資料可以分為靜態資料和動態資料，靜態資料例如是固定不變的資料，動態資料例如是根據不同情況變化的資料。

本實施例中，以元件是使用者介面（User Interface，UI）元件為例，相應的，元件可以保存在用戶端的元件庫中。

以一個公共元件為例，假設該公共元件的鍵值設置為 A，則可以對應 A 設置相應的元件實現類，假設該元件實現類用 B 表示，則可以在元件庫中保存鍵值 A 的具體內容和元件實現類 B 的具體內容，並建立鍵值 A 和元件實現類 B 之間的關聯關係。具體的，鍵值 A 與元件實現類 B 之間的關聯關係可以記錄在元件庫內的元件映射配置表中。

與現有技術中各 APP 各自開發不同的是，公共元件 A 可以被不同的 APP 使用，從而可以避免重複開發和資源浪費。

獲取模組 42，用於在要啟動應用程式的功能項時，根據所述應用程式的資訊和所述功能項的資訊，從服務端獲取特徵資料；

其中，特徵資料可以包括靜態資料和/或動態資料，不同的 APP 可以具有不同的特徵資料。

根據要啟動的功能項的不同，特徵資料也相應是不同的，例如，在要進行頁面內容展示時，特徵資料可以包括：與待展示的頁面內容相關聯的介面資料，以及頁面展示的佈局範本資料。以待展示的頁面內容是商品詳情為例，介面資料例如包括：商品圖片、商品標題、商品價格、商品所在地等商品詳情相關的資訊。佈局範本資料用於描述頁面展示的佈局，例如包括需要使用哪些元件以及這些需要使用的元件的展示順序等。

一些實施例中，所述獲取模組 42 用於根據所述應用程式的資訊和所述功能項的資訊，從服務端獲取特徵資料，包括：

向服務端的適配層發送請求消息，所述請求消息中包含所述應用程式的資訊和所述功能項的資訊；

接收服務端發送的特徵資料，所述特徵資料是所述服務端根據所述應用程式的資訊和所述功能項的資訊獲取的；

其中，所述服務端的適配層能夠接收不同應用程式發送的請求消息。

例如，參見圖 2，在服務端設置有適配層，不同 APP 可以透過用戶端的資料層向服務端的統一的適配層發送請求消息，服務端中可以預先配置應用程式的資訊、功能項的資訊及特徵資料之間的關聯關係，以根據應用程式的資

訊和功能項的資訊獲取相應的特徵資料。應用程式的資訊例如包括：應用程式的名稱和版本號，在商品詳情頁展示時，功能項的資訊例如是要展示的商品的 ID。

確定模組 43，用於根據所述特徵資料確定所述元件庫中需要使用的元件；

例如，特徵資料中可以包括需要使用的元件的資訊，從而確定出需要使用的元件。

啟動模組 44，用於根據所述需要使用的元件，啟動所述功能項。

在確定出需要使用的元件後，可以採用元件庫中保存的具體的元件實現類實現元件的功能，以及，在需要使用的元件是多個時，組合多個需要使用的元件的功能，從而啟動應用程式的功能項。

本實施例中，透過在元件庫中保存元件，且元件能夠被不同的應用程式使用，可以對應所以應用程式需要使用的相同元件開發一次，不需要對應每個應用程式分別開發一次，從而可以提高應用程式的實現效率以及降低資源浪費。

一些實施例中，當所述功能項是頁面內容展示時，所述特徵資料包括：與待展示的頁面內容相關聯的介面資料，以及頁面展示的佈局範本資料，所述佈局範本資料中包含需要使用的元件的資訊，以確定需要使用的元件。

參見圖 5，所述裝置 40 還包括：

建立模組 45，用於建立元件與視圖資料模型之間的

關聯關係；

所述啟動模組 44 具體用於：

根據所述需要使用的元件，以及所述元件與視圖資料模型之間的關聯關係，確定與所述需要使用的元件對應的初始的視圖資料模型；

根據所述佈局範本資料和所述介面資料更新所述初始的視圖資料模型，得到更新後的視圖資料模型；

採用更新後的視圖資料模型進行頁面渲染，展示渲染後的頁面。

一些實施例中，所述建立模組 45 具體用於：

在資料層，建立元件與視圖資料模型之間的關聯關係；

視圖資料模型（view model）用於視圖渲染，包含了 UI 組件的視圖渲染所需的全部資訊。

具體的，可以在資料層中建立元件與視圖資料模型之間的關聯關係，例如，鍵值 A 對應視圖資料模型 C。

參見圖 5，該裝置 40 還包括：

接收模組 46，用於接收所述應用程式的頁面載入器發送的請求消息，所述請求消息中包含所述應用程式的資訊和所述功能項的資訊，以確定要啟動應用程式的功能項，並獲取所述應用程式的資訊和所述功能項的資訊。

應用程式的資訊例如包括應用程式的名稱和版本號。

功能項的資訊是待展示詳情頁的商品的 ID。

當資料層接收到頁面載入器發送的請求消息後，可以

確定要啟動應用程式的商品詳情頁，並向服務端的適配層發送請求消息，以獲取特徵資料。

一些實施例中，參見圖 5，該裝置 40 還包括：

初始渲染模組 47，用於根據初始的佈局範本資料和基本資訊進行頁面的初始渲染，並展示初始渲染後的頁面。

其中，初始的佈局範本資料和基本資訊可以是預先配置的，以便使用者看到基本的頁面內容，如展示商品標題和價格等。

可選的，所述元件是 UI 元件，所述元件庫和所述資料層位於用戶端，所述用戶端還包括：與應用程式對應的業務包，所述頁面載入器位於所述應用程式對應的業務包內。

可選的，所述用戶端還包括其他的能夠被不同的應用程式公用的模組，所述其他的能夠被不同的應用程式公用的模組包括如下項中的一項或多項：核心庫、流程管理模組和生命週期管理模組。

可選的，所述用戶端還包括適配層，所述適配層包括不同應用程式的實現資訊，以便資料層在渲染時根據相應應用程式的實現資訊進行渲染。

上述的具體內容可以參見方法實施例中的相關描述，在此不再贅述。

本實施例中，透過不同的應用程式公用元件，可以實現透過跨 APP 的元件化解決方案，透過一次開發，就能

夠將業務快速應用到移動終端不同的 APP 上，大大的節約了研發成本，同時能夠讓業務快速同步上線。

需要說明的是，在本申請的描述中，術語“第一”、“第二”等僅用於描述目的，而不能理解為指示或暗示相對重要性。此外，在本申請的描述中，除非另有說明，“多個”的含義是指至少兩個。

流程圖中或在此以其他方式描述的任何過程或方法描述可以被理解為，表示包括一個或更多個用於實現特定邏輯功能或過程的步驟的可執行指令的代碼的模組、片段或部分，並且本申請的較佳實施方式的範圍包括另外的實現，其中可以不按所示出或討論的順序，包括根據所涉及的功能按基本同時的方式或按相反的順序，來執行功能，這應被本申請的實施例所屬技術領域的技術人員所理解。

應當理解，本申請的各部分可以用硬體、軟體、韌體或它們的組合來實現。在上述實施方式中，多個步驟或方法可以用儲存在記憶體中且由合適的指令執行系統執行的軟體或韌體來實現。例如，如果用硬體來實現，和在另一實施方式中一樣，可用本領域公知的下列技術中的任一項或他們的組合來實現：具有用於對資料信號實現邏輯功能的邏輯門電路的離散邏輯電路，具有合適的組合邏輯門電路的專用積體電路，可程式設計閘陣列（PGA），現場可程式設計閘陣列（FPGA）等。

本技術領域的普通技術人員可以理解實現上述實施例方法攜帶的全部或部分步驟是可以透過程式來指令相關的

硬體完成，所述的程式可以儲存於一種電腦可讀儲存媒體中，該程式在執行時，包括方法實施例的步驟之一或其組合。

此外，在本申請各個實施例中的各功能單元可以整合在一個處理模組中，也可以是各個單元單獨物理存在，也可以兩個或兩個以上單元整合在一個模組中。上述整合的模組既可以採用硬體的形式實現，也可以採用軟體功能模組的形式實現。所述整合的模組如果以軟體功能模組的形式實現並作為獨立的產品銷售或使用時，也可以儲存在一個電腦可讀取儲存媒體中。

上述提到的儲存媒體可以是唯讀記憶體，磁片或光碟等。

在本說明書的描述中，參考術語“一個實施例”、“一些實施例”、“示例”、“具體示例”、或“一些示例”等的描述意指結合該實施例或示例描述的具體特徵、結構、材料或者特點包含於本申請的至少一個實施例或示例中。在本說明書中，對上述術語的示意性表述不一定指的是相同的實施例或示例。而且，描述的具體特徵、結構、材料或者特點可以在任何的一個或多個實施例或示例中以合適的方式結合。

儘管上面已經示出和描述了本申請的實施例，可以理解的是，上述實施例是示例性的，不能理解為對本申請的限制，本領域的普通技術人員在本申請的範圍內可以對上述實施例進行變化、修改、替換和變型。

【符號說明】

16：查詢請求接收模組

18：查詢模組

20：資訊建立模組

22：資訊發送模組

28：呼叫請求接收模組

30：查詢請求發送模組

32：號碼資訊接收模組

34：呼叫模組

40：號碼請求接收模組

42：號碼分配模組

44：綁定模組

46：呼叫模組

100：伺服器

110：伺服器

120：伺服器

申請專利範圍

1. 一種應用程式的處理方法，其特徵在於，包括：

將預先生成的元件保存在元件庫中；

在要啟動應用程式的功能項時，根據該應用程式的資訊和該功能項的資訊，從服務端獲取特徵資料；

根據該特徵資料確定該元件庫中需要使用的元件；

根據該需要使用的元件，啟動該功能項，

其中，所述根據該應用程式的資訊和該功能項的資訊，從該服務端獲取特徵資料，包括：

向該服務端的適配層發送請求消息，該請求消息中包含該應用程式的資訊和該功能項的資訊；

接收該服務端發送的特徵資料，該特徵資料是該服務端根據該應用程式的資訊和該功能項的資訊獲取的；

其中，該服務端的適配層能夠接收不同應用程式發送的請求消息，

其中，該服務端中係預先配置該應用程式的資訊、該功能項的資訊及特徵資料之間的關聯關係，以根據該應用程式的資訊和該功能項的資訊獲取相應的特徵資料。

2. 根據申請專利範圍第 1 項所述的方法，其中，當該功能項是頁面內容展示時，該特徵資料包括：與待展示的頁面內容相關聯的介面資料，以及頁面展示的佈局範本資料，該佈局範本資料中包含需要使用的元件的資訊，以確定需要使用的元件，該方法還包括：

建立元件與視圖資料模型之間的關聯關係；

該根據該需要使用的元件，啟動該功能項，包括：

根據該需要使用的元件，以及該元件與視圖資料模型之間的關聯關係，確定與該需要使用的元件對應的初始的視圖資料模型；

根據該佈局範本資料和該介面資料更新該初始的視圖資料模型，得到更新後的視圖資料模型；

採用更新後的視圖資料模型進行頁面渲染，展示渲染後的頁面。

3. 根據申請專利範圍第 2 項所述的方法，其中，該建立元件與視圖資料模型之間的關聯關係，包括：

在資料層，建立元件與視圖資料模型之間的關聯關係；

該方法還包括：

接收該應用程式的頁面載入器發送的請求消息，該請求消息中包含該應用程式的資訊和該功能項的資訊，以確定要啟動應用程式的功能項，並獲取該應用程式的資訊和該功能項的資訊。

4. 根據申請專利範圍第 3 項所述的方法，其中，在從服務端獲取特徵資料之前，該方法還包括：

根據初始的佈局範本資料和基本資訊進行頁面的初始渲染，並展示初始渲染後的頁面。

5. 根據申請專利範圍第 3 項所述的方法，其中，該元件是 UI 元件，該元件庫和該資料層位於用戶端，該用戶端還包括：與應用程式對應的業務包，該頁面載入器位

於該應用程式對應的業務包內。

6. 根據申請專利範圍第 5 項所述的方法，其中，該用戶端還包括其他的能夠被不同的應用程式公用的模組，該其他的能夠被不同的應用程式公用的模組包括如下項中的一項或多項：核心庫、流程管理模組和生命週期管理模組。

7. 根據申請專利範圍第 5 項所述的方法，其中，該用戶端還包括適配層，該適配層包括不同應用程式的實現資訊，以便資料層在渲染時根據相應應用程式的實現資訊進行渲染。

8. 一種應用程式的處理裝置，其特徵在於，包括：
保存模組，用於將預先生成的元件保存在元件庫中；
獲取模組，用於在要啟動應用程式的功能項時，根據該應用程式的資訊和該功能項的資訊，從服務端獲取特徵資料；

確定模組，用於根據該特徵資料確定該元件庫中需要使用的元件；

啟動模組，用於根據該需要使用的元件，啟動該功能項，

其中，所述根據該應用程式的資訊和該功能項的資訊，從該服務端獲取特徵資料，包括：

向該服務端的適配層發送請求消息，該請求消息中包含該應用程式的資訊和該功能項的資訊；

接收該服務端發送的特徵資料，該特徵資料是該服務

端根據該應用程式的資訊和該功能項的資訊獲取的；

其中，該服務端的適配層能夠接收不同應用程式發送的請求消息，

其中，該服務端中係預先配置該應用程式的資訊、該功能項的資訊及特徵資料之間的關聯關係，以根據該應用程式的資訊和該功能項的資訊獲取相應的特徵資料。

9. 根據申請專利範圍第 8 項所述的裝置，其中，當該功能項是頁面內容展示時，該特徵資料包括：與待展示的頁面內容相關聯的介面資料，以及頁面展示的佈局範本資料，該佈局範本資料中包含需要使用的元件的資訊，以確定需要使用的元件，該裝置還包括：

建立模組，用於建立元件與視圖資料模型之間的關聯關係；

該啟動模組具體用於：

根據該需要使用的元件，以及該元件與視圖資料模型之間的關聯關係，確定與該需要使用的元件對應的初始的視圖資料模型；

根據該佈局範本資料和該介面資料更新該初始的視圖資料模型，得到更新後的視圖資料模型；

採用更新後的視圖資料模型進行頁面渲染，展示渲染後的頁面。

圖 式

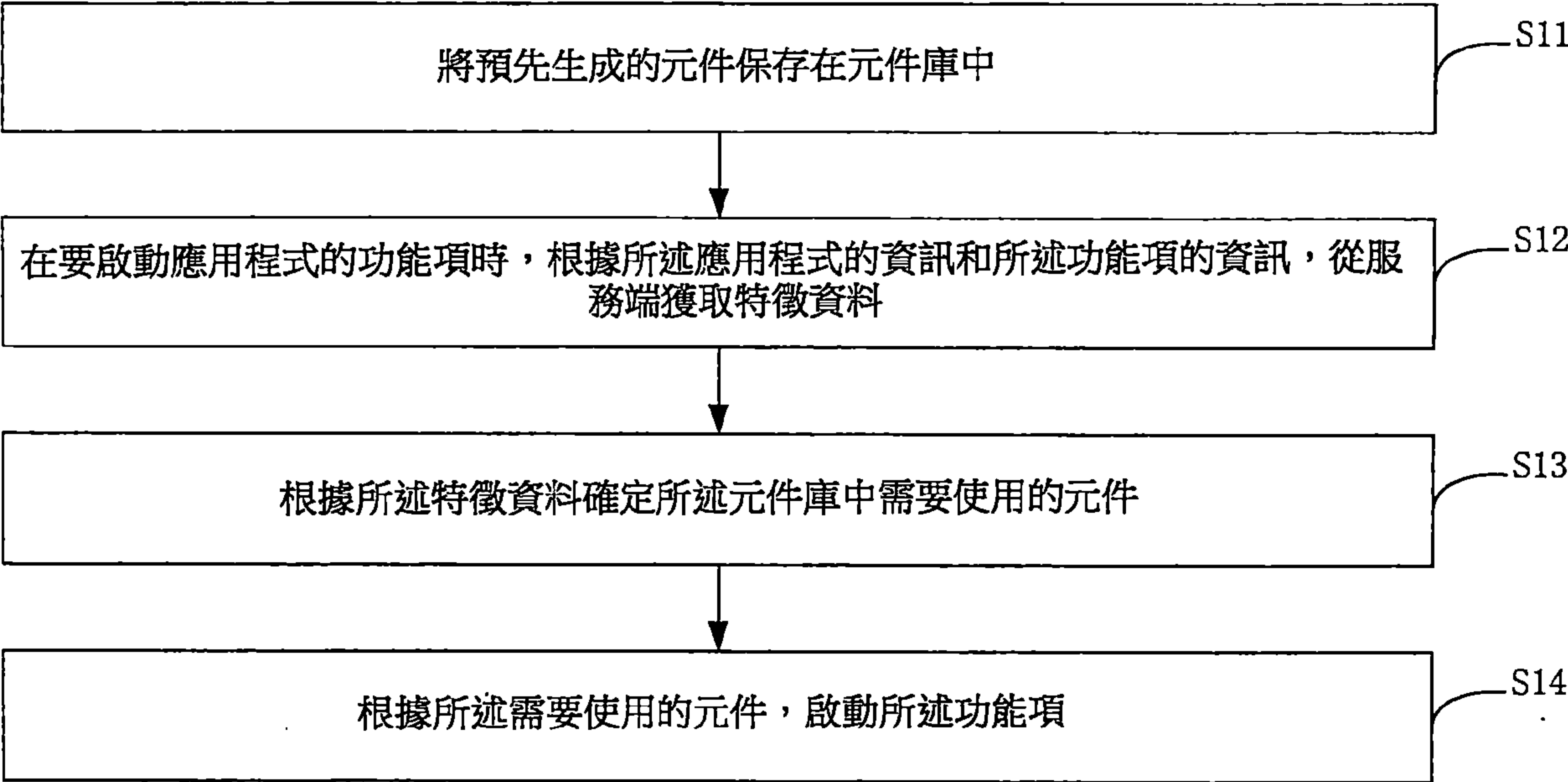


圖 1

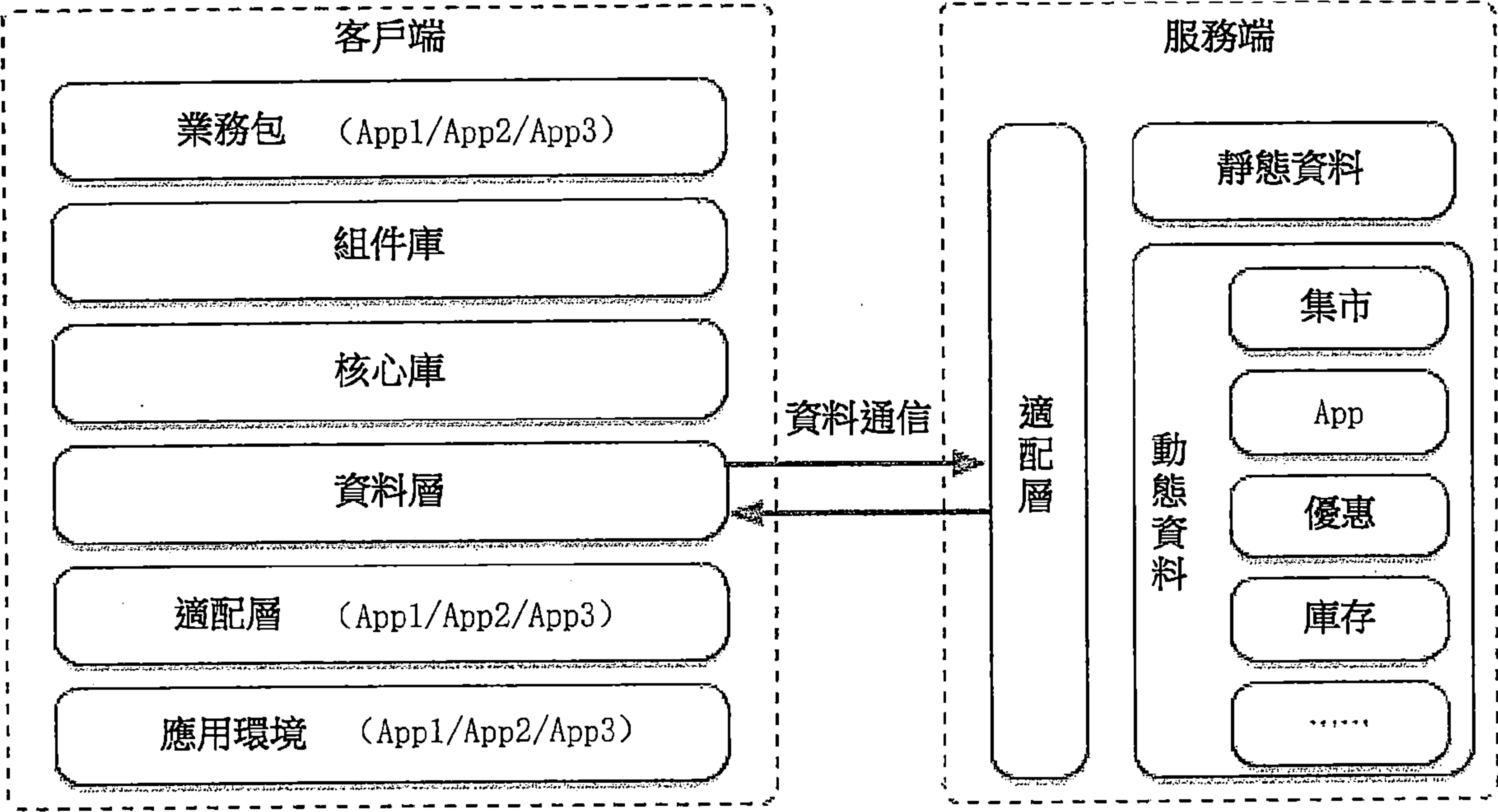


圖 2

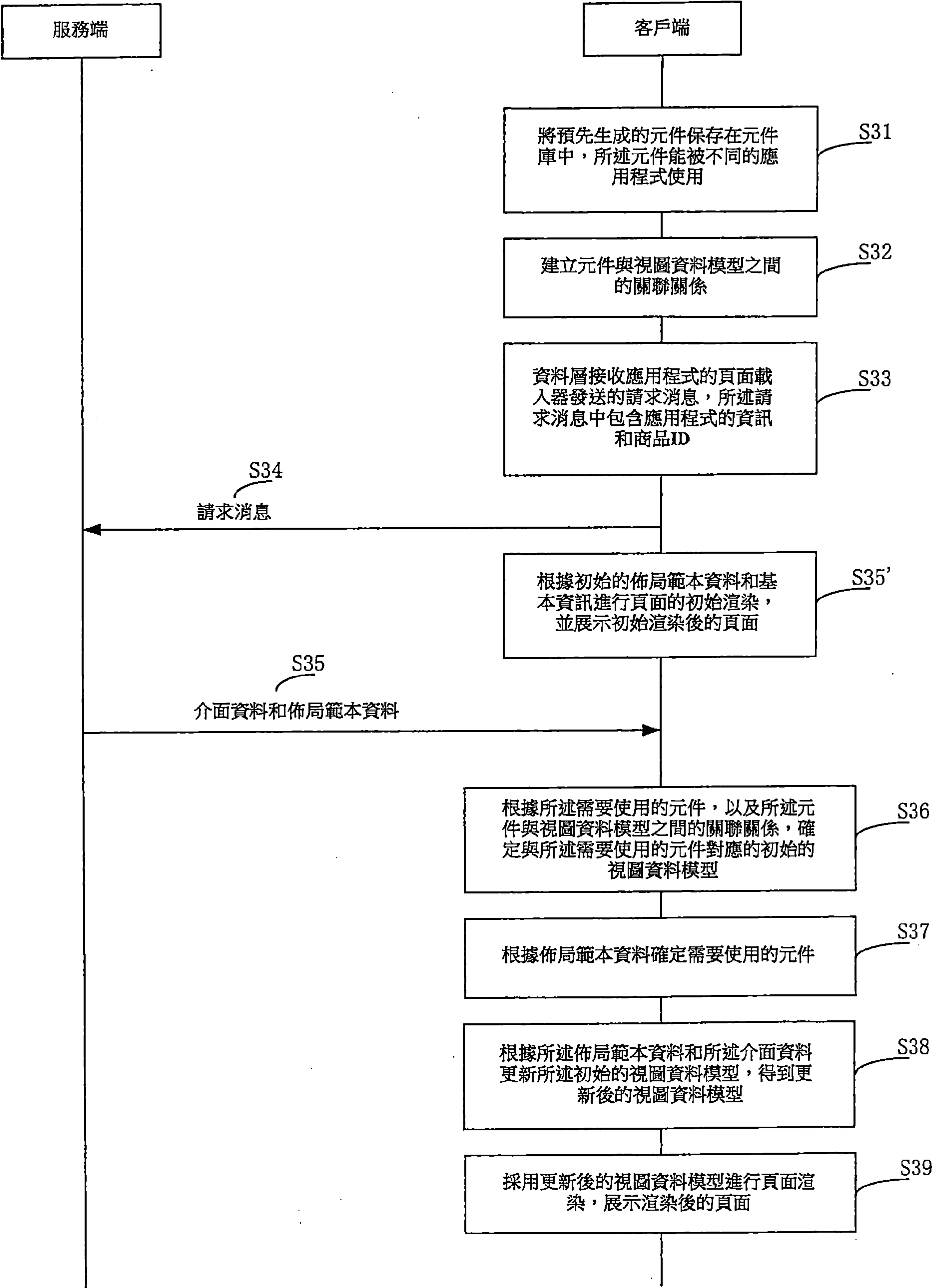


圖 3

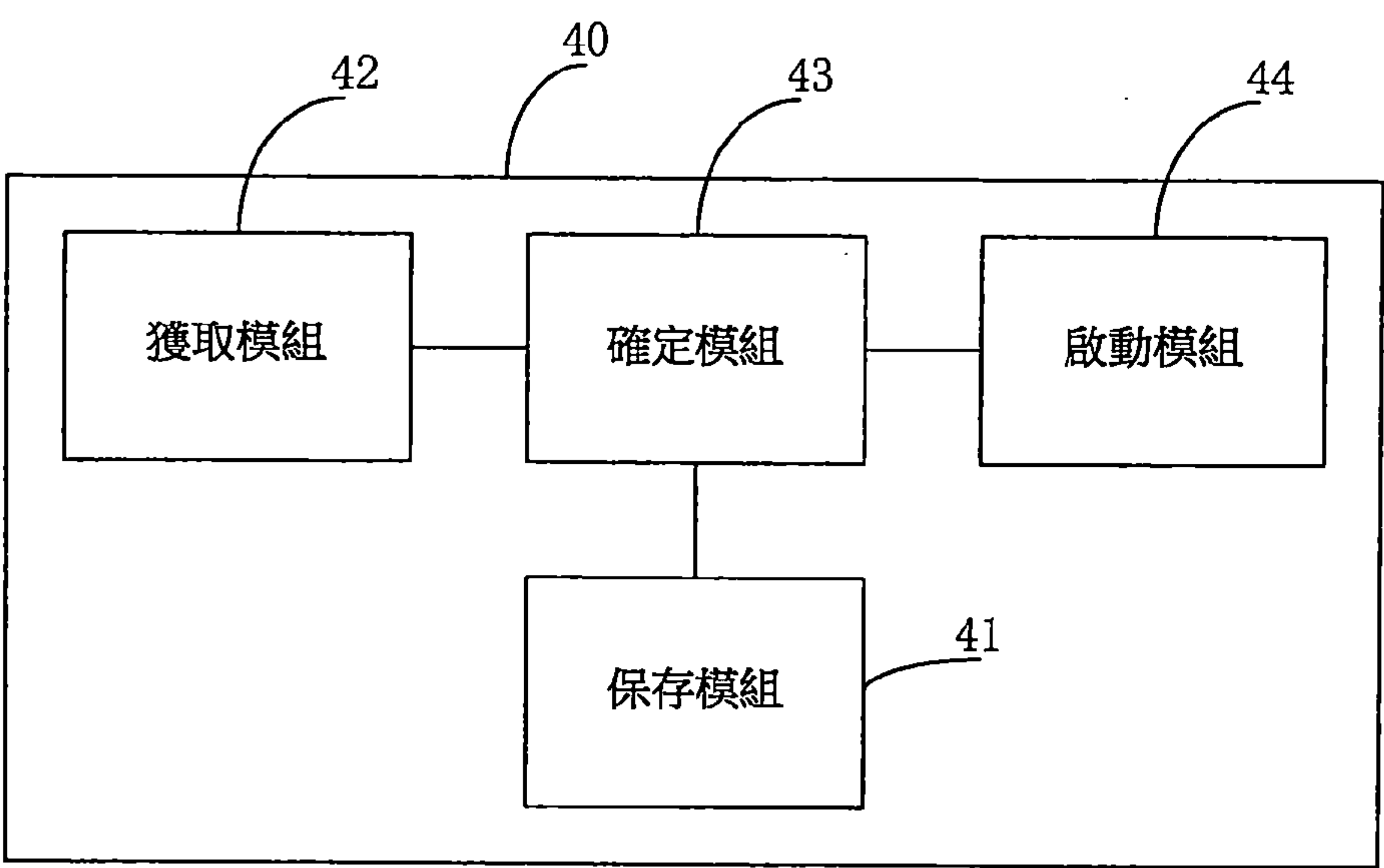


圖 4

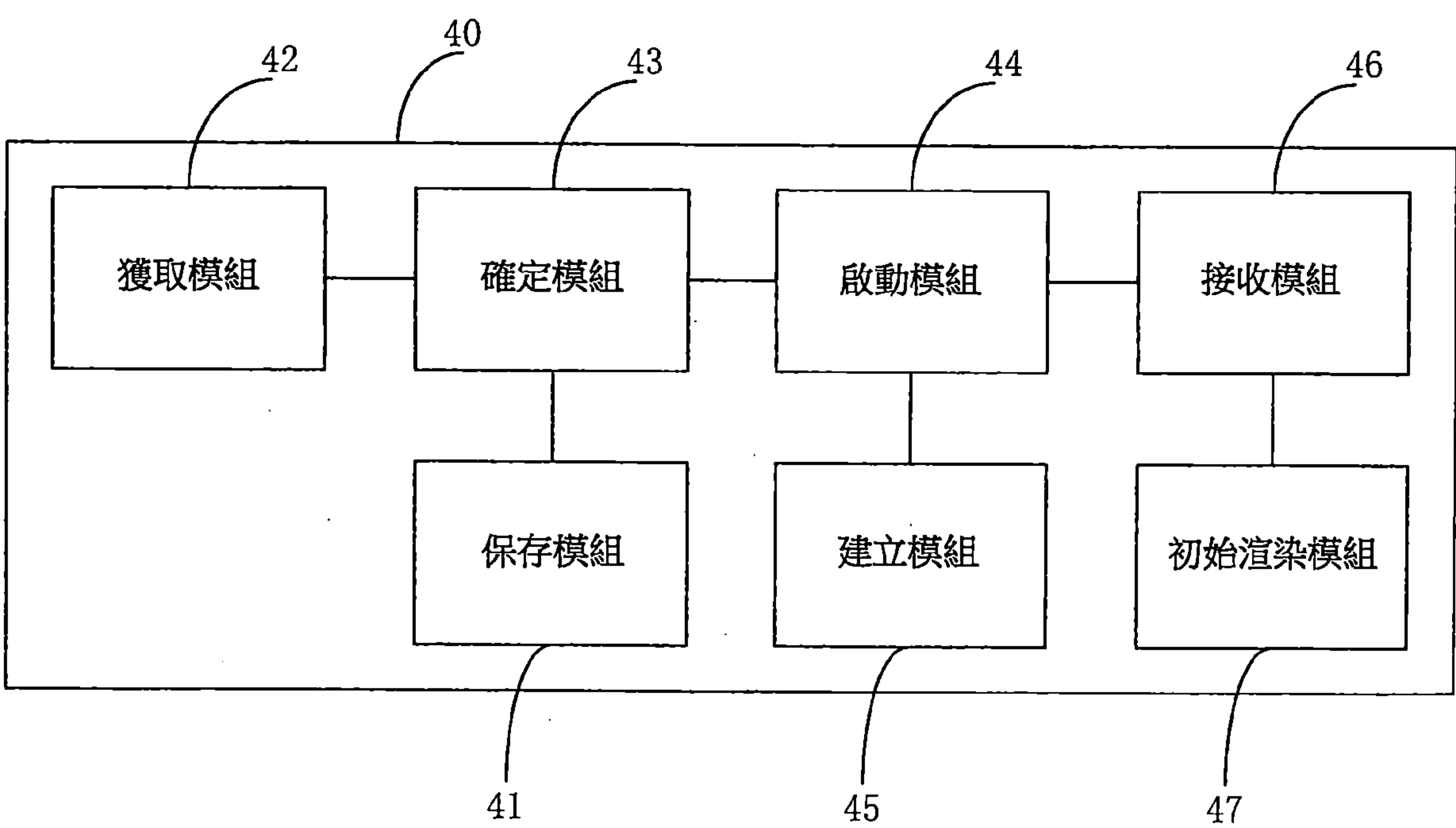


圖 5