



iSAI

手機應用雲端安全管理平台

系統整合開發技術手冊

文件編號: PTP-TG-001

文件日期: 2011/06/20

版 本: 1.2

修 訂 記 錄 表

版次	生效日期	撰 寫 人	修 訂 說 明
1.0	2010/12/03	Alvin Huang	初稿
1.1	2011/05/10	Alvin Huang	增加刪除命令格式說明
1.2	2011/06/20	Alvin Huang	新增 Priority、Keywords 欄位於 Inbox 與 Outbox 兩個資料表內。

目 錄

1	系統整合開發簡介	4
1.1	系統架構說明	4
1.2	軟體開發環境	5
1.3	網路環境	6
2	訊息的傳送與接收	7
2.1	訊息推播的運作流程	7
2.2	訊息資料表溝通介面	8
2.3	訊息頻道(Channel)	8
2.4	訊息內文(Content)與內文類型(ContentType)	9
3	附錄	12

1 系統整合開發簡介

手機應用雲端安全管理平台，(以下簡稱 iSAI)，除了具備手機安全管理能力外，也同時擁有訊息推播的處理機制，提供外部的應用程式或系統可以透過 iSAI 傳遞接收專屬的頻道訊息資訊。本文件用來說明 iSAI 系統開發與整合相關所需要具備的技術資訊。

1.1 系統架構說明

iSAI 之設計架構規劃為雲端平台，系統建置人員可以透過 iSAI 的主控台界面，為不同企業組織產生各自獨立的裝置管理平台及資料庫。每個組織的裝置管理平台也擁有專屬的服務元件與功能模組，用來服務客戶端系統或行動裝置的應用程式。其系統架構如下圖所示：



■ iSAI Console – iSAI 系統主控台

系統建置人員可以藉由此主控台界面，建立/發佈一個專屬於某個組織的裝置安全管理平台，管理該平台所具備的功能模組權限、授權的用戶數量、系統參數...等。

■ iSAI for Organization – 裝置安全管理平台

一個專屬於特定組織的裝置安全管理平台，具備有下列的元件模組：

- **MDM-Mobile Device Management:** 提供系統管理者進行公告發佈、裝置管理、使用者管理、權限設定、訊息追蹤、軟體發佈的網頁管理界面。
- **PSE- Push Server Engine :**負責傳遞裝置管理平台的各種訊息資訊，透過此元件的運作用戶端的訊息才能正確的傳遞給不同的裝置與使用者。

- PSS- Push Server Service：裝置管理平台與外部裝置或系統的訊息傳送與接收，統一透過此網路服務(Web Service)進行溝通。
- ORG DB：專屬於此特定組織的平台資料庫。

■ iSAI Mobile

iSAI 在每一個支援的行動裝置平台(Windows Mobile、Android、iPhone)皆有一個裝置管理的代理人(PMA, Push Mobile Agent)，主要負責訊息的傳送與接收，並依據各個行動裝置平台所支援的處理能力，執行 MDM 所發佈的管理命令。

訊息資訊中會包含頻道代碼，但是 PMA 對於訊息的傳送與接收，基本上只負責確認接收或傳送的完成與否。在訊息接收完成後，會記錄於 PMA 所處的訊息資料庫(MSGDB)內，再各自對應的頻道處理應用程式進行處理。其他應用程式若是要傳送訊息，只要將訊息儲存至 PMA 的訊息資料庫內，PMA 會自動偵測並傳送到伺服器。

■ iSAI Client

iSAI 也支援 Windows 作業系統，同樣也具備一個代理人(PCA, Push Client Agent)負責傳送與接收訊息，但是對象不是一般的使用者，而是需要傳遞訊息至行動裝置端的異質應用系統(例如:BPM、EIP、ERP...等)。PCA 提供以資料庫及 XML 文檔為基礎的溝通界面，可降低整合上的難度。

■ Client System

需要傳遞訊息至行動裝置端的異質應用系統(例如:BPM、EIP、ERP...等)。

■ Mobile App

行動裝置端具備處理 iSAI 頻道資訊的應用程式。

1.2 軟體開發環境

iSAI 系統平台係採用公用的服務標準與介面，作為系統平台內外應用服務的存取接口。針對欲在伺服器端整合的異質系統，可以透過 Web Service 或資料庫存取系統資訊。在用戶端與行動裝置端的系統整合，

則統一透過資料庫作為溝通介面。

在公開的標準服務接口下，開發人員除了要依循不同平台的環境限制外，可以自行決定使用合適的開發工具/軟體/程式語言。進行軟體開發。

下面是 iSAI 系統建議的軟體開發環境說明：

1.2.1 伺服器端/客戶端

作業系統: Windows Server

開發工具: VS.NET / C# / ASP.NET

資料庫: MSSQL

1.2.2 Windows Mobile 行動裝置端

作業系統: Windows Mobile 5 / 6 / 6.1 / 6.5

開發工具: VS.NET / C# / .NETCF

資料庫: SQLCE 3.5

1.2.3 Android 行動裝置端

作業系統: Android 2.0 / 2.1 / 2.2

開發工具: Eclipse / Java / Android SDK

資料庫: SQLite

1.2.4 iPhone 行動裝置端

作業系統: iOS

開發工具: Xcode / Interface Builder / Object-C

資料庫: SQLite

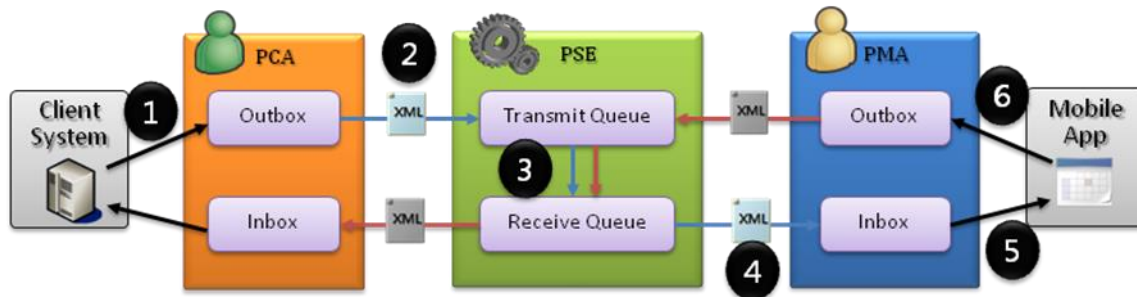
1.3 網路環境

此系統平台的裝置管理機制必需仰賴網路進行通訊。

2 訊息的傳送與接收

訊息推播是作為 iSAI 系統平台管理機制與頻道應用服務擴充的重要功能模組。本章節針對訊息推播的相關技術規格進行說明，以利開發人員進行整合。

2.1 訊息推播的運作流程



流程說明如下：

1. 客戶端的系統將欲傳送的訊息資訊，寫入 PCA 所屬的 Outbox 資料表內，PCA 會自動將訊息傳送至伺服器端。要注意的是，若是訊息本身包含附件檔案資訊，檔案所在位置的實體路徑必須要讓 PCA 具備有讀取的權限，方可正常運作。
2. PCA 會主動將 Outbox 中尚未傳送的訊息，序列化為 XML 格式後，加密編碼後傳送至伺服器端，解密還原訊息資訊後，將其記錄於資料表 TransmitQueue(傳送佇列)中，由 PSE 進行後續處理。
3. PSE 會自動讀取 TransmitQueue 的訊息，並依照訊息資訊中的收件人列表，將訊息分派至各個收件人專屬的 ReceiveQueue(接收佇列)中。
4. 在 ReceiveQueue 中的訊息資訊，會等待 PMA 主動抓取，同樣經過訊息加解密的處理流程，記錄於行動裝置端的 Inbox 資料表內，等待 Mobile App 進行處理。
5. Mobile App 會依照各自的應用處理流程，進行訊息資料的解析處理與呈現。
6. Mobile App 發送訊息的流程與上述的步驟相同，只是接收對象為特定的 PCA 帳號，例如:由 PCA 發送訊息給 10 個不同的使用者，各個使用者的回傳的訊息資訊中，其收件人為 PCA 的帳號，利用此方法 PCA 可以取得這 10 個使用者的回覆資訊。

2.2 訊息資料表溝通介面

在上述的流程中，為了降低異質系統之間系統整合的複雜度，主要的溝通介面是以資料庫做為與異質系統的溝通介面。有下列相關的資料表：

- **Outbox (OutboxAttachment)**：負責記錄傳送訊息(及附件)的資料表。
當後端系統需要傳送訊息給特定的使用者帳號時，只要依照資料表規格，新增一筆記錄到 Outbox 資料表，PCA 便會自動讀取尚未傳送的訊息，將訊息送達 PSE。同時為了客戶端系統整合的方便性，OutboxAttachment 資料表所記錄的附件檔資訊，並不需要客戶端的系統寫入，只要將附件檔存放在 PCA 主機可存取的目錄路徑下即可，PCA 會自動處理相關的資料表對應動作。
- **Inbox (InboxAttachment)**：負責記錄接收訊息(及附件)的資料表，PCA 會主動與 PS 進行通訊，當有符合 PCA 帳號的新訊息時，便自動下載訊息資訊及附件檔至 Inbox 及 InboxAttachment 兩個資料表內。企業後端系統，需要自行定時讀取 Inbox 檢查是否有新的訊息資訊，並在讀取訊息後，將狀態更新為已讀取。
- 在伺服器端的 TransmitQueue & ReceiveQueue 兩個資料表並不開放異質直接存取。

上述相關系統整合用之資料表欄位的定義請參照附錄的資料表欄位說明。

2.3 訊息頻道(Channel)

訊息資訊本身為了區分不同的應用情境，舉例來說，簽核處理與公告訊息，兩者所需要的資訊內容與處理流程是不同的，但是訊息的傳送與接收卻是相同的。因此，系統利用訊息頻道的分流機制，透過定義不同的頻道代碼(ChannelCode)，例如:BPM 代表簽核訊息，BULLTIN 代表系

統公告。

2.4 訊息內文(Content)與內文類型(ContentType)

訊息透過頻道代碼分流機制處理後，其對應的頻道應用程式可以透過訊息內文類型的資訊，決定訊息內文該如何解析處理。例如:ContentType 為 MSG 類型，表示訊息內文為訊息資訊，若是 REPLY 類型，則表示訊息的內文是某個訊息的回覆資訊。

以 BPM 的簽核表單來說，訊息內文的格式可定義為下列的 XML，用來檢視某一份申請單的簽核內容、簽核歷程及簽核結果。

```
<?xml version="1.0" encoding="utf-8" ?>
<content contentType="MSG">
  <!-- <params> 用來識別此表單的資訊，可用此節點的參數紀錄，作為頻道應用程式判斷處理的依據-->
  <params>
    <param name="sheetNo" value="AF-20101010-01"/>
    <param name="sheetType" value="AbsentForm"/>
  </params>
  <!-- <view> 用來呈現簽核表單內容的節點定義，可用page節點進行分頁。-->
  <view>
    <page id="page1" name="摘要">
      Free Text (可以是 HTML Code or Xml)
    </page>
    <page id="page2" name="表單資訊">
      Free Text (可以是 HTML Code or Xml)
    </page>
    <page id="page3" name="簽核歷程">
      Free Text (可以是 HTML Code or Xml)
    </page>
  </view>
  <!-- <replyInfo> 用來定義使用者及頻道應用程式該如何呈現及處理簽核的回覆內容。 -->
  <replyInfo>
    <!--isNullable=true表示此欄位允許回覆的內容為空值-->
    <input id="comment" name="簽核意見" isNullable="true" type="TextBox" defaultValue="無意見"/>
    <!--解析與呈現的程式自行處理Submit型態的輸入界面(例如:button or list)·
    當使用者點選此類型的輸入選擇時，就表示回覆的資訊確認並傳送回覆的訊息-->
    <input id="result" type="Submit" defaultValue=" Approve">
      <option id="btn1" name="同意" value="Approve"/>
      <option id="btn2" name="不同意" value="Disapprove"/>
      <option id="btn3" name="退回" value="Reject"/>
      <option id="btn4" name="放棄" value="Drop"/>
    </input>
  </replyInfo>
</content>
```

上述的 XML 的訊息內文，送達手機端的應用程式後，會交由對應的頻道應用程式負責解析其內容，顯示於手機端的使用者操作介面，供使用者進行簽核後，將簽核結果再次透過 PSE 回傳給 PCA 進行訊息接收。

```
<?xml version="1.0" encoding="utf-8" ?>
<content contentType="REPLY">
  <!-- <params> 用來識別此表單的資訊，可用此節點的參數紀錄，作為頻道應用程式判斷處理的依據-->
  <params>
    <param name="sheetNo" value="AF-20101010-01"/>
    <param name="sheetType" value="AbsentForm"/>
  </params>
  <!--使用者簽核的結果，會依照<replyInfo>所定義的資訊，記錄於submitInfo的對應result欄位-->
  <submitInfo>
    <result id="result" name="簽核結果" value="Approve"/>
    <result id="comment" name="簽核意見" value="No problem"/>
  </submitInfo>
</content>
```

訊息回收機制，當客戶端系統欲將發送過的訊息進行回收時，必須先取得 Outbox 資料表的 MessageId，此欄位是訊息由 Outbox 正確發送後，由伺服器端回傳的訊息代碼。依據此 MessageId，客戶端需要另外產生一筆新訊息，寫入 Outbox 的 Content 欄位。

```
<?xml version="1.0" encoding="utf-8" ?>
<content contentType="RECALL">
  <params>
    <param name="RecallMessageId" value=" 02b2ae73-5b88-4cfe-81a8-34f514471355"/>
  </params>
</content>
```

回收訊息填寫 Outbox 欄位的範例如下：

欄位名稱	欄位資料	備註
Creator	Alvin	
CreateDate	2011/05/10 15:07:00	
TxMessageId	c7cd8812-8c0f-4c5e-a8e6-402aa6f44534	程式自動產生
ChannelCode	SYS	
SourceDomain	ase	
Sender	pcagent	
ReceiverType	U	
ReceiverList	pseReceiver	設定為伺服器端的接收代理人
BccList		
Subject	[Recall]: 02b2ae73-5b88-4cfe-81a8-34f514471355	非必要，僅用來參考。
Content	填入上述的 XML
ContentType	RECALL	
AttachmentList		
AlertFlag	0	
MessageId		

State	10	
TransmitTime	2011/05/10 15:07:00	
StatusTime	2011/05/10 15:07:00	
StatusLog	回收訊息	

以上的訊息格式為 BPM 整合的格式參考，實際整合時，可依據需求進行調整。

3 附錄

3.1 Table Schema

Table Name: <u>Outbox</u>					
英文識別字	中文欄位	資料型態	資料長度	允許空值	說明
TxMessageId	訊息傳送序號	nvarchar	50	PK	用戶端自行定義，建議 GUID
ChannelCode	頻道代碼	nvarchar	20		用戶端自行定義，英文為主
SourceDomain	來源領域名稱	nvarchar	50		必填，用來判斷訊息來自何處，藉以處理未加上 domainName 的帳號資訊。
Sender	寄件者	nvarchar	50		請參照 UserAccount.UA002 用戶帳號欄位
ReceiverType	收件者類型	char	1		指定收件者的類型。 若是 U=User，則 ReceiverList 表示內容為使用者列表。 若是 D=Device，則表示 ReceiverList 的內容為裝置序號列表
ReceiverList	收件者列表	nvarchar	max		收件者列表，多筆使用；分隔，請參照 UserAccount.UA002 用戶帳號欄位
BccList	密件收件者列表	nvarchar	max	V	密件收件者列表，處理方式同上
Subject	主旨	nvarchar	200		訊息主旨
Content	內文	nvarchar	max		訊息內文
ContentType	內文類型	nvarchar	20		用來說明訊息內文的資料類型為何 CMD=命令訊息 MSG=資訊訊息 REPLY=回覆訊息 UPDATE=更新訊息 DELETE=刪除訊息 STATUS=狀態訊息 RECALL=回收訊息
AttachmentList	附件列表	nvarchar	max	V	欲傳送的附件列表，每個附件檔必須給完整的路徑及名稱，用分號(;)隔開。
AlertFlag	通知旗標	bit	1		是否提示使用者有新訊息通知(Y:顯示通知 N:不顯示)
MessageId	訊息識別碼	nvarchar	50	V	伺服器回傳的訊息序號
State	訊息狀態	int			0=初始化， 10=尚未上傳,11=附件上傳中,12=附件上傳完成, 19=上傳完成。 20 = 等待伺服器發送, 29=發送伺服器完成。 97=逾時, 98 = 作廢, 99 = 處理失敗
TransmitTime	傳送時間	nchar	20	V	訊息的傳送時間，用字串記錄，yyyy-MM-dd HH:mm:ss
StatusTime	狀態時間	nchar	20	V	訊息狀態的更新時間，用字串記錄，yyyy-MM-dd HH:mm:ss
StatusLog	狀態記錄	nvarchar	200	V	訊息處理過程中的狀態說明記錄
Priority	優先權	nvarchar	1		E:急件 · H:高優先權 · N:一般(default) · L:低優先權。
Keywords	關鍵字	nvarchar	200	V	訊息傳送時，可自行定義關鍵字在此欄位，並用分隔符號(;)分開，系統在

顯示或是搜尋時可以依據此欄位進行檢索。作為訊息的篩選條件。

Table Name: Inbox					
英文識別字	中文欄位	資料型態	資料長度	允許空值	說明
RxMessageId	訊息接收序號	nvarchar	50	PK	伺服器端回傳的序號，通常是 GUID
ChannelCode	頻道代碼	nvarchar	20		用戶端自行定義，英文為主
SourceDomain	來源領域名稱	nvarchar	50		必填，用來判斷訊息來自何處，藉以處理未加上 domainName 的帳號資訊。
Sender	寄件者	nvarchar	50		請參照 UserAccount.UA002 用戶帳號欄位
Owner	訊息擁有者	nvarchar	max		訊息擁有者，可能是一般使用者，也可能是 ChannelAgent 的帳號
ReceiverType	收件者類型	char	1		指定收件者的類型。 若是 U=User，則 ReceiverList 表示內容為使用者列表。 若是 D=Device，則表示 ReceiverList 的內容為裝置序號列表
ReceiverList	收件者列表	nvarchar	max		原始訊息的收件者列表，多筆使用；分隔 請參照 UserAccount.UA002 用戶帳號欄位
BccList	密件收件者列表	nvarchar	max	V	密件收件者列表，處理方式同上
Subject	主旨	nvarchar	200		訊息主旨
Content	內文	nvarchar	max		訊息內文
ContentType	內文類型	nvarchar	20		用來說明訊息內文的資料類型為何 CMD=命令訊息 MSG=資訊訊息 REPLY=回覆訊息 UPDATE=更新訊息 DELETE=刪除訊息 STATUS=狀態訊息 RECALL=回收訊息
AttachmentFlag	附件旗標	bit	1		此旗標表示訊息含有附件資料
AlertFlag	通知旗標	bit	1		是否提示使用者有新訊息通知(Y:顯示通知 N:不顯示)
MessageId	訊息識別碼	nvarchar	50		伺服器回傳的訊息識別碼，通常是 GUID
State	訊息狀態	int			0=初始化, 10=準備下載,11=附件下載中,12=附件下載完成, 19=下載完成。 20 = 等待用戶處理, 21=用戶讀取中，29=用戶處理完成。 97=逾時, 98 = 作廢, 99 = 處理失敗
TransmitTime	傳送時間	nchar	20	V	訊息的傳送時間，用字串記錄，yyyy-MM-dd HH:mm:ss
ReceiveTime	接收時間	nchar	20	V	訊息的接收時間，用字串記錄，yyyy-MM-dd HH:mm:ss

StatusTime	狀態時間	nchar	20	V	訊息狀態的更新時間，用字串記錄，yyyy-MM-dd HH:mm:ss
StatusLog	狀態記錄	nvarchar	200	V	訊息處理過程中的狀態說明記錄
Priority	優先權	nvarchar	1		E:急件，H:高優先權，N:一般(default)，L:低優先權。
Keywords	關鍵字	nvarchar	200	V	訊息傳送時，可自行定義關鍵字在此欄位，並用分隔符號(;)分開，系統在顯示或是搜尋時可以依據此欄位進行檢索。作為訊息的篩選條件。

Table Name: <u>InboxAttachment</u> 、 <u>OutboxAttachment</u>					
英文識別字	中文欄位	資料型態	資料長度	允許空值	說明
AttachmentId	識別碼	uniqueidentify		PK	自動編號
MessageId	訊息識別碼	nvarchar	50		參考 Outbox/Inbox 的.PK
SeqNo	序號	int			用來排序檔案順序
FileName	檔名	nvarchar	250		
FileType	檔案類型	nvarchar	10		檔案類型
FileSize	檔案大小	int			
FileData	檔案內容	varbinary	max	V	將檔案儲存於資料庫此欄位內。
SourceUri	檔案類型	nvarchar	10	V	表示檔案所在位置(Inbox=>遠端，Outbou=>本機,)

3.2 名詞解釋

略