Contents

[Background 1](#_Toc43193943)

[Device and customer repository 1](#_Toc43193944)

[Device ID and customer ID Synchronization 2](#_Toc43193945)

[Types of PN Requests 3](#_Toc43193946)

[Processing flow during mobile apps first launch: 3](#_Toc43193947)

[System flow for PN 4](#_Toc43193948)

[Scope of enhancement: 4](#_Toc43193949)

[Future enhancement 5](#_Toc43193950)

[Web service for integrated e-alert 5](#_Toc43193951)

### Background

1. The current DSB mobile apps are Mobile banking, 328, DS-Direct & Securities Trading. The 4 apps are installed and work independently.
2. A customer needs to bind his own document id with the 4 mobile apps separately.
3. Each mobile app require to create one corresponding moibileAppID in Catapush (total 4 mobileAppID in 1 catapush account).
4. For each mobile apps, a customer can only bind his document id with one device.
5. System cannot reliably detect uninstallation of mobile apps.
6. To request Catapush / FCM to send PN, we need to specify the target mobile apps.
7. After apps uninstallation, Catapush / FCM cannot send PN to the device.
8. [User id] has 1-to-1 mapping with [Device id]+App ID. EMS is using [User id] to communicate with Catapush only.

### Device and customer repository

1. After first launch of mobile apps, below is stored in Catapush

|  |  |  |
| --- | --- | --- |
| Use id | Device ID/Token | App ID |
| xxx | xxx | xxx |

Below should be stored in BOSS database.

|  |  |  |  |
| --- | --- | --- | --- |
| Use id | Device ID/Token | App ID | Customer document id |
| xxx | Xxx | xxx | NULL |

* If the apps is installed on one more device, one more set of record (user id+device id+app id) will be added.

That is, below should be stored in BOSS database.

|  |  |  |  |
| --- | --- | --- | --- |
| Use id | Device ID/Token | App ID | Customer document id |
| Xxx | Xxx | xxx | NULL |
| Xxx | Xxx | xxx | NULL |

* If the apps is re-installed on the same device, same use id is adopted. Hence, there is no change in Catapush. Yet there is update in BOSS, and will be discussed in next section.

1. After a customer binds his document id with a mobile apps, below should be updated in BOSS. Hence, a customer can have at most 4 records in the below table, one for each mobile apps.

|  |  |  |  |
| --- | --- | --- | --- |
| Use id | Device ID/Token | App ID | Customer document id |
| xxx | Xxx | xxx | xxx |

### Device ID and customer ID Synchronization

All records are synchronized from mobile apps to BOSS online.

1. After Mobile apps installation:
   1. insert record (app id + device id + user id)
   2. if the record already exist, clear customer id
2. After biometric binding,
   1. clear customer id of the existing record with the same app id + customer id
   2. assign customer id with matching app id + device id + user id
3. Examples of record maintenance

Example 1

Time1: customer install mobile apps A. Below record is added.

|  |  |  |  |
| --- | --- | --- | --- |
| Use id | Device ID/Token | App ID | Customer document id |
| Xxx | Xxx | xxx | NULL |

Time 2: customer bind his HKID with device. Customer HKID is then updated

|  |  |  |  |
| --- | --- | --- | --- |
| Use id | Device ID/Token | App ID | Customer document id |
| xxx | Xxx | xxx | NULL 🡪 xxx |

Time 3: customer re-install mobile apps on the same device. The customer ID should be removed.

|  |  |  |  |
| --- | --- | --- | --- |
| Use id | Device ID/Token | App ID | Customer document id |
| xxx | Xxx | xxx | Xxx 🡪 NULL |

Example 2

Time1: customer install mobile apps A on 2 devices. Below record is added.

|  |  |  |  |
| --- | --- | --- | --- |
| Use id | Device ID/Token | App ID | Customer document id |
| Xxx | AAA | xxx | NULL |
| Xxx | BBB | xxx | NULL |

Time 2: customer bind his HKID with device A. Customer HKID is then updated

|  |  |  |  |
| --- | --- | --- | --- |
| Use id | Device ID/Token | App ID | Customer document id |
| xxx | AAA | xxx | NULL 🡪 xxx |
| xxx | BBB | xxx | NULL |

Time 3: customer bind his HKID with device B. Customer HKID is then updated

|  |  |  |  |
| --- | --- | --- | --- |
| Use id | Device ID/Token | App ID | Customer document id |
| xxx | AAA | xxx | Xxx 🡪 NULL |
| xxx | BBB | xxx | NULL 🡪 xxx |

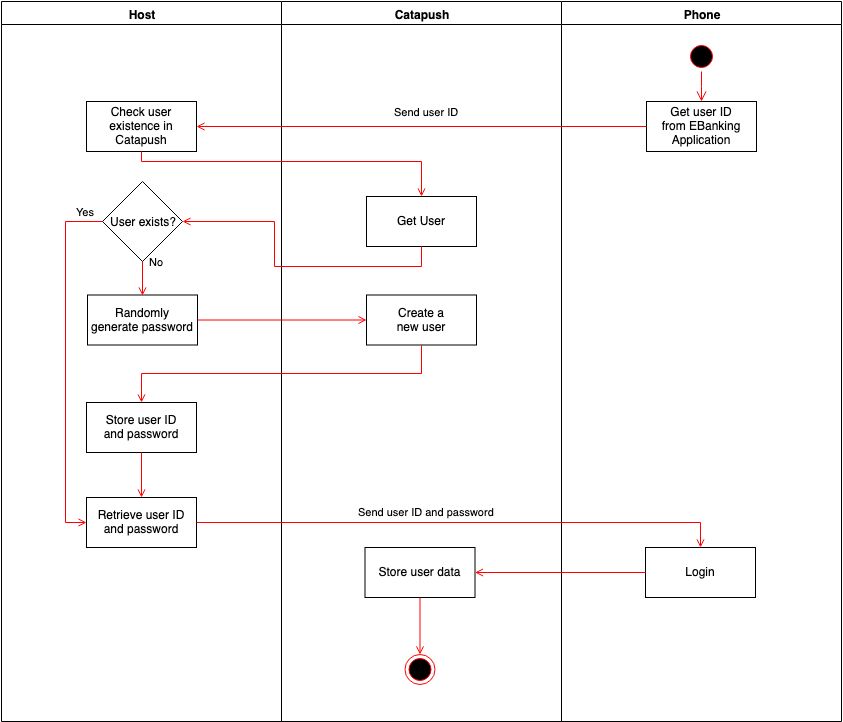
### Types of PN Requests

1. Below are different types of PN requests. All are targeted at a specific mobile apps.

|  |  |  |
| --- | --- | --- |
| **Type of PN** | **A) via Catapush** | **B) via FCM** |
| **Default Plan 1 – Guaranteed**  **(priority) PN>SMS** | Support | Not Support |
| **Default Plan 2 – Non-Guaranteed**  **(priority) PN>email>SMS (attempt only 1 channel)** | Not Support | Support |
| **1) specific customer id with fallback SMS / email** | Support | Not support |
| **2) specific customer id without fallback SMS / email** | Support | Support |
| **3) specific device id** | Not needed | Support |

1. Broadcasting to all customers is the batch mode of (2B).
2. Broadcasting to all devices is the batch mode of (3B).
3. BOSS enquiry is supported for (1A), (2A) & (2B) only.

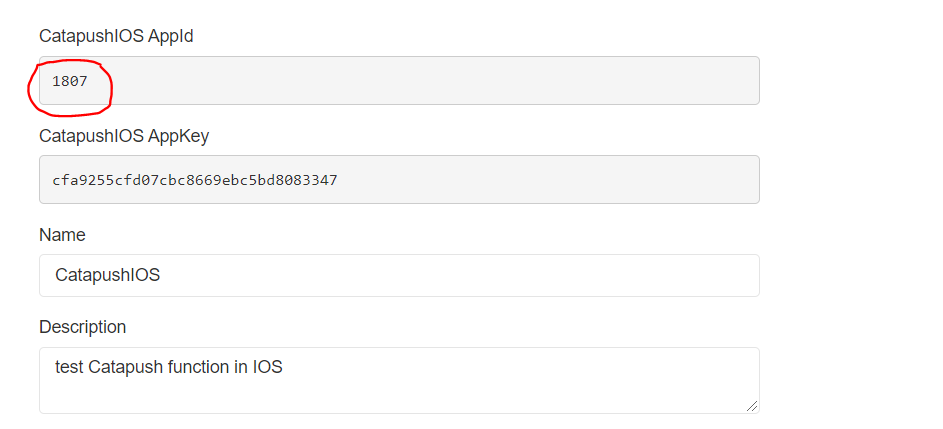
### Processing flow during mobile apps first launch:



Objective:

* + - Generate an unique userID with password
    - Use the generated userID and password to logon catapush, catapush will generate a corresponding deviceID (Token)
    - Host/ boss store the userID and appID for PN.

1. Customer install the app, which contain the appID (provided by catapush, total 4 appID for 4 DahSing App)

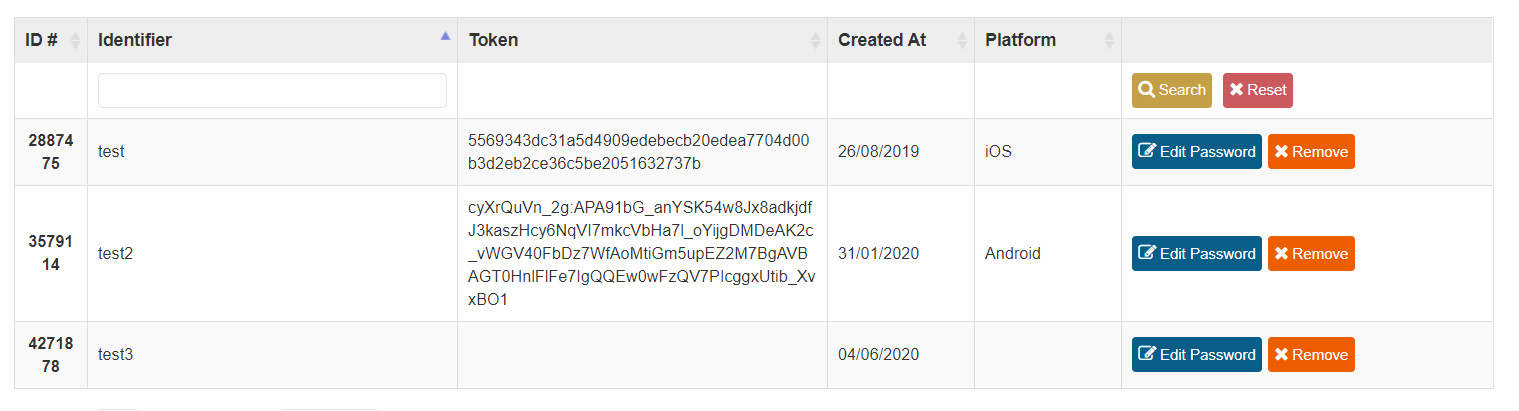


1. Get customer consent to receive push.
2. Phone generate user id and password (why not host to generate?)

Generate user id by phone: Apps could logon without receiving the userID and password from host.

Generate user id by Host: Host require to pass the user ID, password to app for login

1. Mobile 🡪 Host: send appID + userID
2. Host 🡪 Catapush: check if (user id) already exist
3. Catapush 🡪 Host: already exist / not exist
4. Host (Loop 3,4 until user id does not exist)🡪 Catapush: create user id ( appID + userID)
5. Host: keep appID + userID
6. Host 🡪 Phone: send appID + userID
7. Phone 🡪 Catapush: logon appID + userID
8. Catapush will create a corresponding token for this appID+userID



There are 2 ways to process the user initiation process in Catapush

1. APP: Customer consent 🡪 HOST: query and create an unique userID 🡪 APP: app receive the userID + pw and logon catapush 🡪 catapush register the logon device 🡪 Done.

(host has to query the catapush again to ensure the logon process success)

1. APP: Customer consent 🡪 APP: query and create an unqiue userID 🡪 APP: logon Catapush using the userID +pw 🡪 catapush register the logon device 🡪 APP: sent the registered userID and password to host 🡪 host could query catapush to confirm the logon success. 🡪 Done.

### System flow for PN

System flow – 1A

1. Source system call EMS asynchronous web service (info pass to EMS: customer id, target mobile app. Template ID, parameters)
2. EMS acknowledge requesting system
3. EMS generate the push message content
4. EMS look up catapush user id from BOSS database, & add push message log in BOSS
5. If catapush user id NOT found, fallback to SMS / email
6. EMS send push message to Catapush
7. EMS receive delivery status from Catapush
8. EMS update delivery status in BOSS
9. If PN fails, fallback to SMS / email

System flow – 2A

1. Source system call EMS asynchronous web service (info pass to EMS: customer id, target mobile app. Template ID, parameters)
2. EMS acknowledge requesting system
3. EMS generate the push message content
4. EMS look up catapush user id from BOSS database, & add push message log in BOSS
5. If catapush user id NOT found, terminate
6. EMS send push message to Catapush
7. EMS receive delivery status from Catapush
8. EMS update delivery status in BOSS
9. If PN fails, fallback to SMS / email

System flow – 2B & 3B

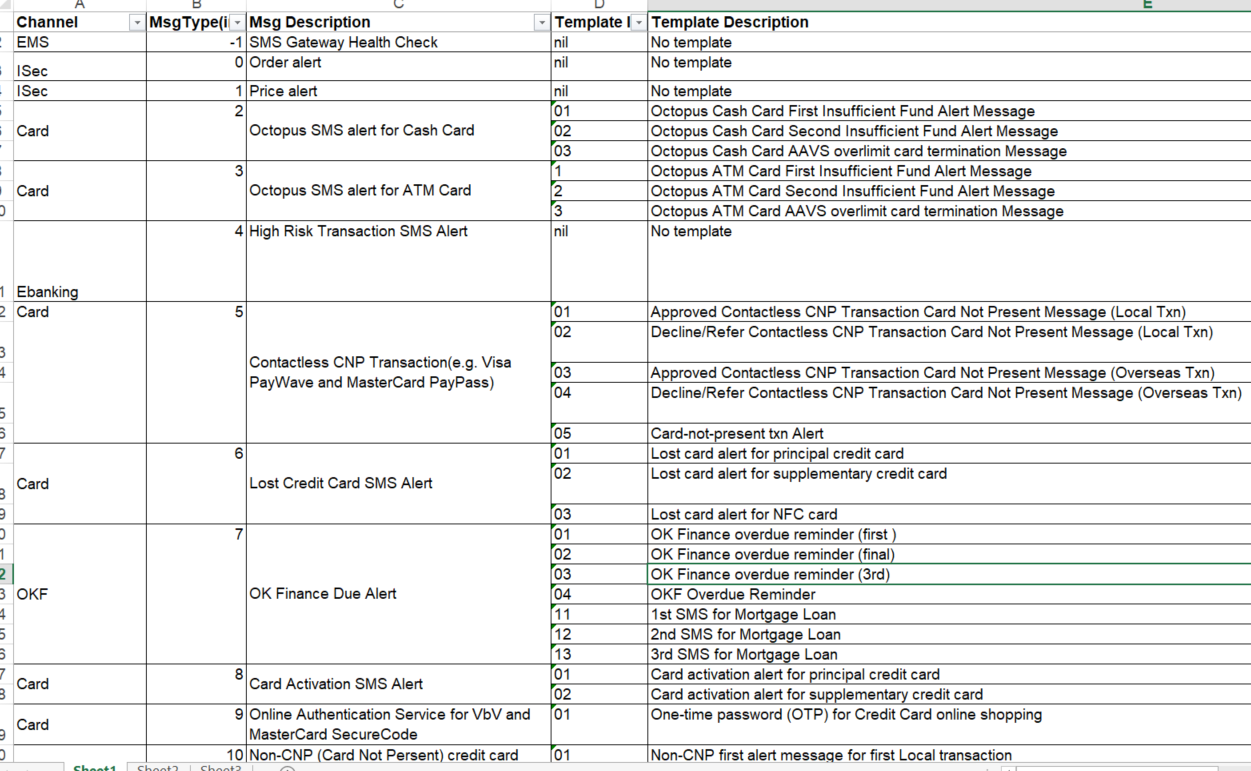
1. Source system call EMS asynchronous web service (info pass to EMS: customer id, target mobile app. Template ID, parameters)
2. EMS acknowledge requesting system
3. EMS generate the push message content
4. EMS look up device id from BOSS database, & add push message log in BOSS
5. If device id NOT found, terminate
6. EMS send push message to FCM

### Scope of enhancement:

1. EMS new web service for integrated e-alert
2. Separate thread pool for the e-alert service coz each call may take several seconds to complete
3. Mobile apps: write/update catapush user id upon app installation
4. Mobile apps: one-off user id creation for existing customers
5. Online synchronize device & customer id from mobile apps to BOSS
6. New EMS web server to receive push message delivery result from Catapush
7. EMS to BOSS log PN message
8. BOSS enquiry & database change

### Future enhancement

1. Users enquiry and maintenance of message templates
   1. ITD will set up the initial version of all message templates
   2. The templates will be listed in an enquiry function (see below sample screen).
      1. Users can search & download the templates. Further details (like UR) can also be downloaded.
      2. In UAT, users can modify and upload templates. Suppose only static text modification only. The templates will reloaded at scheduled time (say, during lunch or night).
      3. Production migration by ITD only



1. Enhancement of Push message
   1. Image attachment
   2. Customer reply
2. Addition of new type of e-alert
   1. WeChat

### Web service for integrated e-alert

Service Interface

1. Below is an example only. Action 2 is either EMAIL or SMS, but not both

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service :**  IntegratedEalert | | | | |
| **Input :** | | | | |
| **Tag** | | **Type** | | **Remark** |
| **EmsHeader** | | **EmsHeader** | |  |
| **AcctNum** | | **String** | |  |
| Msg Content | | String | | For EMS internal use |
| **RefNum** | | **String** | | **Unique reference number from client to check duplication.** |
| ActionFlag | | String | | Mandatory: fire all ealert  Optional: fire the first ealert; and fire the next ealert only if the present one fails |
| eAlertAction | | String[][] | | (Mix occurrence: unbounded)  Dynamic fields for mail merge |
| **Action 1** | | Action[0][0] | | PUSH |
|  | | Action[0][1] | | Msg Type |
|  | | Action[0][2] | | Msg String |
|  | | Action[0][3] | | Target mobile app: MBanking / MTrading / … |
| **Action 2** | | Action[1][0] | | EMAIL |
|  | | Action[1][1] | | Msg Type |
|  | | Action[1][2] | | Msg String |
|  | | Action[1][3] | | FromAddr |
|  | | Action[1][4] | | ToAddr |
|  | | Action[1][5] | | Cc |
|  | | Action[1][6] | | Bcc |
|  | | Action[1][7] | | Subject  If missing, use EMS defined value |
| **Action 2** | | Action[2][0] | | SMS |
|  | | Action[2][1] | | Msg Type |
|  | | Action[2][2] | | Msg String |
|  | | Action[2][3] | | Recipient (Customer mobile number) |
|  | | Action[2][4] | | Language: T/E |
| TemplateContent | | String[] | | (Mix occurrence: unbounded)  Dynamic fields for mail merge  Common parameters for PUSH/SMS/Email |
|  | | TemplateContent[0] | | Customer Name |
|  | | TemplateContent[1] | | Reference No |
|  | | TemplateContent[2] | | Login ID |
|  | | TemplateContent[3] | | Activate Small-Value Transfer in Chinese |
|  | | TemplateContent[4] | | Activate Small-Value Transfer in English |
|  | | TemplateContent[5] | | Daily Limit for Small-Value Transfer in Chinese |
|  | | TemplateContent[6] | | Daily Limit for Small-Value Transfer in English |
|  | | TemplateContent[7] | | Subscribe e-Statement, e-Advice & e-Alert in Chinese |
|  | | TemplateContent[8] | | Subscribe e-Statement, e-Advice & e-Alert in English |
|  | | TemplateContent[9] | | Language Setting for Notification in Chinese |
|  | | TemplateContent[10] | | Language Setting for Notification in English |
| **Output :** | | | | |
| **Tag** | **Type** | | **Remark** | |
| EmsHeader | EmsHeader | |  | |

Please also list out all other new web services to be built: e.g. mobile device registration for PN before binding; mobile device registration for PN after binding.

Questions: can the web services built in REST instead of SOAP?