Assumptions

Except if stated otherwise in a specific use case, the following preconditions are taken for granted:

* The end-user is a registered customer on issuer side
* The end-user has at least one plastic card from this issuer
* The end-user has a Samsung Pay compatible device
* The end-user has an Samsung account
* The plastic card is eligible

Card Enrolment – Samsung Pay initiated

User story yellow Flows (ID&V required)

**Main story (new user, card, device)**

**pre-conditions**

* The end-user is not a registered Samsung pay customer

**scenario**

1. The end-user launches her Samsung Pay application and hits the “+” button in order to enroll a new card in Samsung Pay
2. The end-user enters her card credentials either by selecting a card already registered with Samsung Account (card “on-file”, or by OCR (taking a picture of the card) or manually
3. The user hits the save button and waits for eligibility check.
4. According to the value card.issuer.tnc field present in the POST / Enrollments response the user has to accepts the terms & conditions

**Already existing user and device, new card**

1. The end-user launches her Samsung Pay application, already enrolled cards are displayed, she hits the “+” button in order to enroll another card in Samsung Pay wallet.

The rest of the use case is equivalent.

**Already existing user and card, new device**

The flow is the same as [Main story (new user, card, device)](https://confluence.oberthur.com/display/SPAYT/Card+enrolment+-+User+centric+flows#Cardenrolment-Usercentricflows-Mainstory(newuser,card,device)).

On step 2. The user will be able to select the card already on-file.

**Already existing user, new card and device**

The flow is the same as [Main story (new user, card, device)](https://confluence.oberthur.com/display/SPAYT/Card+enrolment+-+User+centric+flows#Cardenrolment-Usercentricflows-Mainstory(newuser,card,device)).

* [Card enrollment Samsung Pay Initiated](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-CardenrollmentSamsungPayInitiated)
  + [Main flow](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Mainflow)
  + [Existing user and device, new card](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Existinguseranddevice,newcard)
  + [Existing card and new device](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Existingcardandnewdevice)
  + [Error flows](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Errorflows)
    - [Enrollment already ongoing](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Enrollmentalreadyongoing)
    - [Card already tokenized](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Cardalreadytokenized)
    - [Card expired](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Cardexpired)
    - [No card product match (no issuer found)](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Nocardproductmatch(noissuerfound))
    - [Not eligible (PAN)](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Noteligible(PAN))
    - [Not eligible (cardholder)](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Noteligible(cardholder))
    - [Failed to parse request](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Failedtoparserequest)
    - [Decryption error](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Decryptionerror)
    - [Internal system down](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Internalsystemdown)
* [Device Token Enrollment](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-DeviceTokenEnrollment)
  + [Error flows](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Errorflows.1)
    - [Failed to parse request](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Failedtoparserequest.1)
    - [Decryption error](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Decryptionerror.1)
    - [Internal system down](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-Internalsystemdown.1)
* [ID&V](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+enrollment#Usecases-enrollment-ID&V)

Card enrollment Samsung Pay Initiated

Main flow

| **Use Case ID & Name** | **UC #1: Card Enrollment, new user, new card, new device** |
| --- | --- |
| **Description** | Samsung Pay server checks eligibility of a card being enrolled from the Samsung Pay wallet. |
| **Priority** | HIGH |
| **Pre-conditions** | * The end-user is a registered customer on issuer side * The end-user has at least one plastic card from this issuer * The end-user has an Samsung Pay compatible device * The end-user has an Samsung Pay compatible Samsung account * The plastic card is eligible |
| **Triggers** | User initiates card enrollment from the Samsung Pay Wallet. |
| **Main Scenario** | 1. The Samsung Pay server initiates a *POST / Enrollments*:    1. The device and card is unknown to the OT DPS 2. The OT DPS sends an **CheckCardsRequest** and **EnrollCardRequest** to the issuer. 3. The OT DPS returns a successful *POST / Enrollments Response.* |
| **Alternate Scenario** | Always the main scenario is done |
| **Post-conditions** | * A temporary enrollment object aggregating a new card object and a new device object is provisioned in the OT DPS * A tokenRefId that uniquely represent the enrollment process of this card on this walletId has been provided to Samsung Pay servers. * Localized contents provided by the OT DPS match the *"locale"* argument based on the strategy defined in [Internationalization](https://confluence.oberthur.com/display/SPAYT/OnBoarding). |

Existing user and device, new card

| **Use Case ID & Name** | **UC #2: Card Enrollment, existing user, new card, existing device** |
| --- | --- |
| **Description** | Samsung Pay server checks eligibility of a card being enrolled from the Samsung Pay wallet. |
| **Priority** | HIGH |
| **Pre-conditions** | * The end-user is a registered customer on issuer side * The end-user has at least one plastic card from this issuer * The end-user has an Samsung Pay compatible device * The end-user has an Samsung Pay compatible Samsung account * The plastic card is eligible * The device is already registered in the OT DPS |
| **Triggers** | User initiates card enrollment from the Samsung Pay Wallet. |
| **Main Scenario** | 1. The Samsung Pay server initiates a *POST / Enrollments*:    1. The device already registered and card is unknown to the OT DPS 2. The OT DPS sends an**CheckCardsRequest** and **EnrollCardRequest** to the issuer. 3. The OT DPS returns a successful *POST / Enrollments Response.* |
| **Alternate Scenario** | As defined in Main flow. |
| **Post-conditions** | * A temporary enrollment object aggregating a new card object and the already existing device is provisioned in the OT DPS * A tokenRefId that uniquely represent the enrollment process of this card on this walletId has been provided to Samsung Pay servers. * Localized contents provided by the OT DPS match the  *"locale"*  argument based on the strategy defined in [Internationalization](https://confluence.oberthur.com/display/SPAYT/OnBoarding). |

Existing card and new device

| **Use Case ID & Name** | **UC #3: Card Enrollment, existing user, existing card, new device** |
| --- | --- |
| **Description** | Samsung Pay server checks eligibility of a card being enrolled from the Samsung Pay wallet. |
| **Priority** | HIGH |
| **Pre-conditions** | * The end-user is a registered customer on issuer side * The end-user has at least one plastic card from this issuer * The end-user has an Samsung Pay compatible device * The end-user has an Samsung Pay compatible Samsung account * The plastic card is eligible * The card is already registered in the OT DPS |
| **Triggers** | User initiates card enrollment from the Samsung Pay Wallet. |
| **Main Scenario** | 1. The Samsung Pay server initiates a *POST / Enrollments*:    1. The card already registered and device is unknown to the OT DPS 2. The OT DPS sends an **CheckCardsRequest** and **EnrollCardRequest** to the issuer. 3. The OT DPS returns a successful *POST / Enrollments Response.* |
| **Alternate Scenario** | As defined in Main flow. |
| **Post-conditions** | * A temporary enrollment object aggregating a card object and a new device is provisioned in the OT DPS * A tokenRefId that uniquely represent the enrollment process of this card on this walletId has been provided to Samsung Pay servers. * Localized contents provided by the OT DPS match the *"locale"* argument based on the strategy defined in [Internationalization](https://confluence.oberthur.com/display/SPAYT/OnBoarding). |

Error flows

**Enrollment already ongoing**

| **Use Case ID & Name** | **UC #4 : Error on enrollment  (Already on-going)** |
| --- | --- |
| **Description** | Samsung Pay servers send a *POST / Enrollments* request for an already ongoing card enrollment |
| **Scenario** | 1. Samsung Pay servers send a *POST / Enrollments* for a *card* and a *walletId* for which an enrollment already exists or for which a deviceToken without a device  2. The OT DPS returns the same response as previously returned to Samsung Pay servers for this enrollment. |
| **Post-condition** | Samsung Pay servers continue with tokenization of this card on this wallet. |

**Card already tokenized**

| **Use Case ID & Name** | **UC #5 : Error on enrollment  (Card already enrolled)** |
| --- | --- |
| **Description** | Samsung Pay servers send a *POST / Enrollments* Request for an already enrolled card |
| **Scenario** | 1. Samsung Pay servers send a *POST / Enrollments* request for a card and a walletId for which a token has already been issued (***acknowledgePaymentBundle***received) and is not in the deleted state.  2. The OT DPS returns the error as in [TR API]**,** alongside the associated tokenRefId in the *POST / Enrollments* Response. |
| **Post-condition** | Samsung Pay servers recovers by getting a new bundle for this tokenRefId.  No new enrollment object is persisted. |

**Card expired**

| **Use Case ID & Name** | **UC #6 : Error on enrollment  (Card expired)** |
| --- | --- |
| **Description** | Samsung Pay server sends a *POST / Enrollments* request for an expired card. |
| **Scenario** | 1. Samsung Pay servers send a *POST / Enrollments* request for a card that has expired (the ***encryptedPaymentCard.expiryMonth/expiryYear***fields are in the past).  2. The OT DPS returns the error as in [TR API] |
| **Post-condition** | Samsung Pay servers stops enrollment of this card.  No new enrollment object is persisted. |

**No card product match (no issuer found)**

| **Use Case ID & Name** | **UC #7 : Error on enrollment  (Issuer not found)** |
| --- | --- |
| **Description** | Samsung Pay server sends a *POST / Enrollments* for a card that matches none of the card products declared on the OT DPS. |
| **Scenario** | 1. Samsung Pay servers send a *POST / Enrollments* request for a card with a PAN that matches none of the card products declared on the OT DPS. As a result the OT DPS doesn’t know to which issuer this card belongs and has to stop.  2. The OT DPS returns the error as in [TR API] |
| **Post-condition** | Samsung Pay servers stops enrollment of this card.  No new enrollment object is persisted. |

**Not eligible (PAN)**

| **Use Case ID & Name** | **UC #8 : Error on enrollment  (PAN not eligible)** |
| --- | --- |
| **Description** | Samsung Pay server sends a *POST / Enrollments* request, the OT DPS contact the issuer, but the PAN isn’t eligible. |
| **Scenario** | 1. Samsung Pay servers send a *POST / Enrollments* request and the OT DPS contact the issuer.  2. In the CheckCards response the issuer returns an error indicating the PAN isn’t eligible.  3. The OT DPS returns the error as in [TR API] |
| **Post-condition** | Samsung Pay servers stop enrollment for this card.  No new enrollment object is persisted. |

**Not eligible (cardholder)**

| **Use Case ID & Name** | **UC #9 : Error on enrollment  (Cardholder not eligible)** |
| --- | --- |
| **Description** | Samsung Pay server sends a *POST / Enrollments* request, the OT DPS contact the issuer, but the cardholder isn’t eligible. |
| **Scenario** | 1. Samsung Pay servers send a *POST / Enrollments* request and the OT DPS contact the issuer.  2. In the EnrollCard response the issuer returns an error indicating the card holder didn’t match the provided PAN number.  3. The OT DPS returns the error as in [TR API] . |
| **Post-condition** | Samsung Pay servers stop enrollment for this card.  No new enrollment object is persisted. |

**Failed to parse request**

| **Use Case ID & Name** | **UC #10 : Error on enrollment  (Failed to parse request)** |
| --- | --- |
| **Description** | Samsung Pay server sends an improperly formatted *POST / Enrollments* request. |
| **Scenario** | 1. Samsung Pay servers send an improperly formatted *POST / Enrollments* request, the OT DPS fails to parse it.  2. The OT DPS returns the error as in [TR API] |
| **Post-condition** | This raises an alarm on both servers and the support teams investigate what’s going on. |

**Decryption error**

| **Use Case ID & Name** | **UC #11 : Error on enrollment  (Failed to decrypt card.data field)** |
| --- | --- |
| **Description** | Samsung Pay servers send a properly formatted *POST / Enrollments* request with improperly ciphered ***card.data***field. |
| **Scenario** | 1. Samsung Pay servers send a properly formatted *POST / Enrollments* request with improperly ciphered ***card.data***, the OT DPS fails to decipher it.  2. The OT DPS returns the error as in [TR API] . |
| **Post-condition** | This raises an alarm on both servers and the support teams investigate what’s going on. |

**Internal system down**

| **Use Case ID & Name** | **UC #12 : Error on enrollment  (Internal system down)** |
| --- | --- |
| **Description** | Samsung Pay servers send a valid *POST / Enrollments* request but connection to the core fails. |
| **Scenario** | 1. Samsung Pay servers send a properly formatted *POST / Enrollments* request but during processing an internal component is found to be down (ex: unable to reach the Cassandra cluster).  2. The OT DPS returns the error as in [TR API] |
| **Post-condition** | This raises an alarm on both servers and the support teams investigate what’s going on. |

Device Token Enrollment

| **Use Case ID & Name** | **UC #13 : Token Request from Samsung Pay** |
| --- | --- |
| **Description** | Samsung Pay Server asks for a card to be tokenized on a given wallet |
| **Priority** | HIGH |
| **Pre-conditions** | A valid tokenRefId has been provided by the OT DPS to Samsung Pay servers |
| **Triggers** | CheckEligibility passed and Samsung Pay servers tries to tokenize a card. |
| **Main Scenario** | 1. The Samsung Pay server initiates a***POST / Tokens***request as defined in [TR API] for an eligible card using a valid ***cardReferenceId***. 2. The OT DPS finds the enrollment and other entities linked to this***cardReferenceId***and forwards this request to the Issuer:    1. By sending an ***EnrollDeviceToken***to register the token with the issuer and for the issuer to perform risk checking and provide the relevant ID&V options. 3. The issuer generates the token associated with this tokenRefId. Token credentials are generated in the background. 4. The OT DPS returns a successful *POST / TokensResponse* as defined in [TR API] and sets the activation object value using the following mapping:      |  |  | | --- | --- | | **EnrollDeviceTokenResponse.idvMethods** | **POST / TokensResponse.activation** | | * CallCenter * App (N/A for Samsung Pay) * OTPEmail * OTPARS (N/A for Samsung Pay) * OTPSMS | Object present filled with values listed in [TR API] | | * Skipped | No object |     Token metadata are set according to the enrollment data for this card. |
| **Alternate Scenario**  **External TSP** | On step 3 additional TokenRequest is send to external TSP.  Flow continues at step **4.** |
| **Post-conditions** | * The token is generated * The associated initial bundle is ready for download * The device token status is:   + ACTIVE if IDV is not required   + PENDING if activation object was returned |

Error flows

**Failed to parse request**

| **Use Case ID & Name** | **UC #14 : Error on Token Request  (Failed to parse request)** |
| --- | --- |
| **Description** | Samsung Pay server sends an improperly formatted *POST / Tokens*request. |
| **Scenario** | 1. Samsung Pay servers send an improperly formatted *POST / Tokens*request, the OT DPS fails to parse it.  2. The OT DPS returns the error as in [TR API] |
| **Post-condition** | This raises an alarm on both servers and the support teams investigate what’s going on. |

**Decryption error**

| **Use Case ID & Name** | **UC #15 : Error on Token Request  (Failed to decrypt card.data field)** |
| --- | --- |
| **Description** | Samsung Pay servers send a properly formatted *POST / Tokens*request with improperly ciphered ***card.data***field. |
| **Scenario** | 1. Samsung Pay servers send a properly formatted *POST / Tokens*request with improperly ciphered ***card.data***, the OT DPS fails to decipher it.  2. The OT DPS returns the error as in [TR API] . |
| **Post-condition** | This raises an alarm on both servers and the support teams investigate what’s going on. |

**Internal system down**

| **Use Case ID & Name** | **UC #16 : Error on Token Request (Internal system down)** |
| --- | --- |
| **Description** | Samsung Pay servers send a valid *POST / Enrollments* request but connection to the core fails. |
| **Scenario** | 1. Samsung Pay servers send a properly formatted *POST / Tokens*request but during processing an internal component is found to be down (ex: unable to reach the Cassandra cluster).  2. The OT DPS returns the error as in [TR API] |
| **Post-condition** | This raises an alarm on both servers and the support teams investigate what’s going on. |

ID&V

| **Use Case ID & Name** | **UC #17 : OTP ID&V** |
| --- | --- |
| **Description** | ID&V through OTP |
| **Priority** | MEDIUM |
| **Pre-conditions** | * The user has a valid account registered with the issuer backend * The issuer returns at least one of SMS, EMAIL or RECEIVE\_CALL as an available activation method |
| **Triggers** | POST / Tokens response sent with the supported activation.method |
| **Main Scenario**  **OTP ID&V** | 1. The Samsung Pay server requests a list of available activation methods. 2. The OT DPS finds the enrolment object associated to the provided tokenRefId and returns the previously received activation methods attached to the enrolment. 3. The user is provided with a list of available ID&V options among which SMS EMAIL or RECEIVE\_CALL, she selects one of those. 4. Samsung Pay Service Layer requests for an OTP to be sent to the chosen address. 5. The OT DPS receives a POST / Tokens / {reference} request that is forwarded to the issuer as a ***notifyChosenIDVMethod***request. 6. The issuer generates and sends an OTP using the selected method. 7. The user inputs the received OTP through the Samsung Pay mobile application. 8. The OT DPS receives a POST / Tokens / {reference}request that is forwarded to the issuer as a ***VerifyOTP***request. 9. The issuer verifies the OTP and returns a successful response forwarded back to the Samsung Pay server as a successful  answer to the *POST / Tokens / {reference}* |
| **Post-condition** | The token status is active. |

* [Description](https://confluence.oberthur.com/display/SPAYT/Green+path#Greenpath-Description)
* [Flow](https://confluence.oberthur.com/display/SPAYT/Green+path#Greenpath-Flow)
* [Methods](https://confluence.oberthur.com/display/SPAYT/Green+path#Greenpath-Methods)
  + [Example](https://confluence.oberthur.com/display/SPAYT/Green+path#Greenpath-Example)

Description

In the Samsung flows, TR means Samsung Pay Token Requester, TS means "Token Service" meaning IDEMIA TEE DEP.

User enters or scans a card to wallet app. Wallet app passes CHD (cardholder data) to PF (Payment Framework).

PF encrypts the CHD, posts it to Token Requestor. Token Requestor enrolls the card with token service and returns enrollment, terms and conditions, and other card metadata.

Wallet app presents user with terms & conditions. User accepts it. Token Requestor requests token service to provision a token.

Token service provisions an active token. Wallet app syncs the event with Token Requestor through Payment Framework. Token Requestor may send acknowledgement to token service.

Flow

**Card enrollment Green Flow**

Methods

| **Method name** | **API [TR API]**  **TR => TSM** | **API [TR API]**  **TR <= TSM** | **API [IS\_INTERFACE]**  **TSM => Issuer** | **API [IS\_INTERFACE]**  **TSM <= Issuer** |
| --- | --- | --- | --- | --- |
| POST /enrollments | [Check Eligibility](https://confluence.oberthur.com/display/SPAYT/Check+Eligibility) |  |  |  |
| POST /checkCards |  |  | [CheckCardsRequest](https://confluence.oberthur.com/display/PAYM/CheckCardsRequest) |  |
| POST /enrollCard |  |  | [EnrollCardRequest](https://confluence.oberthur.com/display/PAYM/EnrollCardRequest) |  |
| POST /tokens | [Request for Token](https://confluence.oberthur.com/display/SPAYT/Request+for+Token) |  |  |  |
| POST /enrollDeviceToken |  |  | [EnrollDeviceTokenRequest](https://confluence.oberthur.com/display/PAYM/EnrollDeviceTokenRequest) |  |
| POST /notifications |  | [Provision](https://confluence.oberthur.com/display/SPAYT/Provision) |  |  |
| GET /tokens/{id} | [Get token](https://confluence.oberthur.com/display/SPAYT/Get+token) |  |  |  |
| POST /deviceTokenStatusNotification |  |  | [DeviceTokenStatus](https://confluence.oberthur.com/display/PAYM/DeviceTokenStatus) |  |
| POST /report | [Provision event](https://confluence.oberthur.com/display/SPAYT/Provision+event) |  |  |  |

Input from Samsung:

**Example**

- [TR] POST /enrollments  
- [TR] POST /tokens : [Green Path] status.code = PENDING  
- [TS] POST /notifications : (Provision)  
- [TR] GET /tokens/{id} : **status.code = ACTIVE, data.profile exists**  
- [TR] POST /reports : provision.status = SUCCESS, token.status = ACTIVE  
- [TR] GET /tokens/{id} : **status.code = ACTIVE**, data.profile doesn’t exists

Depreciated - flow witout STET and TSP HUB

Notes

Types of Green Flow:

1. Green Flow (Enrolment from Samsung Pay wallet), T&C is requested
2. Green Flow with Enrolment from Issuing Bank Mobile Application (App2App), no T&C required

[Skip to end of metadata](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#page-metadata-end)

* Created by [WOJCICKI Dominik](https://confluence.oberthur.com/display/~DominWOJ), last modified on [Sep 04, 2018](https://confluence.oberthur.com/pages/diffpagesbyversion.action?pageId=105844629&selectedPageVersions=31&selectedPageVersions=32)

[Go to start of metadata](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#page-metadata-start)

* [Main Flow](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-MainFlow)
  + [Description](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-Description)
  + [Flow](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-Flow)
* [Import Card flow](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-ImportCardflow)
  + [Description](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-Description.1)
  + [Flow](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-Flow.1)
* [Flow diagram to share with external](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-Flowdiagramtosharewithexternal)
* [Methods](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-Methods)
* [Input from Samsung](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-InputfromSamsung)
  + [Main Flow](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-MainFlow.1)
  + [Import Card Flow](https://confluence.oberthur.com/pages/viewpage.action?pageId=105844629#Yellowpath-deployDeviceTokenbeforeID&V-ImportCardFlow)

Main Flow

Description

In the Samsung flows, TR means Samsung Pay Token Requester, TS means "Token Service" meaning IDEMIA TEE DEP.

User enters or scans a card to wallet app. Wallet app passes cardholder data (CHD) to PF.

PF encrypts the CHD, posts it to Token Requester. Token Requester enrolls the card with token service and returns enrollment, terms and conditions, and other card metadata.

Wallet app presents user with terms & conditions. User accepts it.

Token Requester requests token service to provision a token. Token service provisions a pending token with a list of secondary authentication methods.

User selects OTP method. Token Requester informs token service of the selection.

Token service generates OTP, returns the format of OTP, and sends the OTP value to user out of band channel such as SMS or email. User receives the OTP, and enters it into wallet app.

Wallet app sends the value to payment framework. Payment Framework sends the value to token service for validation.

Once the value is validated, token service activates the token and sends a notification to Token Requester on token status change.

Token Requester sends a push notification to device. Payment framework picks up the message and performs status sync on device.

Flow

This flow present flow with external TSP (STET).

**Main Yellow Flow - First card entry**

Import Card flow

Description

Wallet app retrieves a set of card-references for user. Wallet app passes a card-ref to PF. PF posts it to TR. TR enrolls the card with TS and returns enrollment, terms and conditions, and other card metadata. Wallet app may present user with terms & conditions. User accepts it. TR requests TS to provision a digital card. TS provision an active digital card. Wallet app syncs the event with TR through PF. TR may send acknowledgement to TS, if TS requires.

Flow

**Yellow Flow - n entry**

Flow diagram to share with external

Methods

| **Method name** | **API [TR API]**  **TR => TSM** | **API [TR API]**  **TR <= TSM** | **API [IS\_INTERFACE]**  **TSM => Issuer** | **API [IS\_INTERFACE]**  **TSM <= Issuer** |
| --- | --- | --- | --- | --- |
| POST /enrollments | [Enrollment](https://confluence.oberthur.com/display/SPAYT/Enrollment) |  |  |  |
| POST /checkCards |  |  | C[heckCardsRequest](https://confluence.oberthur.com/display/PAYM/CheckCardsRequest) |  |
| POST /tokens | [Request for Token](https://confluence.oberthur.com/display/SPAYT/Request+for+Token) |  |  |  |
| POST /enrollCard |  |  | [EnrollCardRequest](https://confluence.oberthur.com/display/PAYM/EnrollCardRequest) |  |
| POST /enrollDeviceToken |  |  | [EnrollDeviceTokenRequest](https://confluence.oberthur.com/display/PAYM/EnrollDeviceTokenRequest) |  |
| POST /notifications |  | [Provision](https://confluence.oberthur.com/display/SPAYT/Provision) |  |  |
| GET /tokens/{id} | [Get token](https://confluence.oberthur.com/display/SPAYT/Get+token) |  |  |  |
| POST /reports | [Provision event](https://confluence.oberthur.com/display/SPAYT/Provision+event) |  |  |  |
| POST /tokens/{id} | [Select of ID&V](https://confluence.oberthur.com/pages/viewpage.action?pageId=88345609) |  |  |  |
| POST /notifyChoosenIDVOption |  |  | [NotifyChosenIDVOption](https://confluence.oberthur.com/display/PAYM/NotifyChosenIDVOption) |  |
| POST /activateDeviceToken |  |  |  | [ActivateDeviceTokenRequest](https://confluence.oberthur.com/display/PAYM/ActivateDeviceTokenRequest) |
| POST /tokens/{id} | [Verify ID&V](https://confluence.oberthur.com/pages/viewpage.action?pageId=88345659) |  |  |  |
| POST /verifyOTP |  |  | [VerifyOTPRequest](https://confluence.oberthur.com/display/PAYM/VerifyOTPRequest) |  |
| POST /deviceTokenStatusNotification |  |  | [DeviceTokenStatus](https://confluence.oberthur.com/display/PAYM/DeviceTokenStatus) |  |

Input from Samsung

Main Flow

**1. If the client send reports(provision) before sending ID&V verification.**  
- [TR] POST /enrollments  
- [TR] POST /tokens : [Yellow Path] status.code = PENDING  
- [TS] POST /notifications (Provision: “Pending token” is ready to be deployed on device)  
- [TR] GET /tokens/{id} : **status.code = PENDING, data.profile exists**  
- [TR] POST /reports : **provision.status = SUCCESS**, token.status = PENDING  
- [TR] GET /tokens/{id} : **status.code = PENDING**, data.profile doesn’t exist  
- [TR] POST /tokens/{id} : [ACTIVATION]  
- [TR] POST /tokens/{id} : [VERIFICATION]  
- [TS] POST /notifications : token.status = RESUMED (Status Change)  
- [TR] GET /tokens/{id} : **status.code = ACTIVE,** data.profile doesn’t exist  
- [TR] POST /reports : **token.event = RESUMED**

**2. If the client send reports(provision) after sending ID&V verification.**  
- [TR] POST /enrollments  
- [TR] POST /tokens : [Yellow Path] status.code = PENDING  
- [TR] POST /tokens/{id} : [ACTIVATION]  
- [TR] POST /tokens/{id} : [VERIFICATION]  
- [TS] POST /notifications (Provision: “Pending token” is ready to be deployed on device)  
- [TR] GET /tokens/{id} : **status.code = PENDING, data.profile exists**  
- [TR] POST /reports : provision.status = SUCCESS, token.status = PENDING  
- [TR] GET /tokens/{id} : **status.code = PENDING**, data.profile doesn’t exist  
- [TS] POST /notifications : token.status = RESUMED (Status Change)  
- [TR] GET /tokens/{id} : **status.code = ACTIVE,**data.profile doesn’t exist  
- [TR] POST /reports : **token.event = RESUMED**

Import Card Flow

Diagram below is depreciated - use it only to compare what has changed in flow from may '17

[Skip to end of metadata](https://confluence.oberthur.com/display/SPAYT/Yellow+path#page-metadata-end)

* Created by [WOJCICKI Dominik](https://confluence.oberthur.com/display/~DominWOJ), last modified on [Sep 04, 2018](https://confluence.oberthur.com/pages/diffpagesbyversion.action?pageId=87689263&selectedPageVersions=24&selectedPageVersions=25)

[Go to start of metadata](https://confluence.oberthur.com/display/SPAYT/Yellow+path#page-metadata-start)

* [Description](https://confluence.oberthur.com/display/SPAYT/Yellow+path#Yellowpath-Description)
* [Flow](https://confluence.oberthur.com/display/SPAYT/Yellow+path#Yellowpath-Flow)
* [Methods](https://confluence.oberthur.com/display/SPAYT/Yellow+path#Yellowpath-Methods)
  + [Example](https://confluence.oberthur.com/display/SPAYT/Yellow+path#Yellowpath-Example)

Description

In the Samsung flows, TR means Samsung Pay Token Requester, TS means "Token Service" meaning IDEMIA TEE DEP.

User enters or scans a card to wallet app. Wallet app passes cardholder data (CHD) to PF.

PF encrypts the CHD, posts it to Token Requester. Token Requester enrolls the card with token service and returns enrollment, terms and conditions, and other card metadata.

Wallet app presents user with terms & conditions. User accepts it.

Token Requester requests token service to provision a token. Token service provisions a pending token with a list of secondary authentication methods.

User selects OTP method. Token Requester informs token service of the selection.

Token service generates OTP, returns the format of OTP, and sends the OTP value to user out of band channel such as SMS or email. User receives the OTP, and enters it into wallet app.

Wallet app sends the value to payment framework. Payment Framework sends the value to token service for validation.

Once the value is validated, token service activates the token and sends a notification to Token Requester on token status change.

Token Requester sends a push notification to device. Payment framework picks up the message and performs status sync on device.

Flow

**Card enrollment Yellow Flow - ID&V using SMS or email OTP**

Methods

| **Method name** | **API [TR API]**  **TR => TSM** | **API [TR API]**  **TR <= TSM** | **API [IS\_INTERFACE]**  **TSM => Issuer** | **API [IS\_INTERFACE]**  **TSM <= Issuer** |
| --- | --- | --- | --- | --- |
| POST /enrollments | [Enrollment](https://confluence.oberthur.com/display/SPAYT/Enrollment) |  |  |  |
| POST /checkCards |  |  | C[heckCardsRequest](https://confluence.oberthur.com/display/PAYM/CheckCardsRequest) |  |
| POST /tokens | [Request for Token](https://confluence.oberthur.com/display/SPAYT/Request+for+Token) |  |  |  |
| POST /enrollCard |  |  | [EnrollCardRequest](https://confluence.oberthur.com/display/PAYM/EnrollCardRequest) |  |
| POST /enrollDeviceToken |  |  | [EnrollDeviceTokenRequest](https://confluence.oberthur.com/display/PAYM/EnrollDeviceTokenRequest) |  |
| POST /tokens/{id} | [Select of ID&V](https://confluence.oberthur.com/pages/viewpage.action?pageId=88345609) |  |  |  |
| POST /notifyChoosenIDVOption |  |  | [NotifyChosenIDVOption](https://confluence.oberthur.com/display/PAYM/NotifyChosenIDVOption) |  |
| POST /tokens/{id} | [Verify ID&V](https://confluence.oberthur.com/pages/viewpage.action?pageId=88345659) |  |  |  |
| POST /verifyOTP |  |  | [VerifyOTPRequest](https://confluence.oberthur.com/display/PAYM/VerifyOTPRequest) |  |
| POST /DeployDeviceToken |  |  |  | [DeployDeviceTokenRequest](https://confluence.oberthur.com/display/PAYM/DeployDeviceTokenRequest) |
| POST /notifications |  | [Provision](https://confluence.oberthur.com/display/SPAYT/Provision) |  |  |
| GET /tokens/{id} | [Get token](https://confluence.oberthur.com/display/SPAYT/Get+token) |  |  |  |
| POST /report | [Provision event](https://confluence.oberthur.com/display/SPAYT/Provision+event) |  |  |  |
| POST /deviceTokenStatusNotification |  |  | [DeviceTokenStatus](https://confluence.oberthur.com/display/PAYM/DeviceTokenStatus) |  |

Input from Samsung

**Example**

- [TR] POST /enrollments  
- [TR] POST /tokens : [Yellow Path] status.code = PENDING  
- [TR] POST /tokens/{id} : [ACTIVATION]  
- [TR] POST /tokens/{id} : [VERIFICATION]  
- [TS] POST /notifications : (Provision)  
- [TR] GET /tokens/{id} : **status.code = ACTIVE, data.profile exists**  
- [TR] POST /reports : **provision.status = SUCCESS**, token.status = ACTIVE  
- [TR] GET /tokens/{id} : **status.code = ACTIVE**, data.profile doesn’t exists

[Skip to end of metadata](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#page-metadata-end)

* Created by [WOJCICKI Dominik](https://confluence.oberthur.com/display/~DominWOJ), last modified on [Jul 23, 2019](https://confluence.oberthur.com/pages/diffpagesbyversion.action?pageId=87689240&selectedPageVersions=29&selectedPageVersions=30)

[Go to start of metadata](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#page-metadata-start)

* [High level description](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-Highleveldescription)
  + [If action triggered from Samsung TR](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-IfactiontriggeredfromSamsungTR)
  + [If action triggered from  Issuer](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-IfactiontriggeredfromIssuer)
* [Suspension / Resumption note](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-Suspension/Resumptionnote)
* [Flow triggered by Samsung](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-FlowtriggeredbySamsung)
  + [Token Suspension](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-TokenSuspension)
  + [Token Resumption](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-TokenResumption)
  + [Token Termination](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-TokenTermination)
    - [Synchronous](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-Synchronous)
    - [Asynchronous](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-Asynchronous)
* [Flow triggered by Issuer](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-FlowtriggeredbyIssuer)
  + [Token Suspension / Resumption](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-TokenSuspension/Resumption)
  + [Token Termination](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-TokenTermination.1)
* [Device Token Metadata Change (Update Card)](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-DeviceTokenMetadataChange(UpdateCard))
* [Methods](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-Methods)
* [Samsung Token Status Management](https://confluence.oberthur.com/display/SPAYT/Card+lifecycle#Cardlifecycle-SamsungTokenStatusManagement)

## High level description

The lifecycle events could be triggered either from Samsung (TR) or from the Issuer (Using Issuer API)

The concept is always the same:

### If action triggered from Samsung TR

* Samsung TR sends to IDEMIA a**PATCH / Tokens{id}**
* IDEMIA solution must perform the requested action (Lock, Unlock, Delete, ...) with the entity managing the Token (TSP, Issuer, ...)
* IDEMIA solution notifies the Issuer about the completion of the action
* IDEMIA solution notifies Samsung TR about the completion of the requested actions **POST / notification**
* Based on this notification, Samsung TR sends**GET / Tokens{id}**for getting the token status

### If action triggered from  Issuer

* Issuer sends to IDEMIA a**request using Issuer API**
* IDEMIA solution must perform the requested action (Lock, Unlock, Delete, ...) with the entity managing the Token (TSP, Issuer, ...)
* IDEMIA solution notifies the Issuer about the completion of the action
* IDEMIA solution notifies Samsung TR about the completion of the requested actions **POST / notification**
* Based on this notification, Samsung TR sends**GET / Tokens{id}**for getting the token status

## Suspension / Resumption note

IDEMIA DEP manage two independent suspension flags. Request from Samsung TR raise the wallet suspension flag and Issuer raise issuer suspension flag. This operation not work crosswise (e.g. Issuer can not set wallet suspension flag to NOT\_SUSPENDED).

Card operate only when both flags are set as NOT\_SUSPENDED.

## Flow triggered by Samsung

### Token Suspension

Token Requester sends a request to token service to suspend a token.

Token service notifies Token Requester of the status change although TSP choose Sync case.

Token Requester sends a push notification to device. Payment framework picks up the message and checks the token status through Token Requester.

Wallet app gets notified on the status change and handles the event. Wallet app acknowledges the status change to payment

framework. Payment framework reports to Token Requester.

Token Requester may send acknowledgement to token service.

### Token Resumption

Token Requester sends a request to token service to unsuspend a token.

Token service notifies Token Requester of the status change.

Token Requester sends a push notification to device. Payment Framework picks up the message and checks the token status through Token Requester.

Wallet app gets notified on the status change and handles the event.

Wallet app acknowledges the status change to payment framework. Payment framework reports to Token Requester.

Token Requester may send acknowledgement to token service.

**Lifecycle initiated from the Samsung: Suspended or Resume**

### Token Termination

#### Synchronous

Wallet app request PF to delete a card. PF sends the request to Token Requester.

Token Requester sends a request to token service for deleting a token.

Token service will return “200 OK” after successfully deleting the token or return “204 No Content” if token is already deleted.

Token Requester / Payment framework / Wallet apps will remove token from their storage.

#### Asynchronous

Wallet app request PF to delete a card. PF sends the request to Token Requester.

Token Requester sendsa request to token service for deleting a token.

Token service sends a notification to Token Requester after successfully deleting the token.

Token Requester sends a push notification to device.

Payment framework picks up the message and query Token Requester to confirm if the token is deleted.

Once confirmed, payment framework notifies the event to wallet app. Wallet apps reports to Token Requester through payment framework.

Token Requester may send acknowledgement to token service.

**Lifecycle initiated from the Samsung: Delete Token**

## Flow triggered by Issuer

Token service sends notification to Token Requester on token status change.

Token Requester sends push messages to device. Payment framework checks token status and notify to wallet app.

Wallet app handles the status change and reports back to Token Requester through payment framework.

Token Requester may send acknowledgement to token service.

### Token Suspension / Resumption

**Lifecycle initiated from the Issuer: Suspended or Resume**

### Token Termination

**Lifecycle initiated from the Issuer: Delete Token**

## Device Token Metadata Change (Update Card)

Payment network send metadata notification to TR. Wallet app fetches the latest metadata from the TR.

TSP should send notification for Metadata again for token that did not send report for Metadata changed from client for a period time after sending a notification for Metadata.

TSP did not receive Metadata Changed report from client, it means client fails to fetch the latest metadata thus TSP should send notification for Metadata again.

**Token metadata change**

## Methods

| **Method name** | **API [TR API]**  **TR => TSM** | **API [TR API]**  **TR <= TSM** | **API [IS\_INTERFACE]**  **TSM => Issuer** | **API [IS\_INTERFACE]**  **TSM <= Issuer** |
| --- | --- | --- | --- | --- |
| PATCH /tokens{id} | [Suspend token](https://confluence.oberthur.com/display/SPAYT/Suspend+token) |  |  |  |
| PATCH /tokens{id} | [Resume token](https://confluence.oberthur.com/display/SPAYT/Resume+token) |  |  |  |
| PATCH /tokens{id} | [Delete token](https://confluence.oberthur.com/display/SPAYT/Delete+token) |  |  |  |
| POST /notifications |  | [Status change](https://confluence.oberthur.com/display/SPAYT/Status+change) |  |  |
| GET /tokens{id} | [Push token](https://confluence.oberthur.com/display/SPAYT/Push+token) |  |  |  |
| POST /reports | [Lifecycle Event](https://confluence.oberthur.com/display/SPAYT/Lifecycle+event) |  |  |  |
| POST /suspendOrResumeDeviceToken |  |  |  | [SuspendOrResumeDeviceTokenRequest](https://confluence.oberthur.com/display/PAYM/SuspendOrResumeDeviceTokenRequest) |
| POST /deleteDeviceToken |  |  |  | [DeleteDeviceTokenRequest](https://confluence.oberthur.com/display/PAYM/DeleteDeviceTokenRequest) |
| POST /deviceTokenStatusNotification |  |  | [DeviceTokenStatus](https://confluence.oberthur.com/display/PAYM/DeviceTokenStatus) |  |

## Samsung Token Status Management

If token status in ACTIVE, any parties can request for token status update (suspend or dispose).

If the last requested parties was card issuing bank, Only card Issuing bank can make the request to token status update.

To Activate SUSPENDED token which is changed by Issuer, Only card Issuing bank can make the request to token status update (RESUME).

| **Token Status** | **instruction.value:RESUME - FMM Unlock - Issuer Initiated** | **instruction.value:SUSPENDED - FMM Lock - Issuer Initiated** | **instruction.value: DISPOSED - FMM Factory Reset - Delete Token By PF - Issuer Initiated** |
| --- | --- | --- | --- |
| PENDING | X | X | O |
| ACTIVE | X | O | O |
| SUSPENDED | O | X | O |
| DISPOSED | X | X | X |

**Token status matrix**

**Token status diagram**

Input from Samsung

Suspension / Resumption

Token disposal

Synchronous

Asynchronous

Token Status Change

Metadata Notification

[Skip to end of metadata](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#page-metadata-end)

* Created by [WOJCICKI Dominik](https://confluence.oberthur.com/display/~DominWOJ), last modified on [Mar 09, 2017](https://confluence.oberthur.com/pages/diffpagesbyversion.action?pageId=105844439&selectedPageVersions=4&selectedPageVersions=5)

[Go to start of metadata](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#page-metadata-start)

* [DeviceToken Lifecycle Management - Issuer initiated](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#Usecases-lifecyclemanagement-DeviceTokenLifecycleManagement-Issuerinitiated)
  + [Device Token suspension](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#Usecases-lifecyclemanagement-DeviceTokensuspension)
  + [Device Token resumption](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#Usecases-lifecyclemanagement-DeviceTokenresumption)
  + [Device Token termination](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#Usecases-lifecyclemanagement-DeviceTokentermination)
* [Device Token Lifecycle Management – Samsung Pay initiated](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#Usecases-lifecyclemanagement-DeviceTokenLifecycleManagement%E2%80%93SamsungPayinitiated)
  + [Device Token deletion](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#Usecases-lifecyclemanagement-DeviceTokendeletion)
  + [Device Token suspension](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#Usecases-lifecyclemanagement-DeviceTokensuspension.1)
  + [Device Token resumption](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+lifecycle+management#Usecases-lifecyclemanagement-DeviceTokenresumption.1)

DeviceToken Lifecycle Management - Issuer initiated

Device Token suspension

| **Use Case ID & Name** | **UC #18 : Device Token suspension from the Issuer** |
| --- | --- |
| **Description** | The issuer requests a deviceToken to be suspended |
| **Priority** | LOW |
| **Pre-conditions** | * The user has an active token deployed on her device |
| **Triggers** | The issuer decides to suspend a customer deviceToken |
| **Main Scenario** | 1. The issuer sends a ***SuspendOrResumeDeviceTokenRequest*** 2. The OT DPS updates the token status to the requested state and notifies Samsung Pay servers about the new status by sending a **POST / notifications** 3. Samsung Pay Servers fetches the new device token status by sending a **GET / tokens / {reference}** 4. The OT DPS returns status=SUSPENDED |
| **Post-condition** | The token status is suspended. |

Device Token resumption

| **Use Case ID & Name** | **UC #19 : Device Token resumption from the Issuer** |
| --- | --- |
| **Description** | The issuer requests a deviceToken to be resumed |
| **Priority** | LOW |
| **Pre-conditions** | * The user has an suspended token deployed on her device |
| **Triggers** | The issuer decides to resume a customer deviceToken |
| **Main Scenario** | 1. The issuer sends a ***SuspendOrResumeDeviceTokenRequest*** 2. The OT DPS updates the token status to the requested state and notifies Samsung Pay servers about the new status by sending a **POST / notifications** 3. Samsung Pay Servers fetches the new device token status by sending a **GET / tokens / {reference}** 4. The OT DPS returns status=ACTIVE |
| **Post-condition** | The token status is active. |

Device Token termination

| **Use Case ID & Name** | **UC #20 : Device Token deletion from the Issuer** |
| --- | --- |
| **Description** | The issuer requests a deviceToken to be terminated |
| **Priority** | MEDIUM |
| **Pre-conditions** | * The user has an active token deployed on her device |
| **Triggers** | The issuer decides to terminate a customer deviceToken |
| **Main Scenario** | 1. The issuer sends a ***DeleteDeviceTokenRequest*** 2. The OT DPS updates the token status to the requested state and notifies Samsung Pay servers about the new status by sending a **POST / notifications** 3. Samsung Pay Servers fetches the new device token status by sending a **GET / tokens / {reference}** 4. The OT DPS returns status=DELETED |
| **Post-condition** | The token status is deleted. |

Device Token Lifecycle Management – Samsung Pay initiated

Device Token deletion

| **Use Case ID & Name** | **UC #21 : Device Token deletion from Samsung Pay** |
| --- | --- |
| **Description** | The user requests a deviceToken to be deleted |
| **Priority** | MEDIUM |
| **Pre-conditions** | * The user has an active token deployed on her device |
| **Triggers** | The user decides to delete a deviceToken from his Samsung Pay wallet |
| **Main Scenario** | 1. Samsung Pay servers send a **PATCH / Tokens / {reference}** request 2. The OT DPS updates the token status to the requested state and save the associated reason. 3. The OTDPS notifies the issuer of the new deviceToken status using ***DeviceTokenStatusChangeNotification***(DELETED) with the associated reason. |
| **Post-condition** | The token status is deleted. |

Device Token suspension

| **Use Case ID & Name** | **UC #22 : Token suspension from Samsung Pay** |
| --- | --- |
| **Description** | The user requests a deviceToken to be suspended |
| **Priority** | MEDIUM |
| **Pre-conditions** | * The user has an active token deployed on her device |
| **Triggers** | The user decides to suspend a deviceToken from his Samsung Pay wallet |
| **Main Scenario** | 1. Samsung Pay servers send a **PATCH / Tokens / {reference}** request 2. The OT DPS updates the token status to the requested state 3. The OTDPS notifies the issuer of the new deviceToken status using ***DeviceTokenStatusChangeNotification***(SUSPENDED) |
| **Post-condition** | The token status is SUSPENDED. |

Device Token resumption

| **Use Case ID & Name** | **UC #23 : Token resumption from Samsung Pay** |
| --- | --- |
| **Description** | The user requests a deviceToken to be resumed |
| **Priority** | MEDIUM |
| **Pre-conditions** | * The user has a suspended token deployed on her device |
| **Triggers** | The user decides to resume a deviceToken from his Samsung Pay wallet |
| **Main Scenario** | 1. Samsung Pay servers send a  **PATCH / Tokens / {reference}** request 2. The OT DPS updates the token status to the requested state 3. The OTDPS notifies the issuer of the new deviceToken status using ***DeviceTokenStatusChangeNotification***(ACTIVE) |
| **Post-condition** | The token status is ACTIVE. |

[Skip to end of metadata](https://confluence.oberthur.com/display/SPAYT/Token+replenishment#page-metadata-end)

* Created by [WOJCICKI Dominik](https://confluence.oberthur.com/display/~DominWOJ), last modified by [VALADIER Jean-Louis](https://confluence.oberthur.com/display/~jeanlval) on [Oct 07, 2019](https://confluence.oberthur.com/pages/diffpagesbyversion.action?pageId=149181366&selectedPageVersions=4&selectedPageVersions=5)

[Go to start of metadata](https://confluence.oberthur.com/display/SPAYT/Token+replenishment#page-metadata-start)

**Token Replenishment**

Device and server can trigger replenishment.

When server triggers, Token service sends a notification to Token Requester to trigger token replenishment. Token Requester sends a push message to device. Payment framework on device picks up the message and requests token replenishment to Token Requester.

Payment framework may also trigger replenishment on a certain condition. Token Requester processes the request and returns a new token use key to payment framework. Payment framework stores the new key into trust zone and notified wallet app of the event. Wallet app acknowledge to payment framework. Payment framework reports to Token Requester. Token Requester may send acknowledgement to token service.

**Token Replenishment (mPure tokens with STET)**

**Same process as the one applied for HCE solution.**

In case of CB (Mpure) the trigger is managed by IDEMIA (receiving inputs from STET about the usage of ATCs).

The provisionned ATC is managed within IDEMIA solution, therefore the solution is able to computed a new set of keys with the associated ATCs

Once pushed to Samsung, the list of provisioned ATCs to pushed to STET.

**Token replenishment**

Methods

| **Method name** | **API [TR API]**  **TR => TSM** | **API [TR API]**  **TR <= TSM** | **API [IS\_INTERFACE]**  **TSM => Issuer** |
| --- | --- | --- | --- |
| POST /notifications |  | [Replenishment](https://confluence.oberthur.com/display/SPAYT/Replenishment) |  |
| GET /tokens{id} | [Push token](https://confluence.oberthur.com/display/SPAYT/Push+token) |  |  |
| POST /reports | [Replenishment event](https://confluence.oberthur.com/display/SPAYT/Replenishment+event) |  |  |
|  |  |  |  |

Input from Samsung

Replenishment

**Token Replenishment (VISA tokens with VTS)**

To be completed

[Skip to end of metadata](https://confluence.oberthur.com/display/SPAYT/Payment+transaction#page-metadata-end)

* Created by [WOJCICKI Dominik](https://confluence.oberthur.com/display/~DominWOJ), last modified on [May 08, 2018](https://confluence.oberthur.com/pages/diffpagesbyversion.action?pageId=87689243&selectedPageVersions=8&selectedPageVersions=9)

[Go to start of metadata](https://confluence.oberthur.com/display/SPAYT/Payment+transaction#page-metadata-start)

Description

Issuer or payment network send transaction notification directly to device. Wallet app fetches transaction details from the issuer or payment network transaction data service.

Flow

**Payment Transaction Notification**

Methods

| **Method name** | **API [TR API]**  **TR => TSM** | **API [TR API]**  **TR <= TSM** | **API [IS\_INTERFACE]**  **TSM => Issuer** | **API [IS\_INTERFACE]**  **TSM <= Issuer** |
| --- | --- | --- | --- | --- |
| POST /paymentTransactionNotification |  |  |  | [PaymentTransactionNotification](https://confluence.oberthur.com/display/PAYM/PaymentTransactionNotification) |
| POST /notifications |  | [Transaction](https://confluence.oberthur.com/display/SPAYT/Transaction) |  |  |
| GET /transactions?token.reference={token.reference} | [Get Token Transactions](https://confluence.oberthur.com/display/SPAYT/Get+Token+Transactions) |  |  |  |
| POST /reports | [VALADIER Jean-Louis](https://confluence.oberthur.com/display/~jeanlval) |  |  |  |

CA transaction policy

Regarding the request to support “**since**”, it is supported today (Even if defined as optional in the Samsung specs)

Sum-up:

1/ Our server stores/keeps up to the last 20 transactions

2/ Based on the Samsung request we provide either:

* Without *since* parameter: All transactions performed in the last 7 days (Maximum of 20)
* With *since* parameter: All transactions performed since the provided date (Maximum of 20)

3) the list of “M” fields (that we are providing/supporting), please find below the list of “Optional” fields that:

* Transaction “DPAN”

[Use cases - transaction](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+transaction)

[Skip to end of metadata](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+transaction#page-metadata-end)

* Created by [WOJCICKI Dominik](https://confluence.oberthur.com/display/~DominWOJ), last modified on [Mar 03, 2017](https://confluence.oberthur.com/pages/diffpagesbyversion.action?pageId=105844491&selectedPageVersions=1&selectedPageVersions=2)

[Go to start of metadata](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+transaction#page-metadata-start)

* [Transaction Management](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+transaction#Usecases-transaction-TransactionManagement)
  + [New transaction reported by the Issuer or TSP](https://confluence.oberthur.com/display/SPAYT/Use+cases+-+transaction#Usecases-transaction-NewtransactionreportedbytheIssuerorTSP)

Transaction Management

New transaction reported by the Issuer or TSP

| **Use Case ID & Name** | **UC #25 : New transaction notifies by Issuer or TSP** |
| --- | --- |
| **Description** | The issuer or  TSP notifies the TSM about a new transaction |
| **Priority** | LOW |
| **Pre-conditions** | * The user has at least a token on her Samsung Pay wallet * The user performs a payment with his deviceToken |
| **Triggers** | The Issuer or TSP detects a transaction is performed and notifies OT DPS about it |
| **Main Scenario**  **New card product** | 1. The Issuer or TSP notifies the OT DPS about a new transaction 2. The OT DPS notifies Samsung Pay servers about this new transaction by sending a **POST / notification** message 3. Samsung Pay servers sends a GET /transactions?token.reference={token.reference} request to the OT DPS 4. The OT DPS returns the list of the transaction details for the token that have been added or updated from requested timestamp. |
| **Post-condition** | Samsung Pay servers have a list of all transactions including the last one. |