Overview

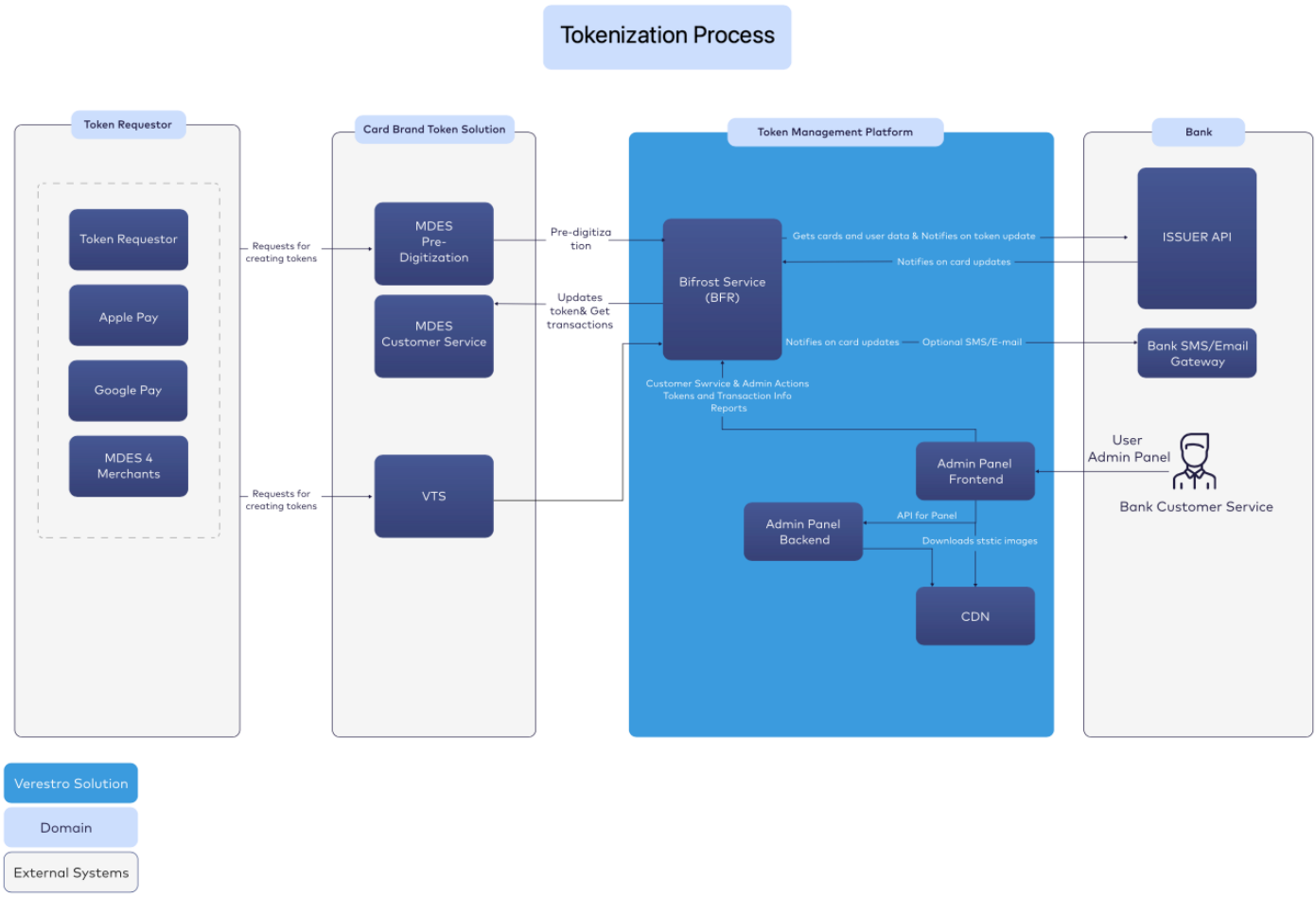
Verestro Token Management Platform is a solution created in order to allow much easier connection to Token Service Providers (TSP) - MDES, VTS. That can be used for card „pre-digitization” from all Token requestors with minimum development on the issuer side. It consists of the following parts:

- Predigitization API - set of processes and requirements that must happen before the payment token becomes ready for use - it will be possible to make payments.
- Token LifeCycle API - Mastercard or Visa API that TMP connects to in order to manage token life cycle.
- Admin Panel - Administration Panel for creating/fetching reports and managing token life cycle - can be used by Issuer's Customer Service.
- PushProvisioning API - allow card issuers the ability to initiate the card provisioning process for Apple/Google Wallet/Click2Pay directly from app.

Benefits for issuing bank or fintech partner

- TMP is created to connect to TSP(MDES/VTS) and enable much easier integration for the Issuer.
- TMP integrates with Token Service Providers (Mastercard MDES, Visa VTS) and provides a single interface for the issuers, so issuers don't have to integrate with both TSP.
- TMP supports various Token Requestors.
- TMP supports different requirements and implementations recommended by Token Requestors.
- TMP has audit and reporting capabilities for the Issuer including Apple Pay reports.
- TMP provides the Token and (optionally) Card Lifecycle Management API.
- TMP provides Admin Panel.
- TMP provides customizable Rule Engine, with risk rules kept in line with Token Requestors and Payment Networks recommendations, allowing issuer to make easier and more reliable tokenization authorizations decisions.
- TMP supports notifications including reminders for the users.
- TMP supports token requestor based velocity controls.
- TMP supports automated token lifecycle management.
- TMP supports In App Push Provisioning and supports In App token activation method.

High Level Overview



Key components

Component	Description
Bank Customer Service	Service that allows direct one-on-one interaction between a consumer using a card and a representative of the bank.
Bank SMS/Email Gateway	Bank Service to send OTP via SMS or email to users. Used only when Issuer wants to send sms/email by themselves.
Verestro SMS/Email Gateway	Verestro TMP gateway to send OTP via SMS or email to users.
Bifrost Service (Verestro TMP backend)	Component responsible for the pre-digitization, token/card lifecycle, reports and push provisioning.
Push Provisioning ApplePay/GooglePay	Allows a mobile application to push a card to a token requestor.

Issuer	The bank or institution responsible for issuing the cardholder their card (debit, credit, prepaid).
MDES/VTS	Token Service Provider which supports digitization (transforming the card into payment token) and is responsible for management, generation and provisioning of transaction credentials to Token Requestor to enable simpler and more secure digital payment experience.
MDES Pre-digitization	MasterCard Digital Enablement Service API with methods to perform pre-digitization process.
Verestro TMP Admin Panel	Panel that can be used by bank for token LifeCycle management, administration and viewing reports.
Token Requestor	Is an entity that requests payment tokens for end-users. Some examples of TRs include digital wallet providers, payment enablers, merchants and IoT manufacturers.

Features overview

Verestro Token Management Platform supports below functionalities:

Area	**Functionalities**
Interface (Payment network APIs)	Mastercard Pre-Digitization API Mastercard Customer Service API Visa Token Service Inbound API Visa Token Service Outbound API
Apple Pay	Manual provisioning Push provisioning Apple Pay payload encryption Apple Pay card list
Google Pay	Manual provisioning Push provisioning Google Pay payload encryption Google Pay card list

XPays (Samsung Pay, Fitbit Pay, etc.)	Manual provisioning Push provisioning Payload encryption Card list
Token Management Portal	Web UI All tokens available at the same time through the same interface Rules & tokenization decisions overview
Identity Verification	SMS OTP Email OTP InApp Verification
Reports	Apple mandatory report Standard report (includes user, device, token, payment history data)
Card Lifecycle Management	Automated Card Lifecycle Management Single Token Lifecycle Management Automated removal of inactive tokens
Decision Paths & Fraud Prevention	Wallet Provider recommendation Wallet Provider reason codes Wallet Provider Device score Account number source Wrong SMS OTP Velocity Wrong CV2 or Exp.Date Velocity Token Limit per Card Token Limit per Device Geolocation Verification Device IP Address Verification Mobile Phone Number Marching

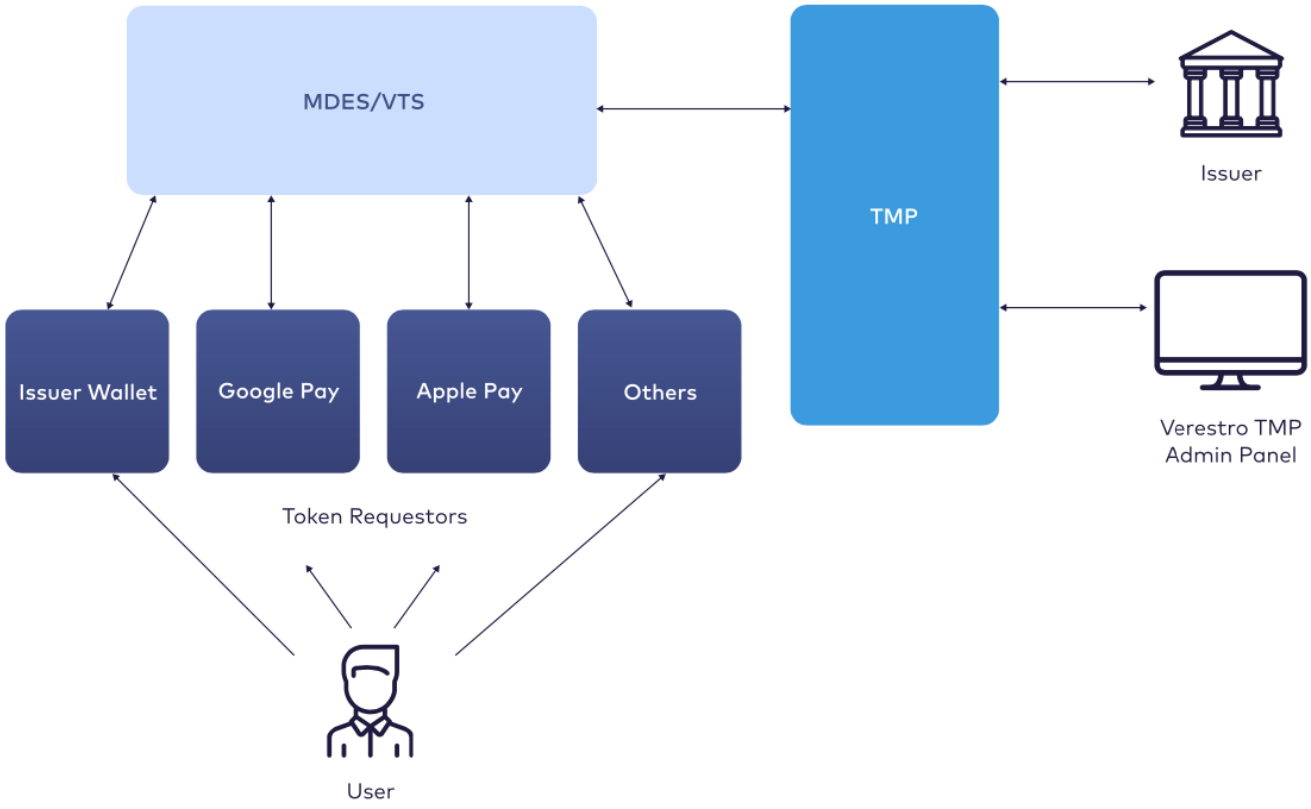
Verestro TMP solution

Verestro Token Management Platform is solution which has been developed to facilitate the process of pre-digitizing cards for the Issuer.

Solution consists of:

- Token Management Platform (Server solution) - backend component.
- TMP Admin Panel - frontend component, dedicated to issuer's customer service teams.

Architecture



Pre-digitization

Pre-digitization is a set of processes that allows to a generation of digital payment tokens to enable simpler and secure digital payment experiences. Simply it turns a payment card into a digital token. In this process, Verestro TMP is taking care of all the requirements from Token Requestors.

For this process, the Issuer needs to expose one API method, which will return card verification result or security code verification result.

Tokenization process
1. User enters the card into Apple Pay/Google Pay or another Token Requestor wallet.

2. TMP receives Authorize Service request from TSP(MDES/VTS) on Pre-digitization API with Card Number, CVC, Exp Date, Device Score, and other tokenization data provided by Token Requestor.

3. TMP checks device score, number of already active tokens, velocity controls and other risk rules, relevant for the case.

4. TMP sends a request to Issuer Card Verification API with a Card Number and receives the Card Status, Card ID, User Phone Number, CVC validation Result, Product Category.

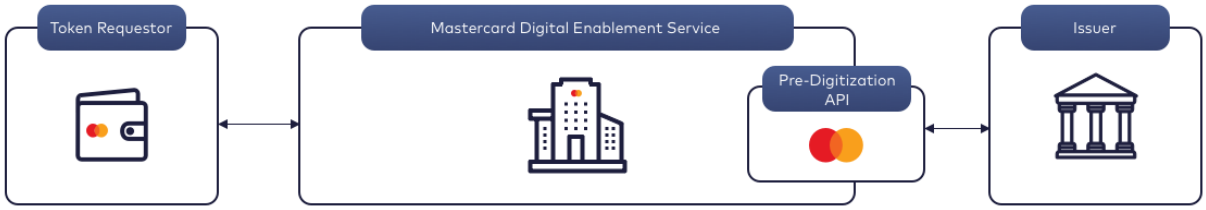
5. TMP returns the decision to TSP (APPROVED/REQUIRE_ADDITIONAL_AUTHENTICATION/DECLINED).

Token activation
If the decision is APPROVED - token is activated instantly after Authorize Service response. Verestro TMP can also notify the issuer if required.

If the decision is REQUIRE_ADDITIONAL_AUTHENTICATION - The message will be displayed to the user with activation options (ex. SMS OTP). After the user selects the activation type, TSP will send a DeliverActivationCode to Verestro TMP. Verestro TMP will send the OTP activation code to the user. After the user enters the OTP, TSP activates the token. The token can also be activated manually via the Administration Panel.

If the decision is DECLINE - a token becomes INACTIVE and cannot be activated again.

When a token is activated, Verestro TMP will receive a notifyServiceActivated call from TSP.



User authentication

There are 4 authentication paths for the user:

- Green Path - Path without user confirmation (authentication) during the token activation process. The payment token is automatically activated.
- Yellow Path - Path with user confirmation (authentication) during the token activation process. Payment token is activated after correct OTP is provided.
- Orange Path - Path with user confirmation (authentication) during the token activation process. Payment token is activated by the Bank after the user's request via call.
- Red Path - Path when the Issuer rejected activation payment token during the token activation process.

Pre-digitization API Sequence Diagram

@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF

```
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
title Green Path
actor User
'comment: actor boundary control entity
User -> "Token Requestor": 1. Tokenize Card
activate "Token Requestor"
"Token Requestor" -> "MDES": 2. AuthorizeService request
activate "MDES"
"MDES" -> "TMP": 3. AuthorizeService request
activate "TMP"
"MDES" <-- "TMP": 4. AuthorizeService response (APPROVED)
"Token Requestor" <-- "MDES": 5. AuthorizeService response (APPROVED)
User <-- "Token Requestor": 6. APPROVED
"MDES" --> "TMP": 7. NotifyServiceActivated
deactivate "TMP"
"MDES" --> "Token Requestor": 8. Service Activated
deactivate "MDES"
"Token Requestor" --> User: 9. Service Activated
deactivate "Token Requestor"
@enduml

@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
```

```
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
title Yellow Path
actor User
'comment: actor boundary control entity
User -> "Token Requestor": 1. Tokenize Card
activate "Token Requestor"
"Token Requestor" -> "MDES": 2. AuthorizeService request
activate "MDES"
"MDES" -> "TMP": 3. AuthorizeService request
activate "TMP"
"MDES" <-- "TMP": 4. AuthorizeService response (RAA)
"Token Requestor" <-- "MDES": 5. AuthorizeService response (RAA)
User <-- "Token Requestor": 6. Activation Methods
User -> "Token Requestor": 7. Choose Activation Method
"Token Requestor" -> "MDES": 8. Choose Activation Method
"MDES" -> "TMP": 9. DeliverActivationCode
"TMP" --> User: 10. DeliverActivationCode (SMS, EMAIL)
deactivate "TMP"
User -> "Token Requestor": 11. Enter activation code
"Token Requestor" -> "MDES": 12. Validate activation code
"MDES" --> "Token Requestor": 13. Service Activated
deactivate "MDES"
"Token Requestor" --> User: 14. Service Activated
deactivate "Token Requestor"
@enduml

@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
```

```
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
title Red Path
actor User
'comment: actor boundary control entity
User -> "Token Requestor": 1. Tokenize Card
activate "Token Requestor"
"Token Requestor" -> "MDES": 2. AuthorizeService request
activate "MDES"
"MDES" -> "TMP": 3. AuthorizeService request
activate "TMP"
"MDES" <-- "TMP": 4. AuthorizeService response (DECLINE)
deactivate "TMP"
"Token Requestor" <-- "MDES": 5. AuthorizeService response (DECLINE)
deactivate "MDES"
User <-- "Token Requestor": 6. Decline
deactivate "Token Requestor"
@enduml
```

Deliver activation code.

This method is called when authorize service returned decision: REQUIRE_ADDITIONAL_AUTHNETICATION(Yellow Path). Account Holder needs to verify himself with one of the available activation methods (e.g. OTP code or call to call center). OTP code can be generated by Verestro TMP or TSP(preferred option).

Verification steps:

- Verestro TMP sends OTP code via SMS or email (configurable option) to the Account Holder, but there is also possibility to do that by the Issuer, in that case Verestro TMP will notify the Issuer and then Issuer sends it to the Account Holder,
- Account Holder is entering received OTP and TSP or Verestro TMP(configurable) is validating it,
- When OTP code is correct, notifyServiceActivated method is called which means that token is activated and ready to use.

MDES Pre-digitization API technical

The process of predigitization presents a set of methods by which Issuer can easily digitize his card. The diagram below shows the flow from the moment the User starts digitizing his card, through its authentication with the activation code (Deliver Activation Code) to the activation of the payment token.

```
@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
actor User
participant "Token Requestor" as TR
participant "Mastercard Digital Enablement Service" as MDES
participant "Token Management Platform" as TMP
participant Issuer as Issuer
User -> TR: 1. Digitize
activate TR
TR -> MDES: 2. Digitize
activate MDES
MDES -> TMP: 3. AuthorizeService
activate TMP
TMP -> Issuer: 4. CardValidation
activate Issuer
Issuer --> TMP: 5. Card Validation Response
deactivate Issuer
TMP --> MDES: 6. Decision and List of Activation Methods
deactivate TMP
MDES --> TR: 7. List of Activation Methods
deactivate MDES
TR --> User: 8. List of Activation Methods
deactivate TR
```

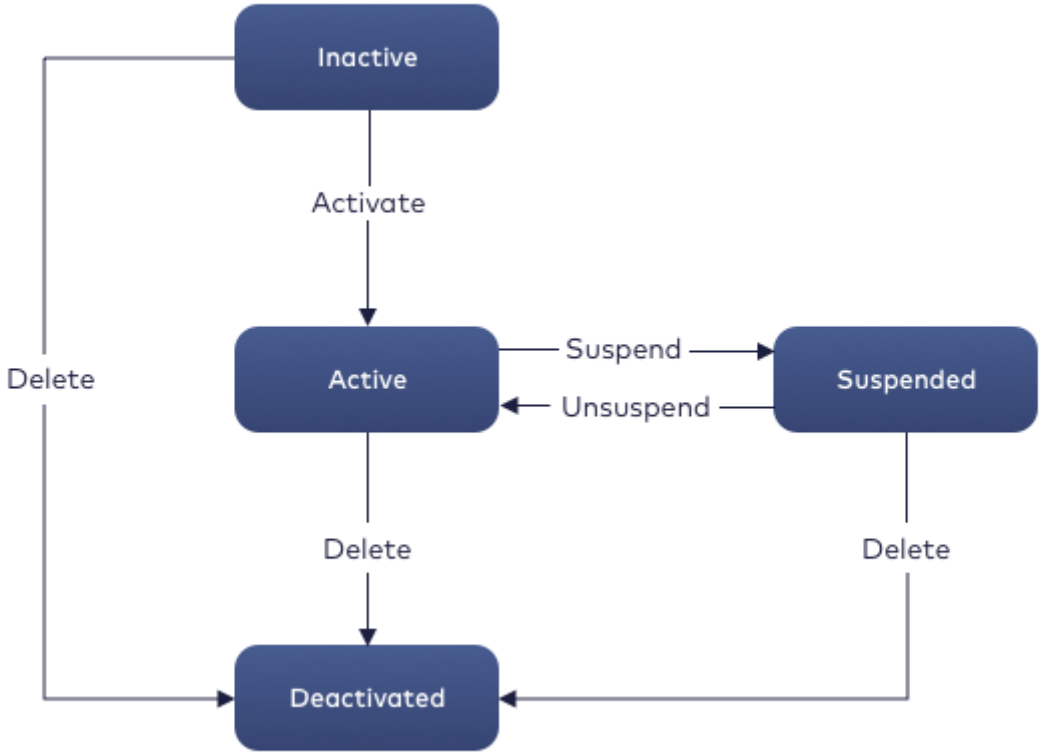
deactivate User
note left: Activation method selected
User -> TR: 9. Deliver Activation Code
activate TR
TR -> MDES: 10. Deliver Activation Code
activate MDES
MDES -> TMP: 11. Deliver Activation Code
activate TMP
TMP -> Issuer: 12. Deliver Activation Code
activate Issuer
Issuer --> User: 13. Send activation code using preferred channel
Issuer --> TMP: 14. Activation Code Delivered
deactivate Issuer
TMP --> MDES: 15. Activation Code Delivered
deactivate TMP
MDES --> TR: 16. Activation Code Delivered
deactivate MDES
TR --> User: 17. Activation Code Delivered
deactivate TR
User -> TR: 18. Enter activation code
activate TR
TR-> MDES: 19. Activate
activate MDES
MDES -> MDES: 20. Token
MDES -> MDES: 21. Token activated (TUR)
MDES -> TMP: 22. NotifyTokenUpdated (TUR, Active)
activate TMP
MDES -> TMP: 23. NotifyServiceActivated
TMP -> Issuer: 24. NotifyServiceActivated
deactivate TMP
MDES -> TR: 25. NotifyServiceActivated
deactivate MDES
TR --> User: 26. Token Activated
deactivate TR
@enduml

Lifecycle

Token lifecycle support token management which can be use directly by user, issuer or customer service using Admin Panel. This feature provides action on token to change token status. Actions what can happened are:

Acivate token → change token status to Active,
Suspend token → change token status to Suspended,
Unsuspend token → change token status to Active,
Delete token → change token status to Deactivated,

The diagram below shows the transitions between payment token statuses.



Automatic lifecycle management is supported via Verestro TMP API. Issuer can call Verestro TMP API to perform any lifecycle actions on card with will result on lifecycle action on all tokens assigned for the card.

Example: User looses his card and calls bank customer services to block the card, bank blocks user's card and automatically calls Verestro TMP to blocks all the tokens for the same card. This solution is compliant with apple automated lifecycle management requirements

Payment Token Lifecycle Management

These diagrams show what the 4 authentication paths look like for a user:

This section describes payment token lifecycle management performed via Issuer or Admin Panel.

Token lifecycle options:

– Token can be suspended/deleted from the consumer app (Apple Pay/ Google Pay). Verestro TMP will receive a TokenUpdate notification from TSP.

– Token can be activated/suspended/unsuspended/deactivated from Administration Panel UI by the Issuer Customer Service Support team.

– Token can be activated/suspended/unsuspended/deactivated via Verestro TMP API.

– All card tokens can be suspended/unsuspended/deactivated via Verestro TMP API. Card renew and replace is also supported.

Thera are a possibility to turn on an automatic notifications depends on any actions on token e. g token was activated or customer didn't finalize token activation.

Suspend, Unsuspend, Deactivate the token

Payment token can be suspended by the Issuer or the Admin Panel.

```
@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
actor User
participant "Token Requestor" as TR
participant "Mastercard Digital Enablement Service" as MDES
participant "Token Management Platform" as TMP
participant "Admin Panel" as AP
AP -> TMP: 1. suspend/unsuspend/deactivate token
activate AP
activate TMP
TMP -> TMP: 2. check the permissions
alt if permissions are compatible
TMP -> MDES: 3. suspend/unsuspend/deactivate token
activate MDES
else if permissions disagree
TMP --> AP: 4. response
end
MDES -> MDES: 5. suspend/unsuspend/deactivate token
```

MDES --> TMP: 6. response
TMP --> AP: 7. response
deactivate AP
deactivate TMP
MDES -> TR: 8. token suspend/unsuspend/deactivate
deactivate MDES
activate TR
TR -> TR: 9. suspend/unsuspend/deactivate token
TR --> User: 10. notification
deactivate TR
@enduml

Payment token can be suspended by the Issuer.

@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
actor User
participant "Token Requestor" as TR
participant "Mastercard Digital Enablement Service" as MDES
participant "Token Management Platform" as TMP
participant Issuer as Issuer
Issuer -> TMP: 1. suspend/unsuspend/deactivate token
activate Issuer
activate TMP
TMP -> TMP: 2. check the permissions
TMP -> MDES: 3. suspend/unsuspend/deactivate token
activate MDES
MDES -> MDES: 4. suspend/unsuspend/deactivate token

MDES --> TMP: 5. response
TMP --> Issuer: 6. token suspended/unsuspended/deactivated
deactivate Issuer
deactivate TMP
MDES -> TR: 7. token suspend/unsuspend/deactivate
deactivate MDES
activate TR
TR -> TR: 8. suspend token
TR --> User: 9. notification
deactivate TR
@enduml

Replace and renew the token

Replace/renew payment token is the process with consists of updating payment token data. Every payment token has an expiry date and linked PAN. When the token is expiring, the Issuer can change the token expiration date.

When a cardholder has a new card (new PAN) and wants to keep the digital token active, it can be achieved by doing "PAN swap" for a token. The issuer can do it for one or all payment tokens associated with one PAN set to the new data. A new product configuration ID can also be associated with one or all tokens associated with a PAN.

The same situation may occur when the cardholder receives a new card (PAN) due to fraud / theft. The issuer changes the payment token data in the same way.

@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}

```
actor User
participant "Token Requestor" as TR
participant "Mastercard Digital Enablement Service" as MDES
participant "Token Management Platform" as TMP
participant Issuer as Issuer
Issuer -> TMP: 1. replace/renew token
activate Issuer
activate TMP
TMP -> TMP: 2. check the permissions
TMP -> MDES: 3. replace/renew token
activate MDES
MDES -> MDES: 4. replace/renew token
MDES --> TMP: 5. response
deactivate MDES
TMP --> Issuer: 6. token replaced/renewed
deactivate Issuer
TMP -> TR: 7. replace/renew token
deactivate TMP
activate TR
TR -> TR: 8. replace/renew token
TR --> User: 9. notification
deactivate TR
@enduml
```

Payment Card Lifecycle Management

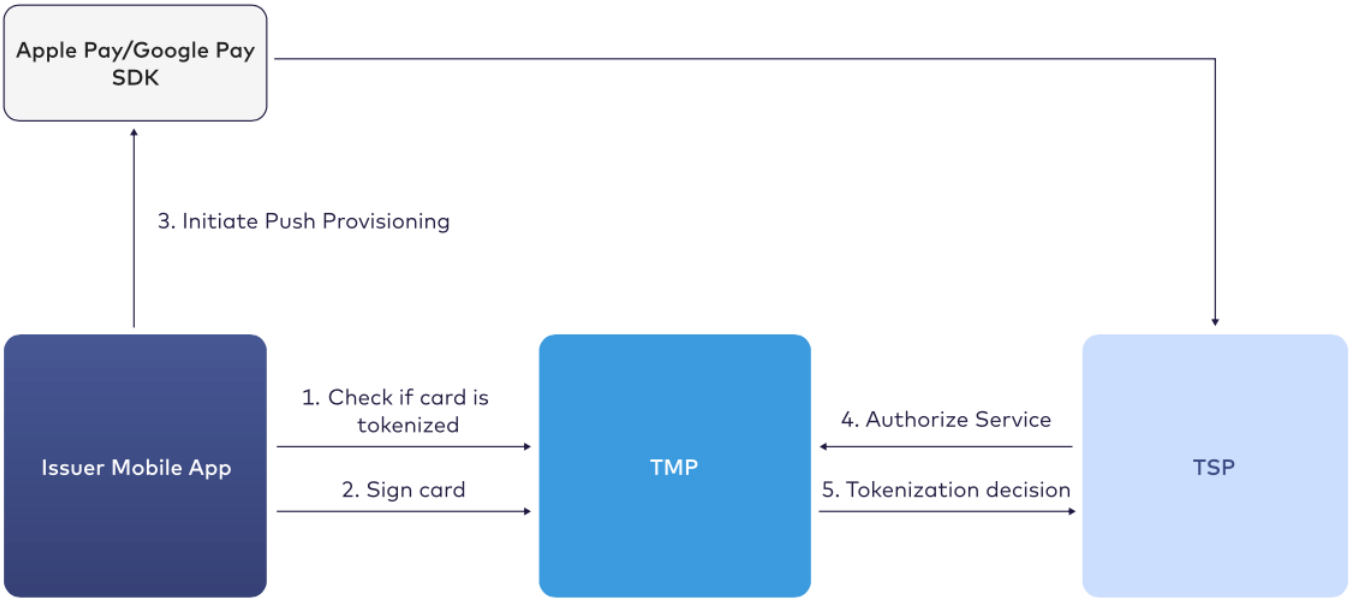
```
@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
```

participant "Issuer" as Issuer
participant "Verestro Token Management Platform" as TMP
participant "Customer Service API" as CSAPI
Issuer -> TMP: 1. Suspend card (CardId)
activate Issuer
activate TMP
TMP -> TMP: 2. Find all tokens for card
TMP --> CSAPI: 3. Suspend tokens
activate CSAPI
TMP --> Issuer: 4. Token Suspended
@enduml

Push Provisioning

Push Provisioning provides the ability to initiate the card provisioning process for Apple/Google Wallet directly from the issuer's app.

Users will find the Push Provisioning feature an extremely convenient method to provision their cards or passes into their devices by avoiding the need to input those details manually.



Verestro TMP Push Provisioning module API:

1. Check if card is tokenized.
2. Sign card .

Verestro TMP Push Provisioning module allows the following flow:

1. Check if card is tokenized - Return information if a card is tokenized on the device, so the Issuer's mobile application can show "Add to Apple/Google pay" button.
2. Sign card - Prepare encrypted and signed payload which can be used by the Issuer's mobile application.
3. Initiate Push Provisioning with Apple Pay/Google Pay SDK.
4. Authorize Service to Verestro TMP.
5. Tokenization Decision returned to TSP (APPROVE/DECLINE).

Admin Panel

Admin Panel for Verestro Token Management Platform is an user interface thanks to we can manage and view tokens and transaction data.

Due to admin panel operators can:

- browse tokens list and see token details,
- change/manage tokens statuses,
- browse transactions list and see transaction details (including statuses).
- generate reports.

Can be used by Issuer Customer Service.

More information we can find in the Verestro TMP Admin Panel Product Overview documentation.

Additional Features

Reports.

Verestro TMP can generate various types of reports: ApplePay, Tokens, Transactions, Activity audit. Reports can be downloaded or generated from Admin Panel. Some of the reports can be generated automatically by Verestro TMP.

User notifications.

Verestro TMP can send custom notifications to Issuer, like:

- OTP code for additional authentication.
- Notifications when a token is activated or deleted.
- Notifications to inactive customers, which didn't perform any transactions after token activation.
- Notifications on abandoned provisioning, when a user didn't finish the full process of token activation.

Jobs.

Verestro TMP can generate/notify or do some other custom task automatically, like:

- Delete inactive tokens after a configured time.
- Generate reports.
- Send notifications.
- Fetch transactions from Customer Service, which can be used for reporting and accessible from administration panel.

Monitoring and Alerting:

- Grafana dashboard with tokenization activity and performance metrics.
- Statistics.
- Error and warning alerting.

Security:

- IP whitelist for API communication.
- Role based access to lifecycle and reporting features.
- x.509 mTLS authentication.
- VPN tunnel support.
- OAuth when connecting to Issuer API.

Revision #48
Created 2022-06-10 06:00:21 UTC
Updated 2026-09-29 15:02:10 UTC by Aleksandra Szczesna