

# Cortina V2 External Prepaid API Support Document

Version V2R01 (Alpha release)

# Nayax



## Confidentiality Declaration

This document is the confidential property of Nayax Ltd. Any reproduction or distribution of this document, in whole or in part, or the disclosure of any of the information contained herein, without the prior written consent of Nayax Ltd. is prohibited. This document and any supplemental material supplied as part of any review must be returned to Nayax Ltd. upon request.

Notice: This document may go through material revisions over time. While the Company will make every effort to keep readers updated as changes occur, care must be taken by the reader to make sure they are reading the latest version. Version numbers appear in the title page and revision history page of this document.

© 2021 Nayax Ltd. All rights reserved. Nayax, the Nayax logo are either trademarks or registered trademarks of Nayax in the United States and/or other countries. All other trademarks or brand names are properties of their respective holders.

## Table of Content

|  |    |
|--|----|
| Revision History.....                                      | 5  |
| 1. Objective.....  | 6  |
| 2. Introduction to Cortina Flow .....                      | 7  |
| 3. Security measures: .....                                | 8  |
| 4. Setup Requirements:.....                                | 9  |
| 5. High Level Typical Flows - Legend: .....                | 11 |
| 6. Preauthorization Transaction Flow: .....                | 12 |
| 6.1 Preauthorization success scenario .....                | 12 |
| 6.2 Start Session Failed .....                             | 14 |
| 6.3 Authorization Request Failed .....                     | 15 |
| 6.4 Confused consumer Scenario .....                       | 16 |
| 6.5 Preauthorization Vend failed.....                      | 17 |
| 6.6 Preauthorization Settlement failed.....                | 18 |
| 6.7 Transaction cancel attempt failed.....                 | 19 |
| 7. Preselection Transaction Flow: .....                    | 20 |
| 7.1 Positive Preselection Scenario.....                    | 20 |
| 7.2 Confused customer scenario.....                        | 22 |
| 7.3 Start Session Failed .....                             | 23 |
| 7.4 Sale Request Failed .....                              | 24 |
| 7.5 Vend failed scenario.....                              | 25 |
| 7.6 sale end notification failed scenario .....            | 26 |
| 7.7 Void failed scenario .....                             | 27 |
| 8. Supported Methods: .....                                | 28 |
| 8.1 Timeout Handling Mechanism .....                       | 28 |
| 8.2 Methods .....  | 29 |
| 9.3 Methods Characterization .....                         | 31 |
| 9.4 Error Code Reasoning Table for supported methods ..... | 32 |
| 9. Appendix.....   | 33 |
| 9.1 Nayax's IP addresses for whitelist: .....              | 33 |
| 9.2 Cortina Security Encryption.....                       | 34 |

|   |    |
|---|----|
| 9.3 /StartSession Authentication & Example..... | 35 |
| 10. Examples (Yaml V2R01) .....                 | 37 |
| 10.1 /StartSession Example: .....               | 37 |
| 10.2 /Authorization Example: .....              | 37 |
| 10.3 /Settlement Example: .....                 | 39 |
| 10.4 /Sale Example: .....                       | 42 |
| 10.5 /Cancel Example: .....                     | 44 |
| 10.6 /Void Example: .....                       | 46 |
| 10.7 /Refund Example:.....                      | 48 |
| 10.7 /SaleEndNotification Example: .....        | 50 |

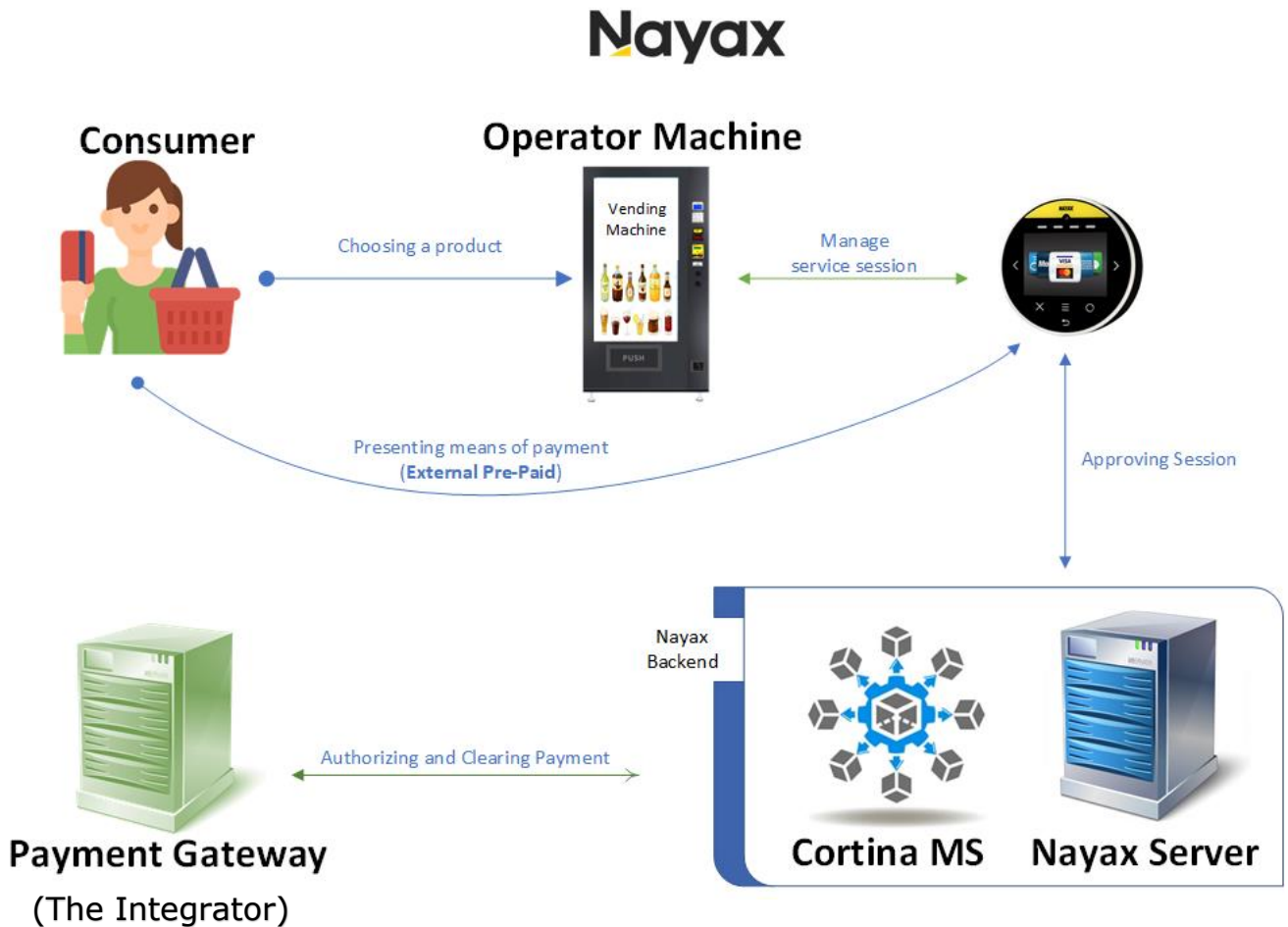
## Revision History

| Revision | Date       | Content                                     | Author      |
|----------|------------|---|-------------|
| V2R01    | 24/07/2021 | Cortina V2 External prepaid – Alpha Release | Lior Louzon |

## 1. Objective

The purpose of this document is to describe the Cortina v2 external Prepaid card transaction flow, as well as API calls and parameters. Payment Service Providers (aka: Integrators) may use the Nayax **External Prepaid (v2)** API to integrate with NAYAX's system and assure acceptance and payment processing.

Nayax **External Prepaid (v2)** goal is to allow operators to use the Nayax POS device to receive payments via external prepaid. In this scenario, the Nayax POS is connected directly to the Operator's machine.



## 2. Introduction to Cortina Flow

Cortina **External PP v2** supports 2 types of transactions:

1. **Preauthorization** – constructed of two separate operations:
  - a. Authorization request – Nayax request to temporarily hold credit in the consumer account balance.
  - b. Settlement request – Nayax request to complete the transaction and to charge partial / full amount from the original authorization request.

Basic consumer preauthorization journey:

### Preauthorization example illustration



Authorize means of payment and default amount



Receiving a service



Settle final amount

In preauthorization the price is being calculated based on actual service/products received after initial card authorization.

2. **Preselection** – performing a sale in one step preselection transaction. Nayax request to charge the consumer for a specific amount in a single request.

Basic consumer preselection journey:

### Pre-selection example illustration



Choosing a product



Proceed to payment



Receiving the product

In preselection the price is already known based on service/product that has chosen.

## 3. Security measures:

### 1. HTTPS Request:

The Cortina API is REST interface over HTTPS using JSON format as input and output.

### 2. Preventing Man-in-the-middle attack and security:

- Notify is the first method in the transaction flow which is called by Mobile App Provider server to notify Nayax of a customer check in. Nayax will send transaction requests (Authorization, Settlement etc.) to pre-defined URL and NOT to the URL that Notify call came from. This is for preventing Man-in-the-middle attack.
- All requests and responses are linked together by the Transaction Id parameter, generated by Mobile App Provider in Notify call, composing a complete transaction.

### 3. Using Security keys:

To keep high security level, Nayax provides unique secret token for each merchant. This token is a pairing of payment method id and integrator (technology provider). This value will be sent by the Mobile App Provider in Notify call and will be validated by Nayax.

Nayax may contact the Mobile App Provider to replace this value from time to time.

### 4. Authenticating Nayax Source (/StartSession) – mandatory

As per project the Integrator and Nayax shall scope the security requirements and may mandate the use of **Start Session Authentication**.

Implementation of this security method is detailed in the [StartSession Authentication](#) & Example in this document.

### 5. Using Encryption: - optional

To provide high level of message confidentiality and integrity, Nayax supports an option of encrypted communication. Every Nayax message contains an EncryptInfo Object that is available for the integrator choice. For more details on the encryption process please see the [Cortina Security Encryption](#).



## 4. Setup Requirements:

There are a few technical configurations required between Nayax and the integrator. Those requirements address security aspects, account setup, testing aspects and others technical prerequisites that the integrator will provide to the Nayax technical point of contact.

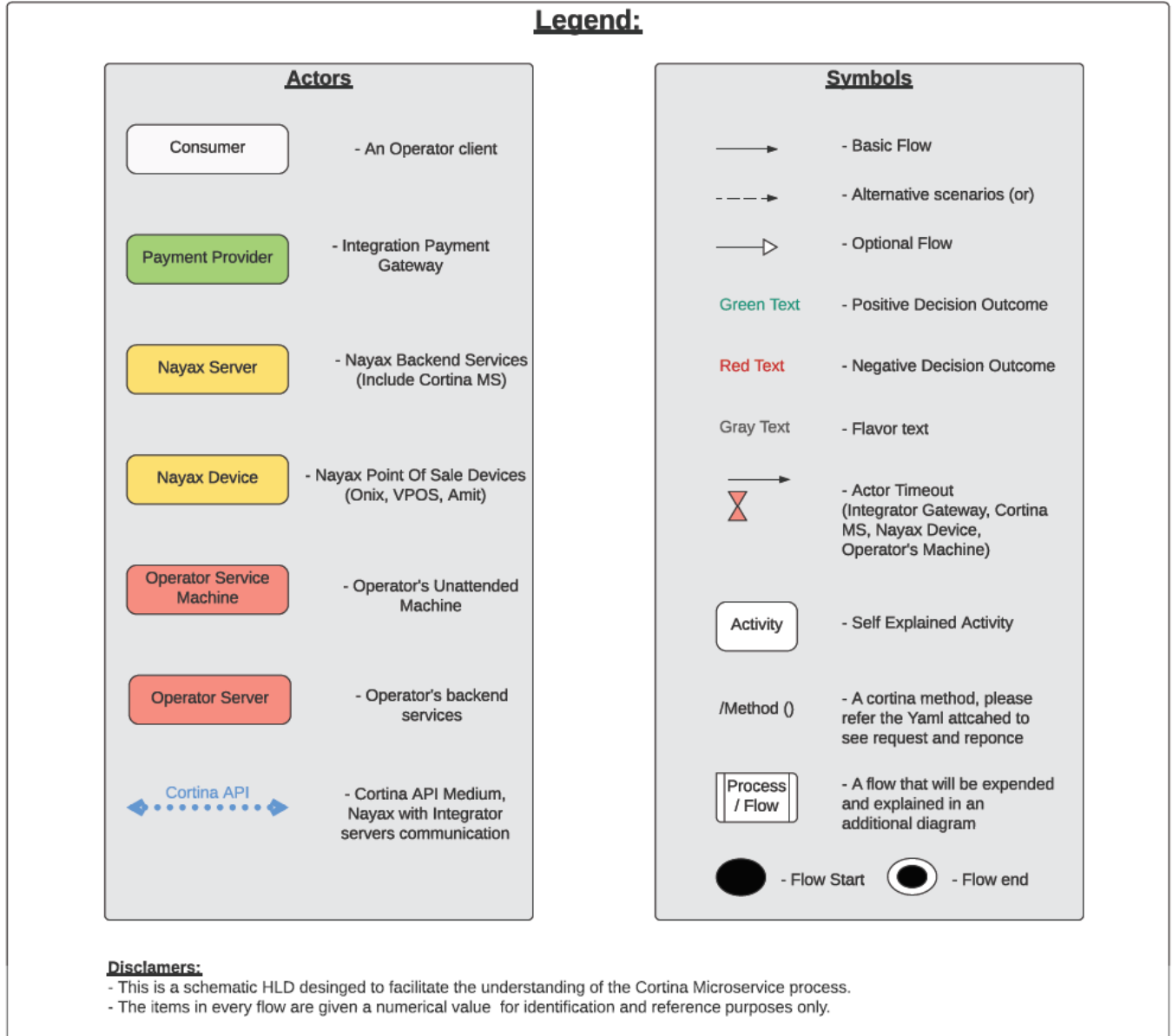
Below is a table with the prerequisites based on topics and providing entity:

|                      | <b>Integrator responsibility to provide</b>  | <b>Nayax responsibility to provide</b>   |
|----------------------|--|--|
| <b>Security</b>      | <ul style="list-style-type: none"> <li>HTTPS <b>test</b> endpoint URL               <ol style="list-style-type: none"> <li>For Cortina routes</li> </ol> </li> <li>HTTPS <b>Production</b> endpoint URL               <ol style="list-style-type: none"> <li>For Cortina routes.</li> </ol> </li> </ul>  | <ul style="list-style-type: none"> <li>IP Whitelisting if needed, see <a href="#">attached</a>.</li> <li>Secret Token (Value, ID)</li> <li>If encryption is used, consult your Nayax TPOC regarding the key generation.</li> </ul> |
| <b>Account Setup</b> | <ul style="list-style-type: none"> <li>provide <b>Nayax server timeout</b> when approaching integrator server. [default is 15 s. possible range is 1-15s]</li> </ul>   |  |
| <b>Testing</b>       | <ul style="list-style-type: none"> <li>In case of magnetic stripe cards, integrator may provide the actual cards or provide us its data. Any other case, integrator will provide physical test cards.</li> <li>Optional - Stub Test settings (Merchant ID / URL)</li> <li>Tests will be performed on the integrators side.</li> </ul>                                      | <ul style="list-style-type: none"> <li>Optional - Relevant Nayax POS devices</li> </ul>  |
| <b>Additional</b>    | <ul style="list-style-type: none"> <li>Optional - in case of a need to use the Balance object in the returning /Sale or /Authorize method, please contact Nayax TPOC for more information.               <pre>"Balance": {   "RegularCreditType": 0,   "RegularCredit": 0,   "RevalueCredit": 0 }</pre> </li> <li>In case of T/O in authorization or settlement</li> </ul> |  |

|  |   |  |
|--|---|--|
|  | <p>request, Nayax server shall perform by default, 2 attempts to cancel/void the transaction with 30 seconds intervals between. For more details see Immediate void for <a href="#">preauthorization</a> or <a href="#">preselection</a>, and contact Nayax TPOC.</p> <ul style="list-style-type: none"><li>• In preselection Transactions by default Nayax shall perform Sale end notification to inform the integrator that the operator has completed the transaction successfully. For more details contact Nayax TPOC.</li></ul> |  |
|--|---|--|

## 5. High Level Typical Flows - Legend:

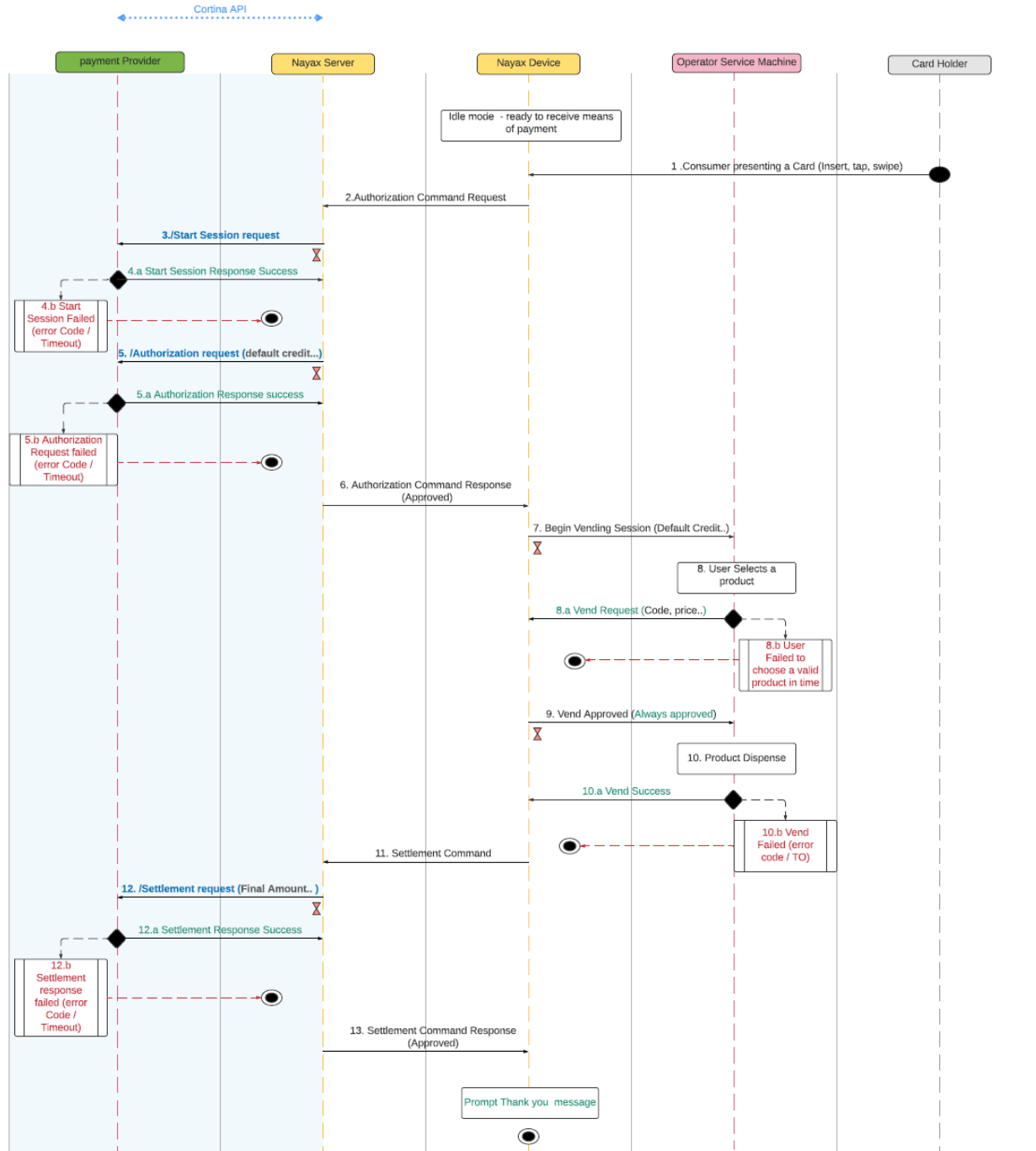
Cortina POS Remote Start V2, High level design legend.



## 6. Preauthorization Transaction Flow:

### 6.1 Preauthorization success scenario

Cortina External PrePaid - Preauthorization Process (Success Scenario)



**Notes:**

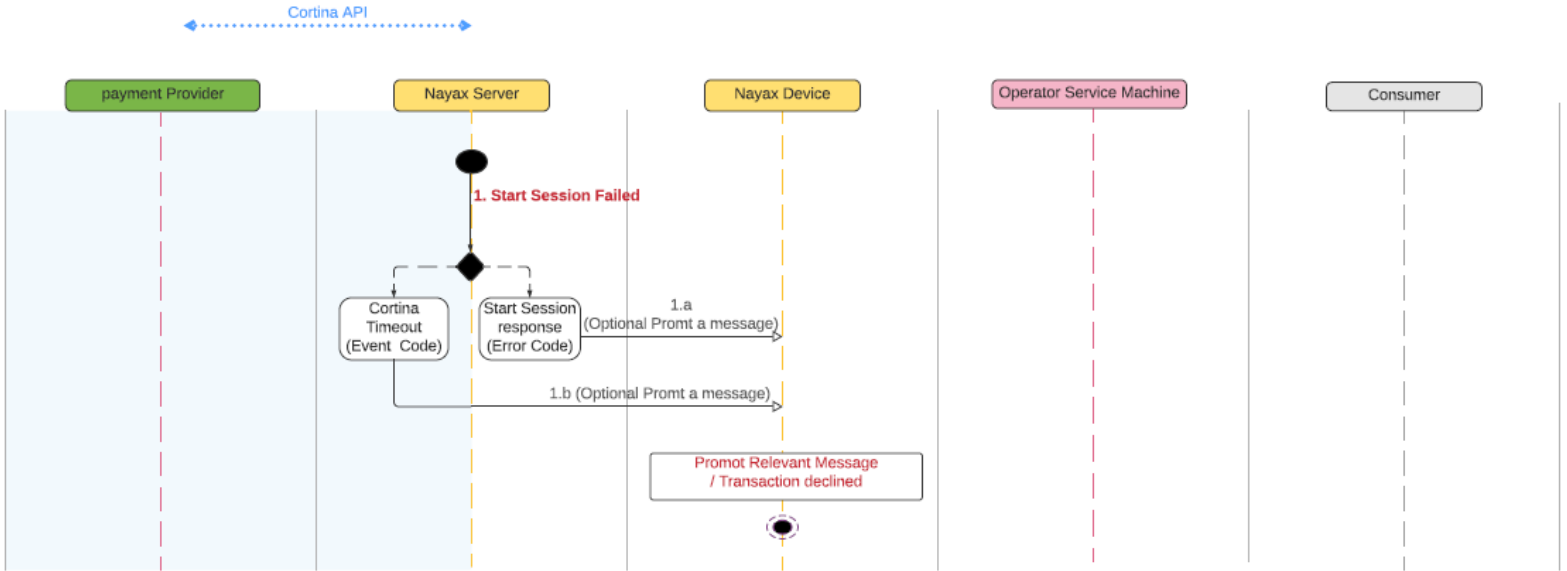
- For activity number: 4.b - Start Session Failed Scenario, please check the following [process](#).
- For activity number: 5.b – Authorization request Failed Scenario, please check the following [process](#).
- For activity number: 8.b – User failed to choose a valid product in time Scenario, please check the following [process](#).
- For activity number: 10.b – Vend Failed Scenario, please check the following [process](#).
- For activity number: 12.b – Settlement response Failed Scenario, please check the following [process](#).

The essence of Cortina external prepaid is:

1. Nayax POS receives external prepaid means of payment from the consumer.
2. Nayax server securing communication with relevant payment gateway.
3. Authorize payment, In preauthorization - receiving approval from the payment gateway to hold a default credit amount for the transaction.
4. Operators' machine provide service accordingly.
5. Settlement payment, In preauthorization after consumer receiving the product, settling the payment on the final amount with the payment provider.

## 6.2 Start Session Failed

Start Session Failed Scenario (4.b)

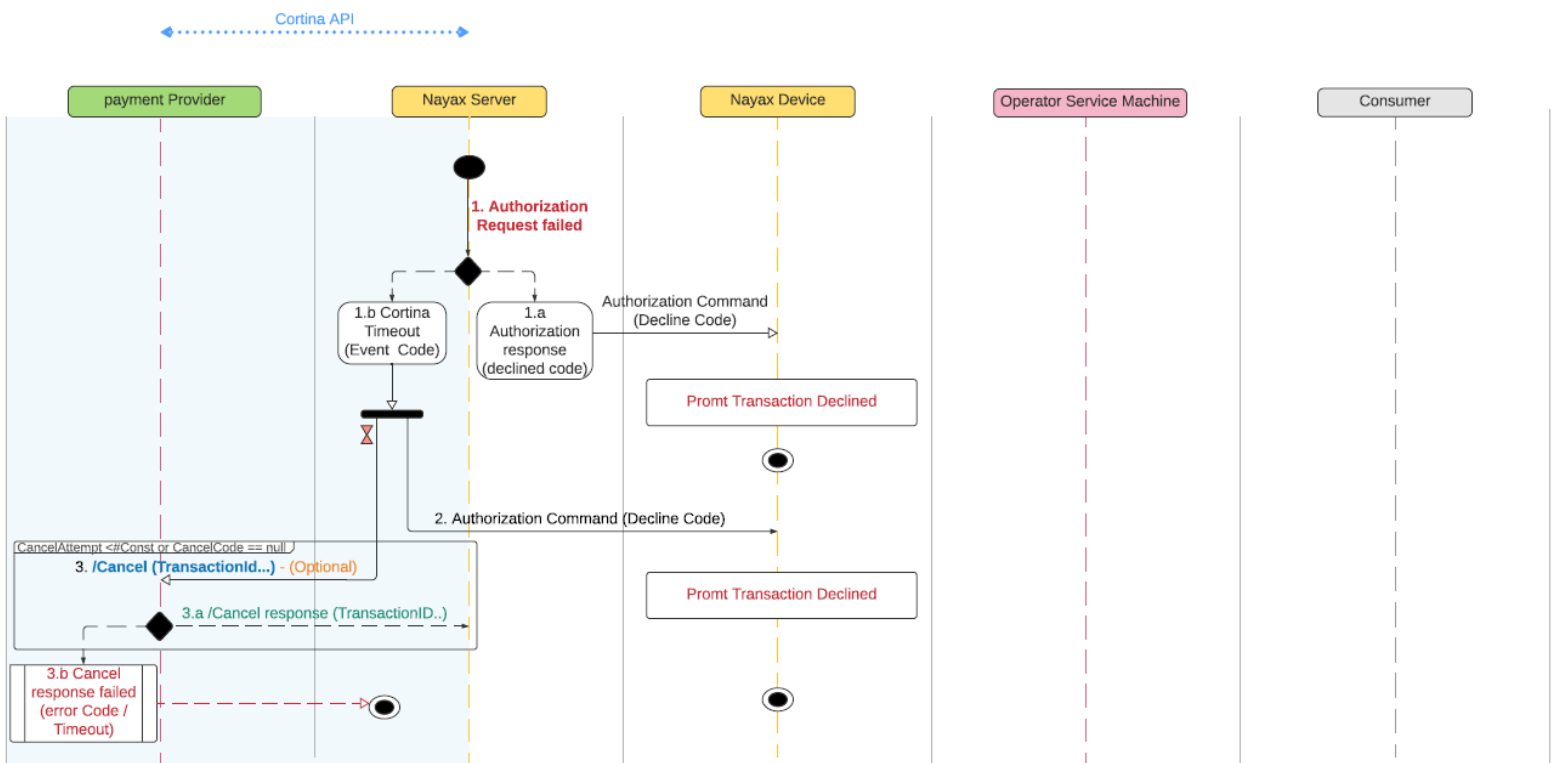


Please note, event "1.a" and event "1.b" are alternatives according to whether the error comes from a Cortina timeout reason or an error response from the integrator or the Operator Server.

Currently prompt "transaction declined".

## 6.3 Authorization Request Failed

Authorization failed Scenario (E.g. 5b)



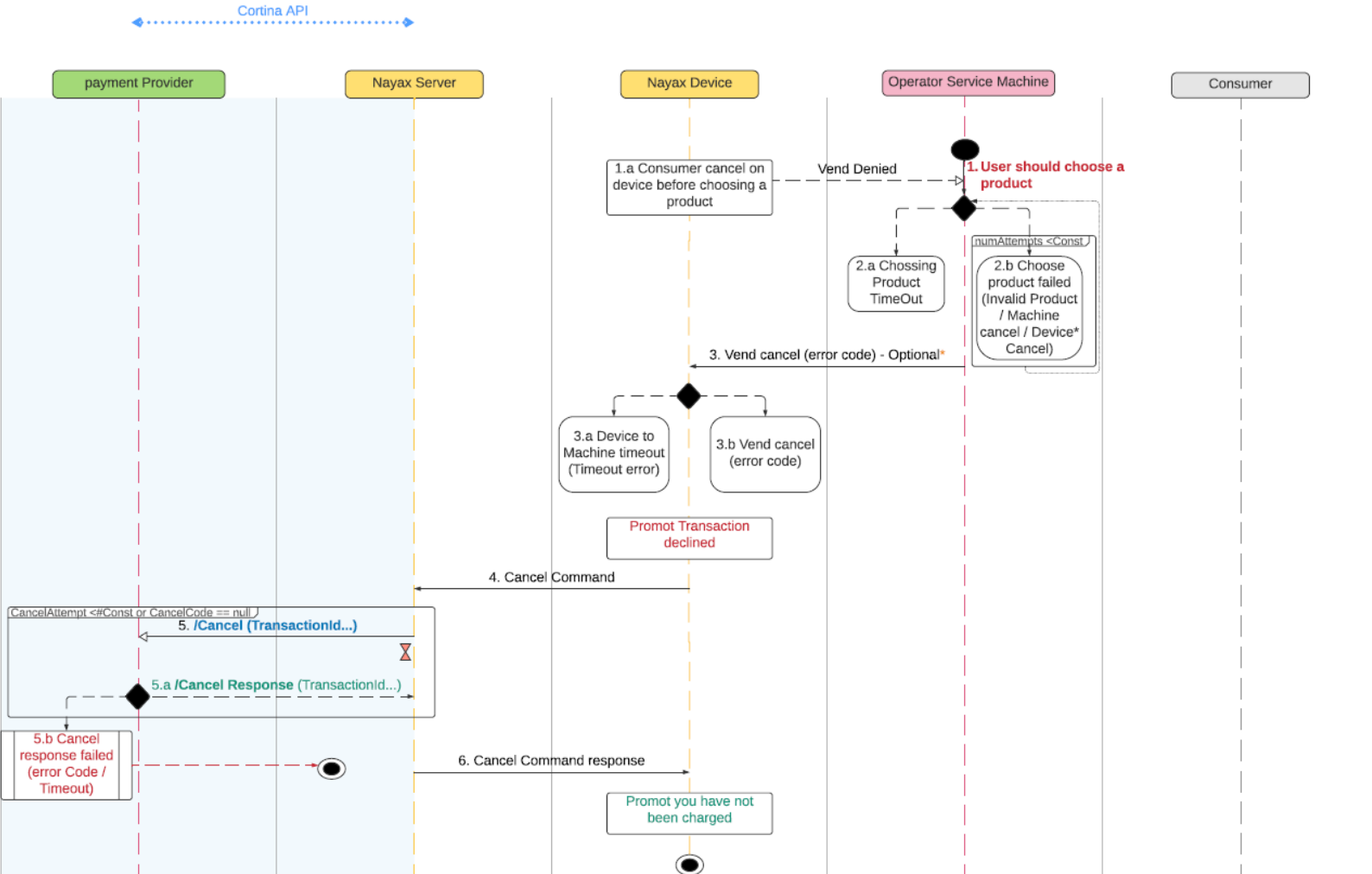
Please note, event "1.a" and event "1.b" are alternatives according to whether the error comes from a Cortina timeout reason or an error response code from the integrator Server.

For activity number: 3.b – Cancel response Failed Scenario, please check the following [process](#).

- Please note, this ability of performing Cancel in case of T/O in Authorization request is called **Immediate Void**, Nayax Server default parameters are: 2 attempts with 30 seconds interval between each call. (For different settings, please contact your Nayax TPOC). Although /Void is optional it is default and highly recommended.

## 6.4 Confused consumer Scenario

Confused customer (E.g. 8b) - Failed to choose a product in a given time or cancel



**Note:**

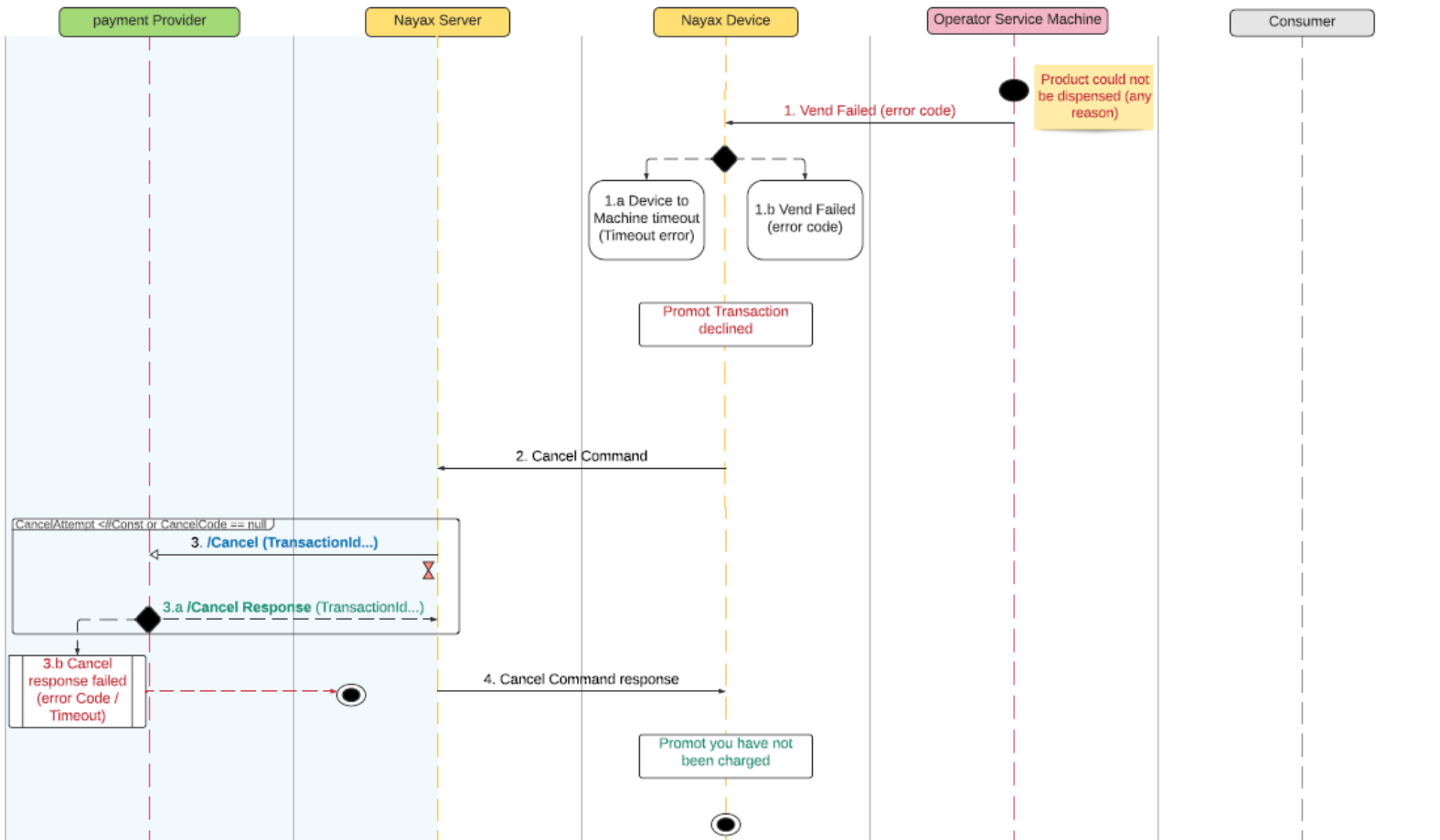
1. Whetehr the machine sends /support vend cancel or not, if a valid product didn't choose within dedicated time will handeled by section 3.a .
2. For activity number: 5.b – Cancel response Failed Scenario, please check the following [process](#).



## 6.5 Preauthorization Vend failed

Preauthorization Vend Fail (E.g. 10b)

← Cortina API →

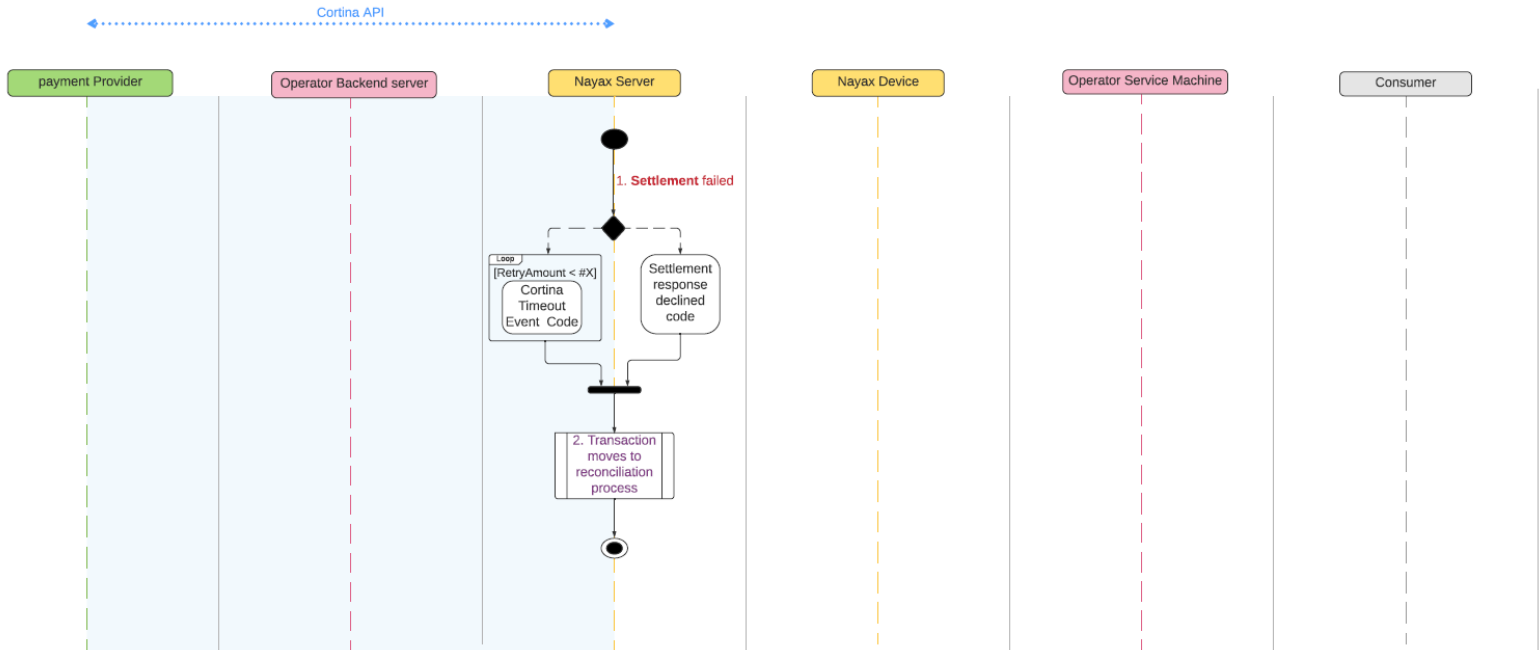


In any case that the Nayax POS is not receiving Vend success in given time, the transaction will fail.

For activity number: 3.b – Cancel response Failed Scenario, please check the following [process](#).

## 6.6 Preauthorization Settlement failed

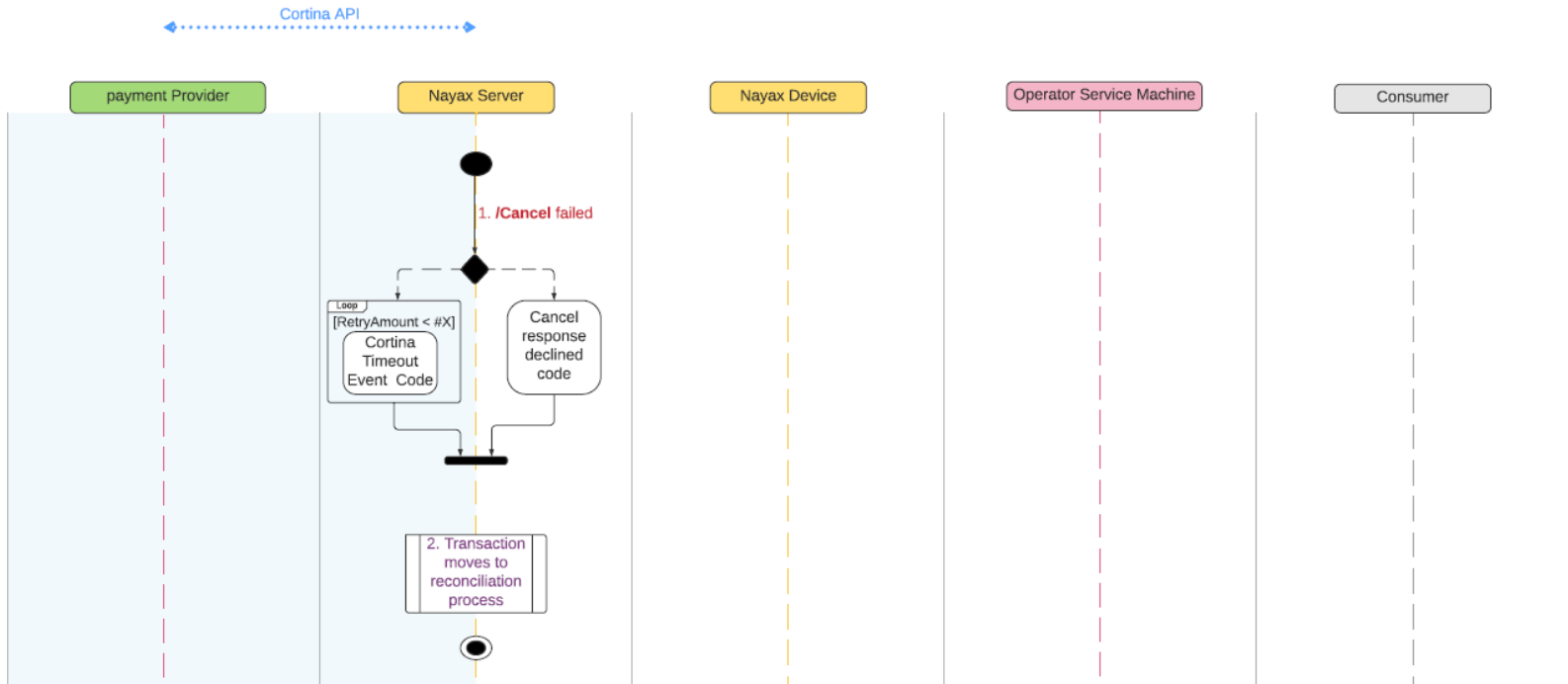
Settlement failed (E.g. 19b)



Please note, Nayax may try to do multiple attempts for settlement before declaring the transaction as lost and sent it into reconciliation process.

## 6.7 Transaction cancel attempt failed

### Preauthorization Cancel Failed

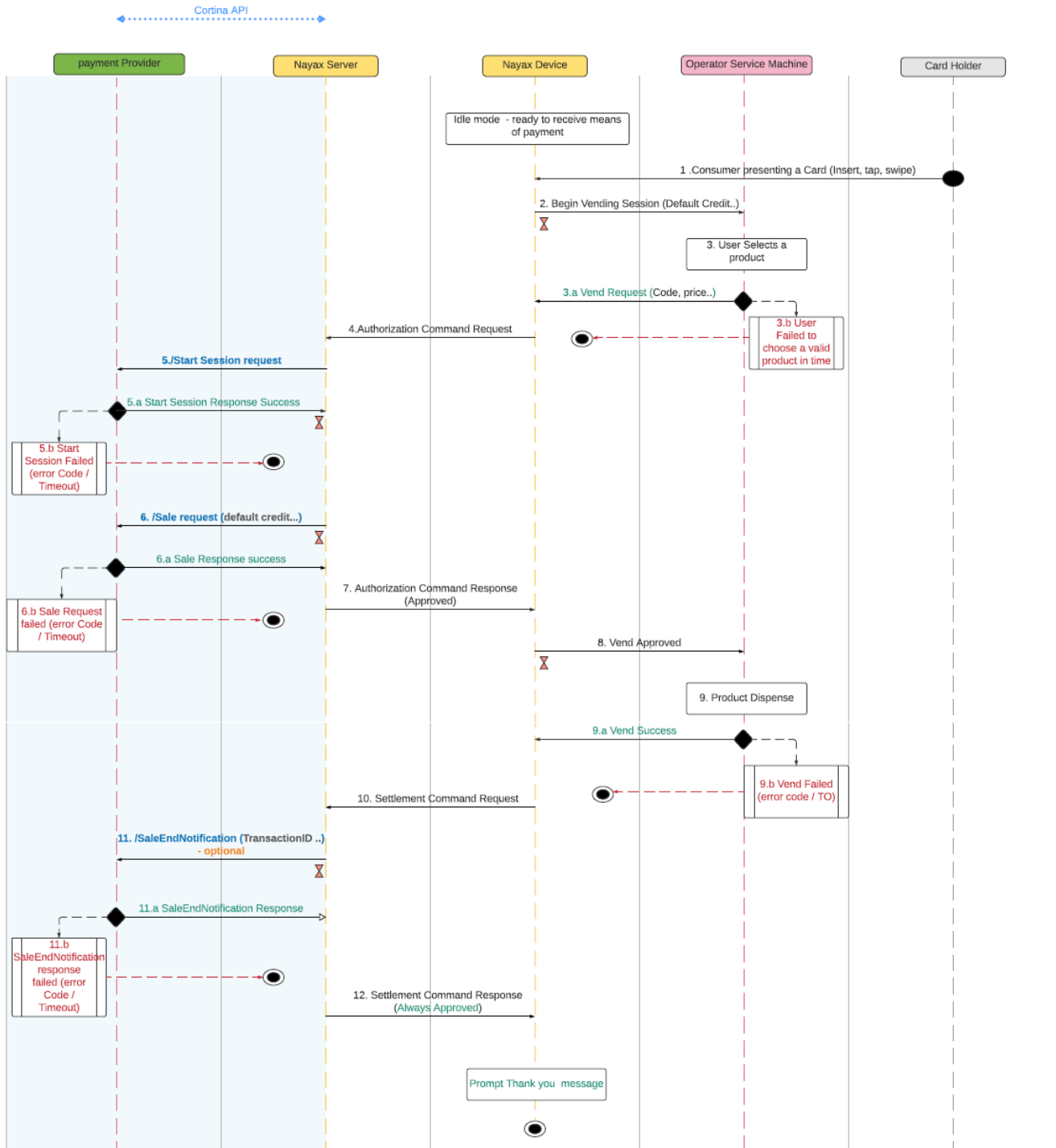


Please note, Nayax may try to do multiple attempts for cancel before sending it into reconciliation process.

# 7. Preselection Transaction Flow:

## 7.1 Positive Preselection Scenario

Cortina External PrePaid - PreSelection Process (Success Scenario)



## Notes:

- For activity number: 3.b – User failed to choose a valid product in time Scenario, please check the following [process](#).
- For activity number: 5.b - Start Session Failed Scenario, please check the following [process](#).
- For activity number: 6.b – Sale request Failed Scenario, please check the following [process](#).
- For activity number: 9.b – Vend Failed Scenario, please check the following [process](#).
- For activity number: 11.b – Sale End Notification Response Failed Scenario, please check the following [process](#).

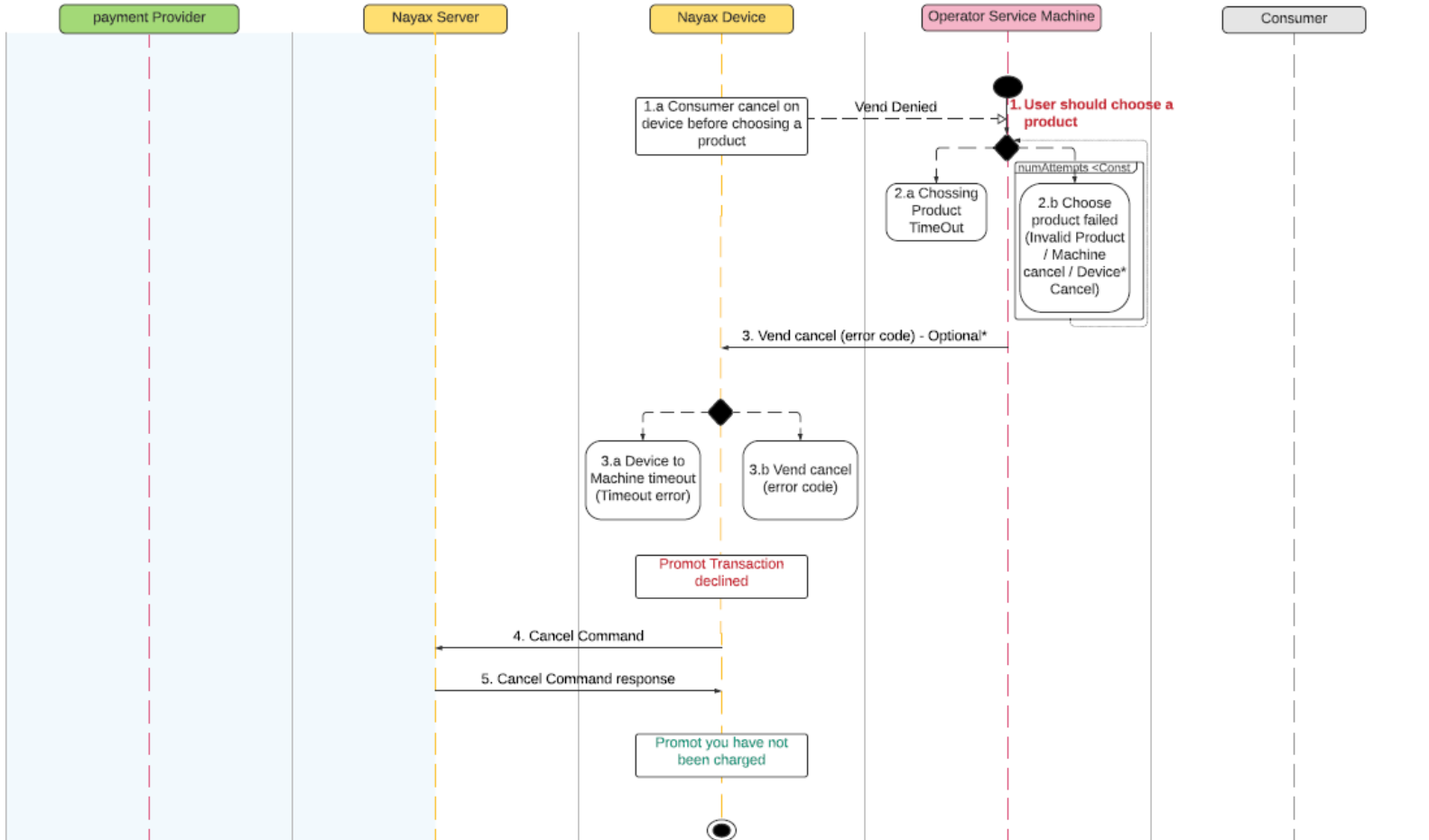
The essence of Cortina external prepaid is:

1. Nayax POS receives external prepaid means of payment from the consumer.
2. Consumer is choosing a product.
3. Nayax server securing communication with relevant payment gateway.
4. Clearing payment - In preselection receiving approval from the payment gateway to clear the final credit amount for the transaction.
5. Operators' machine will provide service accordingly.

## 7.2 Confused customer scenario

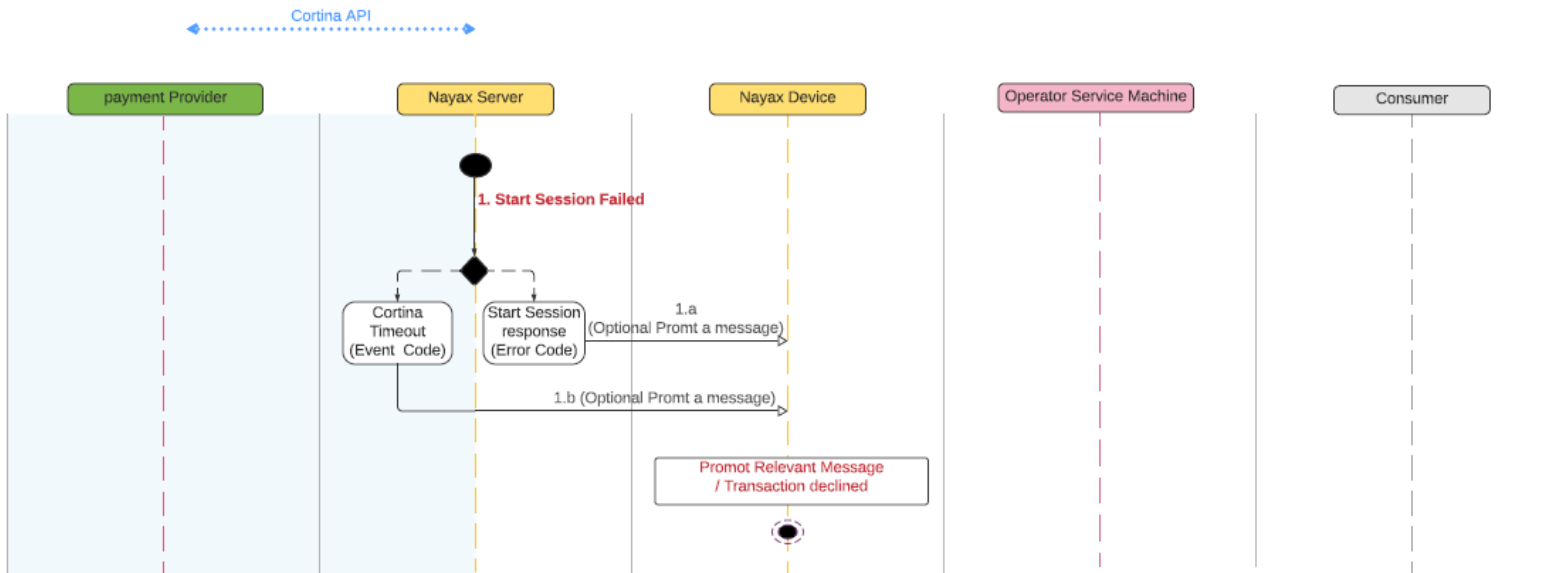
Preselection Confused customer (E.g. 3b) - Failed to choose a product in a given time or cancel

← Cortina API →



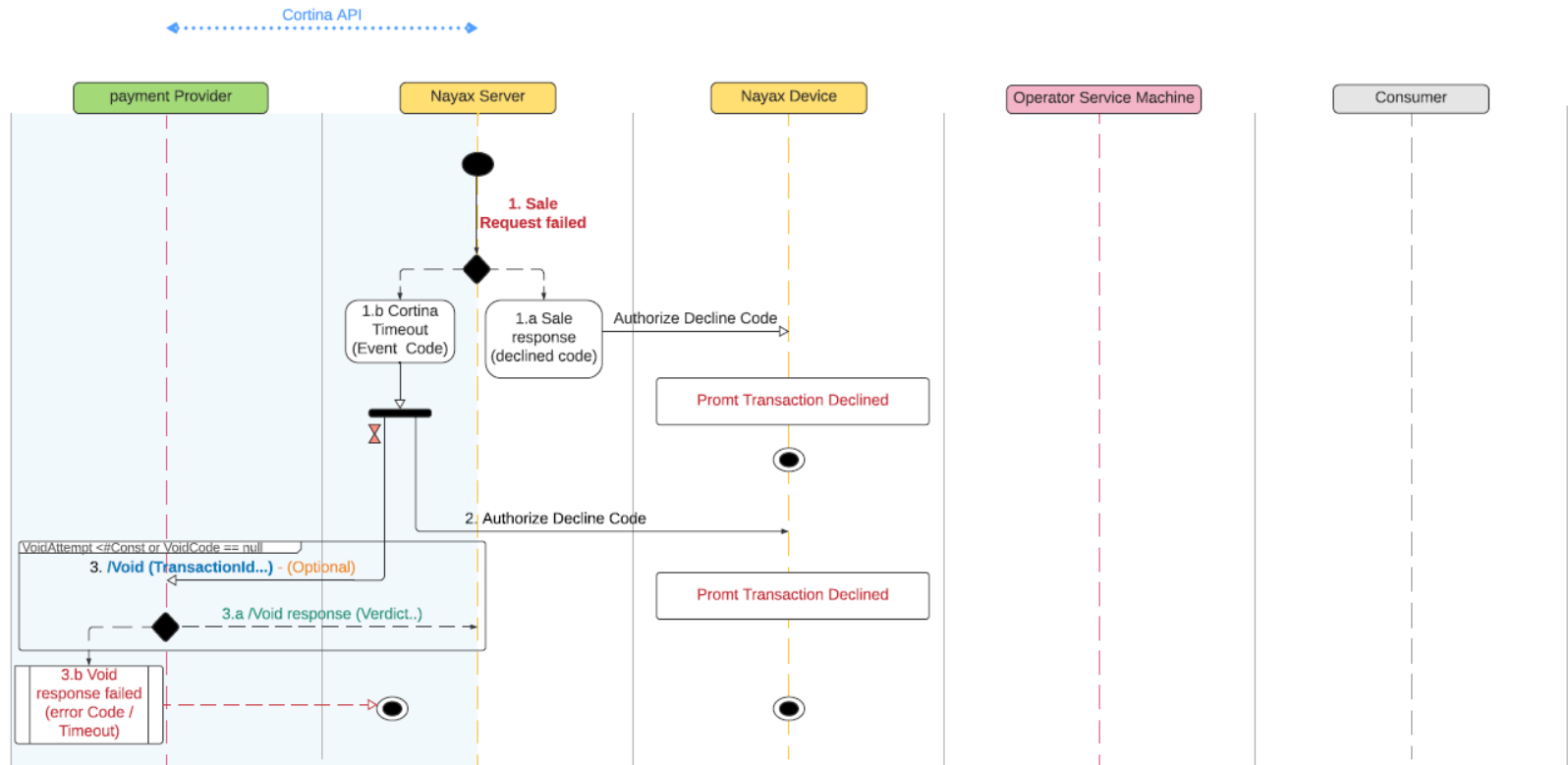
### 7.3 Start Session Failed

Start Session Failed Scenario (5.b)



## 7.4 Sale Request Failed

Sale failed Scenario (E.g. 6b)



Please note, event "1.a" and event "1.b" are alternatives according to whether the error comes from a Cortina timeout reason or an error response code from the integrator Server.

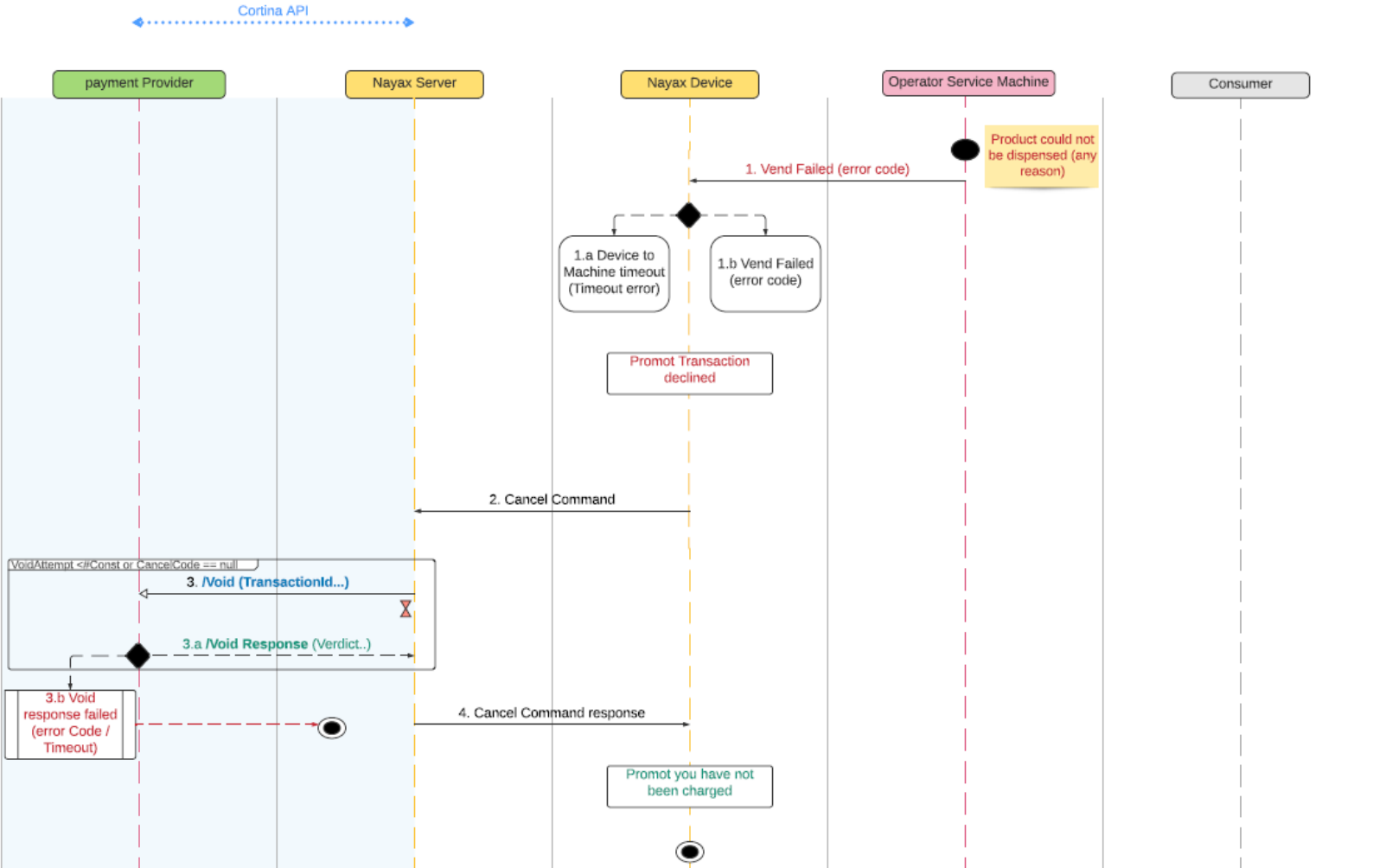
For activity number: 3.b – Void Failed Scenario, please check the following [process](#).

- Please note, this ability of performing Void in case of T/O in Sale request is called **Immediate Void**, Nayax Server default parameters are: 2 attempts with 30 seconds interval between each call. (For different settings, please contact your Nayax TPOC). Although /Void is optional it is default and highly recommended.



## 7.5 Vend failed scenario

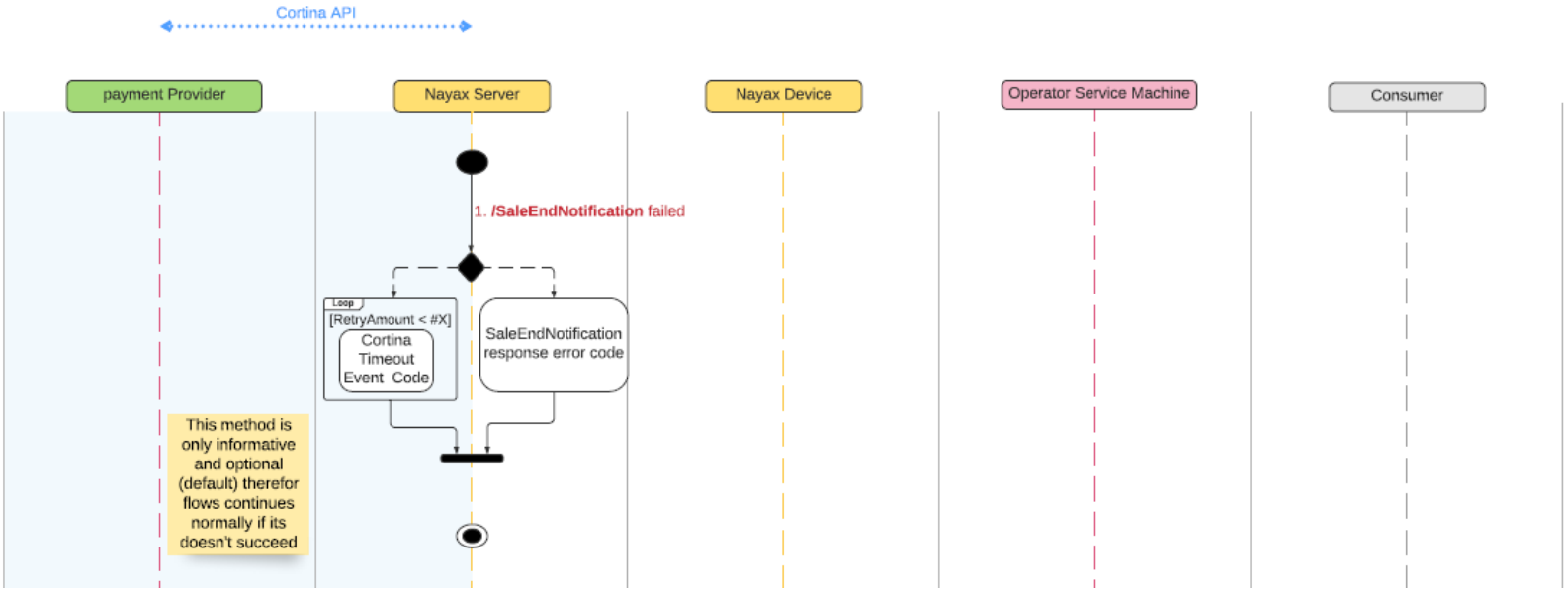
### PreSelection Vend Fail (E.g. 9b)



For activity number: 3.b – Void Failed Scenario, please check the following [process](#).

## 7.6 sale end notification failed scenario

### Preselection Void Failed

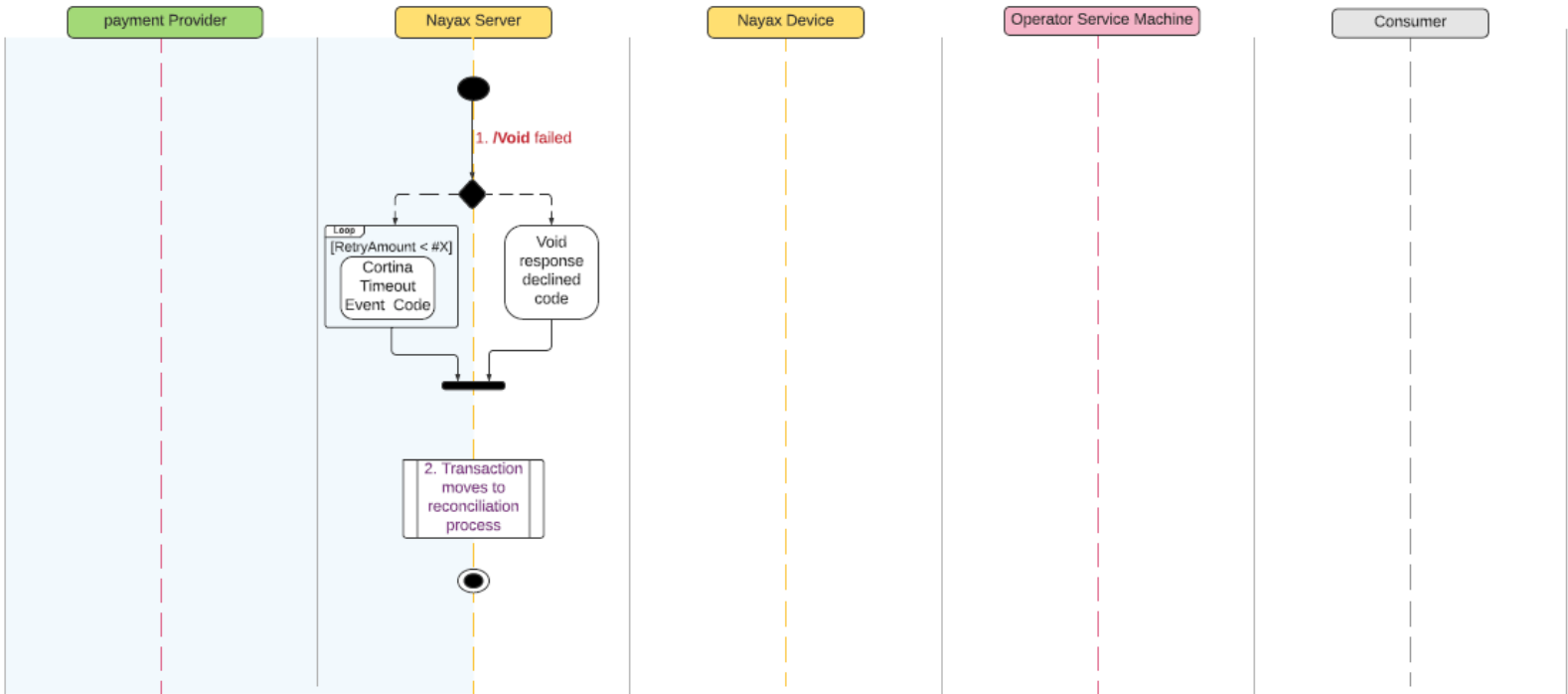


In such case, nothing will happen regarding the transaction as this feature is optional.

## 7.7 Void failed scenario

### Preselection Void Failed

← Cortina API →



Please note, Nayax may try to do multiple attempts for Void before sending it into reconciliation process.

## 8. Supported Methods:

### 8.1 Timeout Handling Mechanism

The timeout handling mechanism for the supported methods is as follows:

- a) Nayax to Integrator Timeout (*#CortinaTimeout*): Nayax uses a default timeout of 15 seconds for its triggered requests to the integrator. This variable is customizable in the setup requirements stage, as an Integer with a range between *#[1000]* to *#[15000]* milliseconds.
- b) In cases of above timeouts, there is an option to immediately revoke transaction with a corresponding **Void\Cancel** call according to relevant flow and predefined settings in the setup requirements section by the integrator.
- c) General Timeouts relations:  
Nayax POS Device timeout > Nayax Server ("Engine") Timeout > Cortina MS Timeout > Integrator Timeout

## 8.2 Methods

### **Keep Alive methods:**

**/Version** - This method is called by Nayax to get the integrator's version details, usually as part of a "Keep Alive" implementation.

### **Start Session methods:**

**/StartSession** – an initial call by Nayax to securely authenticate both Nayax and the 3<sup>rd</sup> party communicating with it regarding a transaction per machine, as described in the relevant appendix of this document. During this process, the integrator also generates a (Numeric) transaction ID for the following calls. If /StartSession is not used the transaction ID shall be generated by Nayax.

### **Transaction Clearing methods:**

**/Authorization** – Used in preauthorization cases (e.g., when a consumer starts a transaction, but the exact clearing amount will be **defined later** according to their uses), this method is called by Nayax and provides the integrator with relevant information to validate and authorize an ongoing transaction by the consumer based on their account. /Authorization request will temporarily hold credit in the consumer account balance as part of the transaction flows. This command must be followed by either settlement or cancel request.

**/Settlement** – Upon successful dispense of a product/service by the operator's machine, Nayax will call this method to complete and capture the transaction with the integrator. This call will include the transaction's final amount to be cleared by the integrator.

**/Sale** – Used in preselection cases (e.g., when a consumer starts a transaction, and the exact clearing amount is **known**), this method is called by Nayax and provides the integrator with relevant information to charge the consumer account balance for a specific amount in a single request.

### **Revoke Transactions:**

**/Cancel** – In events of revoking preauthorization process, this method is called by Nayax and provides the integrator with relevant information to revoke an ongoing instance of preauthorization transaction. Therefore, in such events this method (Cancel) will replace the settlement method and will release the temporarily captured credit in the consumer account. for example, in event such of Vend fail, authorization timeout, consumer manual cancels before choosing a product.

**/Void** – This method is called by Nayax to void a previously successful payment request. For example, in cases such as vend failed, Cortina time out in /sale method, and device failed to produce settlement after a successful /Sale. Although in pre-selection cases /Void may resemble to /Refund method, /Void should always be implemented as it addresses different use cases.

**/Refund** – This method is called by Nayax, within Nayax web system (DCS), to refund full/partial amount of the sale/settlement payment requests or a successful inquiry, for example due to an Application user's complaint.

**Informative:**

**/SaleEndNotification** – Optional method, this method is called by Nayax to inform the integrator about a successful end of sale transaction.

### 9.3 Methods Characterization

#### Detailed Specifications:

Fully detailed Specifications, of the transferred data schemes (Attributes, Types, Applicability, Validation and Description) is available in the attached YAML file, we suggest using [Swagger Editor](#) to read the full characterization descriptions inside the "Scheme" view.

#### Example:

Swagger and YAML for the Cortina v2 external prepared.

## 9.4 Error Code Reasoning Table for supported methods

| Error code | Error description  | Returning Entity |
|------------|--|------------------|
| 1          | "Insufficient funds"   | Integrator       |
| 2          | "Transaction ID unknown"   | Integrator       |
| 3          | "Cardholder exceeded daily / weekly / monthly transaction count limit" | Integrator       |
| 4          | "Cardholder exceeded transaction amount limit"                         | Integrator       |
| 5          | "Suspected Fraud"  | Integrator       |
| 6          | "General system failure"   | Integrator       |
| 7          | "Invalid amount"   | Integrator       |
| 8          | "Request cannot be parsed (format error)"                              | Integrator       |
| 9          | "Transaction not allowed to cardholder"                                | Integrator       |
| 10         | "Missing mandatory parameters"   | Integrator       |
| 11         | "Transaction is already refunded"                                      | Integrator       |
| 12         | "Refund amount cannot be greater than original amount"                 | Integrator       |
| 20         | "No VPN or MQTT enabled"   | Nayax            |
| 30         | "Machine is not responding"  | Nayax            |
| 40         | "Request cannot be parsed (format error)"                              | Nayax            |
| 50         | "Unknown machine Id"   | Nayax            |
| 60         | "Missing mandatory parameters"   | Nayax            |

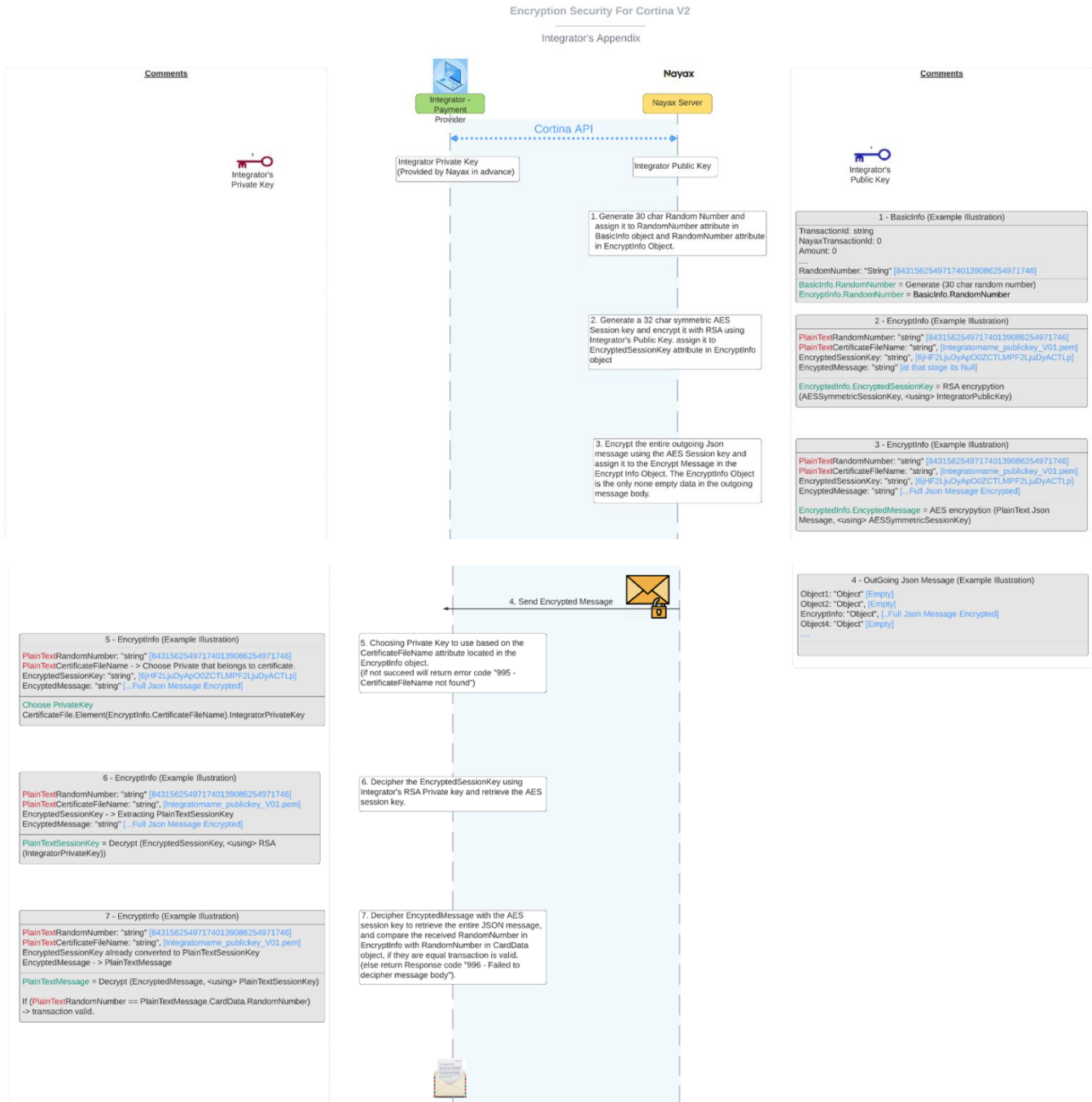


## 9. Appendix

### 9.1 Nayax's IP addresses for whitelist:

- 82.80.58.78
- 37.142.122.78
- 212.179.76.194
- 62.219.16.7
- 62.219.16.8
- 82.80.44.34
- 212.179.76.198
- 82.102.172.206
- 72.28.118.100
- 213.57.117.42
- 137.221.37.116
- 77.247.177.10
- 77.247.183.162
- 77.247.183.163
- 185.159.232.2
- 185.159.233.2
- 185.159.234.2
- 185.159.235.2
- 52.49.25.101
- 52.198.70.228

## 9.2 Cortina Security Encryption



### 9.3 /StartSession Authentication & Example

The /StartSession authentication flow is detailed below with examples values

- Nayax and the integrator have pre-shared the Secret Token (will usually be of 66 chars).  
For example: **mrV3U3nsgGFrE3w5-wnBo\_WCLPce-pZ1awRvTVTkungMIKThTVbj\_fiXdfogclhn0**
- Nayax sends /Start Session Request with the Token ID to be used (e.g. 123456) and a 27 char Random String.  
For example: **123456789qwertyuioasdfghjkl**
- The Integrator generates **Transaction ID** of 36 Numeric Characters  
For example:  
**123456789012345678901234567890123456**
- The integrator prepares a Ciphertext by appending Transaction ID and the received Random string with "=" char separator - to create a 64 char long ciphertext.  
In our example:  
**123456789012345678901234567890123456=123456789qwertyuioasdfghjkl**
- The Integrator shall produce an encryption key from the the last 32 chars of the token (a 256 bit key).  
In our example: **wRvTVTkungMIKThTVbj\_fiXdfogclhn0**
- The Integrator shall encrypt the ciphertext using AES in ECB mode and will send the produced cipher in the /StartSession response  
  
In our example it would produce the following cipher:  
**a0Qnxm4fWMSkzFXiMivn8BDiQVSL6be/NXIICC9HBoAiry6DUdKYPQh/YS1G8nObE6/0o9N4MFuYA7CTAxAnphuNJwBEjgBzKhhgpJ5ggnw=**
- Nayax Decrypts the Cipher using the same key and validates that random number is as sent in the request and that the format is as expected.
- Nayax will then use the deciphered Transaction ID in the following commands like **/Sale, /Authorization, /Void** and Others.
- When receiving these requests, the Integrator shall validate that the Transaction Id was indeed produced by itself in an earlier Start-Session request and is still valid. It is Highly recommended that transactions ID remain Valid for a limited time no longer than 10 minutes for **/Sale** and **/Authorization** requests. However **/Settlement** , **/Void**, **/Cancel** and **/Refund** requests should be

processed regardless of the Transaction ID's validity as they may be called after their preceding **/Authorization** and **/Sale** requests.

## 10. Examples (Yaml V2R01)

### 10.1 /StartSession Example:

Request

```
{ "TokenId": "0",  
  "RandomNumber": "D1UQI1IJ8LT7C2PW67IZALMG472" }
```

Response

```
{ "TranIDCipher":  
  "CGPKw+96Cz5LwlQuuWK4E7SRsVUc5z2FsIRDpW0xMLLeU2FYfU8SXvYZWrgM/I9xAEYbUXCk  
  m+bjd+xsOXsmjC7rG4JUfxJWwaiTs/DgGrU=",  
  "Status": {  
    "Verdict": "Approved",  
    "Code": 0,  
    "StatusMessage": "Cortina V2 Stub Tester"}  
}
```

### 10.2 /Authorization Example:

Request:

```
{  
  "BasicInfo": {  
    "TransactionId": "236584335969834429321847829253667359",  
    "Amount": 0.1,  
    "CurrencyCode": "USD",  
    "CurrencyNumeric": 376,  
    "SiteId": 12  
  },  
  "MachineInfo": {  
    "Id": 869761713,  
    "Name": "TestPayStaticQR V2",  
    "TerminalId": 987654321,  
    "DecimalPlace": 2,  
  }  
}
```

```
"Offset": 2,  
"GroupId": "Beverages",  
"OperatorId": 12345,  
"ZipCode": 10001,  
"Country": {}  
},  
"ActorInfo": {  
"Id": 2001102296,  
"Name": "Far East Dev Zone",  
"OperatorId": 2000305191,  
"OperatorName": "TanyaT_OP",  
"MerchantId": 10011  
},  
"CustomData": {  
"DirectActor": 12312312,  
"Operator": 123121434134,  
"Distributor": "superDistributor",  
"Actor": 12312312,  
"Machine": 131231348  
}  
}
```

Response:

```
{  
"Status": {
```

```
"Verdict": "string",  
"Code": 0,  
"StatusMessage": "string"  
},  
"PaymentInfo": {  
"AuthCode": "333333",  
"AuthAmount": 0.1,  
"SettAmount": 0.1,  
"Token": "Cortina V2 Stub Tester Token",  
"AuthDateTime": "230221101008",  
"SettDateTime": "230221101009",  
"TraceNumber": "4444",  
"AuthSource": "555555",  
"AdditionalData": ""  
}  
}
```

### 10.3 /Settlement Example:

Request:

```
{  
"BasicInfo": {  
"TransactionId": "236584335969834429321847829253667359",  
"Amount": 0.1,  
"CurrencyCode": "USD",  
"CurrencyNumeric": 376,
```

```
"SiteId": 12
},
"MachineInfo": {
  "Id": 869761713,
  "Name": "TestPayStaticQR V2",
  "TerminalId": 987654321,
  "DecimalPlace": 2,
  "Offset": 2,
  "GroupId": "Beverages",
  "OperatorId": 12345,
  "ZipCode": 10001,
  "Country": {}
},
"ActorInfo": {
  "Id": 2001102296,
  "Name": "Far East Dev Zone",
  "OperatorId": 2000305191,
  "OperatorName": "TanyaT_OP",
  "MerchantId": 10011
},
"CustomData": {
  "DirectActor": 12312312,
  "Operator": 123121434134,
  "Distributor": "superDistributor",
  "Actor": 12312312,
```



```
"Machine": 131231348
},
"Products": [
{}
],
"PaymentInfo": {
"AuthCode": "333333",
"AuthAmount": 0.1,
"SettAmount": 0.1,
"Token": "Cortina V2 Stub Tester Token",
"AuthDateTime": "230221101008",
"SettDateTime": "230221101009",
"TraceNumber": "4444",
"AuthSource": "555555",
"AdditionalData": ""
}
}
```

Response:

```
{
"Status": {
"Verdict": "Declined",
"Code": 10,
"StatusMessage": "Ammount is missing"
}
```

```
}
```

#### 10.4 /Sale Example:

Request:

```
{  
  "ReasonCode": 0,  
  "ReasonText": "string",  
  "BasicInfo": {  
    "TransactionId": "236584335969834429321847829253667359",  
    "Amount": 0.1,  
    "CurrencyCode": "USD",  
    "CurrencyNumeric": 376,  
    "SiteId": 12  
  },  
  "MachineInfo": {  
    "Id": 869761713,  
    "Name": "TestPayStaticQR V2",  
    "TerminalId": 987654321,  
    "DecimalPlace": 2,  
    "Offset": 2,  
    "GroupId": "Beverages",  
    "OperatorId": 12345,  
    "ZipCode": 10001,  
    "Country": {}  
  },  
}
```

```
"ActorInfo": {  
  "Id": 2001102296,  
  "Name": "Far East Dev Zone",  
  "OperatorId": 2000305191,  
  "OperatorName": "TanyaT_OP",  
  "MerchantId": 10011  
},  
"CustomData": {  
  "DirectActor": 12312312,  
  "Operator": 123121434134,  
  "Distributor": "superDistributor",  
  "Actor": 12312312,  
  "Machine": 131231348  
},  
"Products": [  
  {}  
]  
}
```

Response:

```
{  
  "Status": {  
    "Verdict": "Approved",  
    "StatusMessage": "transaction approved"  
  }  
}
```

```
},  
"PaymentInfo": {  
  "AuthCode": "333333",  
  "AuthAmount": 0.1,  
  "SettAmount": 0.1,  
  "Token": "Cortina V2 Stub Tester Token",  
  "AuthDateTime": "230221101008",  
  "SettDateTime": "230221101009",  
  "TraceNumber": "4444",  
  "AuthSource": "555555",  
  "AdditionalData": ""  
}  
}
```

### 10.5 /Cancel Example:

Request:

```
{  
  "ReasonCode": 2,  
  "ReasonText": "Cashless cancelled by consumer",  
  "BasicInfo": {  
    "TransactionId": "236584335969834429321847829253667359",  
    "Amount": 0.1,  
    "CurrencyCode": "USD",  
    "CurrencyNumeric": 376,  
    "SiteId": 12
```

```
},  
"MachineInfo": {  
  "Id": 869761713,  
  "Name": "TestPayStaticQR V2",  
  "TerminalId": 987654321,  
  "DecimalPlace": 2,  
  "Offset": 2,  
  "GroupId": "Beverages",  
  "OperatorId": 12345,  
  "ZipCode": 10001,  
  "Country": {}  
},  
"ActorInfo": {  
  "Id": 2001102296,  
  "Name": "Far East Dev Zone",  
  "OperatorId": 2000305191,  
  "OperatorName": "TanyaT_OP",  
  "MerchantId": 10011  
},  
"CustomData": {  
  "DirectActor": 12312312,  
  "Operator": 123121434134,  
  "Distributor": "superDistributor",  
  "Actor": 12312312,  
  "Machine": 131231348
```

```
},  
"PaymentInfo": {  
  "AuthCode": "333333",  
  "AuthAmount": 0.1,  
  "SettAmount": 0.1,  
  "Token": "Cortina V2 Stub Tester Token",  
  "AuthDateTime": "230221101008",  
  "SettDateTime": "230221101009",  
  "TraceNumber": "4444",  
  "AuthSource": "555555",  
  "AdditionalData": ""  
}  
}
```

Response:

```
{  
  "Status": {  
    "Verdict": "Approved",  
    "StatusMessage": "Cortina V2 Stub Tester"  
  }  
}
```

## 10.6 /Void Example:

```
Request  
{  
  "ReasonCode": 3,
```

```
"ReasonText": "Cashless Cancelled by machine",
"BasicInfo": {
  "TransactionId": "236584335969834429321847829253667359",
  "Amount": 0.1,
  "Currency": "ILS",
  "SiteId": 1,
  "MachineAuTime": "230221121009" },
"DeviceInfo": {
  "Id": 0,
  "HwSerial": "0434324119376524",
  "FWVersion": "" },
"MachineInfo": {
  "Id": 869761713,
  "Name": "TestPayDynamicQR V2",
  "TerminalId": "987654321",
  "DecimalPlace": 2,
  "Offset": "2.00",
  "CountryCode": "US",
  "CountryISOCode": "USA",
  "Country": "UNITED STATES",
  "City": "Abbeville",
  "Address": "TestPayDynamicQR V2",
  "ZipCode": "1223311" },
"ActorInfo": {
  "Id": 2001187922,
  "Name": "TanyaT_OP",
  "MerchantId": "123456789" },
"CustomData": {
  "Actor": "",
  "Machine": "{\"dynamic_qr_channels_rec_ids\": \"18,19\"}"
},
"PaymentInfo": {
  "SrvTranId": "236584335969834429321847829253667359",
  "AuthCode": "333333",
```

```
"AuthAmount": 0.1,  
"SettAmount": 0.1,  
"RRN": "111111",  
"Token": "",  
"AuthDateTime": "230221101008",  
"SettDateTime": "230221101009",  
"TraceNumber": "4444",  
"AuthSource": "55555",  
"IsGatewayTimeout": false }  
}
```

Response

```
{  
  "Status": {  
    "Verdict": "Approved",  
    "Code": 0,  
    "StatusMessage": "Cortina V2 Stub Tester"  
  }  
}
```

## 10.7 /Refund Example:

Request:

```
{  
  "ReasonText": "string",  
  "BasicInfo": {  
    "TransactionId": "236584335969834429321847829253667359",  
    "Amount": 0.1,  
    "CurrencyCode": "USD",  
    "CurrencyNumeric": 376,  
    "SiteId": 12
```



```
},  
"MachineInfo": {  
  "Id": 869761713,  
  "Name": "TestPayStaticQR V2",  
  "TerminalId": 987654321,  
  "DecimalPlace": 2,  
  "Offset": 2,  
  "GroupId": "Beverages",  
  "OperatorId": 12345,  
  "ZipCode": 10001,  
  "Country": {}  
},  
"ActorInfo": {  
  "Id": 2001102296,  
  "Name": "Far East Dev Zone",  
  "OperatorId": 2000305191,  
  "OperatorName": "TanyaT_OP",  
  "MerchantId": 10011  
},  
"CustomData": {  
  "DirectActor": 12312312,  
  "Operator": 123121434134,  
  "Distributor": "superDistributor",  
  "Actor": 12312312,  
  "Machine": 131231348
```

```
},  
"PaymentInfo": {  
  "AuthCode": "333333",  
  "AuthAmount": 0.1,  
  "SettAmount": 0.1,  
  "Token": "Cortina V2 Stub Tester Token",  
  "AuthDateTime": "230221101008",  
  "SettDateTime": "230221101009",  
  "TraceNumber": "4444",  
  "AuthSource": "555555",  
  "AdditionalData": ""  
}  
}
```

Response:

```
{  
  "Status": {  
    "Verdict": "Approved",  
    "StatusMessage": "Cortina V2 Stub Tester"  
  }  
}
```

## 10.7 /SaleEndNotification Example:

**Request:**

```
{
```

```
"BasicInfo": {  
  "TransactionId": "string",  
  "NayaxTransactionId": 0,  
  "Amount": 0,  
  "CurrencyCode": "string",  
  "CurrencyNumeric": "string",  
  "SiteId": 0,  
  "MachineAuTime": "string",  
  "TimeoutMS": 0,  
  "IsProductSelected": true  
},  
"DynamicURL": "string"  
}
```

**Response:**

```
{  
  "Status": {  
    "Verdict": "string",  
    "Code": 0,  
    "StatusMessage": "string"  
  }  
}
```