1 INTRODUCTION

1.1 WHAT IS PREAUTHORIZATION OF FUNDS?

When a merchant is paid through one of its e-payment channels, a process is initiated to verify and guarantee the debtor has sufficient funds. This process entails ‘holding’ or ‘reserving’ funds on the debtor’s account (be it the credit limit in case of a credit card payment, or the available bank balance in case of a debit card). The funds are “held” and deducted from the customer’s credit limit (or available bank balance, in the case of a debit card), but are not yet transferred to the merchant. At the time of the merchant’s choosing, the settlement process is initiated, where the funds are actually transferred from the customers’ account to the merchant's account.

1.2 WHY IS THIS NEEDED?

The main reason for preauthorization of funds is because of a two-step process in the payment, consisting of an authorization and a settlement with a time lag in between.

1.3 THE POWER OF MASH UPS USING ISO 20022

This pre-authorisation API contract is using ISO 20022 components and elements. Other open banking APIs that are also using ISO 20022 components (such as PayLater) can be easily used to create mash ups and more easily integrate with bank’s back office systems that are also based on ISO 20022.
2 PAYMENT PRE-AUTHORISATION USE CASES

2.1 LARGE AMOUNT WEB PURCHASE (E.G. HOLIDAY BOOKING)

Payment Pre-Authorisation Activities - Use Case 1
Known Future Date and Amount without blocking

Customer PSU

- Selects purchase (e.g. holidays)
- Confirms purchase and requests pre-authorisation

Merchant

- Receives request from Customer (future date and amount are known)
- Initiates pre-authorisation to Bank

Bank ASPSP

- Records obligation to pay for customer
- Returns confirmation of pre-authorisation

- Records pre-authorisation confirmation
- Validates purchase with guaranteed payment
- Sends validation

- Receives confirmation
2.2 TEMPORARY BLOCKING OF FIXED AMOUNT (E.G. GAS STATION)

Payment Pre-Authorisation Activities - Use Case 2
Gas Station scenario
Max Amount is blocked until finalisation of transaction

Customer
Selects purchase (e.g. request to fill up tank)
Confirms purchase and requests pre-authorization

Receives request from Customer (max amount authorised and blocked)
Initiates pre-authorization to Bank

Records obligation to pay for customer
Returns confirmation with max amount

Records pre-authorization confirmation
Validates purchase with guaranteed payment
Sends validation

Records end of purchase (e.g. transaction is terminated)
Initiates payment with final amount (or max amount when reached) based on payment pre-authorization

Processes payment
Unblocks the pre-authorization
Confirms payment initiation

Processes confirmation of payment
Informs customer

Receives confirmation of initiated payment
2.3 WEB PURCHASE: ASPSP GUARANTEES AMOUNT TO MERCHANT

Online Shop - Pre-authorization with individual payments when line items are shipped

Payment Pre-authorization Activities - Use Case 3

Customer

- Selects multiple items in shopping cart (e.g., basket of items)
- Checks out
- Requests pre-authorization

Merchant

- Receives request from Customer (max amount authorized for total amount of line items)
- Initiates pre-authorization to Bank

Bank - ASPSP

- Records obligation to pay for customer
- Returns confirmation with pre-authorized max amount

Merchant

- Records pre-authorization confirmation
- Validates purchase with guaranteed payment
- Prepares shipment of line items
- Initiates payment with line item amount (until max amount based on payment pre-authorization (Partial or Full))
- Records payment for line item
- Ships line item
- Receives confirmation of shipment

---

If not reached, Misc steps reached:

- Rejects payment
- Rejection Process Not covered here

Merchant - Customer

- Update shipped line items list
- Sends invoice to Customer with all shipped line items.

---

Receive invoice
3 PAYMENT PRE-AUTHORISATION FLOWS
4 PAYMENT PRE-AUTHORISATION DATA-MODEL PROPOSAL

Payment Pre-Authorisation Resource Data Model

- PaymentPreAuthorisation (1..1): Max35Text
  - Status (1..1): StatusCode (Approved/Rejected/Pending)
  - MaximumAmount (0..1): ActiveCurrencyAndAmount
  - Type (0..1): AuthorisationTypeCode (OneOff/Recurrent/CreditLine)
  - Purpose (0..1): Max35Text
  - AllowedMerchants (0..n): PartyIdentification
  - ValidFrom (0..1): ISODatetime or ISODatetime
  - ValidUntil (0..1): ISODatetime or ISODatetime
  - RecurringPaymentData (0..1): (NbOfPayments, FirstDate, LastDate, Frequency)

- ChargesRecord (1..1): UUDv4Identifier
  - Amount (1..1): ActiveCurrencyAndAmount
  - Type (0..1): Max35Text
  - Rate (0..1): PercentageRate
  - Tax (0..1): ChargesTax (Amount, Type and Rate)

- PreAuthorisedPaymentInitiation (0..n)
  - PaymentInitiationId (1..1): UUDv4Identifier
  - PaymentPreAuthorisationId (0..1): UUDv4Identifier
  - PaymentIdentification (1..1): (InstructedId, EndToEndId, UETR)
  - PaymentType (1..1): Full or Partial or Final
  - RequestedExecutionDate (0..1): ISODatetime or ISODatetime
  - AcceptanceDate (0..1): ISODatetime
  - InstructedAmount (1..1): ActiveCurrencyAndAmount
  - UltimateDebtor (0..1): Name
  - Debtor (1..1): PartyIdentification
  - DebtorAccount (1..1): CashAccountIdentification
  - CreditAgent (1..1): FinancialInstitutionIdentification
  - Creditor (1..1): PartyIdentification
  - CreditorAccount (1..1): CashAccountIdentification
  - UltimateCreditor (0..1): Name
  - RequestedExecutionDate (1..1): ISODatetime
  - Purpose (0..1): Max35Text
  - UnstructuredRemittance (0..1): Max140Text

- LineItem (0..n)
  - ItemAmount (1..1): ActiveCurrencyAndAmount
  - Type (0..1): Max35Text
  - Description (0..1): Max35Text