

nexo specification - Part 2

nexo FAST

JUNE 2021

DAVID HOARAU - DAVID.HOARAU@GALITT.COM

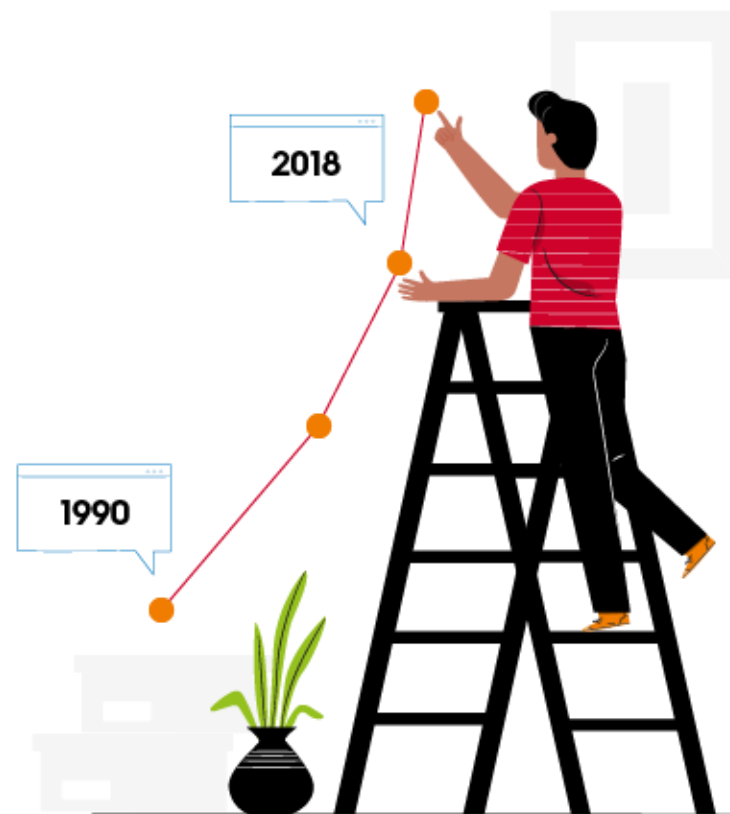
REF: 2- NEXO FAST – V1.03.PPTX

Passion for payments



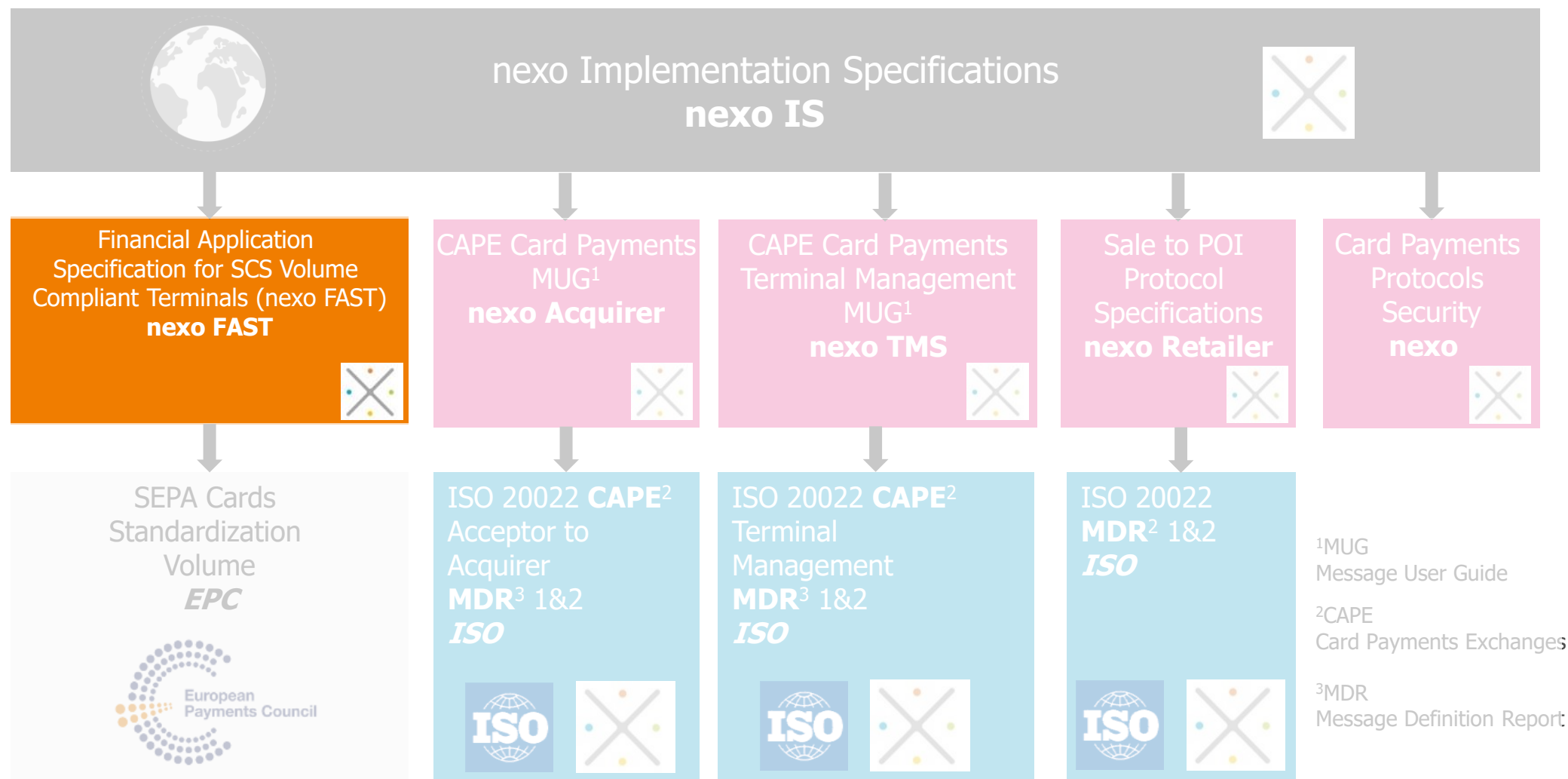
Summary

- 01 **Business considerations**
- 02 **nexo FAST architecture**
- 03 **In scope**
- 04 **nexo FAST application structure**
- 05 **Data model**
- 06 **Interfaces to the host and to the sale system**
- 07 **Out of scope**



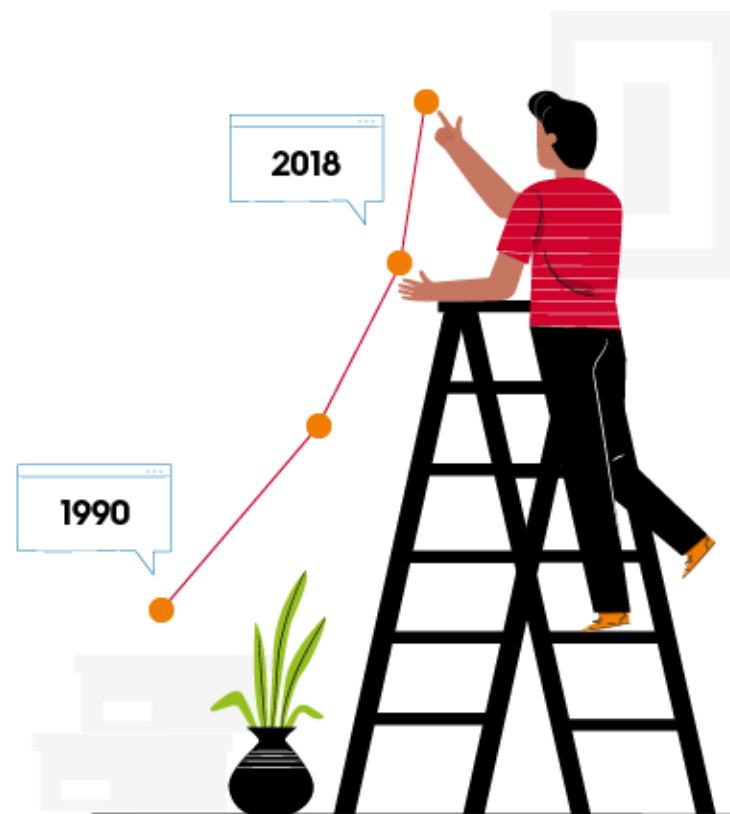
nexo FAST

Reference documents



Summary

- 01 **Business considerations**
- 02 **nexo FAST architecture**
- 03 **In scope**
- 04 **nexo FAST application structure**
- 05 **Data model**
- 06 **Interfaces to the host and to the sale system**
- 07 **Not described by nexo FAST**



Business Considerations



— **Financial Application Specification for Terminals compliant to SEPA Card Framework**

- └ Harmonising EMV implementations on terminals including those which have to be SEPA compliant
- └ Reducing the risks of interoperability obstacles between applications
- └ Supporting a uniform "look and feel" for transactions from the cardholder's perspective
- └ Processing all Schemes within the same application



— **Scope**

- └ Multi-Services (Payment, Cashback, Deferred Payment, Refund, Cancellation...)
- └ Multi-Schemes (Visa, MasterCard, American Express, CB...)
- └ Multiple technologies (Chip, Magstripe, ContactLess, Manual Entry)
- └ Cardholder Verification Method (Online PIN, Offline PIN, Paper Signature...)
- └ Card Authentication (SDA, DDA, CDA)

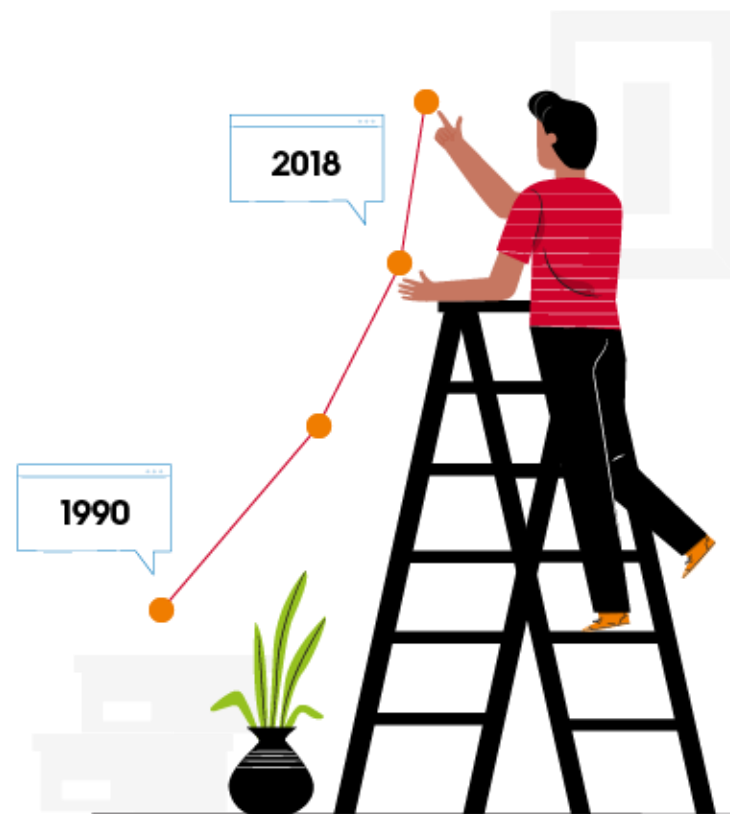


— **Unique application capable of communicating with one or more acquirers for all schemes**

- └ Application behavior according to the application profiles
- └ Application Profile = Parameter set associated with a card application and an acquirer

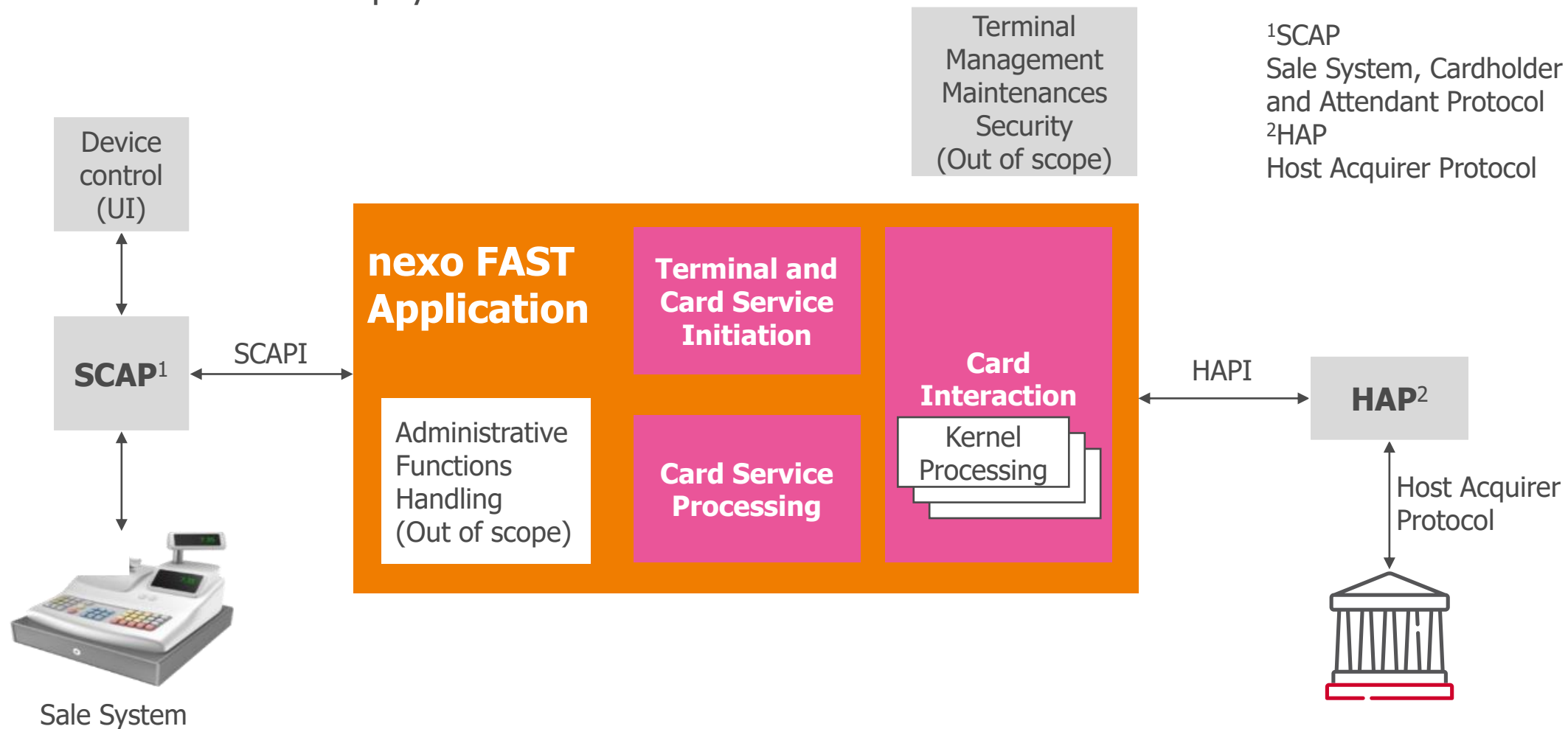
Summary

- 01 Business considerations
- 02 **nexo FAST architecture**
- 03 In scope
- 04 nexo FAST application structure
- 05 Data model
- 06 Interfaces to the host and to the sale system
- 07 Not described by nexo FAST



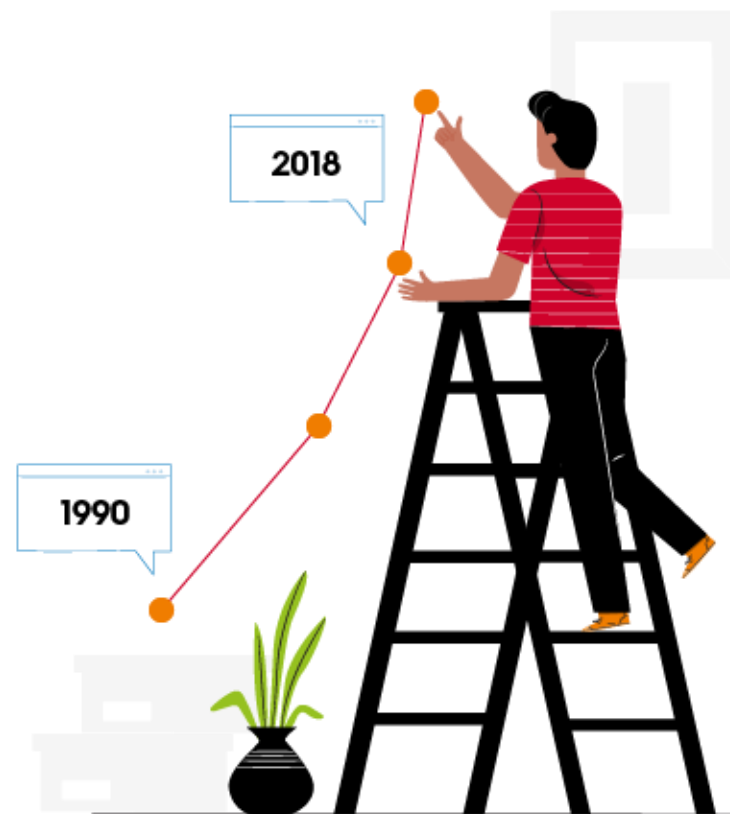
nexo FAST architecture

nexo FAST architecture in a payment context



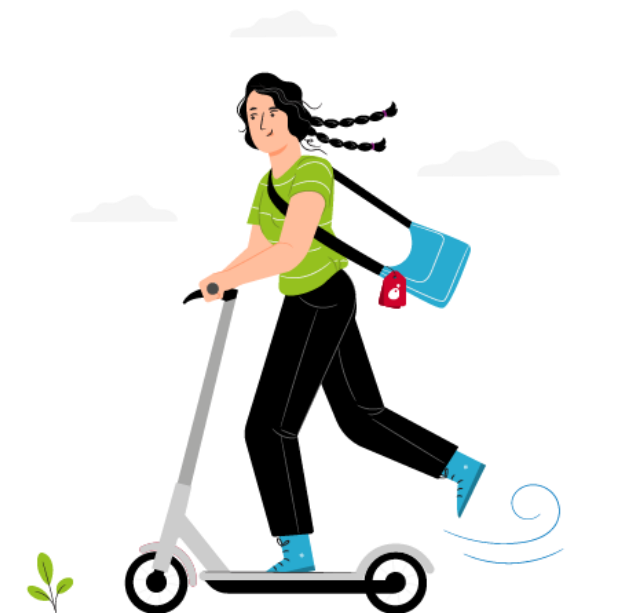
Summary

- 01 Business considerations
- 02 nexo FAST architecture
- 03 **In scope**
- 04 nexo FAST application structure
- 05 Data model
- 06 Interfaces to the host and to the sale system
- 07 Not described by nexo FAST



In scope

Scope for nexo FAST



In scope

Services : Payment Service, Refund Service, Cancellation Service.....

Terminal environments : Online only, Offline with online capabilities and Offline only

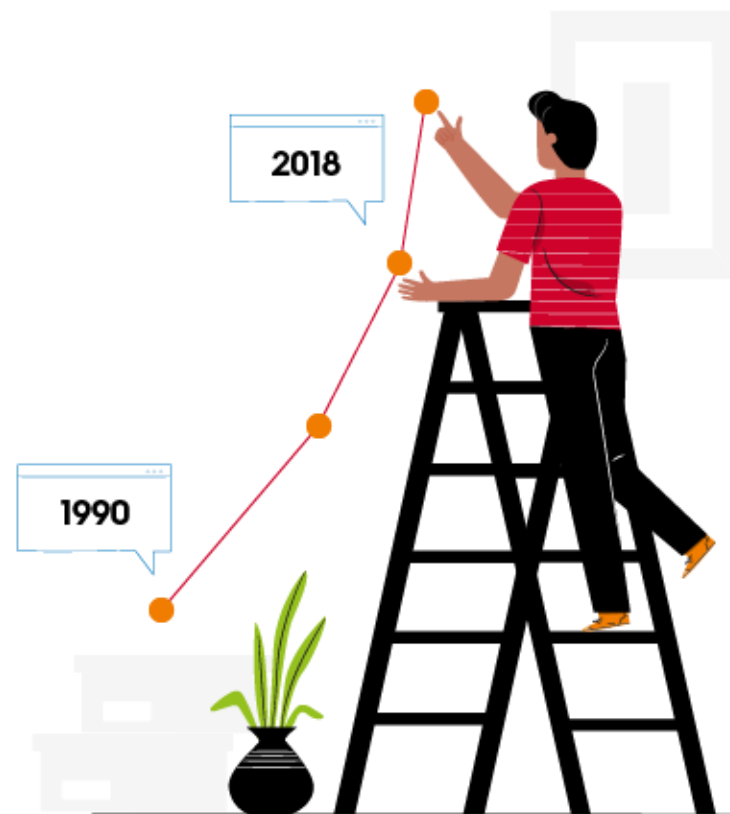
Interfaces : HAPI and SCAP

Data elements used to :

- └ configure a nexo FAST implementation
- └ perform a transaction with a nexo FAST implementation

Summary

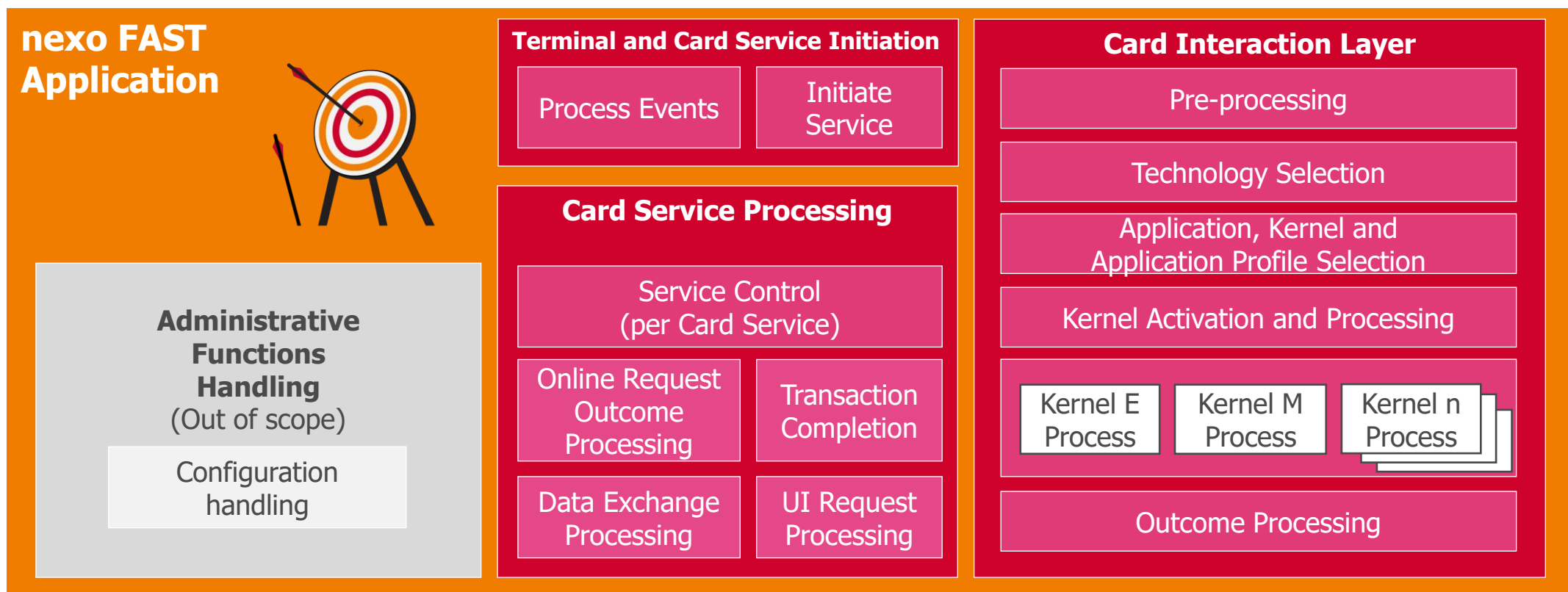
- 01 Business considerations
- 02 nexo FAST architecture
- 03 In scope
- 04 **nexo FAST application structure**
- 05 Data model
- 06 Interfaces to the host and to the sale system
- 07 Not described by nexo FAST



nexo FAST application structure

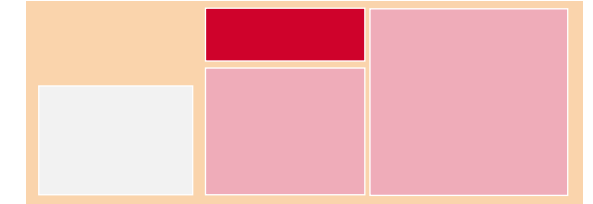
Overview of the nexo FAST application structure

**Terminal
Management
Maintenances
Security
(Out of scope)**

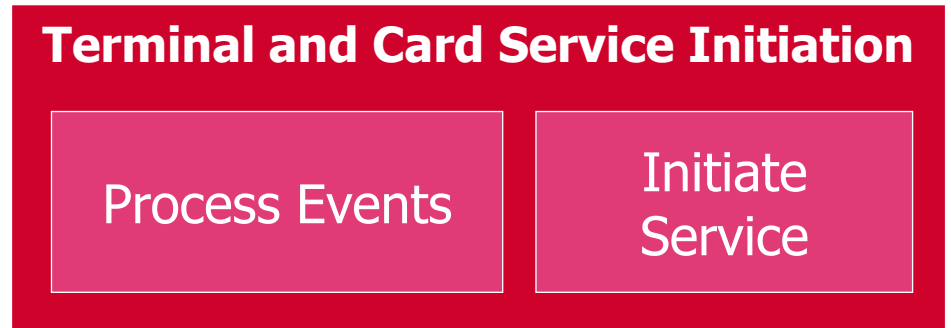


nexo FAST application structure

Terminal and card service initiation

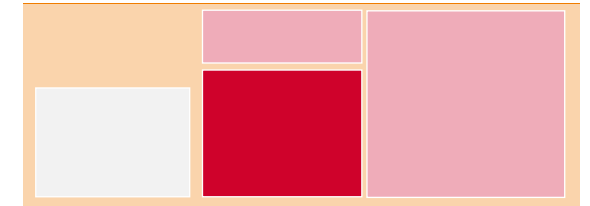


- **Terminal and Card Service Initiation** function is the top-level function of a nexo FAST application
- Financial Application **Start-up**
- « **Default Service** » Initialisation
- « **Idle state** » pending the events sent by SCAP
 - └ CARD INSERTED
 - └ CARD SWIPED
 - └ AMOUNT ENTRY
 - └ MANUAL ENTRY
 - └ REFERENCE ENTRY....
- **Process events and initialize associated services as needed**
 - └ The events required to start a service are configurable (*Allowed Service Start Events*)
 - └ Several events may be necessary to start a service
- **Transaction continues to the block 'Card Service Processing' to process the service**



nexo FAST application structure

Card service processing



— Service Control (Payment, Deferred Payment, Cancellation...)

- Transfer of execution to the '**Card Interaction Layer**' block during card dialogs

— User Interface Request Processing

- Interact with SCAP to send requests to the cardholder and receive answers (e.g. insert/remove card, see phone, ...)

— Data exchange processing

- Manage data exchange between Kernels and nexo FAST (PIN Entry, Exception file checking...)

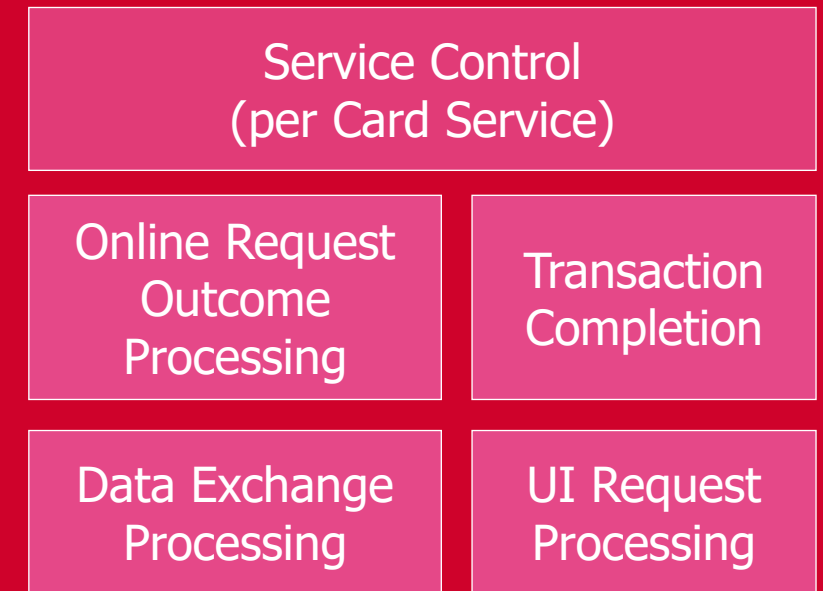
— Online Request Outcome Processing

- Outcome Processing from 'Card Interaction Layer' = ONLINE REQUEST

— Transaction completion

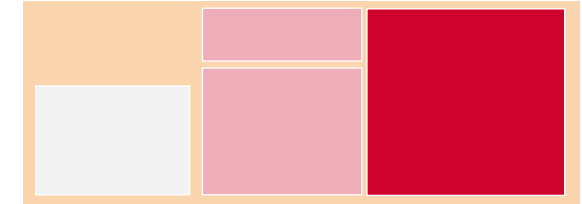
- It is the last step to perform to close the transaction
- Display card removal, receipt printing (SCAP via UI Request Processing)
- Reversal and advice handling (HAP)

Card Service Processing



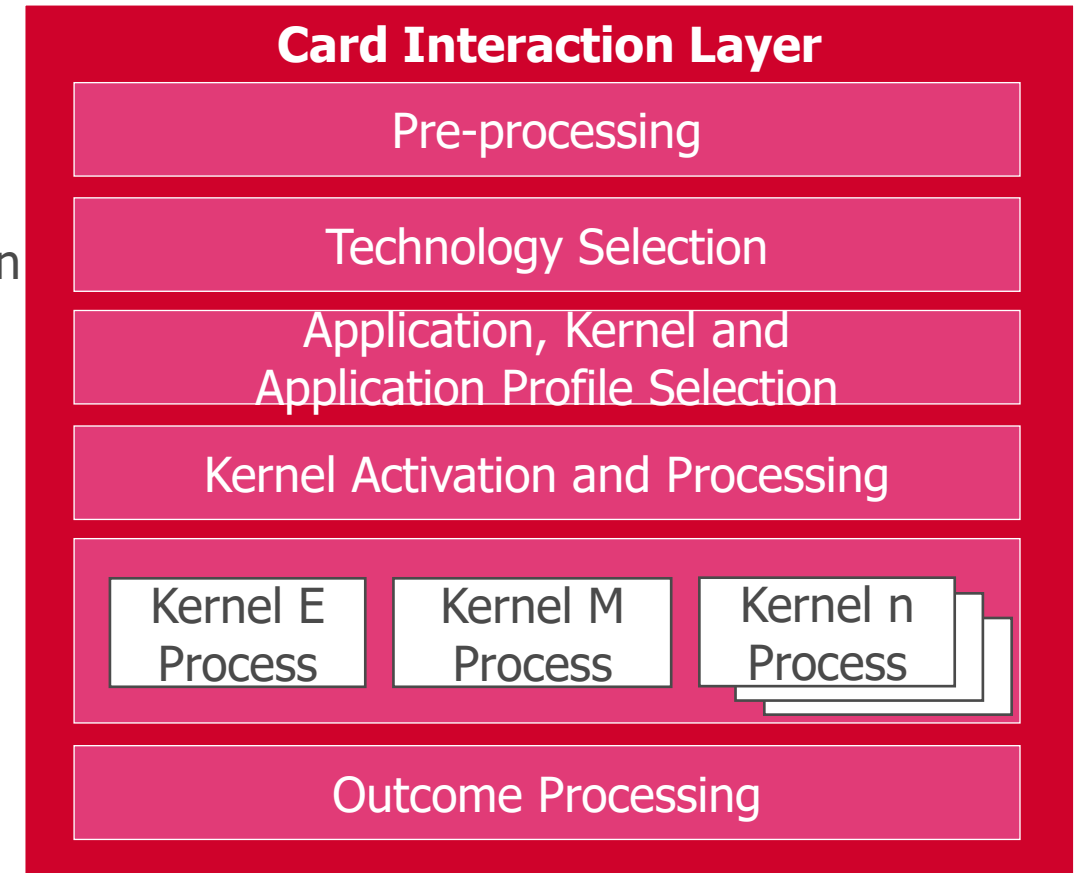
nexo FAST application structure

Card interaction layer



— Functions

- └ Pre-processing
- └ Technology selection (EMV Contact, Contactless, Magstripe, Manual entry)
- └ Application, Kernel and Application Profile Selection
- └ Activation and launch of the selected Kernel
- └ Processing of Kernel 'outcomes'

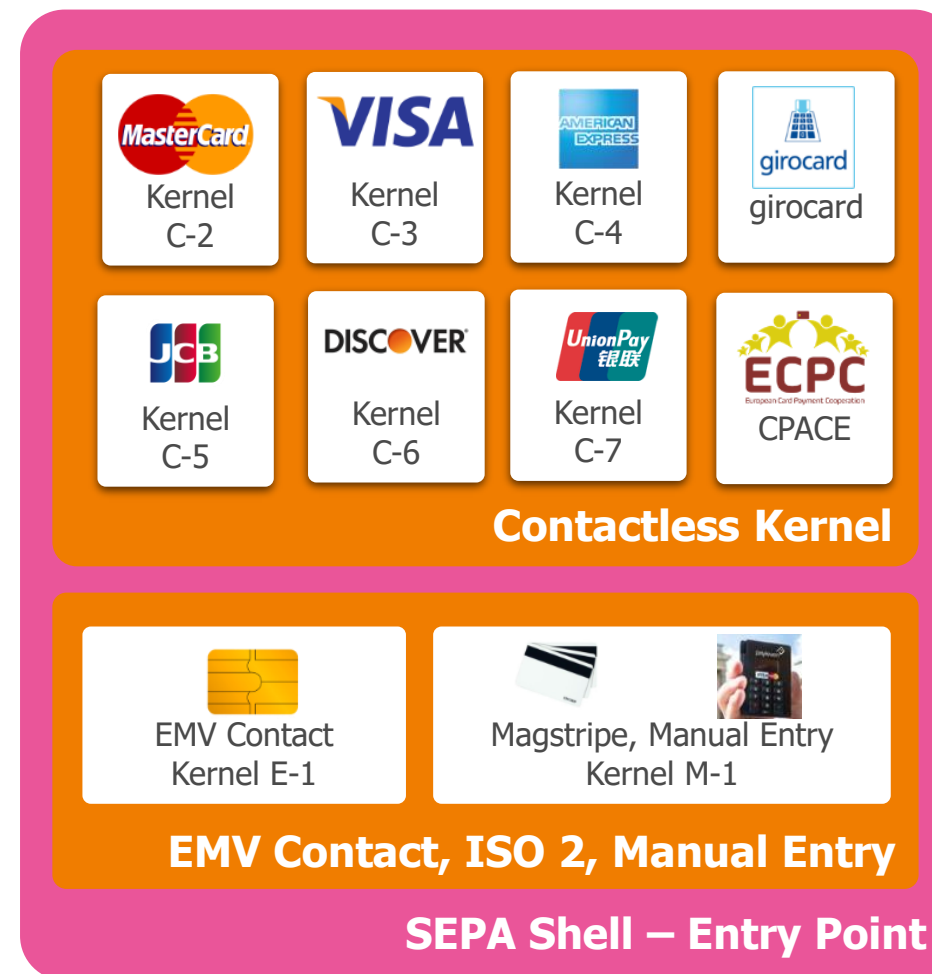


nexo FAST application structure

Kernels integration

— Extends the functions of the entry-point EMV contactless and embeds the Kernels

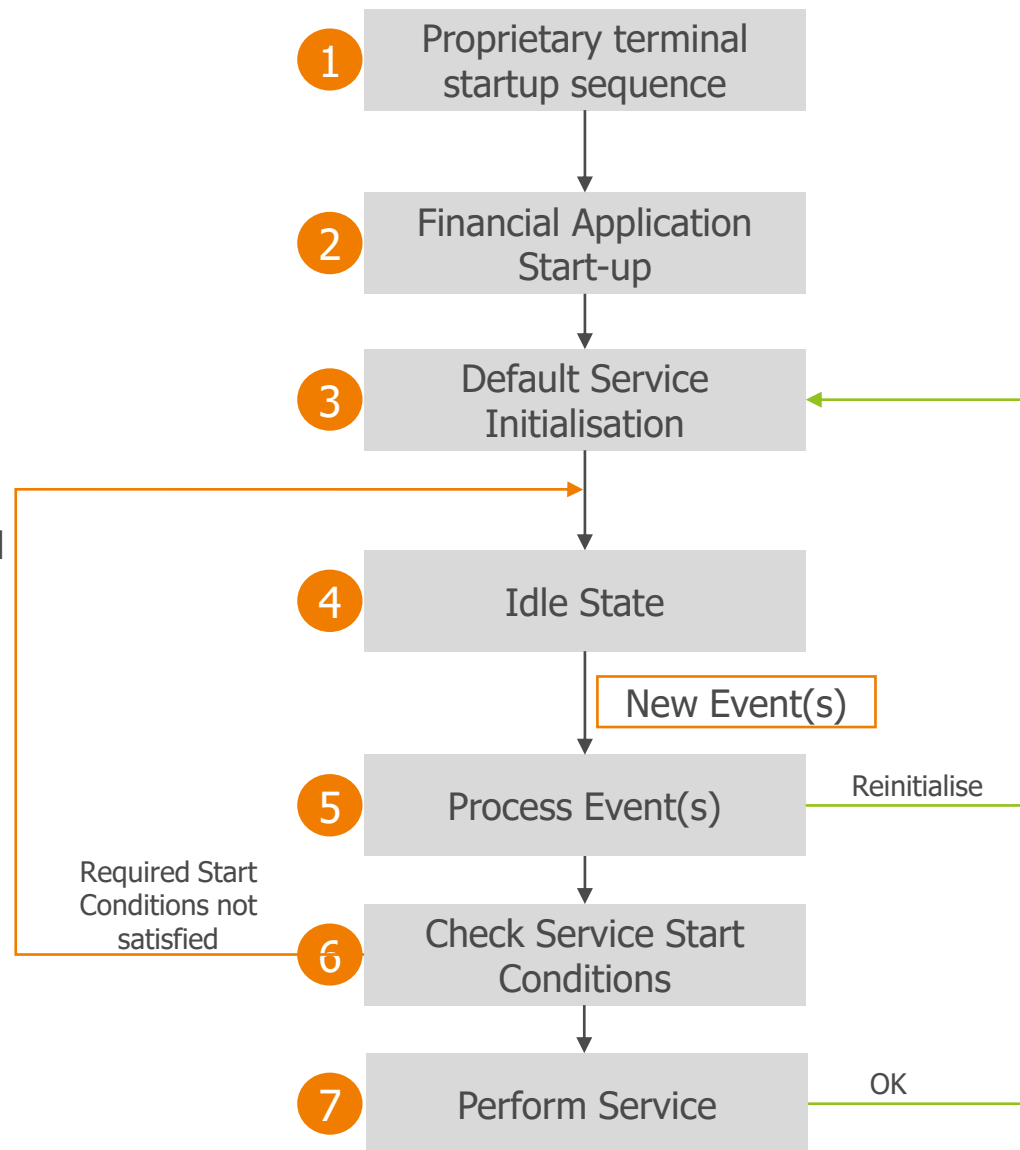
- └ C2-C7, girocard, CPACE : ContactLess
- └ E-1 : EMV contact
- └ M-1 : Magstripe and Manual Entry



nexo FAST application structure

Main flow

- 1 Starting the POI
- 2 Initializing the nexo FAST application
- 3 Launching the service defined as the default service
- 4 Standby state where the POI waits for the arrival of events sent by SCAP
- 5 Evaluation and treatment of events
- 6 Verification of the necessary conditions to launch the requested card service
- 7 Achievement of the card service

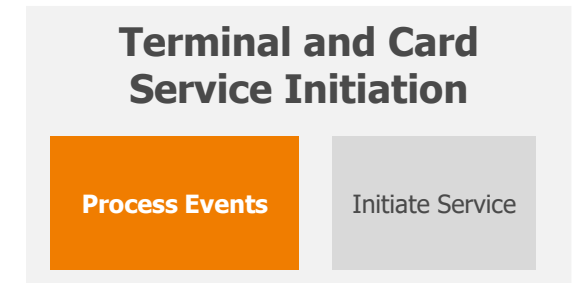
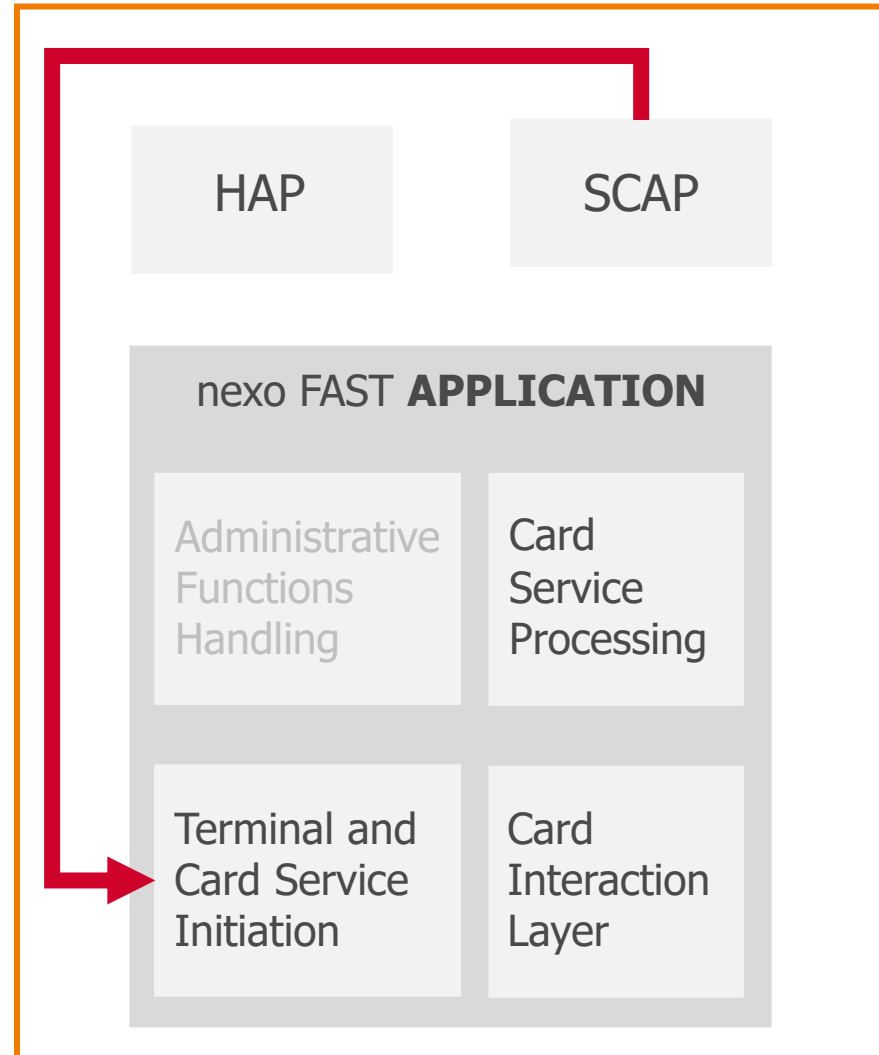


nexo FAST application structure

Payment flow

1

1. Merchant enters the amount
2. « **Process Events** » detects trigger events

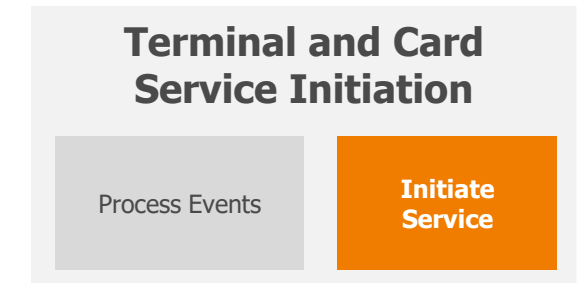
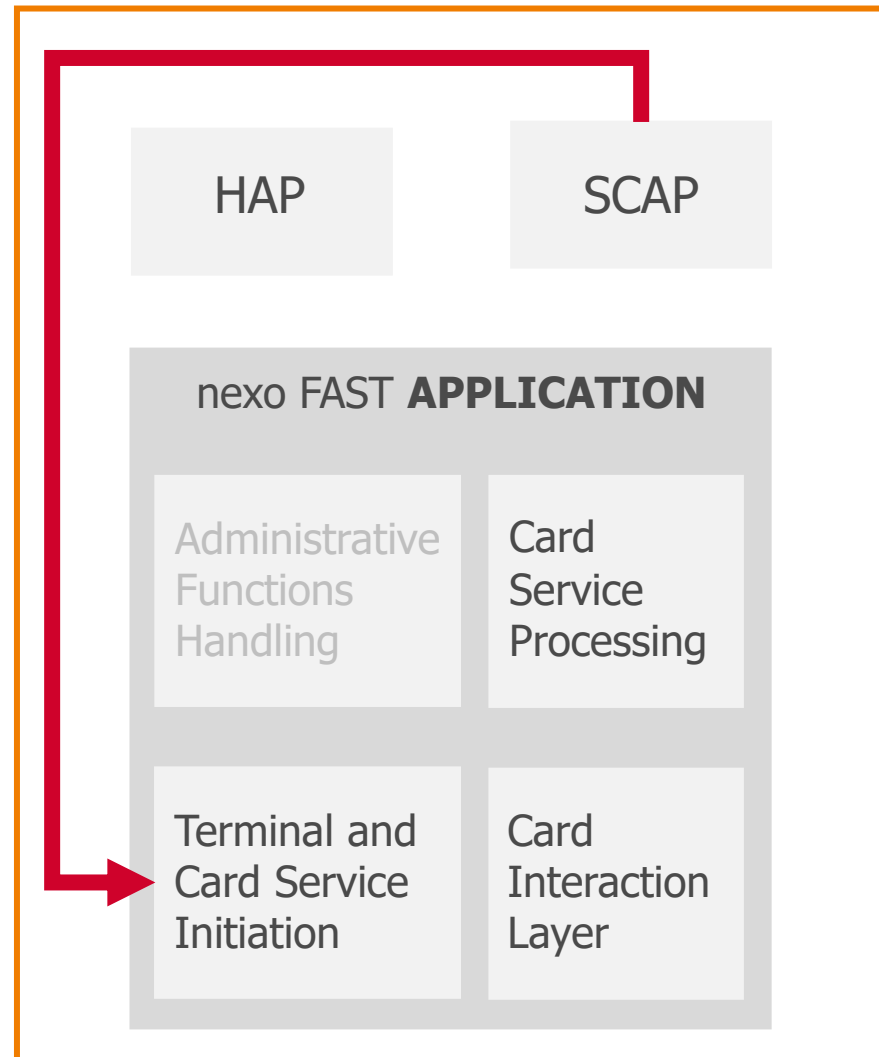


nexo FAST application structure

Payment flow

1

1. Merchant enters the amount
2. « **Process Events** » detects trigger events
3. « **Initiate Service** » initializes associated service

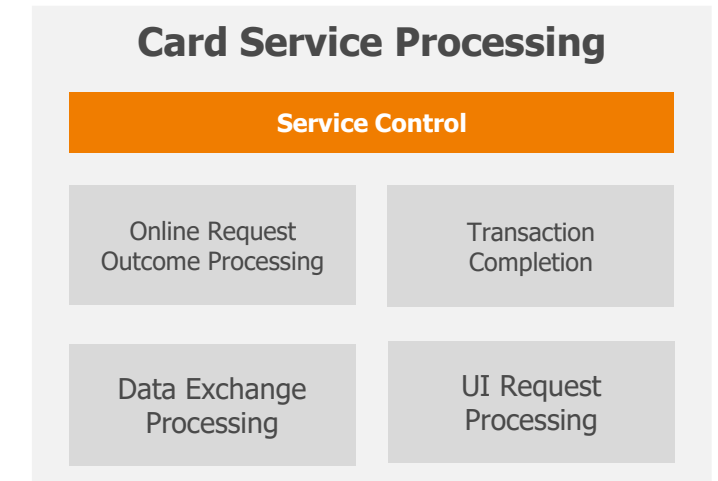
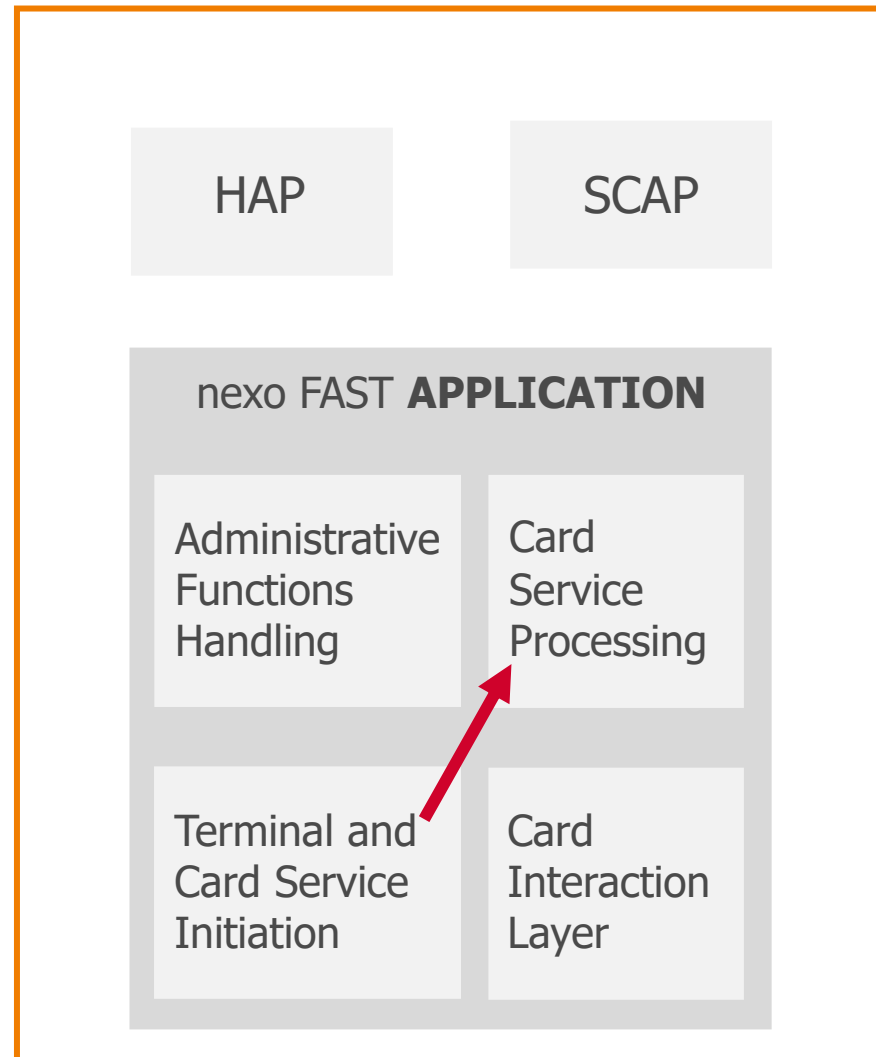


nexo FAST application structure

Payment flow

2

1. Transaction continues to 'Card Service Processing' to process the service
2. **"Service control"** verifies that all data needed by the payment service are present

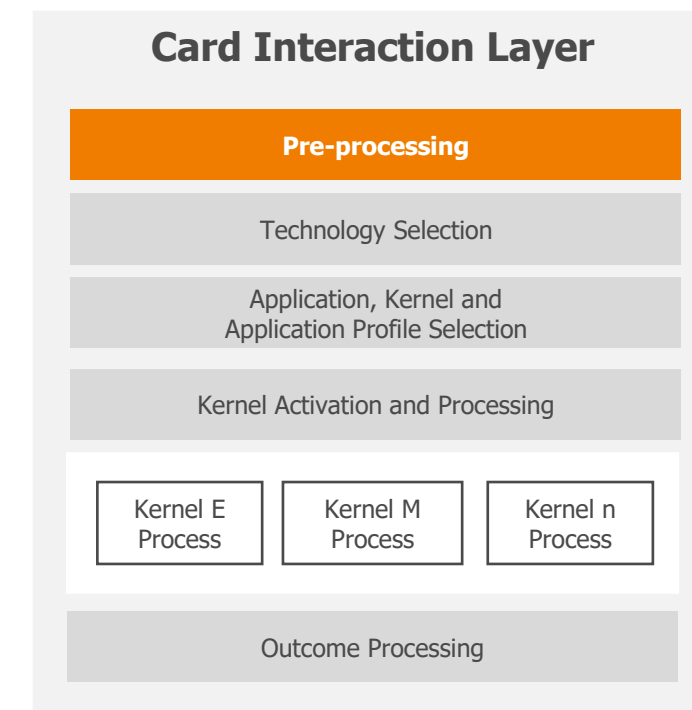
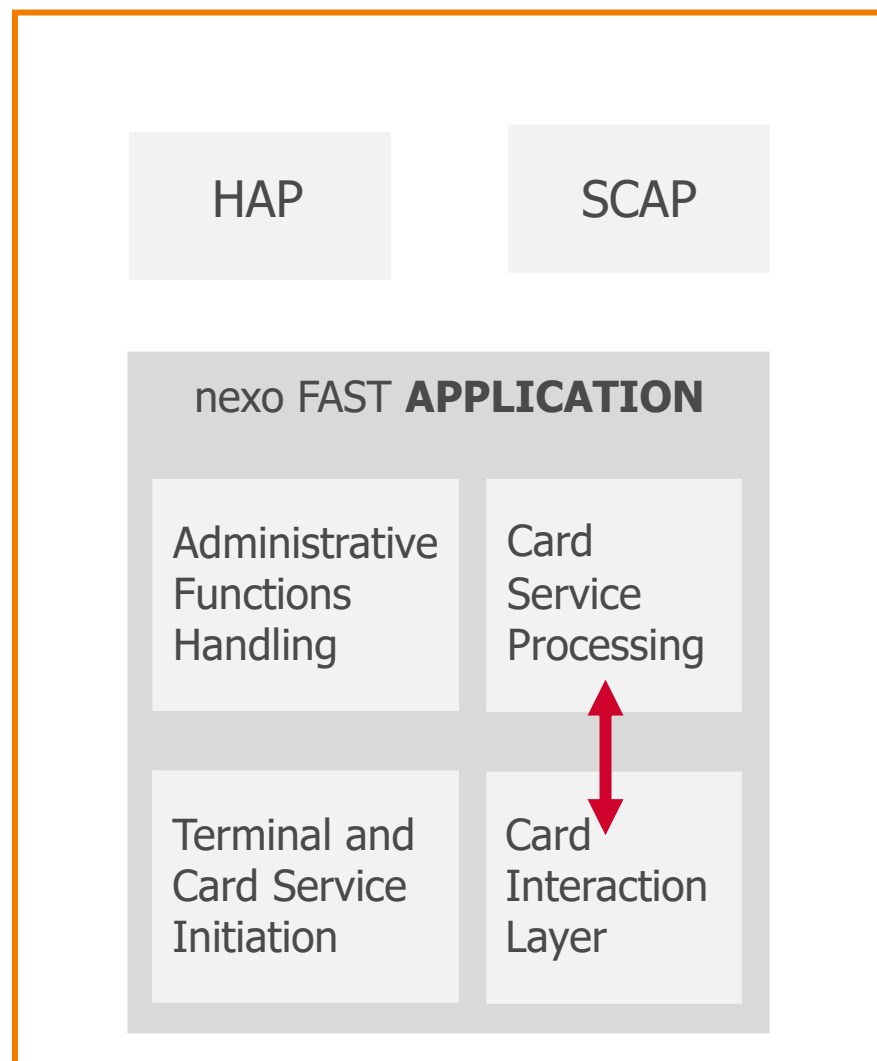


nexo FAST application structure

Payment flow

3

1. Pre-processing

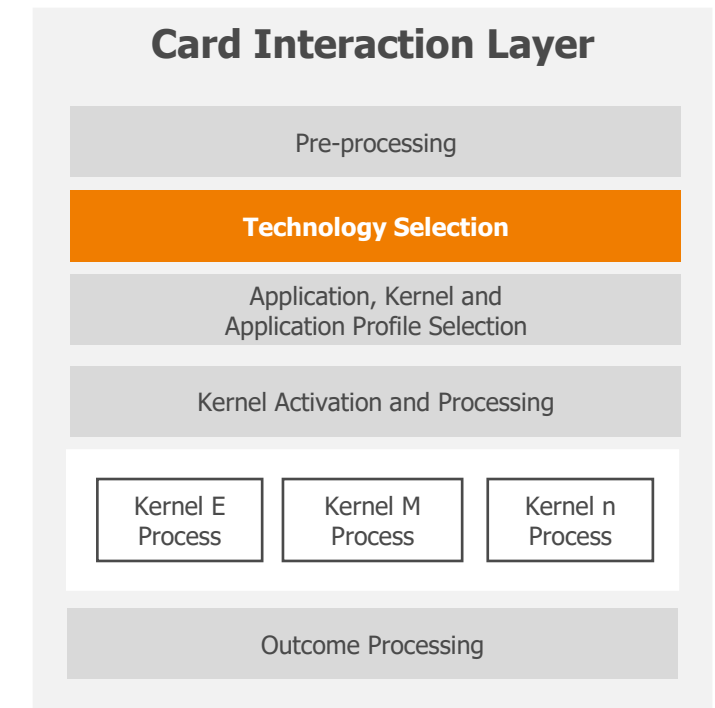
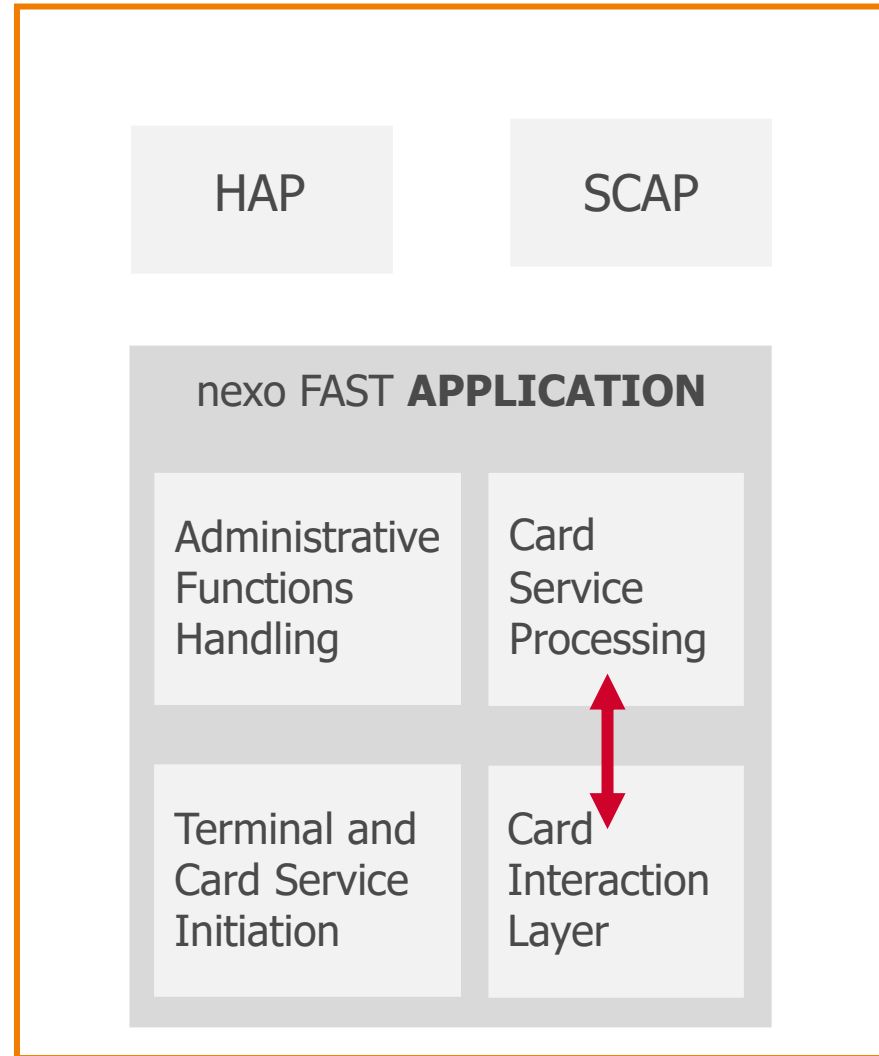


nexo FAST application structure

Payment flow

3

1. Pre-processing
2. Technology Selection

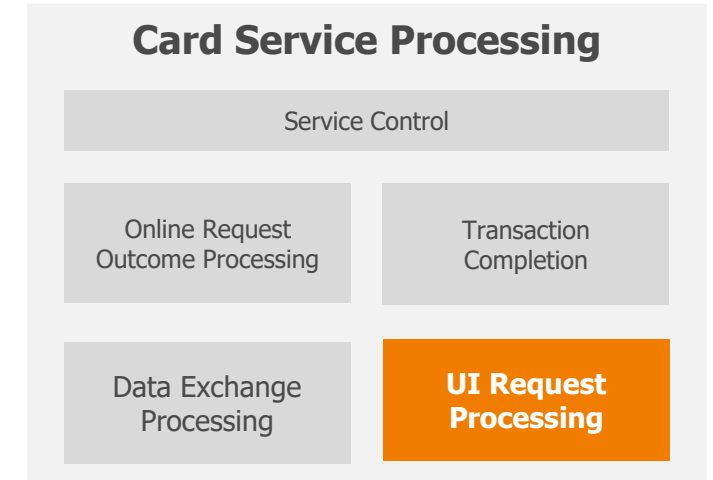
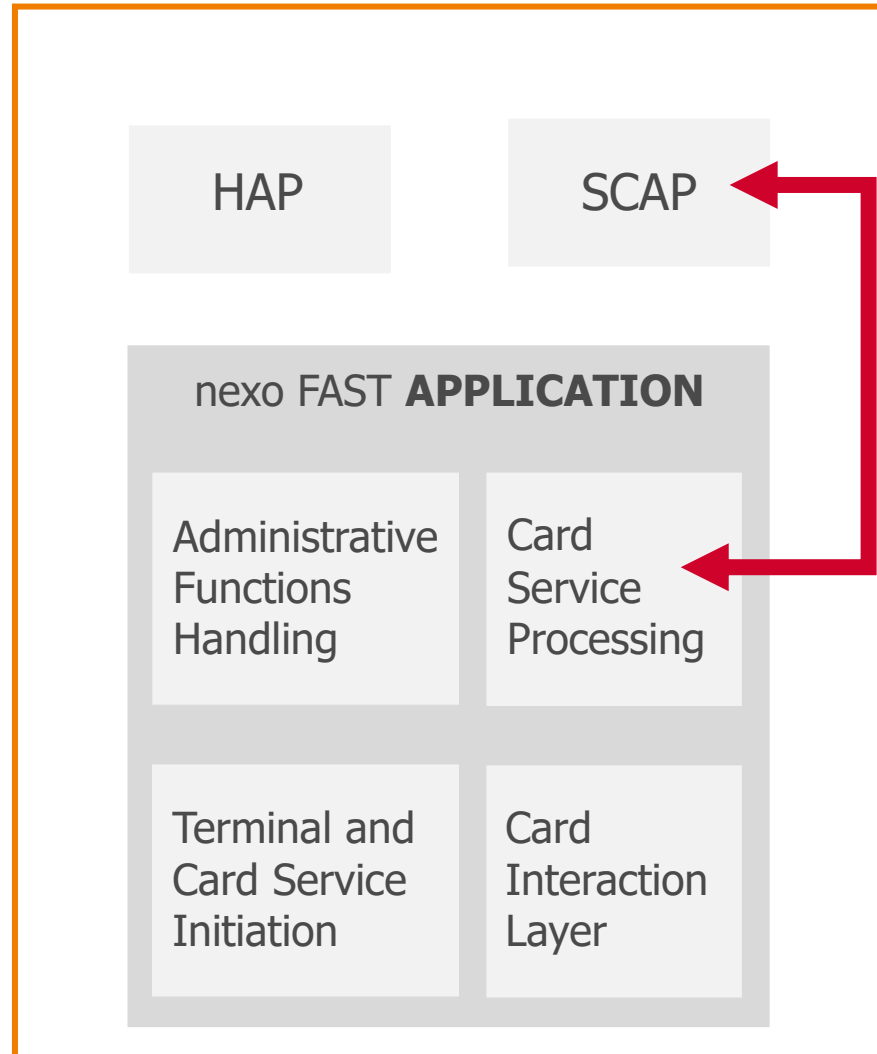


nexo FAST application structure

Payment flow

4

1. Transaction continues to 'Card Service Processing'
2. **"UI Request Processing"** sends to SCAP "INSERT CARD"

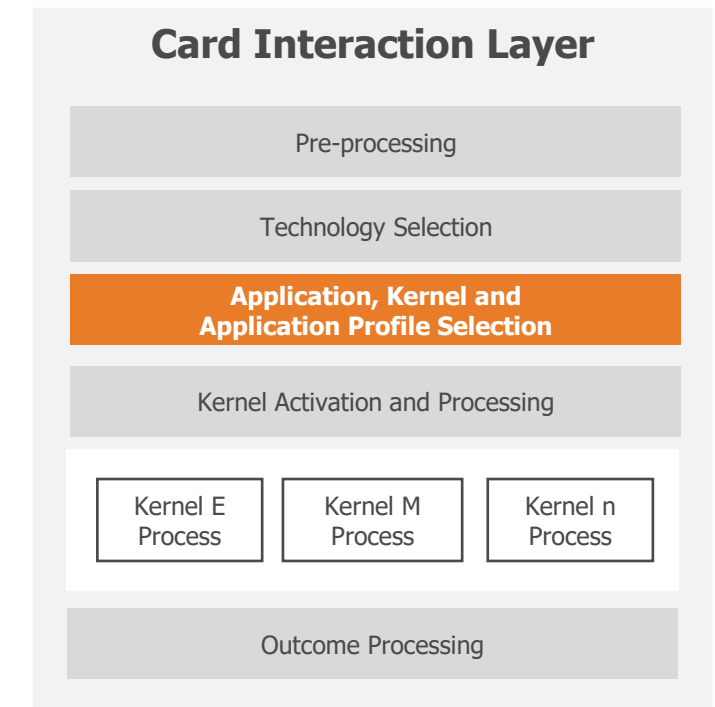
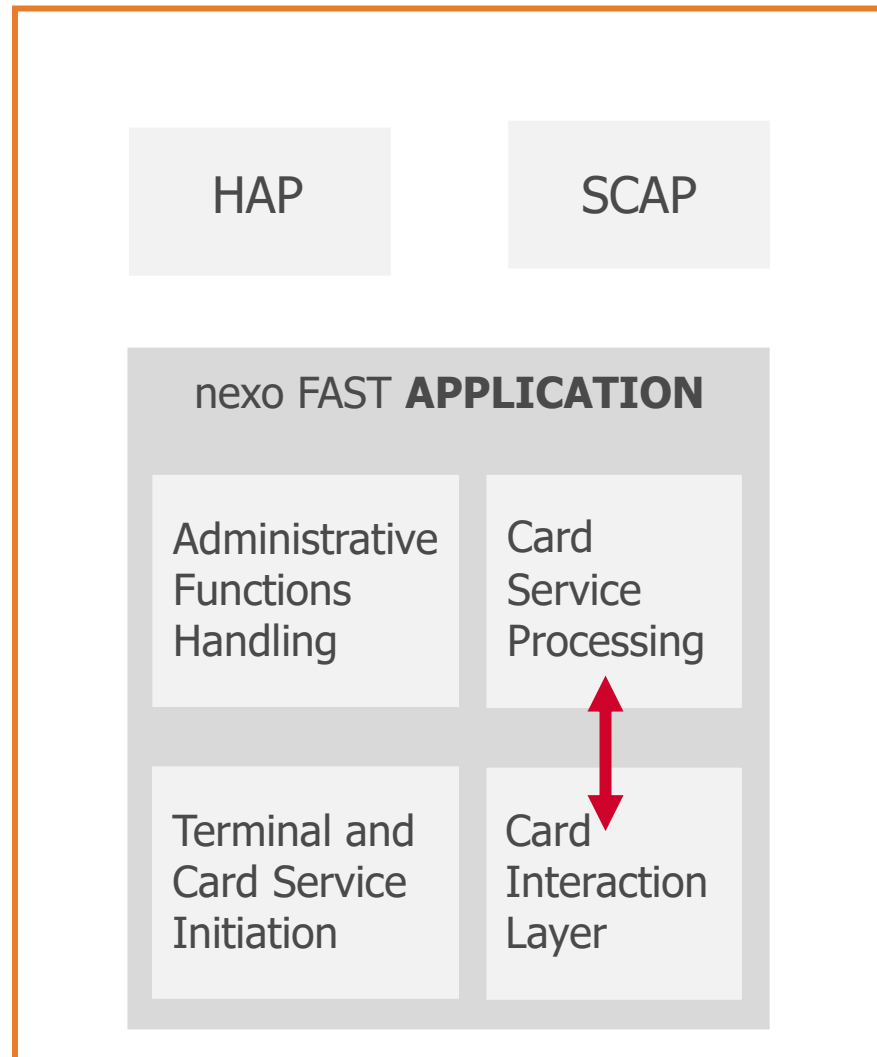


nexo FAST application structure

Payment flow

5

1. Pre-processing
2. Technology Selection
3. Selection of the application, Kernel and Application Profile

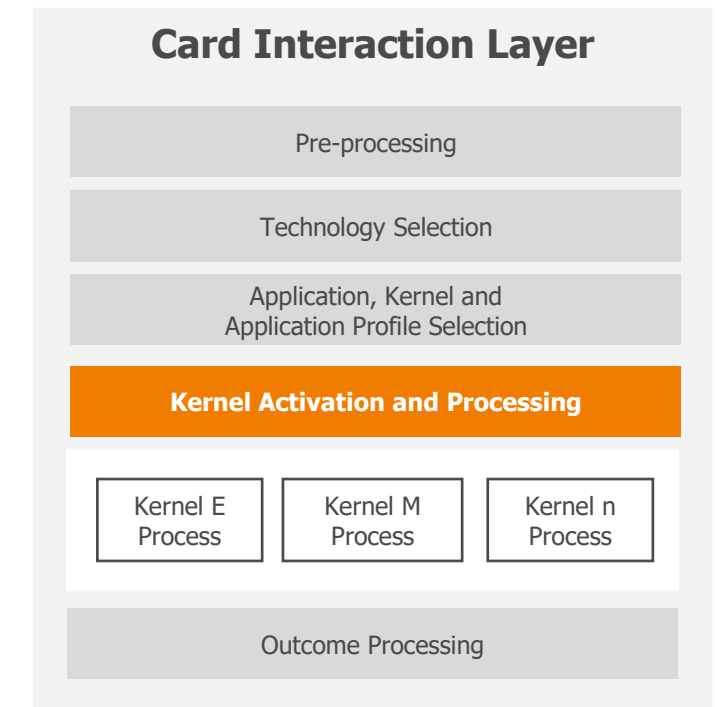
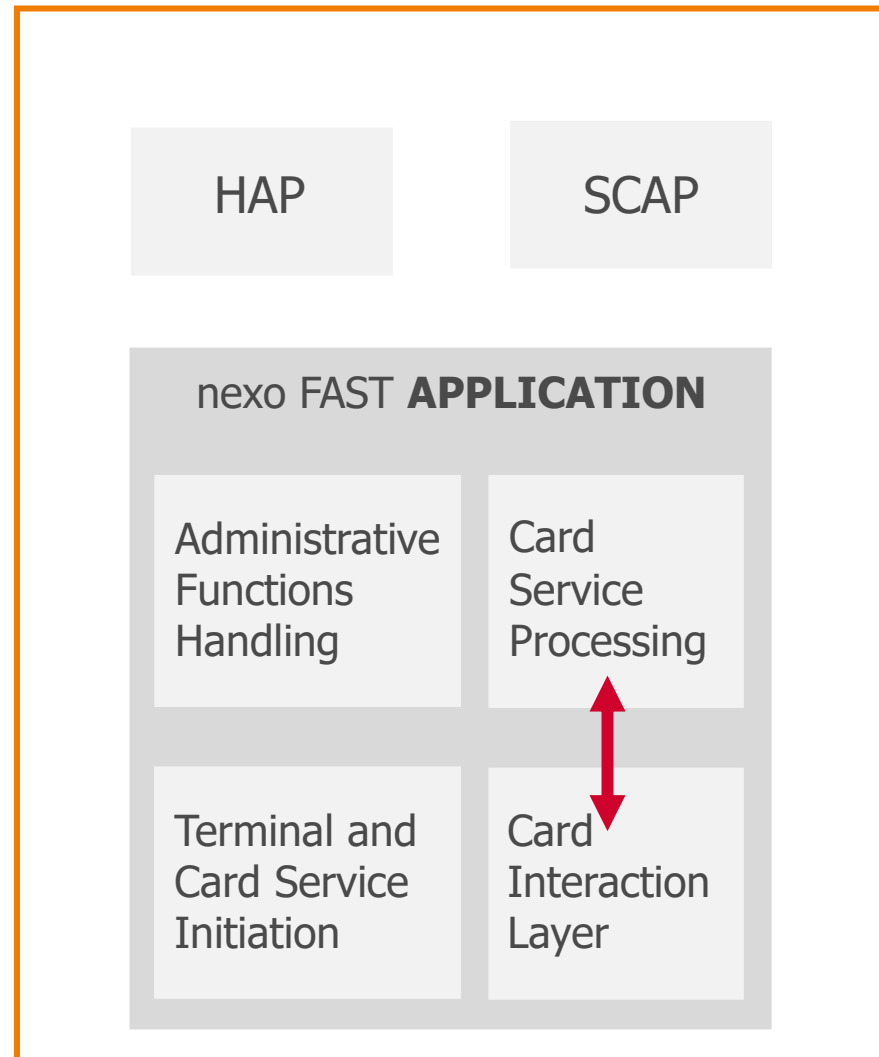


nexo FAST application structure

Payment flow

5

1. Pre-processing
2. Technology Selection
3. Selection of the application, Kernel and Application Profile
4. Kernel data loading

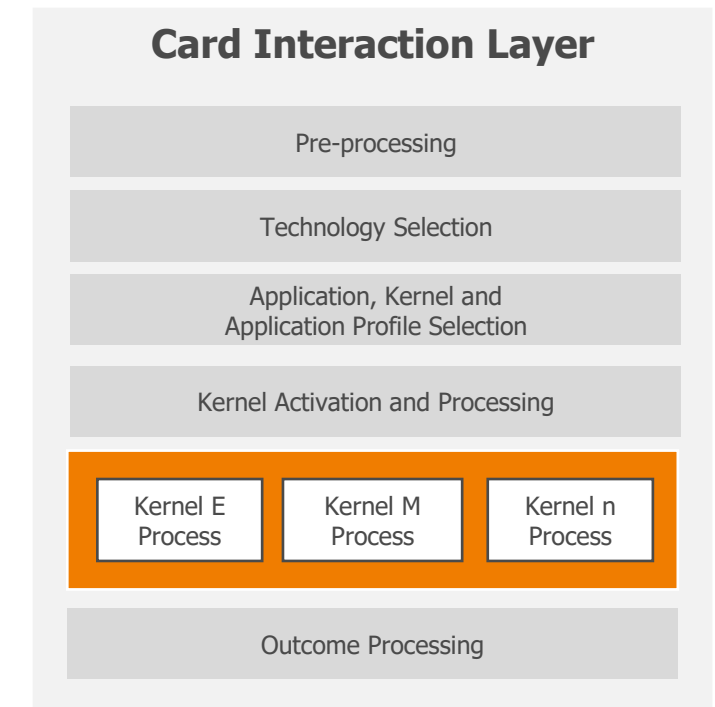
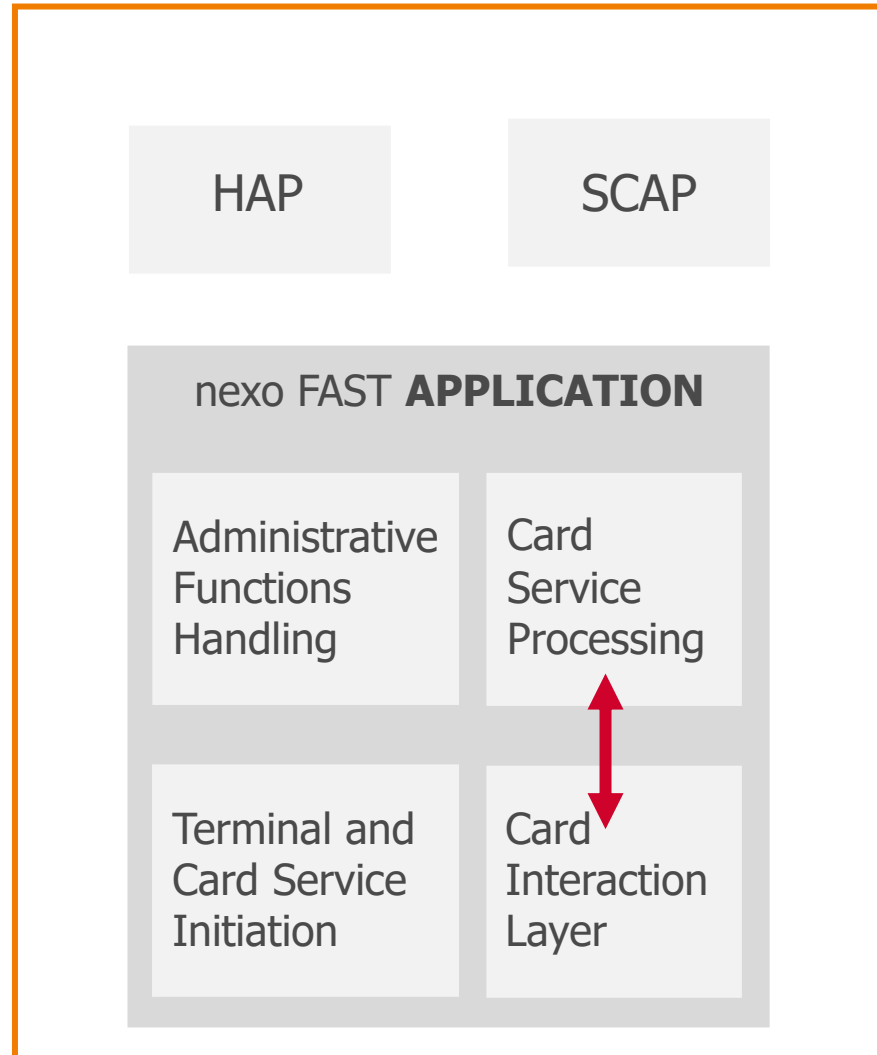


nexo FAST application structure

Payment flow

6

1. Pre-processing
2. Technology Selection
3. Selection of the application, Kernel and Application Profile
4. Kernel data loading
5. Transaction processing

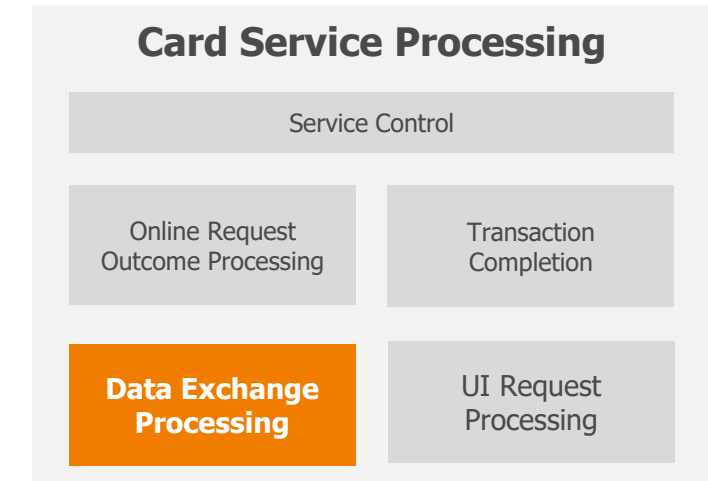
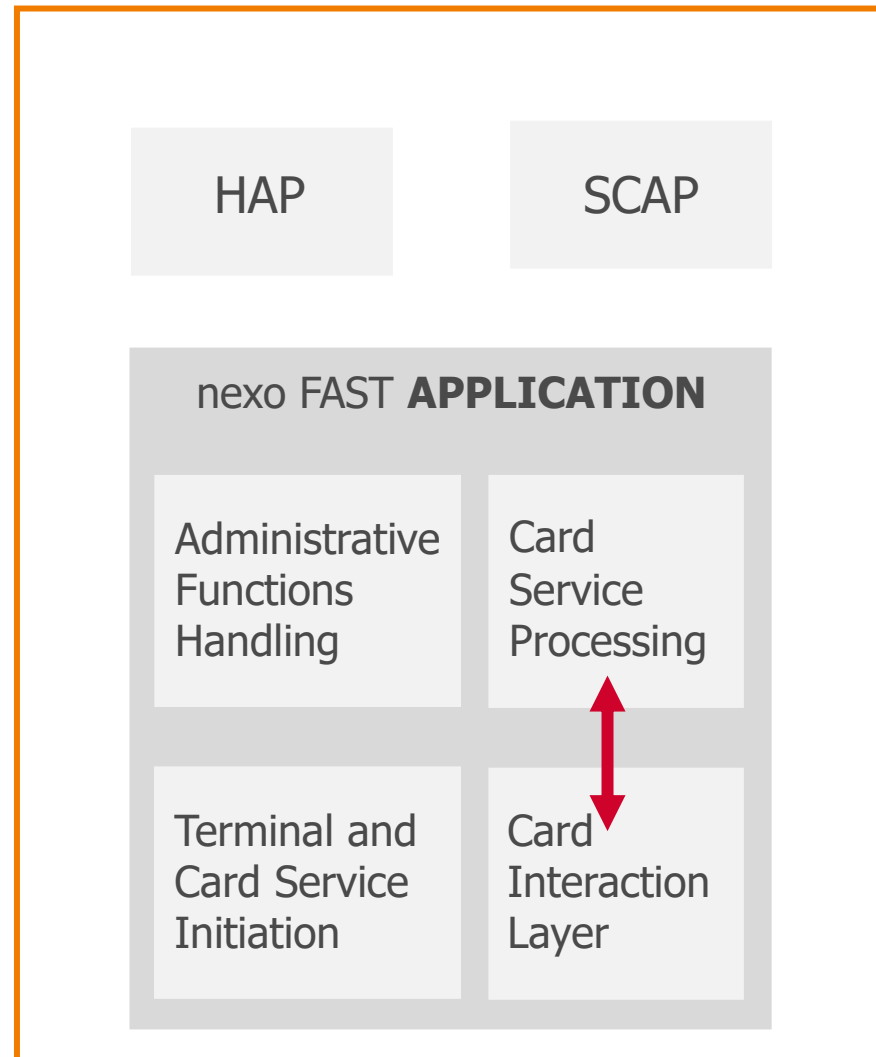


nexo FAST application structure

Payment flow

7

1. Transaction continues to 'Card Service Processing'
2. **"Data exchange processing"** retrieves PIN Entry

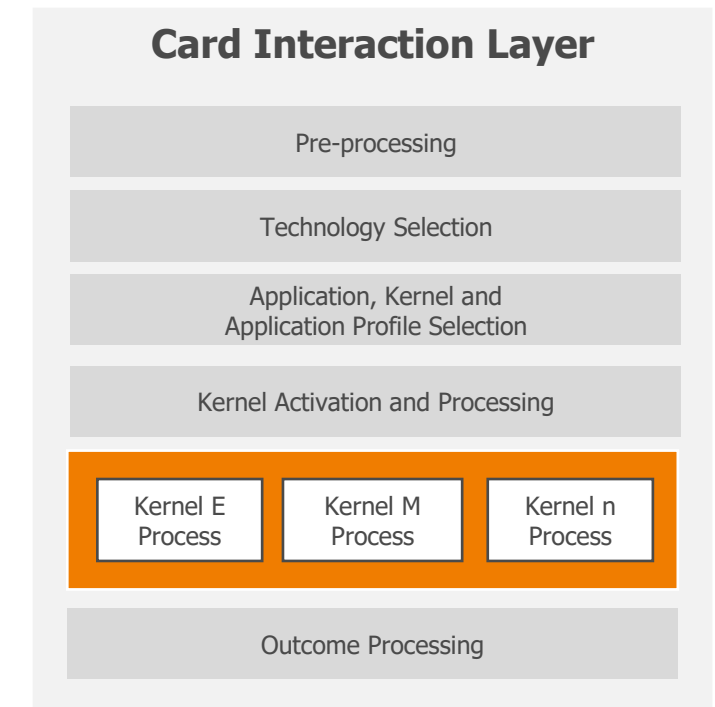
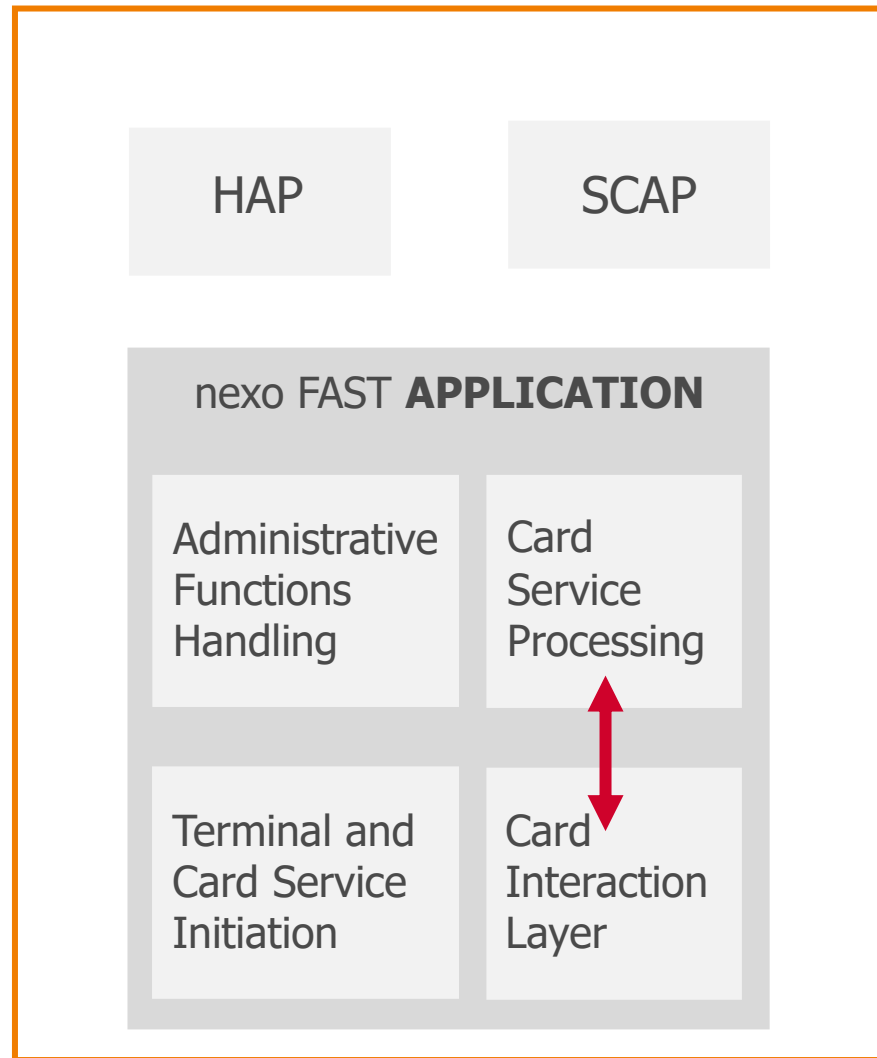


nexo FAST application structure

Payment flow

8

1. PIN verification then back to kernel process
2. Kernel requests online authorisation by setting Outcome to ONLINE REQUEST

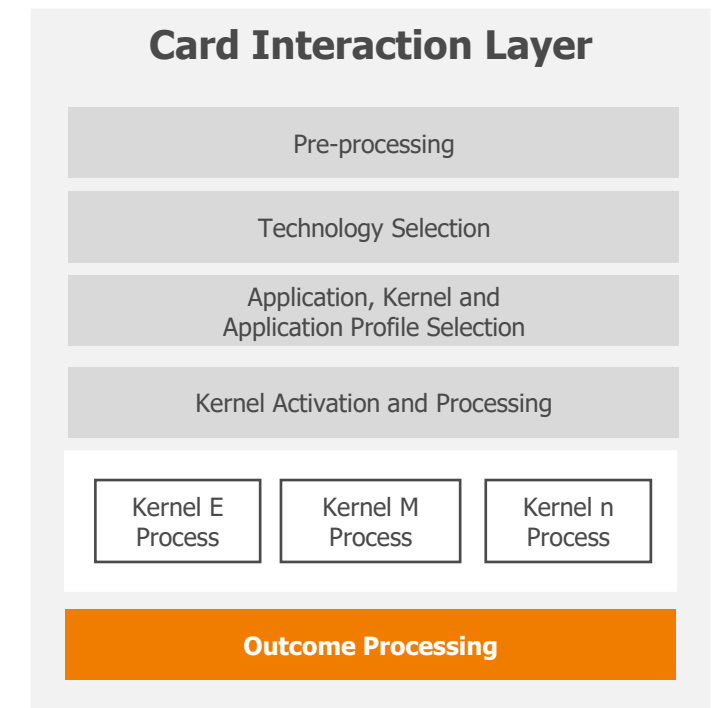
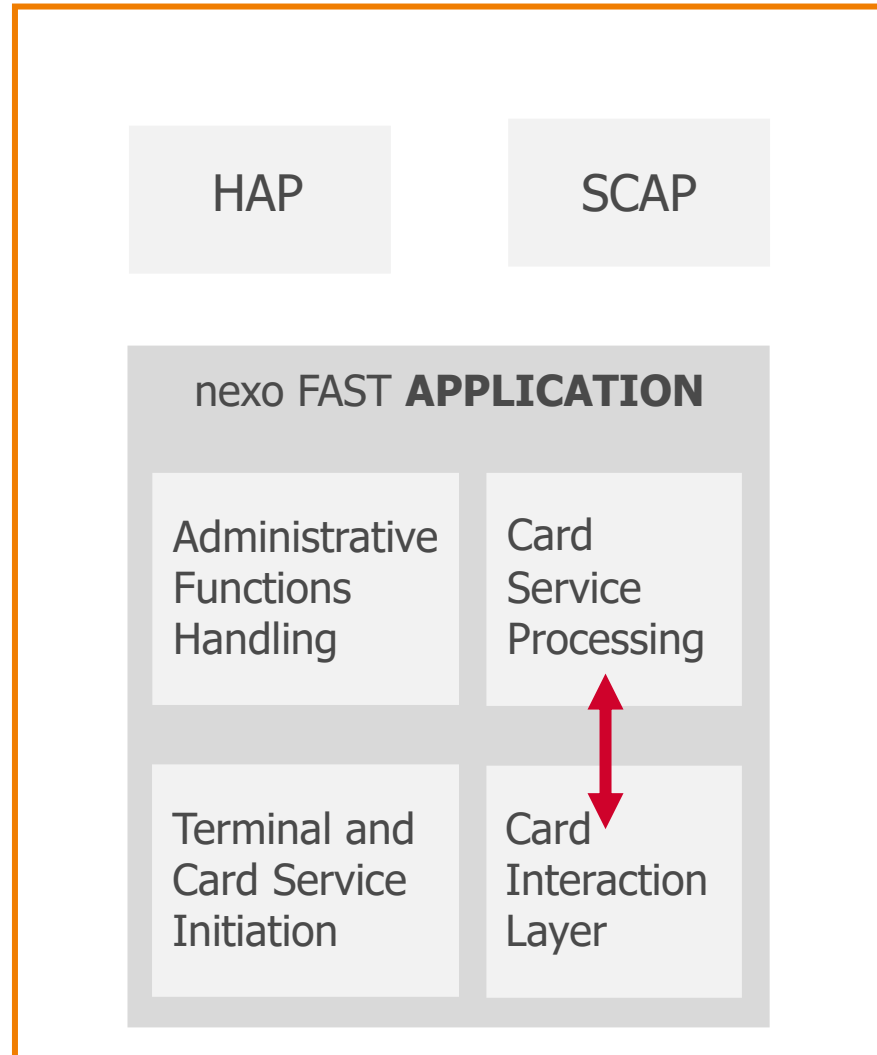


nexo FAST application structure

Payment flow

8

1. PIN verification then back to kernel process
2. Kernel requests online authorisation by setting Outcome to ONLINE REQUEST

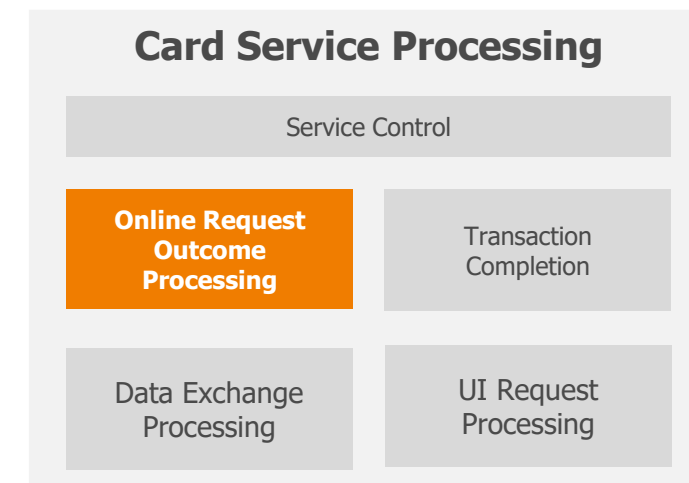
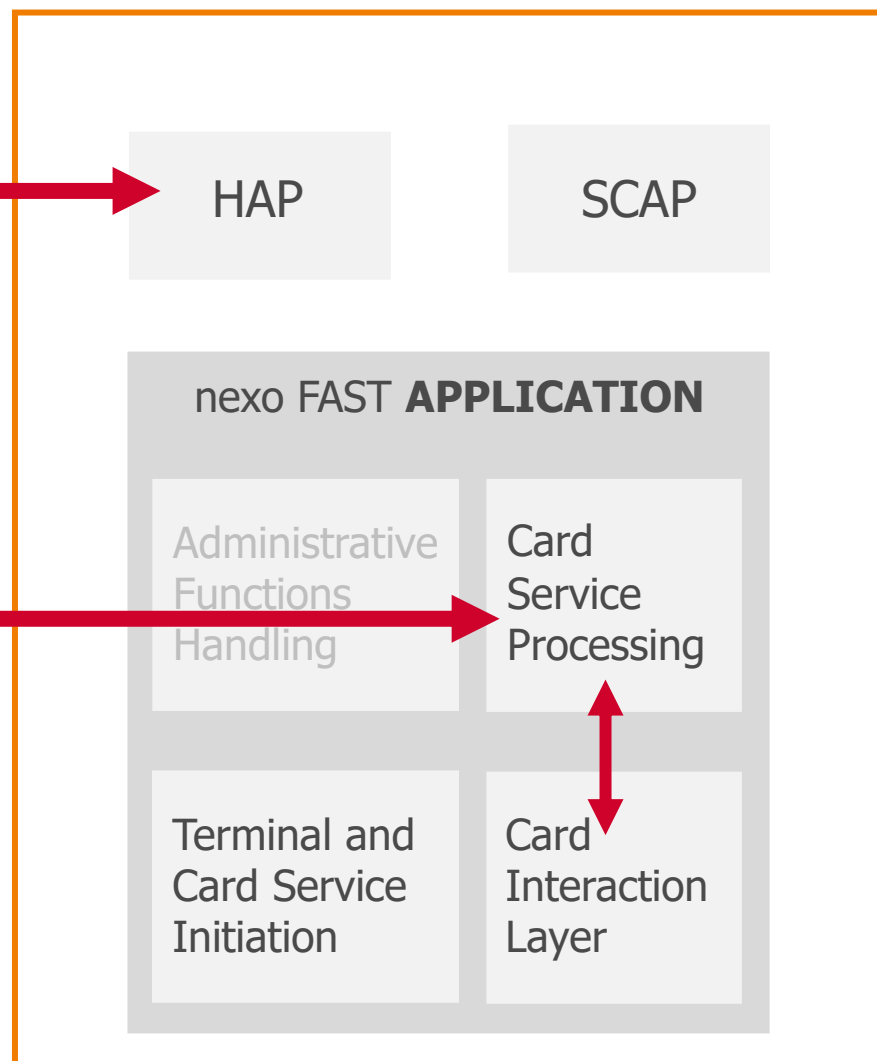


nexo FAST application structure

Payment flow

9

1. Transaction continues to 'Card Service Processing'
2. **"Online Request Outcome Processing"** sends an authorisation request via HAP
3. HAP sends an authorisation response to **"Online Request Outcome Processing"**

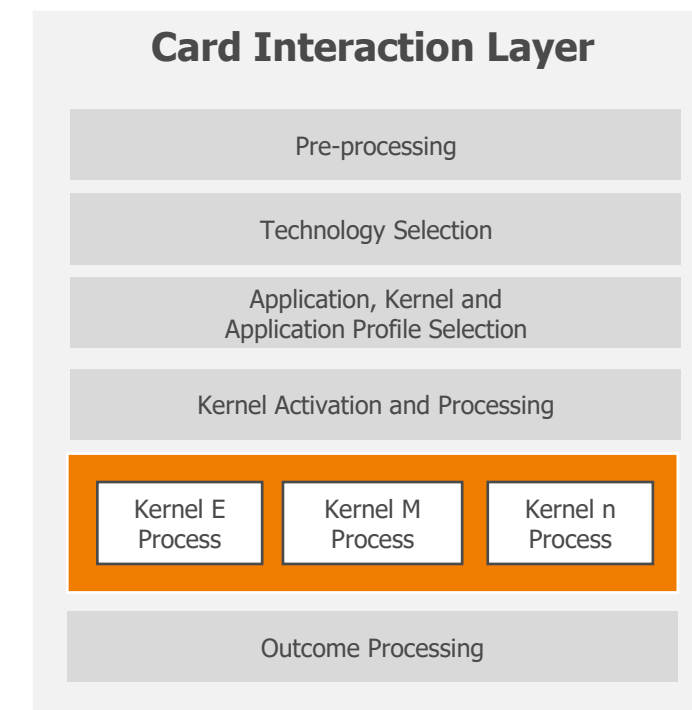
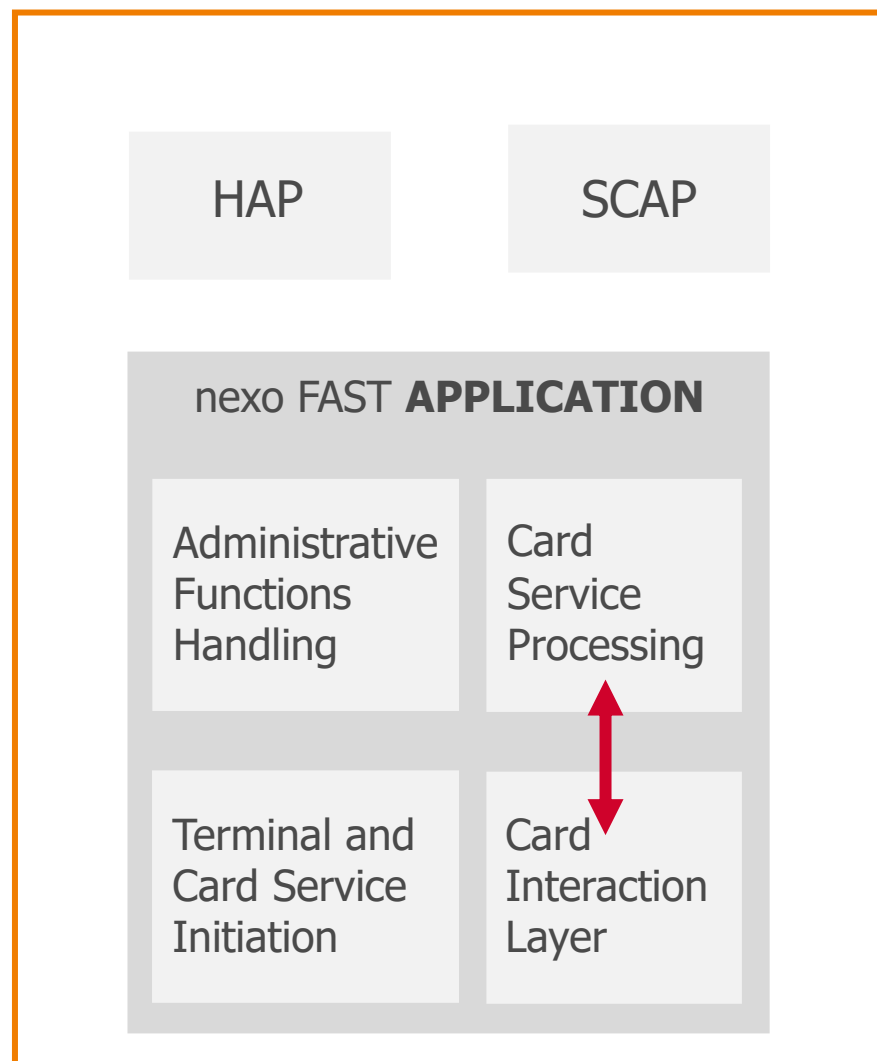


nexo FAST application structure

Payment flow

10

Kernel retrieves and analyses online authorisation response

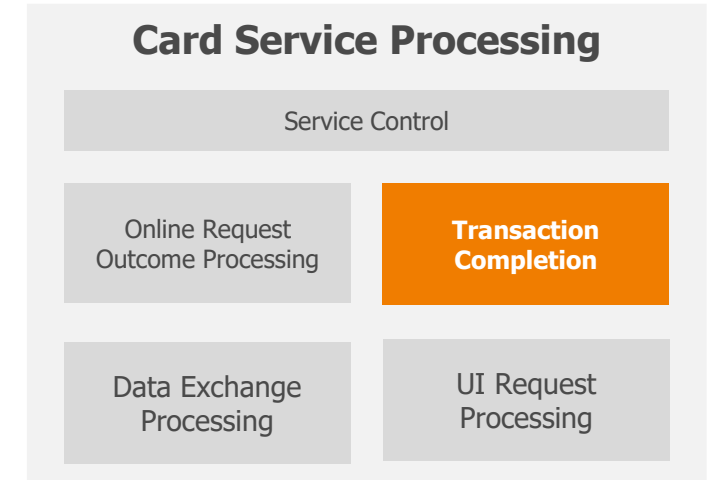
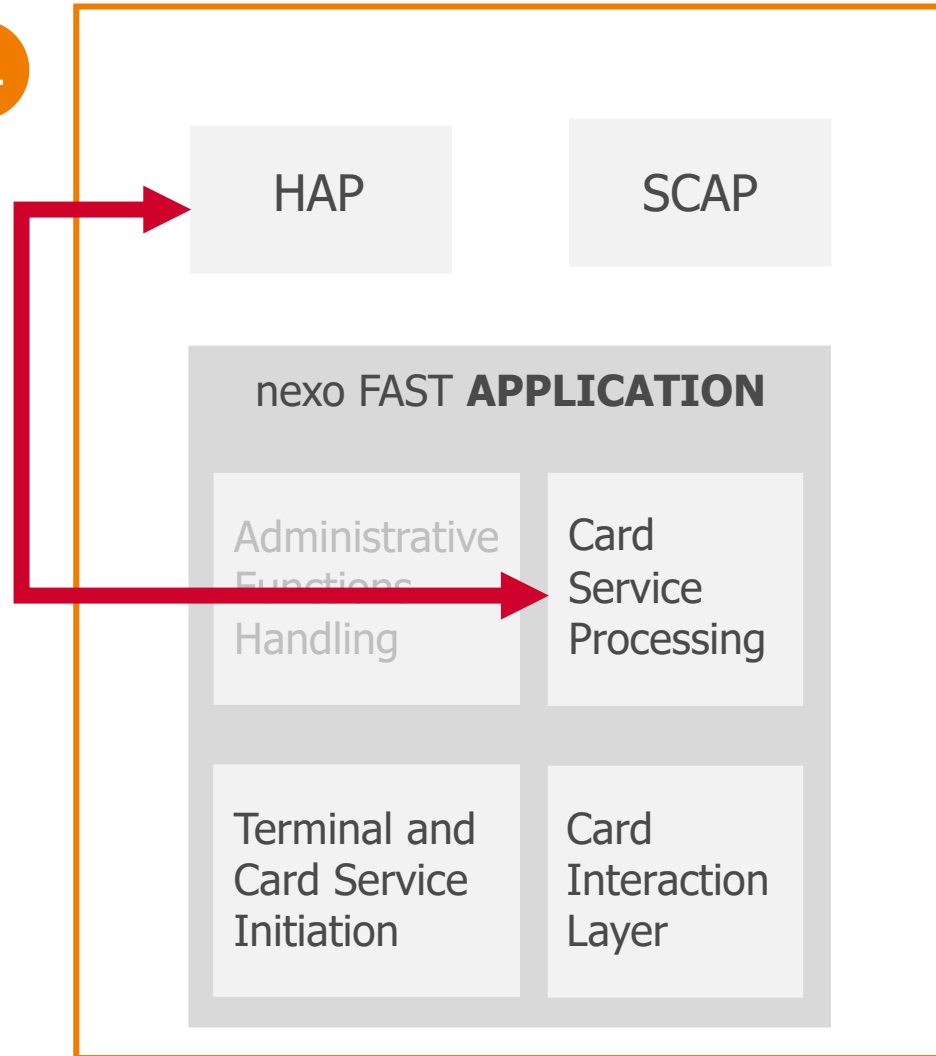


nexo FAST application structure

Payment flow

11

1. Transaction continues to the block 'Card Service Processing' to finalise the transaction
2. **"Transaction Completion"** e.g.: sends a completion advice via HAP, store the transaction for a future batch transfer

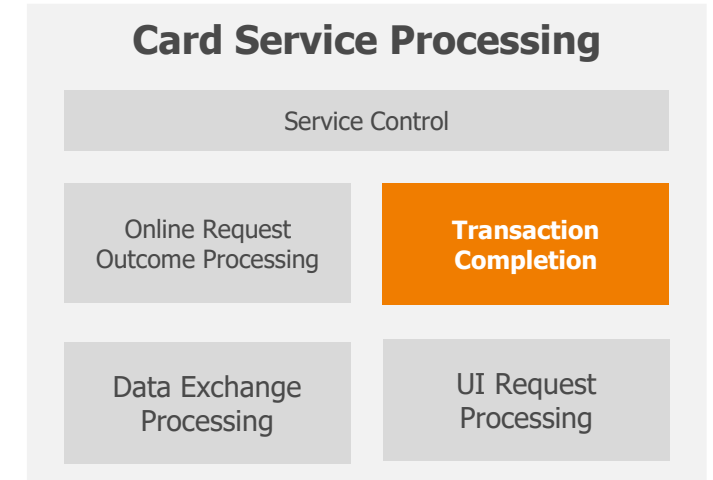
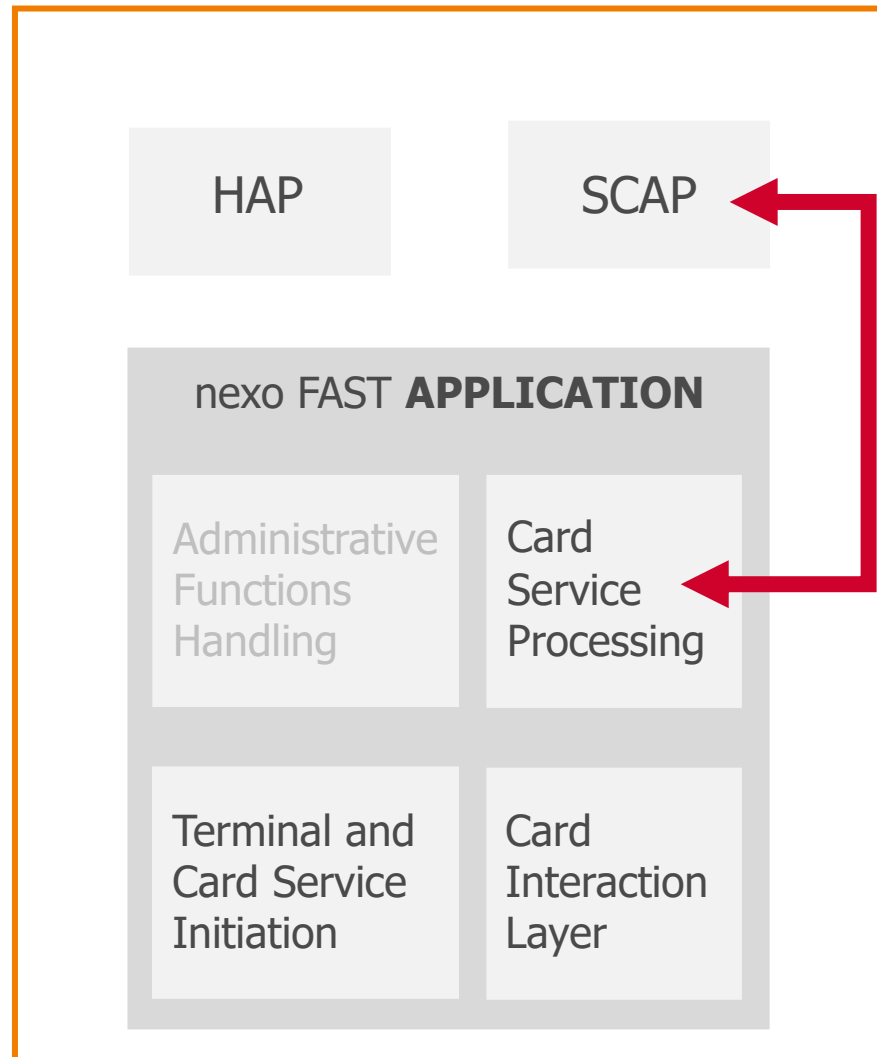


nexo FAST application structure

Payment flow

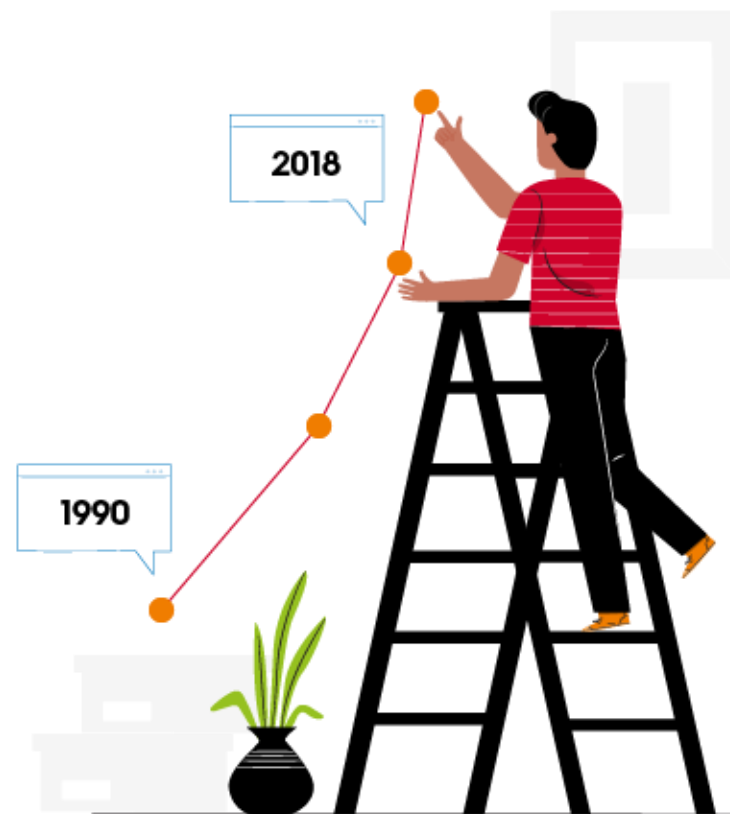
11

1. Transaction continues to the block 'Card Service Processing' to finalise the transaction
2. **"Transaction Completion" e.g.:** sends a completion advice via HAP, store the transaction for a future batch transfer
3. **"Transaction Completion" :** requires receipt printing to SCAP



Summary

- 01 Business considerations
- 02 nexo FAST architecture
- 03 In scope
- 04 nexo FAST application structure
- 05 **Data model**
- 06 Interfaces to the host and to the sale system
- 07 Not described by nexo FAST



Data model

Introduction

¹TMAP

Terminal Management System Protocol

Global data elements defined in the nexo FAST Data Dictionary are classified into 2 main categories

— Terminal Configuration Data

- Set of Terminal Configuration Parameters (modified by TMAP¹ - out of scope)

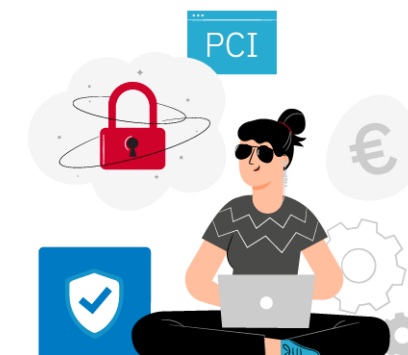
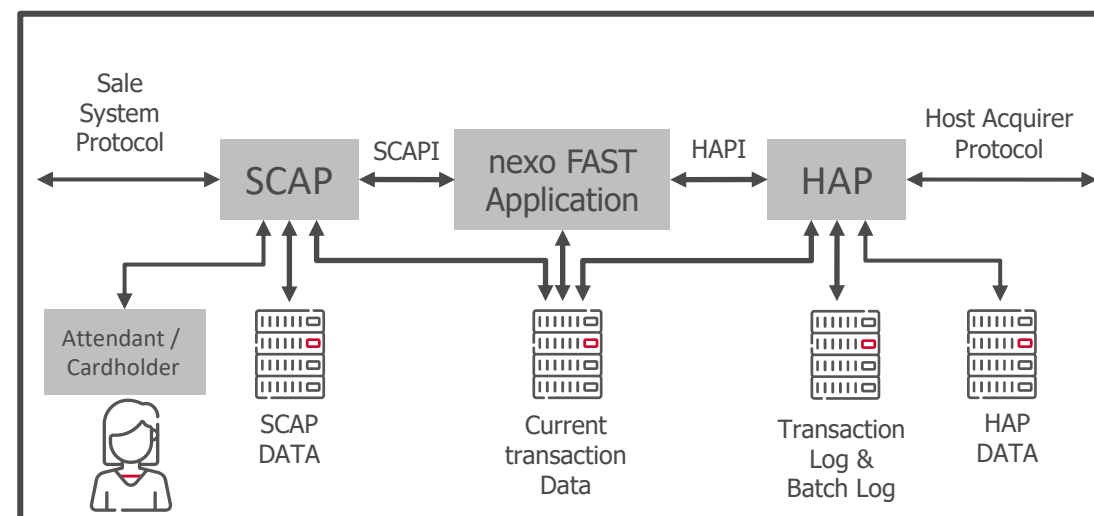
— Terminal Transaction Data

- Local data related to the current financial transaction (modified by HAP and SCAP)

Kernel Data

Card Data

Online Response Data



Data model

Terminal configuration data

— Persistent data, themselves divided into two categories

- Terminal level: one instance in the terminal

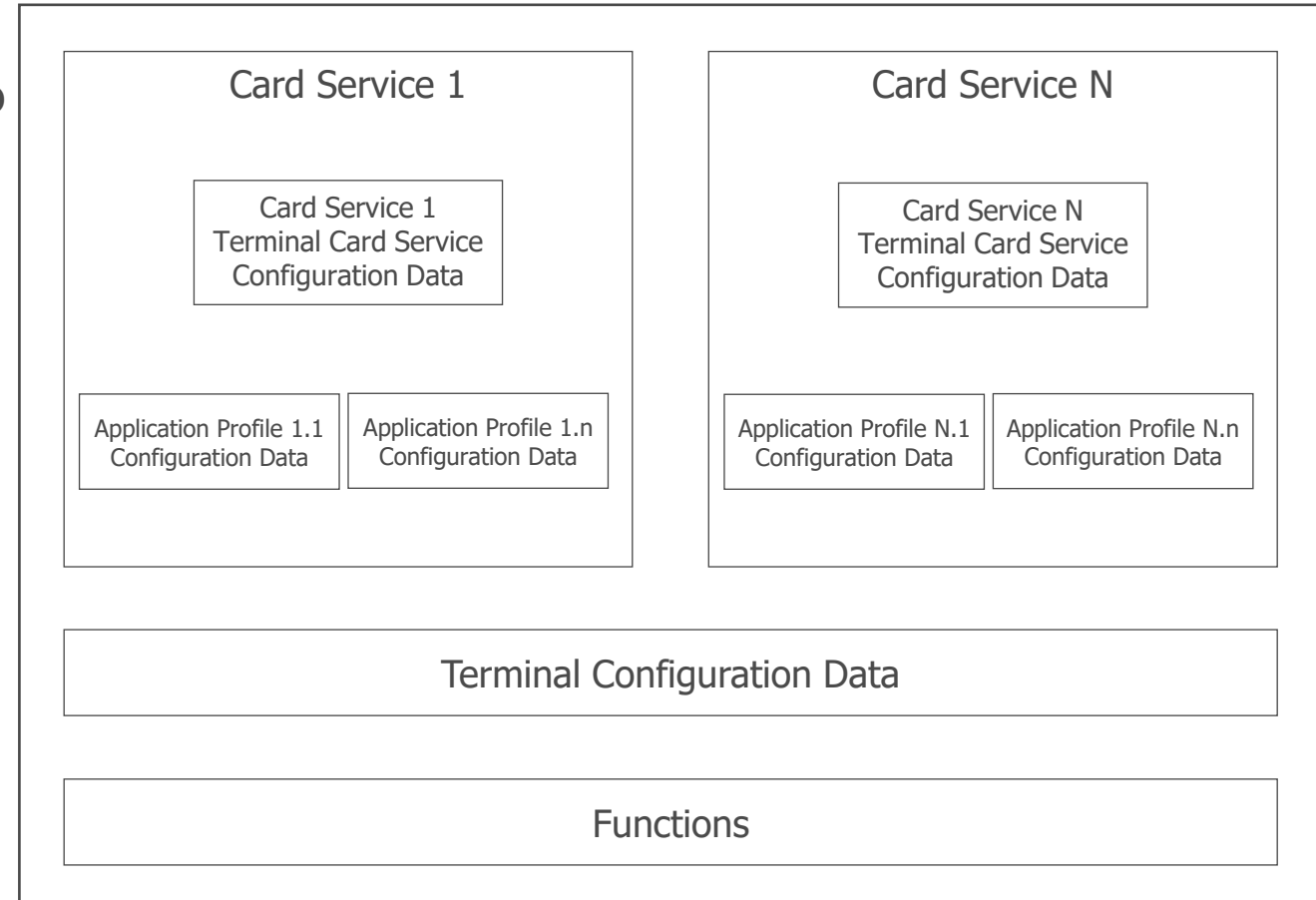
Terminal configuration data: Terminal Country Code, Terminal Type and Command Key Enter Label

Terminal Card Service configuration data: Service Settings

- Application Profile level: one instance configurable per Application Profile

Card Service independent Application Profile configuration data: Terminal Action Code (TAC) - Online, Terminal Floor Limit, and Application Profile Settings (APS)

Card Service dependent Application Profile configuration data: Cashback Maximum Amount, Pre-Authorisation Validity Number of Days, and Application Profile Settings for Cancellation (APSC)



Passion for payments

Data model

Transaction data base



- In nexo FAST, a transaction data base is used to describe the handling of **current transaction** data during financial transaction processing



- **The nexo FAST transaction data base contains**
 - └ The values of **terminal configuration data** elements that are applicable to the **current transaction**
 - └ The current values of
 - Terminal transaction data: Selected Language...
 - Kernel data: Terminal Verification Results (TVR) ...
 - Card data: Application Cryptogram...
 - Online response data: Authorisation Code...

Data model

HAP data storage and HAP search

— HAP Data Storage



- └ HAP stores the transaction data used by the nexo FAST application together with the transaction data needed by the protocol used
- └ HAP stores the financial data for capture
 - Capture by **advice** message after the transaction is finalised
 - Capture using a **batch** message

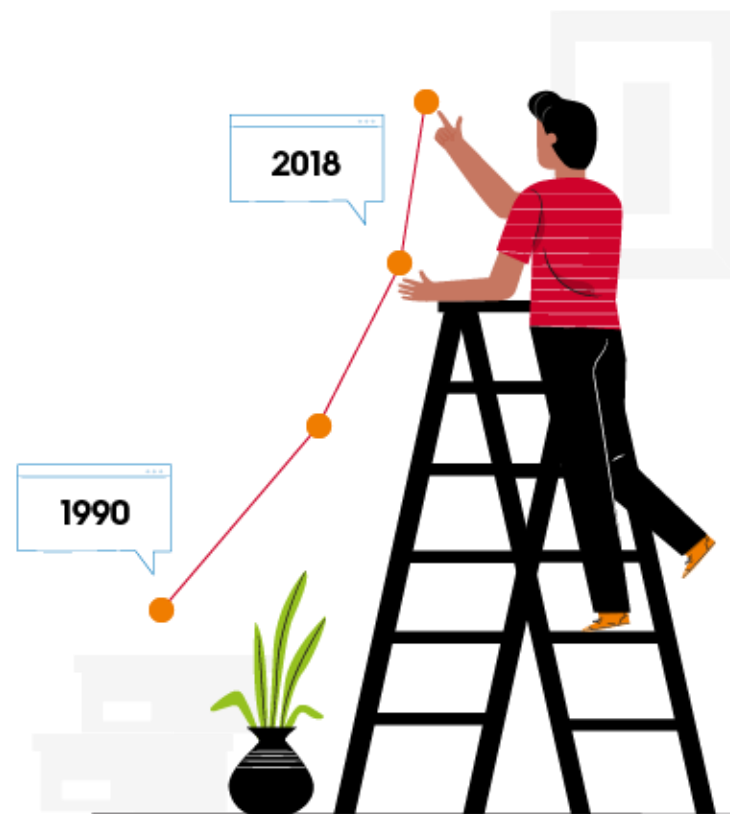
— HAP Search



- └ Search previous transactions with the same **Application Primary Account Number (PAN)** to prevent split sales according to EMV risk management
- └ Search for a transaction to be **cancelled**
- └ Search for a **Deferred Payment** transaction to perform Deferred Payment Completion
- └ Search for a **Pre-Authorised** transaction to avoid duplication of a Pre-Authorisation
- └ Search for a transaction with outcome **VOICE AUTHORISATION** to be able to close the transaction

Summary

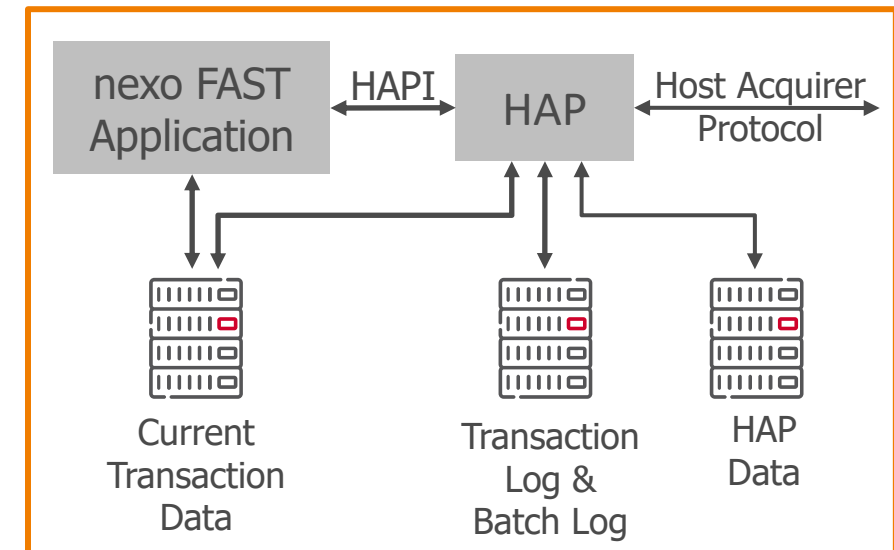
- 01 Business considerations
- 02 nexo FAST architecture
- 03 In scope
- 04 nexo FAST application structure
- 05 Data model
- 06 Interfaces to the host and to the sale system
- 07 Not described by nexo FAST



Interfaces to the host and to the sale system

Interface to the host

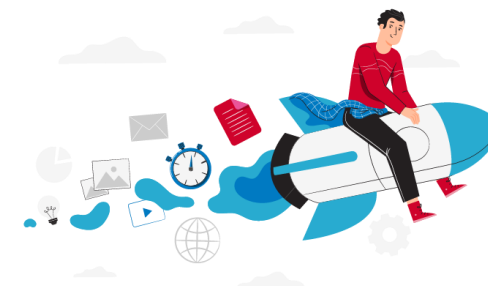
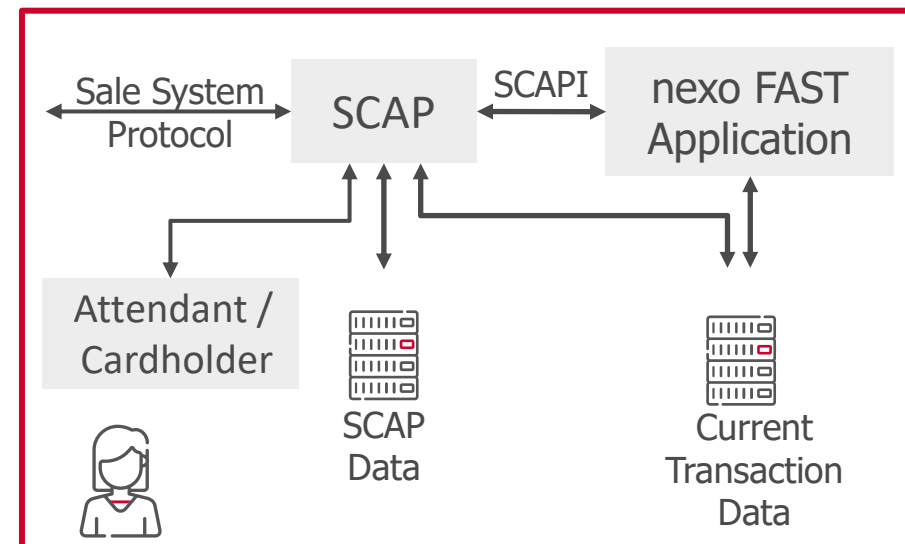
- The application responsible **for handling the acquirer protocol and the storing of transaction** data is called HAP
- The interface between the **nexo FAST** application and **HAP** is called HAPI
- Details of the data used by the HAP functions
 - └ HAP Status
 - └ HAP Online Approval Request
 - └ HAP Transaction Data Storage
 - └ HAP Transaction Finalisation
 - └ HAP Search Transactions
 - └ HAP Log Entry Data Retrieval
- Batch Processing triggering (out of scope)



Interfaces to the host and to the sale system

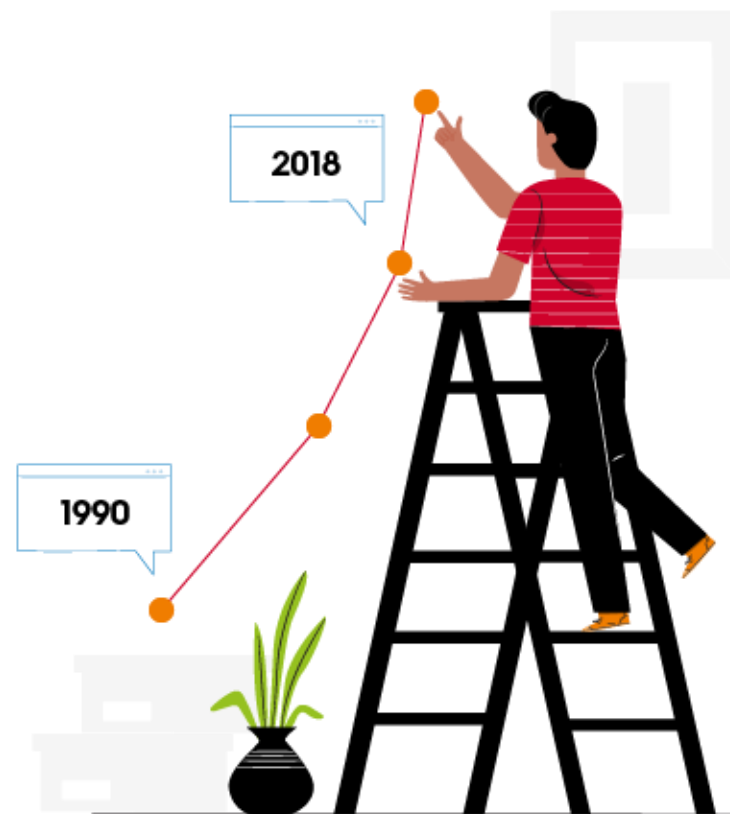
Interface to the sale system

- nexo FAST application needs to **communicate with the cardholder**
 - └ to request or to provide information on status or data of the nexo FAST application or to request the printing of receipts
- SCAP (Sale System, Cardholder and Attendant Protocol application) always **controls the communication** between the nexo FAST application and the cardholder or attendant
- SCAP provides the following functionality:
 - └ Providing information to the nexo FAST application
 - └ Forwarding information from the nexo FAST application to the cardholder, the sale system or the attendant
 - └ Forwarding information to be printed from the nexo FAST application



Summary

- 01 Business considerations
- 02 nexo FAST architecture
- 03 In scope
- 04 nexo FAST application structure
- 05 Data model
- 06 Interfaces to the host and to the sale system
- 07 **Out of scope**



Not described by nexo fast

Scope for nexo fast



Not described by nexo FAST

TMAP functionality

Terminal installation

Non-financial services

Security requirements

Specific requirements for a multi-merchant terminal

Multi-activity terminal

