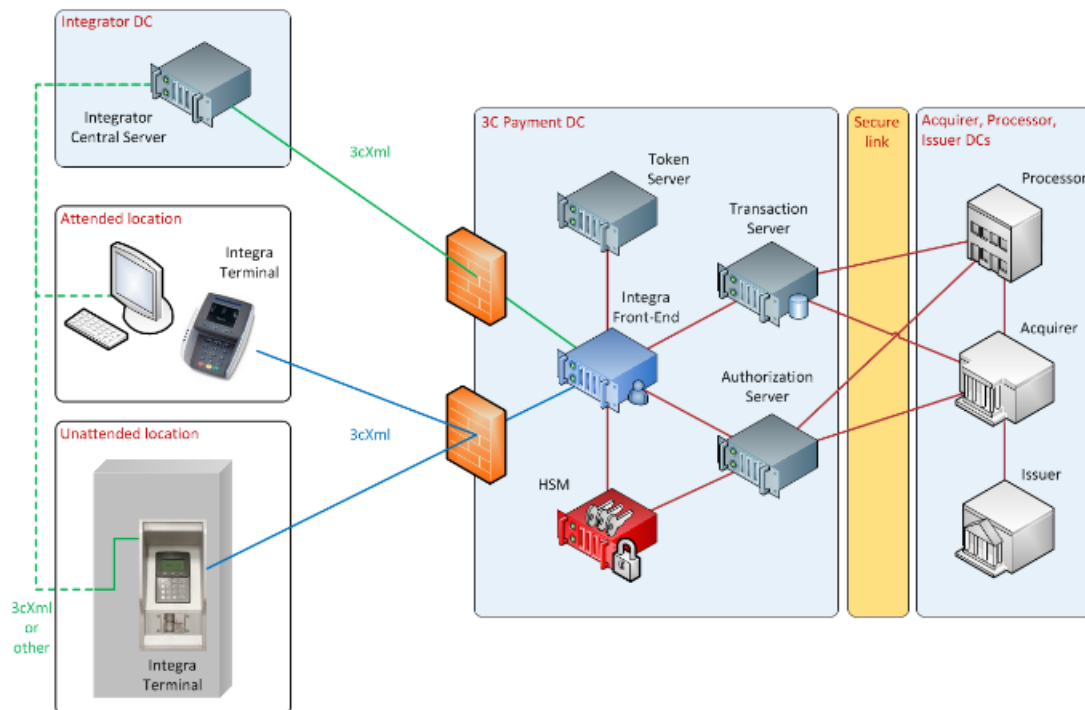


# Planet Payment terminal integrations

## 3cXml "ECR-POS" Protocol

The 3cXml protocol supports 2 means of communication:

- 1) Between the integrator and the Integra Front-End (Integra-FE).  
**NOTE:** The Integra-FE can communicate via 3cXml directly to the protocol in this case as shown in the diagram.
- 2) Between the Integrator and EMV Terminal.



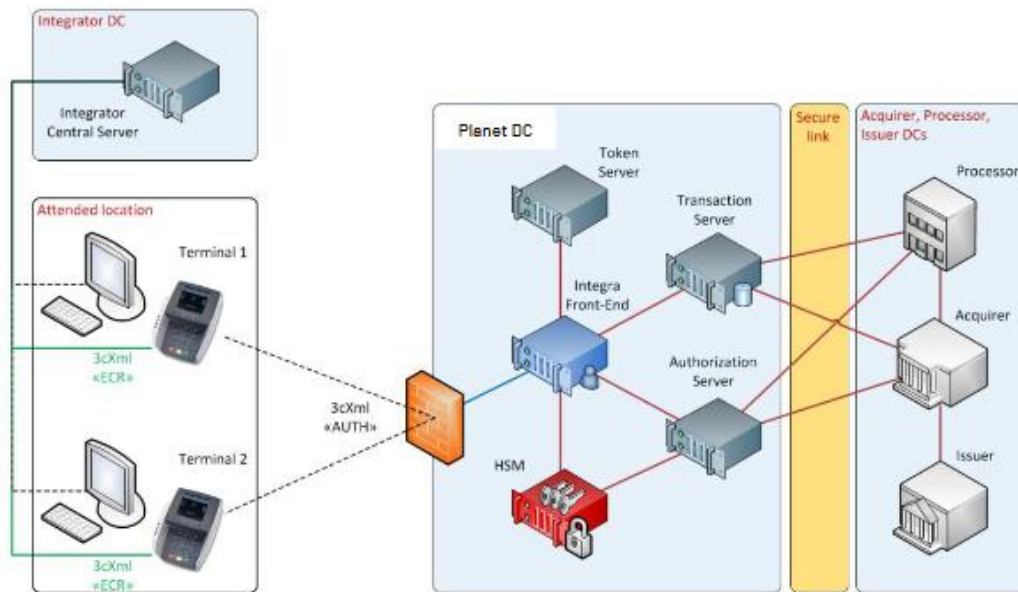
There are two main use cases for the 3cXml protocol:

**ECR:** Communication with the EMV Terminal in order to process payment operations and get the status of the terminal. Here the 3cXml is used as a terminal communication interface protocol.

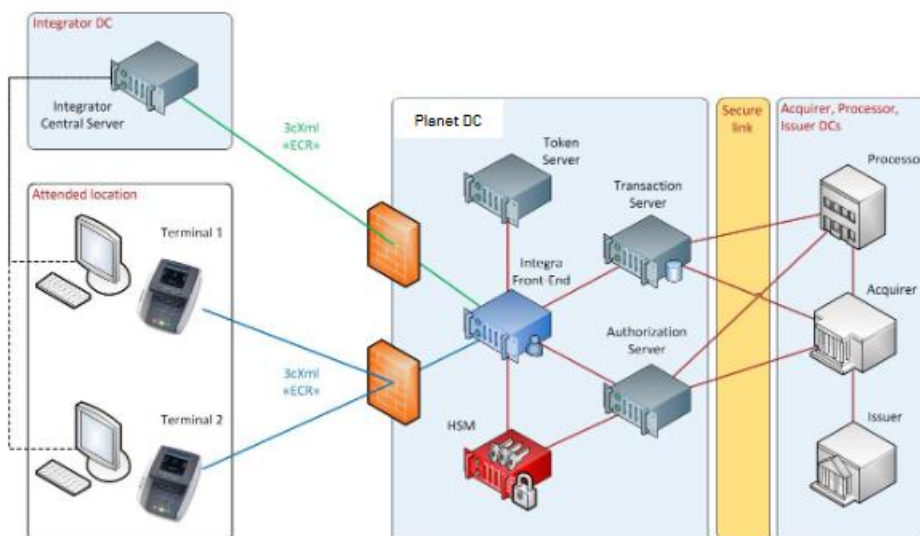
**AUTH:** The integrator controls the EMV terminal/ card reader /etc. and uses 3cXml as an authorization interface protocol, where all the card data is included in the request to further authorize at an acquirer.

There are two deployment options shown in the two diagrams below:

The first one is where the integration connects directly to the EMV Terminal, in order to start a transaction, then the EMV Terminal will attempt a 3cXml Auth request to the Planet Integra Frontend server, to process the information.



In the other use case the integration is connected to the terminals via the central Front-End server at Planet. The integrator POS Server (BC) connect to the Integra Front-End. This request needs terminal identification.



Overview of typical messages used in ECR-POS:

## Overview of typical messages used in this mode "ECR-POS"

For payment processing:

- **EftSettlementEmv - type "Sale-Terminal"**: to process a direct sale (debit) purchase transaction
- **EftSettlementEmv - type "Refund-Terminal"**: to process a direct refund (credit) transaction
- **EftSettlementEmv - type "Sale-Reversal"**: to void a previously created sale transaction
- **EftSettlementEmv - type "Refund-Reversal"**: to void a previously created refund transaction
- **Cancel**: to abort a payment request

For administration (optional):

- **ShiftClose**: to close a reconciliation shift
- **ShiftOpen**: to open a reconciliation shift
- **ShiftCloseOpen**: to close, then re-open a reconciliation shift

For terminal management (optional):

- **EftTerminalControl - command "initialize"**: to initialise an EMV terminal (ip settings, language)
- **EftTerminalStatus**: to retrieve the status of an EMV terminal

Other messages (optional):

- **CardCheckEmv**: read out card data (masked) and provide a card token
- **EftData**: receive the data of a formerly processed transaction

## Payment cycle

A payment cycle is a sequence of payment requests. These request can be initial authorisation (pre-auth), supplemental ones (top-up), and payment completions. Whenever payment cycles are used, references MUST be used. But when dealing with single requests, then they MUST NOT be used.

Reference:

RequesterTransRefNum:

This is the most common and is included in every request related to a payment. It is used to find initial and supplemental request concerning one transaction

Token:

This is built in in Integra, and can be used to reference upcoming requests linked to an initial one. The token can be transaction related or card related. The card related token are card PAN substitutes.

## All EMV Connected to Integra FE

In this mode of operation, the Integra receives EMV authorisation and settlement request without any card data. Integra communicates with the EMV terminal, and the EMV terminal provides the EMV data back to the Integra server. The authorisation and transaction is then done using the terminal and the integrator gets back limited card info, EMV information and transaction receipt.

The **<EmvTerminalId>** field is filled with a terminal ID which is configured in the Integra solution, and Integra can then read the EMV data. This scenario is called **Direct Terminal (DT)**. In this scenario the message types used are: **EftAuthorizationEmv** or **EftSettlementEmv**.

In the situation only one terminal is connected to the Integra server, no terminal ID is required. This scenario is called **Direct Unique (DU)**. The **<EmvTerminalId>** field is empty.

In all EMV messages the field **<ScenarioId>** must be filled in.

Alternatively to filling out the **EmvTerminalId**, the fields **<RequesterStationId>** and **<RequesterLocationId>** can be used, then Integra server then maps these to the correct terminal, however the **TerminalId** must still be in the system, but in the protocol it can be left empty.

## Integrator handles EMV terminal

The integrator handles the EMV terminal and includes all the EMV data which is sent to the Integra-FE server. FE is used as an authorisation host, the authorisation or transaction creation is then done by FE, and result is sent back to the integrator. This scenario is named **Data Integrator (DI)** and uses only the **EftProcessDataIntegratorEmv**.

## Integrator communicates to EMV terminal

Here the Integrator communicates directly with the terminal, which in turn communicates directly with FE, and then FE sends response to the terminal, which in turn sends back response to the Integrator.

## Printing

Some terminals support printing, and both the card holder receipt and also the merchant receipt. If the terminal does not support this, then the data can be sent back via the fields **<PrintData1>** and **<PrintData2>** the first one contains the merchant receipt and the second cardholder receipt.

## Technical details

### Integra Front-end server endpoint (used by terminals)

194.7.101.51 port 14603 ECR

194.7.101.51 port 14604 Auth

194.7.101.45 port 443 Config/Event

194.7.101.45 port 8953 FirmWare

194.7.101.45 port 29423 (DG)

### Terminal passwords

PasswordDeviceSettings	@3cPax2020@
pwdAdminMenu	123
pwdRefund	123
pwdReversal	000
pwdShift	222

pwdUserMenu	1235789
-------------	---------

Integra Front-end server requester endpoint

<https://integrator-uat.six-integra.com:14601/000137/14601>

## Config of terminal

To configure the terminal open the Integra Terminal app.

Press and hold the Planet logo in the top nav bar.

Enter the password for pwdUserMenu.

Then to do update of config do :

Config > Update > Config.

## Request

There are different kinds of request or messages that can be sent to the Integra Front-end server.

- **EftSettlementEmv** which is usually used in the context of transactions
- **CardCheckEmv** which can be used to validate, read a card or get a card token for a specific card number.
- **EftData** which sends back information about an ongoing or former transaction processed by Integra. The field **RequesterTransRefNum** is mandatory to find the info about the transaction and the **result** field contains either:
  - **<empty>**: The request is being processed
  - **A**: the referenced transaction was accepted
  - **R**: the referenced transaction was rejected
  - **E**: the referenced transaction was not found or the **RequesterTransRefNum** is not the ongoing or last transaction.
- **DCCCheck** can be used to check whether a card supports dynamic currency conversion.
- **CheckStatus** is used to check the readiness of the application.
- **ShiftOpen**: this message opens a reconciliation period
- **ShiftClose**: this message closes a reconciliation period.
- **ShiftCloseOpen**: First closes then opens reconciliation period, if it was closed it just opens.
- **Cancel**: This is a message that can cancel a request based on the **SequenceNumberToCancel** property, and can either be Accepted (A), Rejected (R) or not Found (E) which is sent back via the Result property. If it is too late to cancel a transaction then a transaction should be voided a reversal request shall be sent.
- **GetToken**: Get a token from a card PAN
- **GetPAN**: Get a PAN from a token.
- **EFTTerminalStatus**: can be used to fetch different types of info from the terminal like Display, Card, Power, Error, Info.

For EftSettlementEmv messages there are several **EftSettlementType** subtypes.

## Glossary

Integra-FE: Integra FrontEnd (Their servers).

DCC: Dynamic Currency Conversion

## Request Flow with EFT Module

Sale-Terminal and Payment.

First BC Starts a Sale with Sale-Terminal Message And await a response.

- Start Sale-Terminal (Payment)
  - Terminal receives response
    - Terminal Card OK (Result A)
    - Terminal Card Reject (Result R)
    - Terminal Cancel (Result R)
    - Cancel Message.
- Gets Response
  -
- Timeout
  -