Sale to POI Architectures and Models

Retailer Protocol Working Group

Protocol Version 3.1

Document Version 3.1

3 October 2017

Revision History

Version	Date	Object
0.90	17 Nov. 2009	Draft for external consultation
1.0	10 Oct. 2010	Version 1.0
1.9	15 Feb. 2013	Draft for external consultation
2.0	02 April 2013	Version 2.0
2.9	04 May 2015	Draft for external consultation
3.0	3 October 2016	Version 3.0
3.1	21 Novembre 2017	Version 3.1 – Architectures and models chapter in a separate document

Table of Contents

Figures

Configuration Parameters

1 Introduction

1.1 Purpose of the Document

Chapter 2, *Architectures and Models*, starts by a quick presentation of the main actors around the protocol.

The section 2.2, *Environments*, explains the environments of the systems involved directly or not, by the protocol. The section helps to understand the scope of the protocol.

The section 2.3, *Implementation History and Architecture Trends*, introduces a brief history of implementations of cashless payment in the retail market. This history is presented around the two main trends for the implementation. The section helps to understand the modeling explained in the following section.

The section 2.4, *Architecture Models*, presents the four models of architecture considered for implementing the Sale To POI protocol.

The last section of the chapter, section 2.5 *System Components Identification and Relationship*, describes on a practical way how the protocol uses these models of architecture. It first defines how the protocol identifies components of the systems. Then it presents the possible relationships (logical connections) between these components to dialog between systems and implement the communication aspect of the protocol. The section ends by the configuration parameters used to implement these models.

1.2 Revision History – Architecture and models – v3.1

Architectures and models are now in a separate document.

No change since v3.0

2 Architectures and Models

Architecture is fundamental to be successful on the design of any hardware or software product. But the design of a protocol between two systems needs to be as far as possible independent of any implementation of the systems that use it.

The purpose of this chapter is to clearly define the environment of the Sale to POI Protocol and to provide model of architectures on each side of the Sale to POI Protocol.

2.1 Actors and Roles of the Sale to POI Protocol

2.1.1 Customer

A *Customer* is someone who acquires products or services provided by the Sale System.

2.1.2 Cardholder

A Cardholder refers to a customer using a Card to pay or more generally to perform electronic transactions.

2.1.3 Cashier

A Cashier refers to the attendant of the merchant who performs electronic transactions using the Sale and the POI systems. The Cashier is accountable for the money in the cash drawer. This role is not mandatory, for instance in case of unattended sales operation. A Cashier is also called Operator in the following pages of the document.

2.1.4 Site Manager

The Site Manager is a special Cashier who is authorized to perform certain function that a standard Cashier cannot execute.

2.2 Environments

This section presents the environments of the systems involved directly or not, by the protocol. The section introduces briefly and defines these systems, to delimit the scope of the protocol.

2.2.1 Environment of the POI System

The POI System, which is responsible for the processing of electronic transactions[1], is surrounded by three other systems:

- The *Sale System*, which has in charge the management of a purchase made by a customer, which might involve a secure electronic transaction.
- The *Acquirer System*, which processes the transaction according to the rules of the card used by the cardholder for the transaction.
- The *TMS System*, which manages the hardware and software of the POI for the organisation in control of installed base of POI, maintains and downloads the configurations parameters for the processing of transactions and usable cards.

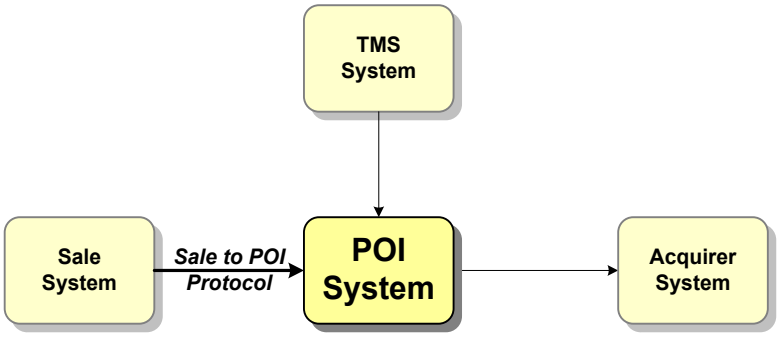


Figure 1: Environment of the POI System

In order to accept cards and realise transactions, the POI has to be initialised by the TMS System.

Processing of transactions at the Sale System, might involve processing of electronic transactions by the POI. Each electronic transaction at the POI might involve processing of a transaction on the Acquirer Host using the EPAS Acquirer Protocol, or other Application Hosts using other protocols.

During the life cycle of the POI, update or removal of hardware, software and configuration parameters are supervised by the TMS System.

[1] The POI system is not necessary responsible of processing all electronic transaction (e.g. local fleet card).

2.2.2 Retailer Environment

In the Store, the retailer's environment includes three systems:

- The *Sale System* is the entity in charge of the store, which provides all the standard functionalities of a point of sale, including the interface with the Delivery and the POI System. It can contain particular functions or services specific to the specialty or the context of the shop (for instance handling of the checkout in a supermarket, of the forecourt in a gas station, of the toll in a car park, of a vending machine…).
- The *Delivery System* has responsibility of delivering goods or services sold to the customer. It is always controlled by or included in the Sale System. This is also the case if the POI and the Delivery Systems are integrated in the same device, as in a pump or a vending machine.
- The *POI System* which, at Sale System's request, handles secure electronic transaction accomplished by a cardholder with an account identifier like a card. It has no direct relationship with the Delivery System.

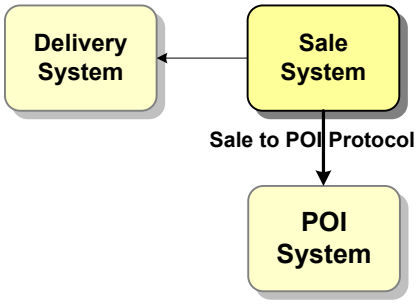


Figure 2: Retailer Environment

2.2.3 Acquirer Environment

The Acquirer is in relation with the two following systems:

- The *POI System* which performs the accepting part of electronic transactions inside the Acceptor System, which processes the sale transactions.
- The *Issuer System*, through Card Scheme Networks, to process transaction authorization and clearing. In some cases, the Acquirer and the Issuer may belong to the same entity and managed by the same host.

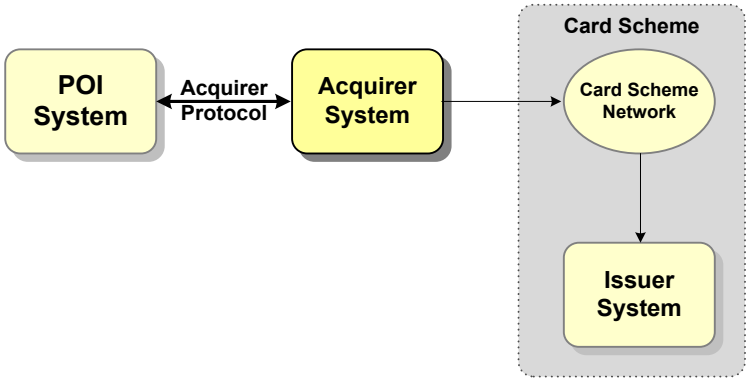


Figure 3: Acquirer Environment

Some Intermediary Agents may take place between the POI System and the Acquirer. They deliver some added value services, and can be a member located on the Acceptor domain, the Acquirer domain or both, depending on the services and responsibilities.

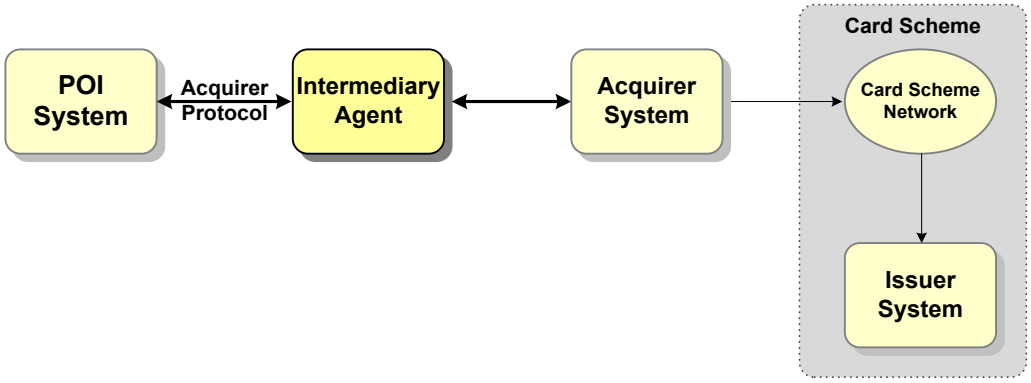


Figure 4: Intermediary Agent

2.2.4 TMS Environment

TMS services are controlled by a unique Master Terminal Manager, which has the total control of the POI System as regards to the security, applicative and maintenance services needed to manage the POI. The Master Terminal Manager can delegate part of the TMS services to one or several other Terminal Managers. Terminal Managers, including the Master can be various actors (e.g. a Manufacturer, an Acceptor, an Acquirer or a third party).

TMS services are provided by the following systems:

- A *Software Management System*, which manages the hardware related firmware, operating system and application software of the POI on behalf of the responsible entity in control of the installed base of POIs.
- A *Parameter Centre*, which stores, maintains and downloads the configuration parameters and cryptographic keys for the processing of transactions and usable cards.
- A *Maintenance and Device Management System*, for diagnosis of the POI component functionalities with limited access rights (e.g. read software versions only).

POI Systems might get the software and parameters from different Terminal Management Systems.

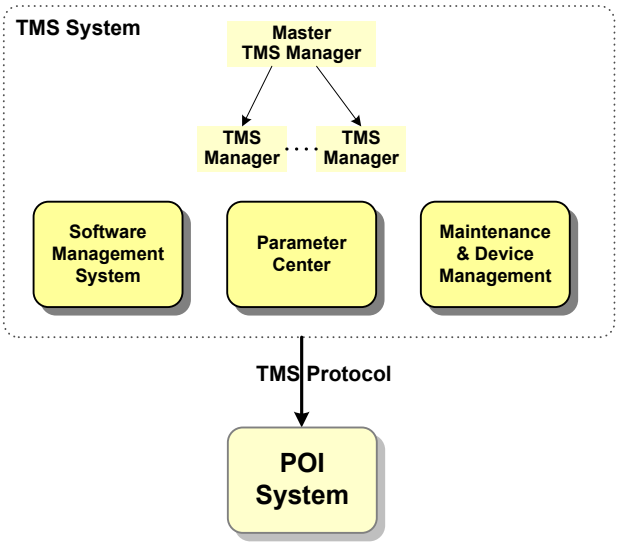


Figure 5: TMS Environment

2.3 Implementation History and Architecture Trends

Payment architectures in the retail/petrol activity are strongly correlated to the market of the sales products. Users, payment solution providers, and financial institution take care of this domain only quite recently.

Payment is only one function among others provided by a Sale software product, and whatever the architecture of the Sale System vis-à-vis POI System, payment still stays an important function of the Sale product for the management of sales amount, payment means, taxes, rebates …

2.3.1 Stand-Beside Systems

The simplest architecture to imagine is to install a standalone payment terminal in the store to be used by one or several cashiers beside the cash registers (or ECR or POS). This was the first and the most primitive POI solution used in the multilane retail, inherited from the standalone POI architecture (see Figure 6: Stand-Beside Systems-a).

To avoid re-entering of the amount and get back some information about the payment, an elementary protocol was designed to link the payment terminal to the cash register (Figure 6: Stand-Beside Systems-b).

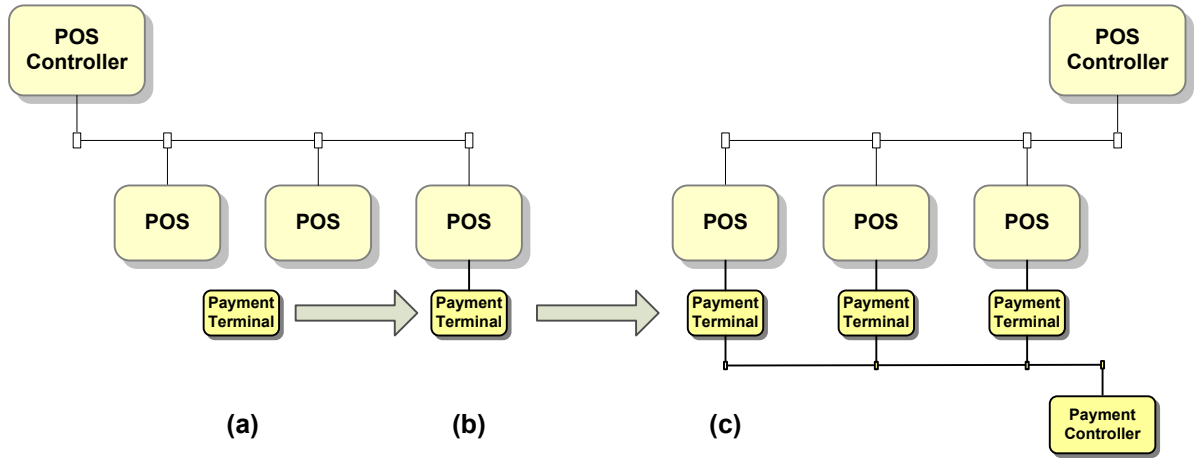


Figure 6: Stand-Beside Systems

Later the market proposed cluster architectures, made up of payment terminals linked by a primitive LAN to a gateway or a payment controller, for the communication with the Acquirer Host. Sometimes, a particular payment terminal plays the role of terminal master to contain shared resources (see Figure 6: Stand-Beside Systems-c).

To optimise payment terminal cost, receipt printer was removed and payment receipt printed by the cash register.

All these kinds of architectures belong to the Clustered POI System architecture model[2].

[2] The POS controller is the Sale Server, the POS are the Sale Terminals, the Payment Controller is the POI Server, and the Payment Terminals are the POI Terminals.

2.3.2 Integrated Systems

Shortly after the first stand beside payment solutions, Sale software products have included electronic payment software and a magnetic stripe card reader, device of the Sale Terminal. So the POS was completely autonomous to make payment initiated by magnetic cards, with the help of the POS controller which integrates the record of payment transactions, as the record of the sale transaction, and the communication software modules with the Acquirer Hosts (see Figure 7: Integrated Systems-a).

Some Sale products were designed to facilitate customisation of locally dependent functions, like tax or payment. So logically, because of complexity of the payment, raise of investment, and difficulty to be on time with the market, payment modules were more and more developed separately.

On the same time, to offer a PIN code verification or encryption, a pin-pad replace the magnetic card reader, connected to the POS with a protocol specific to the manufacturer and a module for this interface.

The POI is then composed of a pin-pad, a payment module in the POS, which handles the pin-pad, process the payment, and dialogue with a payment module in the POS controller (see Figure 7: Integrated Systems-b). To be independent of the Sale product, a payment server can be a substitute for the payment module in the POS controller (see Figure 7: Integrated Systems-c).

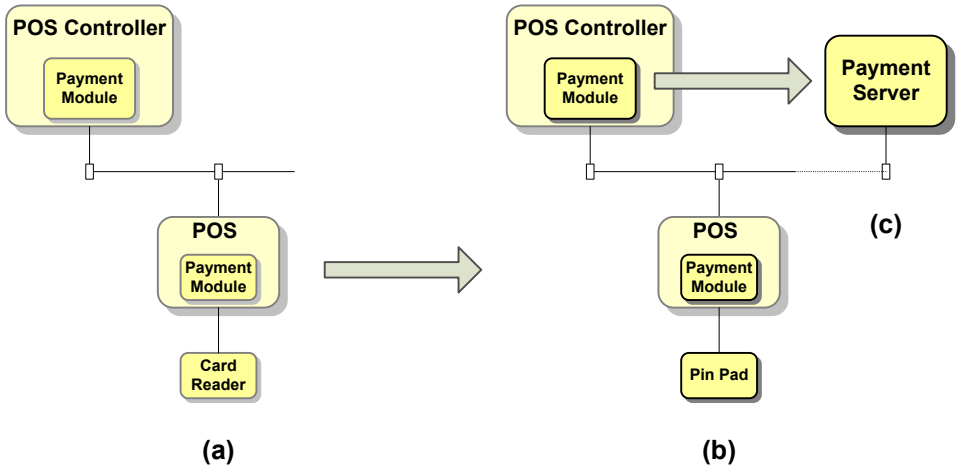


Figure 7: Integrated Systems

These architectures are a mix of Clustered POI and Distributed POI Systems models[3].

The introduction of Ethernet LAN on which Sale System and POI System can both be connected, brings in Distributed POI configurations.

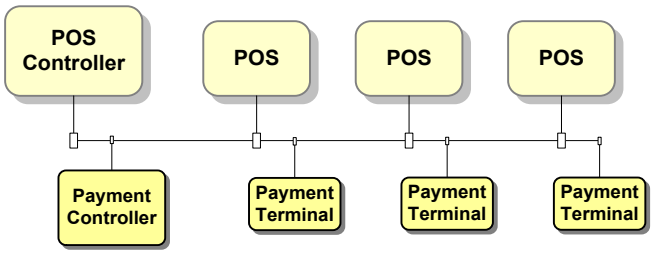


Figure 8: Sharing a Common LAN

[3] The Sale Server is the POS controller, the Sale Terminals are the POS, the POI Server is the payment module in the POS Controller or the Payment Server, and the POI Terminals are the payment module in the POS and the Card Reader or the PIN pad, or the Payment Terminals

2.4 Architecture Models

This section describes the four models of architecture considered to implement the Sale To POI protocol.

2.4.1 Sale System Architecture

The Sale System, as seen by the Sale to POI Protocol's interface includes[4]:

- A set of *Sale Terminals*, which could be physical terminals (Electronic Cash Register), or logical terminals if there is no cashier (unattended transaction). The Sale Terminal manages the whole sale transaction, choice and identification of the purchased items, operations necessary to complete the transaction (payment, loyalty, and other secure operations), the interface with all the participants to transaction.
- A *Sale Server*, which processes all the global sale functions, in particular the recording of the transaction and the database of the system.

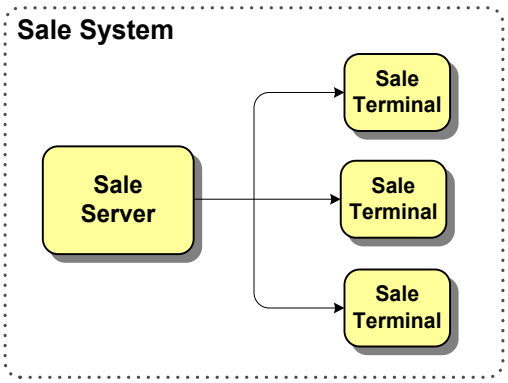


Figure 9: Global Sale System Architecture

The Sale Server and the Sale Terminals are usually joined together with a LAN, allowing communication between a terminal and the server. Traditionally, there are no communications between Sale Terminals.

The Sale Terminal handles all the physical devices required during the processing of the transaction, and delegates to the Sale Server, all the functions consuming substantial resources or functions common to the whole system.

In some contexts, i.e. if there is only one Sale Terminal, the Sale Server can be located in a single Sale Terminal.

[4] Some examples of real and typical architecture of Sale Systems are described in the Examples of *Architectures*.

2.4.2 POI System Architecture

The POI System, as seen by the Retailer Protocol's interface includes[5]:

- A set of *POI Terminals*, a physical or logical terminal, or a pin-pad which contains at the minimum the security services, the low level services, and the user interfaces. It can also include application to process part of the transaction in cooperation with the POI Server, and some synchronisation tasks with the Sale System. In some cases, POI Terminal uses a component of the Sale System as a user interface.
- A *POI Server*, which groups or contains the global functions, as the recording of the transaction or the communication with the Acquirer Hosts or the TMS Server.

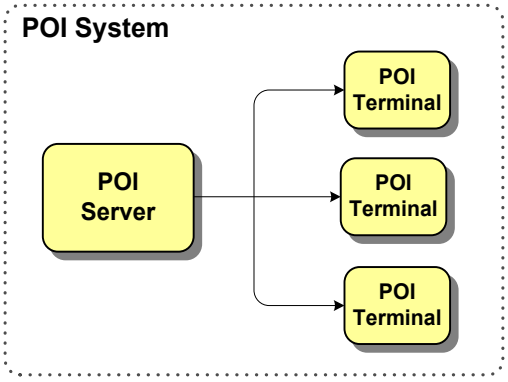


Figure 10: Global POI System Architectures

We can distinguish four classes of POI Systems, which characterize the various POI architectures that are implemented in real systems:

The *Standalone POI*, where the hardware and software components of the POI are concentrated in a single unit (the POI Terminal), without any correlation with the Sale System other than a user interface for the cashier.

The *Connected POI*, where the hardware and software components of the POI are concentrated in a single unit, connected to the Sale System through a retailer's application protocol.

The *Clustered POI System*, where the hardware and software components of the POI are split among several units (POI Terminals and POI Server), and are connected to the Sale System as a device of the Sale Terminal.

The *Distributed POI System*, where the hardware and software components of the POI are distributed among several units (POI Terminals and POI Server), and form a Payment Solution separated from the Sale System (Sale Terminal, Sale Server), with clear interface and responsibilities.

[5] Some examples of real and typical architectures of POI Systems are described in the Examples of Architectures, and can be read before this section.

2.4.2.1 Standalone POI

The Standalone POI architecture, which is reduced to a single POI Terminal, possesses the following characteristics:

- The terminal encloses all the hardware and software components required to process card transactions and manage the POI Terminal.
- The terminal provides user interface to the cardholder for the transactions, and user interface to the cashier for the transactions and the administration of the terminal. There is no communication with the Sale System.
- The terminal manages the communication protocols with the Acquirer Hosts and the TMS Servers.

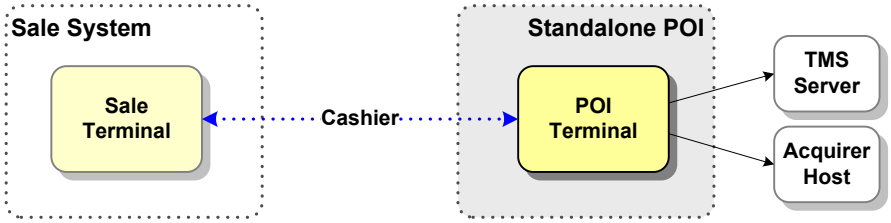


Figure 11: Standalone POI System

By definition, this type of architecture does not use any retailer's application protocol with the Sale System, which is usually composed of a unique Sale Terminal.

2.4.2.2 Connected POI

The Connected POI is a simple configuration where a POI Terminal is connected to a Sale Terminal, without any relation between POI Terminals.

The Connected POI architecture, which can also be reduced to a single POI Terminal, assumes the following characteristics:

- The terminal encloses all the hardware and software components required to process card transactions and manage the POI Terminal.
- The terminal provides user interface to the cardholder for the transactions, and user interface to the cashier for the transactions and the administration of the terminal.
- The communication between the Sale System and the POI Terminal allows performing transactions, printing receipt and status display on the Sale Terminal.
- The terminal manages the communication protocols with the Acquirer Hosts and the TMS Server.

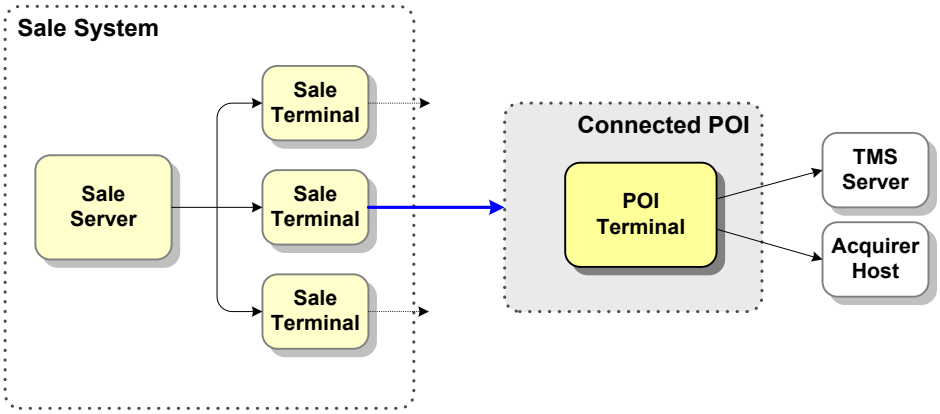


Figure 12: Connected POI Terminal

The Connected POI architecture can use the Sale to POI Protocol as the application protocol between the Sale Terminal and the corresponding POI Terminal. From the view of the POI System by the Sale System, there are as many POI Systems as POI Terminals, since each POI Terminal does not know the presence of other POI Terminals.

In the following specifications, this architecture model will be associated to the following model, the *Clustered POI System*, with the specific case of only one POI Terminal.

2.4.2.3 Clustered POI System

The Clustered POI architecture is characterised by the following features:

- Each Sale Terminal is physically linked to its respective POI terminal, involving an implicit addressing between the two Terminals. The Sale Terminal dialogues with an interface dedicated to the device or a POI software component implemented in the Sale Terminal to interface this POI Terminal.
- Some devices of the Sale Terminal (printer, display, scanner…) are used by the POI System during the transaction.
- Some devices of the POI Terminal (card reader, display) can be used by the Sale System to achieve sale function.
- Global functions as transaction recording and communication management with the Acquirer Hosts or the TMS Server are carried out by the POI Server inside a common platform. These global functions can be implemented inside the Sale Server, a particular Sale Terminal, or a dedicated platform.
- Viewed from the Sale system, each POI Terminal is considered as a peripheral device of the Sale Terminal, in the same respect as a magnetic stripe card reader, a scanner or a printer.

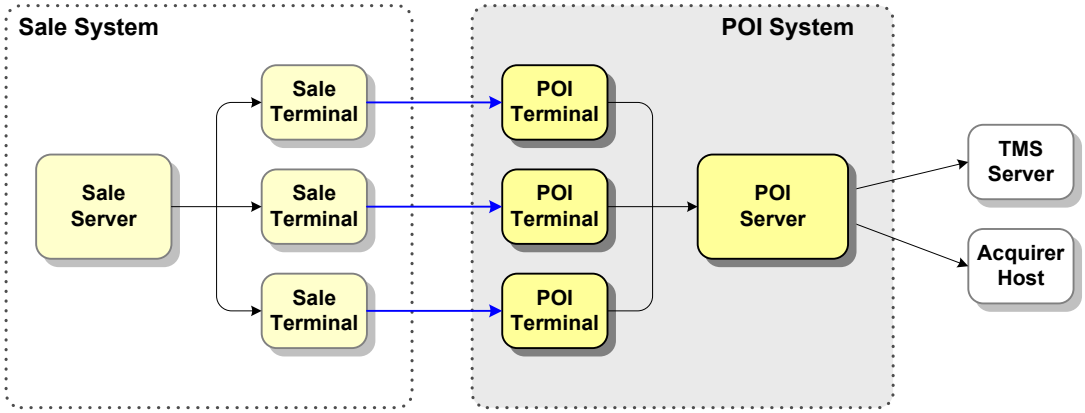


Figure 13: Clustered POI System

Implementation of the POI Server (which can only handle communication device interfaces, or manage global POI functions and located in the Sale System), depends on the configuration of the shop, number of Sale Terminals, volume of transaction, and the level of integration inside the Sale System.

In this architecture, all the global services of the Sale to POI Protocol could go through the link between the Sale Terminal and the corresponding POI Terminal, or between the Sale Server and the POI Server.

For the Clustered POI System, an alternative to the POI Server, as described in the previous figure, is the deportation of the POI Server in a central location in order to keep in the store only the POI Terminals and a communication gateway.

This POI Server manages on a *Central Site* several POI Systems where the POI Terminals are on *Local Sites* in the shops.

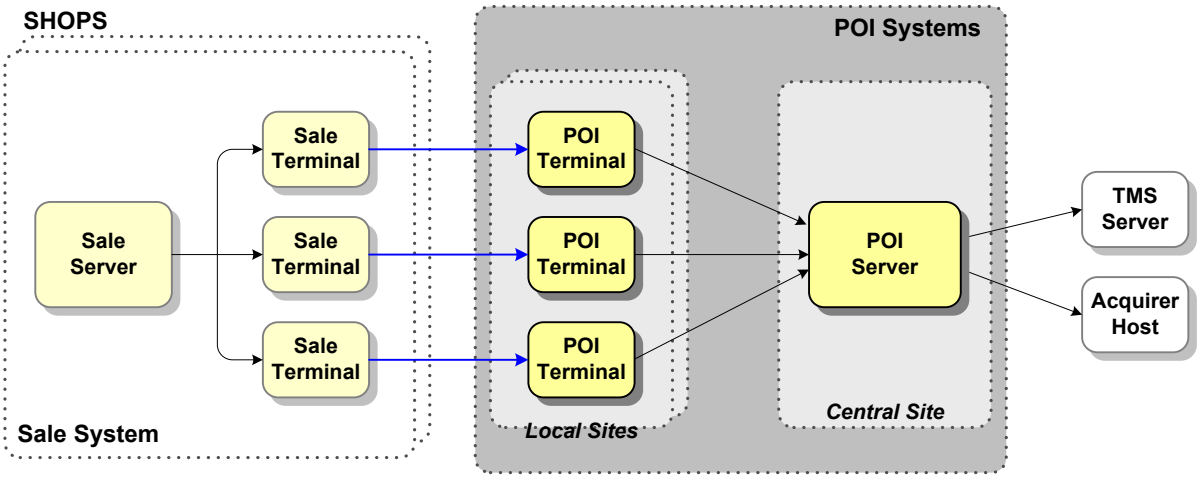


Figure 14: Clustered POI System with a Central Site

2.4.2.4 Distributed POI System

The Distributed POI architecture is characterised by the following features:

- The Sale System and the Distributed POI Systems dialogue with a standard interface in a system-to-system relationship.
- Each Sale Terminal is logically linked to its respective POI terminal, and dialogue with the Distributed POI System via the POI server. There is no implicit addressing between the Terminals, or a POI software component in the Sale Terminal to implement an interface with the POI.
- As the Clustered POI, some devices of the Sale Terminal (printer, display, scanner…) are used by the POI System during the transaction. This data flow carrying out lower level service of the standard interface is clearly identified with specific rules.
- A specific data flow of the standard interface allows the use of some devices of the POI Terminal (card reader, display) by the Sale System to achieve sale function.
- This architecture allows some flexibility for the distribution of functions between the POI Server and the POI Terminal, according to the requirements defined in this document (global functions have to be carried out by the POI Server, secure and stable functions has to be located in the POI Terminal…).

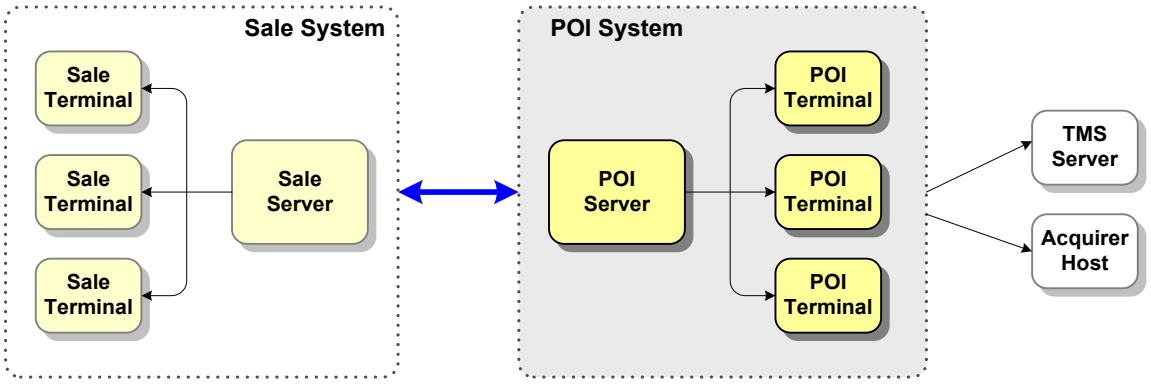


Figure 15: Distributed POI System

The standard interface is governed by some rules implied by the architecture of the POI System:

- Whatever the requester and the provider are (Server or Terminal), this is always a system-to-system interface. This is essentially a matter of implementation of the interface or the protocol.
- Visibility of the POI System: the Sale System can address a particular POI Terminal or the POI Server (i.e. the POI System) if no POI Terminal is involved in the requested service.
- Visibility of the Sale System: the POI System can address a particular Sale Terminal or the Sale Server (i.e. the Sale System) if no Sale Terminal is involved in the requested service.

2.5 System Components Identification and Relationship

The section describes in a practical way how the protocol uses the models of architecture defined in the previous section.

It first defines how the protocol identifies components of the systems.

Then it presents the possible relationships (logical connections) between these components to dialog between systems and implement the communication aspect of the protocol.

The section ends by the configuration parameters used to implement these models.

2.5.1 Identification of Systems and Components

The Sale to POI protocol is designed to interface a unique Sale System to a unique POI System.

In some particular contexts, architecture can include several POI Systems or even several Sale Systems. In this case, the various systems are considered independents, and the protocol allows the management of different systems by their system identification.

As we have seen in the various models of Sale System and POI System, on the protocol point of view, a system is composed of:

1. A *Server*, which covers hardware or software module, and assures the functions global to the System.
2. One or several *Terminals* which takes in charge part of the processing of a transaction, either because the transaction involves some hardware device, either because several transactions have to be processed in parallel and the Terminal stands for the transaction context.
3. One or several *Devices* managed and attached to each *Terminal*. These Devices are used either with a logical identification (e.g. for a Display request), either implicitly by a particular device command (e.g. CardRead request).

As already mentioned, all these system components are a logical view of the Sale and the POI Systems through the Sale to POI protocol interface. The concrete POI Server could be in a wide range of implementations, from a dedicated Point of Sale terminal in the store, to a specific software module on a Sale Terminal.

All Terminals of a given Sale or POI System could have different sets of Devices. The list of available Devices on a Sale Terminal and on a POI Terminal is exchanged at the login step between these two terminals.

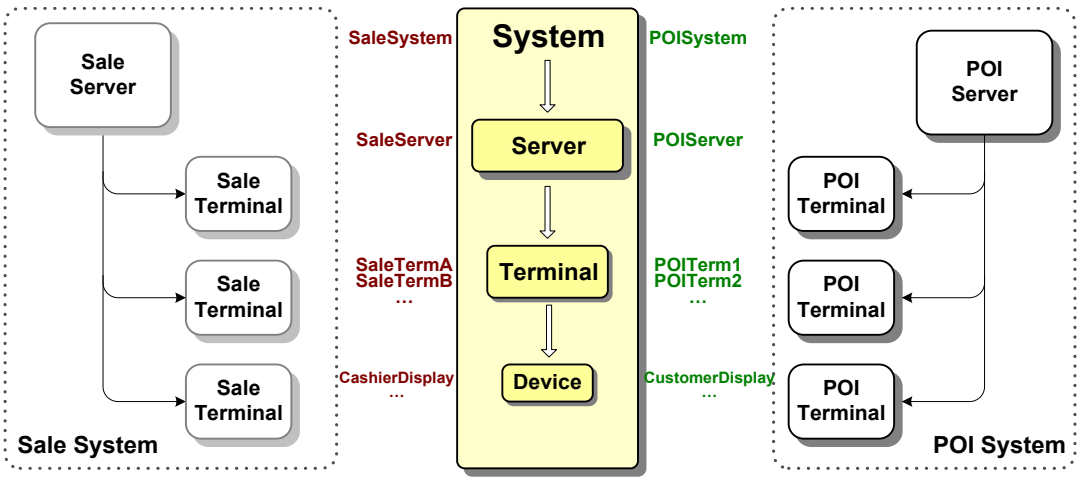


Figure 16: Identification of System Components

These components of a System are identified inside the protocol in a hierarchical way:

1. The System has an identifier, root of its components. This identifier prefixes all components identifiers of the System, and allows the recognition of a System when several Systems are managed in parallel.
2. The Server has a unique identifier, which is prefixed by the System identifier. This identifier could be used in the protocol when a message does not involve a Terminal, or is a dialogue from System to System.
3. Every Terminal has a unique logical identifier, also prefixed by the System identifier. This identifier is used in the protocol for Terminal to Terminal messages.
4. Every Device on a Terminal is identified by a logical identifier in the Terminal. This is a logical device name used in the protocol for Terminal to Terminal messages.

For the remainder of the document, when components' identifiers are mentioned or inside examples of exchanges, the following identifiers values are used:

1. The identifier of the Sale System is "*SaleSystem*", and the identifier of the POI System is "*POISys*tem",
2. The identifier of the Sale Server is "*SaleServer*" (or just "*Srv*"), and the identifier of the POI Server is "*POIServer*" (or just "*srv*"),
3. Identifiers of the Sale Terminals are "*SaleTermA*", "*SaleTermB*",... (or simply "*A*", "*B*",...), and the identifier of the POI Terminals is "*POITerm1*", "*POITerm2*",... (or simply "*1*", "*2*",...),
4. Identifier of the Device's is defined by the label of the data element (e.g. "*CashierDisplay*"[6])

[6] "CashierDisplay" identifies the logical device where information is displayed for the Cashier. This logical device is generally managed by the Sale System, but could also be managed by the POI System.

2.5.2 Multiple Sale or POI Systems

In some particular context, architectures can include several POI Systems or even several Sale Systems. For instance, in a petrol station, the Sale System can manage one POI System for payment indoor and another POI System for the outdoor payment.

In this case:

- Either the various POI (or Sale) Systems are independents or cannot share any information, or cannot use common communication channels, and there are two Sale to POI protocols management. These two protocols cannot mix messages because of the different identification of each POI (or Sale) System.
- Either the various POI (or Sale) Systems can share the same application protocol by any possible implementation, there is only one POI (or Sale) System using the same system identification.

To follow the example of a petrol station, with two different POI Systems for indoor and outdoor payment, the Figure 17: Example of Multiple POI Systems shows the case of two independent POI Systems with two different assigned system identifications (*IndoorPOI* and *OutdoorPOI*).

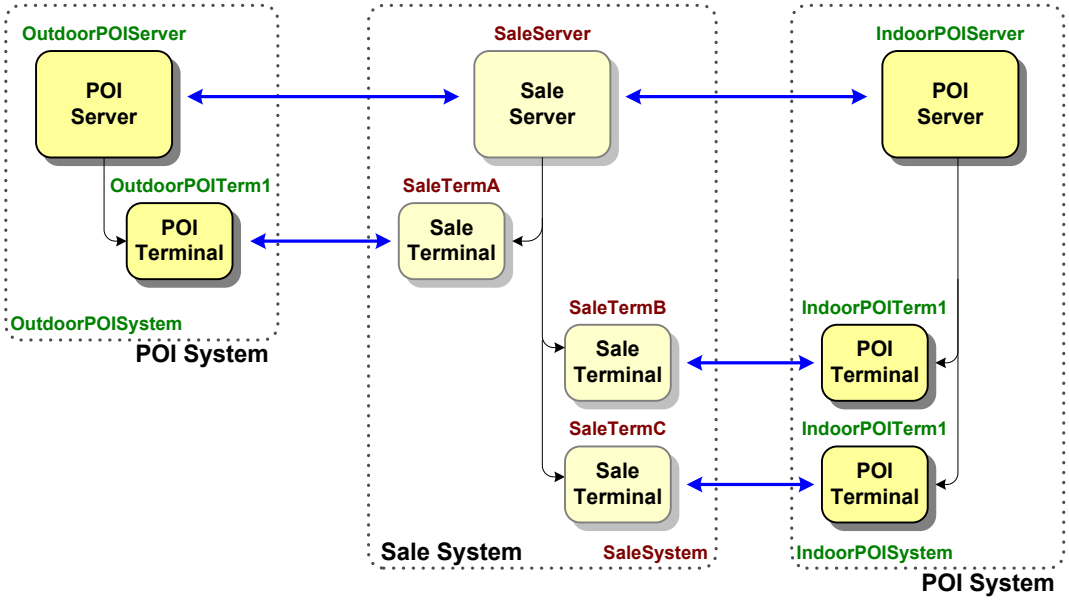


Figure 17: Example of Multiple POI Systems

2.5.3 Logical Connections Between System Components

The Sale to POI protocol exchanges messages between the following pairs of components which are in relation:

- The Sale Server and the POI Server, when the processing of the request does not involve the cooperation of any Terminal,
- The Sale Terminal and the POI Terminal which are used together to make a transaction, with the help or not of the Servers.

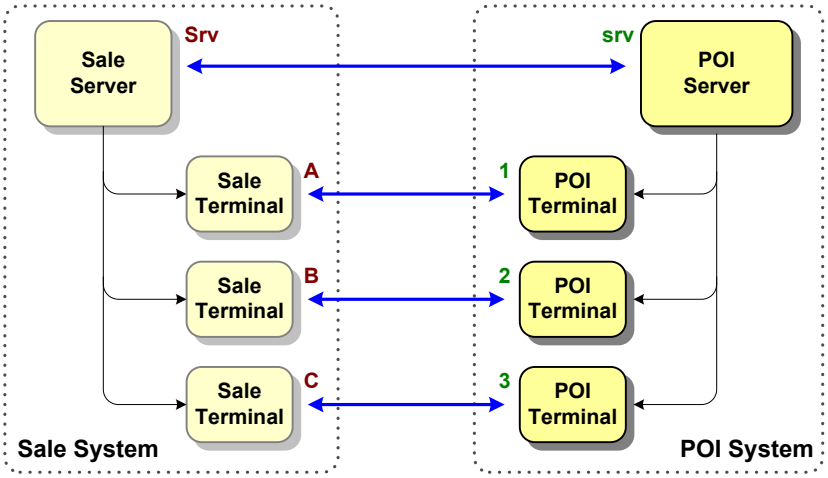
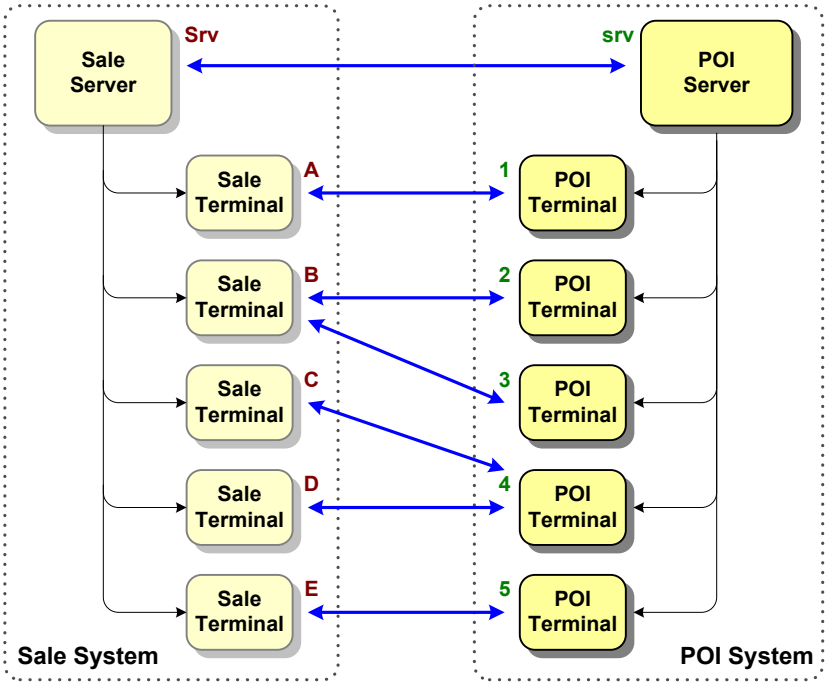


Figure 18: Relationships between System Components

The Figure 18: Relationships between System Components above presents an example of these relationships between components of the POI and Sale Systems.

However, a one-to-one relationship between Sale Terminal and POI Terminal is not required. A Sale Terminal might handle several POI Terminals, and a POI Terminal may be shared between several Sale Terminals as in the example below. In this example, the Sale Terminal B uses the POI Terminals 2 and 3, and the POI Terminal 4 is shared by the Sale Terminals C and D.



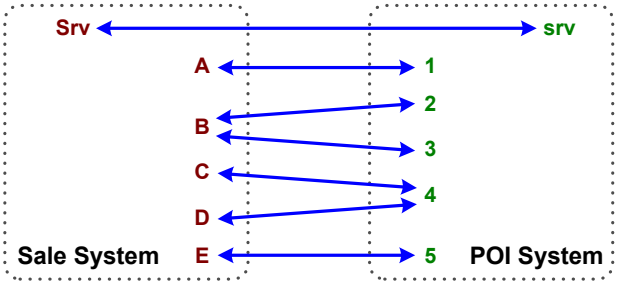


Figure 19: Specific Relationships

These relationships between system components are the channels that support exchanges of messages between the Sale System and POI Systems. They are *Logical Connections* between system components, supported by *Physical Connection* (i.e. transport connections) between the two systems.

Depending on the architecture of the Sale and the POI systems, these related transport connections are established directly between these two components, or through another component.

The two following sections present the various cases of these direct or indirect Logical Connections between or through Terminals and Servers of the Sale and POI Systems.

2.5.4 Logical Connections Between Terminals

A Sale Terminal and a POI Terminal are in relation in the Sale to POI protocol when:

- A transaction is realised in real time on the Sale Terminal, which asks to a particular POI Terminal to process part of the transaction.
- The Sale Terminal needs to use a device or a low-level service of particular POI Terminal.
- The POI Terminal needs to use a device or a low-level service of particular Sale Terminal.

This section details the various transport connections which support the Logical Connections between a Sale and a POI Terminal:

1. The case where all the POI Terminals are independents (i.e. there is no global POI modules, so no POI Server) described in the following chapter *Connected POI Architecture*.
2. The case of the Terminal to Terminal links, where each POI Terminal is connected to a Sale Terminal described in the following chapter *Terminal to Terminal Transport Connections*.
3. The case of the Terminal to Server links, where all the connections to the POI Terminals are carried through the POI Server described in the following chapter *Terminal to Server Transport Connections*.
4. The case of the Server to Server links, where all the connections from the Sale Terminals to the POI Terminals are carried from the Sale Server to the POI Server described in the following chapter *Server to Server Transport Connections*.
5. The case of the Server to Terminal links, where all the connections from the Sale Terminals are carried through the Sale Server described in the following chapter *Server to Terminal Transport Connections*.

2.5.4.1 Connected POI Architecture

The Connected POI is the architecture where all the POI Terminals are independents, without POI Server or global processing. In this case, there is a dedicated link for each pair of Sale and POI Terminals.

On the figure below, links supported by one or several transport connections, are represented by a continuous blue line, and the arrow gives the direction of the association initiation.

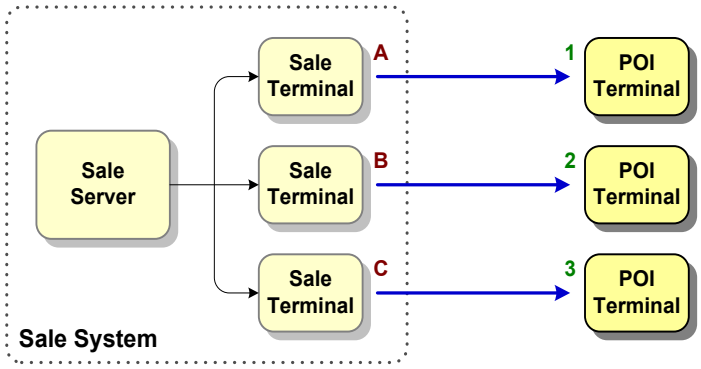


Figure 20: Connected POI Terminal Architecture

2.5.4.2 Terminal to Terminal Transport Connections

This is the typical case of the Clustered POI configuration. Each POI Terminal is connected to a Sale Terminal. In this case, there is also a dedicated link supported by one or several transport connections for each pair of Sale and POI Terminals.

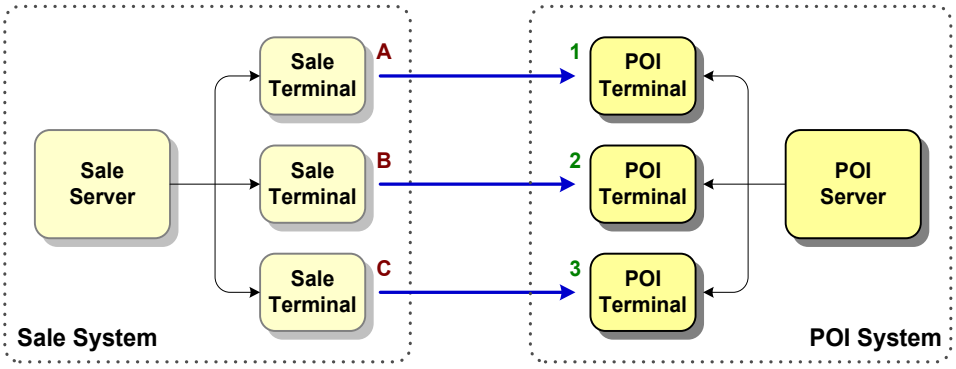


Figure 21: Terminal to Terminal Links

2.5.4.3 Terminal to Server Transport Connections

This is a typical case of the Distributed POI configuration. The POI Server manages globally the interface with the Sale System.

To be in relation with a POI Terminal, a Sale Terminal opens a connection to the POI Server, and will send inside the application messages the logical identification of the targeted POI Terminal.

As the Sale to POI protocol define interface between systems only, the way the POI Server reaches the POI Terminal is POI System implementation dependant (they are denoted by dotted lines between the POI Server and the POI Terminal).

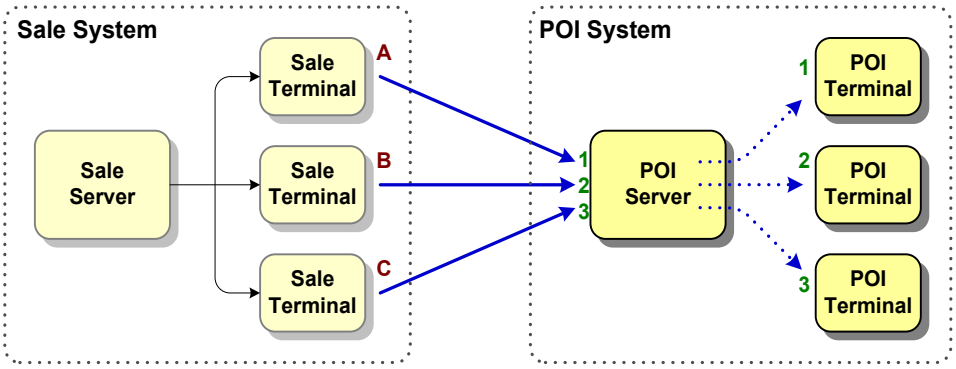


Figure 22: Terminal to Server Links

2.5.4.4 Server to Server Transport Connections

In this case, the Sale Server and the POI Server manage on each side, the Sale to POI interface.

Every transport connection between the Sale Server and the POI Server could be dedicated to a logical link between a Sale Terminal and a POI Terminal. Like the previous case, links and connections management between the Server and the Terminals of the same System are implementation dependant, and have no influence to the Sale to POI protocol (they are also symbolised by dotted lines between the POI Server and the POI Terminal).

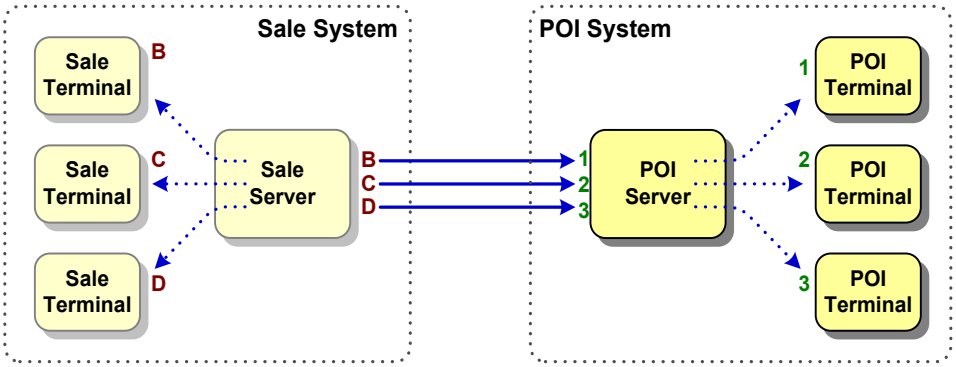


Figure 23: Server to Server Links

2.5.4.5 Server to Terminal Transport Connections

This is the last case, where the Sale Server manages the interface to the POI System directly with the POI Terminals.

There is a transport connection between the Sale Server and each POI Terminals. Once again, the links and connections management between the Sale Server and the Sale Terminals are implementation dependant.

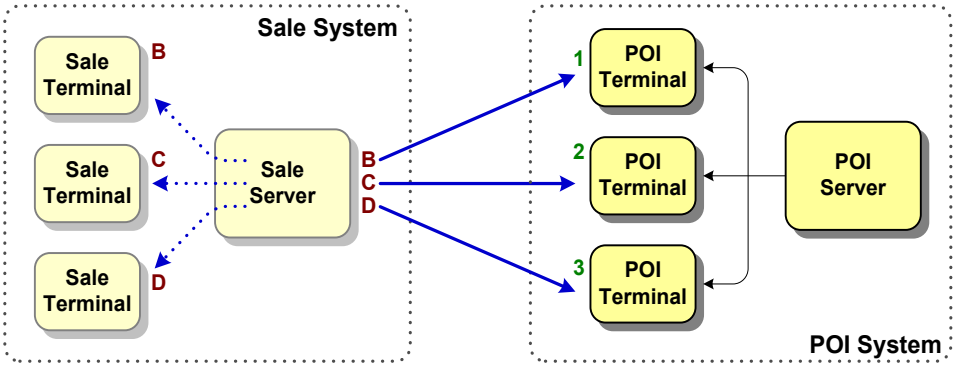


Figure 24: Server to Terminal Links

2.5.5 Logical Connections Between Servers

The Sale Server and the POI Server are in relation in the Sale to POI protocol when a transaction is global to the Systems and does not involve particular Terminal or device.

This section details the various types of associations between the Sale Server and the POI Server:

1. The case of the *Connected POI Architecture*, where all the POI Terminals are independent.
2. The cases of *Direct Server to Server Logical Connections*, whatever the connections from the Sale Terminal to the POI Terminal are carried through.
3. The cases of *Indirect Server to Server Logical Connections*, where the Server to Server connection goes through one or several Terminals.

2.5.5.1 Connected POI Architecture

In the Connected POI architecture, all the POI Terminals are independent without POI Server. So the dedicated link between Sale and POI Terminals conveys the information from both the Sale Terminal and Sale Server. A POI Terminal is like a whole POI System and is then considered by the Sale System as a whole system including a Terminal and a Server.

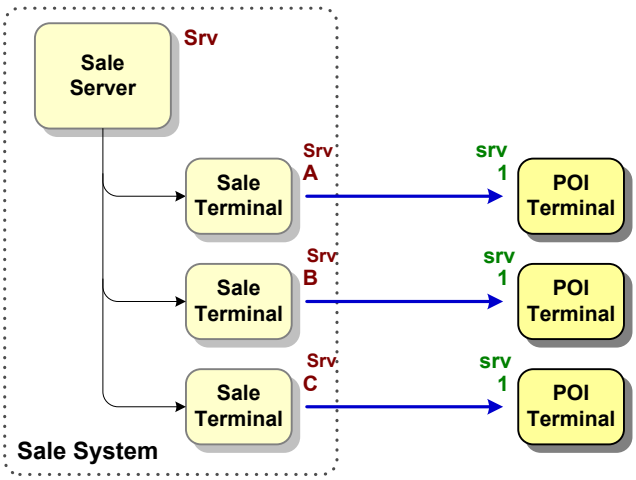


Figure 25: Connected POI Architecture

2.5.5.2 Direct Server to Server Logical Connections

In this case, there is a direct link supported by one or several transport connections between the Sale Server and the POI Server that carries commands for global services.

This type of Server to Server dialogue can be supported by various system architectures of the Sale and the POI Systems, as those we just describe for the links between Terminals.

The first situation is the Clustered POI System architecture where the POI Server is not reachable by the Sale Terminal, but it might be available by the Sale Server.

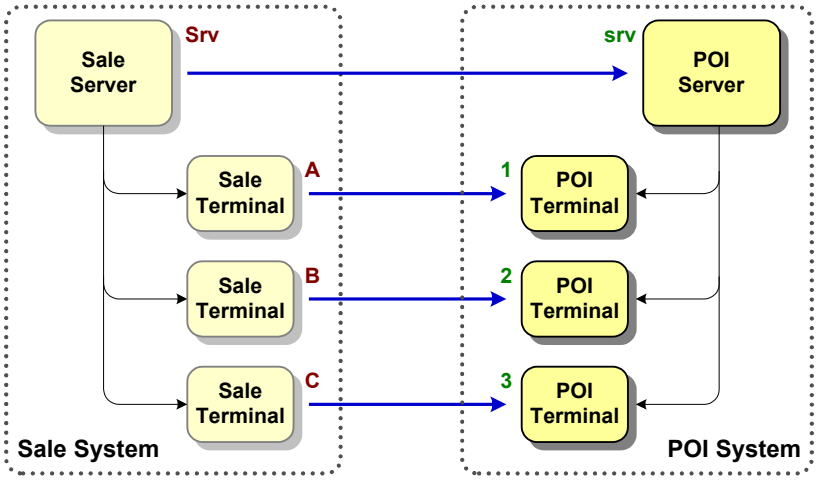


Figure 26: Server to Server Link for Clustered POI

Another case is when the POI Server manages completely the Sale to POI interface. The POI Server receives in addition the Sale Server connections, as presented in the figure below.

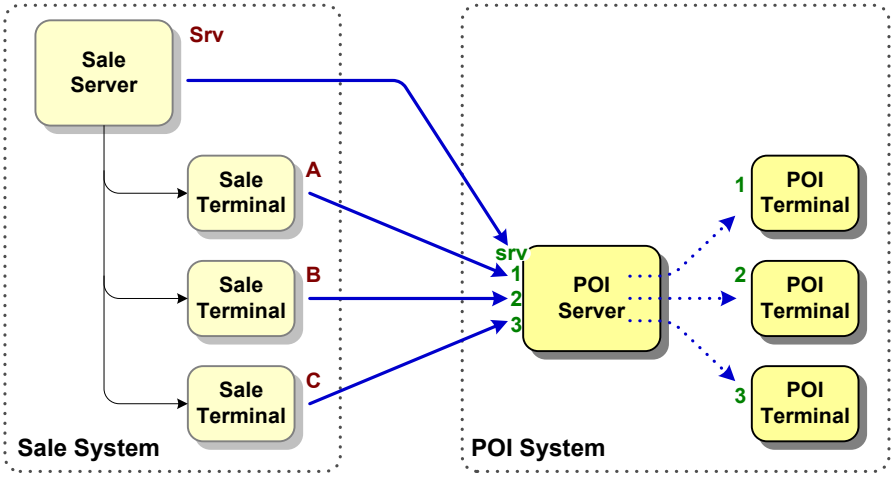


Figure 27: Server to Server Link, Terminal to Server case

An even more manifest case is when Servers at each side manage the Sale to POI interface. All the Terminals and Server transport connections are directly between the two Servers.

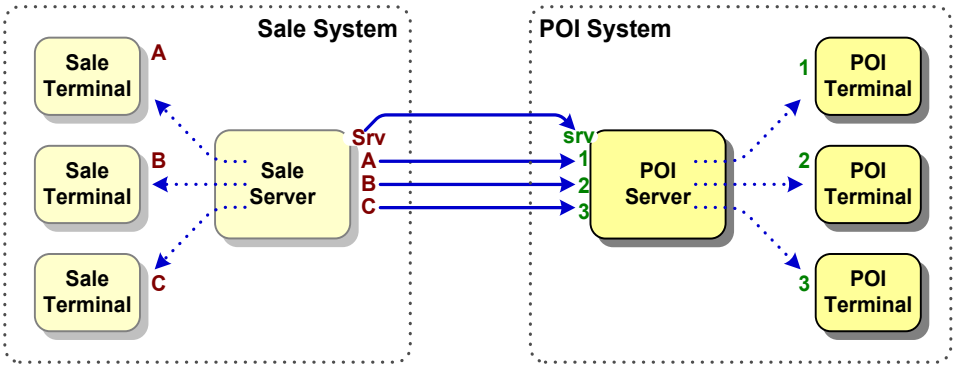


Figure 28: Server to Server Link, Server to Server case

A usual case is when the Sale Server manages completely the Sale to POI interface. The Sale Server opens in addition a connection to the POI Server.

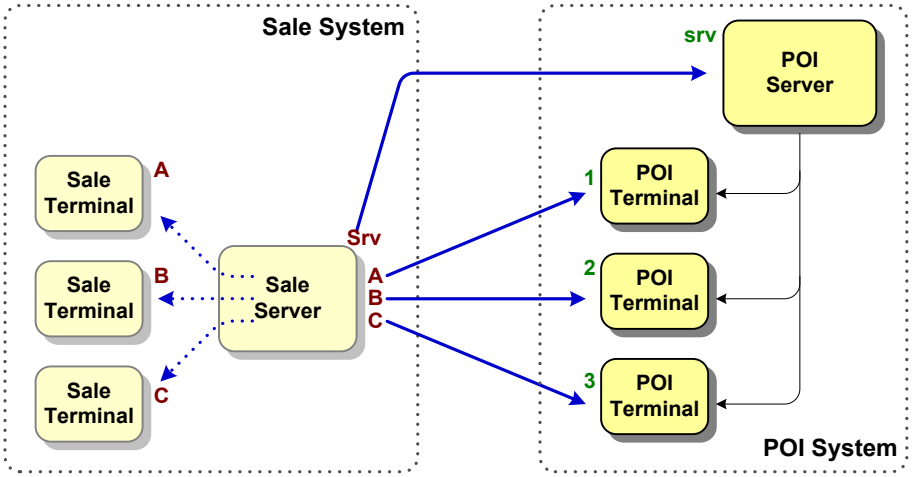


Figure 29: Server to Server Link, Server to Terminal case

2.5.5.3 Indirect Server to Server Logical Connections

Other cases occur when there is no direct links between the Sale and the POI Server to carry global services. We cannot describe all the possible combinations of the Sale POI and Server to Server architectures, but only some typical examples.

The first example is in the Clustered POI System where both the Sale Server and the POI Server use a Sale Terminal to POI Terminal connection to support the Server to Server connection.

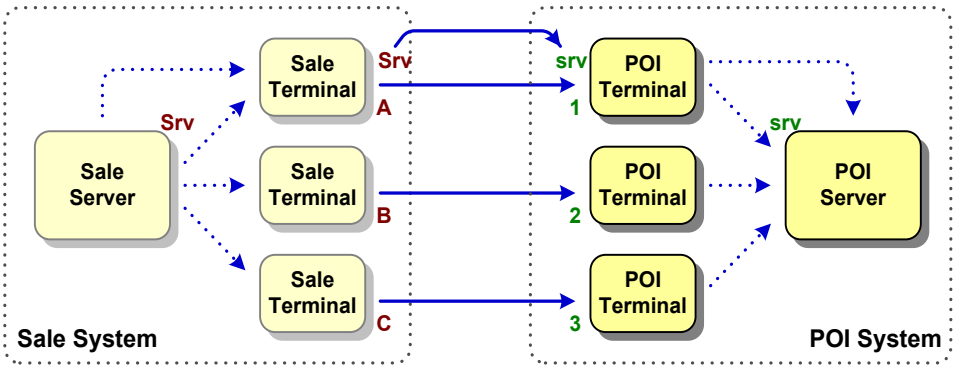


Figure 30: Server to Server Link through Terminal Connections

The second example is in the Distributed POI configuration, where the Sale Server uses a POI Terminal to connect to the POI Server.

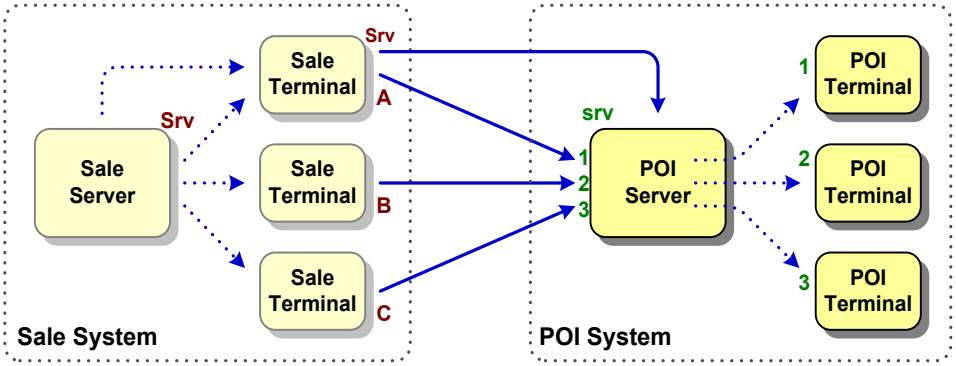


Figure 31: Server to Server Link through Sale Terminal Connection

The last example is where the Sale Server manages the interface to the POI System directly. The Sale Server uses a Sale Terminal to connect to the POI Server.

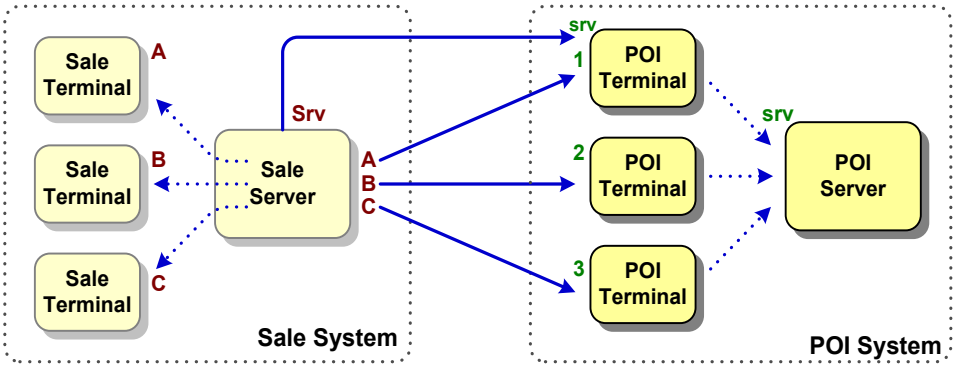


Figure 32: Server to Server Link through POI Terminal Connection

2.6 Configuration and Examples

2.6.1 Architecture Configuration Parameters

All the configuration parameters which are presented below, could be explicit or implicit (i.e. either parameters internal to the implementation, or "hard coded" or "by design" configuration). They are presented to simplify the understanding of the specifications against the various possible architectures.

The Sale and the POI Systems have to know identification of each system to allow exchange of Sale to POI messages.

Name	**POI System Name**
Definition	POI System Name identification.
Usage	Prefix of all the POI System components identifiers POIID
Specification	2.5.1 Identification of Systems and Components

Name	**Sale System Name**
Definition	Sale System Name identification.
Usage	Prefix of all the Sale System components identifiers SaleID
Specification	2.5.1 Identification of Systems and Components

Configuration 1: Sale and POI Systems Identification

For each Terminal to Terminal relationship, the type of link and the identification of the remote Terminal:

Name	**Terminal Relation**
Definition	Type of Terminal to Terminal relationship.
Usage	Transport connection and message dialogue.
Specification	2.5.4 Logical Connections Between Terminals

Label	*Description*
Connected	Connected POI System
Term2Term	Terminal to Terminal link.
Term2Serv	Terminal to Server link.
Serv2Term	Server to Terminal link.
Serv2Serv	Server to Server link.

Name	**Sale (or POI) Terminal Identifier**
Definition	Identification of the Terminal.
Usage	Logical addressing of the Terminal.
Specification	2.5.1 Identification of Systems and Components

Configuration 2: Terminals Identification and Logical Connections

The same information for the Server to Server relationship:

Name	**Server Relation**
Definition	Type of Server to Server relationship.
Usage	Transport connection and message dialogue.
Specification	2.5.5 Logical Connections Between Servers

Label	*Description*
Connected	Connected POI System
Direct	Direct Server to Server link.
Indirect	Indirect Server to Server link.

Name	**Sale (or POI) Server Identifier**
Definition	Identification of the Server.
Usage	Logical addressing of the Server.
Specification	2.5.1 Identification of Systems and Components

Configuration 3: Servers Identification and Logical Connections

3 Examples of Architectures

Considering the variety of architecture and implementation, there are necessary a lot of configurations not covered in this section. Presented implementations retain the terminology used in these concrete implementations, the linked to the wording we have defined in generic architecture is indicated when necessary.

3.1 Petrol Station

A typical petrol station includes two different types of payment:

- *Indoor Payments*, which are comparable to payment organisation in a convenient store,
- *Outdoor Payments*, which are unattended payment mainly at the petrol pump, but also for car wash or other services.

Indoor payments are managed by a Sale System and a Payment System, with the type we have already seen in the previous section.

In the other hand, outdoor payments are quite specifics:

1. The payment terminal inside the pump is often designed by the pump manufacturer, and managed by the pump.
2. The forecourt controller, which constitutes part of the Delivery System, manages all the pumps, and is operated manually or automatically by the cashier or the POS[7].

The logical architecture of a petrol station is presented below, including the Sale System (POS and POS Controller), handles the Delivery System (Petrol Pump distribution and Forecourt Controller) and the POI System (Indoor Payment Terminals, Outdoor Payment Terminals and Payment Server).

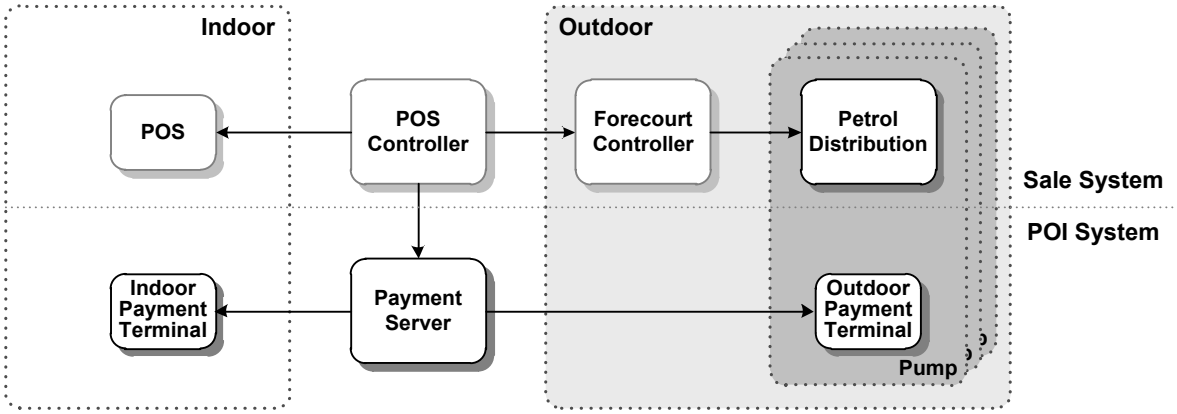


Figure 33: Logical Architecture of a Petrol Station

This is the architecture of Distributed POI System, where in case of outdoor payment terminal, the POI Terminal is associated to a logical Sale Terminal.

[7] Forecourt controller can be used as a communication gateway by the outdoor payment terminal. The outdoor payment terminals could however be linked to the Payment Server independently from the forecourt controller.

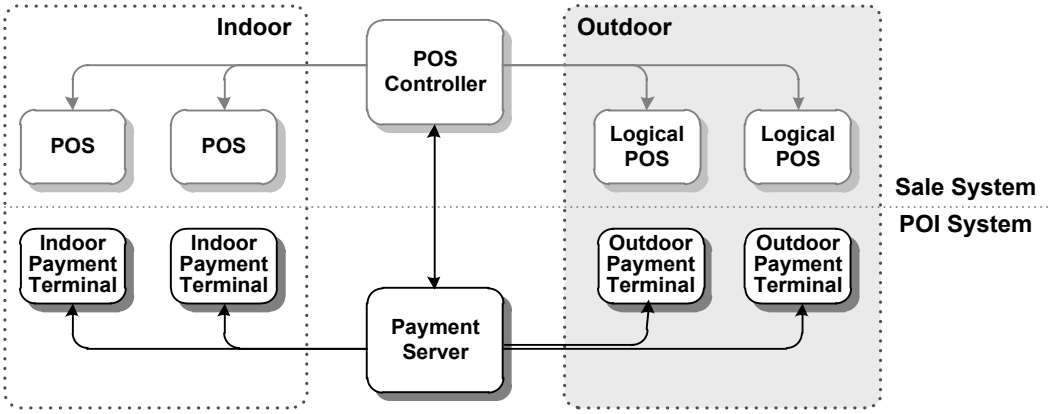


Figure 34: Distributed POI in Petrol Station

Considering the huge investment for a pump manufacturer to develop the payment terminal part of the pump, tendency is to separate these two products. As the unattended outdoor payment terminal offering is quite specific, compared to the pin-pad or standalone payment terminal market, there is a lack of standard to connect an outdoor payment terminal to a payment server.

3.2 Supermarket and Department Store

Supermarket and department store share the following features:

- An important number of lanes or POS,
- The majority of POS (sale terminals) are identical and aligned for supermarket, but some of them are very different and can request specific processing, or can belong to another owner. Some POS can be also semi-attended.

On the other hand, the purchases are processed quite differently:

- For supermarkets, transaction time is quite long, and contains dozen of items per purchase,
- For department store, transaction time is quite short, and contains few items per purchase,

The payment system is more often a Clustered POI System.

Example on Server Client (to be added)

3.3 Car Park

The car park is equipped with three types of payment terminals accessible to the cardholder:

- Inside the car at the parking barrier, registration of the card during the entry in the parking, and unattended payment with the card at the exit.
- Outside the car with automatic payment machines, unattended payment before get back the car to exit the parking.
- On an attended point of sale, inside the car at the barrier gate, or outside the car in the central supervision of the parking.

All these payment terminals are connected to a payment sever located in the parking or centralised, depending on the parking size.

There is a very simple interface between the Sale software in the various parking machine and the payment terminals to request payment. The payment terminal processes the minimum of functions, card reading and user interface, the payment server manages all the other functions.

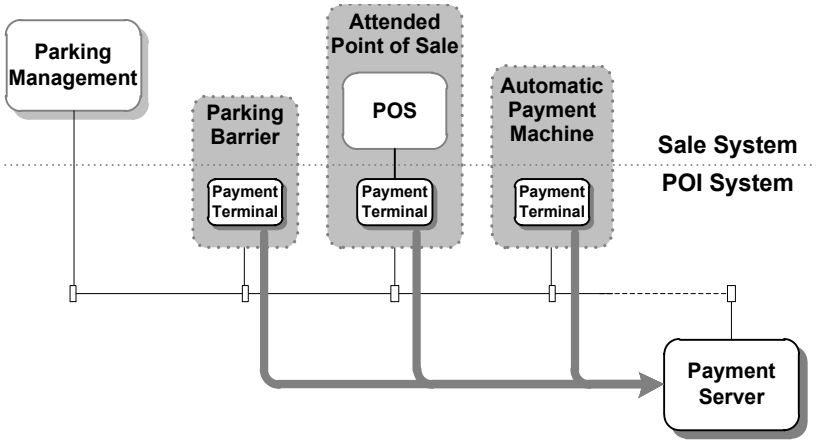


Figure 35: Car Park Architecture

This is the architecture of Clustered POI System, where all the transactions are driven by the POI Terminal and sent to a local or remote POI Server.

3.4 Vending Machine

We differentiate two kinds of vending machines:

1. Those accepting standard card scheme, which include a payment terminal and are on-line (e.g. ticketing).
2. Those accepting mainly cash and pre-paid cards in a closed and off-line environment (e.g. laundry, game, universities, prison, cafeteria…).

These two categories are in a public or private location. They use a communication gateway to send sale information and for the monitoring of the vending machine.

In case of type 1, a payment server can be present to manage several vending machines in the same area, and to reach an Acquirer Host.

In case of type 2, on-line payment solutions start to be installed using the communication gateway for this purpose, with magnetic stripe or contact-less card readers.

3.5 e-Commerce

Payment at home on open networks, realises more and more transactions each year, and are accomplished as MOTO (Mail Order Telephone Order) payment transactions, without a card reader.

Typical architecture for these payments includes:

- A *merchant Web server*, which corresponds to the store and the Sale System, and contains the product catalogue with sophisticated functionalities and the shopping card management.
- A *payment Web server* which corresponds to the POI System, and process payment part of the purchases,
- The *Cardholder* at home, which interacts with these two Web servers with a browser.

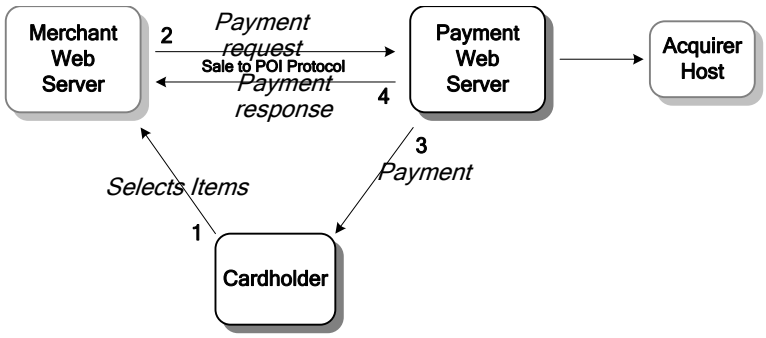


Figure 36: e-Commerce Architecture

The sequence flow of a transaction is the following:

1. The Cardholder chooses products in the catalogue, fills up shopping cart and decides to pay.
2. The merchant Web server prepares and sends a payment request to the payment Web server.
3. The payment Web server processes the payment transaction with the Cardholder and the Acquirer Host.
4. The payment Web server ends the payment and sends the result to the merchant Web server, which finishes the purchase transaction.

Even if the interface between the Cardholder and the payment Web server is very specific, the interface between the merchant Web server and the payment Web server can follow the Sale to POI protocol.

In some cases, a POI component can process the payment at the Cardholder site with a plug-in or a card reader.

Other architectures also propose a centralised role for the Cardholder, where after getting the payment data (1), he is directed to his Issuer host via a card scheme directory (2 and 2'), authenticated and debited by the Issuer (3), and bring the proof of the payment at the Issuer to the Acquirer (4 and 4').

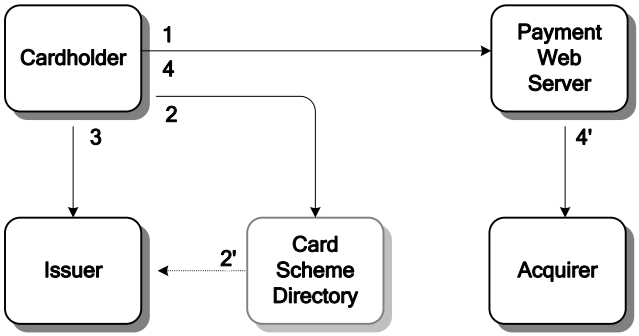


Figure 37: Cardholder Centralised Architecture