



BSI Standards Publication

Intelligent transport systems — Data interfaces between centres for transport information and control systems — Platform-independent model specifications for data exchange protocols for transport information and control systems

National foreword

This Published Document is the UK implementation of CEN ISO/TS 19468:2022. It is identical to ISO/TS 19468:2022. It supersedes PD CEN ISO/TS 19468:2019, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee EPL/278, Intelligent transport systems.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

This publication is not to be regarded as a British Standard.

© The British Standards Institution 2022
Published by BSI Standards Limited 2022

ISBN 978 0 539 14005 7

ICS 03.220.20; 35.240.60

Compliance with a Published Document cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 July 2022.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English Version

**Intelligent transport systems - Data interfaces between
centres for transport information and control systems -
Platform-independent model specifications for data
exchange protocols for transport information and control
systems (ISO/TS 19468:2022)**

Systèmes de transport intelligents - Interface de données entre centres pour les systèmes de commande et d'information des transports - Spécification du modèle indépendant de plateforme pour les protocoles d'échange de données pour les systèmes de commande et d'information des transports (ISO/TS 19468:2022)

Intelligente Verkehrssysteme - Datenschnittstelle zwischen Verkehrszentralen und Steuerungssystemen - Plattformunabhängige Modellspezifikationen für Datenaustauschprotokolle für Verkehrsinformationen und Steuerungssysteme (ISO/TS 19468:2022)

This Technical Specification (CEN/TS) was approved by CEN on 1 February 2022 for provisional application.

The period of validity of this CEN/TS is limited initially to three years. After two years the members of CEN will be requested to submit their comments, particularly on the question whether the CEN/TS can be converted into a European Standard.

CEN members are required to announce the existence of this CEN/TS in the same way as for an EN and to make the CEN/TS available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force (in parallel to the CEN/TS) until the final decision about the possible conversion of the CEN/TS into an EN is reached.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

European foreword

This document (CEN ISO/TS 19468:2022) has been prepared by Technical Committee ISO/TC 204 "Intelligent transport systems" in collaboration with Technical Committee CEN/TC 278 "Intelligent transport systems" the secretariat of which is held by NEN.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN shall not be held responsible for identifying any or all such patent rights.

This document supersedes CEN ISO/TS 19468:2019.

Any feedback and questions on this document should be directed to the users' national standards body/national committee. A complete listing of these bodies can be found on the CEN website.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to announce this Technical Specification: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

Endorsement notice

The text of ISO/TS 19468:2022 has been approved by CEN as CEN ISO/TS 19468:2022 without any modification.

Contents

Page

Foreword.....	vi
Introduction.....	viii
1 Scope.....	1
2 Normative references.....	1
3 Terms and definitions.....	1
4 Symbols and abbreviated terms.....	4
5 Exchange modeling framework.....	5
5.1 Overview.....	5
5.2 Business scenarios and functional exchange profiles.....	5
5.3 Requirements, features and exchange patterns.....	6
5.4 Business scenario: information delivery.....	7
5.4.1 Overview.....	7
5.4.2 Requirements.....	9
5.4.3 Data delivery exchange pattern.....	9
5.4.4 Specific exchange pattern specification PIMs included in this document.....	9
5.5 Business scenario: collaborative ITS services.....	9
5.5.1 Overview.....	9
5.5.2 Data exchange-enabling service request and feedback paradigm.....	10
5.5.3 Requirements.....	11
5.6 Exchange data model.....	11
5.7 Data exchange features.....	11
5.7.1 Context diagram.....	11
5.7.2 Features.....	12
5.8 Exchange pattern modeling using UML.....	16
6 Snapshot pull.....	20
6.1 Overview.....	20
6.2 Exchange pattern messages definition.....	21
6.2.1 Overall presentation.....	21
6.2.2 Exchange pattern definition.....	22
6.2.3 Relevant exchange information in exchange data model.....	23
6.2.4 Exchange messages.....	23
6.3 State diagrams.....	23
6.4 Features implementation description.....	23
6.4.1 Overview.....	23
6.4.2 Subscription contract.....	24
6.4.3 Session.....	24
6.4.4 Information management.....	24
6.4.5 Data delivery.....	25
6.4.6 Self-description.....	27
6.4.7 Communication.....	27
6.4.8 General optimization issues.....	27
7 Snapshot push.....	27
7.1 Overview.....	27
7.2 Exchange pattern messages definition.....	28
7.2.1 Overall presentation.....	28
7.2.2 Basic exchange pattern.....	28
7.2.3 Relevant exchange information in exchange data model.....	29
7.2.4 Exchanged messages.....	30
7.3 State diagrams.....	30
7.4 Features implementation description.....	30
7.4.1 Subscription contract.....	31
7.4.2 Session.....	31

7.4.3	Information management.....	31
7.4.4	Data delivery.....	31
7.4.5	Self-description.....	33
7.4.6	Communication/protocol.....	33
7.4.7	General optimization issues.....	33
8	Simple push.....	33
8.1	Overview.....	33
8.2	Exchange pattern messages definition.....	34
8.2.1	Overall presentation.....	34
8.2.2	Basic exchange pattern.....	35
8.2.3	Relevant exchange information from exchange data model.....	36
8.2.4	List of exchanged messages.....	37
8.3	State diagrams.....	38
8.4	Features implementation description.....	40
8.4.1	Overview.....	40
8.4.2	Subscription contract.....	40
8.4.3	Session.....	40
8.4.4	Information management.....	42
8.4.5	Data delivery.....	42
8.4.6	Self-description.....	43
8.4.7	Communication/protocol.....	43
8.4.8	General optimization issues.....	43
9	Stateful push.....	43
9.1	Overview.....	43
9.2	Exchange pattern messages definition.....	44
9.2.1	Overall presentation.....	44
9.2.2	Basic exchange pattern.....	45
9.2.3	Relevant exchange information from exchange data model.....	47
9.2.4	List of exchanged messages.....	48
9.3	State Diagrams.....	48
9.4	Features implementation description.....	50
9.4.1	Overview.....	50
9.4.2	Subscription contract.....	51
9.4.3	Session.....	51
9.4.4	Information management.....	53
9.4.5	Data delivery.....	53
9.4.6	Self-description.....	54
9.4.7	Communication.....	54
9.4.8	General optimization issues.....	54
10	Simple CIS.....	55
10.1	Overview.....	55
10.2	Exchange pattern and messages definition.....	55
10.2.1	Overall presentation.....	55
10.2.2	Basic exchange pattern.....	56
10.2.3	Relevant exchange information from exchange data model.....	58
10.2.4	Exchanged messages.....	59
10.3	State diagrams.....	60
10.4	Features implementation description.....	61
10.4.1	Overview.....	61
10.4.2	Subscription contract.....	61
10.4.3	Session.....	61
10.4.4	Information management.....	61
10.4.5	Self-description.....	66
10.4.6	Communication/protocol.....	66
11	Stateful CIS.....	66
11.1	Overview.....	66

11.2	Exchange pattern and messages definition	67
11.2.1	Overall presentation	67
11.2.2	Basic exchange pattern	68
11.2.3	Relevant exchange information from exchange data model	70
11.2.4	Exchanged messages	71
11.3	State diagrams	73
11.4	Features implementation description	74
11.4.1	Overview	74
11.4.2	Subscription contract	75
11.4.3	Session	75
11.4.4	Information management	75
11.4.5	Self-description	76
11.4.6	Communication	76
12	Other PIM definitions	76
Annex A	(informative) Methodology presentation	77
Annex B	(normative) Definition of requirements	79
Annex C	(normative) Basic exchange data model and data dictionary	84
Annex D	(informative) Introduction to communications and protocols	117
Annex E	(informative) Major functional exchange profile and exchange patterns for information delivery	122
Annex F	(informative) Data delivery background: stateless and stateful information with information life cycle management	124
Annex G	(informative) Collaborative ITS services (CIS) background	126
Bibliography		139

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*, in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 278, *Intelligent transport systems*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

This second edition cancels and replaces the first edition (ISO/TS 19468:2019), which has been technically revised.

The main changes are as follows:

- UML Communication diagrams have been improved (introduction of Agents and Interfaces to define actor subsystems interactions and addition of [subclause 5.8](#) to describe UML modeling methodology adopted);
- Void input parameter has been defined;
- Description of FEP+EP implementation has been improved with appropriate normative language;
- Publish Subscribe Exchange Pattern has been removed;
- Collaborative Intelligent Transport Systems (ITS) services requirements and features have been reviewed and added in [Clause 5](#) and [Annexes B](#) and [E](#);
- Collaborative ITS service FEP+EP PIM description has been introduced in [Clauses 10](#) and [11](#);
- [Annex C](#) has been reviewed, introducing new classes and attributes to support the implementation of features used in other exchange patterns;
- Annex H has been deleted;
- Certain figures have been improved.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document defines a common set of data exchange specifications to support the vision of a seamless interoperable exchange of traffic and travel information across boundaries, including national, urban, interurban, road administrations, infrastructure providers and service providers. Standardization in this context is a vital constituent to ensure interoperability, reduction of risk, reduction of the cost base, promotion of open marketplaces and many social, economic and community benefits to be gained from more informed travellers, network managers and transport operators.

Especially in Europe, delivering transport policy in line with the White Paper^[13] issued by the European Commission requires co-ordination of traffic management and development of seamless pan European services. With the aim of supporting sustainable mobility in Europe, the European Commission has been supporting the development of information exchange mainly between the actors of the road traffic management domain for a number of years.

This document supports a methodology that is extensible.

To be able to successfully connect systems and start exchanging data in an interoperable and easy way, there is a need to describe and agree on how this exchange ought to be achieved. This is set out in a data exchange specification. Data exchanges in different scenarios can have different needs and requirements. Therefore, several data exchange specifications can be needed.

Data exchange specifications need to address two main issues. Firstly, they model the stakeholders and actors involved in data exchange, each potentially in different roles, as well as abstract exchange patterns for their interactions. Secondly, they select a suitable implementation platform and clearly specify how the abstract scenarios and patterns are effectively implemented on this platform.

The diagram in [Figure 1](#) shows such an abstract communication scenario from the perspective of a road operator who requires data exchange interfaces between the different components of its own operational systems, either between centre-side components or between centre and field devices, but also to exchange information with other road operators or service providers.

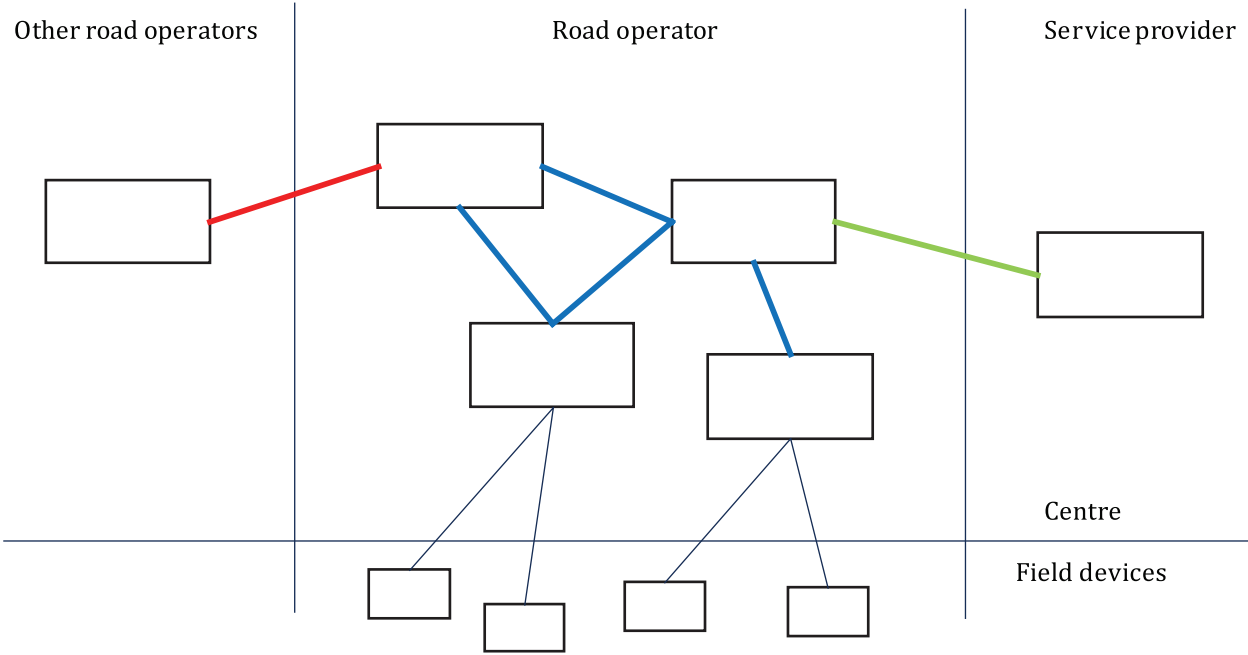


Figure 1 — Abstract communication scenario

While the black links between centre-side components and field devices may use a variety of communication protocols, mostly depending on the physical link conditions, the vast majority of other coloured links between centre-side components, internal to one organization or external to others, are

based on an internet protocol (IP) network and mostly use the transmission control protocol (TCP) transport layer protocol (user datagram protocol, UDP, is also possible in a few cases).

Nevertheless, as the different colours indicate, they can very well have significantly different requirements. Internal links (blue) can reside in one domain of trust, hence do not require protocols compatible with security gateways. This can already be different for links to other road operators (red) and will certainly not hold for links to other types of organizations, like service providers, via the Internet (green).

While different security requirements offer the most striking and obvious example, there are more criteria that can lead to different preferences on different types of links, e.g. scalability, robustness and integration complexity.

In broad terms, the colours blue – red – green form a hierarchy from more internal, closely-coupled, well-integrated systems towards external, loosely-coupled and non-integrated systems. The world of information and communication technology (ICT) offers a broad range of solutions for these different scenarios, offering different advantages and disadvantages. It is clear that the one-size-fits-all principle will not provide the most efficient way of working here. Even on the highest level of abstraction and inside the ICT domain itself, a well-known battle of paradigms between remote-procedure-call (RPC) type service specifications and RESTful architectures exists. The same clusters of options are found in the domain of ITS standards, where for example the European standard for the real-time information interface relating to public transport operations (SIRI; see the EN 15531 series) introduces both concepts as complementary options: Publish-Subscribe and Request-Response.

Furthermore, the ITS station architecture is not in contradiction with this document but is complementary to what is defined in this document. According to the principles and the taxonomy defined in ISO 21217, this document defines a conceptual notion of:

- How two central ITS (sub-)stations could communicate to:
 - deliver information (application data units);
 - negotiate functional service behaviour for collaborating traffic management functions (even if this use case could not directly be matched to ISO 21217 as it is not about information delivery).
- How a central ITS (sub-)station could communicate to deliver information (application data units) to another ITS station with the characteristics of a central ITS station.

This document specifies the process of defining the exchange characteristics by use case-driven feature selection of relevant parameters for the relevant OSI layers as defined in ISO 21217. Two exchange schemas are considered: information delivery and functional service negotiation between central ITS stations.

The drafting of this document was guided by the following principles:

- interoperability, such that different implementations can successfully engage in a data exchange process;
- supporting of legacy implementations which are based on existing (exchange) specification, in order to maximize investments already made by stakeholders;
- addressing other user profiles, not only road operators, thus making this document available to a broader audience;
- reusing existing (communications) standards, in order to reduce implementation complexity and take benefit of proven and already existing solutions for common ICT problems;
- maintaining a clear separation between the payload content and the exchange model.

[Annex A](#) details the adopted methodology for defining this exchange platform-independent model (PIM).

Intelligent transport systems — Data interfaces between centres for transport information and control systems — Platform-independent model specifications for data exchange protocols for transport information and control systems

1 Scope

This document defines and specifies component facets supporting the exchange and shared usage of data and information in the field of traffic and travel.

The component facets include the framework and context for exchanges, the data content, structure and relationships necessary and the communications specifications, in such a way that they are independent from any defined technical platform.

This document establishes specifications for data exchange between any two instances of the following actors:

- Traffic information centres (TICs);
- Traffic control centres/Traffic management centres (TCCs/TMCs);
- Service providers (SPs).

This document can also be applied for use by other actors, e.g. car park operators.

This document includes the following types of information:

- use cases and associated requirements, and features relative to different exchange situations;
- different functional exchange profiles;
- abstract elements for protocols;
- data model for exchange (informational structures, relationships, roles, attributes and associated data types required).

In order to set up a new technical exchange framework, it is necessary to associate one functional exchange profile with a technical platform providing an interoperability domain where plug-and-play interoperability at a technical level can be expected. The definition of such interoperability domains is out of scope of this document but can be found in other International Standards or Technical Specifications (e.g. the ISO 14827 series).

This document is restricted to data exchange. Definition of payload content models is out of the scope of this document.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1 business scenario

high-level description of the interactions that can exist within a system being analyzed or between the system and external entities (called actors) in terms of business functions

Note 1 to entry: See also *use case* (3.21).

3.2 client

entity that receives the information

Note 1 to entry: It is represented in the information delivery *business scenario* (3.1).

3.3 exchange pattern

EP

basic exchange architecture template, described by UML communication diagrams, that identifies the actors in the exchange framework and the available interactions among them, which enable data exchange functionalities as a set of exchange features

Note 1 to entry: Exchange pattern interactions can be described by means of UML sequence diagrams and state machine diagrams in such a way that message-triggering conditions are fully identified and defined alongside any state update based on the subsequent interaction, i.e. exchanged messages and interaction-derived conditions.

3.4 collaborative ITS service

CIS

ITS service (3.7) that can be enabled by combining different “ITS services” that are provided by the combined effort of two to more stakeholders who can have different roles

EXAMPLE Traffic management centres, traffic information centres, service providers.

3.5 functional exchange profile

FEP

selection of data exchange features for a particular *business scenario* (3.1)

3.6 interoperability domain

pair of *functional exchange profile (FEP)* (3.5) and platform selected for implementing a data exchange subsystem

Note 1 to entry: Each *platform-specific model (PSM)* (3.11) document defines an interoperability domain, which ensures that two implementations of this PSM are interoperable and can successfully exchange *payload* (3.9).

3.7 ITS service

processing of information to address specific ITS requirements and implement ITS features such as to manage traffic or deliver information

3.8 payload content model

content model

UML definition of the data structures that can be used to describe travel and traffic information to be exchanged in an exchange system

3.9**payload publication****payload**

bundle of information that is exchanged between two exchange systems containing an instance of the *content model* (3.8)

3.10**platform-independent model****PIM**

document describing the abstract model of the standardized data exchange process in a platform-independent way

Note 1 to entry: This definition is specific to this document.

3.11**platform-specific model****PSM**

document providing the implementation details of a *functional exchange profile (FEP)* (3.5) described in a *platform-independent model (PIM)* (3.10) for a concrete platform

Note 1 to entry: This definition is specific to this document.

3.12**profile-to-platform mapping**

act of defining an *interoperability domain* (3.6)

3.13**pull exchange**

exchange pattern (EP) (3.3) where the exchange of information is originated by the *client* (3.2)

3.14**push exchange**

exchange pattern (EP) (3.3) where the exchange of information is originated by the *supplier* (3.20)

3.15**simple push**

push-based *exchange pattern (EP)* (3.3) that does not require state to be maintained

3.16**snapshot**

set of data providing all of the last known state as opposed to providing partial changes

Note 1 to entry: This definition is specific to this document.

3.17**snapshot pull**

pull-based *exchange pattern (EP)* (3.3) where only the last *snapshot* (3.16) version is exchanged

3.18**snapshot push**

push-based *exchange pattern (EP)* (3.3) where only the last *snapshot* (3.16) version is exchanged

3.19**stateful push**

push-based *exchange pattern (EP)* (3.3) where data describing a communication session is maintained across successive communication within that session

3.20**supplier**

entity that provides the information

Note 1 to entry: It is represented in the information delivery *business scenario* (3.1).

3.21

use case

UC

set of operational interactions between entities (called actors) and a system to ease understanding of the main functions behind such interactions

4 Symbols and abbreviated terms

ASN.1	Abstract Syntax Notation One
BUC	business use case
F&L	freight and logistic
HTTP	hypertext transfer protocol
ICT	information and communication technology
IP	internet protocol
ITS	intelligent transport systems
LOS	level of service
MDA	model-driven architecture
MMI	man-machine interface
pub/sub	publish-subscribe pattern
REST	representational state transfer
RPC	remote procedure call
SOAP	simple object access protocol
SSL	secure sockets layer
TCC	traffic control centre
TMC	traffic management centre
TIC	traffic information centre
TIS	traffic information service
TCP	transmission control protocol
TLS	transport layer security
TMP	traffic management plan
UDDI	universal description discovery and integration
UDP	user datagram protocol
UML	unified modeling language (see the ISO/IEC 19505 series)
VMS	variable message sign

W3C	world wide web consortium
WSDL	web service definition language
WSIL	web services inspection language
WSS	web services security
XML	extensible markup language

5 Exchange modeling framework

5.1 Overview

The model-driven approach is chosen to describe exchange: this leads to describing exchange systems by means of abstract models, named platform-independent models (PIMs), in which the modeling of exchange features is achieved by describing interactions among systems and subsystems as exchange patterns (EPs). These interactions implement system capabilities as features that fulfil exchange requirements requested by specific business scenarios which are used to describe specific uses of exchange.

5.2 Business scenarios and functional exchange profiles

This document is based on business scenarios, i.e. a high-level description of the interactions that can exist within a system being analyzed or between the system and external entities (called actors) in terms of business functions. Business scenarios are derived from application requirements on useful business information required and on technical capabilities enabled by available technologies. FEPs are identified to ensure interoperable services with the restriction of determining one FEP per business scenario for a specific EP, which is an abstract model of available technical platforms.

One business scenario can be supported by more than one FEP. FEPs can be enabled by several EPs ([Figure 2](#)).

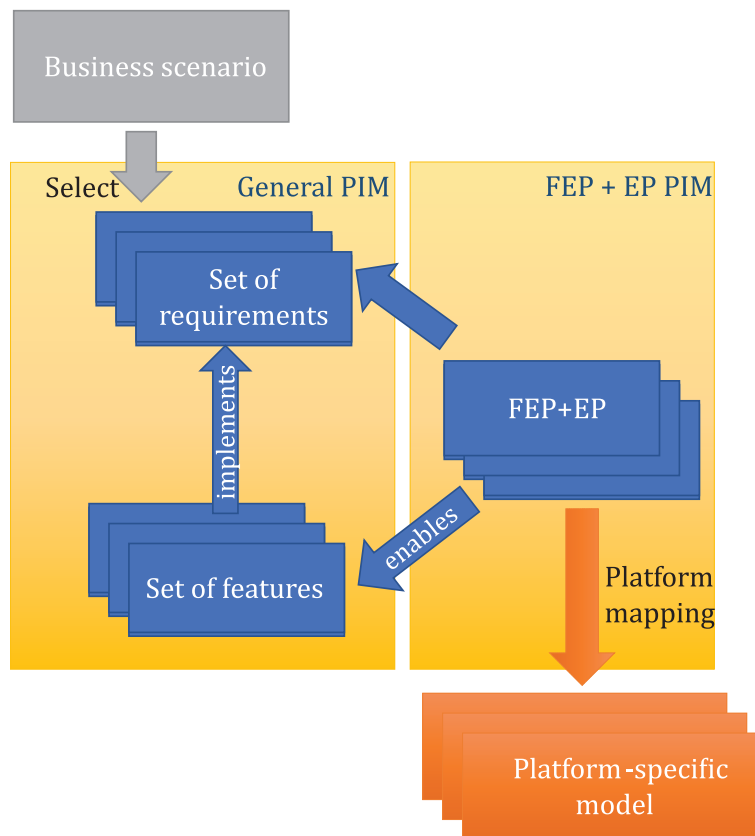


Figure 2 — Business scenario and functional exchange profiles

This document addresses the following business scenarios:

- information delivery,
- collaborative ITS services.

Other business scenarios can be developed in future editions of this document using the same methodology.

5.3 Requirements, features and exchange patterns

Requirements can vary depending on data exchange applications (i.e. use cases to be fulfilled). There are therefore many reasons to consider or ignore any requirement based both on the gathering of data at the supplier system and the usage of the delivered data by the client.

Requirements address the following items:

- information provision,
- communications,
- security,
- financial aspects.

Exchange is defined through enabling features which fulfil data exchange requirements.

Many possible technical EPs are possible, each of which can enable a subset of requirements in different ways. To be interoperable, a client and a supplier shall implement the same platform with the same pattern. Allowing a wide variety of possible EPs, the best suitable option for a use case application is chosen in order to fulfil the necessary set of requirements.

Based on the requirements of a specific business scenario selected for application use case implementation, a set of appropriate exchange features shall be combined into an FEP.

The following schema in [Figure 3](#) represents the domains of PIM and PSM introducing EPs and FEPs.

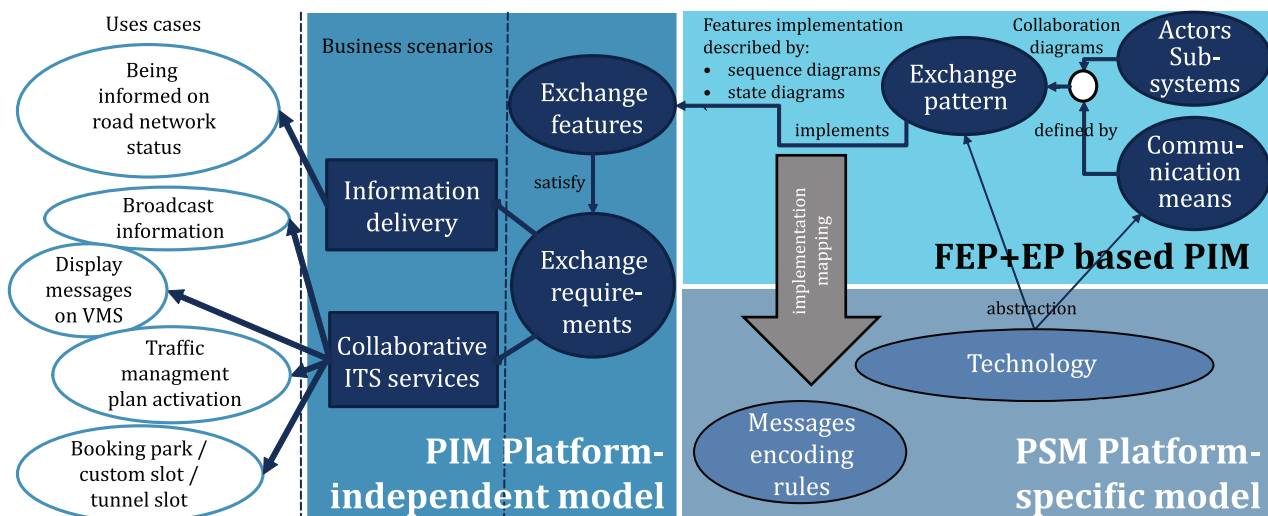


Figure 3 — Business scenario and functional exchange profile (FEP)

5.4 Business scenario: information delivery

5.4.1 Overview

One of the most common applications of a data exchange system is the exchange of traffic and travel information between two nodes. In such a scenario, one node acts as the supplier of the information while the other acts as the intended receiver of that information, i.e. the client.

EXAMPLE This is achieved by using the form of publications, e.g. in the European DATEX II.

The data delivery business scenario considers the actors as defined in [Figure 4](#):

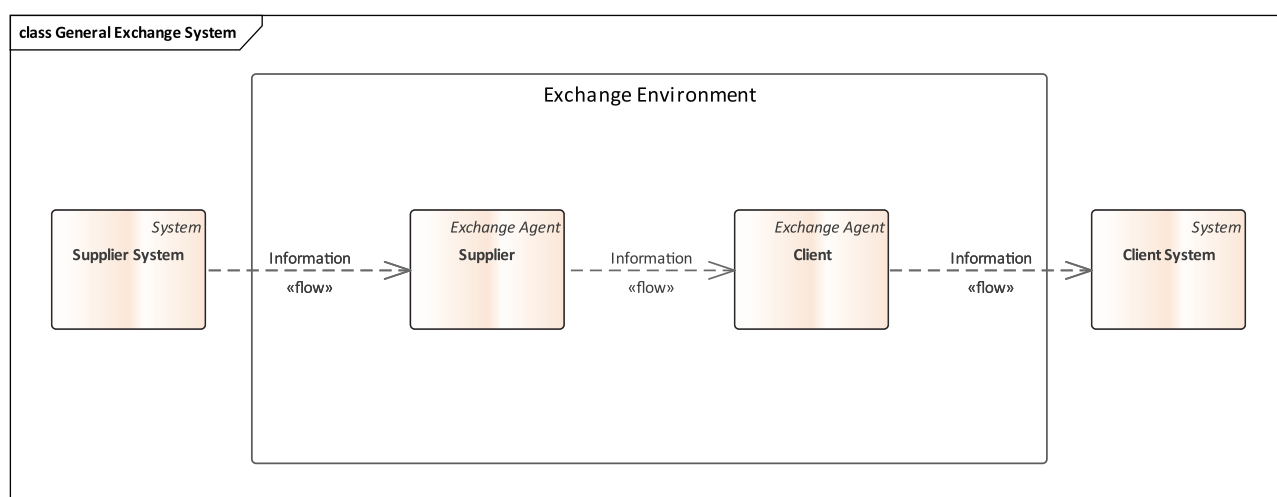


Figure 4 — General data delivery business scenario actors

[Table 1](#) provides the basic definitions for exchange:

Table 1 — Main definitions in exchange

Name	Definition
Supplier system	A system which gathers information (road information) which needs to be conveyed to another system, named "client system". Examples of supplier systems are traffic control centres, traffic information centres or service provider systems, gathering road data from any available source they have.
Client system	A system which needs to update its internal information based on information which is available from another system, named "supplier". Examples of client systems are traffic control centres, traffic information centres or service provider systems.
Exchange environment	The set of components which enables information exchange among client systems and supplier systems via a data exchange protocol.
Supplier	The component of the exchange environment which is devoted to providing data to the client and retrieving them from supplier system.
Client	The component of the exchange environment which is devoted to collecting data from the supplier and delivering them to the client system.

Road and traffic information is gathered in a system named "supplier system". In case this information is needed for any purpose by another system, named "client system", it has to be transferred among the two systems by the exchange environment.

The data delivery business scenario describes the EP and messages which are needed to be exchanged among the supplier and client systems besides the underneath technology and EP. The purpose for which information is exchanged is not considered in this use case description.

As explained in the information delivery background in [Annex F](#), any update of information status at the supplier system shall be replicated to the client system via information delivery. The main objective of information delivery is that information on the client system is updated exactly in the same way as it is in the supplier system without any difference in information values and semantics.

"Exchange message" is defined as the data structure in which the information is coded to transfer information in the exchange system from the supplier to the client.

Assuming S_c = Client status and S_s = Supplier status, exchange is a means to achieving an S_c equivalent to S_s .

Formally:

$$S_s \rightarrow \text{Information} \rightarrow \text{Supplier} \rightarrow \text{Exchange message} \rightarrow \text{Client} \rightarrow \text{Information} \rightarrow S_c$$

This is to say that client system status is updated in an equivalent mode to supplier system status by means of data delivery exchange messages between supplier and client.

The information delivery business scenario scope is implemented by selected exchange features. These selected exchange features enable this scope and other secondary requirements which are based on available features on considered platforms and patterns.

1) Supplier system characterization:

- shall provide data as input to the data exchange environment;
- is mandatory for the information data delivery process.

2) Data exchange environment:

- shall be an environment supporting the exchange of information and data by mean of messages;
- the supplier of the data exchange system shall produce and transmit the messages (notification);
- the client of data exchange system should receive and process the messages.

3) Client system characterization:

- shall receive traffic and travel-related data from the data exchange environment;
- can be either another system for further processing or a simple client for content visualization (the purpose of information exchange is not considered to be relevant in this information delivery business scenario);
- is mandatory for the information data delivery process.

5.4.2 Requirements

[Annex B](#) provides a description of all requirements that apply to specific business scenarios, including information delivery, whether actually used in a particular FEP or not. All requirements are organized into groups based on their characteristics.

NOTE A FEP is a different concept from content profile. The description of the content profiles that can exist in an information delivery scenario is not within the scope of this document. Content profiles are handled depending on the information exchanged.

5.4.3 Data delivery exchange pattern

For data delivery, the concept of EP is introduced, as well as the concepts PIM and FEP (see [Clause 3](#) for definitions).

Examples of EPs are the GET method of HTTP, Pull, Push, PubSub, etc.

Once an EP and a selection of features (FEP) which can be implemented on this EP are set, a set of specification at PIM level for EP+FEP is well-defined.

A PIM for EP+FEP that enables all mandatory requirements is a valid platform for the corresponding business case (information delivery or collaborative ITS services).

5.4.4 Specific exchange pattern specification PIMs included in this document

In the following clauses, PIMs for EP/FEP will be described for the following platforms:

- Snapshot Pull;
- Snapshot Push;
- Simple Push;
- Stateful Push;

[Annex E](#) details the features of the main FEPs considered in this document for the corresponding EP.

5.5 Business scenario: collaborative ITS services

5.5.1 Overview

A service refers to any processing of data which enables a value for the data themselves. Within the field of ITS, ITS services can be considered as the processing of information to address specific requirements, such as to managing traffic and delivering information.

Collaborative ITS services (CISs) are ITS services that can be enabled by combining various ITS services, provided by several stakeholders who may have different roles (e.g. traffic management centres, traffic information centres, service providers).

CIS exchange specifications explore user requirements and define common techniques to address them in order to implement collaborative “ITS services” by different centres. They are based on exchanging

information to be processed by the different nodes and receiving the processing outcomes (feedback), giving a simple mechanism, upon which it could be possible to build more sophisticated workflows, enabling the coordination of operations distributed among many centres.

In the CIS business scenario, the exchange of information among centres is considered as a base mechanism for triggering specific processing and for providing services on this data, so the data exchange layer is to be considered as an ITS services enabler.

A description of ITS services is out of the scope of this document: a detailed description of possible usage of CIS is provided in [Annex G](#).

Examples of ITS services in the different ITS domains are:

- TIS (traffic information services):
 - information delivery,
 - allowed channels.
- TMS (traffic management systems):
 - hard shoulder running,
 - dynamic speed,
 - dynamic lane management,
 - etc.
- F&L (freight & logistic):
 - secure truck parking,
 - etc.

5.5.2 Data exchange-enabling service request and feedback paradigm

The involved systems, which in data delivery had been considered as supplier and client, can be considered in this paradigm respectively as service requester and service provider.

At application level, for implementing a business case, such as management of TMP, workflow management is usually requested; this could be a simple workflow (one shot request and acceptance or rejection) or a more complex one: complex workflow modeling at application level is out of the scope of this document. This document provides the essential tools as building blocks for implementing such a simple workflow in order to raise a service request and provide feedback as exchange of payload to be processed and as processing results.

CIS-enabling mechanisms include:

- service request delivery from service requester to the involved service providers;
- service feedback delivery from service provider to the relative service requester.

The main characteristics of CISs are as follows:

- collaborative: two to many systems interact to achieve a common objective by processing data and to exchange processing results;
- CIS modeling considers the use of a set of "1 to n " exchange connections. Any single CIS triggering request is considered as one service requester and one to many service providers, so that any complex interaction can be considered a combination of single usages from multiple requesters;

- possible lock of resources at service provider sides (to be managed at application or human operator level) are not described in this document.

5.5.3 Requirements

[Annex B](#) provides a description of all requirements that apply to specific business scenarios, including collaborative informative services.

5.6 Exchange data model

To implement some exchange features (such as session management or link monitoring) or security features, extra information is to be added to the informational (payload) content. This extra information, named exchange information, enables messages to convey information and data which are related to client and supplier status, identity, authorization, etc.

Exchange information can be different among EP+FEP selection, but some common general information models are considered. These can be specialized to manage specific EPs and PIMs.

A general UML model for managing a minimum set of information for exchange is provided in [C.2](#).

5.7 Data exchange features

5.7.1 Context diagram

This document defines the features and rules that systems shall implement in order to be able to communicate in the traffic and travel data exchange world.

The context diagram in [Figure 5](#) shows the entities and the features specified in this document and which need to be addressed by the technical implementations. The diagram presents the features in different layers for application, message rules and communication:

- The “Application layer” is used for defining, using and implementing different business and application needs. This document describes the features to deal with how and when the information is published and made available, how subscriptions are managed and how to handle services and transactions. This layer defines the semantics of the communication.
- The “Messages rules layer” defines the features and the rules used for the transport of the messages. This is the layer where the rules (protocol) are defined that enable different systems (suppliers and clients) to communicate and understand each other, i.e. for sending and receiving messages. This layer defines the syntax of the communication.
- The “Communication layer” deals with the support for communication between systems, and defines the protocols and services used by the data exchange applications, e.g. TCP/IP, security, web services. The communication layer deals with rules at the lowest level, i.e. the physical exchange. This layer provides solutions to the defined requirements and features, although it is up to the upper applications to define which protocols to use and how to use them. This layer defines the physical support for the communication.

[Figure 5](#) shows the topics broken up into boxes representing the exchange features and the relation with each layer.

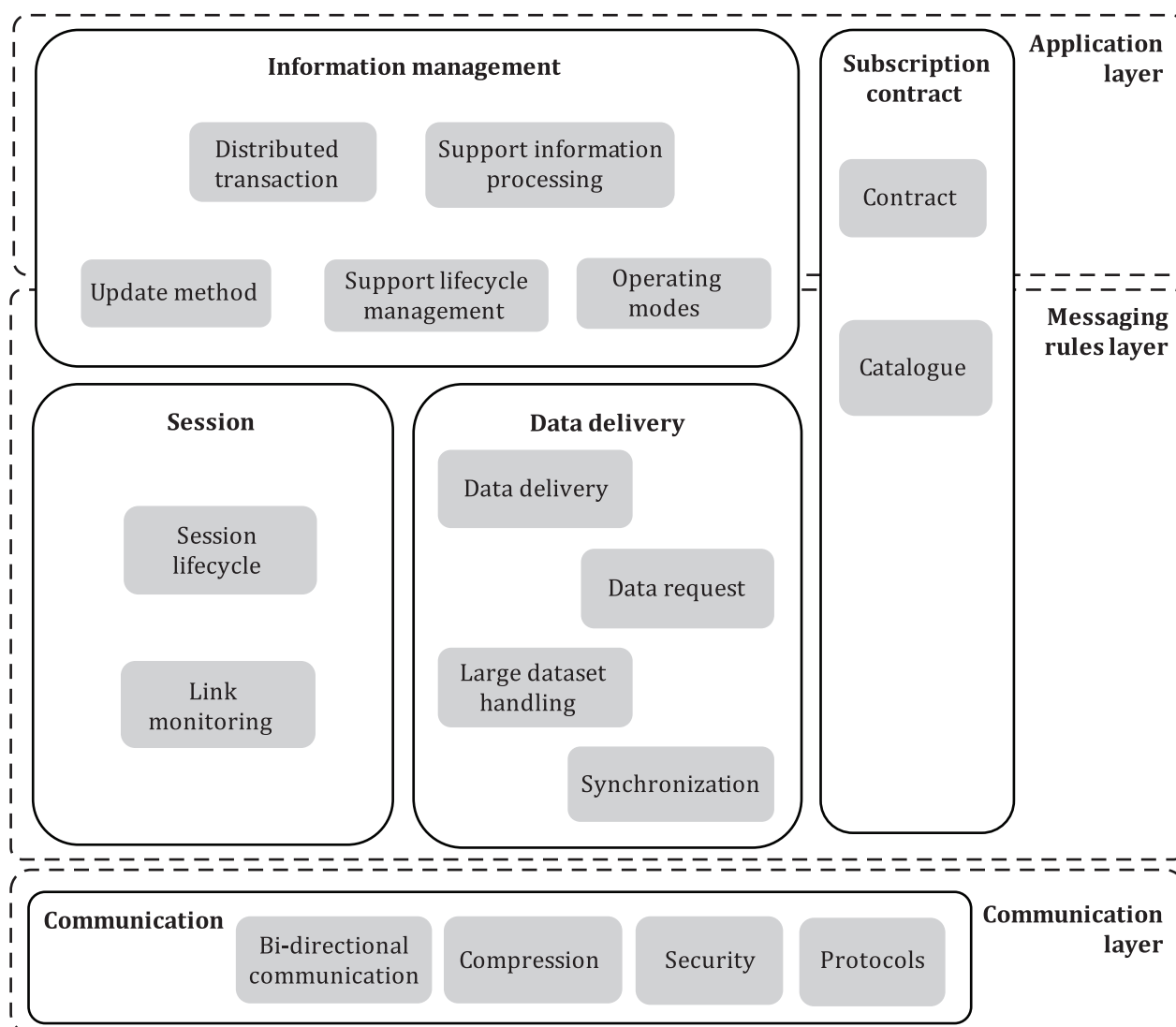


Figure 5 — Context diagram

5.7.2 Features

The features that support the requirements defined by the use cases specified in this document are detailed as follows:

- 1) The “subscription contract” feature group, detailed in [Table 2](#), supports all features related to the contract or agreement, such as:
 - a) Contract and contract life cycle: a model and features that can be used for the support of the information of a subscription contract;
 - b) Catalogue: a model for handling catalogues.

The subscription contract is optional and includes the following features:

Table 2 — Subscription contract: Features and requirements

Features	Requirement type	Requirement name
Contract	Information	Subscription
		Client profiles
		Filter handling
Catalogue	Information	Catalogue exchange

- 2) The “session” feature group, detailed in [Table 3](#), supports all features related to the establishment of a logical session.
- a) Session life cycle: features for managing the life cycle of a logical session (create, manage and terminate);
 - b) Link monitoring: features for link monitoring and control.

The session feature group is optional and includes the following features:

Table 3 — Session: Features and requirements

Features	Requirement type	Requirement name
Session life cycle	Communication	Error handling
		Time-out management
		Session
Link monitoring	Communication	Error handling
		Time-out management
		Full reliability
		Link monitoring and control

- 3) The “information management” feature group, detailed in [Table 4](#), handles the features related to the management of the information and includes features such as:
- a) “Operating modes”: features to specify what portion of the information shall be exchanged;
 - b) “Update methods”: features that let a data exchange system specify when the information should be exchanged;
 - c) “Life cycle management”: features for handling the life cycle management of exchanged payload information for payload for which life cycle is applicable:
 - i) Situation life cycle management
 - i) Filter handling
 - d) “Support information processing”: features for handling directives to process exchanged data and send feedback on processing outcomes;
 - e) “Distributed transaction”: features for handling a transaction on several systems consistently, i.e. an operation is capable of maintaining data consistency among several systems based on undertaken operator actions.

The “information management” feature group includes the features listed in [Table 4](#).

Table 4 — Information management: Features and requirements

Features	Requirement type	Requirement name
Operating modes	Information	Reference datasets for different versions
Update methods	Information	Full audit trail data delivery (all state changes)
Life cycle management	Information	Support for life cycle management
Support information processing	Information	Support for information management
		Support for feedback on information management
Distributed transaction	Information	Distributed transaction
		Distributed atomic transaction

- 4) The “data delivery” feature group, detailed in [Table 5](#), supports all features to exchange data between the supplier and client. In the publish-subscribe pattern, this feature group will support all related interfaces between the producer and the consumer. It supports features such as:
- a) “Data delivery”: features to delivery information by the supplier to the client on a push mode (direct delivery and fetched delivery);
 - b) “Data request”: features to exchange information requested by the client;
 - c) “Large datasets”: support the exchange messages with large volumes;
 - d) “Synchronization”: how to ensure data synchronization between the systems that are communicating.

The “data delivery” feature group includes features and requirements listed in [Table 5](#).

Table 5 — Data delivery: Features and requirements

Features	Requirement type	Requirement name
Data delivery	Information	Extensibility
	Communication	Delivery/response
		Message sequence
		Snapshot data delivery (last known state)
		Exchange quality measures (e.g. response timestamp)
		On occurrence update
		Periodic update
	Security	Client identification
		Supplier identification
	Information	Incremental data delivery

Table 5 (continued)

Features	Requirement type	Requirement name
Data request	Information	Extensibility
	Communication	Request/response
		Message sequence
		Snapshot data delivery (last known state)
		Exchange quality measures (e.g. response times-tamp)
		On occurrence update
		Periodic update
	Security	Client identification
		Supplier identification
Large datasets handling	Information	Data delivered as soon as possible
		Delayed delivery
		Multi-part data delivery
Synchronization	Information	Synchronization
	Communication	With state supplier
		Failed data recovery

- 5) The "communication" feature group, detailed in [Table 6](#), handles all features related to a protocol (close to the physical layer). It is used by the application for the message transport:
- "Communication": describe the communication protocols that can be used;
 - "Security": describe how security features can be implemented;
 - "Compression": how data compression can be used while transmitting data;
 - "Bi-directional communication": enable a client to send back data to a supplier as feedback on data exchanged and processing status of data based on the supplier's processing directive in collaborative ITS services.

The "communication" function is responsible for implementing the following features:

Table 6 — Communication: Features and requirements

Features	Requirement type	Requirement name
Security	Security	Security (data)
		Integrity
		Confidentiality
		State the intended recipient
		Client authentication
		Supplier authentication
		Client authorization
		Non-repudiation
Compression	Communication	Compression
Communication	Communication	Timely responses
Bi-directional communication	Communication	Bi-directional communication

5.8 Exchange pattern modeling using UML

EPs which enable features for selected FEPs can be described by use of certain UML diagrams that describe which actors are involved in the exchange, and the specific interactions among them that enable the features themselves.

The exchange system description assumes that external systems like TCC and TMC are external to the exchange and that exchange agents are the actors who may implement the exchange role of client, supplier, service requester or service provider as defined in the business case.

The [Figure 6](#) describes the main exchange-involved actors as well as other interacting actors external to exchange.

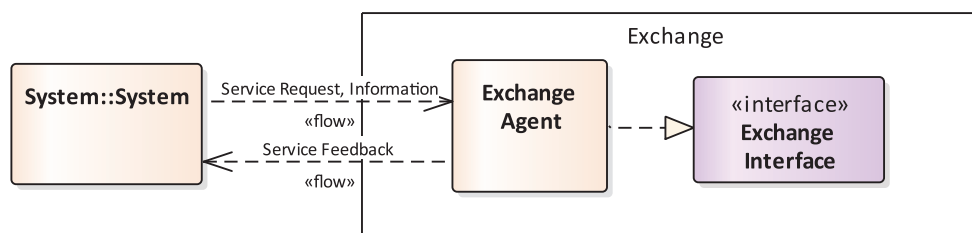


Figure 6 — Exchange actors

Systems (e.g. TCC or TMC system) may act both as client or supplier and for exchange purposes. Systems need to interact with the exchange agent of the same client or supplier type that realizes a system interface enabling the exchange features.

The relative system and interfaces can be specialized based on multiple options for a system to be a client or supplier for data delivery, or a service requester or service provider for collaborative ITS services business scenarios. [Figure 7](#) and [Figure 8](#) show the data delivery exchange actors and the data delivery exchange systems, agents and interfaces, respectively.

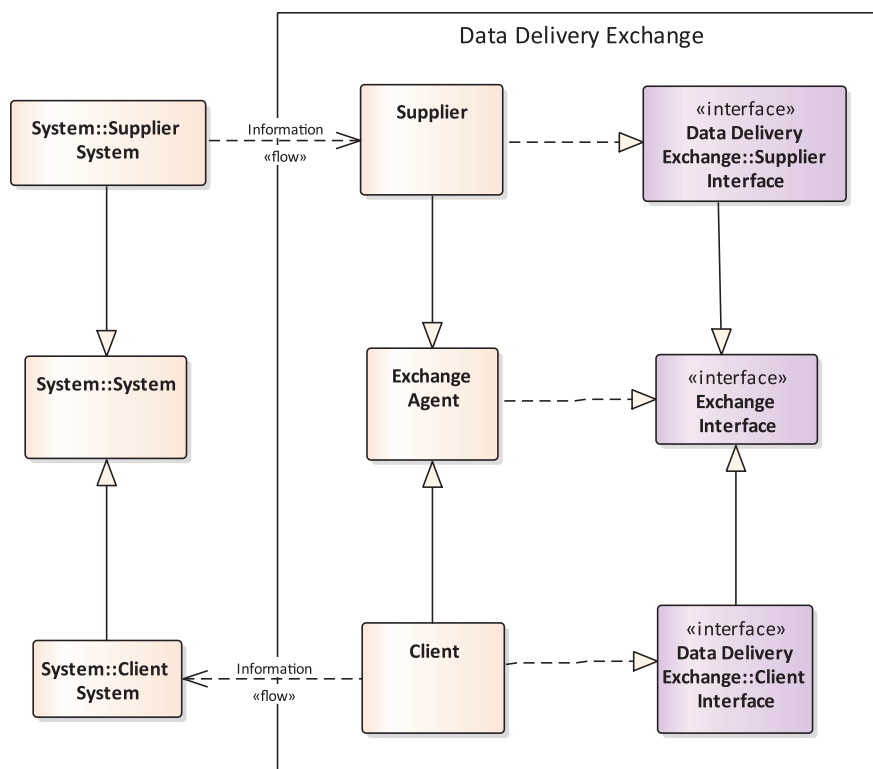


Figure 7 — Data delivery exchange actors definitions

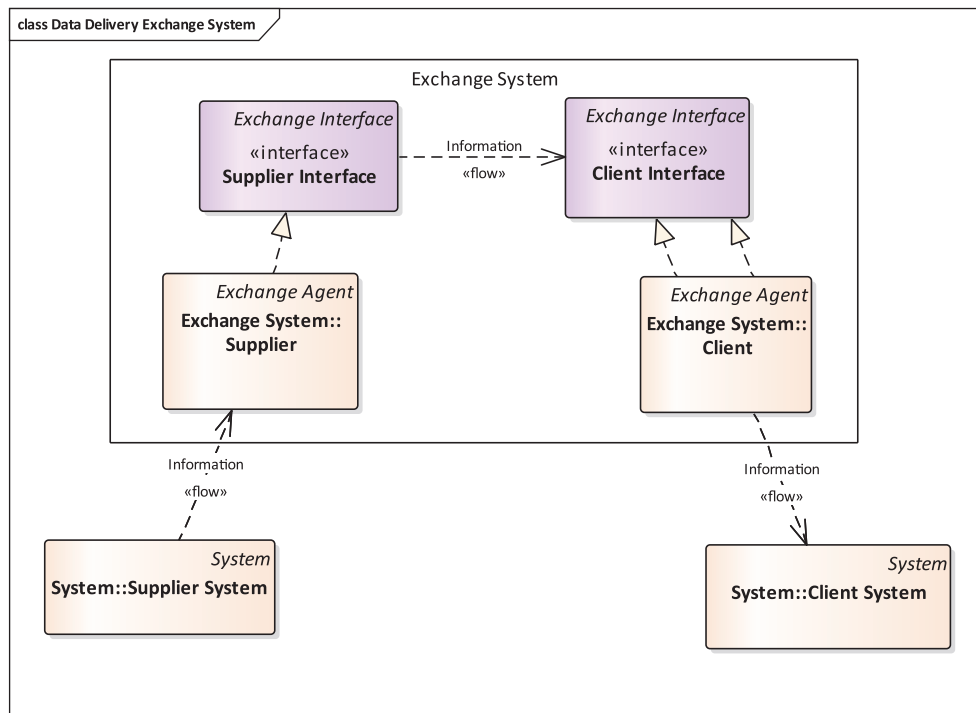


Figure 8 — Data delivery exchange systems, agents and interfaces

Figure 9 and Figure 10 show the systems and exchange agents and interfaces for collaborative ITS services.

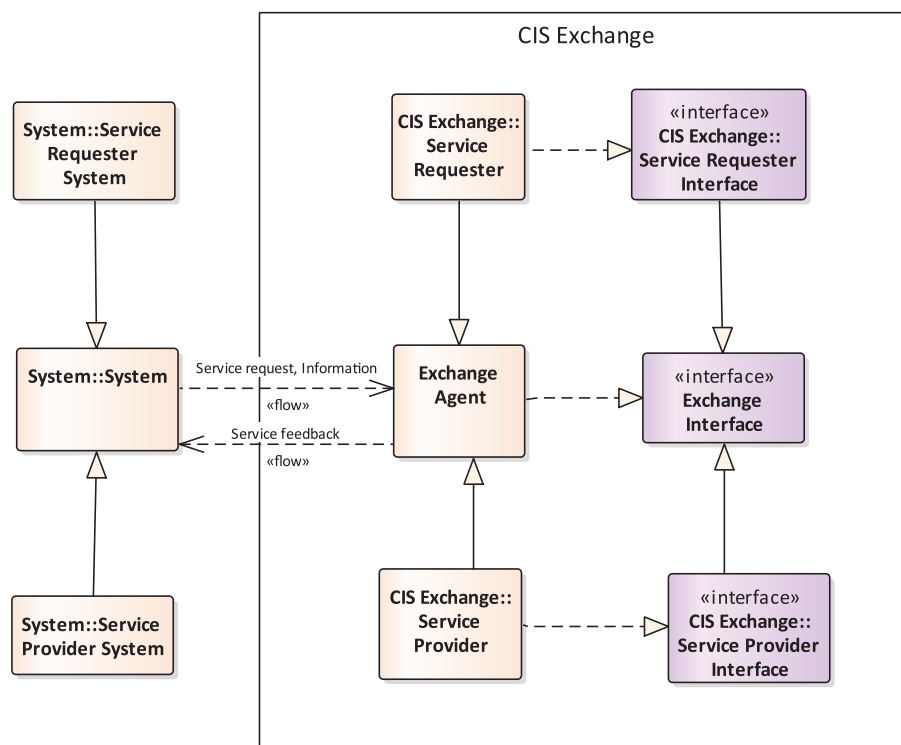


Figure 9 — CIS exchange agents definition

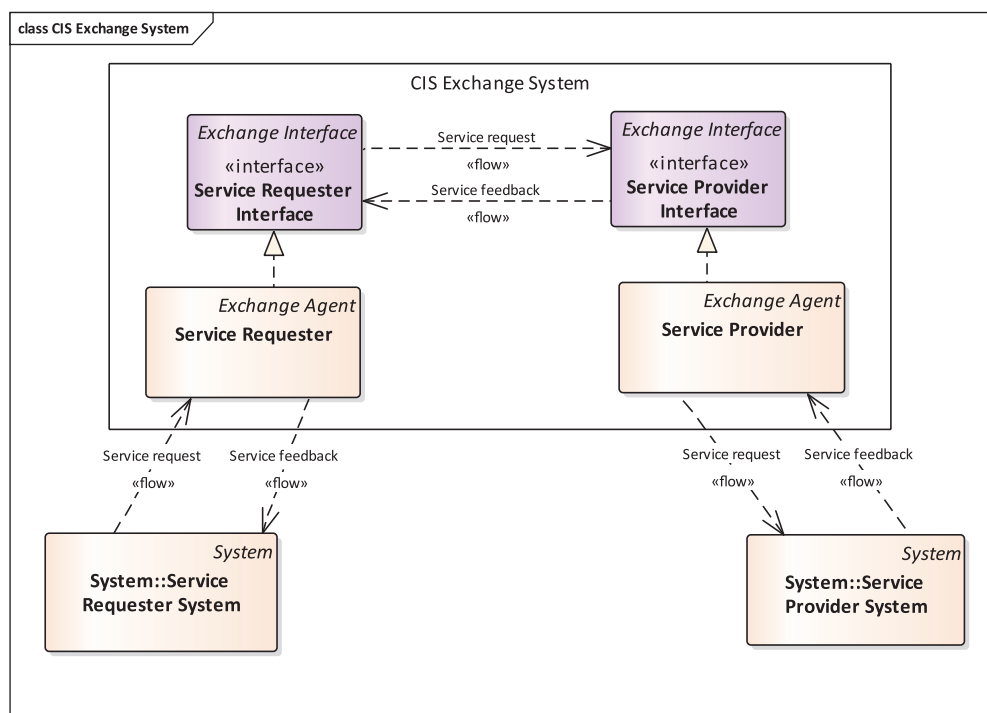


Figure 10 — CIS exchange systems, agents and interfaces

This system, exchange agents and interfaces classes will be used in the FEP+EP specification descriptions in the following clauses to describe the provided interfaces and interactions. Each exchange interface realization will be implemented as a specific specialization for its FEP+EP. For example, snapshot push will have its specific "snapshot push" supplier interface, i.e. the "snapshot push client interface", which will be described as a specialization of the exchange snapshot push client interface. The modeling for a snapshot push client system and its exchange agent and interface is shown in [Figure 11](#).

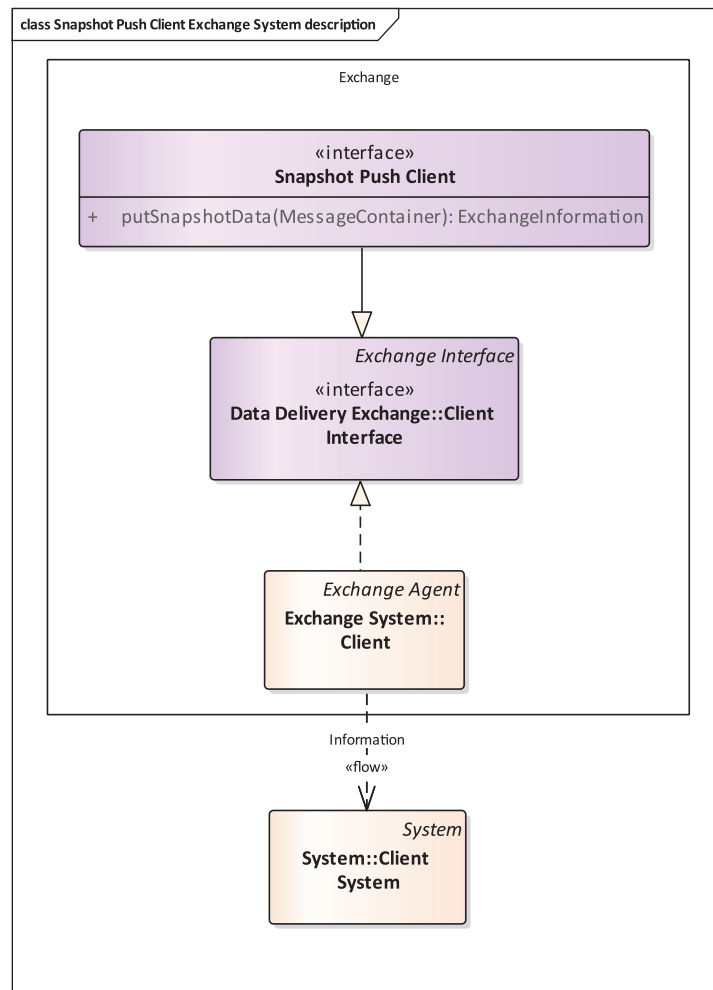


Figure 11 — Sample of "snapshot push" client system and its exchange agent and interface

The relevant communication diagrams will be introduced in the FEP+EP descriptions in the following clauses.

In the description of the interactions among exchange systems and subsystems, UML sequence diagrams will be used. Interactions among actors and their interfaces are described using the actor themselves. The realized specialized interfaces are not mentioned for schema understanding convenience. [Figure 12](#) is an example of the provided sequence diagram.

NOTE In all sequence diagrams introduced in this document, only the interactions among supplier and client are mandatory, and they are intended for the interaction among supplier and client exchange interfaces. Other interactions are introduced for better understanding of the overall system workflows and can be assumed to be generally valid, but those interactions are defined and implemented in the ITS systems in a way which is not defined in this document.

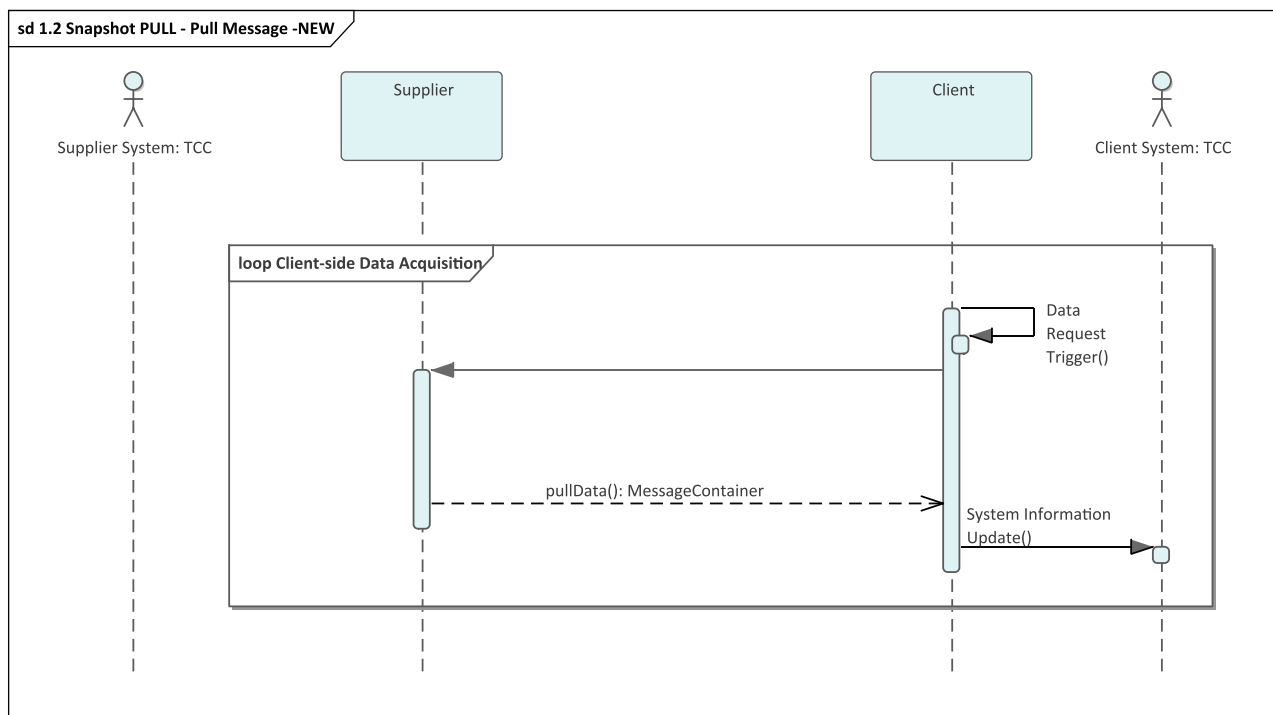


Figure 12 — Example of a sequence diagram

State diagrams can also be used to specify the specific status of exchange operations which will lead to specific interface behaviours such as return messages definition and specific interactions provided (e.g. synchronization, synchronization request). State diagrams are not needed for understanding in simple cases such as stateless FEP+EP so they will not be described in some FEP+EP descriptions.

6 Snapshot pull

6.1 Overview

The “snapshot pull” EP+FEP at PIM level is based on information retrieval by a client from a supplier which delivers a snapshot of information (i.e. all currently available information content) to the client. It can be implemented in several platforms: some examples are XML retrieval of generated XML files by http/get, or supplier exposing a SOAP Web Service method (e.g. named "PULL"), from which currently available data is retrieved by the client.

This “snapshot pull” does not manage session life cycle and link monitoring requirements, as well as synchronization. This feature and related requirements are not considered in this pattern. Synchronization is not required as implicit when delivering snapshots of currently available information content.

To describe the snapshot pull EP+FEP at PIM level, all features are described in a general abstract format, independently from the specific technology platform with which this model will be implemented (e.g. http/get XML, WebService). [Table 7](#) shows a selection of features for snapshot pull.

Table 7 — Selection of features for snapshot pull

Features area	Feature	Snapshot pull implemented
Subscription contract	Contract	N
	Catalogue	N
Session	Session life cycle	N

Table 7 (continued)

Features area	Feature	Snapshot pull implemented
	Link monitoring	N
Information management	Operating modes	Periodic or On Occurrence (i.e. triggered by client conditions)
	Update methods	Snapshot
	Life cycle management	Y, based on snapshot: new and updated content delivered, outdated data not delivered
Data delivery	Data delivery	Y
	Data request	N
	Large datasets handling	N
	Synchronization	Y
Self-Description	Handshake	N
Communication	Security	To be defined at PSM level
	Compression	To be defined at PSM level
	Communication	To be defined at PSM level

6.2 Exchange pattern messages definition

6.2.1 Overall presentation

The information delivery business scenario description and definition states that data exchange is needed to align the information kept by the supplier system into the client system. For this purpose, an exchange system is used which provides tools enabling messages generation and their transfer between supplier and client (see [Figure 13](#)).

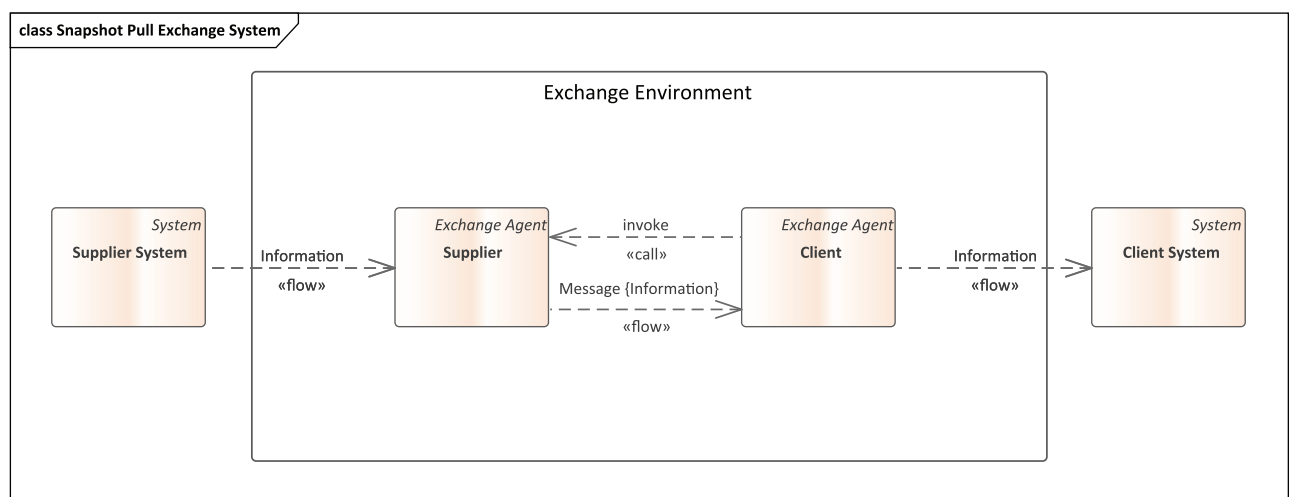


Figure 13 — Snapshot pull exchange actors

The snapshot pull EP is described in the following subclauses.

6.2.2 Exchange pattern definition

In a snapshot pull context, the supplier exchange system provides a mechanism to retrieve currently available and valid data (i.e. a snapshot of information) from an action taken at the client side, which will invoke this specific mechanism offered by the supplier.

In the context of the snapshot pull FEP +EP framework, to enable interoperability among client and supplier, all rules defined in this subclause apply.

A snapshot pull supplier exchange system shall realize a snapshot pull supplier interface that provides a "pullSnapshotData" method for implementing the snapshot pull mechanism.

A snapshot pull client exchange system shall realize a snapshot pull client interface that invokes the "pullSnapshotData" method provided by the snapshot pull supplier interface to retrieve snapshot data.

Figure 14 shows the communication diagram for snapshot pull FEP+EP.

In this FEP+EP framework the client “pulls” messages from the supplier.

The client shall deliver no information to the supplier in the pull request.

The supplier shall return the client pull request by delivering a "MessageContainer" information which includes ExchangeInformation.

NOTE This return message is available to bring some information from the client to the supplier, (e.g. exchange information), which can be used for any further exchange features implementations or application level checks or processing which are out of scope of this FEP+EP specification.

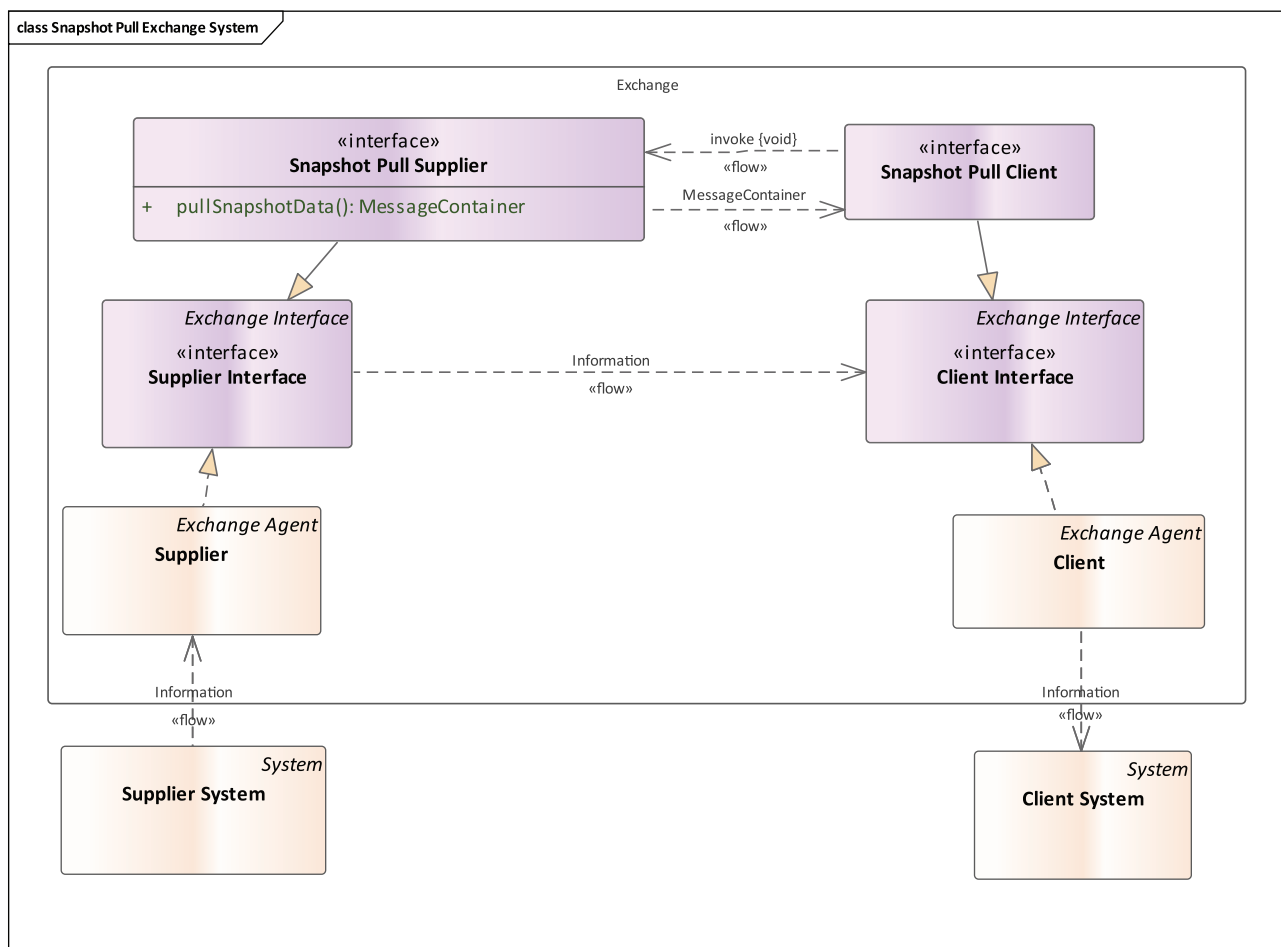


Figure 14 — Snapshot pull exchange subsystems, interface interactions and methods

The client takes the initiative to retrieve the data based on application level requirements which determine the needed exchange operating mode (e.g. on occurrence, condition-triggered or periodic).

6.2.3 Relevant exchange information in exchange data model

No exchange information is needed in this pattern to implement data delivery features. Nevertheless, a basic exchange data model (see in [Annex C](#)) has been provided to allow the implementation to deliver more than one payload content on the same message and further information to allow managing extra features not required by the snapshot pull exchange.

A “MessageContainer” instance should be retrieved using the basic exchange data model as reported in [Figure 14](#). Alternatively, a payload may be retrieved without any further information.

Information related to exchange that should be managed to make application development easier is fully described in the basic exchange data model.

Related exchange context information is:

- Supplier-related information
 - Requirement: supplier identification.

Related dynamic information is:

- Exchange DynamicInformation (provided both by the client and the supplier) wraps information such as exchangeStatus ("online", "offline", "undefined", etc.)
- Message generation timestamp information
 - Requirement: reliable information.

6.2.4 Exchange messages

- Payload message:
 - Payload messages should be delivered wrapped into a container (the basic exchange data model in [Annex C](#) with exchange data applies).
 - Payload messages may contain a payload update timestamp which can be used to understand when the payload has been created/updated for error management and processing saving.

6.3 State diagrams

State diagram are not needed and not developed for stateless FEP+EP as snapshot pull.

6.4 Features implementation description

6.4.1 Overview

This subclause provides a description and the corresponding specification for each feature identified in the context diagram, according to the snapshot pull exchange architecture. The following features are specified:

- subscription contract;
- subscription (also known as session);
- information management;
- data delivery;

— communication/protocol.

6.4.2 Subscription contract

6.4.2.1 Contract

Managed offline, not automated. It assumes information for controls to be implemented in the client to assess the identity of the supplier and authenticate the supplier request in message exchange.

6.4.2.2 Catalogue

Managed offline, not automated.

6.4.3 Session

6.4.3.1 Session life cycle

No session is managed for the current EP+FEP.

6.4.3.2 Link monitoring

Link monitoring is not managed for the current EP+FEP.

6.4.4 Information management

6.4.4.1 Operating modes

The available operating mode for client pull is Periodic, or On Occurrence (i.e. condition-triggered based on client). Pull-exchange is based on client-side conditions.

6.4.4.2 Update methods

The available update method is snapshot, i.e. retrieval of only currently valid data.

6.4.4.3 Life cycle management

Currently available information is included in the payload at a supplier system to prepare message delivery. It can be done at time-out on a cycle basis or at a specific triggering condition as “data updated” condition.

For life cycle management information, a snapshot includes all active information. Outdated information is not delivered in content.

For sampled information, the snapshot information contains the last sampled data available at the supplier site.

Whenever a condition is raised, the supplier system triggers it to the supplier to manage the creation of a payload pull delivery message (see [Figure 15](#)).

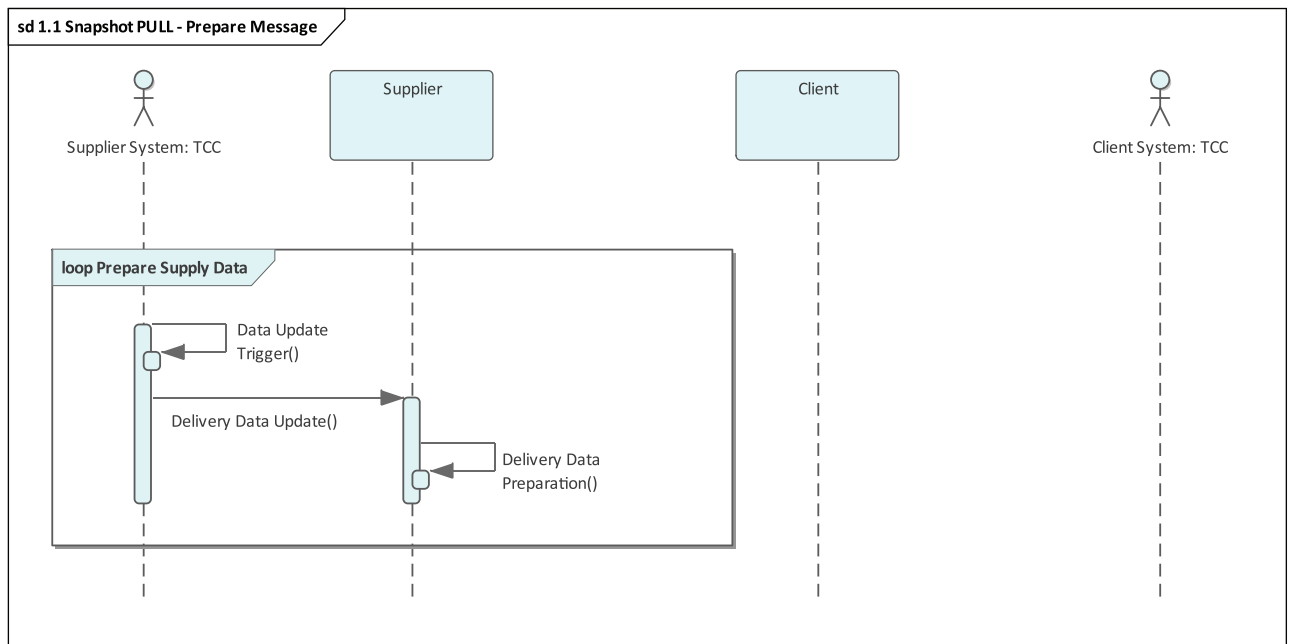


Figure 15 — Snapshot pull payload delivery creation: information management at supplier side

Information management for snapshot pull at client side is implemented as follows: when a client gets a snapshot of the last updated/created items it includes all valid active information. Information which had been delivered and which is not available in the last delivered payload shall not be considered after last delivery, i.e. it has been invalidated either as closed or cancelled information.

6.4.5 Data delivery

6.4.5.1 Data delivery

The sequence diagram for data delivery is as follows in [Figure 16](#):

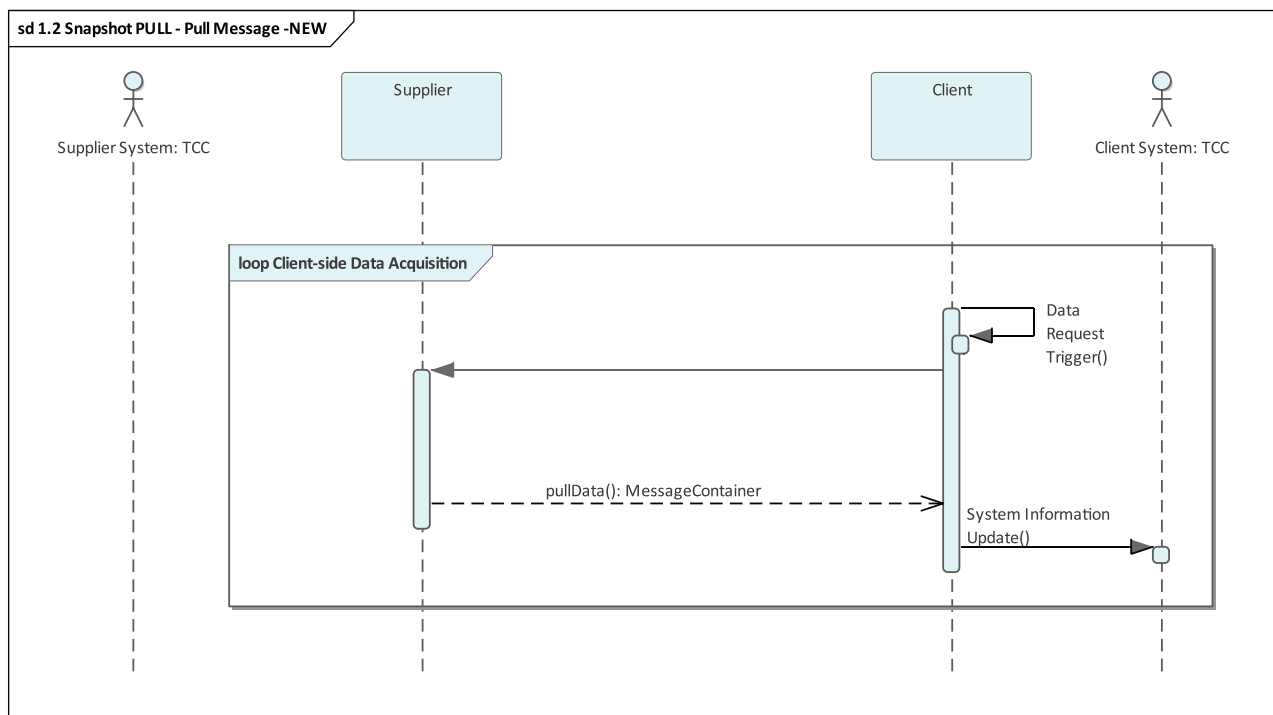


Figure 16 — Snapshot pull sequence diagram for data retrieval: implicit data delivery

When a pullSnapshotData request is triggered from the client to the interface method on supplier SnapshotPull interface, the corresponding snapshot payload(s) available shall be delivered as a return message by the supplier enclosed in a MessageContainer.

NOTE 1 Depending on implementation (e.g. http/get of XML static files generated at supplier side) the payload message is generated by the supplier based on conditions which are only managed by the supplier (e.g. event update or data gathered at the supplier side). In these cases, extra information can be available to implement some optimization, such as bandwidth saving, by not transferring the same data in case no update has been generated. This aspect will be described when applicable for PSM mapping.

NOTE 2 ExchangeInformation delivered in the return message by the supplier may contain any information which can be used to inform the client about client request processing which may be implemented in an optional pullRequest check. A processing for delivery based on client may be implemented by the supplier based on ExchangeInformation and offline subscription agreements. Any processing description for payload delivery based on client which can be implemented on supplier side based on any subscription agreement among client and supplier, are out of scope of this FEP+EP specification.

6.4.5.2 Data request

Not implemented in this pattern.

6.4.5.3 Large datasets handling

Not described in this pattern at PIM level (it may be implemented at PSM level as optimization; see [6.4.8](#)).

6.4.5.4 Synchronization

Implicit synchronization is available as only currently available elements are retrieved by snapshot pull.

6.4.6 Self-description

Handshake is not available.

6.4.7 Communication

Communication features are implemented at PSM level. They are relevant for the specific platform chosen on which the EP will be implemented (e.g. http/XML, Web Services SOAP, REST, etc.).

6.4.8 General optimization issues

Some EP features of any context diagram features groups (e.g. information management, data delivery etc.) allow the implementation of general optimization such as processing saving and bandwidth.

Payload timestamp information is available for client-side processing optimization made at the application level.

Pull message may be generated for all clients reducing processing resources at the supplier-side.

No extra optimization issues are considered in this EP+FEP.

7 Snapshot push

7.1 Overview

The "snapshot push" EP/FEP at PIM level is based on pushing information by the supplier to the client delivering a snapshot of information, i.e. all currently available information content, to the client. This EP can be implemented in several platforms: some examples are XML delivering of generated XML files by http/post, or a client exposing a SOAP web service method (e.g. named "PUSH") when currently available data can be sent by the supplier to the client.

This "snapshot push" does not manage session life cycle and link monitoring requirements, as well as synchronization. These features and related requirements are not considered in this pattern. Synchronization is not required as implicit when delivering snapshots of currently available information content.

To describe snapshot push EP/FEP at PIM level, all features are described in a general abstract format, independently from the specific technology platform in which this model will be implemented (e.g. http/get XML, WebService). [Table 8](#) shows a selection of features for snapshot push.

Table 8 — Selection of features for snapshot push

Features area	Feature	Snapshot push implemented
Subscription contract	Contract	N
	Catalogue	N
Session	Session life cycle	N
	Link monitoring	N
Information management	Operating modes	Periodic or On Occurrence (i.e. triggered by supplier condition)
	Update methods	Snapshot
	Life cycle management	Y, based on snapshot: new and updated content delivered, outdated data not delivered

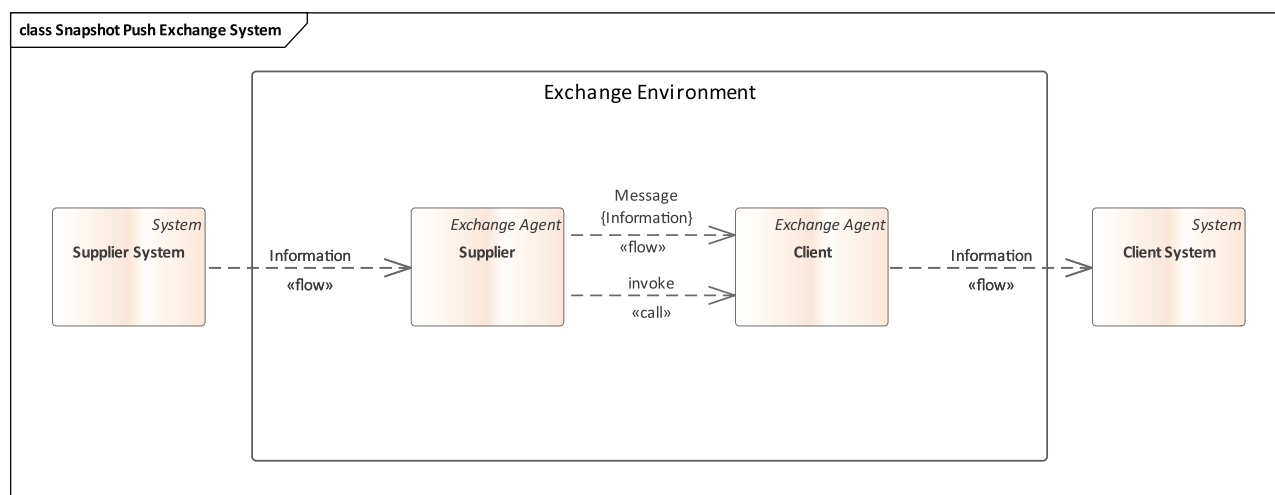
Table 8 (continued)

Features area	Feature	Snapshot push implemented
Data delivery	Data delivery	Y
	Data request	N
	Large datasets handling	N
	Synchronization	Y
Self-Description	Handshake	N
Communication	Security	To be defined at PSM level
	Compression	To be defined at PSM level
	Communication	To be defined at PSM level

7.2 Exchange pattern messages definition

7.2.1 Overall presentation

The information delivery business scenario description and definition imply that data exchange is needed to align the information kept by the supplier system into the client system; for this purpose, an exchange system is used which provides tools enabling message generation and their transfer between supplier and client (see [Figure 17](#)).

**Figure 17 — Snapshot push exchange actors**

The snapshot push EP is described in the following subclauses.

7.2.2 Basic exchange pattern

In a snapshot push context the client provides a mechanism to receive data from an action taken at a supplier site invoking specific resources / methods offered by the client.

The snapshot push client provides a mechanism to the snapshot push supplier to push currently available data, also called "snapshot" of information, i.e. current information at supplier system or last retrieved information for sampled data (see [Annex F](#)).

In the context of the "Snapshot Push" FEP +EP framework, to enable interoperability among client and supplier, all rules defined in this clause apply.

A snapshot push Client exchange system SHALL realise a snapshot push client interface which provides a putSnapshotData method.

A snapshot push supplier exchange system SHALL realise a snapshot push Supplier interface which invokes the putSnapshotData Method provided by the snapshot push Client interface to deliver snapshot data.

Figure 18 shows the communication diagram for Snapshot Push FEP + EP.

In this FEP+EP framework the supplier “pushes” messages to the client.

The client shall acknowledge the received message by a return information to the supplier. This return information shall be coded as ExchangeInformation.

NOTE This return message is available to bring some exchange information from the client to the supplier which can be used for any further exchange features implementations or application level checks or processing which are out of scope of this document.

The supplier takes the initiative to deliver the data based on application level requirements which determine the needed exchange operating mode (e.g. on occurrence or periodic).

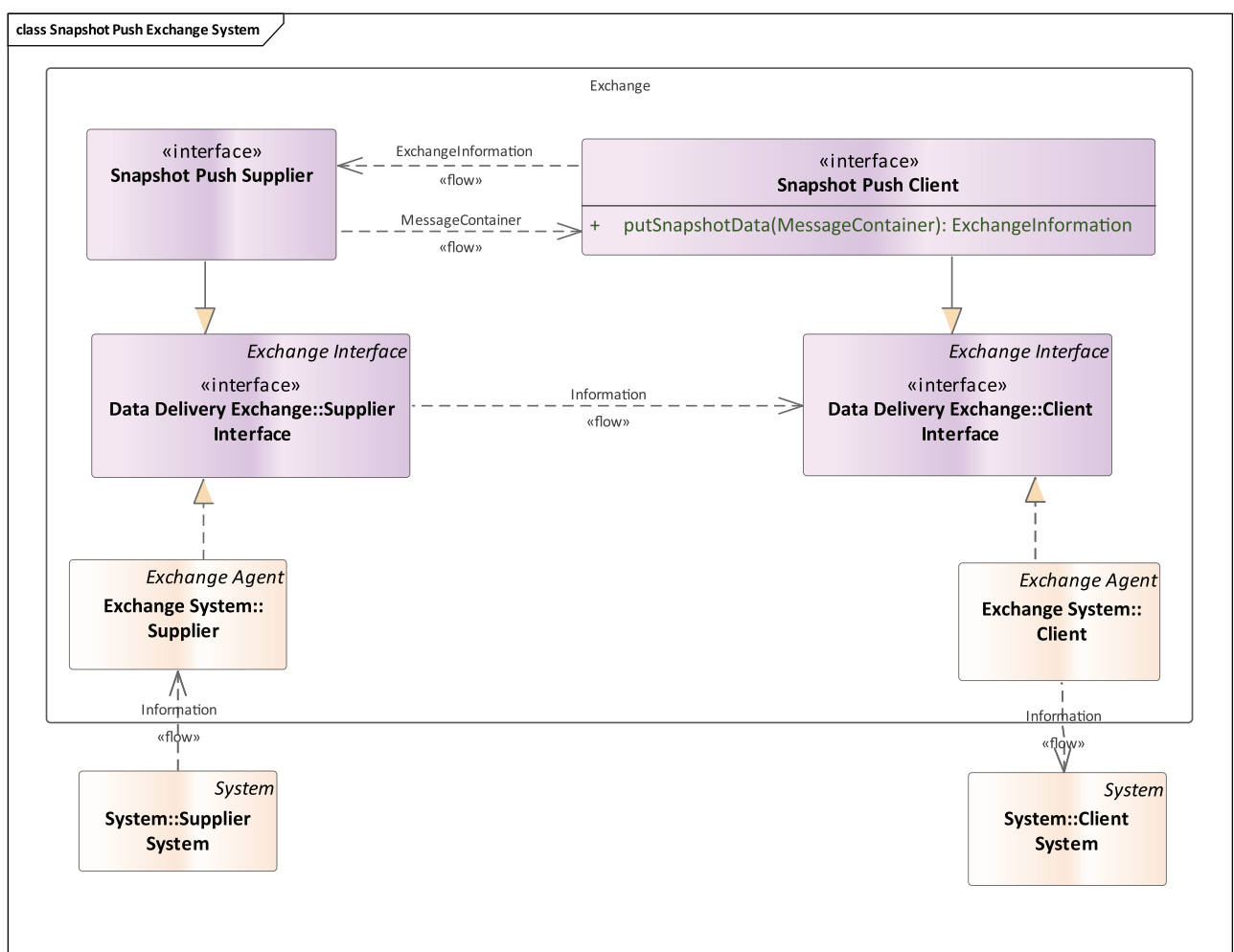


Figure 18 — Snapshot push exchange subsystems, interfaces interactions and methods

7.2.3 Relevant exchange information in exchange data model

No extra exchange information is needed in this pattern to implement any described features.

A basic exchange data model has been provided to allow the implementation to deliver more payload contents in the same message and further information to allow managing some extra features not

required by the basic snapshot push exchange. The usage of the exchange data model wrapping is for harmonization with other exchange patterns such as "simple push" or "stateful push".

A container should be retrieved using basic exchange data model as reported in the previous figure. Alternatively, a payload may be delivered.

An ExchangeInformation is returned to convey information about exchange operation and connection status.

Related exchange context information is:

- Supplier-related information
 - Requirement: supplier identification.
- Client-related information
 - Requirement: client identification.

Related dynamic information is:

- Exchange DynamicInformation (provided both by the client and the supplier) wraps information such as exchangeStatus ("Success", "Fail", "Close Session Request", "Snapshot Synchronization Request")
- Message generation timestamp information:
 - Requirement: timely, reliable information, session management.

7.2.4 Exchanged messages

- Payload message.
 - Simple payload messages can be exchanged within this FEP+EP.
 - Payload messages should be delivered wrapped into a container (see basic exchange data model in [Annex C](#)) with exchange data.
 - Payload messages contain payload update timestamp which can be used to understand when payload has been updated for error management and processing saving.

7.3 State diagrams

State diagrams are not needed and not developed for stateless FEP+EP as snapshot push.

7.4 Features implementation description

This subclause provides a description and the corresponding specification for each feature identified in the context diagram, according to the snapshot pull exchange architecture. The following features are specified:

- subscription contract;
- subscription (also known as session);
- information management;
- data delivery;
- communication/protocol.

7.4.1 Subscription contract

7.4.1.1 Contract

Managed offline, not automated. It assumes information for controls to be implemented in the client to assess the identity of supplier and authenticate the supplier request in messages exchange.

7.4.1.2 Catalogue

Managed offline, not automated.

7.4.2 Session

7.4.2.1 Session life cycle

No session is managed for the current EP+FEP.

7.4.2.2 Link monitoring

Link monitoring is not managed for the current EP+FEP.

7.4.3 Information management

7.4.3.1 Operating modes

Available operating mode for snapshot push is “Periodic”, or “On Occurrence” (i.e. condition-triggered based on supplier) push-based on supplier-side conditions.

7.4.3.2 Update methods

Available updated method is snapshot, i.e. retrieval of only currently valid data.

7.4.3.3 Life cycle management

Currently available information is included in the payload at a supplier system to prepare message delivery. It can be done at time-out on a cycle basis or at a specific triggering condition as “data updated” condition.

For life cycle management information, snapshots include all active information, outdated information is not delivered in content.

For sampled information, the snapshot information contains the last sampled data available at supplier site.

Information management for snapshot push is implemented as follows: when a client gets a snapshot of the last updated/created items, including all active items, it has to check for information which has been removed from the payload to deduce it has been invalidated (i.e. closed or cancelled).

7.4.4 Data delivery

7.4.4.1 Data delivery

For all clients with an active subscription, whenever a payload delivery condition is triggered, the supplier system shall manage the creation of a payload push message in a MessageContainer and shall deliver it by its SnapshotPush supplier interface to the corresponding SnapshotPush method available via the SnapshotPush client interface.

Figure 19 illustrates the sequence diagram which describes the interaction amongst the exchange supplier and its clients.

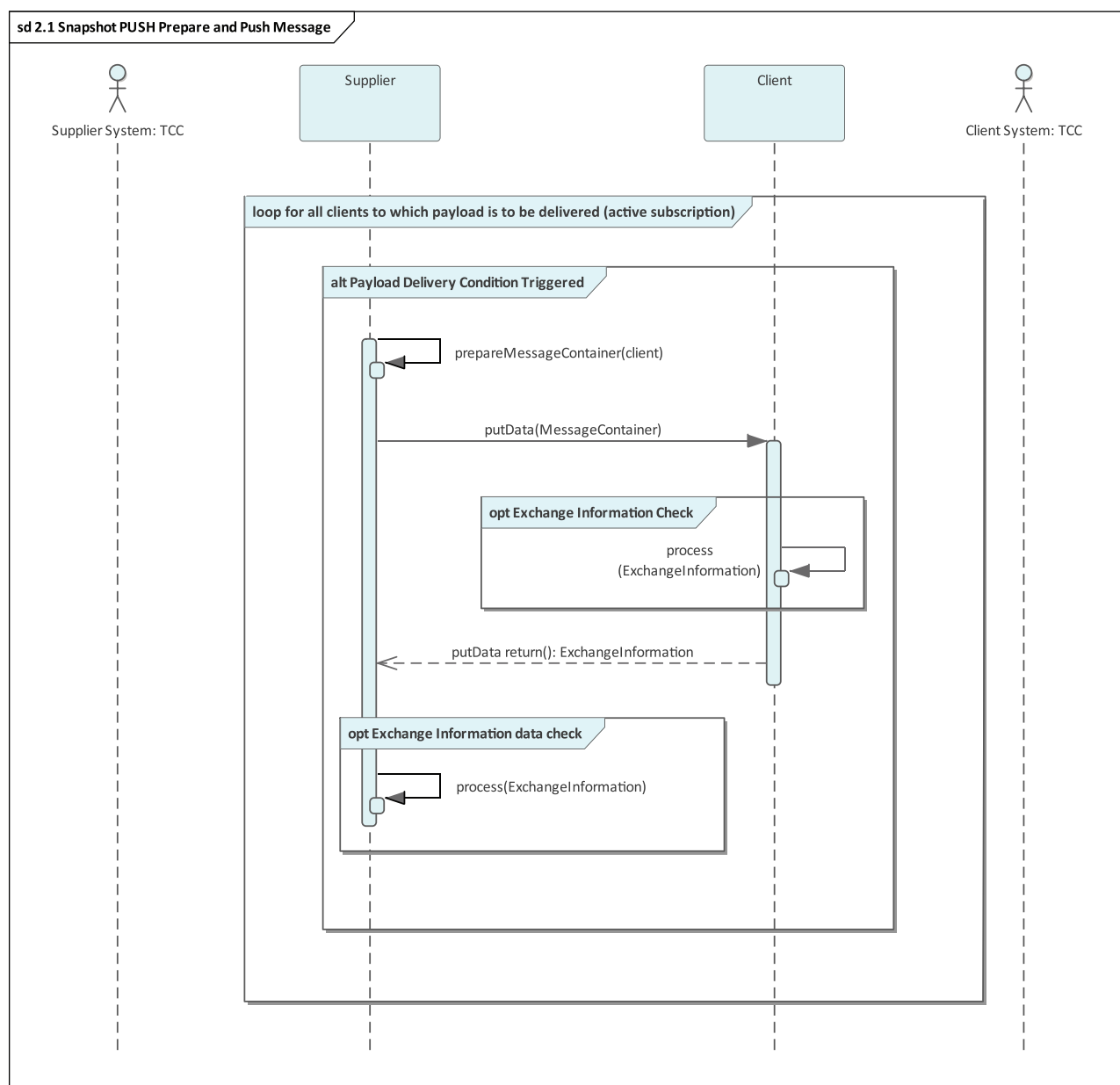


Figure 19 — Snapshot push sequence diagram: information management at supplier side

NOTE 1 Snapshot push message generation can create different a payload based on the client to which messages are to be delivered based on subscription information. Optional payload delivery processing based on ExchangeInformation and client subscriptions information are out of scope of this FEP+EP specification and are not described.

The client may verify the message delivered by the supplier and shall return an ExchangeInformation message to the supplier with information back to the supplier about the delivered message (i.e. return status and exchange status defined in basic exchange data model) which may be processed by the supplier to implement optional exchange checks.

NOTE 2 ExchangeInformation provided by the supplier and the client can optionally be checked to implement optional features enabled by the supplier and defined by the supplier and the client in the optional subscription process which is out of scope of this FEP+EP specification.

NOTE 3 ExchangeInformation delivered in the return message by the snapshot push client can contain any information which can be used to inform the snapshot push supplier about any client request processing, which can be implemented in an optional check. Optional processing to deliver payload information based on agreements among client and supplier can be implemented by the supplier based on offline subscription agreements. These processing descriptions and specifications are out of scope of this FEP+EP specification.

7.4.4.2 Data request

Not implemented in this pattern.

7.4.4.3 Large datasets handling

Not described in this pattern at PIM level. May be implemented at PSM level (see [7.4.7](#)).

7.4.4.4 Synchronization

Implicit synchronization is available as only currently available elements are retrieved by snapshot push.

7.4.5 Self-description

Handshake is not available.

7.4.6 Communication/protocol

Communication features are implemented at PSM level. They are relevant for the specific platform chosen on which the EP will be implemented (e.g. http/XML, Web Services SOAP, REST, etc.).

7.4.7 General optimization issues

Some EP features of any context diagram features groups (e.g. information management, data delivery, etc.) allow the implementation of general optimization such as processing-saving and bandwidth.

Payload timestamp information is available for client-side processing optimization made at the application level.

A push message may be generated for all clients reducing processing resources at the supplier side.

No extra optimization issues are considered in this EP+FEP.

8 Simple push

8.1 Overview

The "simple push" EP/FEP at PIM level is based on information messages sent or pushed by the supplier to a client. It can be implemented in several platforms: some examples are push of generated XML content by http/post, or a client providing a SOAP web service method "push" by which data can be provided by the supplier to the client.

This "simple push" adds extra features to the basic snapshot push EP to manage link monitoring, as well as synchronization/realignment in case of communication lacks or system maintenance. This mechanism will be detailed at PIM level in the following subclauses.

To describe the EP/FEP at PIM level all features are described in a general abstract format, independently from the specific technology platform in which this model will be implemented. (e.g. http/get XML, WebService).

Table 9 — Selection of features for simple push

Features area	Feature	Simple push available
Subscription contract	Contract	N
	Catalogue	N
Session	Session life cycle	N
	Link monitoring	Y
Information management	Operating modes	Periodic or On Occurrence based on triggering of supplier condition
	Update methods	Snapshot, Single Element Update, All Element Update
	Life cycle management	Y
Data delivery	Data delivery	Y
	Data request	N
	Large datasets handling	Y, optional
	Synchronization	Y, optional
Self-Description	Handshake	N
Communication	Security	At PSM level
	Compression	At PSM level
	Communication	At PSM level

8.2 Exchange pattern messages definition

8.2.1 Overall presentation

The information delivery business scenario description and definition state that data exchange is needed to align the information kept by the supplier system into the client system; for this purpose, an exchange system is used which provides tools enabling message generation and their transfer between supplier and client (see [Figure 20](#)).

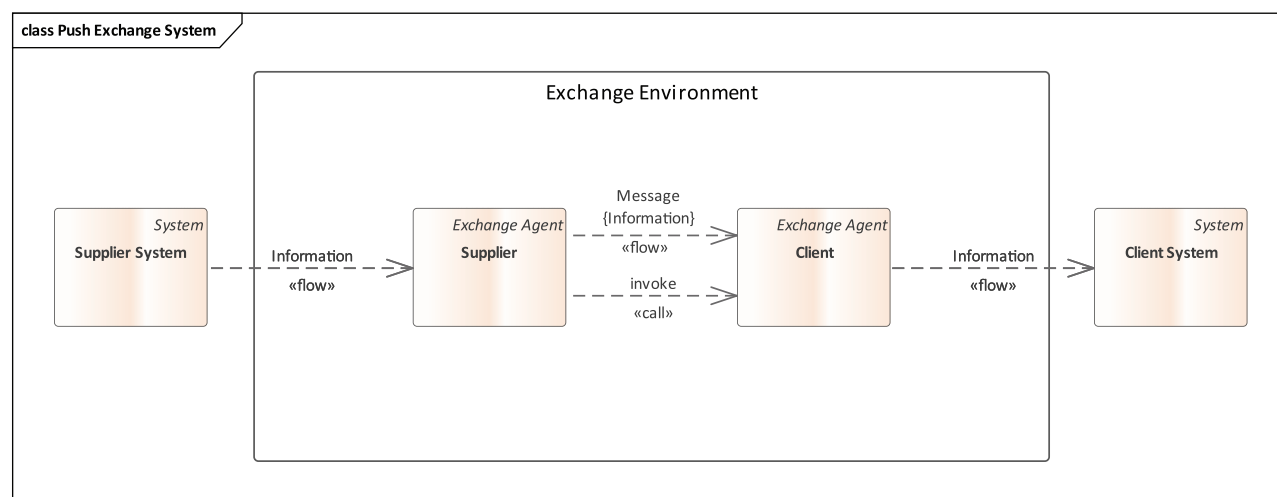


Figure 20 — Simple push exchange actors

The simple push EP is described in the following subclauses.