

DATEX II

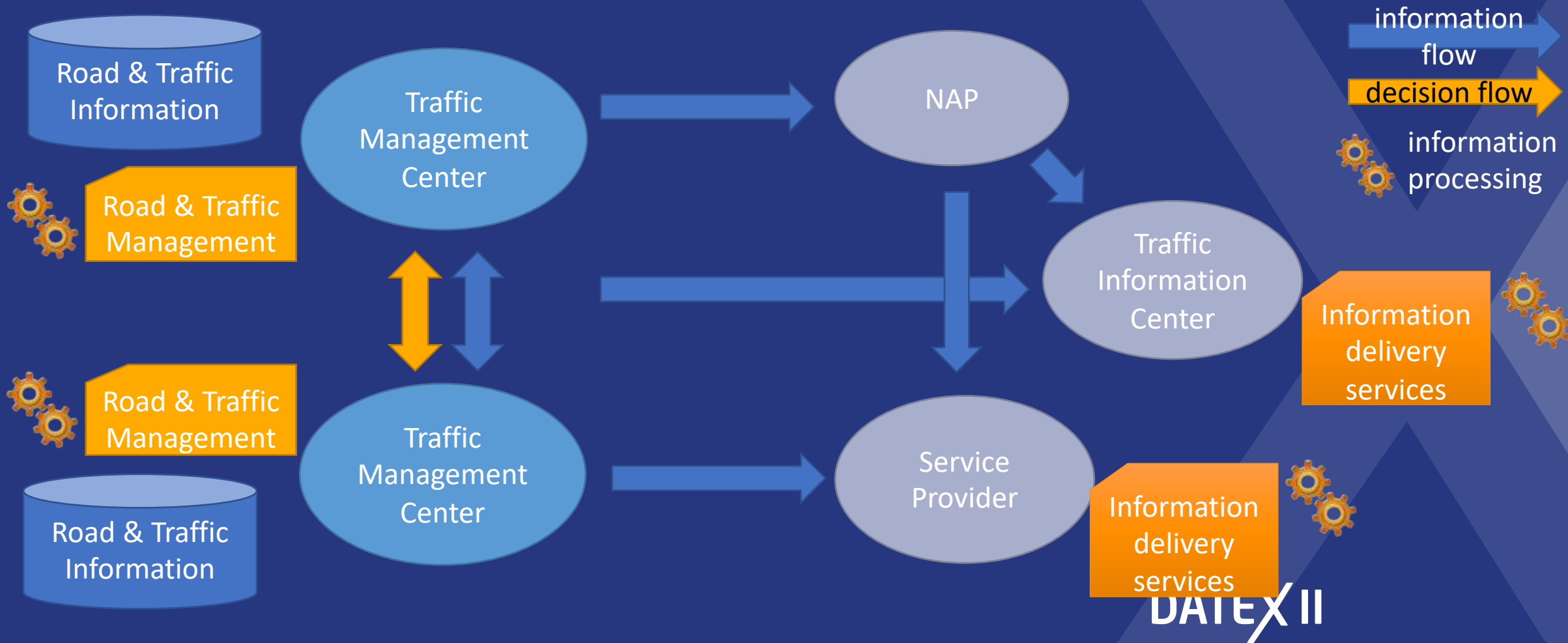
Exchanging DATEX II encoded content

Exchange 2020 specification release

6TH FORUM
WEBINAR SERIES

Please ask your questions in the Q&A

Exchange Specification supporting DATEX II



Exchange Specification Rationale

- **DATEX II specification set (EN 16157 family series)** refer to encoding Payload Content: **Road and Traffic Data**
 - **Payload content** may be shared simply as XML file on http server
 - **«Basic» Exchange Specification** available for DATEX II since v1.0 use HTTP/get to retrieve an «information snapshot».
 - all valid content is delivered in any message, independently of previously retrieved or updated content.
- **DATEX II Exchange Specs v1.0 → v2.x**
 - Fully working for http/get «snapshot» of information
 - Easy, robust
 - Lack of requirements for all use cases (error management, session management, bandwidth saving, timely delivery, etc)
 - Push and Pull SOAP WebServices specification description not completed
 - Realignment synchronisation via not harmonised specification
 - Further requirements needed for Traffic Management features use cases

<https://docs.datex2.eu/exchange/2020/rationale/index.html> [link](#)

DATEX II

6th Forum Webinar series



Exchange 2020 Specification

- Exchange specification analysis led to several solutions based on different requirement selection for the different use cases
 - **several Exchange Pattern** (Snapshot Pull, Simple Push, etc)
Used for
 - **distinct Business Scenarios**
- **Standardised by CEN/ISO with support of DATEX II organisation**
 - **TS 19468 – Platform independent model specifications for data exchange protocols for transport information and control systems**
 - **TS 14827-4 – Data interfaces between centers for Intelligent transport system (Profile B)**

Note: previous Exchange2018 specification are deprecated

DATEX II

6th Forum Webinar series





Information Exchange among Centres

Objectives



Information Delivery

What happens ?
How I manage

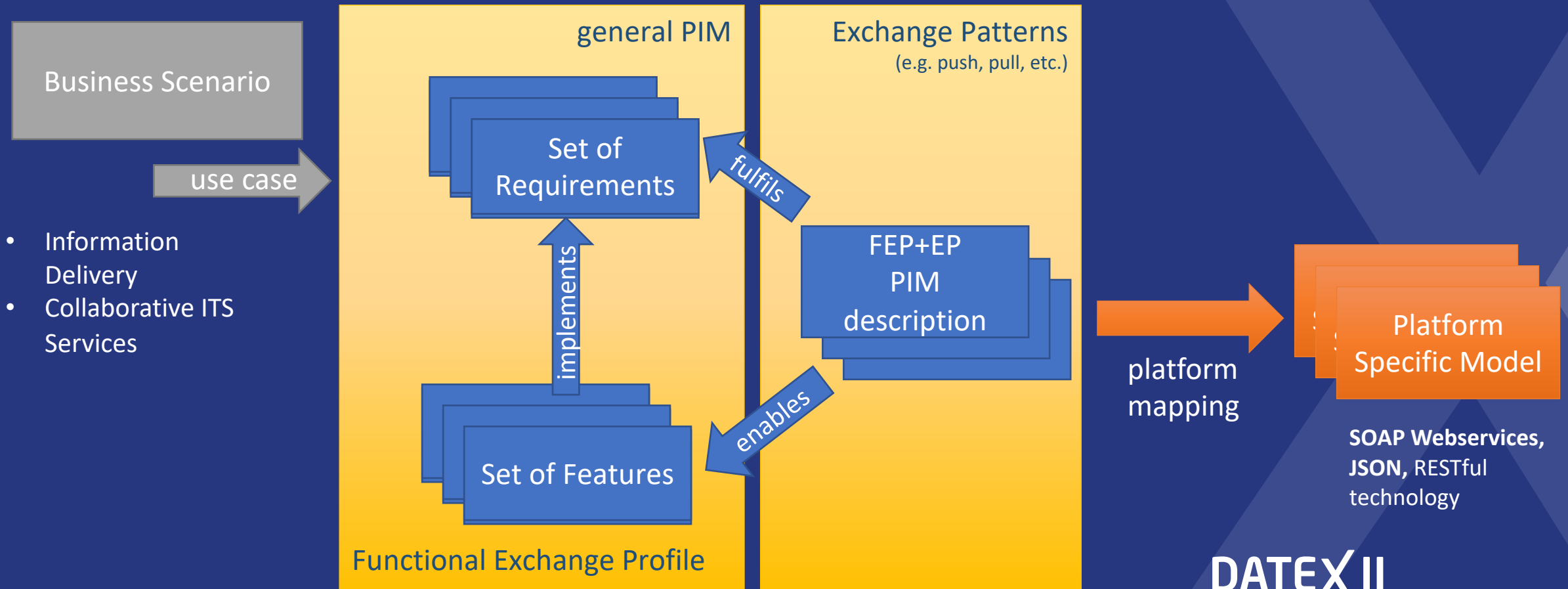
Collaborative Traffic and Road Management

How to deal ? Improving Safety, Environment, Comfort

DATEX II



Model Driven Approach



DATEX II

Requirements & Features

- Stateless Sessionless Exchange vs Stateful and session managed Exchange
- Snapshot update (all information exchanged)
- Delta update (only exchange updated information)
- Implementation costs Supplier side or Client Side
- Etc.

DATEX II

6th Forum Webinar series



Exchange Patterns selection

Operating modes

- Pull → Client initiated → Client retrieves data from supplier
- Push → Supplier initiated → Supplier deliver data to client

Update method

- Snapshot update → all available information is delivered always
- Incremental update → only updated information is delivered

DATEX II

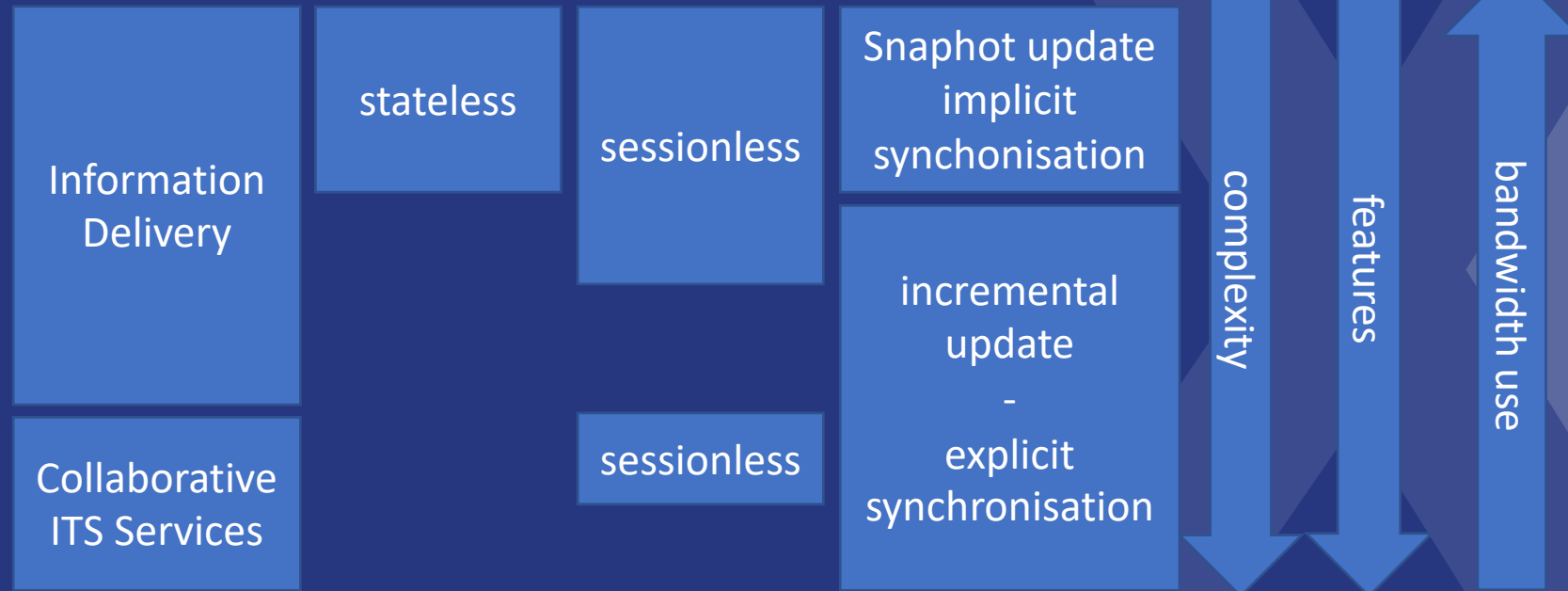
6th Forum Webinar series



Exchange Patterns and Functional Exchange Profiles

Selected EP+FEP to be used for DATEX II

- Snapshot Pull
- Snapshot Push
- Simple Push
- Stateful Push
- Simple CIS
- Stateful CIS



DATEX II

6th Forum Webinar series

FEP+EP selection : use case driven

- Use case driven Requirement selection

See [Exchange Guide](https://docs.datex2.eu/exchange/2020/userguide/index.html)

<https://docs.datex2.eu/exchange/2020/userguide/index.html>

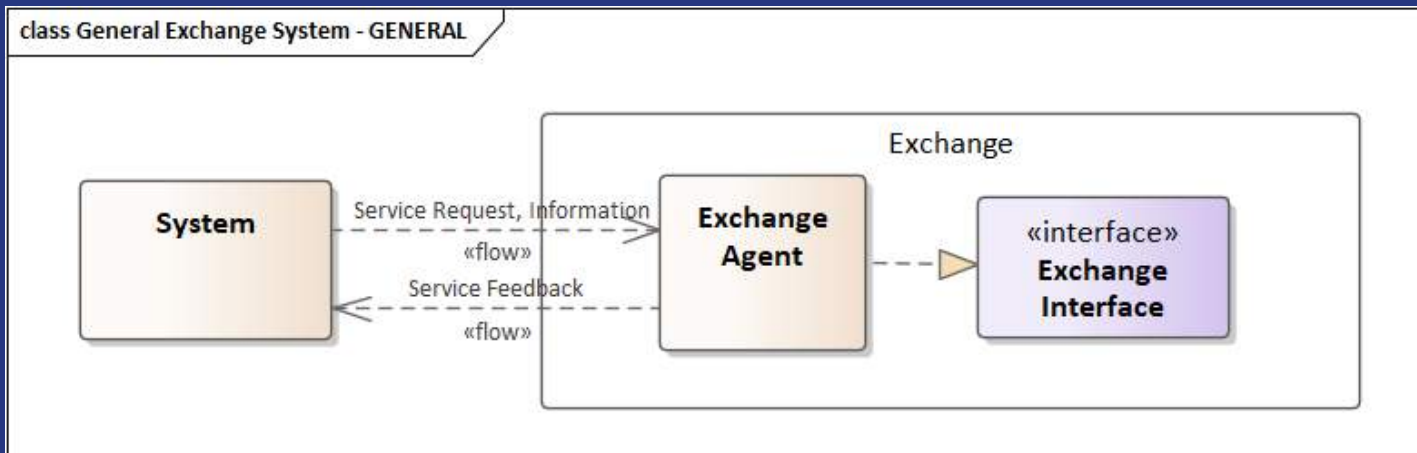
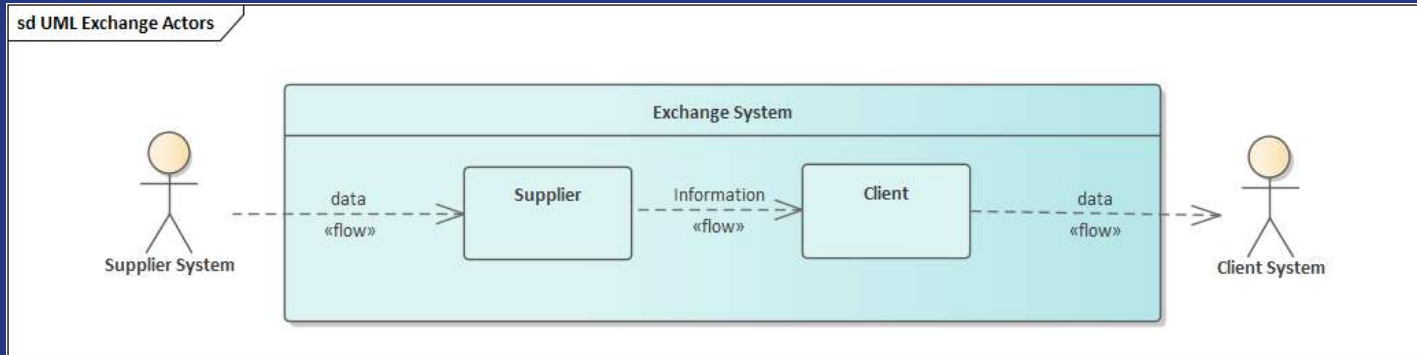
| Use Case | Business Scenario | Available Exchange FEP+EP |
|---|----------------------------|-----------------------------------|
| Delivery of Road Traffic Data from Sensors for standard processing | Data Delivery | Snapshot Pull Snapshot Push |
| Delivery of Road Traffic Data from Sensors for real time processing | Data Delivery | Simple Push (Snapshot + Delta) |
| Delivery of Road Traffic Data for ITS delivery Services such broadcast or VMS Setting or Traffic Management | Data Delivery | Simple Push Stateful Push |
| Delivery of Road Traffic Data for Traffic Management / VMS Setting | Collaborative ITS Services | Simple CIS Stateful CIS |

DATEX II

6th Forum Webinar series



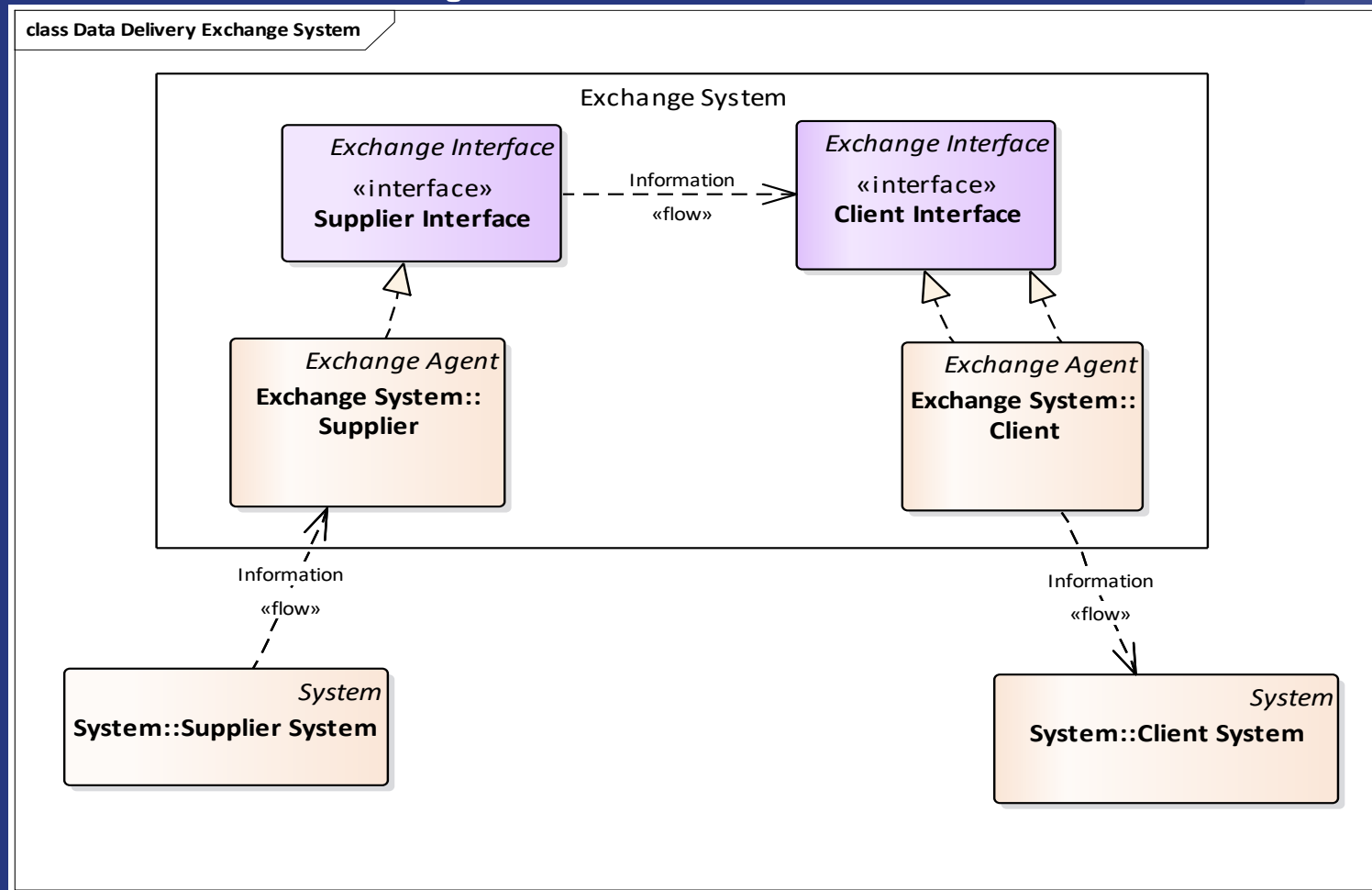
Exchange Pattern description via UML Interface Diagram



DATEX II

6th Forum Webinar series

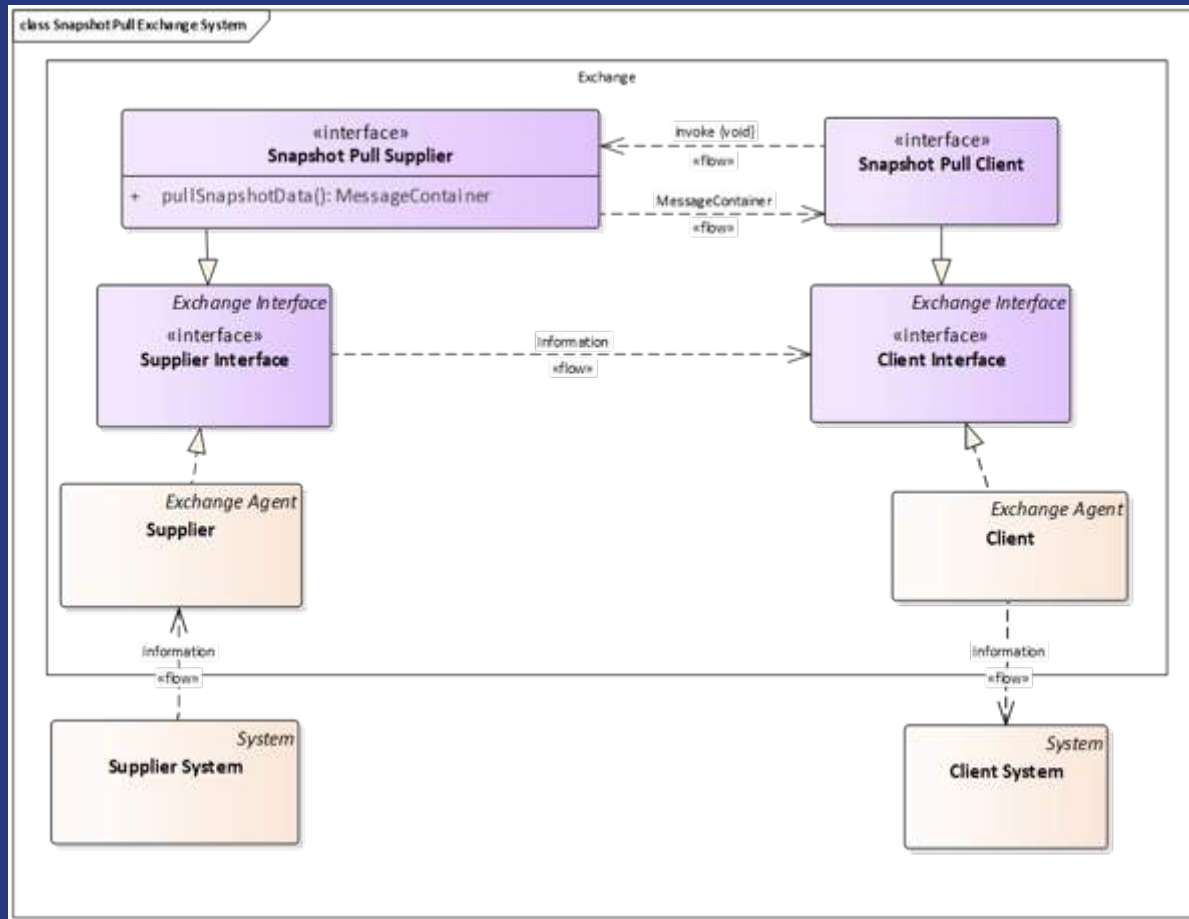
Interface Description Information Delivery



TEX II

6th Forum Webinar series

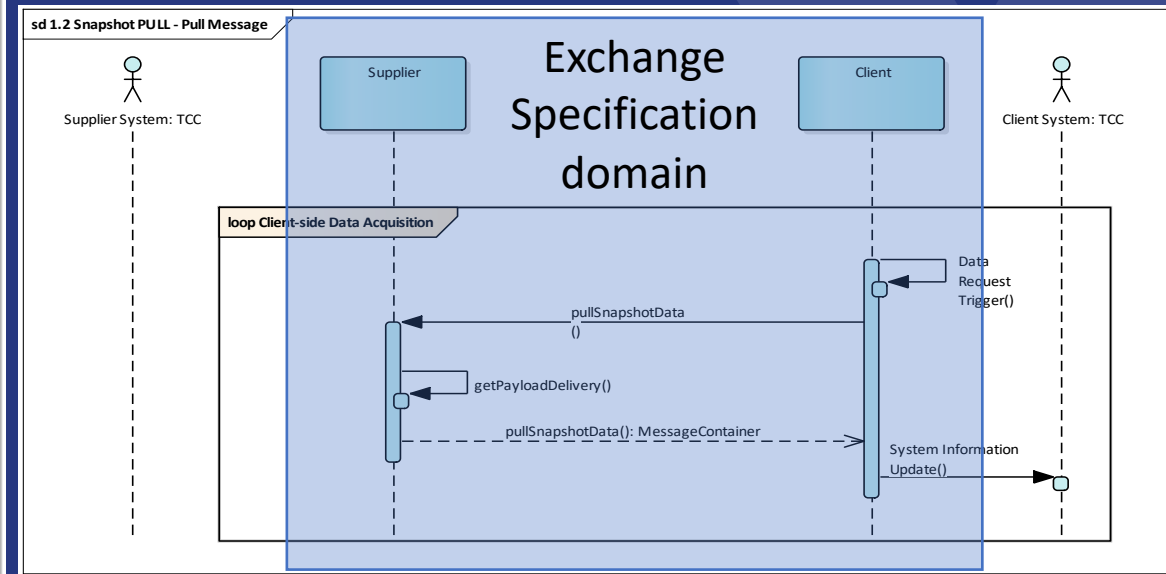
Snapshot Pull specialised Interface description



Client retrieves a snapshot of information, implicit synchronisation

1 simple method, void input

- pullSnapshotData



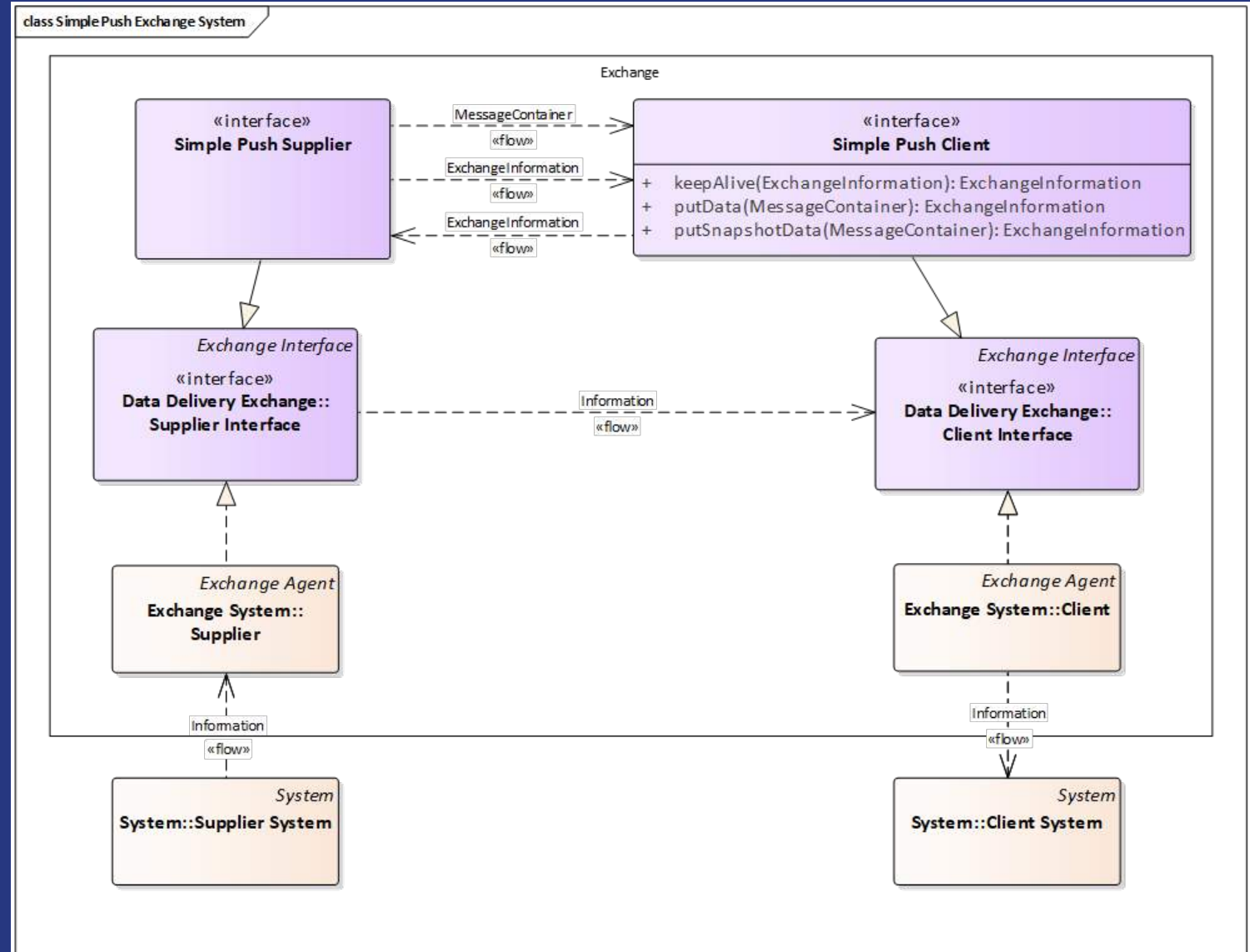
DATEX II

6th Forum Webinar series

Simple Push Interface Description

3 methods

- **putSnapshotData**
 - Explicit synchronisation vs implicit synchronisation in Snapshot Exchange Pattern
- **putData**
 - Depending on use case
 - All updated information, e.g. Measured data, Travel times
 - Single updated elements, e.g. situation, VMS Status, etc.
- **keepAlive**

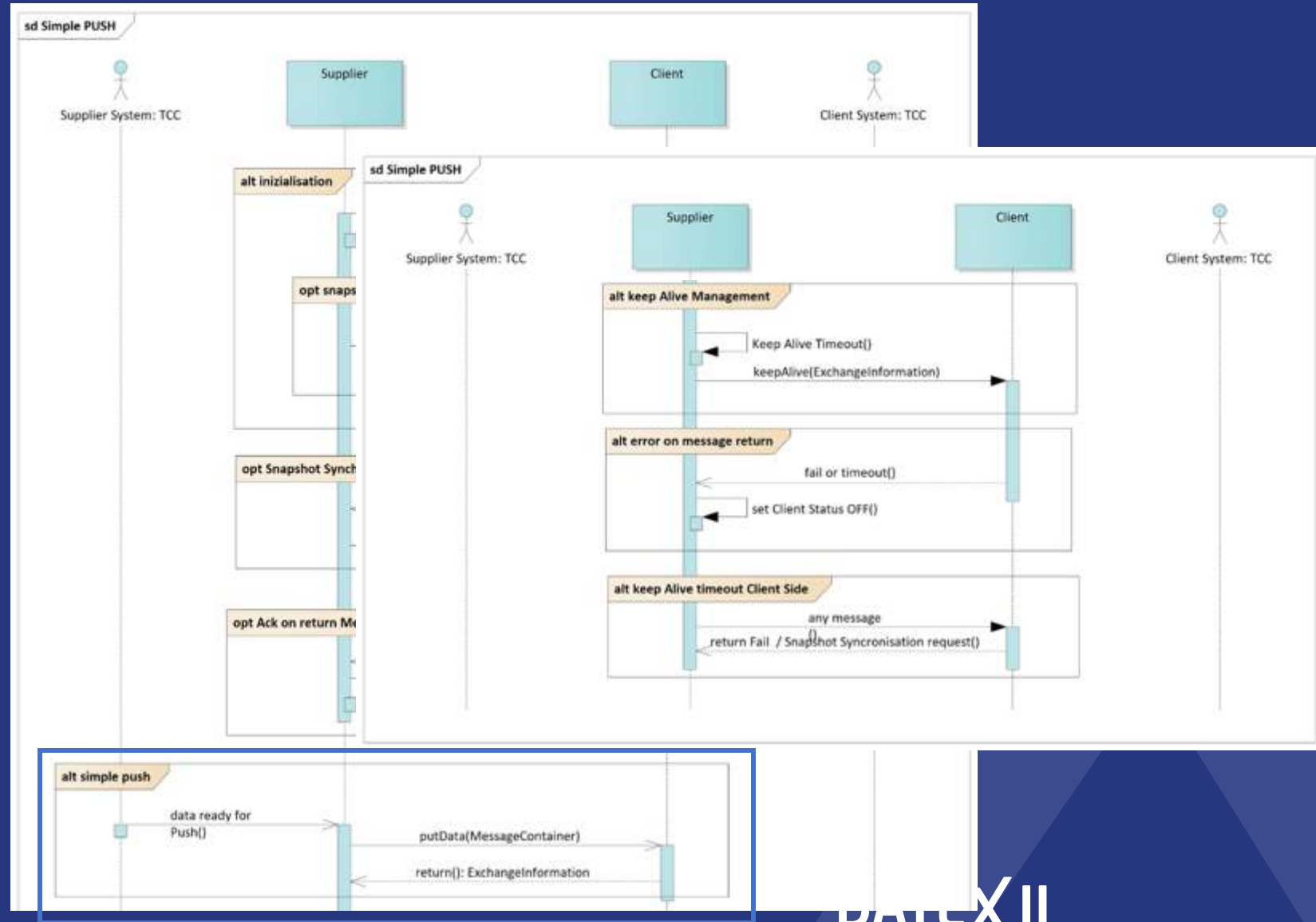


DATEX II

6th Forum Webinar series

Simple Push Sequence Diagrams

Supports the features implementation description

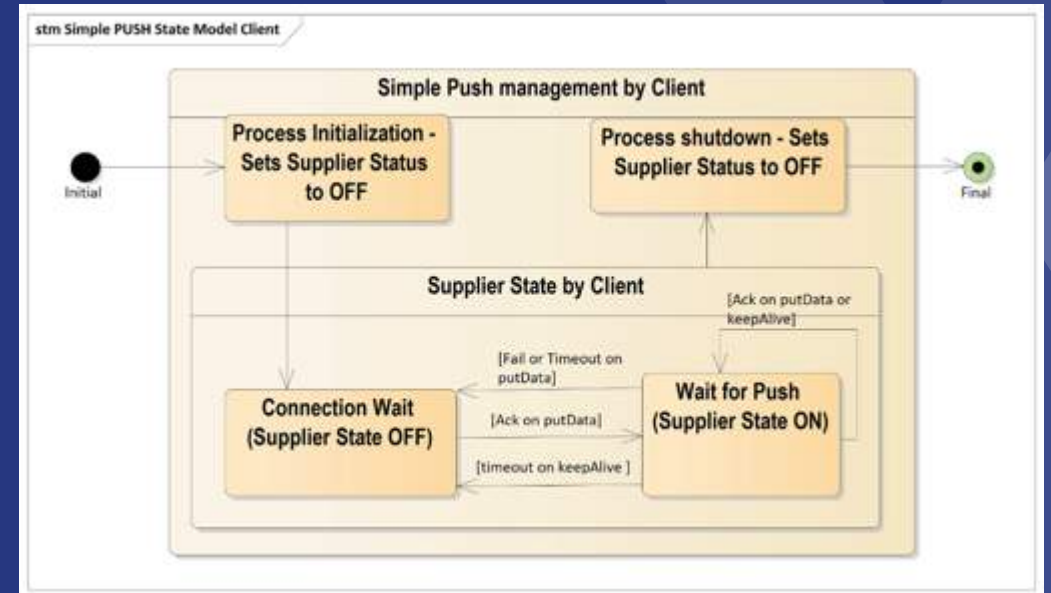
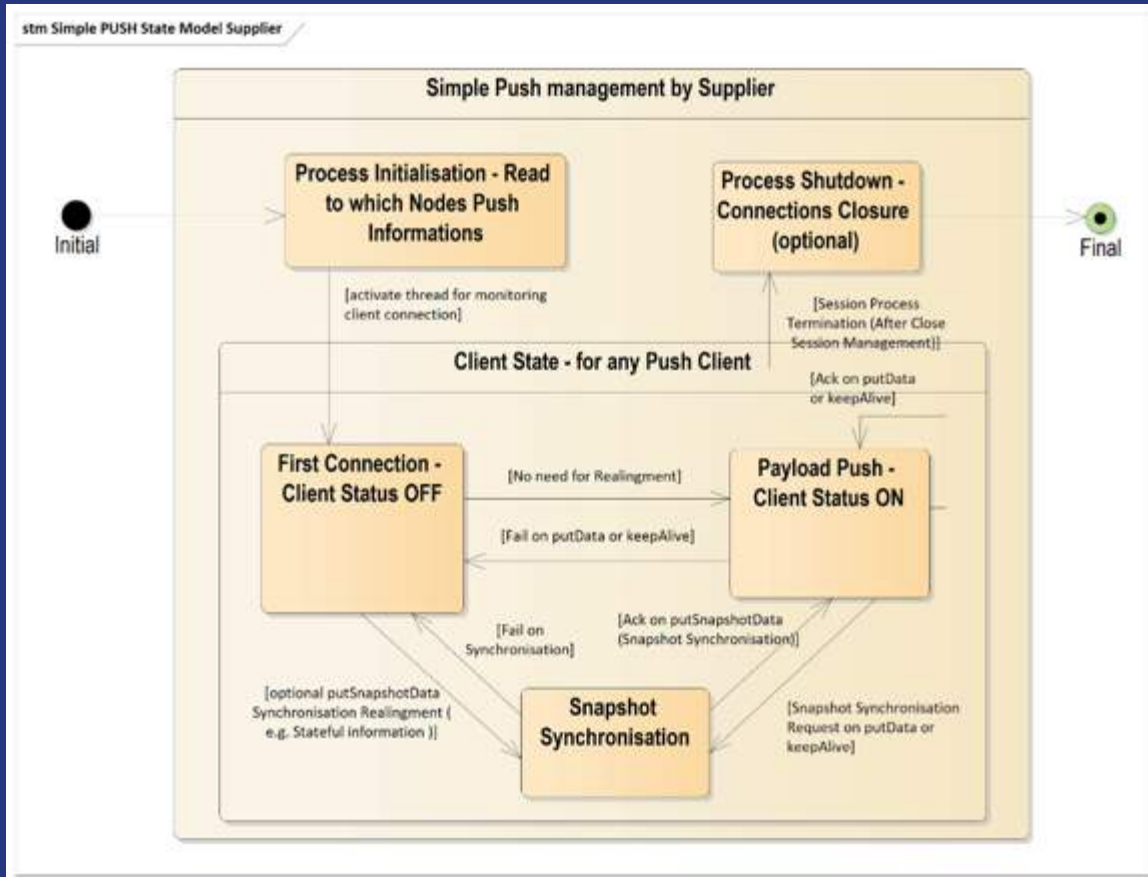


DATEX II

6th Forum Webinar series

Simple Push State Diagrams

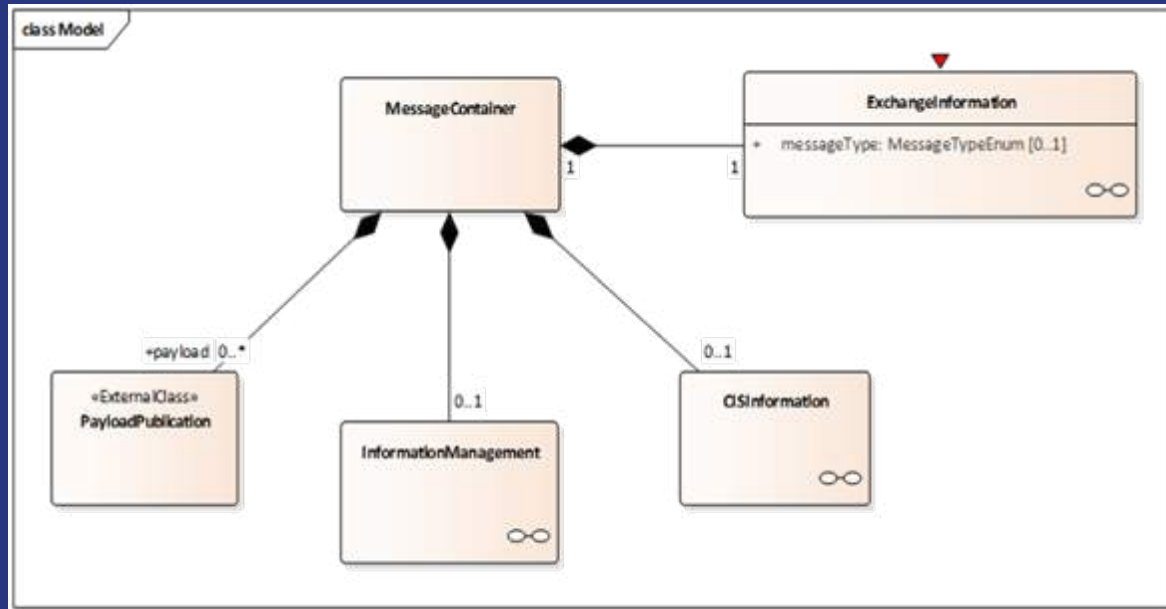
Supports the features implementation description



DATEX II

6th Forum Webinar series

Basic Exchange Data Model



Depending on the **Exchange Pattern**, data are needed to manage and enable different **exchange features**.

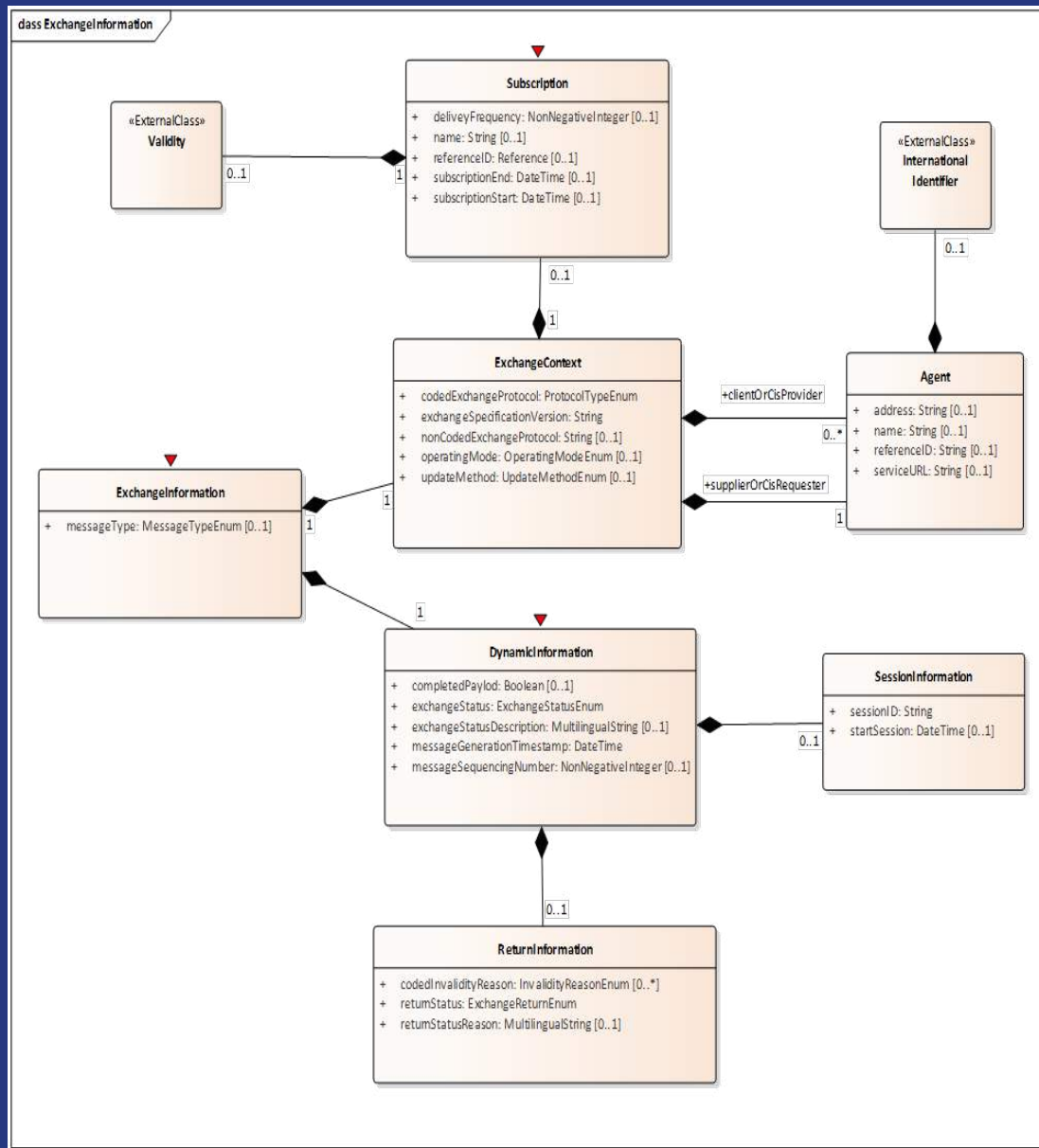
As DATEX II payload is based on Road Traffic Data a **MessageContainer structure** had been defined to wrap Payload to allow implement exchange features

This BEDM model had been designed using same DATEX II Modelling Methodology which may derive XSD, ASN.1, JSON schema encoding for exchange Data

DATEX II

6th Forum Webinar series

Exchange Information



Static Information

Dynamic Information

ATEX II

6th Forum Webinar series

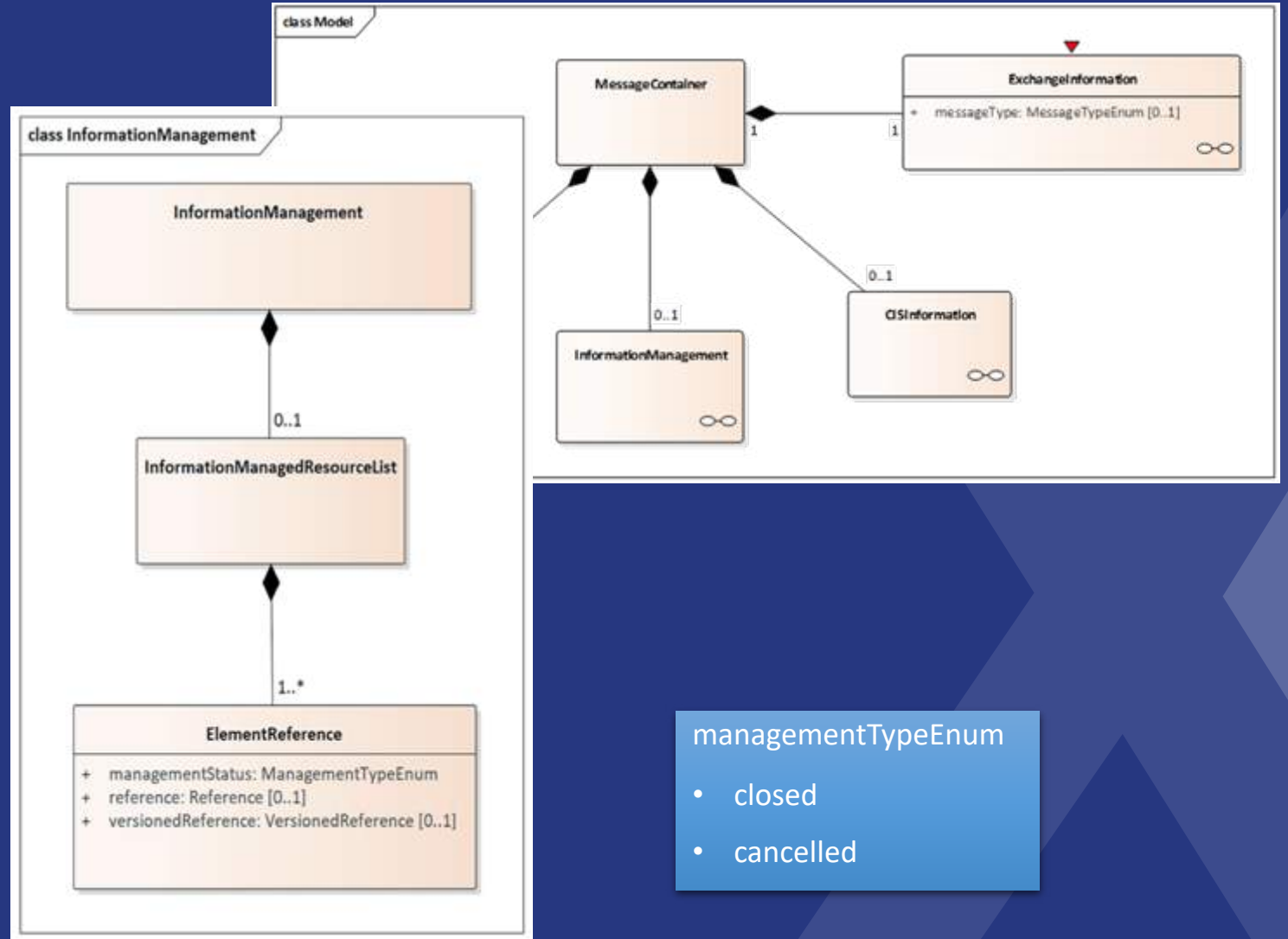


Information Management

Used for Simple and Stateful Push

Single element update operating mode

which may be combined with Snapshot Synchronisation



managementTypeEnum

- closed
- cancelled

DATEX II

6th Forum Webinar series

Exchange Data Profile

- The BEDM manages data which can be used to enable several features, the design principle is extensive.
- **Profiling Exchange Data** for the specific **Exchange Pattern** reduces implementation complexity and enable optimised exchange management.
 - Snapshot Pull FEP+EP only include Payload and reduced **ExchangeInformation**
 - Simple and Stateful Push include **InformationManagement** besides ExchangeInformation
 - Stateful Push include dynamic **SessionInformation** which is not needed in SimplePush and Snapshot Push and Pull.
- Profiled xsd for all FEP+EP are available on documentation website
- DATEX II Webtool embed functions to enable profiled exchange selection will be delivered.

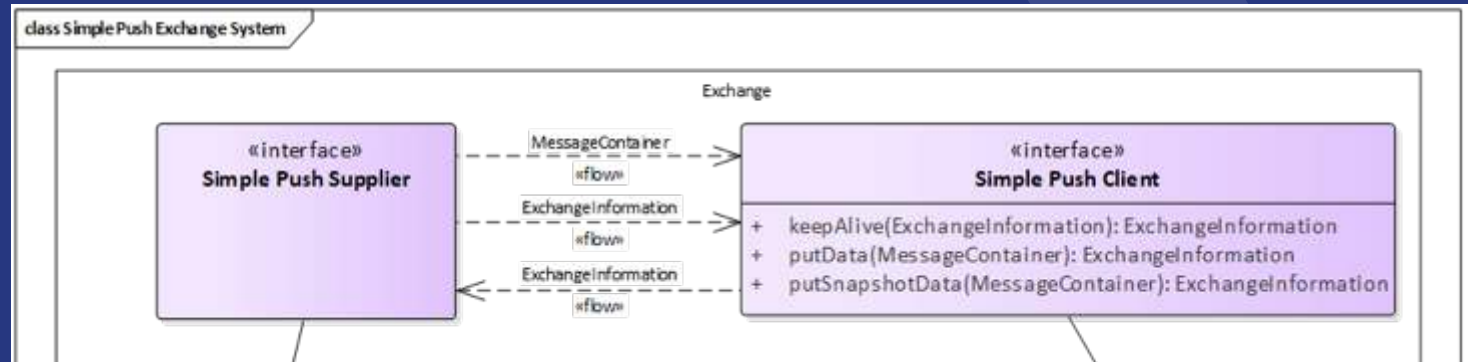
DATEX II

6th Forum Webinar series



PSM mapping for Web Services

- UML Interface Diagram Methods are mapped to WSDL methods
- Message Type definition are defined using BEDM Data Structures generated xsd



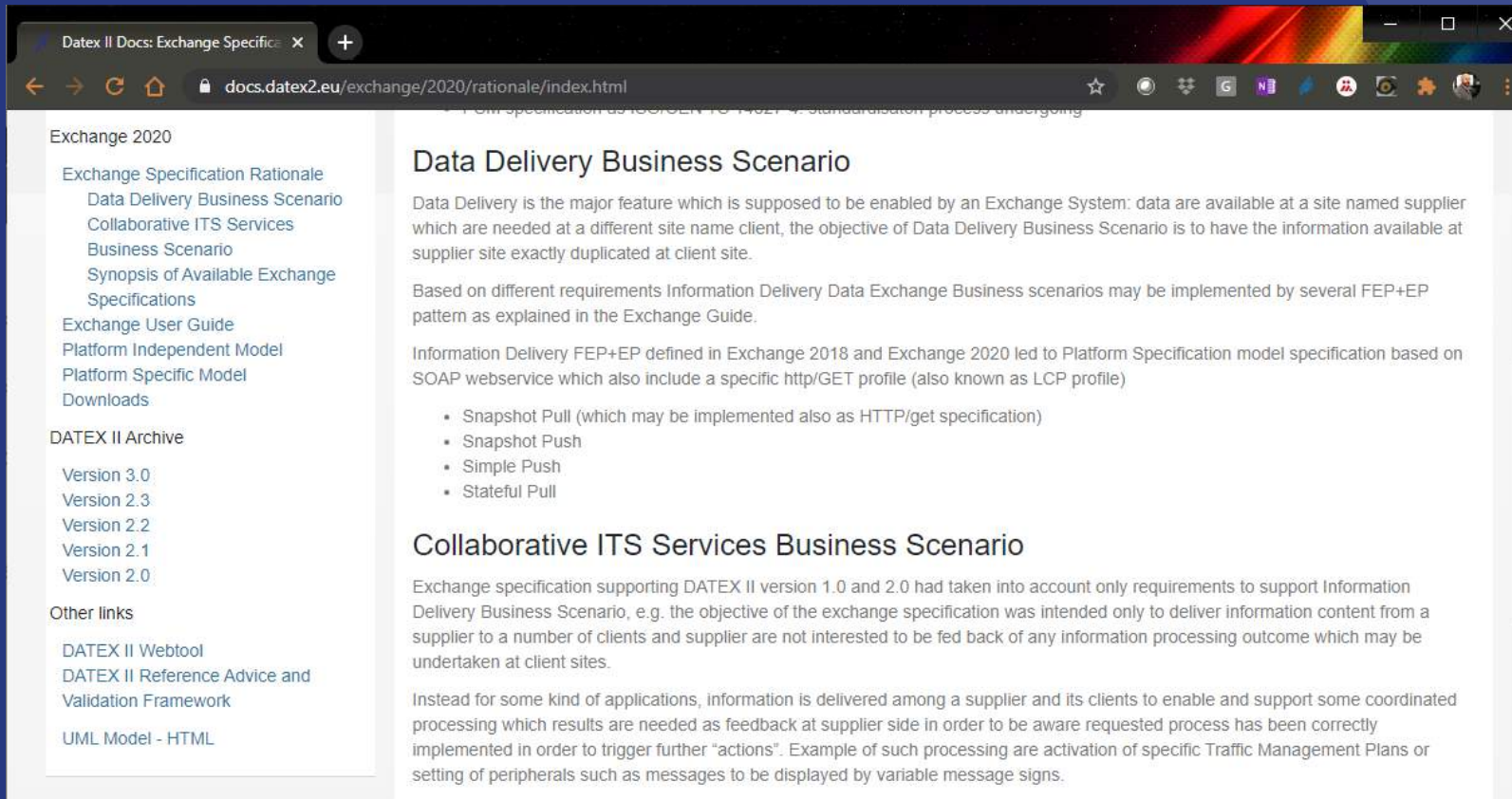
```

<portType name="simplePushInterface">
  <operation name="putData">
    <input message="tns:putData" type="con:MessageContainer" />
    <output message="tns:putDataOutput" type="ex:ExchangeInformation" />
  </operation>
  <operation name="keepAlive">
    <input message="tns:keepAlive" type="ex:ExchangeInformation" />
    <output message="tns:keepAliveOutput" type="ex:ExchangeInformation" />
  </operation>
  <operation name="putSnapshotData">
    <input message="tns:putSnapshotData" type="con:MessageContainer" />
    <output message="tns:putSnapshotDataOutput" type="ex:ExchangeInformation" />
  </operation>
</portType>
  
```

```

<types>
  <xs:schema targetNamespace="http://datex2.eu/wsdl/simplePush/2020">
    <xs:import namespace="http://datex2.eu/schema/3/messageContainer" schemaLocation="http://datex2.eu/schema/3/messageContainer.xsd" />
    <xs:import namespace="http://datex2.eu/schema/3/exchangeInformation" schemaLocation="http://datex2.eu/schema/3/exchangeInformation.xsd" />
    <xs:element name="putDataInput" type="con:MessageContainer" />
    <xs:element name="putDataOutput" type="ex:ExchangeInformation" />
    <xs:element name="putSnapshotDataInput" type="con:MessageContainer" />
    <xs:element name="putSnapshotDataOutput" type="ex:ExchangeInformation" />
    <xs:element name="keepAliveInput" type="ex:ExchangeInformation" />
    <xs:element name="keepAliveOutput" type="ex:ExchangeInformation" />
  </xs:schema>
</types>
  
```

Exchange Documentation available



The screenshot shows a web browser window with the URL docs.datex2.eu/exchange/2020/rationale/index.html. The page content is as follows:

Exchange 2020

- Exchange Specification Rationale
 - Data Delivery Business Scenario
 - Collaborative ITS Services Business Scenario
 - Synopsis of Available Exchange Specifications
- Exchange User Guide
- Platform Independent Model
- Platform Specific Model
- Downloads

DATEX II Archive

- Version 3.0
- Version 2.3
- Version 2.2
- Version 2.1
- Version 2.0

Other links

- DATEX II Webtool
- DATEX II Reference Advice and Validation Framework
- UML Model - HTML

Data Delivery Business Scenario

Data Delivery is the major feature which is supposed to be enabled by an Exchange System: data are available at a site named supplier which are needed at a different site name client, the objective of Data Delivery Business Scenario is to have the information available at supplier site exactly duplicated at client site.

Based on different requirements Information Delivery Data Exchange Business scenarios may be implemented by several FEP+EP pattern as explained in the Exchange Guide.

Information Delivery FEP+EP defined in Exchange 2018 and Exchange 2020 led to Platform Specification model specification based on SOAP webservice which also include a specific http/GET profile (also known as LCP profile)

- Snapshot Pull (which may be implemented also as HTTP/get specification)
- Snapshot Push
- Simple Push
- Stateful Pull

Collaborative ITS Services Business Scenario

Exchange specification supporting DATEX II version 1.0 and 2.0 had taken into account only requirements to support Information Delivery Business Scenario, e.g. the objective of the exchange specification was intended only to deliver information content from a supplier to a number of clients and supplier are not interested to be fed back of any information processing outcome which may be undertaken at client sites.

Instead for some kind of applications, information is delivered among a supplier and its clients to enable and support some coordinated processing which results are needed as feedback at supplier side in order to be aware requested process has been correctly implemented in order to trigger further "actions". Example of such processing are activation of specific Traffic Management Plans or setting of peripherals such as messages to be displayed by variable message signs.

<https://docs.datex2.eu/exchange/2020/> [URL Link](#)

DATEX II

6th Forum Webinar series

Exchange 2020 demonstrator

- Proof of concept of WSDL implementation by Java SOAP, JAXB, SOAPUI.
- It shows dynamically managed interaction and messages samples for all possible WSDL methods (including CIS) based on specific profile

<https://exchangedemo.datex2.eu/Exchange2020Client/> [Link](#)

Exchange 2020 Client

WSDL Operation

openSession

formatted output

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2020 rel. 2 sp1 (x64) (http://www.altova.com)
by Fabrizio Paoletti (AUTOSTRADE PER L'ITALIA SPA) -->
<con:exchangeInformation
xmlns:ex="http://datex2.eu/schema/3/exchangeInformation"
xmlns:con="http://datex2.eu/schema/3/messageContainer"
modelBaseVersion="3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://datex2.eu/schema/3/messageContainer
DATEXII_3_MessageContainer.xsd">
  <ex:exchangeContext>
    <ex:codedExchangeProtocol>other</ex:codedExchangeProtocol>
  </ex:exchangeContext>
</con:exchangeInformation>
<ex:exchangeSpecificationVersion>Exchange2020</ex:exchangeSpecificali
nVersion>
  <ex:supplierOrCisRequester>
    <ex:address>MySupplierAddress</ex:address>
    <ex:internationalIdentifier
xmlns:com="http://datex2.eu/schema/3/common">
      <com:country>IT</com:country>
      <com:nationalIdentifier>IT400</com:nationalIdentifier>
    </ex:internationalIdentifier>
  </ex:supplierOrCisRequester>
</ex:exchangeContext>
</ex:exchangeSpecificationVersion>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<con:exchangeInformation modelBaseVersion="3"
xsi:schemaLocation="http://datex2.eu/schema/3/messageContainer
DATEXII_3_MessageContainer.xsd"
xmlns:inf="http://datex2.eu/schema/3/informationManagement"
xmlns:com="http://datex2.eu/schema/3/common"
xmlns:loc="http://datex2.eu/schema/3/locationReferencing"
xmlns:ex="http://datex2.eu/schema/3/exchangeInformation"
xmlns:con="http://datex2.eu/schema/3/messageContainer"
xmlns:roa="http://datex2.eu/schema/3/roadTrafficData"
xmlns:cis="http://datex2.eu/schema/3/cisInformation"
xmlns:vms="http://datex2.eu/schema/3/vms"
xmlns:sit="http://datex2.eu/schema/3/situation"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ex:exchangeContext>
    <ex:codedExchangeProtocol>
      other
    </ex:codedExchangeProtocol>
    <ex:exchangeSpecificationVersion>
      Exchange2020
    </ex:exchangeSpecificationVersion>
  </ex:exchangeContext>
</con:exchangeInformation>
```

Restore default Upload File Show WSDL

Thanks for listening

DATEX II Act. 5

<https://datex2.eu/activity-5>

Fabrizio Paoletti

autostrade // Tech

autostrade // per l'Italia

fpaoletti@autostrade.it

DATEX II

6th Forum Webinar series

