FIX TRADING COMMUNITY
France Trading Conference 2025

– Protocol Interoperability –

Thursday 20th November 2025

Hanno Klein
FIX Technical Director
GTC EMEA Co-Chair
Senior Standards Advisor, FIXdom

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

Agenda

- The problem of protocol interoperability
- Understanding Orchestra
- Using Orchestra to support protocol interoperability

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

Protocol Interoperability

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

What is the problem?

- Protocol versions and customizations
  - Supporting multiple versions externally that need to be mapped to a single internal version.
  - Supporting customized versions that need to be mapped to a single “version”.
- Internal vs external protocols
  - Internal systems using FIX that needs conversion to a protocol defined by a regulator, e.g. SEC-CAT .
  - Internal systems using a proprietary protocol that needs conversion to an industry standard, e.g. FIX.
- Trade life-cycle
  - For example, FIX Protocol used for front office communication and ISO 20022 used for the back office.
- Semantic interoperability of protocols
  - Examples above are all about semantic interoperability and require skills of a business analyst.
  - Automating the semantic mapping of data across different protocols (e.g. FIX and ISO 20022) requires a common representation (a.k.a. logical model) that abstracts from a specific encoding.
  - FIX has developed the Orchestra Technical Standard to support semantic mapping between versions of a single protocol as well as across different protocols (standard or proprietary).

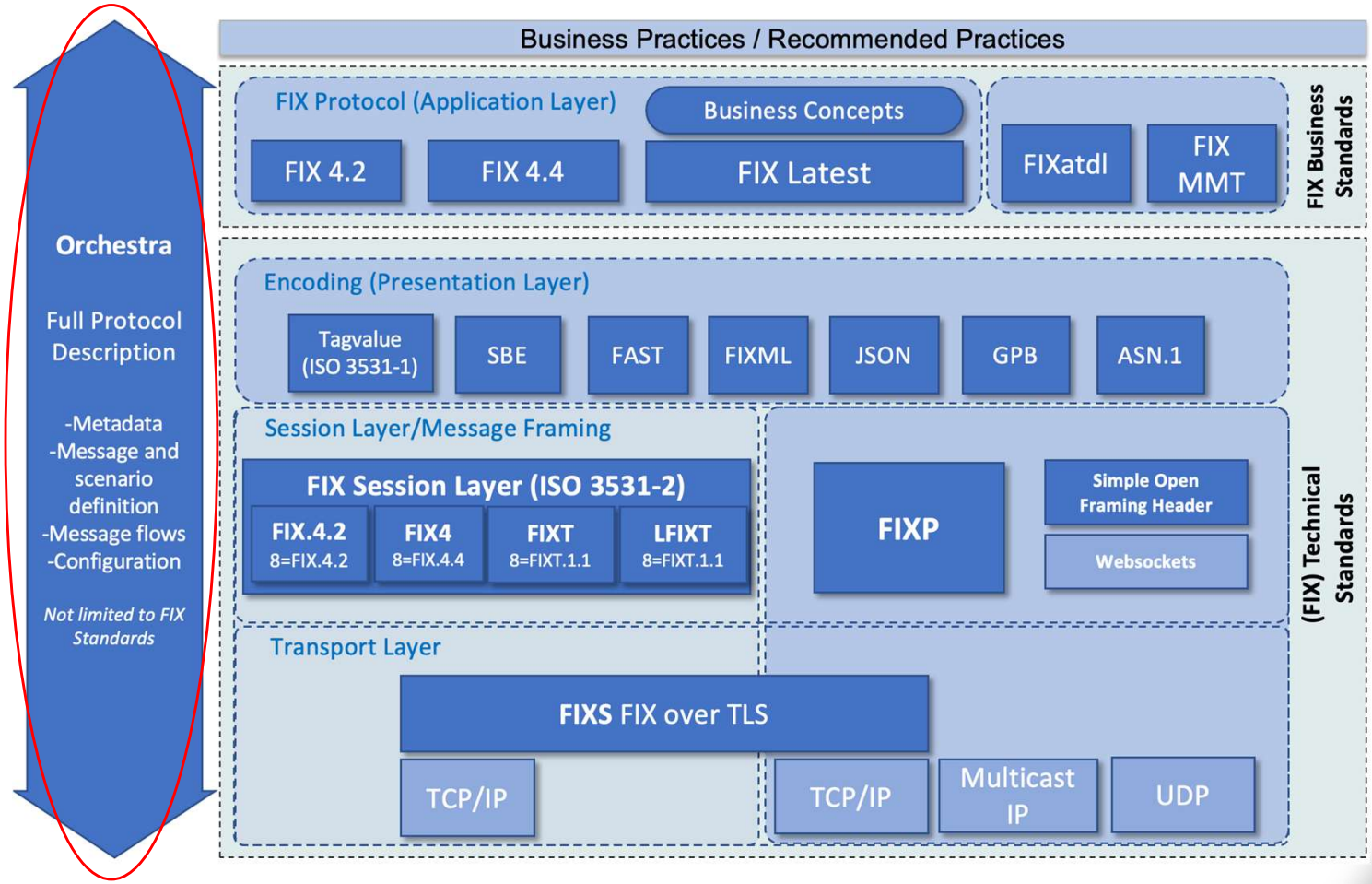
Syntax vs semantic interoperability

- Syntax interoperability
  - Requires the conversion between different encodings of data ➔ EASY.
  - For example, the FIX message NewOrderSingle may be encoded with TagValue or FIXML.
  - A conversion between two NewOrderSingle messages using a different syntax is straightforward and does not require to understand the semantics of the data elements.
  - A client order identified with “ORD001” uses the logical data element “ClOrdID”. TagValue encodes this as `11=ORD001` whereas FIXML encodes the same information as `ID="ORD001"`.
- Semantic interoperability
  - Requires the conversion between different logical models of data ➔ COMPLEX.
  - For example, FIX 4.2 identifies the executing firm of an order with the data element ExecBroker. FIX 4.3 changed this and introduced the repeating group Parties, each instance having 3 data elements (PartyID, PartyIDSource, PartyRole).
  - The data element ExecBroker is semantically equivalent to the data elements PartyID and PartyRole, regardless of the encoding (PartyIDSource is not available as information in FIX 4.2).
- Why is it important to separate the two?
  - They are very different problems to solve but are often bundled together, i.e. one logical model and syntax is converted to another model and syntax, e.g. FIX TagValue to ISO 20022 XML.
  - A single logical model (e.g. FIX or ISO 20022) can have multiple alternate syntaxes, requiring a multitude of converters unless one abstracts from the syntax and focusses on the conversion of logical models.

Understanding Orchestra

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

FIX Standards



FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

Orchestra Technical Standard

What it is...	What it is NOT...
▪ Machine-readable standard for metadata describing the content and behavior of an electronic messaging interface. ▪ Protocol agnostic, i.e. applicable to the FIX Protocol, other industry standard protocols (e.g. ISO 20022), regulatory protocols (e.g. SEC-CAT), and proprietary protocols (e.g. defined by exchanges and clearing houses). ▪ Syntax agnostic, separating business semantics from the wire format (e.g. TagValue, FIXML, SBE, JSON, ISO 20022 XML). ▪ Language to define messages, groups, components, fields, code sets, codes, generic datatypes together with their descriptions. ▪ Language to define scenarios, presence rules, workflows, actors, state changes. ▪ Language to support audit trail of changes (a.k.a. pedigree) down to a single code of a code set.	▪ Replacement for or new version of the FIX Protocol. ▪ Only applicable to the FIX Protocol. ▪ New encoding standard for FIX messages in addition to TagValue, FIXML, SBE, FAST. ▪ Requires to make changes to FIX messages. ▪ Software application, tool or product. ▪ Requires a license to use it. ▪ Limited to metadata related to the application layer.

Metadata -> Logical –> Physical (Syntax)

Metadata						
Message	Group	Component	Field	Code Set	Code	Datatype

Message for the entry of a new limit order from firm X for 100 shares.

Logical Data Elements (FIX Protocol)					
MsgType (NewOrder Single)	**Parties** (Executing Firm X)	**OrderQtyData OrderQty** (100)	**ClOrdID** (ORD001)	**OrdType** (Limit)	...

	Physical Data Elements (Wire Format)					
TagValue	35=D	453=1 448=X 447=N 452=1	38=100	11=ORD001	40=2	...
FIXML	<Order...>	<Pty ID="X" R="1" Src="N">	Qty="100"	ID="ORD001"	Typ="2"	...
SBE	0e00	030158014e	6400	4f5244303031	32	...

Orchestra Standard (XSD Schema file)

```
<xs:complexType name="messageType">
 <xs:sequence>
  <xs:element name="structure" minOccurs="0">
   <xs:complexType>
    <xs:choice maxOccurs="unbounded">
     <xs:element name="groupRef" type="fixr:groupRefType"/>
     <xs:element name="componentRef" type="fixr:componentRefType"/>
     <xs:element name="fieldRef" type="fixr:fieldRefType"/>
    </xs:choice>
   <xs:complexType>
  <xs:element>
 <xs:sequence>
<xs:complexType>
```

Orchestra Representation (XML file)

```
<fixr:message msgType="D" id="14" name="NewOrderSingle">
 <fixr:structure>
  <fixr:fieldRef presence="required" id="11"/>
  <fixr:groupRef id="1012"/>
  <fixr:componentRef presence="required" id="1003"/>
  <fixr:componentRef presence="required" id="1011" which="oneOf"/>
  <fixr:fieldRef presence="required" id="40"/>
 <fixr:structure>
</fixr:message>
```

FIXML Schema | **SBE Schema** | **JSON Schema** | **ISO 20022 XML Schema**

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

Orchestra XML sample

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<fixr:repository xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:fixr="http://fixprotocol.io/2020/orchestra/repository"
    name="Orchestra RoE Example Version 1.0" version="1.0">
    <fixr:metadata>
        <dcterms:title>My Rules of Engagement</dcterms:title>
        <dcterms:publisher>FIX Trading Community</dcterms:publisher>
        <dcterms:rights>Copyright © FIX Protocol Ltd.</dcterms:rights>
        <dcterms:date>2024-10-16</dcterms:date>
    </fixr:metadata>
    <fixr:datatypes>…
    <fixr:codeSets>…
    <fixr:fields>…
    <fixr:components>…
    <fixr:groups/>
    <fixr:messages>…
</fixr:repository>
```

The complete Orchestra XML file for FIX Latest has 164,528 lines as of EP299!

- Orchestra XML uses English keywords to identify logical elements.
- Logical elements have attributes and some have nested logical elements.
- Rules may be attached to elements to express conditional requirements across logical elements.
- All elements may have annotations that can be used to automatically generate documentation.
- Datatypes are logical and can be mapped to physical datatypes of multiple encodings.
- Meta-data uses DC Terms (ISO 15836) as standard.

```
<fixr:field type="String" baseCategory="SingleGeneralOrderHandling" baseCategoryAbbrName="ID"
    id="11" name="ClOrdID" abbrName="ClOrdID" added="FIX.2.7" updated="FIX.Latest" updatedEP="282">
    <fixr:annotation>
        <fixr:documentation purpose="SYNOPSIS">
            Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.)
            (identified by SenderCompID(49) or OnBehalfOfCompID(115) as appropriate). Uniqueness must be
            guaranteed within a single trading day. Firms, particularly those which electronically submit
            multi-day orders, trade globally or throughout market close periods, should ensure uniqueness
            across days, for example by embedding a date within the ClOrdID(11) field.</fixr:documentation>
    </fixr:annotation>
</fixr:field>
```

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

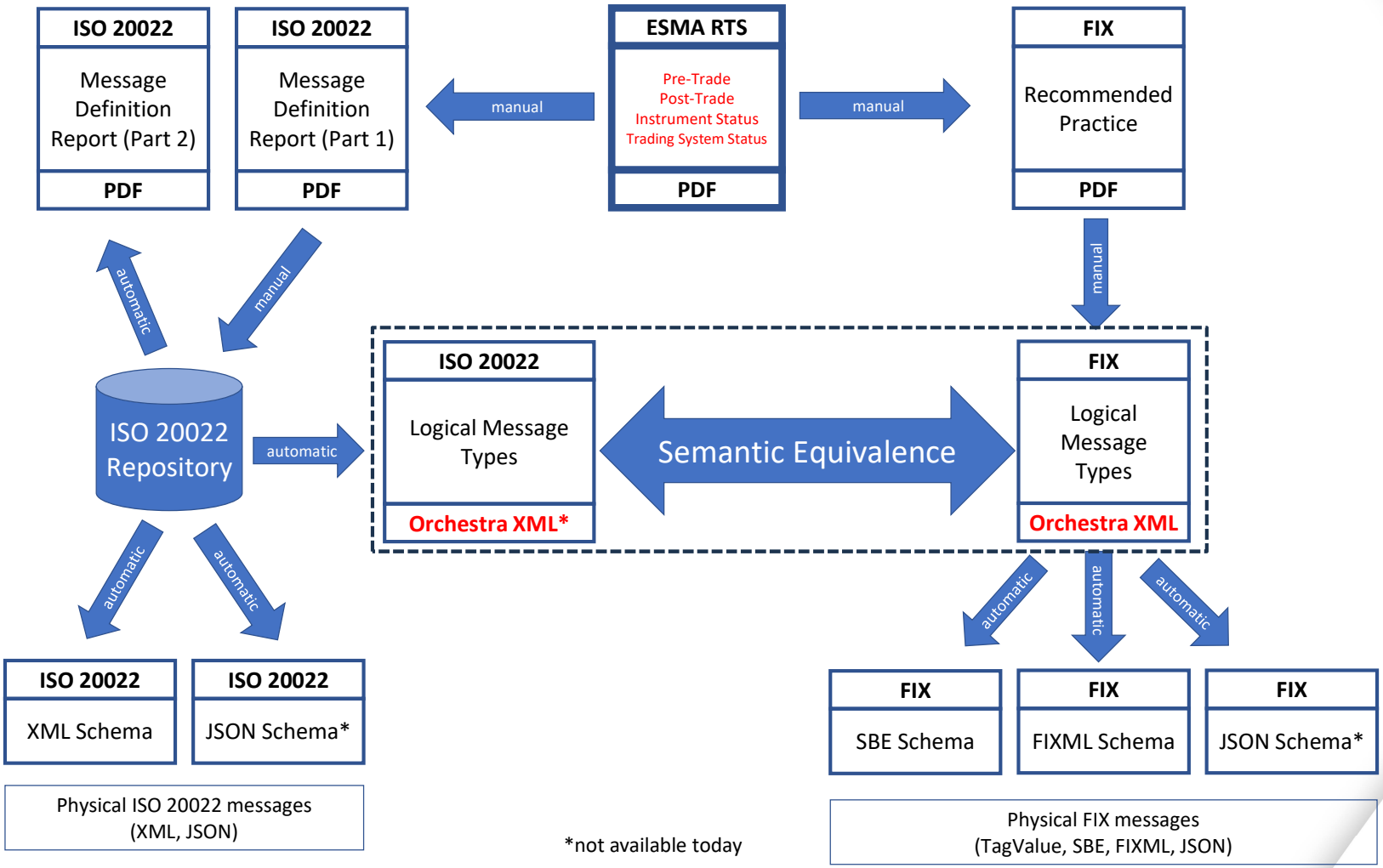
Using Orchestra for Interoperability

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

Use case: EU Consolidated Tape

- Background
  - The regulator (ESMA) is picking a single provider per asset class (bonds, equities, derivatives).
  - ESMA defines the technical requirements (e.g. latency) as well as the data elements that have to be contributed, consolidated and distributed by means of a Regulatory Technical Standard (RTS).
- Mapping the RTS to ISO 20022
  - ESMA is mapping the RTS to (new) messages of the ISO 20022 logical model.
  - ISO 20022 has not been used for market data in the financial industry and primarily supports an XML syntax that is insufficient for the given technical requirements.
  - FIX Global Technical Committee is using Orchestra to create a machine-readable representation of the logical ISO 20022 messages.
- Mapping the RTS to FIX
  - FIX Consolidated Tape WG has mapped the RTS data elements to logical FIX messages that can use any syntax supported by FIX (e.g. TagValue or Simple Binary Encoding).
  - The logical FIX messages can be provided in a machine-readable form in Orchestra together with the references to the RTS data elements.
- Using Orchestra for FIX and ISO 20022
  - Orchestra XML files created from the same RTS are semantically equivalent and can be used to create or validate logical messages using the respective protocol.
  - Syntax schemas can be generated from the Orchestra XML files by adding datatype mappings for the relevant encodings and can be used to create or validate the physical messages (wire format).

Use Case: EU Consolidated Tape



FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

Semantic Mapping of RTS to FIX Protocol

Table 15: FIX mapping for RTS Annex III Table 3

#	RTS Field Identifier	FIX Field(s)
1	Entry date and time	TrdRegTimestamp(769), TrdRegTimestampType(770)=36 (Update time), TrdRegTimestampOrigin(771)=C (Contributor)
2	Instrument identification code	SecurityID(48), SecurityIDSource(22)=4 (ISIN)
3	Currency	Currency(15)
4	Best bid	MDEntryType(269)=0 (Bid), MDEntryPx(270)
5	Best bid volume	MDEntryType(269)=0 (Bid), MDEntrySize(271)
6	EBBO timestamp	TrdRegTimestamp(769), TrdRegTimestampType(770)=34 (Reference time for BBO), TrdRegTimestampOrigin(771)=P (Publisher)
7	MRMTL	MostLiquidMarketID(3102)
8	Best offer	MDEntryType(269)=1 (Offer), MDEntryPx(270)
9	Best offer volume	MDEntryType(269)=1 (Offer), MDEntrySize(271)
10	Dissemination date and time	TrdRegTimestamp(769), TrdRegTimestampType(770)=11 (Publicly reported), TrdRegTimestampOrigin(771)=P (Publisher)
11	Publication date and time	TrdRegTimestamp(769), TrdRegTimestampType(770)=11 (Publicly reported), TrdRegTimestampOrigin(771)=C (Contributor)

CTP output of pre-trade data for continuous order book trading

Outlook

- Open-source community for tools
  - Orchestra is a FIX Technical Standard, i.e. a specification document.
  - Tools are highly recommended to create and maintain Orchestra XML files.
  - FIX Trading Community has already provided a number of useful tools for Orchestra (https://www.fixtrading.org/standards/orchestra-tools/)
  - FIX members can also provide free or open-source tools supporting the use of Orchestra.
  - FIX Global Technical Committee is looking for FIX members interested in developing an open-source community on GitHub (https://github.com/FIXTradingCommunity).
  - Please contact fix@fixtrading.org if you are interested in supporting the open-source initiative!

- Mapping schema for Orchestra
  - Orchestra standard currently consists of two schemas.
  - Repository schema covers metadata for the application and presentation (a.k.a. encoding) layer.
  - Interfaces schema covers metadata for the session and transport layer.
  - The definition of a third schema for semantic mapping between logical models represented with Orchestra would allow automatic message conversion between different protocols.
  - Development of the Orchestra standard is driven by FIX members and the Orchestra Subcommittee as part of the FIX Global Technical Committee.
  - Please contact fix@fixtrading.org if you are interested in joining the Orchestra initiative!

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL