# FIX TRADING COMMUNITY
## EMEA Trading Conference 2026

– Protocol Interoperability –

Thursday 5th March 2026

**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
- Orchestra and ISO 20022
- Orchestra and Protocol Interoperability

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

# The Problem

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

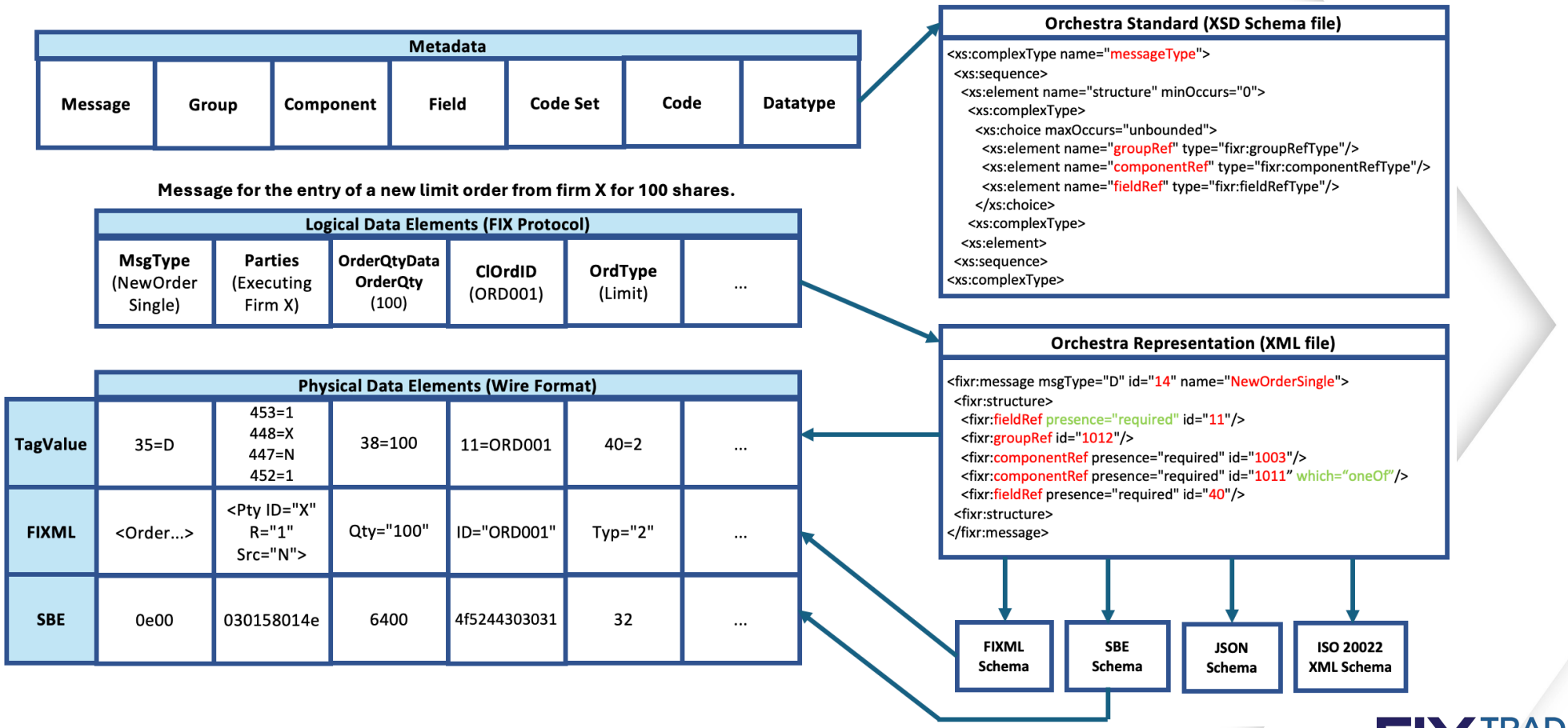
# What is the problem?

- 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 internal 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.
- Different protocols across the 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 logical model that abstracts from a specific encoding.
  - FIX Protocol has a different logical model than ISO 20022 but both can be represented with Orchestra.
  - FIX has developed the Orchestra 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 Protocol 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.

# Metadata -> Logical –> Physical (Syntax)




FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

# Orchestra and ISO 20022

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

# Orchestra Standard

| What it is... | What it is NOT... |
| --- | --- |
| ▪ Machine-readable standard for metadata describing the content and behavior of an electronic messaging interface.<br>▪ 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).<br>▪ Syntax agnostic, separating business semantics from the wire format (e.g. TagValue, FIXML, SBE, JSON, ISO 20022 XML).<br>▪ Language to define messages, groups, components, fields, code sets, codes, generic datatypes together with their descriptions.<br>▪ Language to define scenarios, presence rules, workflows, actors, state changes.<br>▪ 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.<br>▪ Only applicable to the FIX Protocol.<br>▪ New encoding standard for FIX messages in addition to TagValue, FIXML, SBE, FAST.<br>▪ Requires to make changes to FIX messages.<br>▪ Software application, tool or product.<br>▪ Requires a license to use it.<br>▪ Limited to metadata related to the application layer. |

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

# Orchestra and ISO 20022

- Orchestra is a publicly available standard for meta-data
  - Initially developed for the FIX Protocol and its core encodings TagValue and FIXML
  - It is protocol agnostic and can also be used for other logical models
- ISO 20022 has a business model in addition to a logical model
  - The ISO 20022 business model defines components and elements, not messages
  - The ISO 20022 logical model uses subsets of the business components and defines logical messages with them
  - ISO 20022 supports the transformation of logical messages to encodings, currently XML and ASN.1 (there is also a whitepaper for JSON)
- Machine-readable version of ISO 20022 (a.k.a. erepo) is publicy available
  - ISO 20022 business and logical model are both part of the erepo
  - Very large XML file (https://www.iso20022.org/iso20022-repository/e-repository) in EMF format based on the ISO 20022 ecore implementation metamodel
  - FIX has successfully completed a PoC to convert ISO 20022 logical messages to Orchestra
  - Orchestra is well suited to bridge the gap between logical FIX and ISO 20022 messages, e.g. the EU Consolidated Tape is defined in both logical models

# Example: BenchmarkCurveName

## FIX Protocol (Orchestra)

```
<fixr:codeSet type="String" id="221" name="BenchmarkCurveNameCodeSet">
  <fixr:code value="EONIA" id="221001" name="EONIA">
  <fixr:code value="EUREPO" id="221002" name="EUREPO">
  <fixr:code value="Euribor" id="221003" name="Euribor">
  <fixr:code value="FutureSWAP" id="221004" name="FutureSWAP">
  <fixr:code value="LIBID" id="221005" name="LIBID">
  <fixr:code value="LIBOR" id="221006" name="LIBOR">
  <fixr:code value="MuniAAA" id="221007" name="MuniAAA">
  <fixr:code value="OTHER" id="221008" name="OTHER">
  <fixr:code value="Pfandbriefe" id="221009" name="Pfandbriefe">
  <fixr:code value="SONIA" id="221010" name="SONIA">
  <fixr:code value="SWAP" id="221011" name="SWAP">
  <fixr:code value="Treasury" id="221012" name="Treasury">
  <fixr:code value="FEDEFF" id="221013" name="FedFundRateEffective">
  <fixr:code value="FEDOPEN" id="221014" name="FedOpen">
  <fixr:code value="EURIBOR" id="221015" name="EURIBOR">
  <fixr:code value="AUBSW" id="221016" name="AUBSW">
  <fixr:code value="BUBOR" id="221017" name="BUBOR">
  <fixr:code value="CDOR" id="221018" name="CDOR">
  <fixr:code value="CIBOR" id="221019" name="CIBOR">
  <fixr:code value="EONIASWAP" id="221020" name="EONIASWAP">
  <fixr:code value="ESTR" id="221021" name="ESTR">
  <fixr:code value="EURODOLLAR" id="221022" name="EURODOLLAR">
  <fixr:code value="EUROSWISS" id="221023" name="EUROSWISS">
  <fixr:code value="GCFREPO" id="221024" name="GCFREPO">
  <fixr:code value="ISDAFIX" id="221025" name="ISDAFIX">
  <fixr:code value="JIBAR" id="221026" name="JIBAR">
    ...
<fxr:codeSet>
```

## ISO 20022 (Orchestra)

```
<fixr:codeSet name="BenchmarkCurveName2Code" id="978" type="String">
  <fixr:code value="WIBO" name="WIBOR" id="979"/>
  <fixr:code value="TREA" name="Treasury" id="980"/>
  <fixr:code value="TIBO" name="TIBOR" id="981"/>
  <fixr:code value="TLBO" name="TELBOR" id="982"/>
  <fixr:code value="SWAP" name="SWAP" id="983"/>
  <fixr:code value="STBO" name="STIBOR" id="984"/>
  <fixr:code value="PRBO" name="PRIBOR" id="985"/>
  <fixr:code value="PFAN" name="Pfandbriefe" id="986"/>
  <fixr:code value="NIBO" name="NIBOR" id="987"/>
  <fixr:code value="MAAA" name="MuniAAA" id="988"/>
  <fixr:code value="MOSP" name="MOSPRIM" id="989"/>
  <fixr:code value="LIBO" name="LIBOR" id="990"/>
  <fixr:code value="LIBI" name="LIBID" id="991"/>
  <fixr:code value="JIBA" name="JIBAR" id="992"/>
  <fixr:code value="ISDA" name="ISDAFIX" id="993"/>
  <fixr:code value="GCFR" name="GCFRepo" id="994"/>
  <fixr:code value="FUSW" name="FutureSWAP" id="995"/>
  <fixr:code value="EUCH" name="EuroSwiss" id="996"/>
  <fixr:code value="EUUS" name="EURODOLLAR" id="997"/>
  <fixr:code value="EURI" name="Euribor" id="998"/>
  <fixr:code value="EONS" name="EONIASwaps" id="999"/>
  <fixr:code value="EONA" name="EONIA" id="1000"/>
  <fixr:code value="CIBO" name="CIBOR" id="1001"/>
  <fixr:code value="CDOR" name="CDOR" id="1002"/>
  <fixr:code value="BUBO" name="BUBOR" id="1003"/>
  <fixr:code value="BBSW" name="BBSW" id="1004"/>
</fixr:codeSet>
```

# Example: Financial Transaction Reporting

| Name | Type | Multiplicity |
|---|---|---|
| TransactionIdentification | Max52Text | [1..1] |
| ExecutingParty | LEIIdentifier | [1..1] |
| InvestmentPartyIndicator | TrueFalseIndicator | [1..1] |
| SubmittingParty | LEIIdentifier | [1..1] |
| Buyer | PartyIdentification79 | [1..1] |
| Seller | PartyIdentification79 | [1..1] |
| OrderTransmission | SecuritiesTransactionTransmission2 | [1..1] |
| Transaction | SecuritiesTransaction3 | [1..1] |
| FinancialInstrument | FinancialInstrumentAttributes5Choice | [1..1] |
| InvestmentDecisionPerson | InvestmentParty1Choice | [0..1] |
| ExecutingPerson | ExecutingParty1Choice | [1..1] |

ISO 20022 logical component

*SecuritiesTransactionReport7*

(used in auth.016)

- Elements either have a simple type, e.g. Max52Text, or are a component (complex type), e.g. PartyIdentification79.
- Elements with a code list are simple types.
- Names of elements only need to be unique inside a component.
- All elements are defined with a "local" cardinality, e.g. [0..1] means optional and at most once.
- Only one of the elements inside a choice component can be used and the component name ends with "Choice".
- Multiple elements may represent the same component, e.g. buyer/seller.
- Variations of components are explicit, using a number to support unique names, e.g. SecuritiesTransaction3.

Source:
https://www.iso20022.org/standardsrepository/type/SecuritiesTransactionReport7

# Example: Financial Transaction Reporting

```
<fixr:message id="1432" name="FinancialInstrumentReportingTransactionReportV03_auth_016_001_03">
  <fixr:structure>
    <fixr:groupRef id="1437" instanceName="Transaction" presence="required"/>
    <fixr:groupRef id="2827" instanceName="SupplementaryData"/>
  </fixr:structure>
</fixr:message>
```

```
<fixr:group name="ReportingTransactionType3Choice" which="oneOf" id="1437">
  <fixr:groupRef id="1441" instanceName="New" presence="required" implMaxOccurs="1"/>
  <fixr:groupRef id="2832" instanceName="Cancellation" presence="required" implMaxOccurs="1"/>
  <fixr:groupRef id="2827" instanceName="SupplementaryData"/>
</fixr:group>
```

```
<fixr:group name="SecuritiesTransactionReport7" id="1441">
  <fixr:fieldRef id="8" instanceName="TransactionIdentification" presence="required"/>
  <fixr:fieldRef id="9" instanceName="ExecutingParty" presence="required"/>
  <fixr:fieldRef id="10" instanceName="InvestmentPartyIndicator" presence="required"/>
  <fixr:fieldRef id="14" instanceName="SubmittingParty" presence="required"/>
  <fixr:groupRef id="1465" instanceName="Buyer" presence="required" implMaxOccurs="1"/>
  <fixr:groupRef id="1465" instanceName="Seller" presence="required" implMaxOccurs="1"/>
  <fixr:groupRef id="1679" instanceName="OrderTransmission" presence="required" implMaxOccurs="1"/>
  <fixr:groupRef id="1698" instanceName="Transaction" presence="required" implMaxOccurs="1"/>
  <fixr:groupRef id="1964" instanceName="FinancialInstrument" presence="required" implMaxOccurs="1"/>
  <fixr:groupRef id="2606" instanceName="InvestmentDecisionPerson" implMaxOccurs="1"/>
  <fixr:groupRef id="2613" instanceName="ExecutingPerson" presence="required" implMaxOccurs="1"/>
  <fixr:groupRef id="2628" instanceName="AdditionalAttributes" presence="required" implMaxOccurs="1"/>
  <fixr:groupRef id="2727" instanceName="TechnicalAttributes" implMaxOccurs="1"/>
  <fixr:groupRef id="2827" instanceName="SupplementaryData"/>
</fixr:group>
```

- Orchestra attribute “instanceName” is used for the ISO 20022 element names.
- ISO 20022 components are defined as groups with a cardinality defined wherever they are used.
- Orchestra attribute “which” is used for ISO 20022 choice components.
- Orchestra supports having the same element multiple times in a single message, group or component.
- Orchestra attribute “presence” used for mandatory elements inside a message or group.

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

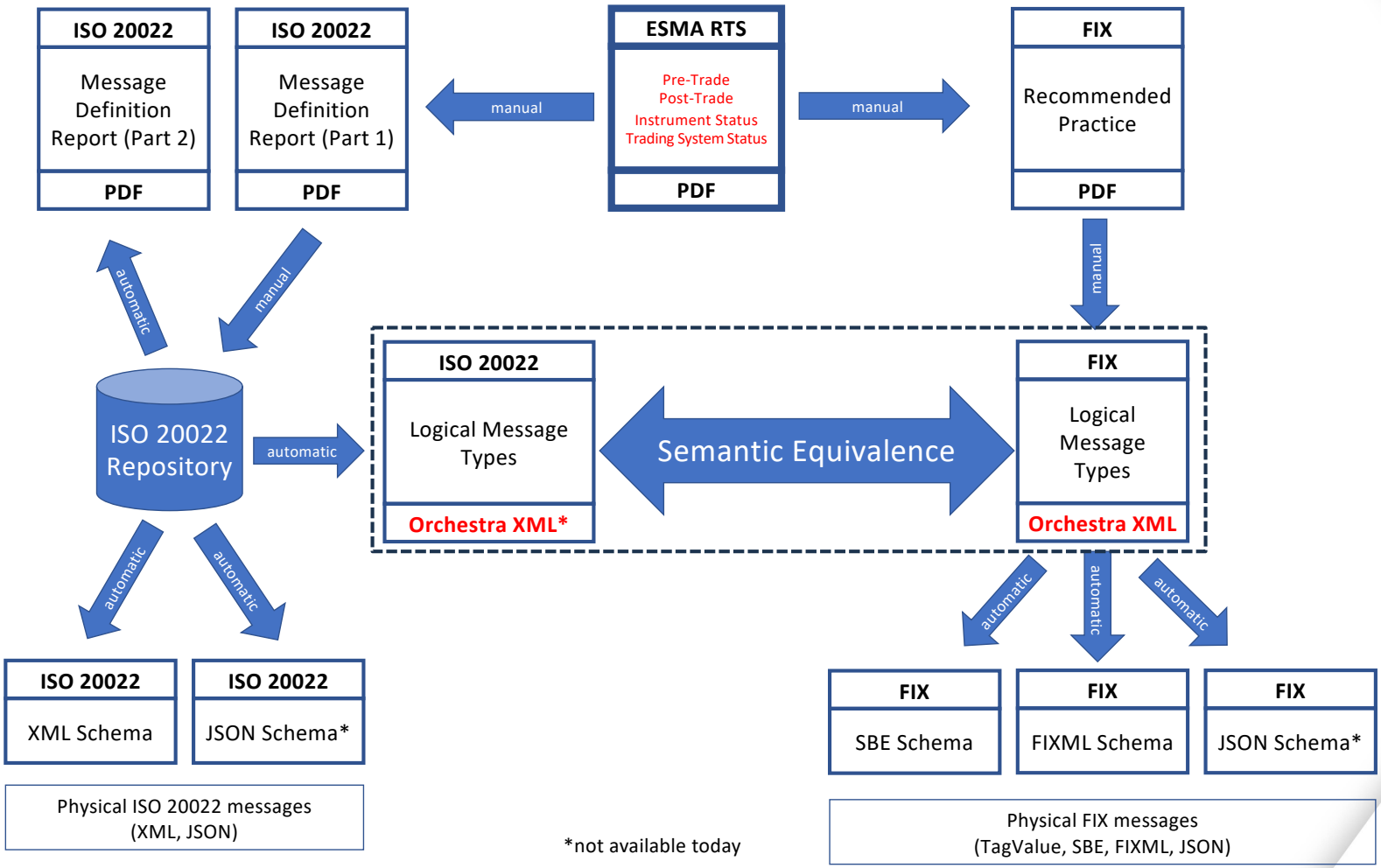
# Orchestra and Protocol Interoperability

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 logical 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

# Message Mapping between FIX and ISO 20022

| # | FIX Message | ISO 20022 Message |
|---|---|---|
| 1 | SecurityStatus(35=f) | InstrumentStatusReport (auth.120) |
| 2 | TradingSessionStatus(35=h) | TradingSystemStatusReport (auth.121) |
| 3 | MarketDataSnapshotFullRefresh(35=W) | PreTradeInputMarketDataReport (auth.122) |
| 4 | MarketDataSnapshotFullRefresh(35=W) | PreTradeOutputMarketData Report (auth.123) |
| 5 | MarketDataIncrementalRefresh(35=X) | PostTradeMarketDataReport (auth.124) |

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

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

# Element Mapping of RTS to ISO 20022

| # | RTS Field Identifier | ISO 20022 Element |
|---|---|---|
| 1 | Entry date and time | EntryDateTime |
| 2 | Instrument identification code | ISIN |
| 3 | Currency | Currency |
| 4 | Best bid | BestBid |
| 5 | Best bid volume | BestBidVolume |
| 6 | EBBO timestamp | CalculationDateTime |
| 7 | MRMTL | MostRelevantMarket |
| 8 | Best offer | BestOffer |
| 9 | Best offer volume | BestOfferVolume |
| 10 | Dissemination date and time | DisseminationDateTime |
| 11 | Publication date and time | PublicationDateTime |

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

# Detailed Mapping between FIX and ISO 20022

| # | FIX Field(s) | ISO 20022 Element |
|---|---|---|
| 1 | MDFullGrp/TrdRegTimestamps/TrdRegTimestamp(769), TrdRegTimestampType(770)=36 (Update time), TrdRegTimestampOrigin(771)=C (Contributor) | PreTradeData/OutputData/ContinuousOrderBook/EntryDateTime |
| 2 | Instrument/SecurityID(48), SecurityIDSource(22)=4 (ISIN) | PreTradeData/ISIN |
| 3 | MDFullGrp/Currency(15) | PreTradeData/OutputData/ContinuousOrderBook /Currency |
| 4 | MDFullGrp/MDEntryType(269)=0 (Bid), MDEntryPx(270) | PreTradeData/OutputData/ContinuousOrderBook/Bid/BestBid |
| 5 | MDFullGrp/MDEntryType(269)=0 (Bid), MDEntrySize(271) | PreTradeData/OutputData/ContinuousOrderBook/Bid/BestBidVolume |
| 6 | MDFullGrp/TrdRegTimestamps/TrdRegTimestamp(769), TrdRegTimestampType(770)=34 (Reference time for BBO), TrdRegTimestampOrigin(771)=P (Publisher) | PreTradeData/OutputData/ContinuousOrderBook/CalculationDateTime |
| 7 | MostLiquidMarketID(3102) | PreTradeData/MostRelevantMarket |
| 8 | MDFullGrp/MDEntryType(269)=1 (Offer), MDEntryPx(270) | PreTradeData/OutputData/ContinuousOrderBook/Offer/BestOffer |
| 9 | MDFullGrp/MDEntryType(269)= 1 (Offer), MDEntrySize(271) | PreTradeData/OutputData/ContinuousOrderBook/Offer/BestOfferVolume |
| 10 | MDFullGrp/TrdRegTimestamps/TrdRegTimestamp(769), TrdRegTimestampType(770)=11 (Publicly reported), TrdRegTimestampOrigin(771)=P (Publisher) | PreTradeData/DisseminationDateTime |
| 11 | MDFullGrp/TrdRegTimestamps/TrdRegTimestamp(769), TrdRegTimestampType(770)=11 (Publicly reported), TrdRegTimestampOrigin(771)=C (Contributor) | PreTradeData/PublicationDateTime |

# Mapping of logical elements

- Messages
  - One-to-one, i.e. a single message maps to another single message, e.g. SecurityStatus(35=f) maps to InstrumentStatusReport (auth.120)
  - One-to-many, i.e. a single message maps to two or more messages (or vice versa), e.g. MarketDataSnapshotFullRefresh(35=W) maps to PreTradeInputMarketDataReport (auth.122) and PreTradeOutputMarketData Report (auth.123)
  - Orchestra scenarios can be used to create multiple versions for a 1:n situation
- Groups and components
  - Same as messages for cases where there are semantically equivalent groups and components, i.e. Orchestra scenarios can be used
  - A component may map to individual elements that have not been combined into a component
  - Instances of groups may map to individual fields, e.g separate instances of the TrdRegTimestamps group map to EntryDateTime, CalculationDateTime, DisseminationDateTime, PublicationDateTime
- Fields
  - One-to-one i.e. a single field maps to another single field, e.g. Currency(15) maps to Currency
  - One-to-many, i.e. a single field maps to two or more fields (or vice versa), e.g. SecurityID(48) and SecurityIDSource(22) map to ISIN
  - Fields may also map to one-to-one to codes and vice versa, e.g. TradeType(3006)=64 (Benchmark) maps to BENC in ISO 20022 code list ExternalPostTradeEquityCode
- Codes
  - One-to-one mapping unless code maps to a field (see previous bullet)