FIX TRADING COMMUNITY
EMEA Trading Conference 2026

– Update from the Orchestra Subcommittee –

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

- Understanding Orchestra
- Orchestra Version 1.1 (progress report)
- Orchestra and ISO 20022

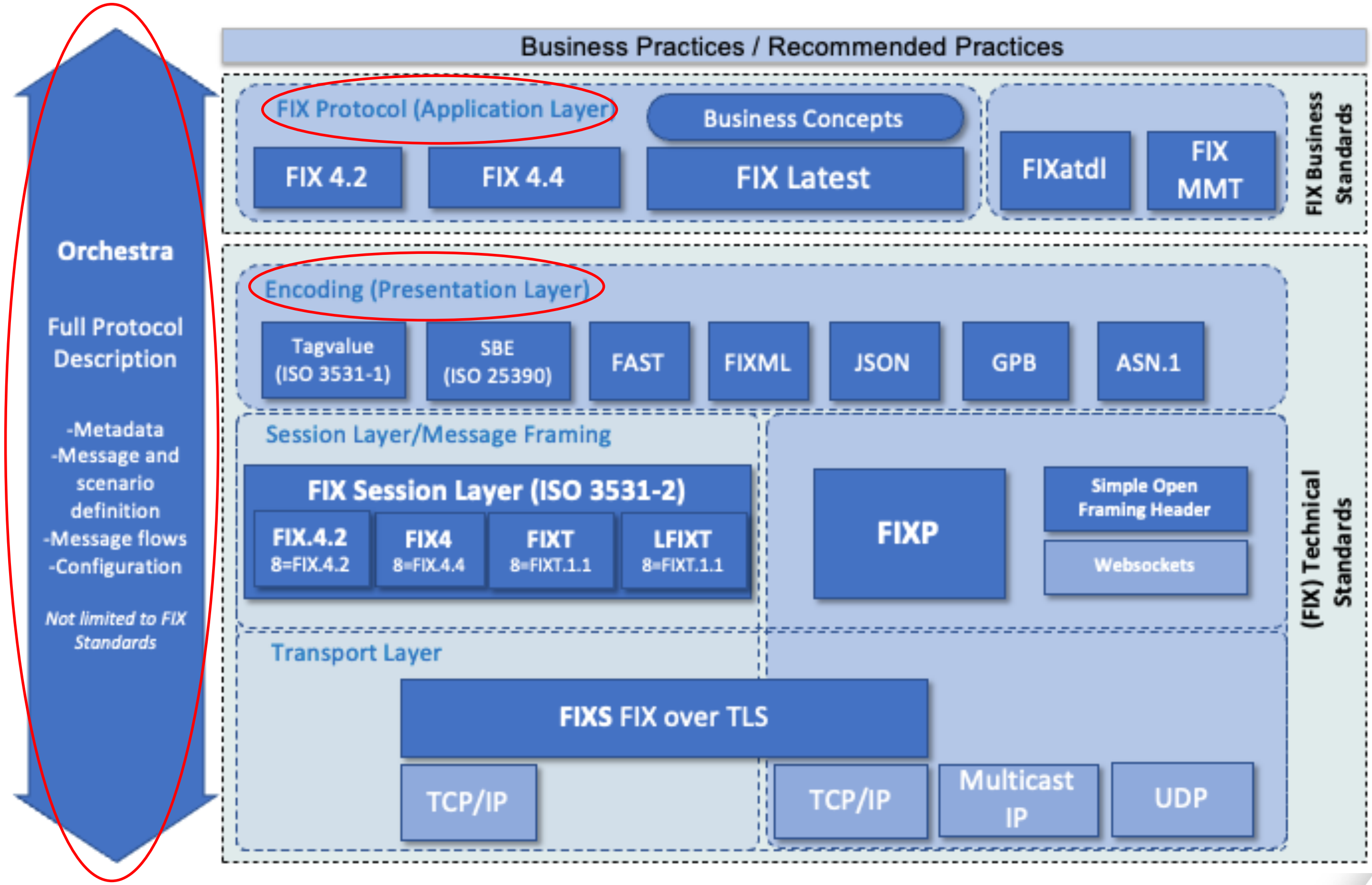
Understanding Orchestra

Orchestra Standard

- The Orchestra Standard is developed by the FIX Orchestra Subcommittee
- Version 1.0 was released in February 2021 and supports a complete logical model of an electronic interface
- Version 1.1 is currently work in progress, having published two release candidates (RC1 and RC2) so far
- A version typically has multiple release candidates to address different areas for enhancements and they are made available to the general public
- Release candidates are not necessarily backward compatible as pilot implementations may identify issues with proposed enhancements
- After the Beta Release of a version, only non-breaking changes, clarifications and corrections can be made
- Orchestra Version 1.1 RC3 should become available for public comment during the second quarter of 2026

If you are a FIX member and interested in Orchestra, please join the group by sending an email to the FIX Program Office (fix@fixtrading.org)

FIX Standards



FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

Orchestra Technical Standard

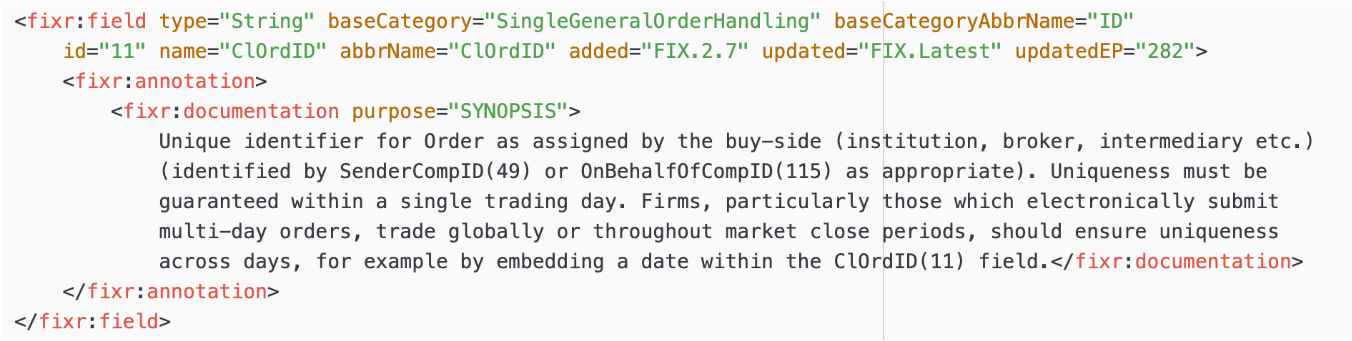
What it is...

- 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.

What it is NOT...

- 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.

Example: Defining ClOrdID(11) in Orchestra



```
<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 Protocol uses the Orchestra Standard to make it machine-readable
- Syntax is XML, but the names of elements (e.g. “field”) and attributes (e.g. “type”) represent the “language” of the Orchestra Standard
- Annotation elements support automated generation of Rules of Engagement
- Pedigree attribute, e.g. “added” support an audit trail of changes
- Encodings can use attribute values for the wire format
    - Attribute “id” is used for the tag number of the TagValue encoding
    - Attribute “abbrName” is used for the XML attribute of the FIXML encoding

Orchestra Version 1.1 (progress report)

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

Orchestra v1.1 – Release Candidates

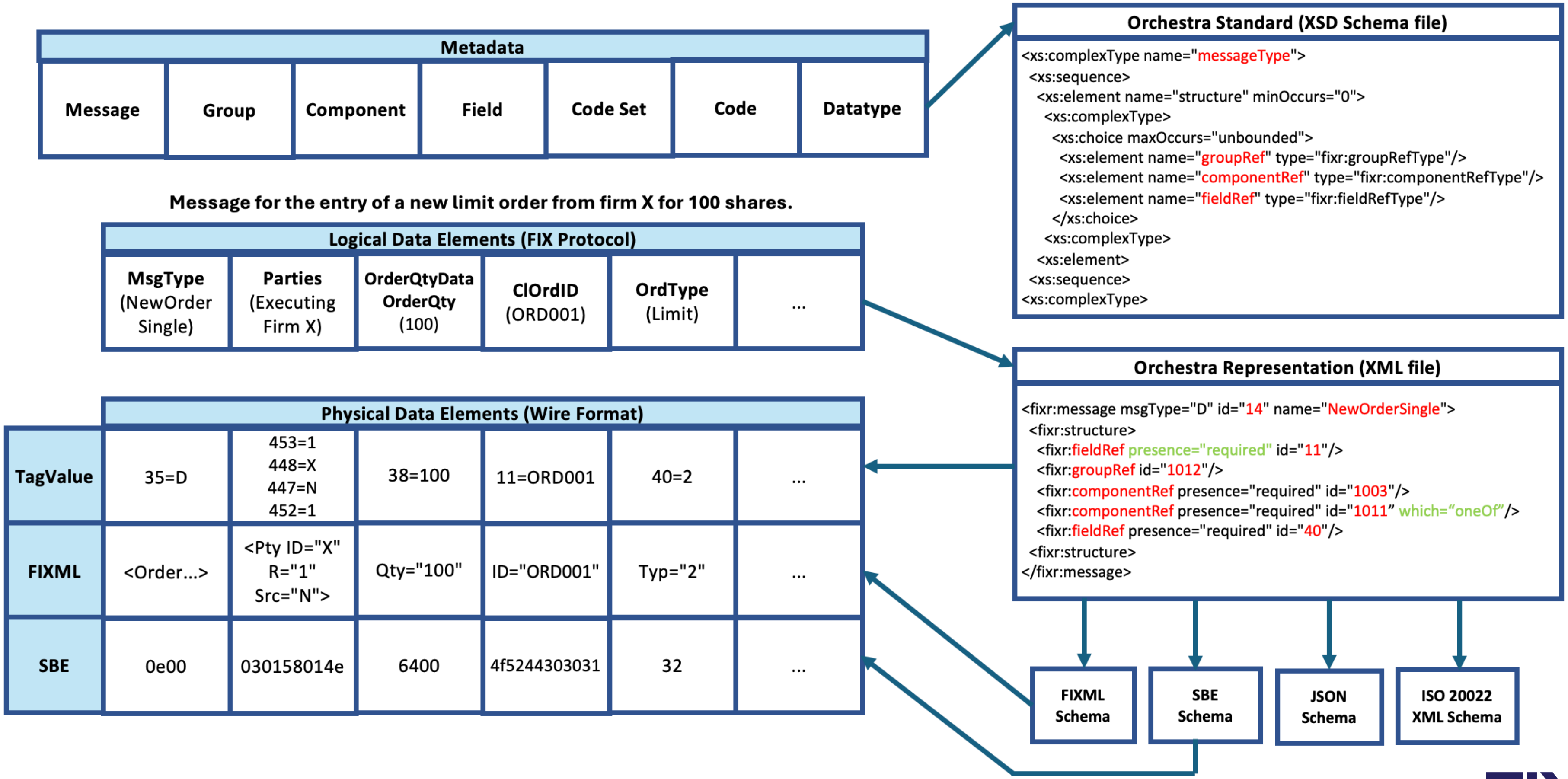
- The first two release candidates for version 1.1 have been published in November 2023 (RC1) and November 2024 (RC2)
- RC1 made changes and extensions, for example:
  - Scenarios not only as references but with standalone definitions
  - Distinction between fields having a simple datatype and a code set
  - Support for annotations of top-level elements
  - Support for xi:include
- RC2 made changes and extensions, for example:
  - Add concept of scenario relationships
  - Extend and define documentation purposes beyond synopsis and elaboration
  - Recommendations for the usage of DC terms (repository metadata)
- RC3 focusses on the introduction of a generic framework supporting arbitrary encoding information

Encoding Information in Orchestra

- What is the problem?
  - FIX TagValue and FIXML encodings use logical attributes (id, MsgType, abbrName) from Orchestra as encoding information for the wire format
  - RC2 only has explicit encoding information as part of the mappedDatatype element
  - Different encodings may require very different attributes to describe the rules for the transformation from the logical to the physical layer
  - Adding encoding-specific attributes to the standard would require a new Orchestra version for every new encoding to be supported
  - Some encodings (e.g. FIXML, SBE, JSON) have their own schemas

- RC3 Concept for Multiple Encodings
  - Orchestra only defines a framework of how to add encoding-specific information
  - The framework supports the generation of encoding-specific schemas
  - Encoding-specific information is supported at the top level and at the level of each element type in Orchestra that is relevant for the wire format or for schema generation
  - Encoding-specific information is defined as an external schema by the provider of the encoding and its attributes can be used inside an Orchestra file

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

External Schemas for Encoding Information

- Standard XML schema (XSD file) with a namespace definition, e.g. xmlns:orchSbe="http://fixprotocol.io/2025/orchestra/encoding/sbe"
- Definition of encoding attributes as XML elements, e.g. byteOrder, and supported values as enumerated XML restrictions

XSD file for SBE Encoding Information

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:orchSbe="http://fixprotocol.io/2026/orchestra/encoding/sbe"
        targetNamespace="http://fixprotocol.io/2026/orchestra/encoding/sbe"
        elementFormDefault="qualified” attributeFormDefault="unqualified">

       <xs:element name="byteOrder">
          <xs:simpleType>
             <xs:restriction base="xs:string">
                <xs:enumeration value="Unspecified"/>
                <xs:enumeration value="LittleEndian"/>
                <xs:enumeration value="BigEndian"/>
             </xs:restriction>
          </xs:simpleType>
       </xs:element>

        <xs:element name="blockLength" type="xs:nonNegativeInteger"/>
        <xs:element name="offset" type="xs:nonNegativeInteger"/>
        <xs:element name="defaultStringLength" type="xs:nonNegativeInteger"/>
</xs:schema>
```

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

Using External Schemas in Orchestra

- Orchestra XML file needs to include the namespace “orchSBE” defined in the external schema to make the four SBE attributes (byteOrder, blockLength, offset, defaultStringLength) available.

- Whenever using an SBE attribute from the external schema, it has to be prefixed with the namespace, e.g. orchSBE:byteOrder, allowing its validation.

- The usage of an attribute depends on its meaning:
  - Attribute “byteOrder” is not specific to an element, it can only be used at the top level
  - Attribute “blockLength” defines the length of a message, group or component and can only be used where these are defined
  - Attribute “offset” defines the position of an element inside a message, group or component and can only be used where such an element is referenced
  - Attribute “defaultStringLength” defines the size of a string in the absence of a value and can be used for SBE schema generation where explicit length is required

Using External Encoding Schemas in Orchestra

- Repository references the external schema for SBE by declaring its namespace

- Orchestra element encodingStandards defines general information with a single element encodingStandard for SBE to define byte order as big endian and strings to have a default length of 4

- Definition of group MyGroup includes Orchestra element encodings with a single element encoding for SBE in two places
  - Defines offset of first field to be zero
  - Defines length of entire group to be 20

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<fixr:repository xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:fixr="http://fixprotocol.io/2026/orchestra/repository"
          xmlns:orchSbe="http://fixprotocol.io/2026/orchestra/encoding/sbe"
          name="Encoding with SBE" version=”1.0">

 <fixr:encodingStandards>
    <fixr:encodingStandard name="SBE" displayName="Simple Binary Encoding">
       <fixr:encoding>
          <orchSBE:byteOrder>BigEndian</orchSBE:byteOrder>
          <orchSbe:defaultStringLength>4</orchSbe:defaultStringLength>
       </fixr:encoding>
    </fixr:encodingStandard>
 </fixr:encodingStandards>

 <fixr:groups>
    <fixr:group id="1" name="MyGroup">
       <fixr:fieldRef id="1">
          <fixr:encodings>
             <fixr:encoding standard="SBE">
                <orchSbe:offset>0</orchSbe:offset>
             </fixr:encoding>
          </fixr:encodings>
       </fixr:fieldRef>
       <fixr:encodings>
          <fixr:encoding standard="SBE">
             <orchSbe:blockLength>20</orchSbe:blockLength>
          </fixr:encoding>
       </fixr:encodings>
   </fixr:group>
 </fixr:groups>

</fixr:repository
```

Orchestra and ISO 20022

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 Type is defined in both logical models (see separate presentation)