

Fall 2020



Don Mendelson, FIX Technical Architect

Jim Northey, GTC Co-chair, Americas and ISO TC68 Chair

Orchestra Assumptions...

- Large **scale** investment in FIX infrastructure not likely
- Few firms are providing resources to address operational inefficiencies
- Therefore, any automation must be tactical and incremental
- MUST address key FIX operational challenges
 - FIXatdl configuration
 - Certificate deployment issues
- Any automation / process improvement must be applicable to non-FIX protocols
 - FIX now has multiple encodings and multiple session layers
 - Many venues have proprietary binary interfaces

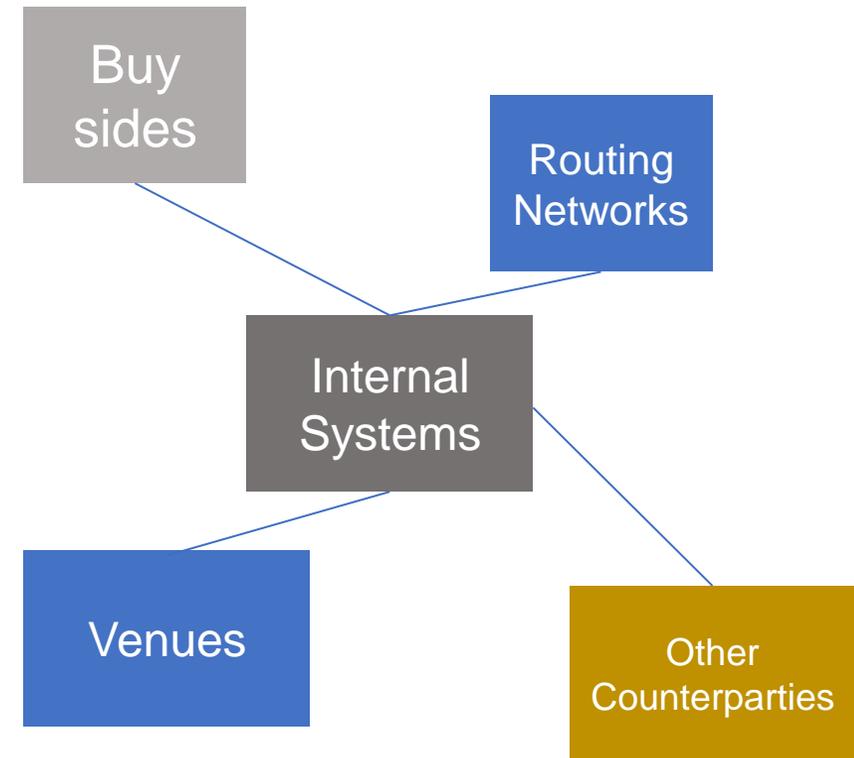
However...

- Unmet needs
 - Operational inefficiencies
 - Operational risks
 - Takes too long to onboard trading partners
 - Translation / normalization effort



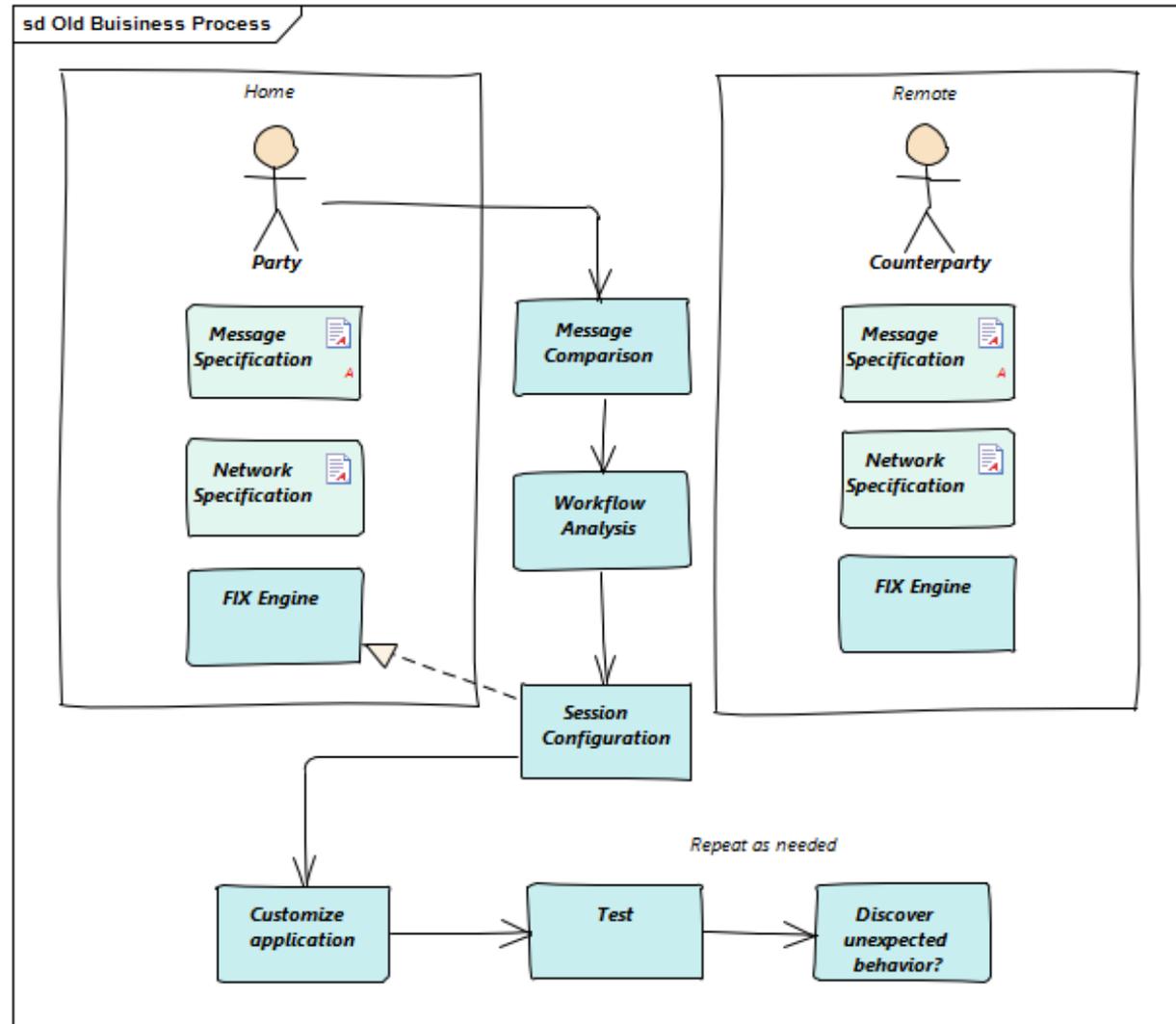
Where are the current pain points?

- Ambiguity and incompleteness of rules of engagement
- Must support multiple flavors of FIX
- Multiple internal messaging applications
- Multiple venue gateways



How to engage a FIX counterparty without Orchestra

“In short, the information we have is sparse and not directly actionable...”



Not to mention error prone as well!

Remember early Computer Interfaces?

- Having to configure device drivers?
- Setting DIP Switches on interface cards?
- Each interface needed an expert to get operational
- Then came the magic of plug and play



The goal of Orchestra is to make financial messaging protocols more plug and play



What is Orchestra?

Two aspects of Orchestra



Define the service...

Fields

Messages

Scenarios

Components and Groups

Codesets and values

Conditional Rules

Workflows

Actors and state transition

Deploy the service...

Application Layer

Version

FIXatdl

Encoding Layer

Session Layer

counterparty identifiers

security

Transport Layer

Connection details

FIX Orchestra enables DevOps best practices



What is FIX Orchestra?



What it is

- A standard for exchanging machine-readable rules of engagement
- Describes your service offering
- How we deliver the standard
 - FIX Latest
 - FIX.4.2
 - FIX Session Layer
 - Recommended practices

What it is *NOT*

- A product
- A wire protocol
- A replacement for your existing infrastructure
- A FIX version

Two interchangeable representations of Orchestra

Orchestra XML

Used widely in operations

Standard in the industry

Tablature

Simple

Human Readable

Human Writeable!

GitHub Markdown

md2orchestra

orchestra2md

Both are machine readable!

Here is an example in both XML and Markdown

```

</fixr:message>
  <fixr:message msgType="W" id="10116" name="MarketDataSnapshotFullRefresh">
    ...
  </fixr:message>
  <fixr:message msgType="D" id="10208" name="NewOrderSingle">
    <fixr:structure>
      <fixr:componentRef presence="required"/>
      <fixr:fieldRef id="11" presence="required"/>
      <fixr:fieldRef id="1" presence="required"/>
      <fixr:fieldRef id="18"/>
      <fixr:fieldRef id="111"/>
      <fixr:fieldRef id="55" presence="required"/>
      <fixr:fieldRef id="1300"/>
      <fixr:fieldRef id="54" presence="required"/>
      <fixr:fieldRef id="60" presence="required"/>
      <fixr:fieldRef id="38" presence="required"/>
      <fixr:fieldRef id="40" presence="required"/>
      <fixr:fieldRef id="44"/>
      <fixr:fieldRef id="99">
        <fixr:rule presence="required">
          <fixr:when>OrderType=4 or OrderType=5</fixr:when>
        </fixr:rule>
        <fixr:fieldRef id="59"/>
      </fixr:fieldRef id="432">
        <fixr:rule presence="required">
          <fixr:when>TimeInForce = 6</fixr:when>
        </fixr:rule>
        <fixr:fieldRef id="528" presence="required"/>
        <fixr:componentRef presence="required"/>
      </fixr:structure>
      <fixr:annotation>
        <fixr:documentation contentType="text/markdown">The New Order – Single
        message type is used by institutions wishing to electronically submit
        orders to a broker for execution.</fixr:documentation>
      </fixr:annotation>
    </fixr:message>
  <fixr:message msgType="F" id="10209" name="OrderCancelRequest">
    ...
  </fixr:message>
  <fixr:message msgType="G" id="10210" name="OrderCancelReplaceRequest">
    <fixr:structure>
      <fixr:componentRef presence="required"/>
      <fixr:fieldRef id="41" presence="required"/>
    </fixr:structure>
  </fixr:message>

```

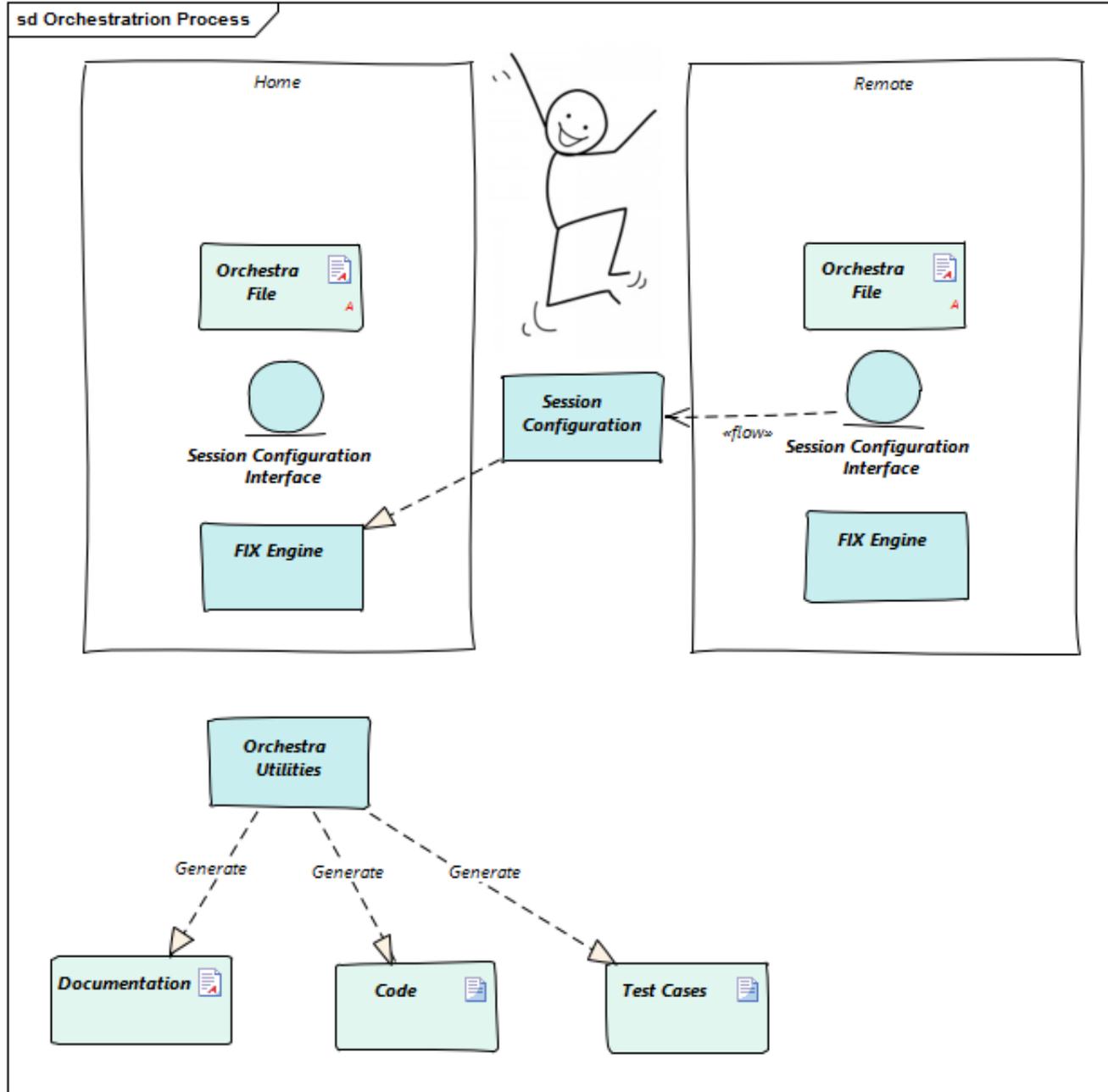
Message NewOrderSingle type D

The New Order – Single message type is used by institutions wishing to electronically submit orders to a broker for execution.

Name	Tag	Presence	Scenario
StandardHeader	c	required	
ClOrdID	11	required	
Account	1	required	
ExecInst	18	optional	
MaxFloor	111	optional	
Symbol	55	required	
MarketSegmentId	1300	optional	
Side	54	required	
TransactTime	60	required	
OrderQty	38	required	
OrdType	40	required	
Price	44	optional	
StopPx	99	required when OrderType=4 or OrderType=5	
TimeInForce	59	optional	
ExpireDate	432	required when TimeInForce = 6	
OrderCapacity	528	required	
StandardTrailer	c	required	

Message OrderCancelRequest type F

How to engage a FIX counterparty with Orchestra





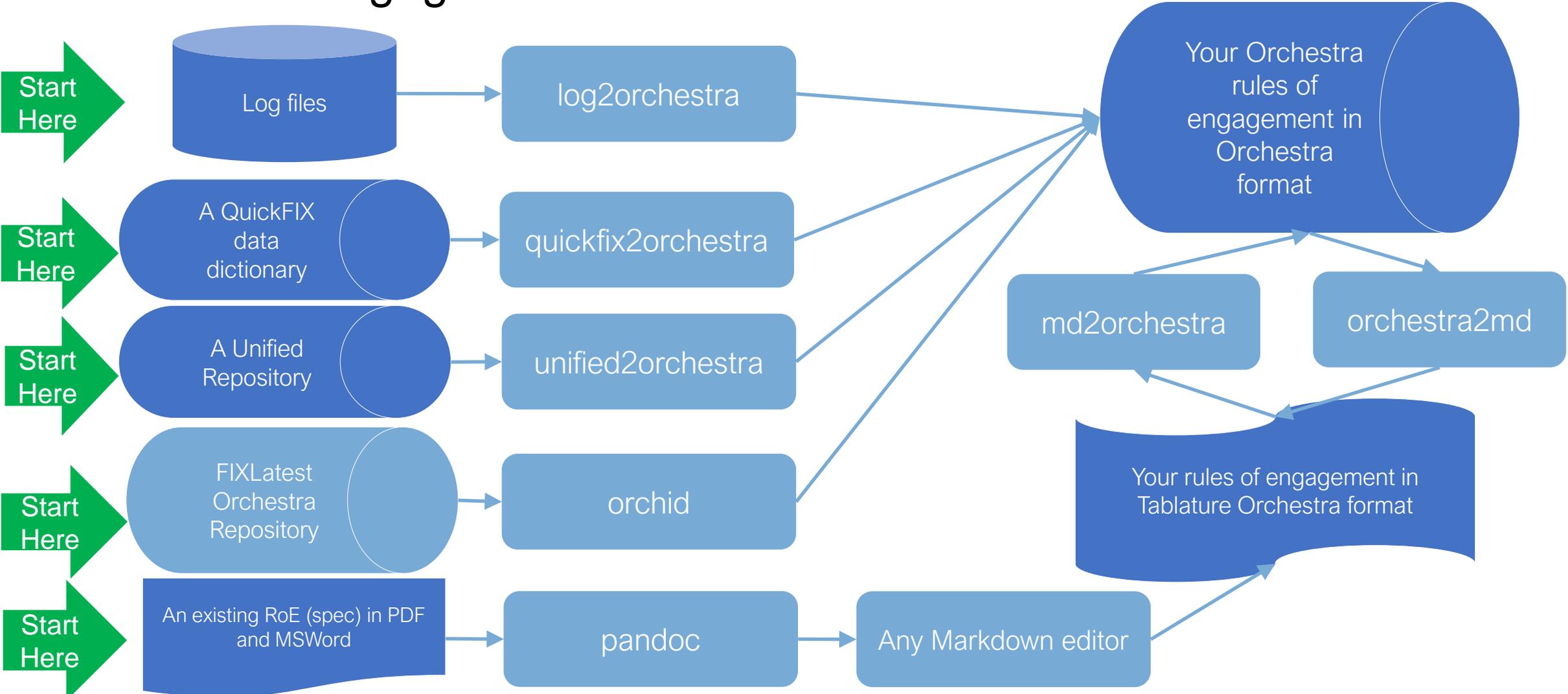
How can you use Orchestra?

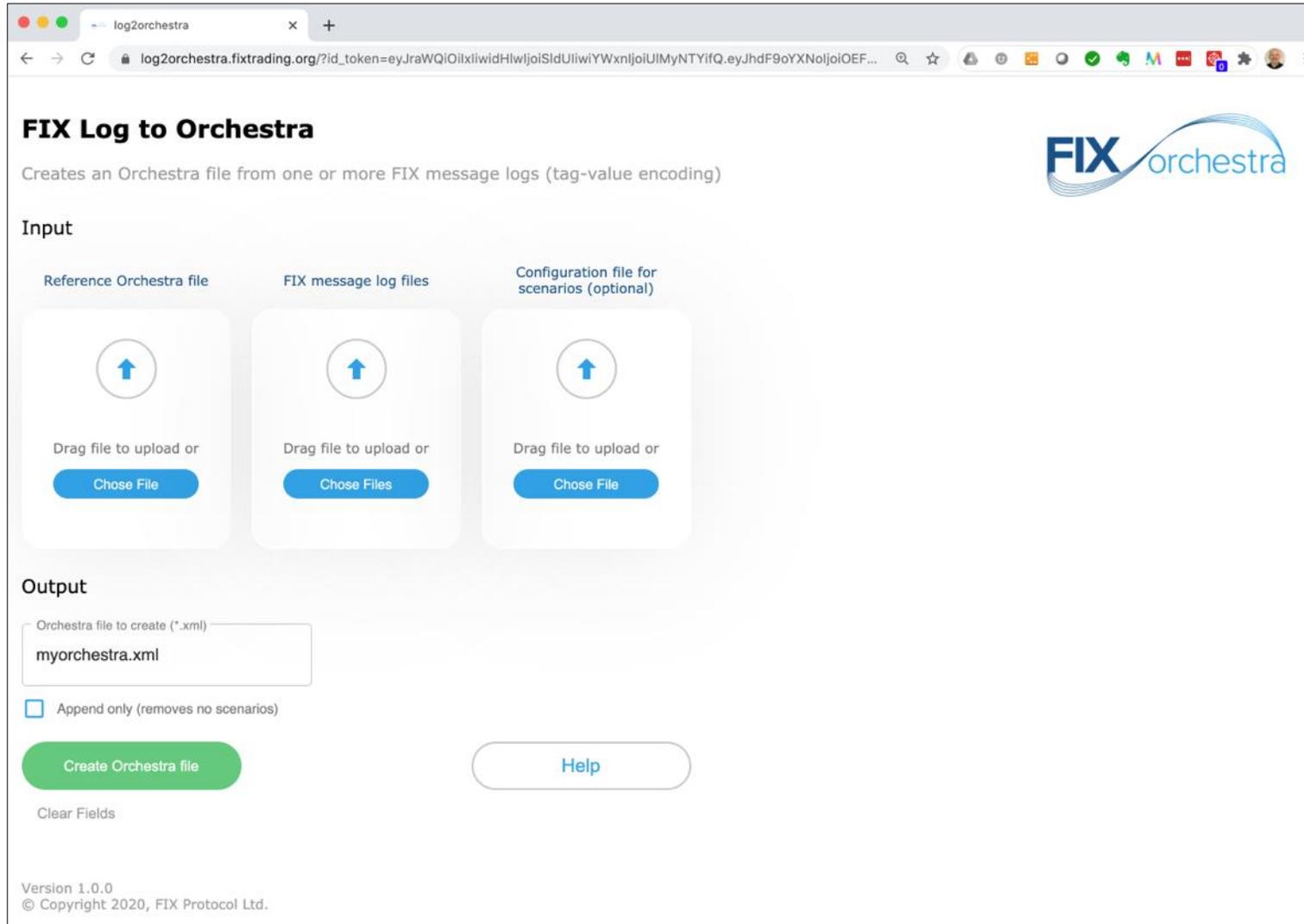
How can you benefit from Orchestra right now?

- Looking for a simpler way to create a richer and less ambiguous rules of engagement document?
- Need to document and understand your internal electronic trading infrastructure better?
- Just want to share message definitions and conditional fields?
- Want to extend to describe message responses, scenarios, and basic states?
- Want to fully model in detail the FIX service?

We can do it! let's start today!

Realizing your own orchestra machine readable rules of engagement





The screenshot shows a web browser window with the URL `log2orchestra.fixtrading.org/?id_token=eyJraWQlOiIiIiwidHlwIjoiaSdUliwiYWxnIjoiaUIMyNTYifQ.eyJhdF9oYXNoljoioiOEF...`. The page title is "FIX Log to Orchestra". Below the title is a description: "Creates an Orchestra file from one or more FIX message logs (tag-value encoding)". The "FIX orchestra" logo is in the top right corner.

Input

- Reference Orchestra file**: A card with an upload icon, the text "Drag file to upload or", and a "Chose File" button.
- FIX message log files**: A card with an upload icon, the text "Drag file to upload or", and a "Chose Files" button.
- Configuration file for scenarios (optional)**: A card with an upload icon, the text "Drag file to upload or", and a "Chose File" button.

Output

- A text input field labeled "Orchestra file to create (*.xml)" containing the text "myorchestra.xml".
- A checkbox labeled "Append only (removes no scenarios)" which is currently unchecked.
- A green "Create Orchestra file" button.
- A "Help" button.
- A "Clear Fields" link.

Version 1.0.0
© Copyright 2020, FIX Protocol Ltd.

A group of tooling to support Orchestra

log2orchestra

Ability to derive an orchestra file from

Atom/VS Code/ Zettlr/GitHub

Editors for tablature



FIXimate

Ability to generate custom
FIXimate

md2orchestra orchestra2md md2interface Interface2md

Converts between
tablature and orchestra

docgen

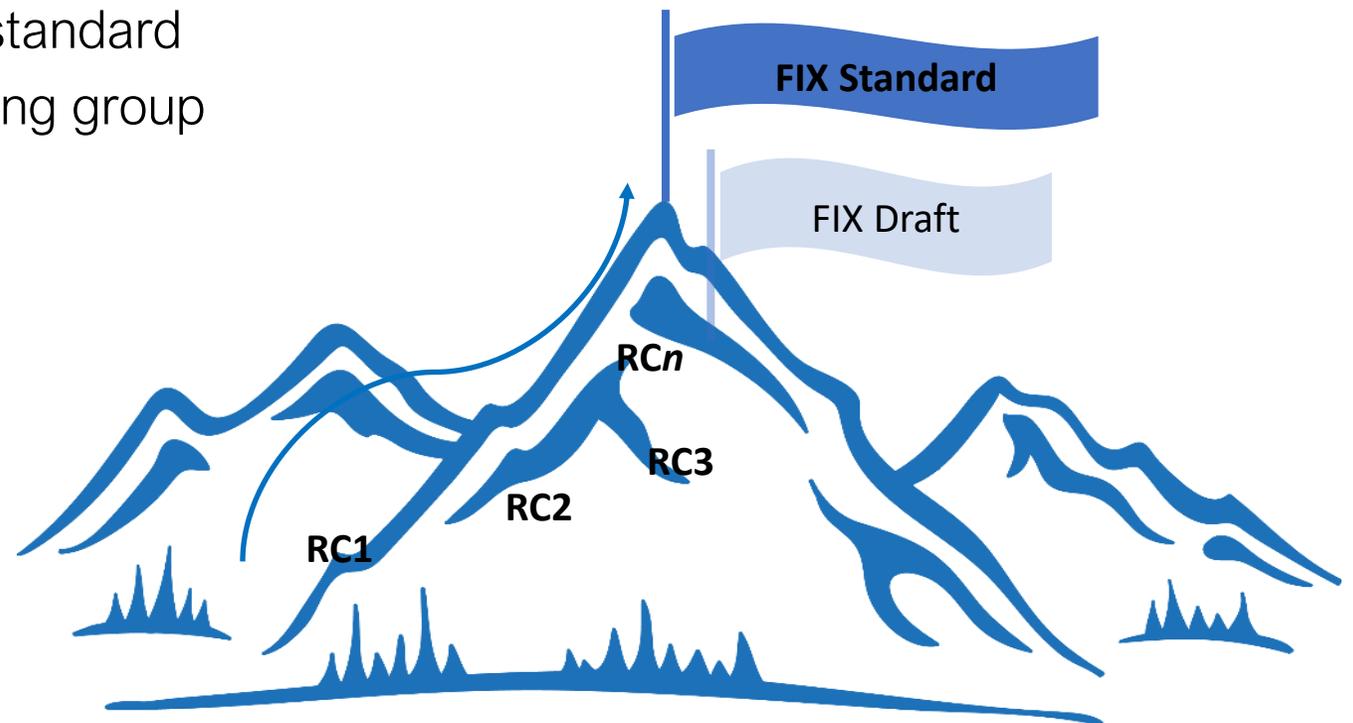
Generates an HTML version
of specification with
workflow diagrams and
state diagrams

FIX Orchestra supports innovation

- Possible uses and tools
 - Generate and run conformance tests
 - Capture best practices as an Orchestra file instead of text
 - Regulate internal flows within a large organization as well as between counterparties
 - Orchestra is a contract for behavior – use it to generate an emulator for testing
 - Analyze FIX logs for conformance to specified behavior
 - Let's go further...
 - Generate Execution Management, Order Management, Smart Order Routing, Order Matching behavior based upon exchange of state machine descriptions contained within FIX Orchestra files

FIX A New Level of Openness

- FIX develops technical standards in an open manner
- Technical Standards Process
 - Anyone can propose a technical standard
 - Anyone can work on a technical standard
 - Work is done by a technical working group
 - Available to everyone on GitHub



In conclusion...

- Orchestra is a standard to define an interface for service offerings
- You don't need to modify the internals of your applications
- Orchestra can be used to define your service in one place within *version control*
- Simple scripts can be created to read Orchestra files and update configuration files for various services
- You can start to benefit from Orchestra immediately with minimal investment



How can you get involved?

Want to get more involved?



FIX Orchestra
Working Group

MOST
Monitoring,
Onboarding,
Software Testing
Working Group

Create your
specification

Test open source and member
based tools and services