FIX India Training Session

Mumbai

September 2012

Hanno Klein, Senior Vice President

Deutsche Börse Group

Co-Chair FIX Global Technical Committee

Agenda

- Status of FIX adoption
- FIX High Performance Working Group
  - Standard Binary Encodings
  - Session Layer Optimization
  - Application Layer Optimization
- Case Study: Deutsche Börse Group

Work in progress

Current Status of FIX Adoption

- Increasingly rich application layer covering all asset classes as well as the vast majority of workflows in different environments
- Strong adoption of FIX 5.0 SP2 for clearing and market data where alternate transports are used (MQSeries, AMQP, UDP Multicast)
- Slow transition of trading and order routing applications from existing FIX 4.x engines to FIX 5 and FIXT 1.1 session layer
- Usage of FIX 5 concepts in FIX 4.x implementations (directly via tags from higher versions or indirectly via UDFs)
- Lack of high performance session layer and simple binary encoding main reasons for exchanges to continue use of proprietary protocols

FIX High Performance Working Group

- Problem
  - FIX is perceived to be slow and not suitable for high performance environments
  - FIX does not provide a simple binary encoding
  - Exchanges usually provide a FIX interface only in addition to a proprietary one

- Objectives
  - Adapt FIX to optimize high frequency transactions handling
  - Optimize FIX application level semantics for “lean” messages and flows
  - Optimize FIX session level semantics to support quick recovery and negotiation
  - Support additional encodings to optimize encoding/decoding speed versus bandwidth usage

- Subgroups
  - Application Layer Subgroup
  - Session Layer Subgroup
  - Encoding Subgroups (Simple Binary, Google Protocol Buffers, ASN.1)

Standard FIX Syntax – Status Quo

- FIXatdl, tag=value and FIXML all have ASCII encodings, only FAST offers a binary encoding
  - FIXatdl
    - <Strategy name="Tazer1" uiRep="Tazer" wireValue="Tazer" version="1" fixMsgType= "D" providerID="ABC">…
  - FIX tag=value
    - 8=FIX.4.2^9=92^35=A^49=BOFASEC0^…
  - FIXML
    - <IOI IOIID="4711" TransTyp="N" Side="2" Qty="200"…
  - FAST
    - FAST Templates describe message layouts
    - 81 84 41 4C CC 01 EA 91 82 E0 B1 FF 99 E0 B0
- FAST was designed for highly repetitive market data and is fairly complex (presence maps, stop-bit encoding, operators)
- FAST does not allow direct access to a single field (requires byte-parsing)
- Binary encodings increase performance but reduce wire-level legibility

Standard Binary Encodings – Requirements

- Support all FIX message types, fields and data types
- Support high performance encoding/decoding (vs minimum bandwidth)
- Support compact encoding (base data type vs implementation)
- Support non-ambiguous data encoding of FIX data types (e.g. NaN)
- Support standard data encodings (e.g. IEEE)
- Support random field access
- Support optimization agreed by counterparties (templates)
- Support automated translation between encodings (incl. tag=value)
- Support historical data (metadata, versioning)

Simple Binary Encoding

- Header information (message type, message length)
- Fixed length fields without metadata (values only)
- Fixed length repeating groups (preferrably at end of message)
- XML metadata statically describes the messages
  - Message layouts and field lengths
  - Value ranges (minimum, maximum) and special values (e.g. NULL)
- Timestamp fields
  - Single field with semantic defined by metadata (e.g. unit=nanosecs/epoch=Unix)
  - Date only and time only fields straightforward (unit=day or epoch=today)
- Price fields
  - Single field with implicit precision defined by metadata (e.g. decimals=4) or embedded as part of the value (n bytes for mantissa and m bytes exponent)
- String fields
  - Fixed length with or without termination character (e.g. for passwords)
  - Variable length via explicit length field prior to string field (e.g. for error texts)

Google Protocol Buffers Encoding

"Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data"

- Open specification
  - Publicly available (https://developers.google.com/protocol-buffers)
  - Flexible, extensible
  - Good performance
- Binary tag-value format, i.e. embedded metadata
  - Variable size messages
  - Absence of optional fields from wire format
  - No impact of new optional fields on existing applications
- External definition of data type encoding
  - https://developers.google.com/protocol-buffers/docs/encoding
  - Task: mapping of FIX data types to Protobuf data types

ASN.1 Encoding

ASN.1 (Abstract Syntax Notation One) defines the structure of messages to be exchanged between peer applications independently of local representation.

- International Standard
  - Support available from ASN.1 Project (http://www.itu.int/ITU-T/asn1/)
  - Widely used in mobile telephony, videoconferencing, air-ground communication
- Encoding Rules define wire format of application messages
  - Basic Encoding Rules (BER) use length-tag-value representation
  - Packed Encoding Rules (PER) do not transmit tags and length-value as needed
  - Octet Encoding Rules (OER) are like PER but 8-bit as smallest encoding unit
- ASN.1 Schema
  - Defines message structures and allows automatic generation of software code
  - Task: mapping of FIX messages, components, data types to ASN.1 types

Session Layer Optimization

- Current FIX session layer has not fundamentally changed since its introduction with FIX 2.7 back in 1995
- FIX Session Protocol version 1.1 (FIXT 1.1) was introduced with FIX 5.0 in 2006
- FIXT 1.1 has also not changed the basic concepts for message sequencing and session recovery
- HPWG is looking at two options
  - Optimization of FIXT 1.1 towards FIXT 2.0
  - New high performance session layer in addition to FIXT
- Topics of discussion
  - Session negotiation (e.g. encoding, recovery)
  - Recovery models (no recovery, windowed, asymmetrical)
  - Session level instructions (e.g. cancel on disconnect, throttles)

Application Layer – Alignment Issues

- Data Types
  - ASCII versus binary
  - MultipleCharValue, MultipleStringValue
- Entity Identification
  - IDs for orders, quotes, trades may be assigned by sender or receiver
- Transaction Models
  - 1:n translation (replication)
  - n:1 translation (bundling)
- Recovery Models
  - FIX Session Layer
  - FIX Application Sequencing

Application Layer – Semantic Verbosity

- Mandatory versus optional fields
  - AvgPx is only optional as of FIX 5.0
  - Order Cancel Request requires not only an order identifier and an instrument but also the side and order quantity
- Explicit versus implicit information
  - Single Execution Report for IOC/FOK orders
  - FIX 5.0 SP1 Specification Volume 7: Exchanges and Markets
- Message bundling
  - Multiple fills of the same order in a single Execution Report (SP1)
  - Mass action messages that can be optimized by receiver
- Echo of input from requests
  - Order submitter does not need attributes that do not change

Application Layer Optimization - Goals

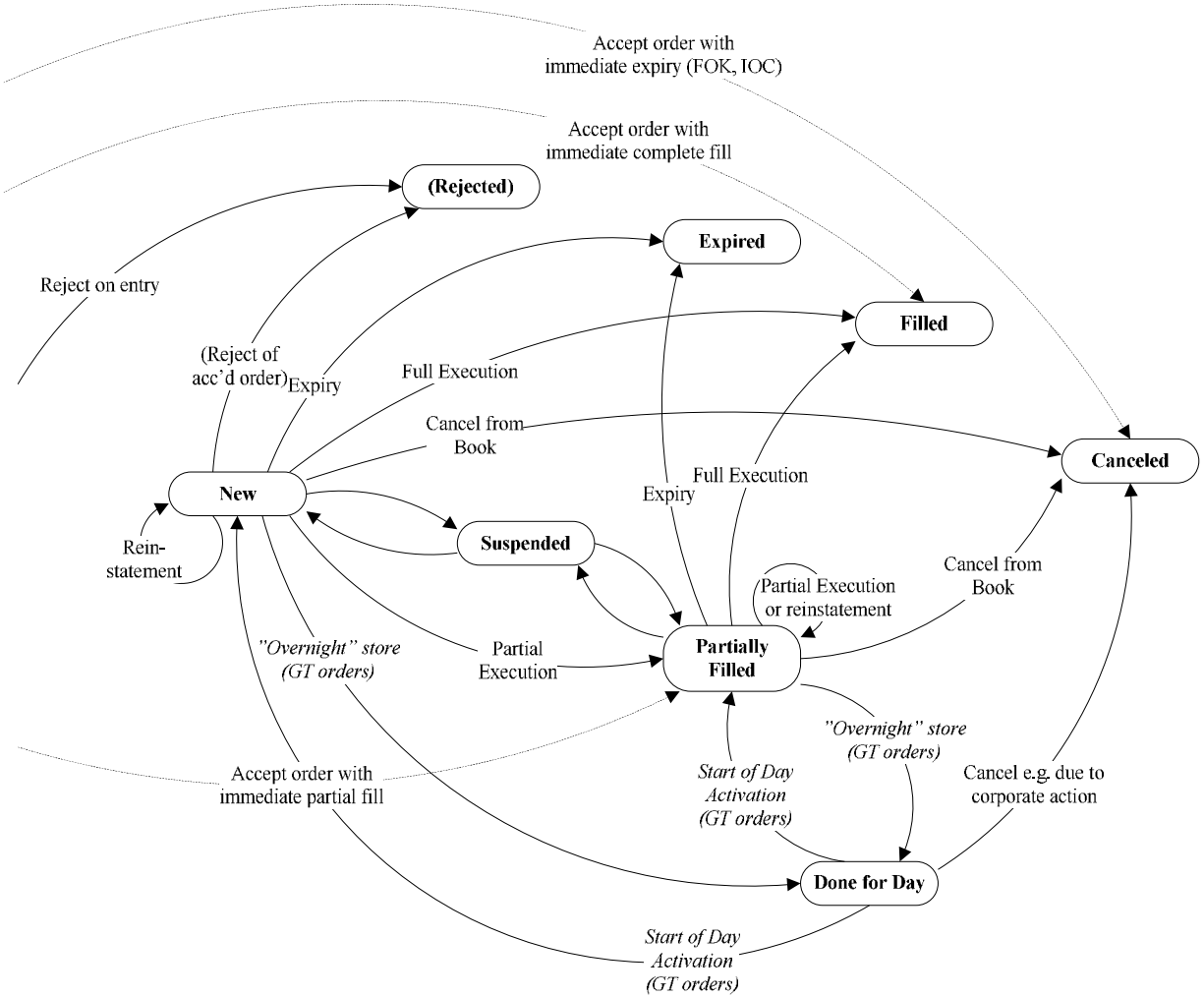
- Provide High Performance Guidelines
  - Document best practices on how to use FIX for high performance
  - Document high performance message flows

- Compile Gap Analysis
  - Propose new messages, fields and valid values
  - Extend existing concepts to better support high performance

Application Layer Optimization - Topics

- Mass Actions and Executions for Orders and Quotes
  - Mass entry of orders similar to quotes (MassQuote)
  - Mass actions for quotes similar to orders (OrderMassActionRequest/Report)
  - Mass execution of complex orders and quotes (leg executions, multi-instrument)
- Order State Information for High Performance Workflows
  - Fields OrdStatus and ExecType are key to understand an ExecutionReport
  - Event aggregation, i.e. single response for multiple sub-events that occur as part of an atomic matching engine transaction (e.g. IOC order)
- Application Session Types
  - Request/response session with minimal echo of input fields
  - Listener session for drop copy receivers with complete context
- Application Sequencing Extensions
  - FIX supports gapless sequence of application level messages for a given ApplID
  - Additional concept of unique application message keys for parallel processing

FIX Order State Changes



Conclusion

- FIX is a universal language for the financial industry, not just a technology.
- FIX can be used for many different interface types in combination with the appropriate transport.
- High performance can be achieved with FIX by integrating FIX semantics into the core system and using a binary transport.
- FIX Protocol Ltd. (FPL) will add a choice of standard binary protocols to its family of standards in 2013
- FIX 4.x Gateways using the FIX Session Protocol continue to have value for users that are not latency-sensitive and need easy access

Questions?

BACKUP

Case Study – Deutsche Börse Group

- Scope
  - Deutsche Börse technology supports many different exchanges and clearing houses across multiple asset classes (equities, bonds, derivatives, commodities)
  - Member interfaces comprise trading, clearing and market data
  - Interface strategy decided to transition away from proprietary interfaces with large footprint on member site to open, standard interfaces without any of our software needing to run on member hardware

- History of Events
  - FIX 4.2 / 4.4 Gateway to Xetra, Eurex, ISE trading since 2004 / 2006
  - FAST for Xetra and Eurex market data since 2007
  - FIXML 5.0 SP2 over AMQP for Eurex risk management since 2010
  - FIX 5.0 SP2 semantics for ISE trading since 2010
  - FIX 5.0 SP2 over FAST for Xetra and Eurex market data since 2011
  - FIXML 5.0 SP2 over AMQP for Eurex clearing since 2011
  - FIX 5.0 SP2 semantics for Eurex trading to launch in December 2012

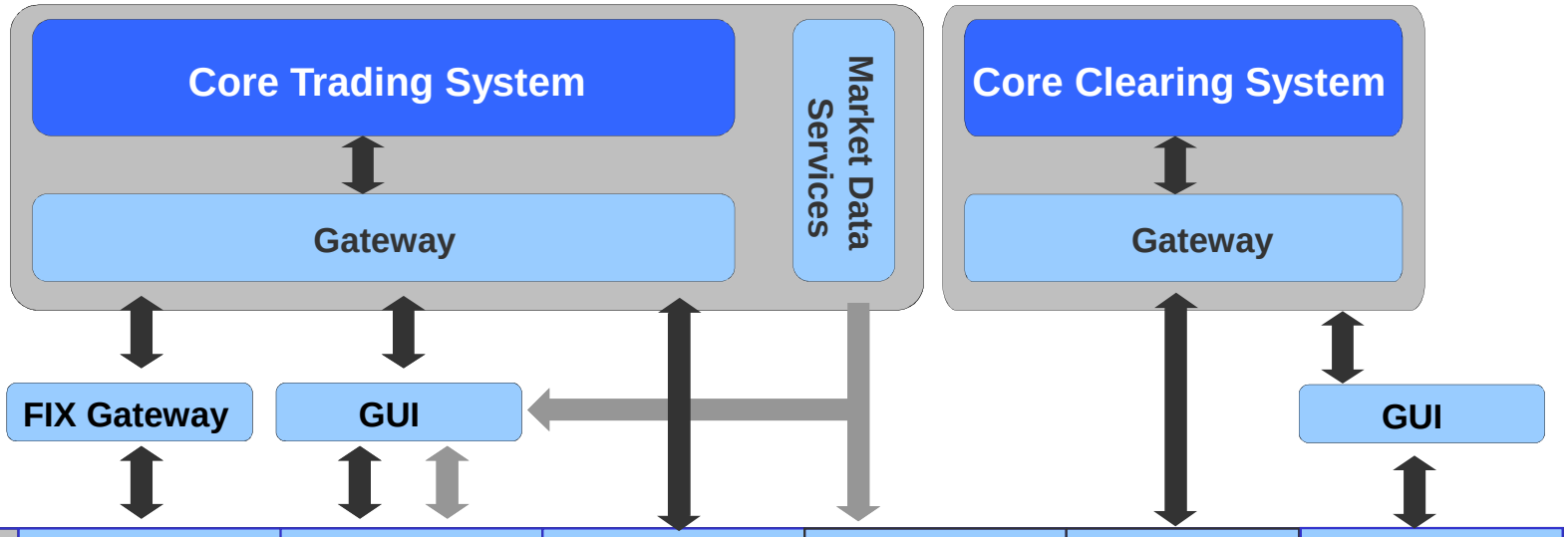
Case Study – Deutsche Börse Group

Standard Interface Protocols used for Trading and Clearing:

- FIX Protocol (Financial Information eXchange Protocol)
- FIXML (FIX Markup Language)
- FAST (FIX Adapted for STreaming)
- FpML (Financial products Markup Language)
- GPB (Google Protocol Buffers)[1]
- AMQP (Advanced Message Queuing Protocol)
- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)

[1]Currently only used for internal interfaces

Interface Architecture Framework



Protocol	FIX over FIX	Interactive	FIX over Binary	FIX over FAST	FIXML over AMQP	Interactive
Typical User	Order Routers	Standard User	Market Maker Algo Trader 3rd Party GUI	Market Maker Algo Trader 3rd Party GUI	Clearing Firm 3rd Party GUI	Standard User
Transport Layer	FIX 4.x, FIXT 1.x TCP/IP	N/A	**Binary** TCP/IP	FAST 1.x UDP Multicast	AMQP	N/A
Application Layer	FIX 4.x, FIX 5	N/A	FIX 5.0 SPx	FIX 5.0 SPx	FIXML 5.0 SPx	N/A

Functional Scope of Interfaces

- Areas of functionality
  - Order routing, interactive trading, algorithmic trading
  - Reference data (instruments, users)
  - Market data (book depth, state changes)
  - Clearing (on-exchange trades, OTC business)

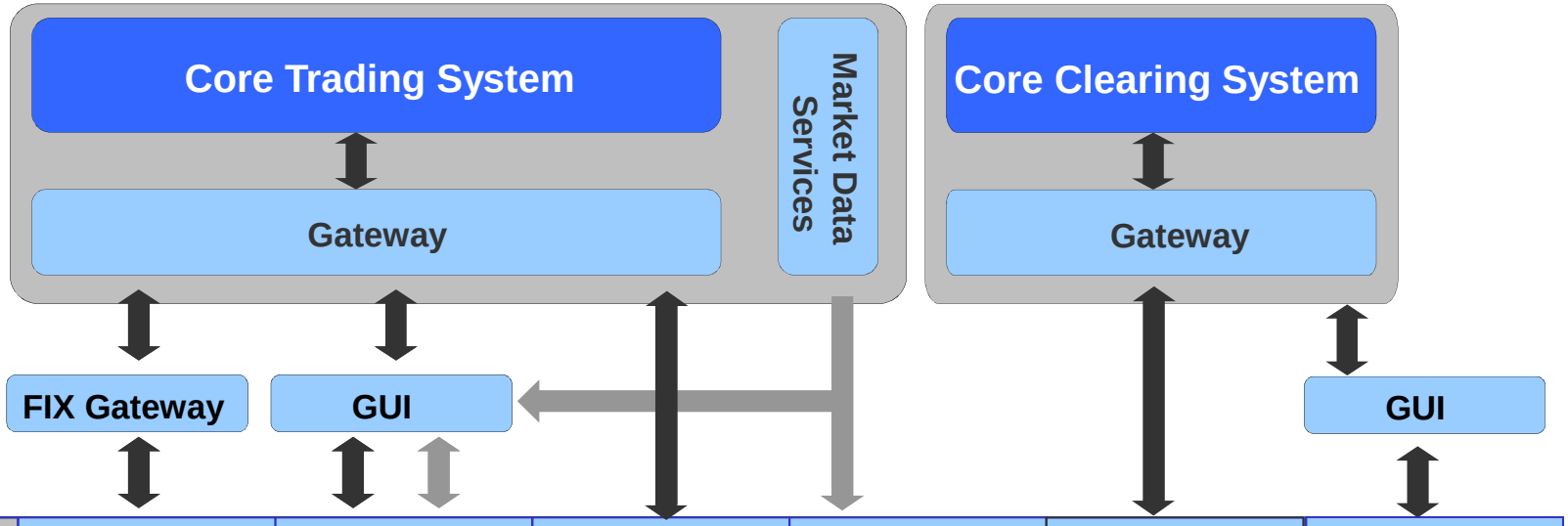
- Sub-areas of functionality
  - Order maintenance, mass quotes, request for quotes
  - Mass actions (deletions, (de)activations)
  - Trade notifications, trade reporting, trade adjustments
  - Give-up/Take-up, position maintenance
  - Administration (e.g. users, risk limits, news, stop button)

- Design Principles
  - Different areas of functionality require different types of transport layers
  - FIX semantics can be used for all programmable interfaces

Deutsche Börse Group Interfaces using FIX

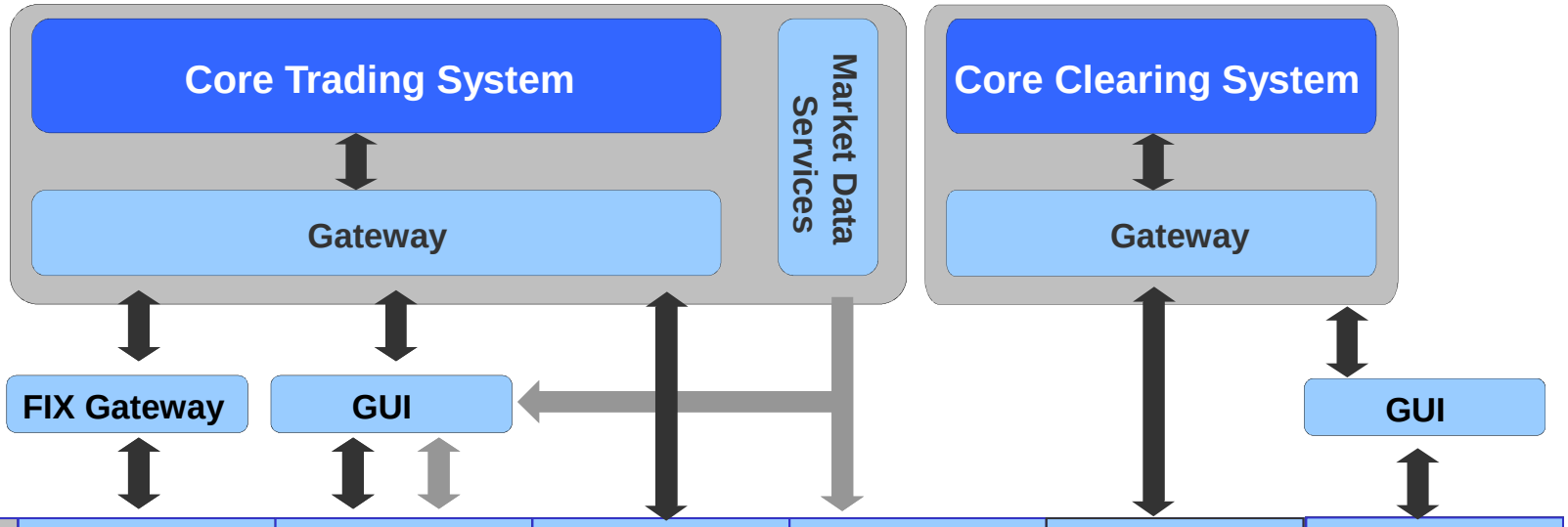
- International Securities Exchange (ISE)
  - FIX 5.0 SP2 over Binary for Trading and Market Making
  - FIX 5.0 SP2 over FAST 1.2 for Reference and Market Data
  - FIX 4.2, 4.3, 4.4 over FIX Session for Order Routing
- Xetra
  - FIX 4.2, FIX 4.4 over FIX Session for Order Routing
  - FIX 5.0 SP2 over FAST 1.2 for netted Market Data
- Eurex
  - FIX 4.2, FIX 4.4 over FIX Engine for Order Routing
  - FIX 5.0 SP2 over Binary for Trading and Market Making
  - FIX 5.0 SP2 over FAST 1.2 for Reference and (netted) Market Data
  - FIXML 5.0 SP2 over AMQP for real-time Risk Management
  - FIXML 5.0 SP2 over AMQP for Clearing & OTC Trade Entry

ISE Trading Interfaces and Standards



Protocol	FIX over FIX	Interactive	FIX over Binary	FIX over FAST	N/A	Interactive
Typical User	Order Routers	Standard User	Market Maker Algo Trader 3rd Party GUI	Market Maker Algo Trader 3rd Party GUI	Clearing Firm 3rd Party GUI	Standard User
Transport Layer	FIX 4.2, FIX 4.4 TCP/IP	N/A	Proprietary TCP/IP	FAST 1.1, FAST 1.2 UDP Multicast	N/A	N/A
Application Layer	FIX 4.2, FIX 4.4	N/A	FIX 5.0 SP2	FIX 5.0 SP2	N/A	N/A
Interface	ISE Order Routing System	PrecISE	Direct Trading Interface	Market Data Interface	N/A	N/A

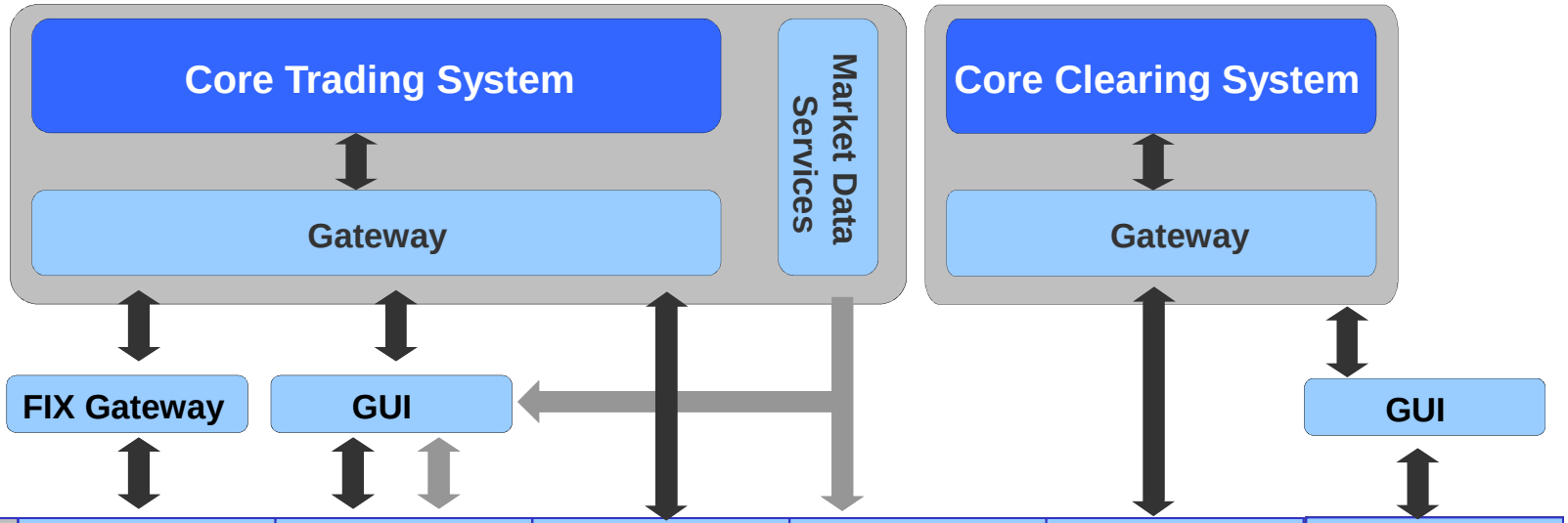
Xetra Trading/Clearing Interfaces and Standards



Protocol	FIX over FIX	Interactive	VALUES over Binary	FIX over FAST	VALUES over Binary	Interactive
Typical User	Order Routers	Standard User	Market Maker Algo Trader 3rd Party GUI	Market Maker Algo Trader 3rd Party GUI	Clearing Firm 3rd Party GUI	Standard User
Transport Layer	FIX 4.2, FIX 4.4 TCP/IP	N/A	Proprietary TCP/IP	FAST 1.1, FAST 1.2 UDP Multicast	Proprietary TCP/IP	N/A
Application Layer	FIX 4.2, FIX 4.4	N/A	VALUES	FIX 5.0 SP2	VALUES	N/A
Interface	FIX 4.2 Gateway FIX 4.4 Gateway	Xetra J-Trader	VALUES	Market Data Interface, Enhanced Broadcast Solution[1]	CCP	CCP GUI

[1]Enhanced Broadcast Solution has a proprietary application layer

Eurex Trading/Clearing Interfaces and Standards



Protocol	FIX over FIX	Interactive	FIX over Binary	FIX over FAST	FIXML over AMQP	Interactive
Typical User	Order Routers	Standard User	Market Maker Algo Trader	Market Maker Algo Trader 3rd Party GUI	Clearing Firm 3rd Party GUI	Standard User
Transport Layer	FIX 4.2, FIX 4.4 TCP/IP	N/A	Proprietary TCP/IP	FAST 1.1, FAST 1.2 UDP Multicast	AMQP	N/A
Application Layer	FIX 4.2, FIX 4.4	N/A	FIX 5.0 SP2	FIX 5.0 SP2	FIXML 5.0 SP2	N/A
Interface	FIX 4.2 Gateway FIX 4.4 Gateway	Eurex Trader GUI Eurex Admin GUI	Enhanced Transaction Interface	(Enhanced) Market / Reference Data Interface	Clearing FIXML	Eurex Clearing GUI

Functional Scope of Eurex Trading Interfaces

	FIX over FIX	Interactive	FIX over Binary	FIX over FAST
Order maintenance	YES	YES	YES	NO
Mass quotes	NO	NO	YES	NO
Request for quote	NO	YES	YES	YES
MM parameters	NO	YES	YES	NO
Mass actions	YES	YES	YES	NO
Execution notifications	YES	YES	YES	NO
Event notifications	YES	YES	YES	NO
Strategy creation	YES	YES	YES	YES
Reference data	NO	YES	NO	YES
Market data	NO	YES	NO	YES
State changes	NO	YES	NO	YES
User administration	NO	YES	NO	NO