

# Market Data Systems Testing

*Recommended Practices for AI-enabled testing of market data, ticker plant, consolidated tape, Securities Information Processor and direct-feed solutions that handle market data from trading venues*

---

# Table of Contents

<b>Executive Summary</b>	<b>3</b>
<b>Stakeholders and Drivers behind Consolidated Tapes</b>	<b>3</b>
<b>Market Data Systems Quality Assessment</b>	<b>4</b>
Market Data Parameters	4
Functional and Non-functional Requirements for Market Data Systems	5
Testing at the Intersection of Functional and Non-functional Requirements	7
<b>Market Data Systems Testing: Main Challenges</b>	<b>8</b>
<b>Recommended Practices for Market Data Systems Testing</b>	<b>9</b>
Testing with Dedicated Test Environments	9
Testing with Trading Platform Simulators	10
A Hybrid Approach to Testing	12
<b>Advanced Results Analysis to Enhance Ticker Plant Quality Assessment</b>	<b>13</b>
<b>Recommended Market Data Systems Testing Practices: Conclusion and Main Takeaways</b>	<b>14</b>

---

## Executive Summary

As the European Securities and Markets Authority commits to developing a [pan-European consolidated tape system](#), and while the US Options Price Reporting Authority (OPRA) [proposes the dynamic channel rebalance](#) for its feed and the US exchanges work with the Securities and Exchange Commission (SEC) to introduce [further efficiencies to the US National Market System \(NMS\)](#), the issue of coordinating efforts and aggregating the industry's best practices to ensure the quality and reliability of market data feeds becomes especially acute.

Years of experience with testing some of the industry's mission-critical infrastructures in the trading and post-trade space enable Exactpro to make a meaningful contribution to the discourse. These recommended practices identify the main functional and non-functional market data systems requirements, the challenges typically encountered in their testing, and the test approaches proven most effective in assessing and controlling the quality of quote aggregation, processing – including the accompanying data enrichment – and dissemination.

Highlighting the unique architecture of market data solutions and surrounding components, the paper elaborates on an AI-enabled approach to independently assessing their quality to enhance their functionality, resilience, and compliance. The approach is rooted in data-driven testing principles and involves the creation and maintenance of market simulators (digital twins). Market simulators help streamline test scenario generation and, ultimately, help create an extensive and yet highly resource-efficient test library. The paper also presents test coverage analysis methods and challenges involved in planning and assessing the test coverage of complex systems.

---

## Stakeholders and Drivers behind Consolidated Tapes

These recommended practices can be used as a reference guide for efforts in developing or improving the quality of market data systems such as ticker plants or direct-feed solutions (the terms “market data systems” and “ticker plants” will be used interchangeably in the paper). The recommended practices are equally applicable to **consolidated tape systems** such as the US **Securities Information Processors (SIPs)** tasked with disseminating the **National Best Bid and Offer (NBBO)**, the **Canadian Best Bid and Offer (CBBO)** system as well as the pan-European consolidated tape system, plans for which are currently underway.

Other stakeholders include issuers, institutional market participants on the buy- and sell-side, retail investors, execution venues, clearing, settlement and custody systems operators, regulators and analytics providers that all benefit from the transparency provided by timely access to accurate market data.

High-quality and consistent market data and a well-coordinated market structure is a precondition for effective consolidated tape systems that aim to provide market participants with shared access to a full and precise cross-asset representation of the market. Due to the variety of trading and execution venue types and their corresponding instruments, market data tends to be disseminated per asset class via dedicated tapes, which presents an additional testing challenge.

The recommended best practices outlined in the paper aim to help individual market data providers as well as consolidated tape infrastructure providers ensure the quality and resilience of their feeds.

---

## Market Data Systems Quality Assessment

Market data volumes are growing at a rapid pace. The ability to handle the entire stream of market data and the capacity to do it continuously for each venue and each financial instrument are the main characteristics of the high-load systems that ticker plants are. Market data systems / ticker plants are mission-critical, low-latency infrastructures aggregating, processing, enriching and disseminating market data to groups of subscribers over normalized protocols in real time. In the algorithmic trading setting heavily reliant on timely and accurate data for informed decision-making, ticker plant testing is a vital aspect of price discovery that underpins the quality and reliability of the entire infrastructure. With both buy-side and sell-side firms, retail investors as well as financial market infrastructures (FMIs) across the globe relying on their multicast feeds, market data quality is a global-scale issue.

These recommended practices summarise Exactpro's testing and test coverage analysis methodology that leverages artificial intelligence (AI) and machine learning (ML) methods to explore the quality and resiliency of ticker plant systems and identify outstanding issues. The approach can also help to pinpoint gaps in existing test libraries, to increase the quality and efficiency of established test practices.

AI Testing is a cross-asset technology-agnostic technique for test library generation and optimisation effectively used for validating the quality of integration of market and reference data feeds, among other data flows integral to the functioning of FMIs. Exactpro's data-driven software testing practice is rooted in [award-winning synthetic test data generation capabilities](#) that have been successfully used across projects with major global FMIs for over 15 years.

To evaluate and validate the quality of financial systems, it is important to be able to verify their behaviour under various conditions. For ticker plants, as well as for a major part of systems in the capital markets space, this behaviour is governed, among other elements, by upstream sources' events, their reference data as well as the ticker plant's own reference data – which creates much of the context for intra- and inter-system data flows.

### Market Data Parameters

Market data features the following parameters per financial instrument: **Ticker Symbol, Last Trade Price, Best Bid, Best Offer, ISIN** (International Securities Identification Number) or other ticker identifiers, **Exchange Code, Trade Time, Open/Close Prices**. Depending on the complexity of the exchange, market data can be processed by the electronic venue's internal components and enriched with additional information, e.g. the daily **turnover, VWAP**, and more detailed information about the stock or the derivative, i.e. Reference Data, including, for instance, parameters of traders, the market, trading sessions, and instruments. Reference Data (also known as Static Data) is instrument information that does not change in real time, e.g. the ISIN, price at the close of the previous day's trading session (Close Price), Currency, the Circuit Breaker parameters that are typically represented as percentages of the last trade price (Dynamic or Static Circuit Breaker Tolerances (%)), and so on.

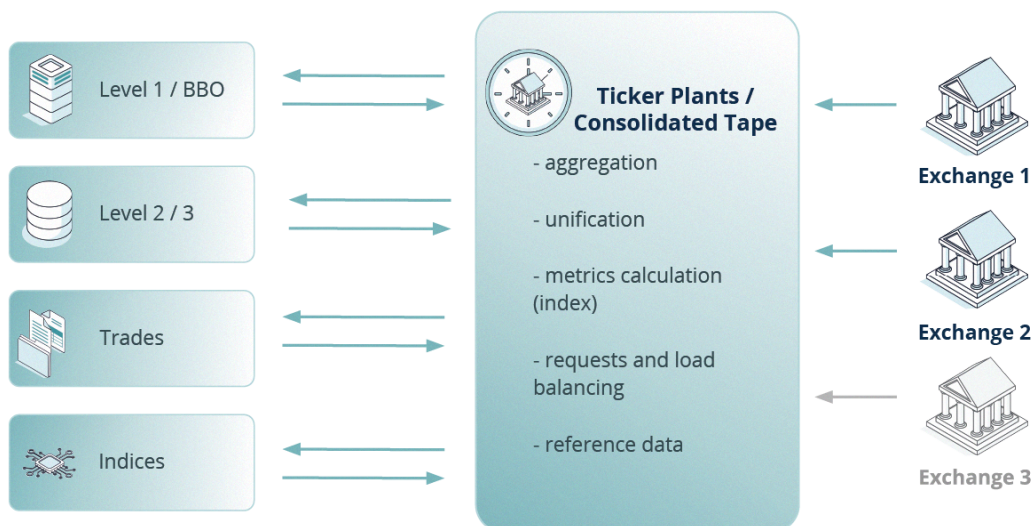
The information is disseminated via a number of protocols: text-based protocols such as FIX/FAST (Financial Information Exchange and FIX Adapted for Streaming, respectively) or STAMP (Securities Trading and Access Message Protocol), binary protocols like ITCH, Pillar or HSVF (High Speed Vendor Feed). There are also cases when data is being streamed over the WebSocket API. To transmit quote information, many electronic exchanges use both standard and bespoke protocols. Traders often lack the capacity to develop applications that would collect quote information required for their work. This supports the need for centralised systems capable of efficient aggregation of market data from the spectrum of financial data transmission protocols.

## Functional and Non-functional Requirements for Market Data Systems

Ticker plant systems (represented in a simplified way in **Fig. 1**) provide collected market data to the traders or other software applications in a normalized or unified format. Based on the accumulated market data, ticker plants often calculate additional parameters, supplementing the static data about fixed-income products, equities, derivatives, currencies and other financial instruments. A key characteristic of ticker plant systems is their ability to collate homogeneous values of the disseminated quotes – price levels with volumes, trades, best bids and best offers, session states, market announcements, and so on – and distribute the quote data real-time based on the subscription levels and specific requests from clients, as well as store the disseminated market data.

The types of market data services mentioned below are widely used in the electronic financial instruments trading industry:

- *Level 1* is information about best Bid/Ask (aka TOP, Top of Book), OHLC (Open, High, Low, Close), Volume.
- *Level 2* is information about Level 1 and the Order Book (5-10 depth levels of the order book, Bid/Ask Price and Quantity). Level 2 can contain attributed or anonymous orders and use models, such as MBO (Market by Order – an individual breakdown of orders in each price level) and MBP (Market by Price – an aggregated view of orders within a price level).
- *Level 3 – Total Order Book View* is more complete detailed information compared to Level 2, which offers significantly more depth in market information – full depth of book for all trading securities.
- A consolidated *Best Bid and Offer (such as NBBO / CBBO)* is information about the best available ask price and the best available bid price for each security across all exchanges where the stock is listed.
- *News* is the latest corporate news.
- *Indices* are indices-related data that is usually represented as calculated statistical measures of changes in various security prices; each index related to different markets has its own calculation methodology.



**Fig. 1** A simplified ticker plants / consolidated tape system schema

Based on the description of a ticker plant system and its main characteristics noted above, we can provide a set of requirements the system should comply with. This set of characteristics is followed closely in testing. To facilitate the assessment of each of the characteristics, we have divided them into functional and non-functional ones. From a functional standpoint, the following requirements can be defined:

Functional Requirements	Description
<b>Quote information collection</b>	Quotes are collected from multiple market data suppliers such as stock exchanges, OTC markets, banks and other financial data providers. These venues disseminate information via various data transmission protocols.
<b>Real-time data collection and processing</b>	The market data information like quotes, trade prices, volumes, best bids and best offers, and National best bids and offers is acquired, processed and enriched as market events happen.
<b>Reference data processing</b>	Correct processing of the reference data provided by the source venues (which happens to be provided in different formats).
<b>Data normalisation</b>	Converting the diverse data and data formats into a consistent, unified format enabling it to be further distributed to subscribers via a unified protocol.
<b>Aggregating quotes information according to various described methods</b>	Aggregating the market data allows ticker plants to reduce throughput and latency. The aggregation methods or snapshot data are usually represented in the form of logical rules which require functional validation.
<b>Enriching the market data information</b>	Processing the market data in order to enrich the system's functionality, e.g. provide contextual information such as timestamps, source identifiers, and so on, or statistics calculations such as <b>VWAP, Turnover, Trade High/Low, 52 week Trade High/Low</b> , etc.
<b>Subscription levels and customisation of subscriber requests</b>	Provisioning of market data according to the clients' requests, their preferences or subscription levels (for example, <b>Level 1 / BBO, Level 2 / 3, Time and Sales (T&amp;S), News, Indices, Option chains</b> , etc.).
<b>Replaying and recovering market data</b>	Such functionalities provide mechanisms to <b>replay</b> and <b>recover</b> missed data for subscribers or connected downstream systems, should subscribers experience any data gaps in the market data streams.
<b>Historical data about quotes</b>	Providing recorded historical data about quotes through the real-time channels and per the subscribers' requests.

Due to the nature of high-load systems and their potential non-determinism, non-functional characteristics have to be factored into the testing on par with functional checks. From a non-functional standpoint, ticker plants are subject to the requirements listed below:

Non-Functional Requirements	Description
<b>Low-latency market data processing</b>	Providing fast processing of the quotes' data streams received from the source venues <b>in real time</b> .
<b>Real-time market data dissemination to subscribers and processing their requests</b>	Real-time dissemination of the enriched and calculated (like NBBO, High/Low Trade Prices, etc.) market data throughout primary and secondary channels (also called A and B feeds or main and retransmission distribution networks) and fast processing of requests received from clients and quotes data, depending on their subscription level and client request type, e.g. separately for a traded instrument or a group of instruments, or the entire market.

<b>Continuous working efficiency</b> of the system	Ensuring <b>high uptime</b> and <b>working system availability</b> and implementation of reliability through the <b>failure and recovery mechanisms</b> of the consolidating data feed handlers.
<b>Operability</b> of the market data system	Providing effective and efficient system management mechanisms.
System <b>monitoring</b> and <b>logging</b>	Providing the ability of system monitoring and <b>alerting</b> , i.e. the availability of applications to monitor the system and <b>operate</b> its components with alerts for any signs of failure or current issues, including the systems' non-functional characteristics such as performance, latency and throughput. Allowing <b>proactive manual intervention</b> where/if needed. The related logging information for further troubleshooting and analysis.
Providing <b>throughput</b>	The capability of handling <b>high volumes</b> of data without <b>performance</b> degradation and applying the Fair Access Rule at the same time, especially during peak market activity.
Providing <b>fault tolerance</b>	<b>Redundancy:</b> duplication of the components in case one fails, so that another could take over seamlessly. <b>Failover:</b> automatic detection of the failures and switching to the backup components/systems without human intervention. <b>Load balancing:</b> data processing distribution across multiple nodes or systems to avoid the occurrence of a single point of failure from overloading.
<b>Disaster recovery</b>	Ensuring robust disaster recovery plans to handle <b>data center</b> failure events with minimal data loss. Evaluation of <b>recovery time</b> in such scenarios.
<b>Maintainability</b>	During very high system uptime, system <b>updates</b> or <b>patches</b> must be within the limits of <b>minimal downtime or disruption</b> .

## Testing at the Intersection of Functional and Non-functional Requirements

The inclusion of both functional and non-functional validation checks into the test library is not enough. In a complex distributed system, it is crucial to understand the possibility of hidden interdependencies that will only reveal themselves at the intersection of the two. Some potential situations of non-determinism occurring under specific non-functional conditions can be found below.

- Any complex system must sustain a certain level of concurrency. Some issues appear when several events happen simultaneously (race conditions). When several events converge in the component at the same time, the state of the message or component can get corrupted. Such race conditions can appear when a high-frequency load is applied.
- Production-like randomisation adds more complexity to the system events that could happen within the scenario, and ticker plant components may behave differently in such complex scenarios combined with load conditions.
- Integer overflow in certain scenarios – the fields in the statistics messages that are constructed through calculation and fields representing incremental identifiers can overflow when system events leading to these cases are generated repeatedly.
- Dynamic mass events in the source upstream system or in the ticker plant which happen during the Daily Life Cycle (DLC), i.e. opening, closing, auction uncross, etc. The state of the system when these DLC events happen should be verified with the load applied against the system.

Having defined a set of functional and non-functional requirements to be met by ticker plants – and keeping in mind the importance of testing at the intersection of functional and non-functional characteristics – we should also identify the main challenges that software testing activities should address.

---

## Market Data Systems Testing: Main Challenges

Due to the dynamic nature of electronic trading and the structural complexity of the infrastructure and its interconnected process flows, a ticker plant testing approach must be able to account for the following challenges:

### Functional requirements and business-related challenges

- Accurate processing of quotes collected via multiple sources with different transportation protocols and recovery mechanisms.
- Coexistence of disparate and complex data formats and protocols.
- Data normalisation and enrichment mechanisms which vary due to the different sources and calculation mechanisms for enrichment.
- Permutations of the subscription layers and requests leads to a high number of combinations and challenges with stream consistency.
- Increased volumes of historical data due to multiple sources and enrichment mechanisms.
- Operability, maintainability and system monitoring challenges due to a variety of the logic components in such distributed systems.
- Adherence to strict regulatory requirements.

### Infrastructure and non-functional requirements-related challenges

- High data volumes that are prone to fluctuations
- Ultra-low latency real-time performance
- Possibility of data loss or corruption
- Possibility of interrupted delivery of data
- Time synchronization across the infrastructure
- Integrations with surrounding systems

### FPGA-related challenges

The ultra-low-latency handling of market data is made possible via the use of FPGA technology. Its high cost and distinctive architecture – implementing the algorithm partly in hardware and partly in software – incur another set of unique challenges:

- *Cost efficiency* of the environment: it makes it difficult to dedicate *a sufficient number of test environments* – a major contributing factor of a successful QA process – to the testing process.
- *Maintainability* and *operability* of ticker plant environments.
- *Test case development*: the hybrid nature of the application makes it an extremely complicated process to develop a proper set of test cases with a comprehensive level of coverage.



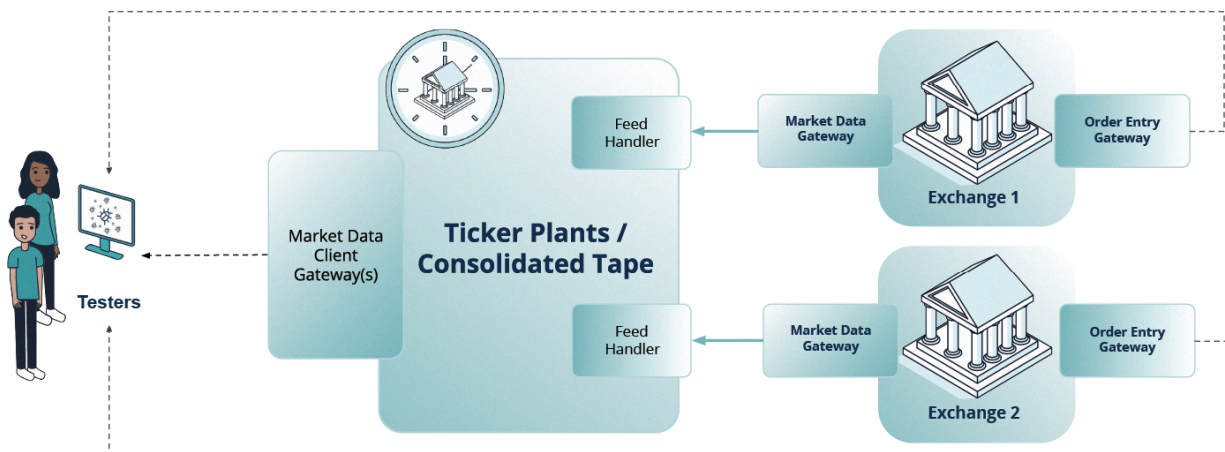
# Recommended Practices for Market Data Systems Testing

Equipped with the main functional and non-functional characteristics of market data systems and the understanding of the main challenges, we can move on to defining the testing approach most suitable for the task. The two testing approaches proposed (illustrated in a simplified way in **Fig. 2** and **Fig. 3**) can be implemented separately or as a hybrid solution, depending on project conditions.

## Testing with Dedicated Test Environments

“Dedicated test environments” stand for the environments made available by the upstream systems / exchanges under test for integration with third-party applications. In this setup, and when the exchange test system is fully available for market data system testing, the methods used in trading systems quality assessment can be applied (for a detailed description, please refer to the [ARTEX-Exactpro case study on AI-enabled Software Testing for ARTEX MTF](#)). As part of the approach, AI-based models help to generate test scenarios with sufficient permutations of relevant variables, parameters and system actions. These test scenarios are then enhanced with additional ticker plant-related interface validations.

In this approach, the existing test libraries created for ticker plant source venues (exchanges) are reused via conversion to a test library of a more complex system under test (one involving both the source venue and the ticker plant outputs). Since the ticker plant's dissemination logic is known from specifications, the test library conversion process can be fairly easy: the ticker plant's dissemination logic is applied to the events in the test library. In other words, this approach extends AI-enabled models of trading systems with AI-based models of corresponding ticker plants. Such a dynamic testing approach allows verifying a fairly large spectrum of scenarios of correct market data system functioning and operation.

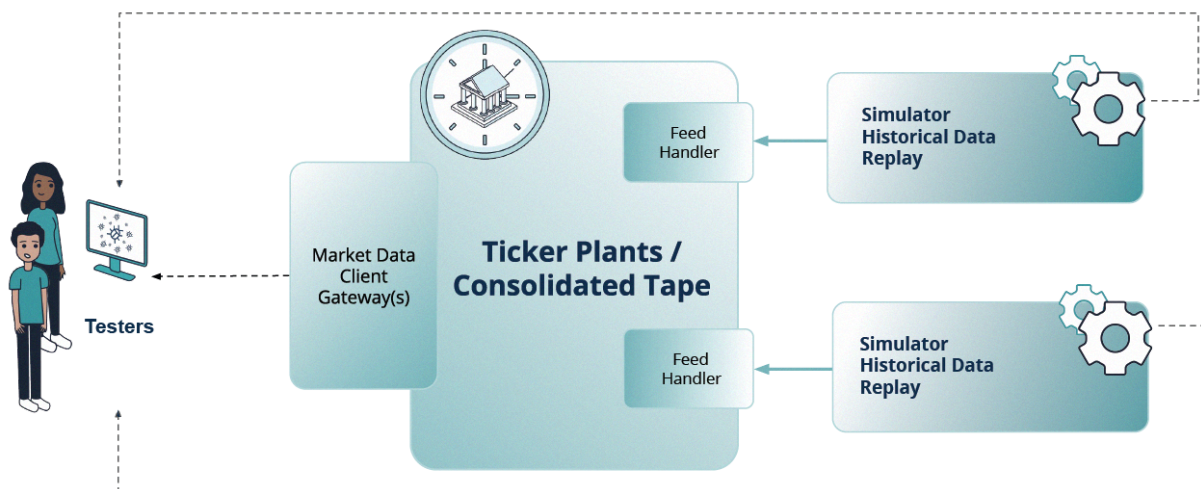


**Fig. 2** Approaches to testing ticker plants / consolidated tapes: testing with dedicated test environments

Due to low availability of real upstream test systems or simply insufficient historic coverage of conditions required for testing, real data can not always be obtained or may not be representative enough to meet all testing needs. To enable comprehensive testing – especially in the context of using AI, with its high demand for massive datasets and the sharp move away from hard-coded data and towards data siloed in configuration files – software testing benefits from system modelling/simulation with the use of synthetic data. The data can be hand-crafted (produced manually to hit specific conditions) or generated automatically based on specific rules, or be a combination of both.

## Testing with Trading Platform Simulators

Another approach that can be implemented is **testing by using trading platform simulators** (models, AI-enabled digital twins) and **historical data replay tools** developed based on the information available about the venue and/or obtained using various testing and data mining techniques. A market (or exchange) simulator is an interactive computer program developed in order to simulate the possibilities and features of real market models. A simulator must emulate real events that take place on an exchange: the trading itself, DLC changes, the calculation of auction prices, opening and closing prices and their publication, etc. A simulator has an API, similar to one a real exchange would have, with elements of controlling, scheduling and configuring responses sent to subscribers (including ticker plant feed handlers).



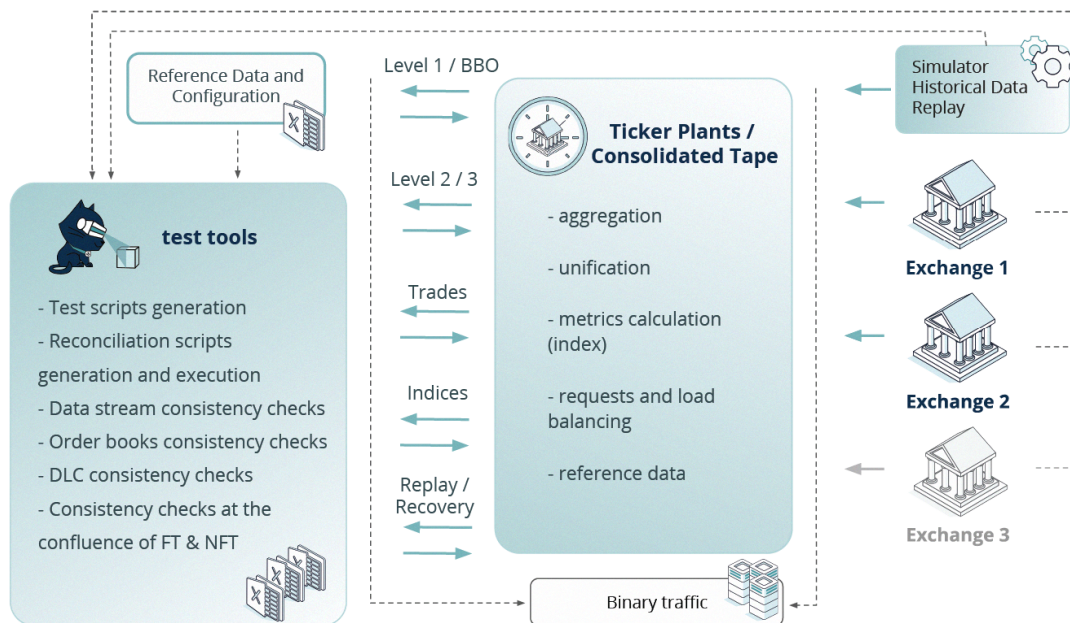
*Fig. 3 Approaches to testing ticker plants / consolidated tapes: testing by using trading platform simulators and historical data replay tools*

Such exchange simulators allow having fuller control over the events generated for a Ticker Plant, which significantly expands testing capabilities and, ultimately, enables more extensive test coverage. Simulators allow creating the necessary load comparable with data streams characteristic of real highly loaded exchange and broker systems. Over the years, Exactpro has developed a variety of simulators collating insights and experience from multiple projects. These AI-enabled digital twins of the exchanges are enriched with a variety of [test scenario generation algorithms](#) based on the source venue complexities, interfaces and field permutations.

Using historical data in testing creates a controlled environment, enabling the generation of repeatable tests under known conditions – which is impossible to do with live data. Such an approach also represents additional tooling for performance benchmarking: with historical data, it is possible to simulate various market conditions, from quiet to highly volatile periods, to test system scalability and performance.

In the schematic depiction of testing a ticker plant system (**Fig. 4**), we combine both test approaches. While executing such tests, we collect binary input sent to the system under test and binary output traffic generated by the system, for parsing and further interpretation for additional layers of the validations.

As visible from the **Fig. 4** diagram, in both approaches, there is a common layer of the verification process which can be re-used and applied to testing functional and non-functional requirements (for a more in-depth breakdown of this process, you can read our case study on the [Dynamic Verification of Input and Output Data Streams for Market Data Aggregation and Quote Dissemination Systems](#)).

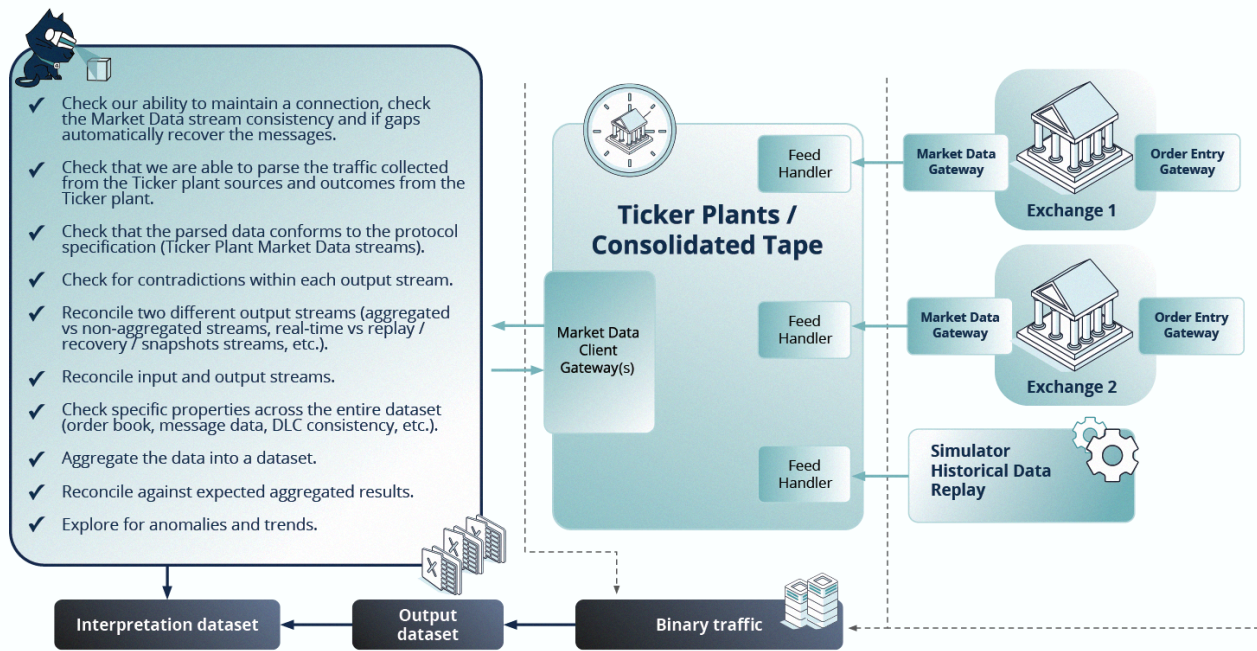


**Fig. 4** Approaches to testing ticker plants / consolidated tapes

The goal of this common layer is validating the Ticker plant's output streams for data consistency. The verifications included here aim to check whether:

- the values of the added quotes are correct (price, quantity, order side, other related calculations, etc.);
- the order change is correctly applied to the order book;
- the book changes accordingly when an order is removed or traded and/or amended to an aggressive trade;
- the order book is constructed correctly in the case of multi-level trades;
- the priority of the constructed orders is correct, the addition of multi-level orders (<10 price levels) is correct;
- the connections to the market data streams are correctly maintained and the data received is consistent; in case of gaps, the lost messages are recovered correctly;
- the traffic collected from the ticker plant sources and outcomes from the ticker plant are parsed without errors;
- the parsed data conforms to the protocol specification (ticker plant market data streams);
- there are any contradictions within any of the output streams;
- upon reconciliation, two different output streams (aggregated vs non-aggregated streams, real-time vs replay / recovery / snapshot streams, etc.) return any inconsistencies, discrepancies or other anomalies;
- the reconciliation of input and output streams produces any inconsistencies;
- specific properties (the order book, message data, the DLC, and others) demonstrate consistency across the full dataset;
- the reconciliation of separate streams against expected aggregated results (once the aggregated data is constructed based on the available rules) returns any inconsistencies;
- other anomalies or trends are observed.

The diagram below highlights the simplified process of binary traffic collection, output data clustering into datasets and interpretation of those according to the specification and common-sense logic.



*Fig. 5 Interpreting the outcome of test actions: traffic analysis*

The “simulator” test approach is indispensable. There is a multitude of situations involving various price positions and their changes as well as a large number of exchanges that disseminate quotes from different platforms. All such situations and combinations are structured in their entirety by using simulator models and, therefore, eliminating the need to spend time on arranging static pre-conditions.

Another way of obtaining relevant market data streams for testing is the generation of a random – or controlled – set of trading combinations that would lead to various changes in quote aggregation and dissemination systems. The same applies when testing non-functional requirements with high-volume data injection or an emulation of trading cycles, including all changes from the production platform. In both scenarios, the outgoing flow is also processed and structured by simulators and can be used in the “simulator” test approach, as described above.

It may seem that full test coverage for ticker plant systems is only possible with the help of simulators. In reality, it is not the case. The main obstacle is the fact that the emulation of a trading platform with 100% accuracy is impossible. In particular, it holds true for the interaction between the Ticker Plant and various markets. The reason for this is that any exchange connectivity protocol specification only contains a limited description of system behavior under various circumstances. Hence, even a simulator built according to the specification with 100% precision and including the smallest nuances will still have the logic implemented “at the discretion” of its developers.

## A Hybrid Approach to Testing

[The analysis of the data collected from different testing projects](#) has demonstrated that both testing methods – the one utilizing exchange simulators and the one utilizing dedicated test environments – have their upsides and downsides. Each can be used independently during appropriate project stages. For example, simulators are especially valuable at the early project stages (when simple functionalities of the system are being developed and tested, and the ticker plant system is not yet ready for the integration with the upstream systems). They also provide unparalleled value for extensive non-functional testing exploration. Load testing activities tend to require powerful simulators and/or historical data replay due to the fact that dedicated upstream test environments may be inferior to simulators in their ability to withstand the required load. Simulators are more reliable when stable performance benchmarking is needed. However, the ability to utilize dedicated test environments is the best option when it comes to integration testing or user acceptance testing.

A hybrid technique has, in fact, demonstrated the best return on investment. The two approaches can be used interchangeably, depending on the project stage or any current circumstances. For instance, temporary maintenance of an upstream system, a blocker bug in it or the absence of access to an upstream system would cause reliance on a simulator.

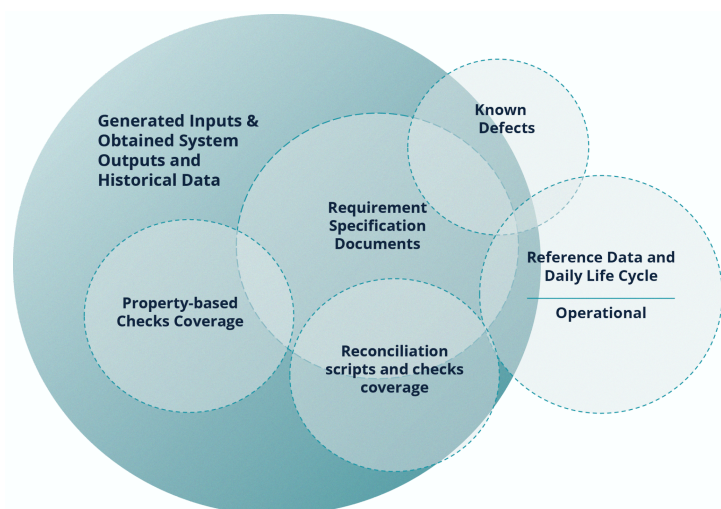
## Advanced Results Analysis to Enhance Ticker Plant Quality Assessment

Considering that the approaches to testing ticker plant systems comprise multiple validation layers, the **assessment of test coverage** becomes an important part of the testing process. The smallest meaningful unit of test coverage in a data-driven practice is a coverage point. A coverage point is an annotated part of code, a field in an individual message, a permutation of particular fields, a sequence of messages, a sequence of actions or a state in the system under test – or its model – that characterises system behaviour. The goal of software testing is to check as many relevant coverage points as possible, with the minimal reasonable hardware and time resources spent.

The resulting test coverage encompasses validations performed at each testing layer through each test approach applied. Each coverage point has a pool of possible acceptable values. Some of them have limited variability, and some have an infinite amount of possible enumerated values. Thus, the overall system test coverage delivered by a given test library cannot be expressed as a single number. It is a **function** derived from coverage measures of each coverage point within the **validation layer**, times the number of validation layers. The input and output of such a function – along with the data collected with test executions – form the **coverage evaluation framework**.

**Fig. 6** schematically illustrates a coverage evaluation framework, highlighting the areas uncovered by validation layers (for instance, test scenarios related to manual market operation interventions happening throughout the DLC are usually not generated as part of the initial input into the systems under test; the missing coverage areas are represented as regions outside the Generated Inputs & Obtained System Outputs and Historical Data circle) and overlapping areas of the validation layers. The overlap is caused by some coverage points being targeted by the multi-layered approach in several types of validations, ensuring deep and comprehensive system exploration.

This approach allows us to explore the market data systems and their source venues more thoroughly and leverage ML capabilities for the generation of advanced computer-assisted checks, their execution, parametric results analysis and reporting. It enables system exploration across multiple strategic validation layers, in addition to the requirements specification documents coverage analysis commonly used in various regulated industries.



**Fig. 6** Test Coverage Evaluation Framework

---

## Recommended Market Data Systems Testing Practices: Conclusion and Main Takeaways

- The question of efficient quality validation for market data / direct-feed / ticker plant / consolidated tape is especially acute in the context of the [EU and UK plans to build a pan-European consolidated tape system](#), and of the [US exchanges](#) and [OPRA proposing](#) to further improve the US NMS consolidated tape distribution.
- Exactpro's independent AI-enabled Testing addresses unique market data system requirements and challenges, such as the need to handle large volumes of highly disparate data transmitted from diverse sources via a variety of protocols, as well as to enrich, customise and restore the aggregated data, where appropriate.
- The two ticker plant testing approaches – testing via simulators or historical data replay tools and testing with dedicated test environments provided by exchanges – can be implemented separately or as a hybrid solution. The hybrid approach ensures the most extensive test coverage.
- In a complex distributed system, it is crucial to understand – and cover with testing – the possibility of hidden interdependencies that will only reveal themselves at the intersection of Functional Testing and Non-functional Testing.
- Following comprehensive data-first testing principles allows for widespread data unification, process automation – including AI-enabled enhancements in the form of test library generation and optimisation – versatile parametric analyses and continuous digital model improvement. Establishing and maintaining a data-first practice ultimately leads to developing an extensive and yet highly resource-efficient test library.
- Leveraging big data and system modelling makes the approach asset-class-agnostic and transferrable across any use cases supported by protocol-based interactions. Exactpro's proposed models and test libraries are the result of years of experience working with some of the world's leading trading, clearing and settlement and market data infrastructures.
- AI Testing can be used to assess the quality of systems that are not yet covered or are only partially covered by tests, as well as the analysis of existing test libraries, to obtain new information about the systems' behaviour and to improve their quality via increasing test library coverage.
- Regardless of the complexity of the systems and integrated components, market data / ticker plant / consolidated tape system providers can benefit from independent AI-enabled assessment of their systems quality to improve their functionality, resilience, and compliance.