Chapter 7
Information Technology

Matthias Kluber

7.1 Introduction

Superior *information technology (IT)* is the essential success factor in any exchange organization worldwide, regardless of the instruments being traded for a panoply of asset classes that can range from shares and bonds to derivatives and commodities.

The days of trading pits with brokers milling around in colorful jackets, taking client orders over the telephone, are history. A modern stock exchange today is first and foremost an IT service provider.

In this brave new world of advanced technologies, the following key characteristics will determine the service quality of a stock exchange. Together, they will drive the design of its underlying IT systems:

- *Reliability*: A stock exchange is a critical component of the macroeconomic infrastructure, comparable to transport systems, communication networks, and energy supply. Every day, millions of investors rely on the *availability* of equity markets, and on the predictable execution of their orders.
- *Transparency*: The exchange should provide complete and timely information to the market about the operational state of its systems and the market's behavior. The relevant information includes the status of the current order book, and of individual member transactions and traded prices.
- *Integrity*: The exchange technology must prevent unpredictable system behavior even in exceptional circumstances, such as uncontrolled process flow by automated trading programs (Mad Machines) of "member installations" or faulty orders (Fat Fingers). Orders that cascade in an uncontrolled way because of these exceptional circumstances may lead to brief bursts of extreme market activity and, in so doing, can trigger a *Flash Crash*.

M. Kluber (✉)
Deutsche Börse Group, Mergenthalerallee 61, Eschborn 65760, Germany
e-mail: Matthias.kluber@deutsche-boerse.com

R. Francioni, R.A. Schwartz (eds.), *Equity Markets in Transition*,
DOI 10.1007/978-3-319-45848-9_7

- *Fairness*: All exchange members are treated equally. In today's markets, exchanges achieve fairness by transparently offering a menu of standardized connectivity options, rather than by having a one-size-fits-all interface with the exchange.
- *Low Latency*: As markets move at ever faster speed, members rely on immediate system response and instant transaction processing. This kind of transaction processing, provided at the highest speed enabled by the latest technology, is particularly relevant for *high-frequency trading (HFT)* and *algorithmic trading*.
- *Predictability*: Members expect consistent system performance irrespective of the system load. In fast market scenarios in particular, systems should operate as usual, i.e., without any delay in transaction processing and market data distribution. Performance may degrade under exceptional volumes in other systems, but the same does not hold for exchange systems because they need to be highly scalable and maintain sufficient headroom to cope with peak loads.
- *Easy Access*: A regulated public exchange should be open for a diverse set of trading members, each with different business models and investment motives. The connectivity options should correspond with various technical requirements and expertise as well as the members' geographical locations. Technical barriers should be minimized for market entry to fulfill this easy access, e.g., with so-called *Zero Footprint*[1] connections.

These key characteristics apply to a broad range of market models and exchange systems. The specific characteristics of each equity market and its membership structure will ultimately determine how these principles are applied by the IT systems. For example, in a traditional floor-trading environment, low latency would signify the prompt display of prices on a screen within a few seconds after a trade is executed. In today's high-performance trading systems, transactions are processed end to end in less than a thousandth of a second. A billionth of a second can matter hugely for some members who are deploying market-sensitive trading strategies and algorithms.

These aforementioned design principles must be manifested in the building blocks of the exchange's technical environment (see Fig. 7.1). We will take a closer look in the corresponding sections.

However, such design principles require substantial capital investment for their implementation, from concept to reality. In the process, they often even compete with each other as we will see in the following sections.

- Core processing is the heart of exchange functionality. This is where order books are maintained and trades are executed by matching orders according to the rules of a market model.
- Transaction interfaces and market data interfaces are both critical for secure and fair member access to the exchange functionality; they keep the architecture efficient and scalable. Standardized *gateways* manage the market members' access to the core processing.

[1] Zero Footprint connections do not require special exchange software or hardware installations and maintenance at the member site.

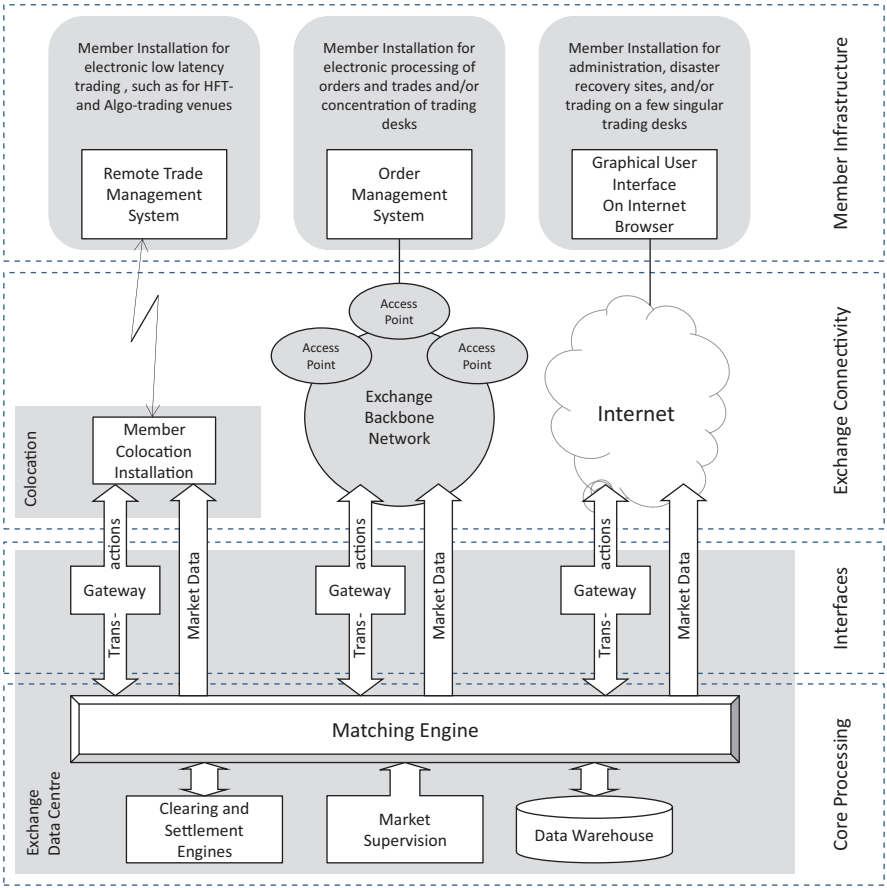

Fig. 7.1 High-level overview of building blocks in an exchange system

- Exchange connectivity is the connectivity options that member firms may deploy; they are compatible with the scale of their operations and trading and investment style. Exchanges typically offer a highly portable access option via the Internet that is suitable for "human traders" at smaller firms, or for use at *disaster recovery* locations. Larger trading desks and client-driven order routing businesses would generally connect via access points in an exchange's *Wide Area Network (WAN)*. Proprietary traders pursuing short-term strategies with high transaction throughput and extremely fast response requirements often opt for a *co-location* site. In a co-location facility, their trading engines that are controlled from a remote trade management installation reside in close physical proximity to an exchange's core processing center.
- Member infrastructure is the technical infrastructure that members have to build and maintain to connect to an exchange.

In the closing section of this chapter, Sect. 7.7, I describe how exchange organizations measure, control, and publish system performance information.

7.2 Core Processing

In the 1990s, the first generation of electronic exchange trading systems specifically designed for high availability made use of specialized *computer operating systems* designed for uninterrupted service with minimal downtime for maintenance. Many exchanges deployed Tandem NonStop[2] and OpenVMS[3], both of which are now part of the Hewlett-Packard Company. Today, state-of-the-art trading systems are typically built on Linux, the Unix-like computer operating system. Linux is "open source," meaning that access is based on a model of collaborative software development[4]. Red Hat or SUSE and other vendors select from the existing Linux modules and hardware drivers to package complete distributions that conform to their customers' needs and the available *server* hardware. Because the Linux software is free, vendors generate revenue mainly by offering software maintenance contracts. They provide support services and will deliver software patches in the case of software bugs, or to offset any incompatibilities between software modules and the hardware.

High-performance trading systems, unlike most general computing environments, are not built upon software *virtualization* layers. It should be noted that these layers would shield the application code from the underlying server hardware and the computing in the *central processing units (CPU)*. This virtualization is very useful for mainstream computing in optimizing hardware utilization, facilitating software development, minimizing maintenance efforts, and, therefore, reducing costs. However, because this adds overhead in the processing, in liquid exchange markets, virtualization is inadequate under the extreme performance requirements for a *matching engine*. High-performance trading systems are generally designed to operate at extreme speed, without delays, even under high load. Consequentially, the capacity specifications are laid out with ample headroom. The utilization of system resources should be on the low end to avoid bottlenecks at peak loads. Not surprisingly, exchanges use high-performance servers with multicore processors; and interconnections between servers have high *bandwidth*, at least 10 Gigabit per second, or more.

The technical setup of the server hardware also has to be optimized: Regular system maintenance activities, from hardware memory checks to fan control, are technically controlled via so-called system management interrupts in the computer operating system. What are interrupts? These will temporarily stop application processing and, in so doing, allocate resources to these maintenance activities. By fine-tuning these interrupts, system performance becomes more predictable.

---

[2] Tandem NonStop was introduced in 1976 and includes a server line as well as the integrated computer operating system NonStop OS.

[3] The computer operating system OpenVMS's predecessor VAX/VMS was released by Digital Equipment Corporation in 1977.

[4] Open-source software is made available with a license in which the copyright holder allows to use and to change the software for free. Open-source software is often developed in a collaborative public manner.

Linux vendors typically assemble two types of software distributions: a general-use distribution that utilizes a stock *kernel* and *real-time* kernels deployed in high-performance environments that depend on extreme processing predictability: In this latter type, the process scheduler within a computer operating system allocates time slots to various processes. As long as the CPU is busy with a process, all other processes have to wait for their next time slot. Real-time kernels force interruptions of the active process on a very granular basis. The result is that all other processes frequently have the chance to become active, and to react to events. Consequently, this makes the reaction time more predictable overall. The downside is the additional overhead attributable to more regular switching between the different processes (Fig. 7.2).

The average processing time might be increased using real-time kernels, but the predictability of the processing time is optimized and the fat tail of the performance distribution is minimized. A fat tail refers to outlier events, e.g., a transaction taking exceptionally long to complete.

The processing of financial instruments with separate order books (e.g., for different shares) can be distributed on separate physical server hardware. Even if the processing occurs on one physical server, it still can be scheduled in parallel by different threads of instructions (one per order book) on multiprocessor systems. Trading systems can therefore be very scalable. The impact of hardware failures or performance issues can be contained within a subset of instruments.

Duplicating key components of the trading architecture is the way to maximize reliability and availability in modern exchange trading systems. Order books can be maintained in two instances, primary and backup, and the transactions processed in parallel on both instances. One component will then be actively used while the other runs in standby mode. This process allows for a seamless failover in case of a

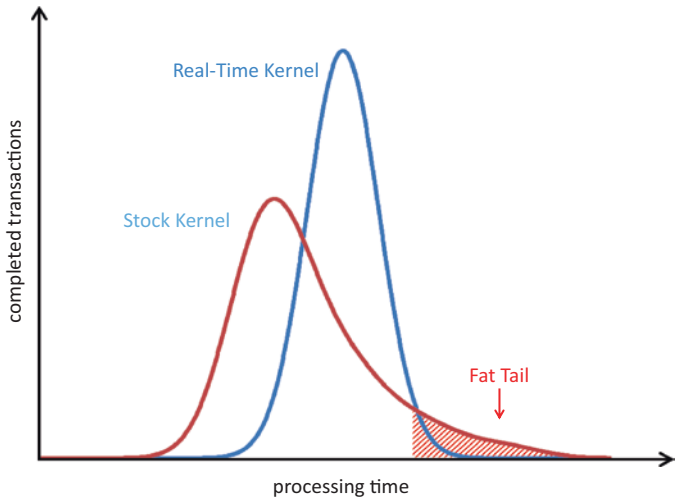

Fig. 7.2 Processing performance with real-time kernel versus stock kernel

defective hardware component. There is an upside to such *scalability* and reliability: High-end trading systems can just about host an unlimited number of instruments and asset classes, and support different market models, multiple market supervision entities, and diverging trading calendars.

However, there is an important limiting factor for the trading system capacity: All transactions for a specific order book[5] must be processed sequentially due to time prioritization of matching events during continuous trading. Therefore, the order book for any instrument must reside on one specific location in the memory. Any changes to the order book can only be consecutively applied, one after another. Distributing the processing of order book updates on more than one processing unit would require that these distributed units lock the memory containing the central order book for the time of the update, i.e., prevent other processing units writing to the memory. This is time consuming, even if measured in microseconds; it also limits the maximum throughput. Hence, state-of-the-art trading systems today concentrate the core matching for one instrument on a single CPU to avoid this extra time and expense. The corresponding order book information should reside in the Level 1 cache, i.e., the fastest memory closest to this CPU. The time to process an order book update by this CPU will then be the overarching limiting factor for a liquid instrument in the entire exchange system.

Pipelining: The concept that balances overall system throughput and the time span of individual order book transactions is called "pipelining." To optimize transaction times, one should ideally take all the steps in an order book transaction with a single CPU, and within the associated Level 1 cache. These steps include preparing the change in the order book—for example, receiving, decoding, and checking the transaction data—and then the update of the order book itself along with certain follow-up steps. The follow-up may consist of generating the relevant market information and the synchronous response to the member, encoding and sending the transaction data. Processing order book transactions end to end in this manner would block the CPU until all steps are completed. It is of interest, therefore, to identify some elementary steps that can be distributed over several CPUs within one matching engine. Breaking up transactions into a series of elementary steps is known as "pipelining." In this way, certain elementary steps can be distributed and executed in parallel by different CPUs, rather than by processing each entire transaction sequentially. More system overhead time is used in distributing these steps, because of the required data transfer between the CPUs. Nevertheless, each individual CPU uses less time than otherwise for the entire individual order book transaction.

Now suppose that the architecture of a high-performance matching engine can arrange an order book transaction into individual steps with a processing time of 15 µs (microseconds, a millionth of a second) for the longest single step (shaded step in Fig. 7.3). This order book can be updated approximately 67,000 times a second. The sum of all individual processing times (i.e., the processing time of the entire order book transaction, including the 5 µs overhead time per step) would be 75 µs. However, the additional system overhead times would be avoided if all steps

[5] The order book of a traded instrument is the list of the interests of buyers and sellers with price and quantity.

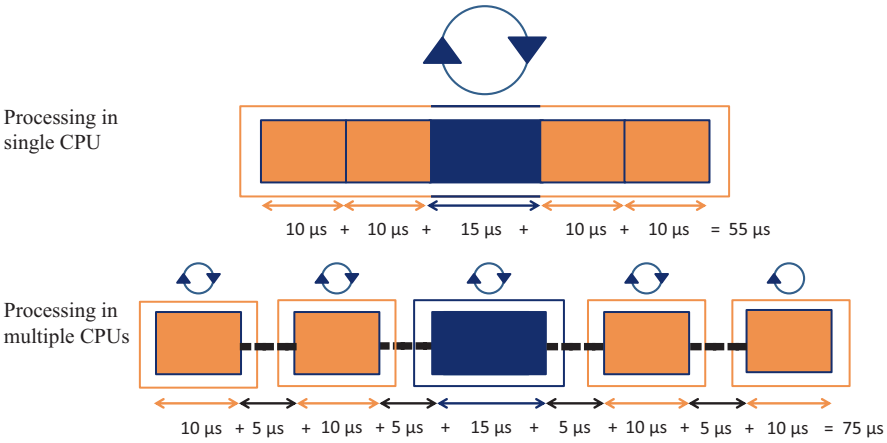

Fig. 7.3 Total processing time per order book transaction

were executed by a single CPU. That, in turn, would reduce the order book update transaction time to 55 µs. Nevertheless, in this case the overall system throughput would then be reduced to 18,000 transactions per second.

This trade-off between elementary processing speed and throughput capacity illustrates how exchange systems must be designed and optimized for specific use cases and market situations. In an exceptionally fast market, it would be conceivable to receive more than 18,000 order book update requests per second, or 18 order book update requests per 1 ms. On a single CPU processing under the assumptions that underlie Fig. 7.3, some transactions would have to be stored in *queues*, waiting for sequential processing, which would negatively impact the performance of the market. Here, an optimization for more throughput seems an appropriate response. If throughput is not as important as the reduction of transaction time, the trading system architecture should steer clear of cutting the transaction into such small pieces. In our example, a single order book update transaction, if processed on a single CPU in the core matching engine, can be accelerated from 75 to 55 µs, end to end.

Queue Handling: Generally, queue handling is a difficult challenge in the design of trading architecture. Typically, the sizing of a high-performance trading system will cater to ample headroom to avoid queuing and other capacity bottlenecks. Trading systems, even in fast market situations, should not slow down at all. However, in exceptional circumstances, the traffic flow might become congested. Sophisticated mechanisms need to allow the system to respond in an elastic manner. A minor system stutter may otherwise become amplified and eventually lead to an overall standstill. *Flow control* models (similar to models used for road traffic simulations) allow the trading system to gracefully slow down temporarily. Any performance degradation or slowdown represents an undesirable state for a trading system. But an escalating capacity overload and eventual standstill of the entire system are even worse. It must be unconditionally avoided.

To this end, customizable transaction limits are kept at the system gateways to prevent members from sending excessive transaction volumes. Transactions can be limited in two ways: The maximum rate of incoming transactions can be specified, or the number of open, not yet completed transactions per member can be limited. In both of these extremely rare cases, members will receive an error message from the gateway if they try to send a transaction that exceeds these limits.

We will now describe other mechanisms that can reduce system latency even further.

Optimistic Response: One of the most time-consuming aspects of transaction processing is writing information to a secure and persistent storage medium. An information update usually is synchronously stored on a storage disk, or other hardware device. To achieve an even higher confidence level, the data may then be copied to a second storage device in a geographically separate location. Once these written instructions are completed and confirmed, a transaction will be finalized and a response sent to the initiator. Finally, once this response is received, an initiator can rest assured that his or her transaction has been completed and safely stored.

To accelerate processing, a trading system can permit members to request an "optimistic response," as soon as the transaction is processed in the CPU, and once the order book is updated in the Level 1 cache memory. With this setup, one will receive a quick response; however, this response may not be reliable. The information in the memory could be irretrievably lost if, for example, there is a hardware problem. Alternatively, the member will receive the response once the order book update has been written reliably on a storage device. However, the storage device may also be lost if there's a larger disaster in the exchange data center. These responses, therefore, can only serve as a preliminary indication of the successful completion of the transaction. The legally binding confirmation of orders and trades will have to follow after the information is copied to the storage device in a second, geographically distinct data center. As an example, synchronous copying of data onto a fast, solid-state disk in a second data center 100 km away will take approximately an additional millisecond.

Transferring messages from one server to another is another source of latency in a trading system. For distributed computing in particular, multiple messages must be sent between clusters of servers. Not surprisingly, the standard communication protocols will add substantial overhead time to the transaction times. Once again, these are overheads measured in a few microseconds. Nonetheless, in a communication cluster with several nodes, these can result in a significant expansion in processing time.

Exchange operators therefore pay special attention to the messaging architecture for the transfer of data between processes. For instance, an incoming transaction related to a specific instrument must be forwarded to the matching engine that hosts this instrument's order book. Market data in turn must be sent out to the various member interfaces. The messaging architecture can either be customized for the exchange, or a low-latency vendor solution could be adopted. For trading systems with high throughput, it is essential to avoid a configuration with a central dispatch function that distributes incoming transactions to their target matching engines. A central broker in this approach would once more create a bottleneck. A distributed

messaging middleware using *IP Multicast* technology and *remote direct memory access (RDMA)* leads to higher scalability and resilience. Transmission overhead can be significantly reduced by deploying RDMA. Moreover, a process in one server can write data directly into the memory of another server without involving their operating systems. This leads to high-throughput and low-latency networking, which is especially useful in massively parallel computer clusters.

Here's an example of the effective use of RDMA technology: The *InfiniBand* architecture of interconnecting computers with high-speed links and low latency to transmit data between each other via IP Multicast protocol.

Tuning these high-end trading systems for the highest possible performance and throughput described above effectively minimizes execution risk for exchange members. Nevertheless, further safeguards are required to prevent unintentional market movements such as Flash Crashes. In today's breathtaking speed of computer-based trading, human supervision of the market can be far too slow to control sudden and challenging market developments. The rare but much publicized Flash Crashes highlight how massive losses in market capitalization can occur within a blink of an eye. There are numerous possible causes for Flash Crashes: a programming error in an algorithmic trading engine (the "mad machine" phenomena), or an erroneous (fat finger) order entry by a *screen-based trader, to name* two.

Several safeguards for these risks are described in the following section:

- Transaction limits
- Functional checks
- Member-triggered emergency exits
- Function of *volatility interruptions*

If the number of transactions from an individual member exceeds predefined limits, a first line of defense against mad machines and fat fingers is the ability of gateways to reject transactions from this member, or even to disconnect the member's session. A second line is functional checks and predefined thresholds in the system. If a trader who is only authorized to buy or to sell up to a certain value accidentally confuses quantity and price, he might just not be able to send an order.

Sophisticated trading systems support the configuration of detailed authorization schemes, including risk and exposure limits for individual groups and specific traders. If certain limits are exceeded, these will first provide warnings, and then slow down or stop a member. Trading systems may also allow members to introduce price reasonability checks so that the entered price does not significantly differ from the price on the market.

Members must have control over their market exposure, particularly in exceptional situations. The emergency exits they need include stop buttons for clearing members (which will cut off some or all of the traders under their sponsorship), market maker protection parameters, and the automatic cancellation of orders if the technical connection to the exchange be interrupted.

The most important safeguard against Flash Crashes potentially is volatility interruptions: Here's how a volatility interruption works: If a market in an instrument moves so quickly that its price shifts outside a predefined range, a trading

system can automatically halt the continuous trading. The market supervision team then has time to initiate an auction, so that members may review their positions and adjust pending orders before continuous trading in the instrument resumes.

7.3 Transaction Interfaces

Exchanges generally provide members with a range of different transaction interfaces. *Front-end trading applications*: Via these interfaces, members can submit transactions to the exchange trading engine, gateways being the standard entry points for the transactions. They also serve as a *firewall that* shields core processing from direct connections to the members.

Exchanges use various concepts for gateways. If a single gateway per traded instrument is configured (and all transactions from all members are directed to this sole gateway), the exchange can easily serialize all transactions for an instrument. In this way, it can implement a first-in, first-out service. For high availability, this single logical gateway would typically be implemented in a redundant hardware setup. However, if the exchange wants to support many members and instruments in a high-performance environment, connectivity via a single logical gateway does not scale well (Fig. 7.4).

When gateways serve many instruments, exchanges can evenly distribute member connections across the gateways as shown in Fig. 7.5. The gateways intermediate member connectivity, and relieve the core matching engine from supporting many individual member connections. That is because many members connect to a single gateway, and the gateways in turn connect to the matching engines.

Even with standard gateway hardware, the concept in Fig. 7.5 may still lead to slight variations in the gateways' and associated network links' performance. Time priority is assigned to orders only when they reach the matching engine. Therefore, extreme latency-sensitive members will always try to identify the "fastest" gateway at any point in time, i.e., the gateway through which they can reach the matching engine first. These members may choose to establish sessions on all of these gateways and then send their transactions in parallel, or to use their own methods to identify the fastest gateways, such as analyzing technical performance data.

Minimizing the impact of technical requirements by the exchange on member infrastructure and architecture is one general design principle of interfaces. Traditionally, exchanges have required members to install and maintain specialized exchange hardware or software (or both) on their premises. Today, members no longer need to install exchange hardware or software to connect to the exchange's *back end*. This is called "Zero Footprint Access." Modern trading architectures can be accessed without the need for specific hardware, operating systems, programming language, and compiler versions. That's as long as they support the general communication components, like *TCP/IP*.

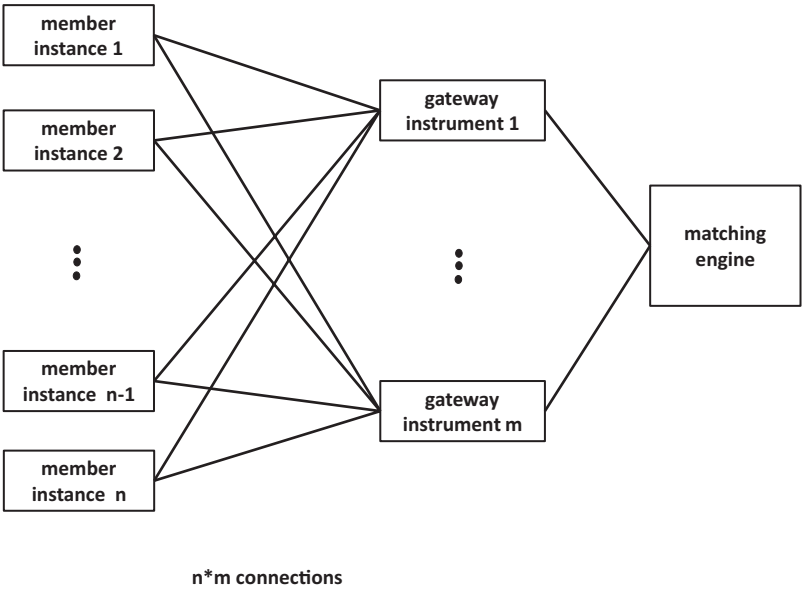

Fig. 7.4 Single dedicated gateway per traded instrument

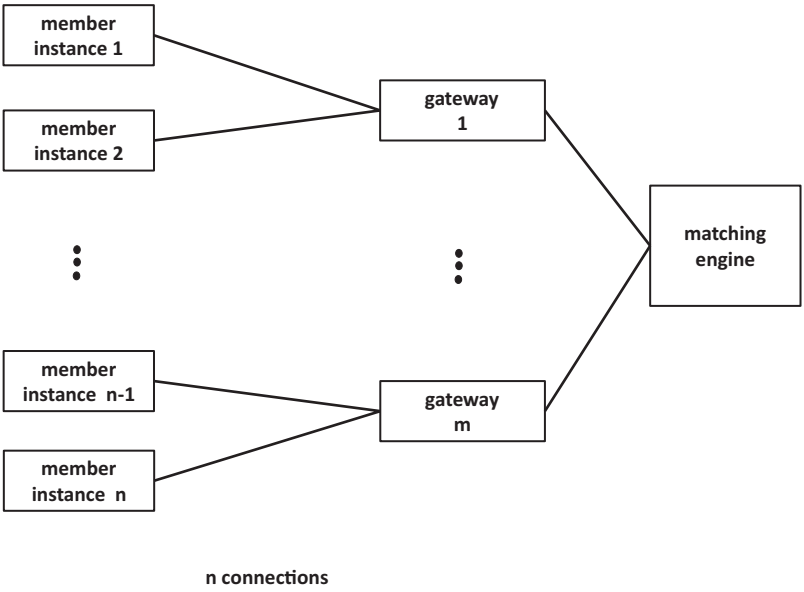

Fig. 7.5 Member connections to dedicated gateways serving multiple instruments

Transaction interfaces on exchange systems are typically asynchronous, message based, and session oriented. Members order their sessions from the exchange. Member software applications are connected to the trading system by opening a TCP/IP connection to an exchange gateway.

There are different design approaches for these interfaces and the corresponding gateways: Exchange proprietary interfaces allow high-performance access and full trading functionality. They support market making/quoting and additional services such as trading support information, or member-specific risk control messages. These proprietary interfaces are for members who require the highest throughput and the lowest latency. Messages exchanged between the member and the exchange across proprietary interfaces are, nonetheless, very similar to the standardized layout and content definitions of the *Financial Information eXchange (FIX)* protocol. The FIX protocol is optimized for traditional buy-side investors rather than for proprietary traders or market makers. Therefore, messaging efficiency can often be enhanced by small deviations from the FIX standard. These customizations may include a proprietary session layer with modified message headers, trailers, or additional user-defined fields and messages. The result is improved efficiency and performance that allows functional gaps in the protocol to be filled.

Exchanges may further support various session types within their proprietary interface specifications, for example:

- High-frequency sessions
- Low-frequency sessions
- FIX sessions

These session types can differ by their throughput limits and functionality. The pricing of these session types may reflect the way a member makes use of the exchange's technical infrastructure.

Members might submit a large quantity of order messages and other transactions to the trading system, resulting in a relatively small number of trades. The ratio of system transactions to trades will often exceed 100. Indeed, an exchange may also charge members for the number of transactions they are allowed to submit on this premise: The required capacity and the cost of the trading system depend more on the message volumes and less on the number of actual trades executed.

The high-frequency sessions offered by some exchanges are intended for market makers and HFT firms. These sessions accept higher transaction rates and allow members to make more intensive use of the exchange infrastructure. To minimize latency, the corresponding high-frequency gateways will, for example, accept only non-persisting orders, i.e., orders that are only kept in the Level 1 cache memory and not synchronously written to a storage disk. Data replication and recovery of trade events are restricted. The hardware of these gateways consists of powerful, dedicated, stand-alone servers that support special features like real-time kernel (see Sect. 7.2), kernel bypass, and *field-programmable gate arrays (FPGA)* for optimized latency and minimized variance.

A kernel bypass (a mechanism on network interface cards) allows data packages to be transferred straight to the application without being buffered in the operating system's memory. FPGAs allow configuring and optimizing microchips for very specific use.

Low-frequency sessions allow more functionality but, at the same time, they also restrict the number of transactions a member can submit. In addition, some exchanges offer special back-office sessions that serve only a subset of the low-frequency session functionality (mainly trade confirmations). The server hardware for the corresponding gateways will be less rigorously optimized for latency and performance.

Exchanges may also offer access via FIX gateways as an alternative to proprietary transaction interfaces. Members may prefer a FIX connection in order to standardize their connections to various exchanges. This is a point-to-point service based on the technology and industry standards of TCP/IP, FIX, and the FIX session protocol. The FIX protocol is not as flexible and efficient as an exchange proprietary interface, and it may limit performance and functionality. For example, a standard FIX session will not support the full scope of functionality for market making and quote submission that most exchanges offer. The exchange might offer two kinds of FIX sessions depending on the intended use of the FIX interface: (1) for order management and (2) back-office FIX sessions for the receipt of detailed trade confirmations organized by member business units.

7.4 Market Data Interfaces

There is a fundamental component for a fair and reliable market: An exchange system must provide order book and trade information as rapidly and transparently as practical to members. Order book information will be made available up to a specified depth based on the member's requirements. The order book data may either be refreshed upon each single order book change, or else be sent via a consolidated update that transmits all order book changes within a certain time interval. The consolidated update method can save bandwidth and be used for highly liquid order books.

Most exchanges use IP Multicast to broadcast market data given that all members should receive the same data simultaneously. IP Multicast is a method of sending data packages to a group of intended recipients in a single transmission. These packages are automatically copied within the network and distributed to several destinations based upon a receiver's subscriptions, in contrast to the TCP/IP protocol for individual transmissions between one sender and one receiver.

In trading systems, members subscribe to the market data streams for certain groups of instruments. However, IP Multicast packages are not necessarily delivered in sequence and lost packages are not automatically resent. That's because they are transmitted via the unreliable *User Datagram Protocol (UDP)*. IP Multicast transmission may generally work predictably and without interruption, but there is no flow control mechanism that guarantees delivery of a package. In fact, the receiv-

ing system at the member site will have to observe the sequence number provided by the exchange system and identify potential gaps, or correct the sequence of the incoming data stream. An exchange system, seeking to cope with the potential loss of IP Multicast packages, will typically disseminate its market data from the matching engine via two distinct IP Multicast streams over two separate network connections. A member system will then listen to both streams, and forward the IP Multicast package which it receives, first for further processing. In this way, it can fill potential gaps in one stream with data received via the other stream.

Market data streams have a highly volatile volume structure. A fast market environment can lead to a self-amplifying effect[6], creating sudden bursts of market data. In liquid instruments, these bursts can happen within a fraction of a second they are called "*microbursts*." The size of a microburst is limited only by the overall processing and delivery capacity of the trading system. This capacity limit can be fairly high with many instruments traded in parallel on distributed systems. But when markets move swiftly, a member doesn't want these high volumes of market data being queued and delayed in their transmission. Trading decisions might otherwise be based on outdated information.

Here are two possible solutions to avoid, or to minimize delays:

1. For very latency-sensitive members—HFT companies and certain algorithmic traders depending on their strategy—a network infrastructure with ample headroom capacity can be used to avoid queuing even during a microburst. The average data transmission rates may be a few Megabits per second. Some members, however, install network connections of 10 Gigabits per second; or even 40 Gigabits per second, i.e., more than a thousand times the average throughput.
2. For many other business models (like screen-based trading), this excessive volume of market data cannot be reasonably processed. Exchange systems therefore offer a "netted" or "pulsed" market data stream. In this stream, order book updates and trades are summarized within a certain time interval. Only the status at the end of the interval is distributed. Sophisticated trading systems permit exchange operators to specify the netting interval separately by traded instrument, or even dynamically depending on the overall volume. The maximum throughput requirements can be better controlled with such netted market data streams. They will not exceed a pre-calculated limit.

In managing bandwidth, IP Multicast has this advantage: Members may individually select certain streams that are essential for their business for subscription. A stream contains the information pertinent to a group of traded instruments. Data transmission via IP Multicast is not 100 % reliable, so exchange systems let members request missing data packages. Alternatively, the system will publish snapshot messages on dedicated streams so that members can reconstruct the order book in the event of gaps in the data received via the normal streams.

Sometimes it is difficult to distinguish between an inactive market and a connectivity issue. This can be the case when members listen to and receive no data in the market data

[6] One order may trigger a cascade of subsequent reactions from other market participants.

stream of a less liquid product. Therefore, exchanges tend to send out "heartbeat messages" on a market data stream. If members receive nothing but the heartbeat, this signals that the market is quiet at that moment. If no heartbeat message is received, a technical alert is raised. It is then obvious that there is a technical issue with the connection, either on the exchange's side or within the member's infrastructure.

7.5 Exchange Connectivity

Exchange organizations need to attract market participants and order flow on a global scale to provide liquidity. Easy, secure, and reliable access for members is fundamental to the exchange business model. A wide range of trading strategies, often requiring different connectivity requirements, may be pursued by members. In response, exchange organizations generally offer a wide range of options for connecting to the exchange system.

The most rudimentary (but sometimes fully sufficient) connection is via the public Internet. Most member firms, however, need a higher level of reliability and guaranteed performance levels. Hence, exchanges often offer connectivity via a dedicated private Wide Area Network, or WAN. Then there are the requirements of technology-driven and latency-sensitive members, HFT traders and many algorithmic traders included. To satisfy this group, exchanges typically also provide co-location facilities as an additional connectivity option. These latency-sensitive trading firms, often connected to multiple exchanges, are willing to pay a significant premium for the fastest connections. Communication technologies, such as *microwave* transmission, are in use in this speed-vital environment.

Connectivity Options: A standard cost-effective way to accomplish direct and simple connectivity is by connecting the member's *front office* to the exchange system via the public Internet. A high level of security can be achieved when using appropriate encryption mechanisms despite the inherently unpredictable nature of Internet data transmission. For small trading firms it is a simpler matter: They may just need a *virtual private network (VPN)* Internet connection and a few standard desktop computers with an Internet browser to easily access multiple exchange systems via the *graphical user interfaces (GUIs).* The GUIs are provided by the exchanges. A VPN is a point-to-point Internet connection through an encrypted data transmission tunnel. It prevents unauthorized third parties from accessing or manipulating data transferred over the Internet.

The reliability and performance of Internet connections cannot be guaranteed because exchange organizations have no control over the Internet infrastructure. These features, however, are critical for the majority of the members.

Business models depend on fast, reliable access to market data provided by the trading system. Hence, exchanges also offer access via dedicated private WANs. These are strictly separated from traffic carried for third parties and protected against unauthorized access. Some exchange organizations offer access

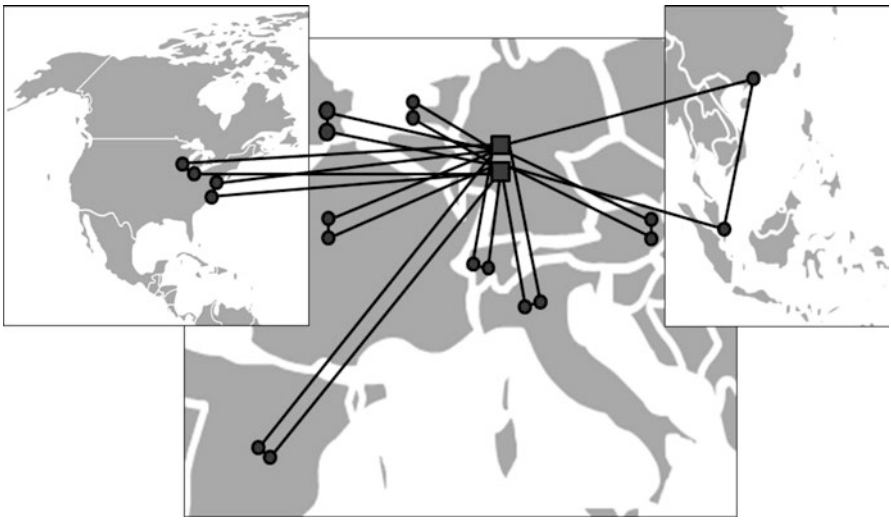

Fig. 7.6 Deutsche Börse's Wide Area Network N7

via specific commercial *extranet* providers; others operate their own global WAN to preserve full end-to-end control between the member installations and the exchange infrastructure.

WANs to connect members with the markets operated by a single-exchange organization are generally built in a star topology. Multiple connectivity centers in different countries and continents—also called points of presence, or access points—are directly linked to the trading system at the center of the star, via the shortest possible path (Fig. 7.6).

Members connect to their closest access point via private telecommunication links, provided by either the member or the exchange organization. Some exchanges ask their members to connect to their connectivity centers, and others provide end-to-end connectivity with options for *redundancy* and bandwidth.

Exchanges develop their trading systems and network infrastructure for full redundancy since reliability is of the utmost importance. The effort and investment in backup infrastructure are substantial. A trading system is typically duplicated, choosing from two options: (1) Both parts are actively used and load balanced over two data centers; for example, the matching engines for one half of the traded instruments are hosted in one data center, and the others are in the other data center. (2) Alternatively, the active primary and the passive backup systems are located in two distinct data centers. Critical data are copied synchronously between the two data centers.

In the event of a large-scale fault in one data center, the installation in the second center will need to take control. For this purpose, exchanges usually select two geographically distinct data center locations to avoid a simultaneous outage in both of them. The cause, for example, could be a regional power interruption, an earthquake, or an extreme weather condition.

Aside from the trading system itself, the network infrastructure and the access points need to be implemented in a fully redundant manner. Access points, similar to data centers, are also installed in pairs of two, and they are interlinked to provide a seamless failover. Each pair of access points is connected via two *backbone* links with the two data centers hosting the trading system. Exchange organizations minimize the risk of a simultaneous outage of both backbone connections. This is accomplished by using different network providers with the highest service level each, whose routes are guaranteed to be physically separate from each other. Sufficient analysis is necessary because seemingly diverse routes can easily turn out to use the same underlying infrastructure, e.g., the same sea cable. A single outage on this infrastructure might then interrupt both supposedly diverse connections. Consequently, a member firm, even a sprawling regional financial community, may be disconnected from the exchange system. It is not so unusual, for example, for the anchor of a fishing boat at sea to cause damage to a major underwater cable[7]. To make things even more problematic, network routes are dynamically altered by the telecommunication providers.

Let me explain: Two routes that have been on separate paths in the past may suddenly share certain underlying infrastructure components after an automatic switch. Hence, the carrier network optimization mechanisms and the routing of individual cables must be verified right down to street level. This will avoid, for example, single points of failure, and unnecessarily long routes. Exchanges monitor network connectivity 24 h per day, enabling them to restore services promptly after a disruption, and to minimize the risk of a complete disconnect or breakdown. In the best-case scenario, this happens before a member would even notice any service degradation.

To offset costs, some exchanges build their own WAN not in a star topology but rather in a ring topology as depicted on the right-hand side of Fig. 7.7. A ring requires fewer backbone connections and less hardware.

In this topology, multiple exchange system locations can be interconnected via a single loop. Every exchange location acts as a connectivity center for all other locations. The members connect to the closest exchange installation. This topology incorporates a natural redundancy because information can flow in both directions around the ring, and a further duplication of links is not required.

However, because an outage of two or more links would impact multiple locations, this topology provides a lower level of redundancy than the star topology. Moreover, network latency in a ring is typically higher than in the star topology. That's because the connection path from a member to the desired exchange installation is on average longer than in the star design.

Algorithmic traders and HFTs create their trading strategies from exchange market data streams, so for them extremely fast processing of market data and equally fast transmission of their order flow are crucial. In fact, low latency is an essential prerequisite for most of these members. Moreover, they must be able to analyze a market situation and react instantly.

[7] Specifically in 2008, a series of sea cable disruptions impacted the data traffic between Europe and the Middle East and Asia.

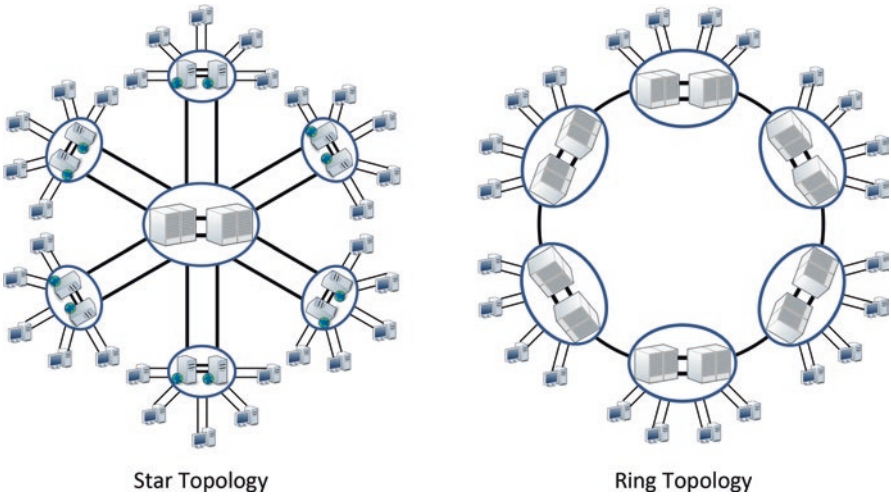

Fig. 7.7 Star network and ring network topology

This is not a new principle. Fast access to market information has always been a key success factor for traders in securities and commodities markets. In the past, timing for traders was a matter of days, hours, minutes, or seconds; nowadays, the time and speed requirements have accelerated, reaching nanosecond turnaround, or a billionth of a second. And so, even the smallest time delay by data transfer from one location to another should be minimized. For the smallest elementary data package, or a bit, it takes 5 μs to travel 1 km through a fiber cable. These 5 μs can make a world of difference; ultimately, this can determine the success or failure of a trading strategy.

Many exchange organizations, seeking to achieve the lowest possible latency for speed-sensitive traders, offer co-location facilities. Under this arrangement, members may install their hardware in exactly the same data center that hosts the exchange back end. Members may then connect locally via so-called cross-connect cables. By minimizing the cable length, it is possible to reduce latency to an absolute minimum, the latency between the member infrastructure and the trading system. Indeed, co-located installations may encounter order round-trip times of approximately a 100 μs—just by cutting out the otherwise inevitable delay from long-distance data transmission.

Some exchanges may also take this step to minimize network hops[8] for these latency-sensitive traders, implementing special high-frequency gateways (see Sect. 7.3) and a dedicated low-latency switching infrastructure. Low-latency switches will use the cut-through technique, which is a method of packet switching. The switch will begin forwarding a packet as soon as the destination address is processed. This method avoids the usual store and forward processing. There is a drawback—relying on the destination devices for error handling.

[8] A network hop represents a networking device on the path between sender and receiver.

This connection concept is highly relevant for certain HFT and algorithmic strategies because co-location installations will always connect faster to the exchange system than any other installation outside co-location. These advantages have lured a large and diverse trading community around specific co-location centers. At Deutsche Börse's co-location center, for instance, more than 150 members are present, including Hudson River Trading, Jump Trading, and Optiver.

As a way to ensure defined service levels between members located in different rooms of the co-location data center, some exchanges use a standardized cable length between the member installation and the exchange infrastructure; others will charge their members contingent on their speed advantage.

In order to limit the impact of a potential data center outage, exchanges play defense, generally preferring to distribute their back-end systems over two redundant data centers. Then there is data transfer and data replication between these data centers. Because it causes additional latency, the trading system infrastructure may be centralized on a single data center campus. Nonetheless, to guarantee the highest possible reliability, a trading system infrastructure would typically be distributed over two separate rooms in the data center. Separate air conditioning and power infrastructure are the ideal arrangement. At the same time, a secondary system must be maintained in a separate, geographically distinct data center to respond to the risk of a complete outage. Data are copied (asynchronously or synchronously) to the secondary data center to allow a market restart after a primary data center failure.

Many latency-sensitive algorithmic traders and HFT firms trade on multiple venues in far-flung global financial centers from New York and Chicago in the USA to London and Frankfurt in Europe—and beyond. Trading strategies on one venue in one city will depend on market data from another venue in another city. With such strategies, speed of data transmission between the market locations has the highest priority. Several competing members will want to be the first to hit an order book.

These market participants are willing to invest in communication infrastructure that allows faster data transmission than the standard telecommunication links between financial centers. They routinely look for ever faster connections between the market back-end locations. A brisk competition for the lowest possible latency has emerged[9]. That has led to some very expensive connection concepts that may deliver speed advantages in the microseconds.

Transmission technologies such as long-distance microwave communication, millimeter waves, and laser links[10] are up to 50 % faster than ordinary fiber cable connections. These speed advantages are directly connected with the physics of light propagation.

[9] Some years ago, telecommunication providers started to deliberately construct short cable connections in nearly straight lines of sight between financial centers. That is despite costs being much higher than they would be for standard routing along existing rail lines or highways.

[10] Wireless connectivity options provide faster alternatives in contrast to cable-based connectivity options.

Consider this: While information transmitted via microwaves achieves nearly the speed of light in a vacuum, i.e., 300,000 km per second, data transmission speeds in fiber cable do not exceed 200,000 km per second. Microwave connections are in use between the major market locations in New York and Chicago and between London and Frankfurt. The round-trip time of a microwave connection between London and Frankfurt could, theoretically, be about 2 ms shorter than that of a fiber connection. There are also microwave connections to the landing points of transatlantic cables; however, the idea of installing a series of levitating microwave antennas over the Atlantic still remains science fiction today.

There is a constraint in microwave transmission: It requires straight line-of-sight propagation, and so it relies on a tightly spaced sequence of antennas between sender and receiver. Because the signal weakens rapidly with distance, it needs to be amplified every 50–60 km. Microwave transmissions are also affected by weather conditions and are, therefore, less reliable. The data transfer rates of approximately 150 Megabit per second are also much smaller than in a fiber cable.

Full market data cannot be transferred easily, so members have to diligently filter the most relevant information for transmission. Smaller wavelength, such as millimeter waves, is necessary to increase the bandwidth. Millimeter waves achieve transfer rates of up to 2 Gigabit per second. Unfortunately, millimeter waves must be amplified every 10–15 km because they are even more vulnerable to weather conditions.

The next step to further improve the signal strength and bandwidth would be the data transfer via laser. Test deployment of this is already happening at some highly specialized technology companies.

7.6 Member Infrastructure

Exchange members need to implement a technical infrastructure to connect to the central exchange systems. These infrastructure at member sites vary significantly. They are heavily dependent on members' business models and trading strategies, as well as on their potential customers' requirements.

In the past, many exchanges required that members install special dedicated devices for the particular exchange on their premises (for example to run servers with special software provided by the exchange). The maintenance of these devices would be either the member's responsibility with guidance by the exchange or the exchange would remotely manage the device from their operating centers. A member who connects to several exchanges would have to host and potentially manage a diversified environment of bespoken devices.

State-of-the-art exchange systems nowadays apply the Zero Footprint approach. It is no longer necessary to maintain exchange software at the member site, since the protocols and interfaces to connect to the exchange systems are open and standardized. Instead, members can freely choose suitable hardware

and computer operating systems and install their preferred front-end software. In doing so, this may connect to all of the exchange markets that are required by their trading strategy.

Extremely latency-sensitive and technology-savvy members will invest significant effort into creating and optimizing what this software will run on, specifically, the front-end software and hardware platform. These members will typically co-locate their installations at the exchange data centers and, in some cases, they might even deploy specifically designed hardware components such as FPGAs, or self-developed network switches. Others may use third-party software which act as a concentrator for connections to several exchanges. Not surprisingly, there is a most dynamic market for exchange connectivity and order management software.

Front-office software for trading, either custom developed or off the shelf, will receive and display market data with numerous customizable views. Traders can enter, modify, or delete orders for different markets and instrument classes, including basket trades. The front-office software then routes these order messages to the appropriate exchange interface.

Traditional order routing systems forward orders automatically to a predefined exchange. Today's smart order routing mechanisms will flexibly choose to internalize orders, or distribute them between the venues, or forward them to the venue with the best execution capability. Front-office systems increasingly include capabilities for real-time analytics. That allows members to track a trader's performance visually, to set risk limits, and to perform further complex analysis.

Big data, a manifest trend in IT in general, is of particular interest to some short-term investors. The correct investment conclusions from a vast amount of input data can create successful business models in proprietary trading driven by technical market signals. Such algorithmic trading is generally supported by complex, high-performance front-office software. Some vendors provide modular building blocks; in other words, a firm may configure, customize, and run their own algorithms without requiring any special software development skills.

Members active on a variety of market venues will have to diligently design the architecture of the front-office infrastructure. The goal here is huge: The infrastructure should be capable of moving massive amounts of data across the globe, supporting a 24-h trading desk in a follow-the-sun rotation. Regarding performance and latency, it will be critical to select the right geographical location for these front-end installations.

As an alternative, or as a supplement for front-office software, some exchanges also offer their own native front-end GUI. This exchange GUI provides some market data views as well as trading and administrative functions. Workstations that run the native exchange GUI could be connected to the exchange's trading system via the Internet. Yet, they can deliver remarkably good performance by deploying efficient data protocols and transmitting only the stripped raw data.

Alternatively, GUI access can also be implemented over the exchange's private WAN. The exchange GUI solution, as with the other interfaces between the exchange and member, may no longer require that members maintain exchange software at their sites. In some configurations, it is relatively simple: a member only needs a standard desktop computer with an Internet browser and a *Java Runtime Environment (JRE)* to

run the GUI. But few members choose an exchange GUI as their preferred solution for actual screen-based trading. Instead, they may prefer to use a software solution with multi-exchange capability depending on the number of exchanges they trade on.

However, the vast majority of members do use the native GUI for other reasons: An exchange GUI may be a sensible choice for a few terminals in a disaster recovery installation, or for an on-site backup. It can also serve as a reference to cross-check the data displayed by the front-office software otherwise used.

Then there is risk management, an increasingly important core component in any front-office system. Agency trading firms as well as proprietary traders will need to control their risk exposure both pre- and post-execution. Therefore, the front-office software will typically connect to a real-time risk management system.

Risk can quickly accumulate and exceed given limits, unless an actual exposure is tightly monitored by an agency trading firm for each downstream client, or by a proprietary trading firm for their own traders. Built-in system safeguards must take immediate action when this occurs. A notable example: The disruption in equity trading caused by a glitch at Knight Capital Group on August 2012 temporarily destabilized trading in nearly 150 New York Stock Exchange-listed stocks. Knight had inadvertently deployed testing software, and consequently suffered a trading loss of US$440 million in less than an hour. This was a reminder that well-designed risk management safeguards are essential.

Robust risk management also requires certain *post-trade* functionality. Once a trade has been executed, it will be reported by the clearing house, either to the member's *back office* or to the designated clearing firm. In the latter case, it is presumed that the member has a clearing arrangement with a partner. *Middle-office* and back-office support is typically provided by one of a few market-leading software solutions.

This post-trade functionality is basically straightforward, yet very critical. Post-trade facilities maintain and manage aggregated positions, and provide the tools to assess underlying exposures for an individual book or across multiple instruments. Moreover, post-trade facilities analyze positions per trader, or for a trading desk, even for an entire firm.

7.7 Time Management and Performance Monitoring

Exchange organizations must provide full transparency for each single transaction due to their special economic significance and major financial impact. In fact, many members expect an exchange system to provide information about the exact point in time a message hits the exchange. More precisely, they expect to know when it enters the exchange gateway, and which subsequent chain of events will be triggered and when.

The best way to provide this kind of transparency is to *time stamp* every message at each step of its path through the electronic trading system. Naturally, this only makes sense if the clocks of the different devices in the system have exactly the

same time and run at the same speed. Not surprisingly, the clocks need to be synchronized very frequently to ensure that this is so.

In general, this is accomplished by using a network protocol, such as the traditional *Network Time Protocol (NTP)* for clock synchronization between devices in a computer network. Time, as provided by a reference clock, is being propagated throughout the network. An accuracy rate of approximately 1 ms that can be accomplished with NTP is not necessarily sufficient for low-latency trading systems.

Exchange organizations must be able to handle fast-moving markets. Time resolution in the sub-microsecond regime is required. Moreover, the electronic exchange system itself is a highly complex system, so a synchronized time signal throughout this system all the way down to member installations is desirable. These requirements can only be met with a more sophisticated time management protocol.

To that end, the *Precision Time Protocol (PTP)* which is typically used is able to handle hundreds of servers, achieving a much higher level of accuracy than the standard NTP. Exchanges deploy specific hardware timing components to achieve extremely high accuracy within the exchange infrastructure, and the member co-location installations. A single, highly precise reference clock is the sole source for time synchronization. This clock will typically use the *global positioning system (GPS)* signal; it provides accuracy to a fraction of a microsecond.

Still, because exchange infrastructure is highly critical, one may not want to depend exclusively on the GPS. A standard radio time signal could be used as well. The radio time signal would serve as a reference and backup, in case the GPS signal is lost or may have been manipulated.

Time protocols measure the delay caused by information transfer between devices. They are, therefore, able to propagate the appropriate time within the network. The transfer time is calculated by averaging the forward and the return time (Fig. 7.8).

The calculation of the signal delay

$$\Delta t = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}$$

assumes that the forward and the return times are equal. In practice, this is often not the case. One major problem: potential queuing in the timing devices during high workload. Such queues cause delays and differences in the transfer times, a

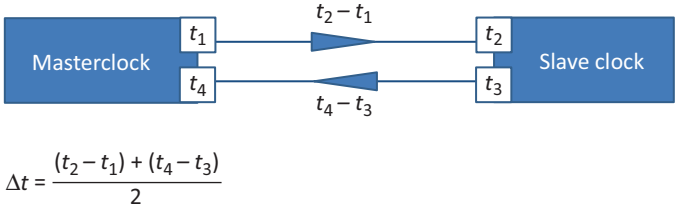

Fig. 7.8 Averaging delay when synchronizing time in a geographically dispersed installation

problem that can be solved by hardware time stamping. The hardware does the time stamping as soon and as fast as possible after the arrival of a message, and as late and fast as possible before the message leaves, a process which avoids software queues.

Devices in the exchange environment regularly receive a precise time. Still, their own systems' internal clocks may still have a slight drift, i.e., a bit too slow or bit too fast. When the device receives the next precise time, it will have to adjust its own system clock accordingly. This adjustment can be done in two different ways:

- By abruptly jumping to the right time, which adjusts the clock instantaneously, but may cause a shift in the chronologic order of the specific device.
- In the form of a smooth and gradual convergence. That means that it takes more time to adjust the clock but the chronologic order of the specific device is conserved.

The second approach to synchronization is preferred because, for exchange organizations, chronological order is highly important.

Exchanges and their members can assign precise time stamps to messages at crucial processing steps based on very accurate time synchronization. Hence, time stamps on these servers can be used to analyze one-way transport times. Figure 7.9 diagrams a typical example of a member sending an order request message, and being answered by a private order response message and a public order book update.

Figure 7.9 can be interpreted as follows:

- The time stamp t_1 can be taken by the member application when the request is sent.
- t_3 is taken by the exchange gateway when the request is read on the member's side of the gateway.
- t_5 is taken by the exchange matching engine when the request is read there.
- t_7 is measured at the time when the matching engine maintains the order books.
- t_6 is taken by the exchange matching engine when the response is sent from the matching engine to the gateway.
- t_4 is taken by the exchange gateway when the response is sent from the gateway to the member.
- t_2 can then be taken by the member application when the response is received.
- t_8 is taken by the market data interface, before the information is sent to the member.
- t_9 can again be taken by the member application when the respective market data arrives.

Only time differences like $(t_2 - t_1)$ can be analyzed in case of non-synchronized times. That is because discrepancies in absolute clock times are eliminated by taking the difference.

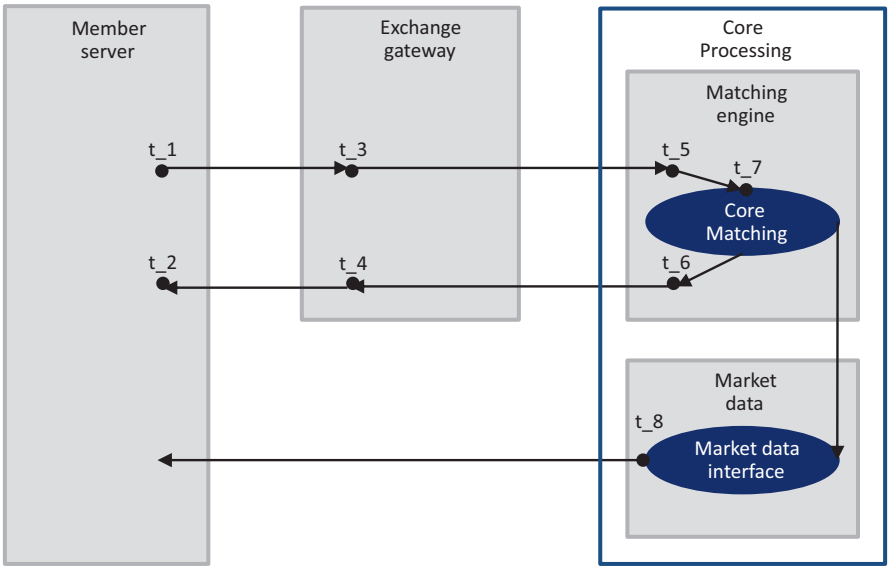

Fig. 7.9 Time stamps in the order processing event sequence

Exchanges may use the above as well as additional time stamps to offer advanced trade traceability to their members. When match occurs in the exchange order book, member order request messages trigger further conditional messages. Examples are order event and trade confirmation messages. By time stamping these downstream messages, and linking them to their parent messages via unique identifiers, exchanges can build entire message trees. The exchange can track the complete life cycle of a message and subsequent events in this way. Intelligent assignment mechanisms make it possible to add these time stamps with minimal impact on overall performance and latency.

The technical support staff at an exchange, with this complete data history, can conduct detailed performance analysis, troubleshooting, as well as capacity management. And because some or all of these time stamps are also available to members, there is full end-to-end transparency. That means that trading firms may analyze system behavior and optimize their infrastructure accordingly.

7.8 Conclusion

In this chapter, we have seen that today's equity markets depend on state-of-the-art information technology. A fully electronic trading environment must balance competing objectives, including reliability, transparency, and high performance.

The exchange needs to account for and to reconcile a diverse set of technical and functional requirements within the exchange member community. The expectations of latency-sensitive market participants have proven to be the strongest driver for innovative and pioneering technology concepts in equity trading systems.

The next generation of technology will continue to transform the exchange system architectures and the exchange ecosystems as a whole. Blockchain technology, cloud computing design principles, big data processing, and mobile computing, to name a few, will create new unprecedented opportunities to shape the future of the financial industry.

Further Reading

St. Hammer, *Architects of Electronic Trading: Technology Leaders Who Are Shaping Today's Financial Markets*, Wiley, 2013

Trading Technologies Ltd., *A Strategic Study of Stock Exchange Technology*, Metal Bulletin, 2003

C. Zaloom, *Out of the Pits: Traders and Technology from Chicago to London*, University of Chicago Press, 2006

D. Lohfert, "Battling the bottleneck: an analysis of the new Eurex platform", *Automated Trader Magazine*, Issue 30, Q3 2013, pp. 16–20. Also available at: http://www.automatedtrader.net/articles/analysis/144302/battling-the-bottleneck-an-analysis-of-the-new-eurex-platform.

H. Cumberland, 2014. Futures & Options World: "Riding along on the crest of a wave". Available at: http://www.fow.com/3308917/riding-alon.html.

St. Hammer, 2014. Futures & Options World: "The 'Race to Zero' continues with faster hardware". Available at: http://www.fow.com/3315483/The-Race-to-Zero-continue.html.

D. Wigan, 2013. Futures & Options World: "Microwave arms race gathers speed". Available at: http://www.fow.com/3248074/Microwave-arms-race-gathers-pace.html.