part of eex group

M7G Functional Documentation

16.11.2026

Ref. 2.0

Content

1. Introduction 4

1.1 Overview of Entities and Their Relationships 4

1.2 Entity Life Cycle 6

2. Market 7

3. Product 8

3.1 Definition 8

3.2 Product State 8

3.3 Product Properties 8

3.4 Contract Properties 10

3.5 Product Types 10

4. Contract 12

4.1 General Information 12

4.2 Contract Phase and State 12

4.3 Contract Generation 13

5. Orders 14

5.1 Common Properties of Orders 14

5.2 Regular Orders 14

5.3 Ephemeral and Market Orders 16

5.4 Order Lifecycle 17

6. Order Book 18

6.1 Price-Time Priority 18

6.2 Maintaining of Up-to-Date View of Order Book 18

6.3 Order Book Indicators 18

6.4 Relationship between Contract and Order Book 18

6.5 Self-Match Prevention and Member Trading Restriction 19

© EEX AG

6.6 Crossed Order Book 19

7. Implied Orders 20

7.1 Rules for Implication 20

8. Trade 21

8.1 Trade Cancellation 21

9. Members 22

9.1 Member Lifecycle 23

10. Trading Groups 24

10.1 Other Electronic Access (OEA) 24

10.2 Trading Groups for Regular Members 24

10.3 Trading Groups for OEA Members 24

10.4 Trading Group Lifecycle 25

11. Users 26

11.1 User Roles and Rights 26

11.2 User Lifecycle 28

12. Self-Match Prevention 29

13. Data Interfaces 30

14. Authentication 31

14.1 User Credentials and Tokens 31

14.2 Authentication by User Type 32

15. Live API 33

15.1 Trading Interface 33

15.2 Market and Reference Data Interface 34

15.3 Common Usage Examples 35

15.4 Concepts 36

16. Historical API 39

16.1 Data Retention 39

© EEX AG

1. Introduction

The M7 Gas (M7G) Platform is the central Trading System for the EEX Gas Spot Market. It is designed to provide all Market Participants with a unified and robust environment for their trading activities, from order entry through to execution.

Participants can perform full order and trade management within a continuous matching environment for efficient price discovery. M7G natively supports integrated spread trading.

A key feature of M7G is its public Application Programming Interface (API), which provides all trading participants with direct, standardized, and efficient access to the platform's trading functions. This modern API is specifically designed to support a wide range of strategies, including algorithmic trading, by allowing low-latency interaction with the market. Independent Software Vendors (ISVs) as well as participants with proprietary systems can use this API to integrate their applications and interact directly with the EEX Gas Spot Market's Order Books.

1.1 Overview of Entities and Their Relationships

1.1.1 Market

A Market is a covering term for all Products on M7G.

1.1.2 Product

A Product is a general definition of the conditions a commodity is traded in specific area for specific delivery period (hour, day, etc.).

1.1.3 Contract

Based on Product properties M7G automatically generates Contracts for specific delivery date and time intervals (e.g. from May 15 06:00 to May 16 06:00).

1.1.4 Order Book

An Order Book represents a real-time list of all active Buy and Sell Orders for a Contract. M7G generates an Order Book once the Contract becomes tradable. When Contract becomes tradeable M7G creates an Order Book - place where Orders are listed.

© EEX AG

1.1.5 Order

An Order is an expression of interest of a Client to buy or sell a defined number of quantified Gas Spot Contracts at a desirable price.

1.1.6 Implied Order

An Implied Order indicates trading opportunity in Order Book which is allowed by relations between multiple Contracts via Spread trading. This form of implication shows the trading opportunity underneath consisting of multiple Trades with a guaranteed execution. Each Order Book update contains a new set of Implied Orders valid for the moment of the update; Implied Orders do not have any revisions.

1.1.7 Trade

A Trade represents the final outcome of the trading process i.e. when the Buy and Sell order successfully match (when the price of a Buy Order is equal or higher than the price of a Sell Order). A Trade is always referencing one Buy Order and one Sell Order.

1.1.8 Member

A Member represents a top-level Trading Participant authorized by EEX to participate in trading. Members usually correspond to legal entities such as companies, brokers, and auctioneers. For the administration and oversight of trading activities on a Member, designated Users, known as Supervisors, might be requested.

1.1.9 Trading Group

A Trading Group is a subdivision of a Member used to organize, control and separate trading activities within that company. A Member can have one or more disjunctive Trading Groups which define visibility of Orders and Trades for their respective Users. Trading Groups can also have restricted access for entering Orders for some Products.

1.1.10 User

A User represents the lowest-level of a Trading Participant hierarchy. It corresponds to a single human or algorithmic trader account who interacts with M7G. Each User belongs exactly to one Trading Group (except Supervisor).

© EEX AG

1.2 Entity Life Cycle

1.2.1 Revision Rules

With an exception to Implied Orders, all entities have a life cycle which starts with a revision 1. After every change on an entity, this revision increments by 1, and an update with a full description of the changed entity is sent via Live API. Some properties of an entity might be hidden for a Client due to visibility rights, thus Client might receive an updated revision of an entity where the only changed property is the revision itself.

1.2.2 Entity State and Life Cycle

Each entity in M7G can transition between several non-terminal states during its life cycle (for example, a Product may be in Trading, Halt, or Inactive states). When an entity becomes obsolete, M7G changes its state to Archived, which is a generic state applied to all entities. Once archived, the entity is no longer accessible via the Live API, and clients will not receive any further updates related to it.

1.2.3 Entity State Inheritance

The state of an entity may be inherited from its parent entities or set manually on a specific entity. The Effective State combines inherited state from parent entity (or several entities, for example, for a Spread Product) and manual state which might be set by Market Operator. This calculation is performed by M7G to avoid extra implementation on a Client side. Clients will receive only the Effective State of an entity via API.

© EEX AG

2. Market

The Market and its state define which trading activities are available on M7G. Depending on the selected state, either all products remain tradable, only spread products can be halted, or trading can be halted for all products.

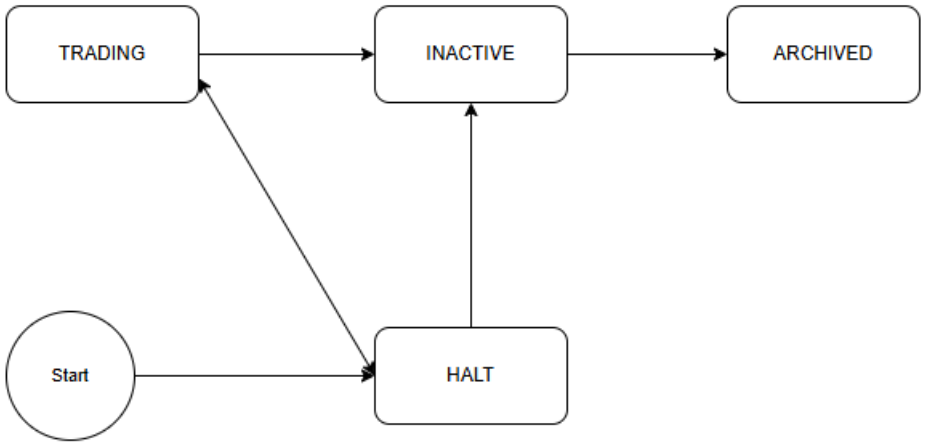
© EEX AG

3. Product

3.1 Definition

M7G supports Spot trading of Outright, geographical Spread, and gas quality Spread commodity Products (referred as Outright and Spread Products further on).

3.2 Product State



Product is always in one of the following states: Trading, Halt, Inactive, or Archived. Market Operators are allowed to modify tradability on a Product by changing its state between Trading and Halt. Inactive and Archived states are used when product is being decommissioned from M7G.

3.3 Product Properties

At the product level, we define Properties that remain constant and describe the fundamental characteristics of the product. These include the Market Area where the gas is delivered, the Gas Quality, the Balancing Type, Maturity, Currency, and Quantity Unit.

3.3.1 Maturity

3.3.1.1 Hourly

An individual full hour on a given date within which delivery takes place (e.g. 2023-10-29 01:00 - 02:00).

© EEX AG

3.3.1.2 Day

Delivery Period from configurable time of any given day until configurable time of the following calendar day (for example at 06:00 CET/CEST). Contract Trading Period starting at configurable time two days before delivery and ends at configurable time of the day of delivery (for example at 03:00 CET/CEST). To ensure a consistent trading duration, the Contract Trading Period for Day Contracts is extended for those starting on a Sunday, Monday, or Tuesday, as well as for Contracts whose tradability would overlap with a Bank Holiday. Every Daily Contract has Trading Period that includes exactly two Working Days, plus any number of Non-Working Days (such as Weekends or Bank Holidays) that may occur during that time.

3.3.1.3 Within-Day

The remaining Delivery Period of the current Gas Delivery Day at the time the transaction is concluded. For the Market Areas with Daily Balancing the Delivery Period is the entire Gas Delivery Day. For the Market Areas with Hourly Balancing the Delivery Period is the remaining delivery hours of the Gas Delivery Day.

3.3.1.4 Weekend

Delivery Period from configurable time of the first Gas Delivery Day of the Delivery Period (generally Saturday at 06:00 CET/CEST) until configurable time of the first calendar day after the end of the Delivery Period (generally Monday at 06:00 CET/CEST). For EEX Natural Gas Spot Products, the delivery period also includes Gas Delivery Days immediately before or after a weekend if those days are classified as holidays under the EEX Trading Calendar for Natural Gas Products.

3.3.2 Market Area

Market Area (MA) is a geographic region within which gas is traded freely (i.e. without reflecting grid capacity constraints), usually on national level (e.g. PEG, French MA). Market Area is the single geographical unit in EEX Gas Spot Market, the Market model is not fragmented further and there are no functional differences between Gas Hub and Virtual Trading Point MAs.

© EEX AG

Outright Products are always scoped to a single MA (e.g. TTF WD), Spread Products may target one (e.g. THE L/THE H WD) or two MAs (e.g. THE/TTF WD).

For the full list of the available Market Areas, check the System Configuration document.

3.4 Contract Properties

When Contracts are generated additional properties are populated by M7G from the setup of corresponding Product.

3.4.1 Quantity Properties

Lot Size defines the granularity of allowed quantities which need to be between Minimum and Maximum Quantity. Quantity Decimal Shift property puts the value into context (e.g. 240 translates to 240 MWh/day on a Product with MWh/day Quantity Unit where the Quantity Decimal Shift is set at 0 and to 24.0 MWh/day where the Quantity Decimal Shift is set to 1).

3.4.2 Price Properties

Tick Size defines the granularity of allowed prices which need to be between Minimum and Maximum Valid Price. Also Minimum and Maximum Sane Price and Price Warning Threshold are defined to provide guidance to Clients when to apply warnings against entering unusual Price. Price Decimal Shift property puts the value into context (e.g. 50000 translates to 50.000 EUR on a Product with EUR Currency where the Price Decimal Shift is set at 3).

3.5 Product Types

3.5.1 Outright Products

An Outright Product is a single, standalone product that is traded independently, without being combined with or dependent on any other product. It is defined by a specific Maturity (delivery period), Market Area, and Gas Quality.

3.5.2 Spread Products

A Spread Product is a combination of two Outright products, typically referred to as legs, where one leg is bought and the other is sold simultaneously. Spread Products represent

© EEX AG

the price difference between two Outright Products, either as a trading opportunity indicated by the Market in the form of an Implied Order or as a Regular Order submitted by a Market Participant. They serve two purposes:

- Providing the Market Participants with a convenient option to execute sets of Trades (i.e. Trade Legs) at known conditions and with guaranteed match.
- Allowing the Market to increase traded volumes in a relatively less-liquid Outright Products through propagation of trading opportunities to other Outright Product's Order Books.

Currently, M7G supports geographical and quality Spreads.

3.5.2.1 Primary and Secondary Market Area

For the purpose of Spread Trading, individual Market Areas (and by implication Outright Products) are designated either as Primary or Secondary. In specific cases, a Spread Product may also be created between two different Gas Qualities in one Market Area, one again designated Primary and the other Secondary Market. This designation impacts interpretation of the Spread direction and by convention follows the liquidity in respective Market Areas - the less liquid Outright Product is considered the Secondary, the more liquid the Primary. The name of the Spread Product is in a format Secondary/Primary.

3.5.2.2 Spread Order Breaking

When Buy Order on a Spread Product is matched, the Order is broken into a Buy Order on a Secondary Product and a Sell Order on a Primary Product. When Sell Order on a Spread Product is matched, the Order is broken into a Sell Order on a Secondary Product and a Buy Order on a Primary Product.

3.5.2.3 Reference Price

Since a Spread Price represents only the Price difference between the Secondary and Primary Products, reference price is normally required to calculate the outright Trade Prices of the resulting legs. In the M7G Technical Go Live version, this reference pricing mechanism is not available. Therefore, the matched Spread Price is temporarily used as the Trade Price for the resulting trades.

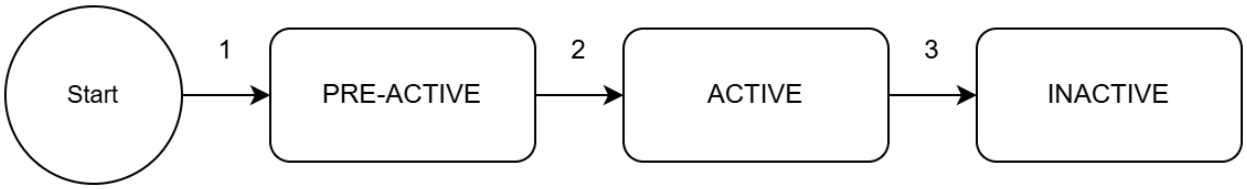
© EEX AG

4. Contract

4.1 General Information

A Contract represents a specific delivery period of the underlying product. Contracts are generated and managed through their lifecycle by M7G based on Product properties. The Product defines how a Commodity is traded (for example, hourly trading), while the Contract applies this setup to a specific delivery time period (e.g. 22 December 2025, 18:00–19:00). Orders and Trades are associated with specific Contract instances.

4.2 Contract Phase and State



image

Each Contract has a Phase (Pre-Active, Active, Inactive) which is managed automatically by M7G based on a planned Trading Period of the Contract.

Pre-Active Contract has been generated by M7G, but it is not yet active because the start of Trading Period is in the future. In the Active Phase, the Contract is available for trading. The Active Phase lasts for the full duration of the Trading Period. When the Contract enters the Inactive Phase, its scheduled tradability is considered complete and no further trading is permitted.

Each Contract has also a State (Trading, Halt). The Contract State defines whether the Contract in Active Phase can be Traded. The State of a Contract can be switched manually by the Market Operator.

Contract Phase and State together determine:

- What information about the Contract is visible to the Market
- What actions Market Participants can perform (such as entering Orders)

© EEX AG

4.3 Contract Generation

Contracts are automatically generated by M7G ahead of their Trading and Delivery Period, amount of pregenerated Contracts is based on the Maturity.

© EEX AG

5. Orders

An Order is an expression of interest of a Market Participant (a Member, practically represented by a Trader) to buy or sell commodity at specified quantity at a desirable price.

5.1 Common Properties of Orders

M7G supports three types of Orders: Regular, Ephemeral and Market. All types of Orders must be created for a specific Contract, and, depending on the Order Type, the User must define the following properties during creation: Side, Quantity and Price.

5.1.1 Side

Orders are created as either Buy or Sell Orders. While a Buy Order represents an instruction from a Market Participant to acquire a commodity in exchange for money, a Sell Order represents an instruction to deliver the commodity in exchange for money.

5.1.2 Quantity

The Quantity of an Order. The value is an integer and does not contain any indication of the quantity unit. The Product of the Contract defines the Quantity Unit.

5.1.3 Price

For Regular and Ephemeral Orders, the User must define a maximal Price at which a Buy Order can be matched or the min Price at which a Sell Order can be matched. It is also known as Order Limit Price in the trading industry. The Price is used to determine the execution priority of an Order based on the Price-Time Principle. The Price value is an integer and may be negative. It does not include any indication of currency. The Product of the Contract defines the Currency Unit.

5.2 Regular Orders

A Regular Order is an Order that, if not completely matched at the time it is created, is placed in the Order Book. The Order remains there until it is fully executed, put into hibernation, or deleted by either the User or M7G (for example, due to validity restrictions). Partial matching

© EEX AG

is permitted for this Order type. In addition, the User may define a Validity Restriction, which is available only for Regular Orders.

5.2.1 Validity Restriction

By applying Validity Restrictions, Users can control the specific period during which an Order is active and available for matching in the Order Book.

Users may select one of the following Validity Restriction types:

Validity Restriction	Description
Good till Cancelled (GTC)	This Order remains active in the Order Book until it is deleted by the User or the Trading Period ends.
Good till Time (GTT)	For this Order the User specifies a date and time until which the Order remains active in the Order Book. The GTT expiration must not be set beyond the end of the Trading Period.
Good For Session (GFS)	This Order remains active only for a duration of a session from which it was created.

5.2.2 Regular Order Modification

An Order modification refers to any change made by a Market Participant to the state or properties of an Order that has already been entered into M7G. Each successful modification results in the creation of a new Order revision, while the Order ID remains unchanged. Modification of an Order may lead to a change in its priority in the Order Book, depending on the nature of the modified properties.

5.2.3 Order Modification with Priority Change

Hibernated Orders are not part of the Order Book thus they do not have a priority. Rules below relate only to active Orders. Order modifications that lead to the priority change are:

© EEX AG

- any change of Price
- Quantity increase

5.2.4 Order Modification without Priority Change

All other Order modifications are without priority change.

5.3 Ephemeral and Market Orders

Ephemeral and Market Orders are eligible to be executed against available market liquidity only at the moment of their creation. If the execution conditions cannot be met at the submission time, the Order is assigned a revision with a deleted state by M7G. Ephemeral and Market Orders never enter the Order Book.

When creating an Ephemeral Order, the User must define the same properties as for a Regular Order: Side, Quantity and Price. When creating a Market Order, the User is not defining Price, their Order is executed at the best available price in the Order Book. These properties have the same meaning and constraints as described in Regular Orders section.

5.3.1 Execution Restriction

Instead of a Validity Restriction, the User must define an Execution Restriction for an Ephemeral and Market Order. Based on the selected Execution Restriction partial matching may be applied or not. One of the following Execution Restrictions may be applied:

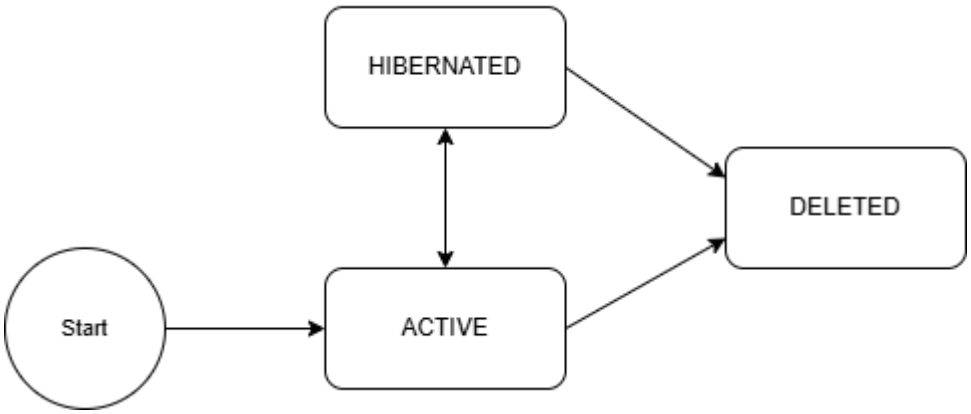
Execution Restriction	**Description**
Immediate or Cancel (IOC)	These Orders can be partially matched. Any quantity that cannot be matched immediately is cancelled and does not enter the Order Book.
Fill or Kill (FOK)	These Orders can be matched only for entire quantity and against one or multiple orders, otherwise the Order is cancelled and does not enter the Order Book.

The complete list of Order attributes is available in the Trading Interface documentation.

© EEX AG

5.4 Order Lifecycle

At any point in time, an Order is always in one of the following states: Active, Hibernated or Deleted.



Active Orders are exposed to the market in the Order Book and can be matched by Orders on the opposite Side. When Orders are Active (and thus in the Order Book) they can be modified, deleted or hibernated.

An Order in the Hibernated state is not eligible for matching. The Order remains stored in M7G but is suspended from trading activities. It can still be modified, deleted or activated. Hibernation does not represent a permanent removal of the Order.

An Order in the Deleted state is permanently removed from the trading process. Deleted Orders are no longer eligible for matching and cannot transition to any other state. Orders may be deleted either by the User or automatically by M7G, for example due to expiration, execution rules, or the end of the Trading Period.

5.4.1 Order Execution

In case of Full Order execution, the Order quantity is exhausted and the Order is removed from the Market. In case of Partial execution of a Regular Order, the Order remains Active with updated quantity and retains its Order ID (only a new revision is created). When a partially executed Order has a remaining quantity below the Minimum Quantity defined for the related Contract, the remainder of Order is deleted.

© EEX AG

6. Order Book

The Order Book is a real-time list of Buy (Bid) and Sell (Ask) Orders for a specific Contract in M7G. Orders are ranked according to the Price-Time principle and are displayed individually with their respective Quantity and Price information, with no aggregation performed by M7G. An Order Book generally cannot be crossed, meaning it does not contain Buy and Sell Orders with overlapping prices, except as described in section Crossed Order Book.

6.1 Price-Time Priority

Price-time priority is a standard order-matching algorithm that prioritizes Orders based on the best price first, and then the earliest submission time (or Order modification with a priority change). Submission time (or Order modification with a priority change) of the latest underlying Order is used to prioritize Implied Orders.

6.2 Maintaining of Up-to-Date View of Order Book

To maintain an accurate local representation of the Order Book for a given contract, a client must first retrieve an initial snapshot of the Order Book upon connection. Subsequently, the client must process real-time incremental updates (deltas), including order additions, modifications, and deletions, in accordance with the snapshot-and-updates synchronization model described in section Market State Synchronization.

6.3 Order Book Indicators

With each change of the Order Book, M7G updates a set of indicators and provides them to the Market for better awareness of its state. The indicators include the price, quantity, and time of the last Trade; the direction of the price movement; and the highest and lowest traded prices since the start of the Trading Period.

6.4 Relationship between Contract and Order Book

When the scheduled Trading Period of a Contract starts, the Order Book entity becomes retrievable via API. Even when the Contract is halted, Order Book is still retrievable via API, although it is empty. Once the Trading Period ends, the Order Book is deleted.

© EEX AG

6.5 Self-Match Prevention and Member Trading Restriction

To prevent self-trading, any aggressive Order that would match against an existing Order from the same Trading Entity is automatically canceled. This mechanism also prevents the Order Book from becoming crossed in such scenarios.

6.6 Crossed Order Book

In specific scenarios, the sides of an Order Book may include Orders with overlapping Prices. This constitutes a compromise of continuous matching and the application of the Price-Time principle. A crossed Order Book is impossible for Regular Orders but can occur in rare cases involving Implied Orders because the implied-order generation logic is intentionally limited to a maximum depth of two.

© EEX AG

7. Implied Orders

With Spread Trading feature, M7G creates Implied Order indicating trading opportunities in Order Books of relevant Contracts. Implied Orders are created by M7G and inserted into Order Books of either a Spread Product or an Outright Product that participates in Spread Trading (e.g. ETF/THE Day, ETF Day or THE Day Order Books). Unlike Regular Orders inserted by Traders, Implied Orders are implicitly calculated by M7G based on passive Regular and Implied Orders. When an Implied Order is matched, a set of Trades is created with the constituent Orders. Details of constituent Orders are provided by M7G for each Implied Order.

7.1 Rules for Implication

All Active Orders are considered for Implied Order generation (not only the top Orders in every Order Book). Implied Order of the depth of one consist of two underlying Regular Orders. Implied Orders of the depth of two consist of a Regular and Implied Order with a depth of one, or of two Implied Orders of a depth of one. Implied Orders of a depth of three or more are not supported by M7G. M7G informs the Market only about Implied Orders to the certain accumulated quantity calculated from the top of the Order Book. See the System Configuration document for applicable value. Implied Orders below this threshold are not displayed but are still considered for matching if an aggressive Order would match all orders on better position in the Order Book.

M7G allows Spread Products to be created with Outright Products which apply different Balancing types. For these combinations, the following rules apply:

- The Spread Products is configured with an explicit quantity unit (e.g. MW or MWh/d) by the Market Operator, no automated unit assignment is in place
- M7G recalculates Order Quantities of affected Outright Orders to the Spread Product unit (e.g. 20 MW on a DAY Contract would become 480 MWh/d on a Spread configured MWh/d as the Quantity unit)
- Only combinations with the same base unit are allowed. For example, a Spread Product can be created between an Outright in MW and an Outright in MWh/d, but not between an Outright in ktherm/d and an Outright in MW

© EEX AG

8. Trade

A Trade is executed when two opposing Orders (a Buy and a Sell) successfully match within the System. Upon matching, the Trade establishes a binding legal obligation: the Seller must deliver the commodity to the Buyer, and the Buyer must pay at the agreed-upon Match Price.

8.1 Trade Cancellation

However, there is still an opportunity to cancel a Trade. Trade cancellation is a rare process that may be initiated when a Trading Participant makes an objectively observable mistake while entering an Order into the Order Book, resulting in a transaction Price that differs significantly from the reference Price. Cancellation can be performed by a Market Operator upon request, submitted via phone call or email.

© EEX AG

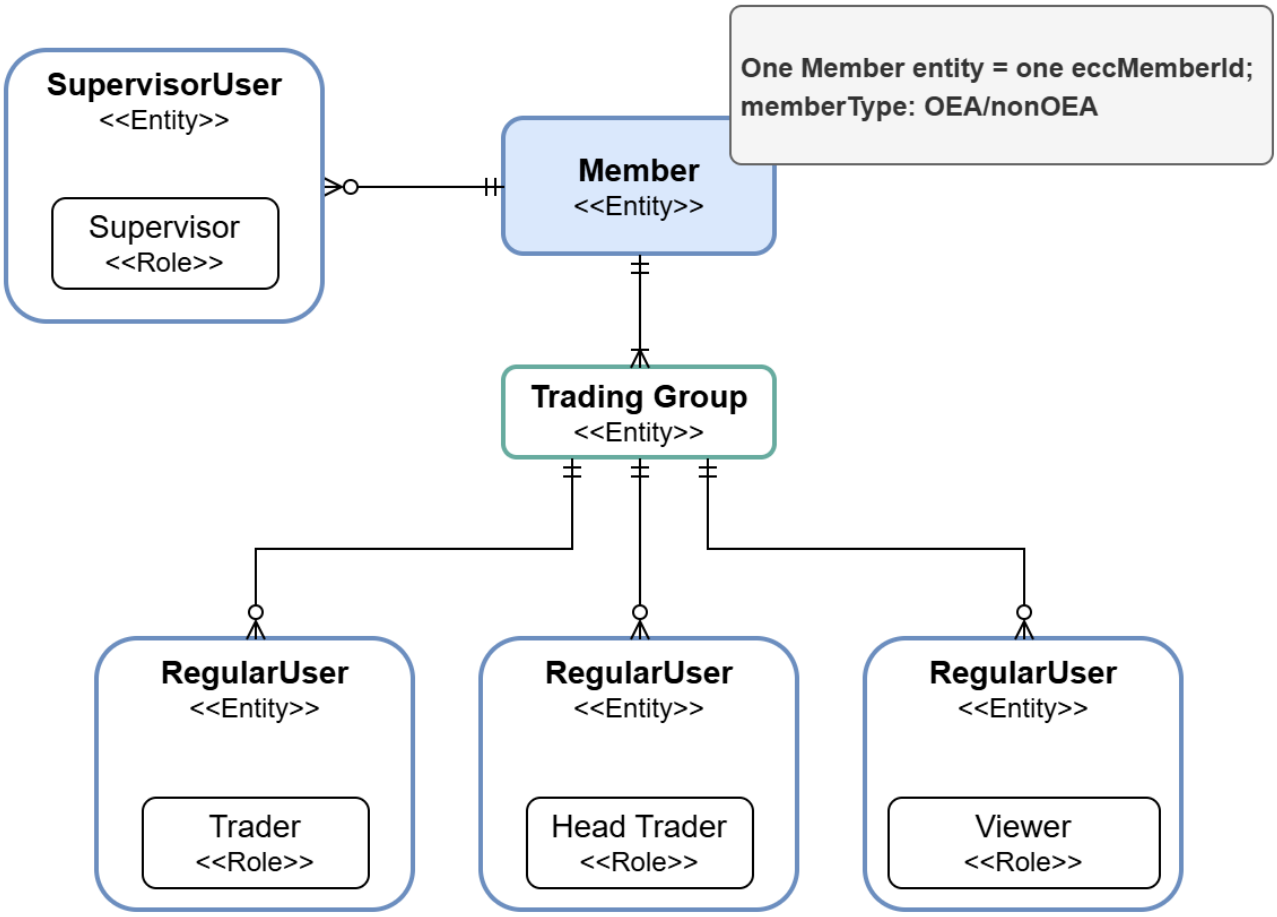
9. Members

Members represent companies who trade on the Exchange. They are represented by Viewers, Traders, Head Traders and Supervisors. The Member structure allows the following:

- Each Member can designate a Supervisors who have the rights to cancel Orders of this Member.
- Each Member's internal hierarchy is represented by Trading Groups. Each Trading Group consists of Users, where each User has exactly one Role assigned (Head Trader, Trader, or Viewer).
- Each Member is assigned a single unique Clearing ID.
- Depending on the Member type, specific trading rules apply. For users of non-OEA Members, it is not possible to trade between different Trading Groups under the same Member. Users of OEA (Other Electronic Access) Members can be assigned to one of two Trading Group types: OEA or Own. Users of Trading Groups of type Own are not permitted to trade with users of other Own Trading Groups.

Visibility of private data depends on whether the User is of the Supervisor Type. More details can be found in the section User's Roles and Rights. The illustration below provides a Member structure example.

© EEX AG



9.1 Member Lifecycle

Within M7G, a Member may be in one of the following states at any given time: Active or Suspended. If a Member is suspended, all Users belonging to that Member are also suspended and do not have access to the Exchange. Suspension is not permanent, and a Member may be reactivated at any time, subject to applicable rules and permissions.

© EEX AG

10. Trading Groups

All Users except those with the role of Supervisor must be setup as belonging to a Trading Group. A Trading Group can be of type OEA or Own, depending on the Member type. OEA Members may have Trading Groups of both types (OEA and Own), whereas non-OEA Members may only have Trading Groups of type Own. Each Trading Group can represent a separate company or a smaller organizational unit required for business purposes. Users belonging to a Trading Group can only view transactional data (full details of Orders and Trades) related to their own Trading Group. The number of Trading Groups per Member is not limited, regardless of the Member type.

10.1 Other Electronic Access (OEA)

Other Electronic Access is intended for companies that wish to trade on the Gas Spot Exchange but are unable to become Members themselves due to various constraints, such as entry fees, company size, or approval requirements. In such cases, certain Members, referred to as Access Providers, may offer these companies the ability to trade under the Access Provider's Member ID.

The Access Provider remains legally and operationally responsible for any transactions concluded under its Clearing ID.

10.2 Trading Groups for Regular Members

For Regular Members, Traders belonging to different Trading Groups (with type Own) under the same Member are not allowed to trade with each other. This is ensured by Self-Match Prevention.

10.3 Trading Groups for OEA Members

Trading Groups are primarily created to represent companies that trade through Other Electronic Access (OEA). Traders belonging to different Trading Groups of type OEA under the same OEA Member are allowed to trade with each other. Users belonging to Trading Groups of type Own are not allowed to trade with users of any Own Trading Groups of the same OEA Member. However, they may trade with users belonging to OEA Trading Groups.

© EEX AG

10.4 Trading Group Lifecycle

Within M7G, a Trading Group may be in one of the following states at any given time: Active or Suspended. If a Trading Group is suspended, all Users belonging to that Trading Group are also suspended. A Trading Group may be suspended by the Market Operator, or automatically by M7G. Suspension is not permanent, and a Trading Group can be reactivated.

© EEX AG

11. Users

A User is either a human or a technical entity that interacts with M7G. Within M7G, Users are assigned one of five roles: Supervisor, Head Trader, Trader, Viewer, or Market and Reference Data User (MRD User).

A Supervisor User is always assigned to a Member and has limited rights in trading activities. Supervisor Users are responsible for oversight rather than direct trading. Further details regarding their permissions are described in the User Roles and Rights section.

Head Trader and Trader Users represent human users or technical entities authorized to receive public and private Market and Reference Data and to perform trading activities in M7G. Viewer Users are authorized to receive public and private Market and Reference Data but are not permitted to perform trading activities. All Head Trader, Trader, and Viewer Users must belong to a Trading Group.

A MRD User can only connect to MRDI and is intended to be used to open a single common Market and Reference Data channel with M7G.

Each User is assigned exactly one Role. Once assigned, a User's Role cannot be changed. To use a different Role, a new User must be created and assigned that Role.

11.1 User Roles and Rights

The sections below describe all available Roles and their corresponding rights within M7G.

11.1.1 Supervisor

A Supervisor User has visibility into and rights to cancel Orders across all Trading Groups belonging to the associated Member, but is not authorized to create or modify Orders.

11.1.1.1 Permissions/Entitlements:

- Public information (info about Products, Contracts, Order Book, etc.).
- View all the Users of their Member.
- View private information about Orders and Trades that belong to the Users of their Member.
- Delete Orders placed by the Users of their Member.

© EEX AG

- View all Trades of their Member.

11.1.2 Head Trader

Head Trader is a Trader with extended rights in scope of their Trading Group.

11.1.2.1 Permissions/Entitlements:

- Public information (info about Products, Contracts, Order Book, etc.).
- View private information about Orders and Trades that belong to the Users of their Trading Group.
- Modify/delete Orders on behalf of other Users inside their Trading Group.
- Enter/modify/delete own Orders.
- View all Trades of their Trading Group.

11.1.3 Trader

A Trader can perform trading activities in the name of their Member.

11.1.3.1 Permissions/Entitlements:

- Public information (info about Products, Contracts, Order Book, etc.).
- Enter/modify/delete only own Orders.
- View private information about Orders and Trades that belong to the Users of their Trading Group.

11.1.4 Viewer

A Viewer can only monitor activity of their Trading Group, they are not eligible for trading.

11.1.4.1 Permissions/Entitlements:

- Public information (Info about Products, Contracts, Order Book, etc.).
- View private information about Orders and Trades that belong to the Users of their Trading Group.

11.1.5 Market and Reference Data User

A Market and Reference Data User who is only allowed to connect to MRDI, intended to be used by ISVs.

© EEX AG

11.1.5.1 Permissions/Entitlements:

- Public information (Info about Products, Contracts, Order Book, etc.).

11.2 User Lifecycle

Within M7G, a User may be in one of the following states at any given time: Active or Suspended. An Active User is permitted to log in to M7G and perform activities in accordance with the Role assigned to the User. A Suspended User is not permitted to log in to M7G. A User may be suspended either by a Market Operator or automatically by M7G. Suspension is not permanent, and a User may be reactivated at any time, subject to applicable rules and permissions.

© EEX AG

12. Self-Match Prevention

Self-Match Prevention (SMP) is a mechanism designed to prevent a Member's own Orders in the same Contract from matching against each other. Currently, SMP applies only to Orders within the same Contract and does not prevent self-matches across different Contracts resulting from Implied Orders. M7G currently supports the Cancel Aggressive (CA) SMP behavior. Under this behavior, an incoming aggressive Order is matched against passive Orders on the opposite side of the Order Book on an Order-by-Order basis. If execution of the incoming Order would result in a self-match, SMP is triggered and the remaining unexecuted quantity of the aggressive Order is immediately canceled. Any quantity already executed prior to the self-match detection remains unaffected.

© EEX AG

13. Data Interfaces

For trading instructions and retrieval of Market Data, M7G provides 2 distinct APIs – Live (Websocket with protobuf) and Historical (REST with JSON payload). Each has its own endpoints and its own specific usage. Live API provides current information (updates of the Order Book, updates on Market Entities etc.), it is designed to be fast. Its scope strictly targets current Market State and interactions.

Historical API provides historical data with potentially slower response.

© EEX AG

14. Authentication

User authentication is managed centrally through the M7G Identity Provider.

14.1 User Credentials and Tokens

All users authenticate with M7G Authentication Endpoint using unique usernames and passwords distributed via the established onboarding process. These credentials must be kept confidential. Upon successful credentials verification, M7G Authentication Endpoint issues limited-lifetime access tokens required to authorize access to the M7G Live API and the Historical API. If a token expires, users must re-authenticate by sending a new API request to M7G Authentication Endpoint to obtain a new access token or a refresh token.

For successful authentication with M7G Authentication Endpoint, credentials from the Credentials document need to be sent with the request. The process of obtaining the token is described below.

1. Send an HTTP POST request to the M7G Authentication Endpoint (example below):

```
Header:
Content-Type: application/json

Body:
{
"login": "<provided-login>",
"password": "<provided-password>"
}
```

2. M7G Authentication Endpoint Returns Response with Access Token (example below):

```
Response:
{"token": <token-string>}
```

3. For the M7G Live API Access Token is parsed on the Clients' side and used in a new WebSocket LoginRequest to an M7G Endpoint. For Historical API the token is provided in HTTP header `Authorization: Bearer $token`.

4. If successful, connection is established

As soon as the User is authenticated, the Session remains active until disconnected.

© EEX AG

14.2 Authentication by User Type

The system supports distinct authentication flows based on user roles:

- Independent Software Vendors (ISVs): ISVs authenticate to the Market Reference Data Interface (MRDI) using a dedicated MRD User account, and to the Trading Interface (TI) using the credentials of the individual Traders they represent.
- Independent Traders: Traders who are not represented by an ISV authenticate directly to both the TI and MRDI using their assigned credentials. Access to trading and market data functionalities is strictly restricted until the user successfully authenticates and presents a valid access token.

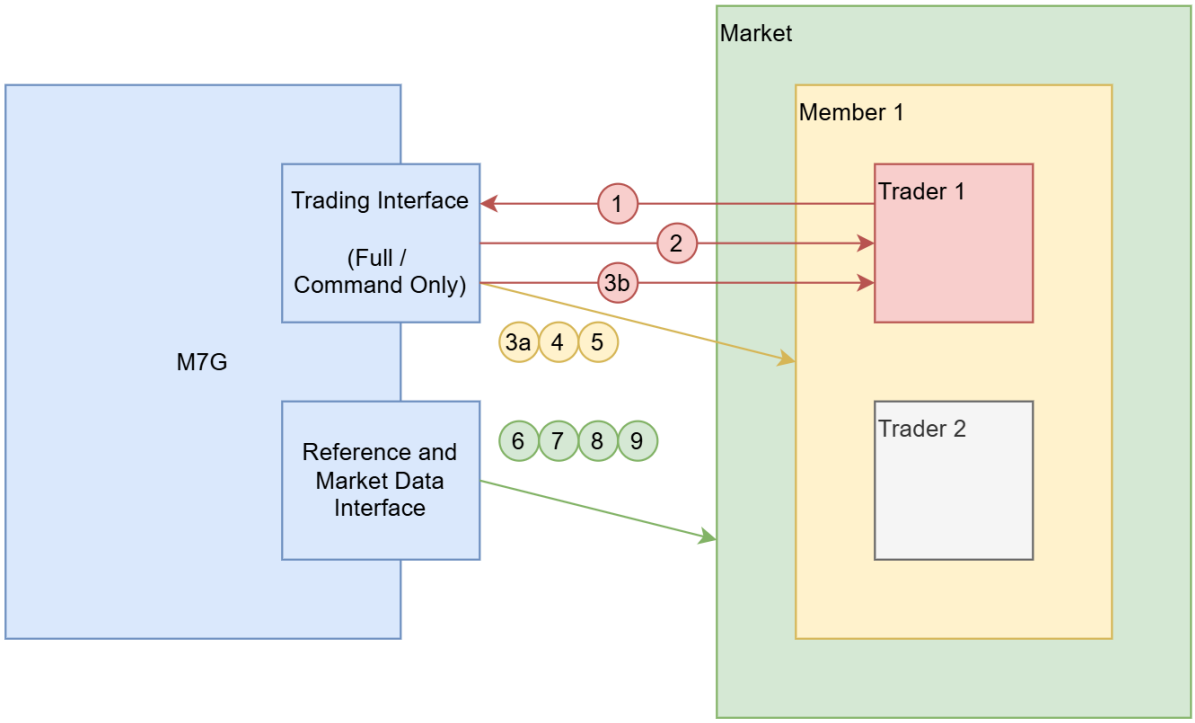
Successful authentication and possession of a valid token are required before access to trading and market data functionalities is granted.

© EEX AG

15. Live API

There are three Streams for communication with M7G, Trading Interface (TI), Trading Interface - Command Only and Market and Reference Data Interface (MRDI)

The diagram below shows Data flows on each interface. For reference, see Trading Interface and Market and Reference Data Interface sections below.



15.1 Trading Interface

Trading Interface is designated for Trading activities in user-specific context. This includes submitting commands receiving direct responses to these commands, and getting private updates

Trading Interface is used to:

- 1. send a trading instruction (e.g. to create or modify Orders)

© EEX AG

- 2. receive direct (i.e. only to the initiating User) responses to invalid instructions (e.g. error on an attempt to enter an Order on an expired Contract)
- 3. receive immediate response about a request (e.g. update on the created Order)
- (3a) - on Organization Unit level
- (3b) - on their own command
- 4. receive updates on Orders and Trades belonging to the User's Trading Group or Member, depending on the User's role (e.g. update on an Order entered by a Trader of the same Trading Group)
- 5. receive Reference Data on private entities relevant to a User - (e.g. suspension of User's Member)

To log in to Trading Interface, a Market Participant sends LoginRequest and receives one of these LoginResponses:

- SessionEstablished
- LoginFailed

The Market Participant can also receive InterruptedByAnotherLogin message any time. It is the last message in the stream, immediately after the session is dropped.

15.1.1 Trading Interface - Command Only

The Trading Interface - Command Only is designed for Trading activities in user-specific context, where the Traders can send trading instructions and receive direct responses without the broader Trading Group scope of immediate responses, snapshots or reference data (Diagram - Data Flows 1, 2, 3b).

15.2 Market and Reference Data Interface

Market and Reference Data Interface broadcasts information to all Market Participants. This includes general Market Data such as Order Book updates, reference data like opening of Contracts for trading, Market State changes, and reports on matched Trades with anonymized data.

Market and Reference Data Interface is used to receive:

© EEX AG

- 6. Order Book snapshots – public data on all Orders
- 7. Order Book updates (Order Book deltas)
- 8. public information of all Trades
- 9. public Reference Data - updates about Products, Contracts (e.g. Contract / Market Area Trading Halts)

To log in to MRDI, a Market Participant sends OutboundEventsRequest and receives one of following responses:

- OutboundEventsSuccess with a Snapshot of all market data
- OutboundEventsError with a message explaining the error

After establishing successful session, the Market Participant receives Transaction messages (updates of current state of entities they need to apply) including updates of Order Book data, public information on Trades or updates of Contracts.

15.3 Common Usage Examples

The examples below illustrate the described concepts.

15.3.1 Example 1: Happy Flow

- A Sell Order O1 placed by Trader T1 of Member MA is present in the Order Book
- Trader T2 of Member MB sends an Order Entry request for Buy Order O2 via Trading Interface. Immediately on arrival:
    - Order O2 is partially matched on the exchange with Order O1
    - Order O1 is fully matched
- All Supervisors of MA and all the Users belonging to the same Trading Group as T1 (including T1) receive in the Trading Interface:
    - Order Update with
        - Full match of Order O1
        - Trade Capture Report with private data for Order O1 on the Sell side
- All Supervisors of MB and all the Users belonging to the same Trading Group as T2 (including T2) receive in the Trading Interface:
    - Order Update with

© EEX AG

        - Creation of Order O2
        - Partial match of Order O2 with Matched Quantity Update
    - Trade Capture Report with private data for Order O2 on the Buy side
- All Users (of MA, MB and others) receive in the Market and Reference Data Interface
    - Public Order Book Update Delta with
        - Removal of Order O1 and
        - Creation of Order O2 with the remaining Quantity
    - Trade Confirmation of the resulting Trade with anonymized data (i.e. no Member / User data are exposed)

15.3.2 Example 2: Error Response

- A Trader T1 of Member MA sends an Order Entry request for a Buy Order O3 via Trading Interface. Order O3 is being sent on a Contract which has recently expired.
- T1 receives via the Trading Interface
    - An Error Response indicating Order O3 has been rejected by the exchange
- No other Users, of MA or other Members, receive any communication on Trading or Reference and Market Data Interface.

15.4 Concepts

The concepts described below apply to Trading and Market and Reference Data Interfaces.

15.4.1 Message Sequencing

All Messages sent by M7G carry an incremental transactionIndex referencing the internal event that triggered them. The transactionIndex does not increase by a fixed step, but it is strictly monotonically increasing — each new Update always has a higher transactionIndex than the previous one.

15.4.2 Market State Synchronization

Both API streams use a snapshot-and-updates model for real-time Market synchronization. Upon successful Session establishment and authentication, the Client receives an initial Snapshot representing the current data on the Market. The Snapshot serves as the baseline

© EEX AG

for all subsequent updates. The stream then delivers updates (deltas) describing incremental changes to the Market. To maintain synchronization with the server, Clients must process updates in the order they are received and apply them to their local state.

15.4.2.1 Resume on Interrupt

During login to TI or MRDI, the Client can optionally request resumption from a specific transactionIndex. In this mode, M7G does not send a Snapshot; instead, it streams all Updates following the requested transactionIndex unless transactionIndex is too old (see the System Configuration document for details).

15.4.3 Periodical Transaction messages

M7G sends a Transaction message on TI and MRDI with a frequency defined in the System Configuration document. These Transaction messages are initiated by M7G clock and serve purpose of informing about timed events like scheduled change of Contract State, expiration of an Order with GTT set, etc. Even in case of no activity to report the Transaction message is sent, but it doesn't contain any entity changes. This feature should be used to evaluate the state of Trading Client's session. If Client didn't receive Transaction message for 3 periods, the Client should assume the session is no longer healthy and reconnect by establishing a new session.

15.4.4 No Synchronization Between Streams

Streams guarantee ordering of events only strictly within themselves. The streams are independent of each other with the responsibility for synchronization between them resting with the Client. The measure to keep track is transactionIndex of individual events (a number which allows recognizing in which order individual events happened). TransactionIndex sequence is common for both streams and transactionIndex value is included in each message sent by M7G backend

15.4.5 Independent Streams

The target design allows connecting to Trading and Reference and Market Data Interface separately by the same User. For example, it is possible to connect just to the Market and Reference Data stream to consume public data exclusively.

© EEX AG

15.4.6 Rate Limits

To protect M7G from malfunctioning or non-compliant Clients and to ensure fair Market conditions, M7G enforces Rate Limits on incoming Requests. M7G tracks the number of Requests received from each Client within a defined time interval and rejects any Requests that exceed the configured threshold (see the System Configuration document for details).

15.4.7 Data Retention

The Live API enforces strict data retention limits on inactive entities, such as expired Contracts or removed Orders. For details on specific retention periods consult the System Configuration document.

© EEX AG

16. Historical API

M7G Provides the Historical API (HAPI) where historical information on Reference Data and Trading entities can be retrieved. It is a REST API with JSON message structure. A User can log in to HAPI with the same credentials as to the Live API. HAPI supports retrieval of information on Orders, Trades, and Contracts and allows filtering of the desired response for a range of attributes (e.g. Time Period or specific entity ID).

HAPI provides a full history of States an entity went through. For an Order that was created and accepted into the Order Book, then partially matched and eventually removed from the exchange due to the underlying Contract's expiration, HAPI will provide all three states with full details and respective revisions and timestamps of the underlying events.

The following conditions apply:

- HAPI is synchronized with the Market; however, there may be a delay between the time an event occurs on the Market and when the corresponding data becomes available through HAPI. Although this delay is typically minimal, Trading Clients should use the Live API to obtain time-critical information.

16.1 Data Retention

HAPI enforces retention limits on historical Reference Data and Trading entity data. For details on specific retention periods consult the System Configuration document.

More information on HAPI can be found in the API documentation.

© EEX AG