System Architecture for Automated Trading Systems

RUSH to Algo!

Agenda

- System Architecture of a Traditional Trading System
- Manual Trading Vs Algorithmic Trading System
- System Architecture of an Automated Trading System
- Various Market Data transmission methodologies
- Interactive Order Sending to exchanges
- Deep dive into Complex Event Processing (CEP) Module
- Deep dive into the Order Manager
- Future Trends

System Architecture of a Traditional Trading System

Traditionally a trading system would consist of …

System Architecture of a Traditional Trading System

Traditionally a trading system would consist of

- A system to read data from the market

System Architecture of a Traditional Trading System

Traditionally a trading system would consist of

- A system to read data from the market
- A storehouse of historical data

System Architecture of a Traditional Trading System

Traditionally a trading system would consist of

- A system to read data from the market
- A storehouse of historical data
- A tool to analyze historical data

System Architecture of a Traditional Trading System

Traditionally a trading system would consist of

- A system to read data from the market
- A storehouse of historical data
- A tool to analyze historical data
- A system where the trader can input his trading decisions

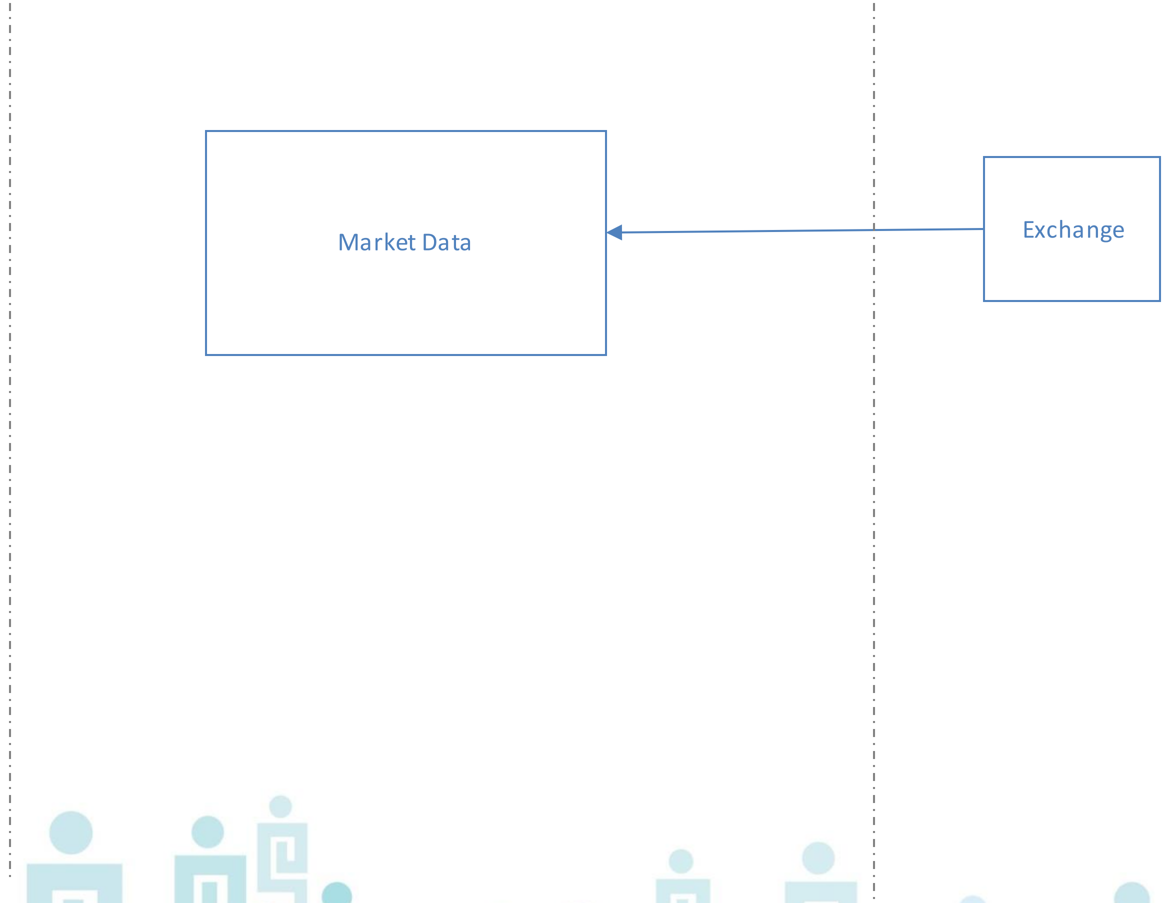
System Architecture of a Traditional Trading System

Traditionally a trading system would consist of

- A system to read data from the market
- A storehouse of historical data
- A tool to analyze historical data
- A system where the trader can input his trading decisions
- A system to route orders to the exchange

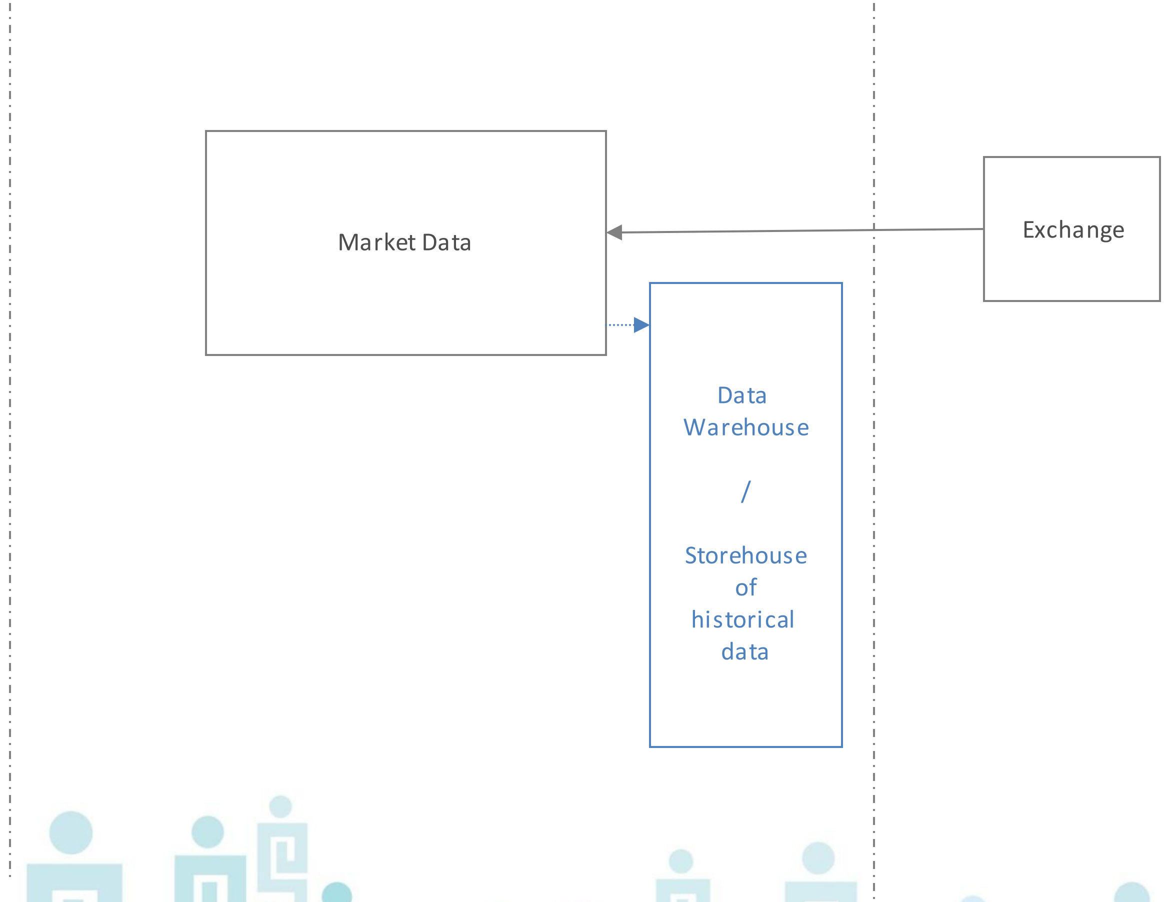
System Architecture of a Traditional Trading System

A system to read data from the exchange (market data adaptor)



System Architecture of a Traditional Trading System

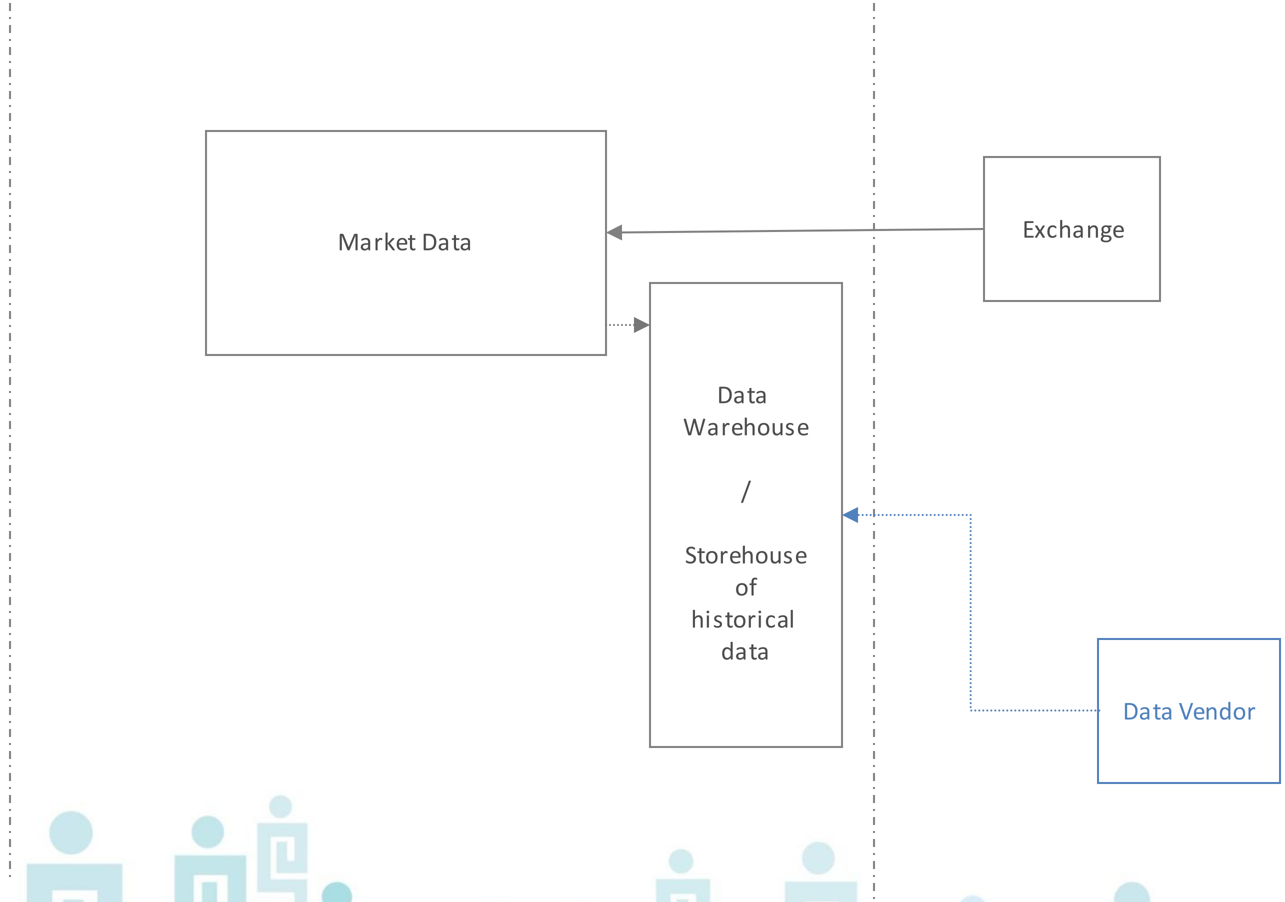
A storehouse of historical data



System Architecture of a Traditional Trading System

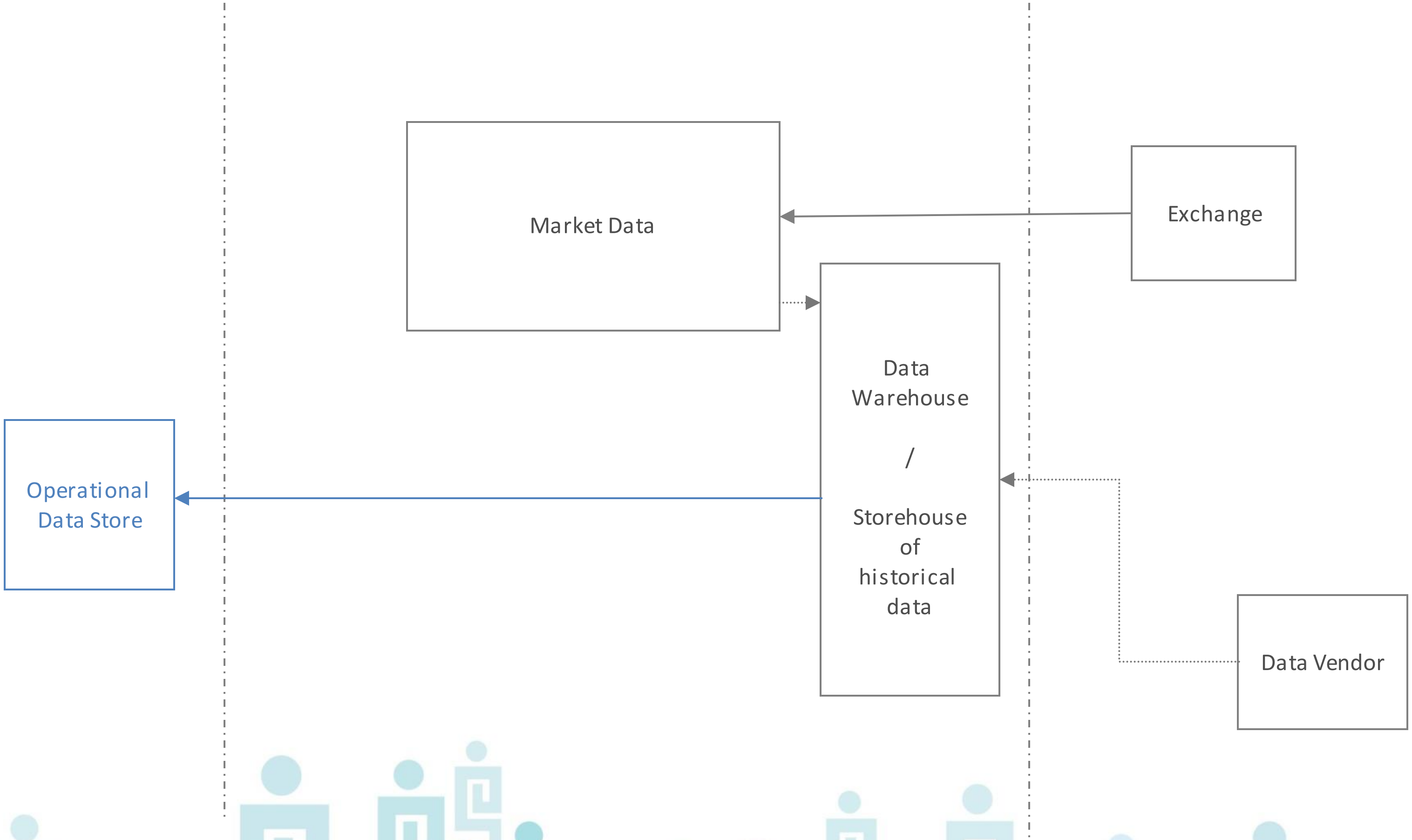
A storehouse of historical data

(which could also be procured from third party vendors)



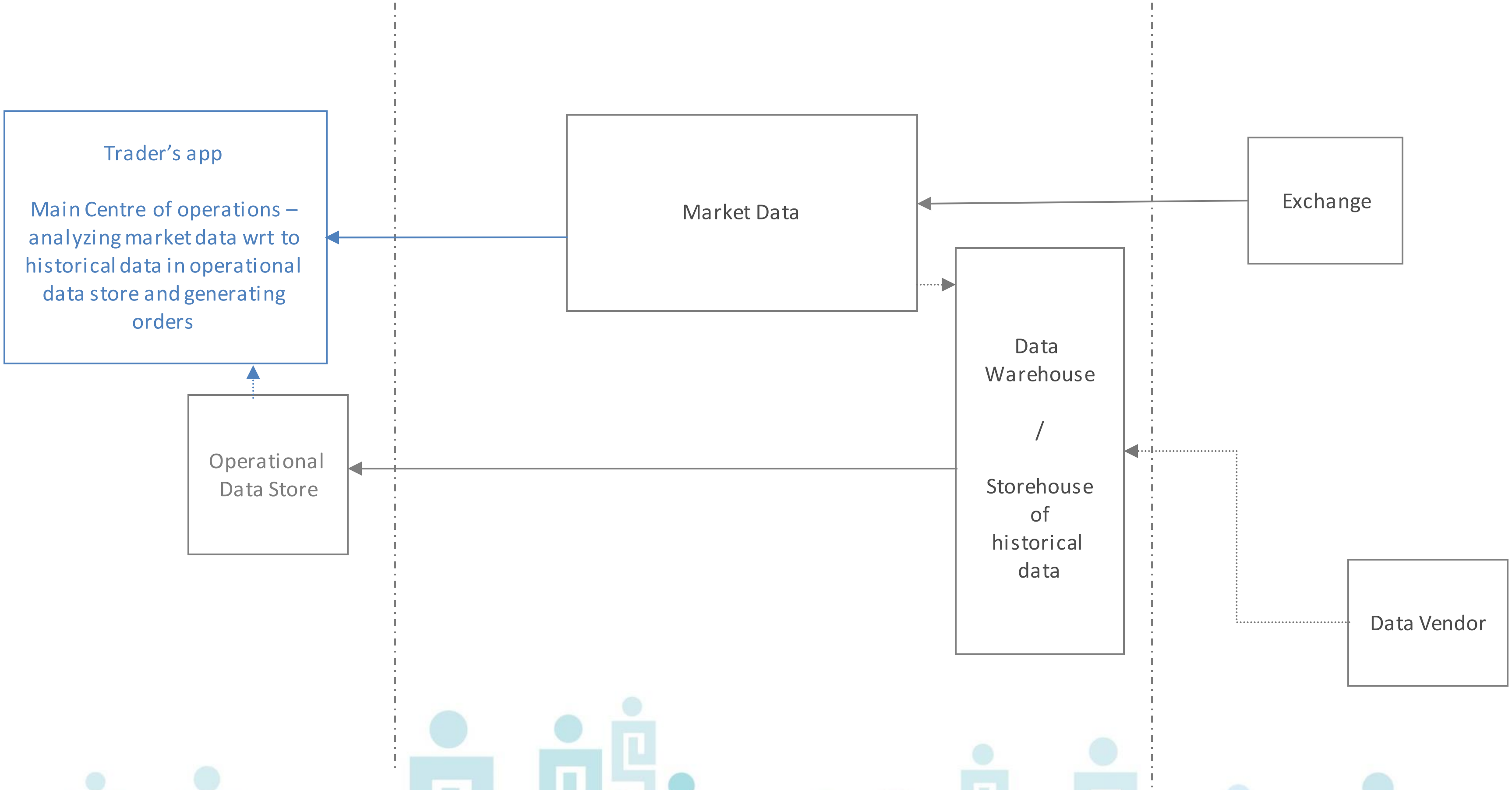
System Architecture of a Traditional Trading System

A subset of the information is stored locally in an operational data store for individual use



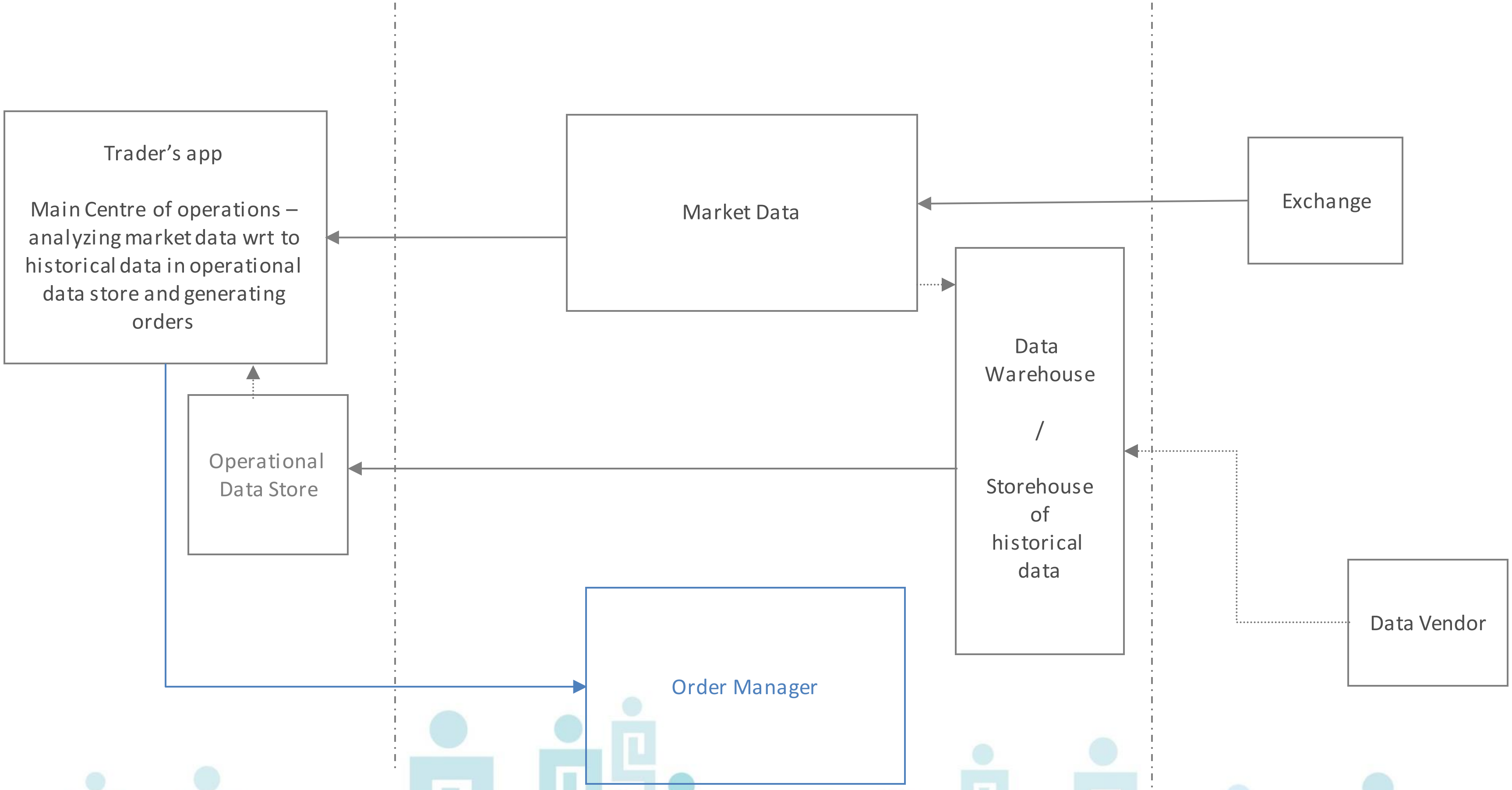
System Architecture of a Traditional Trading System

The trader’s tool would then analyze current data against patterns discovered in the operational data store


Trader’s app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Data Vendor

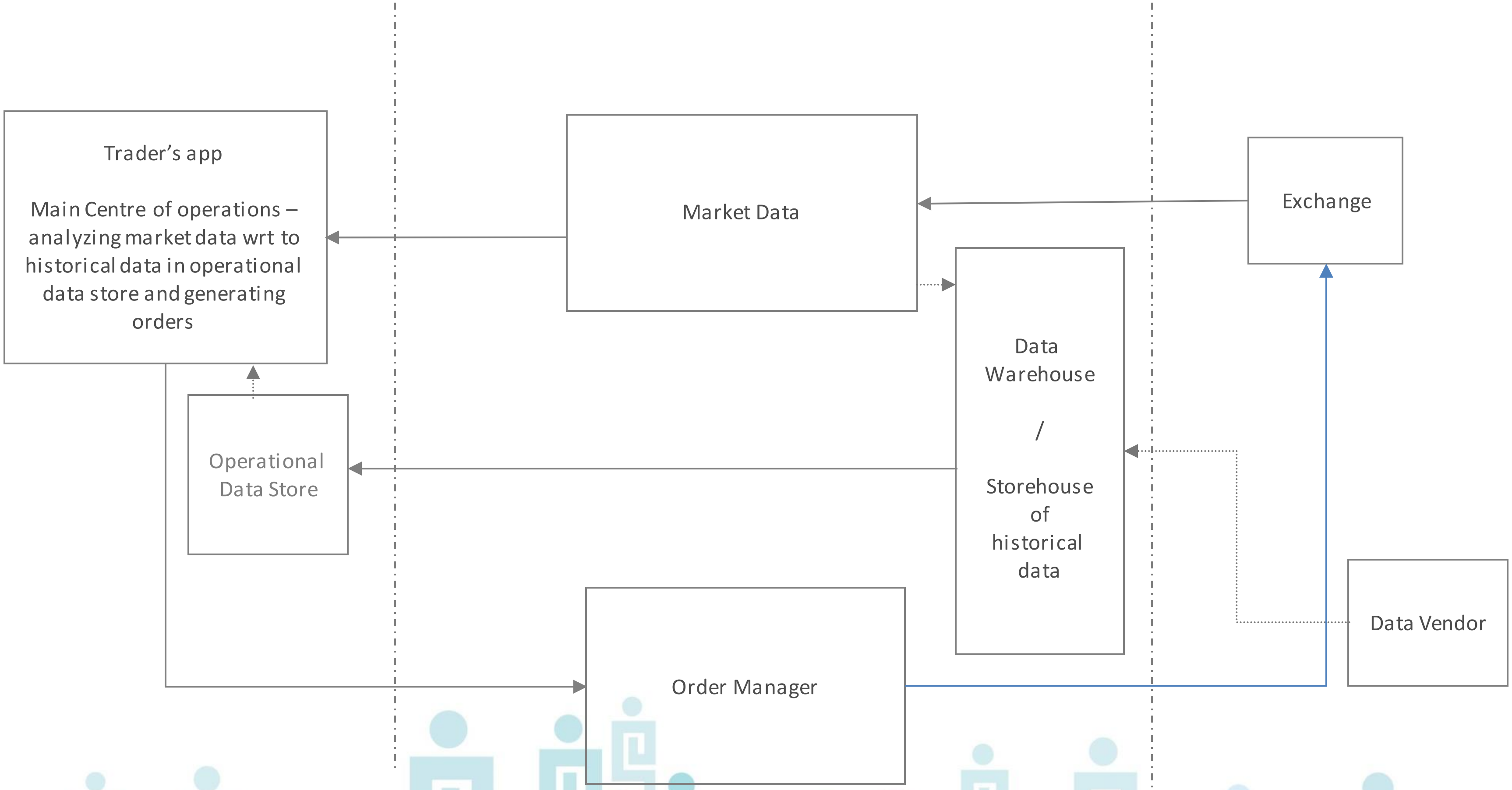
System Architecture of a Traditional Trading System

The trader's tool would then generate orders which will be forwarded to the order management tool


Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Order Manager
Data Vendor

System Architecture of a Traditional Trading System

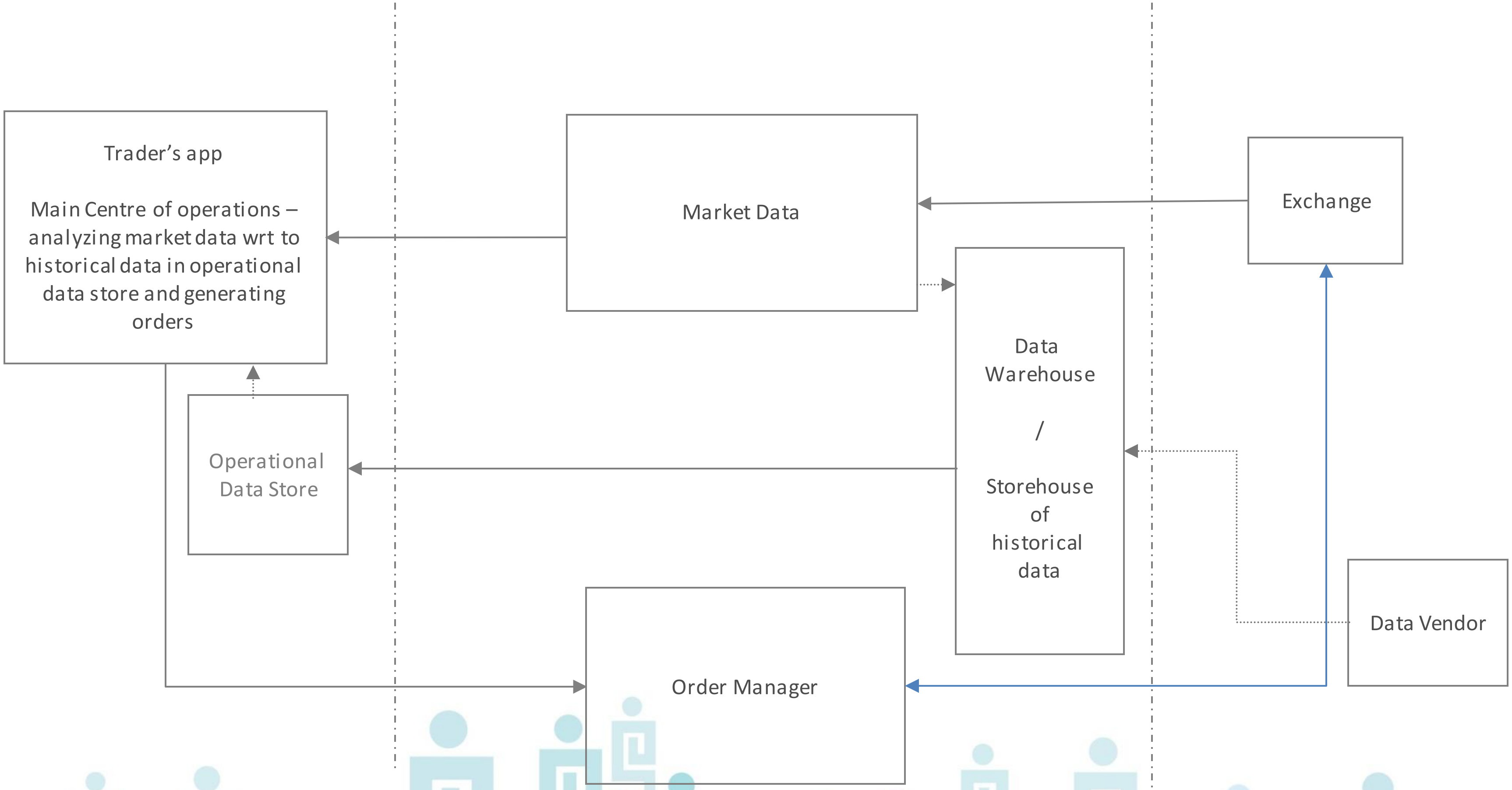
The order manager would then route the orders to the exchange


Trader's app

Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Data Vendor
Order Manager

System Architecture of a Traditional Trading System

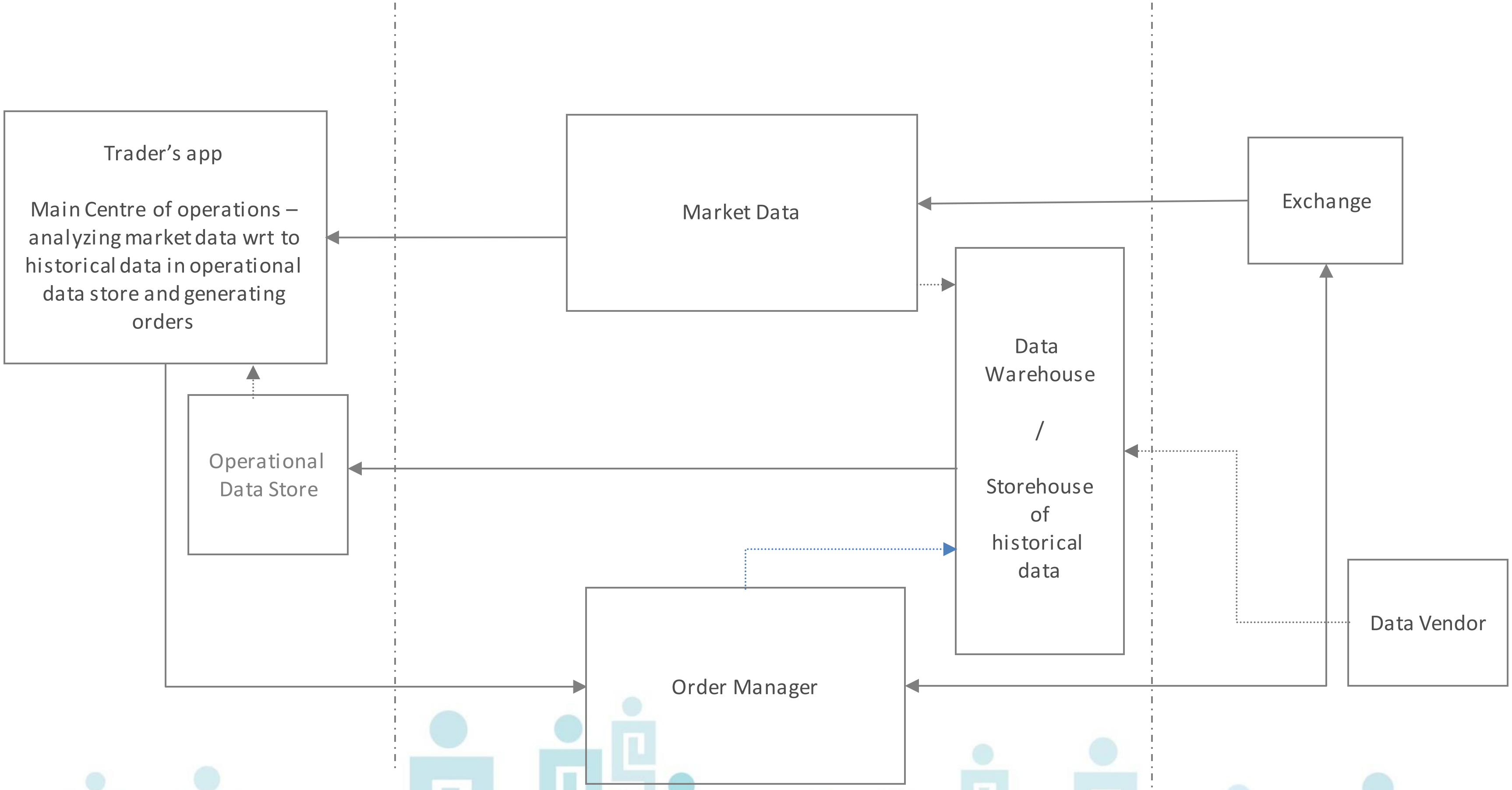
The order manager would also get response from the exchange about executions


Trader's app

Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Order Manager
Data Vendor

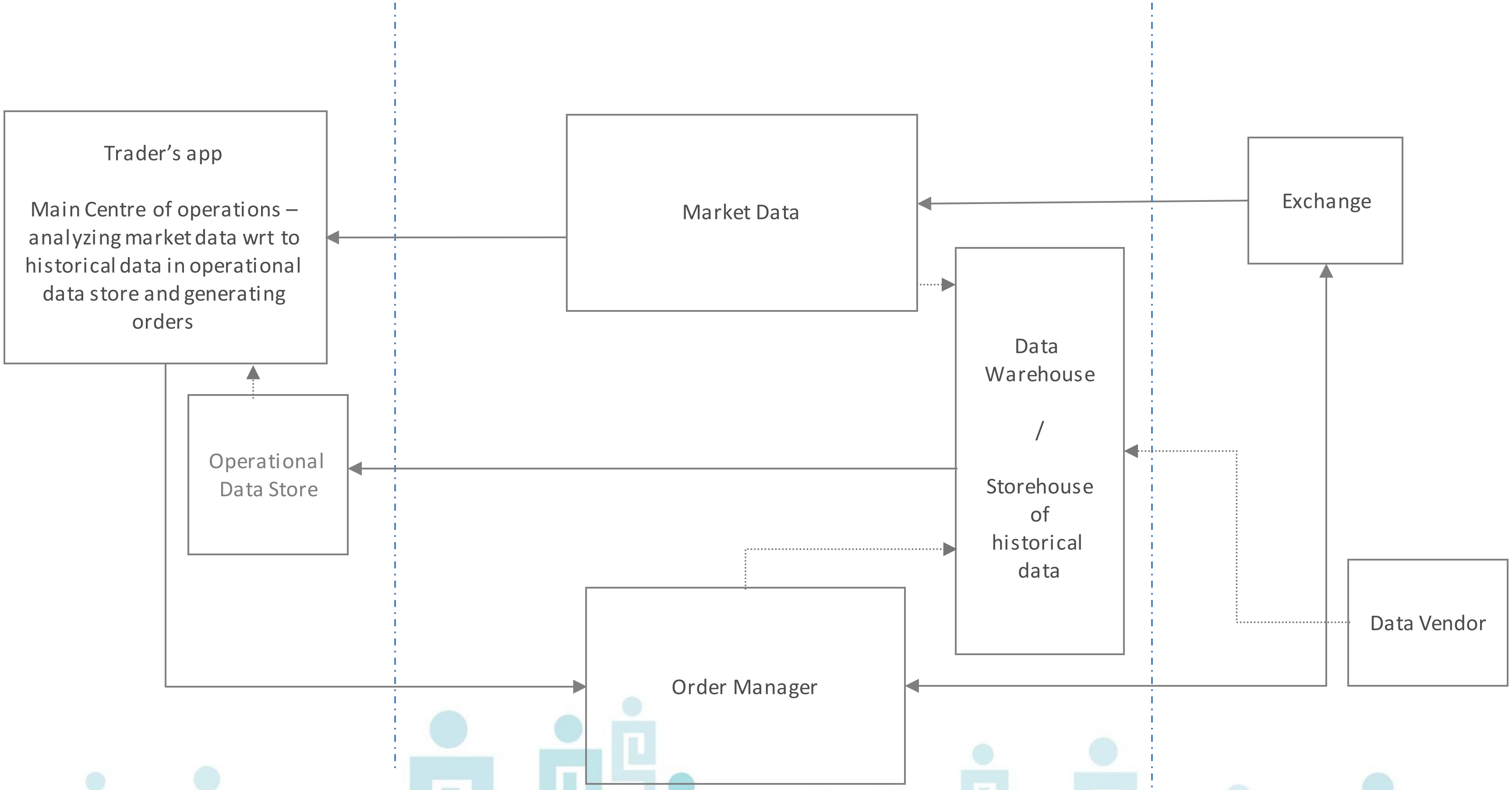
System Architecture of a Traditional Trading System

The data warehouse would also probably store records of orders sent out


Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Order Manager
Data Vendor

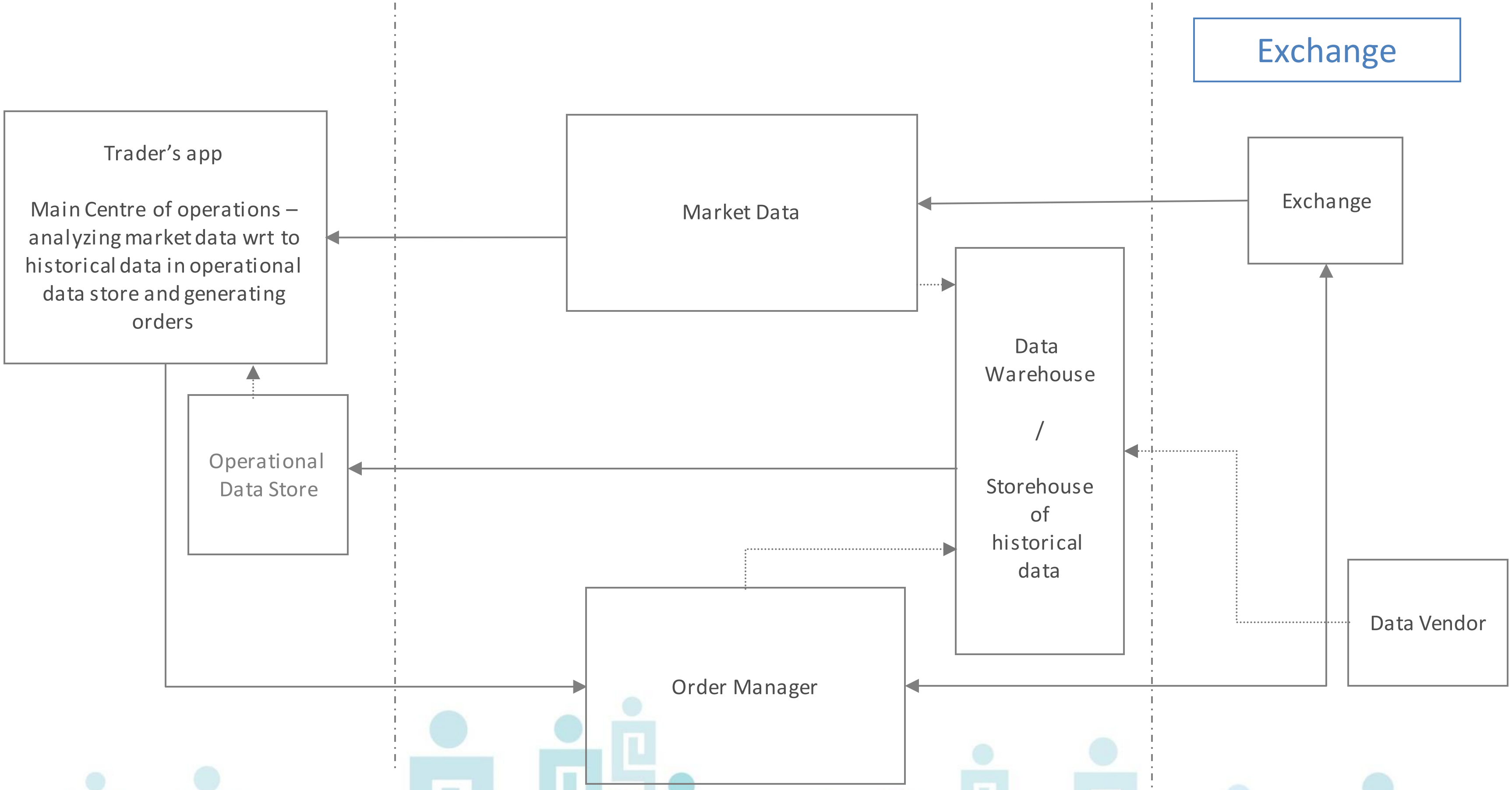
System Architecture of a Traditional Trading System

The whole system could be broken down to three components


Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Order Manager
Data Vendor

System Architecture of a Traditional Trading System

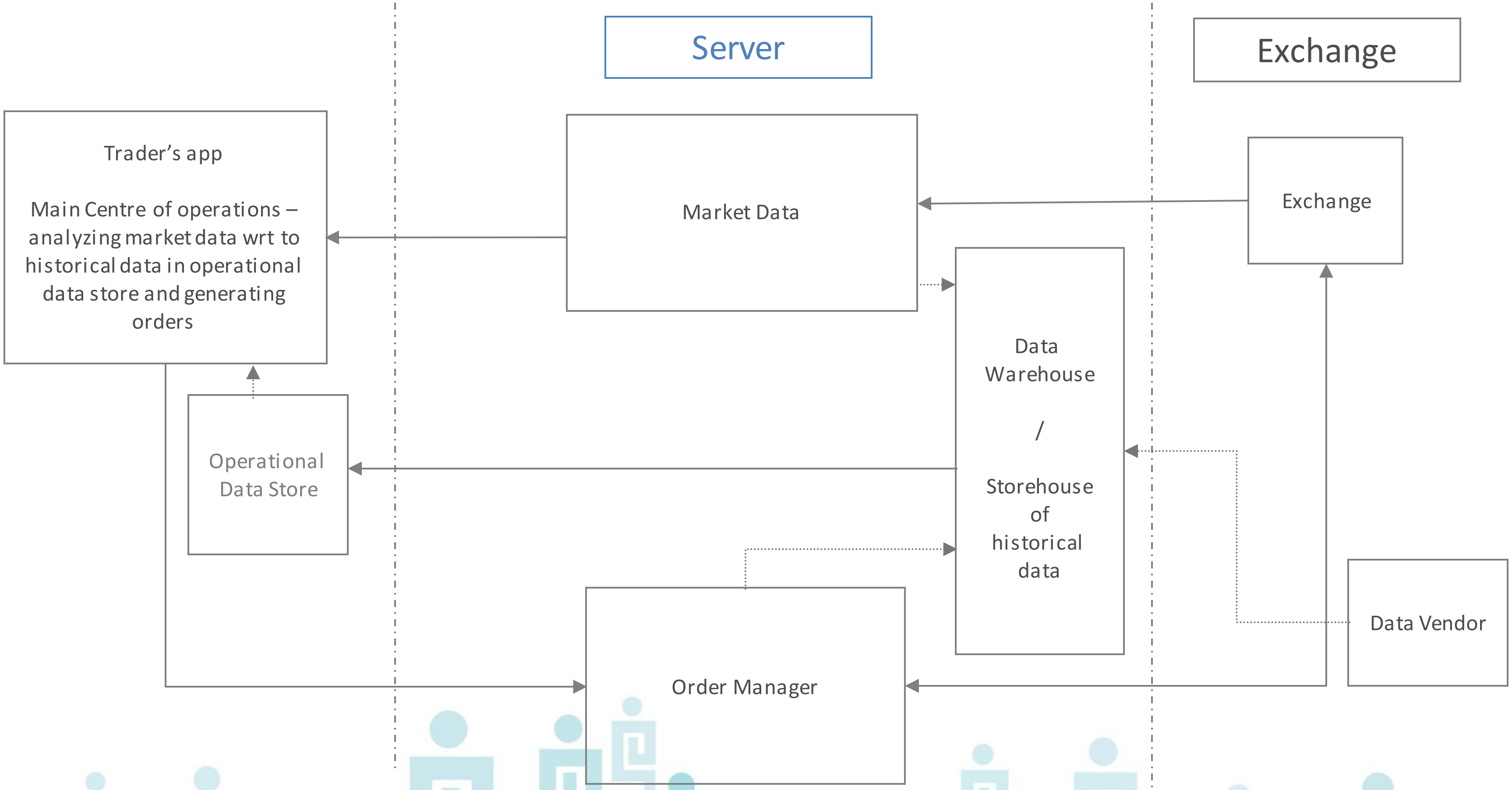
The exchange (and other data sources) – i.e. the external world



Exchange
Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Data Vendor
Order Manager

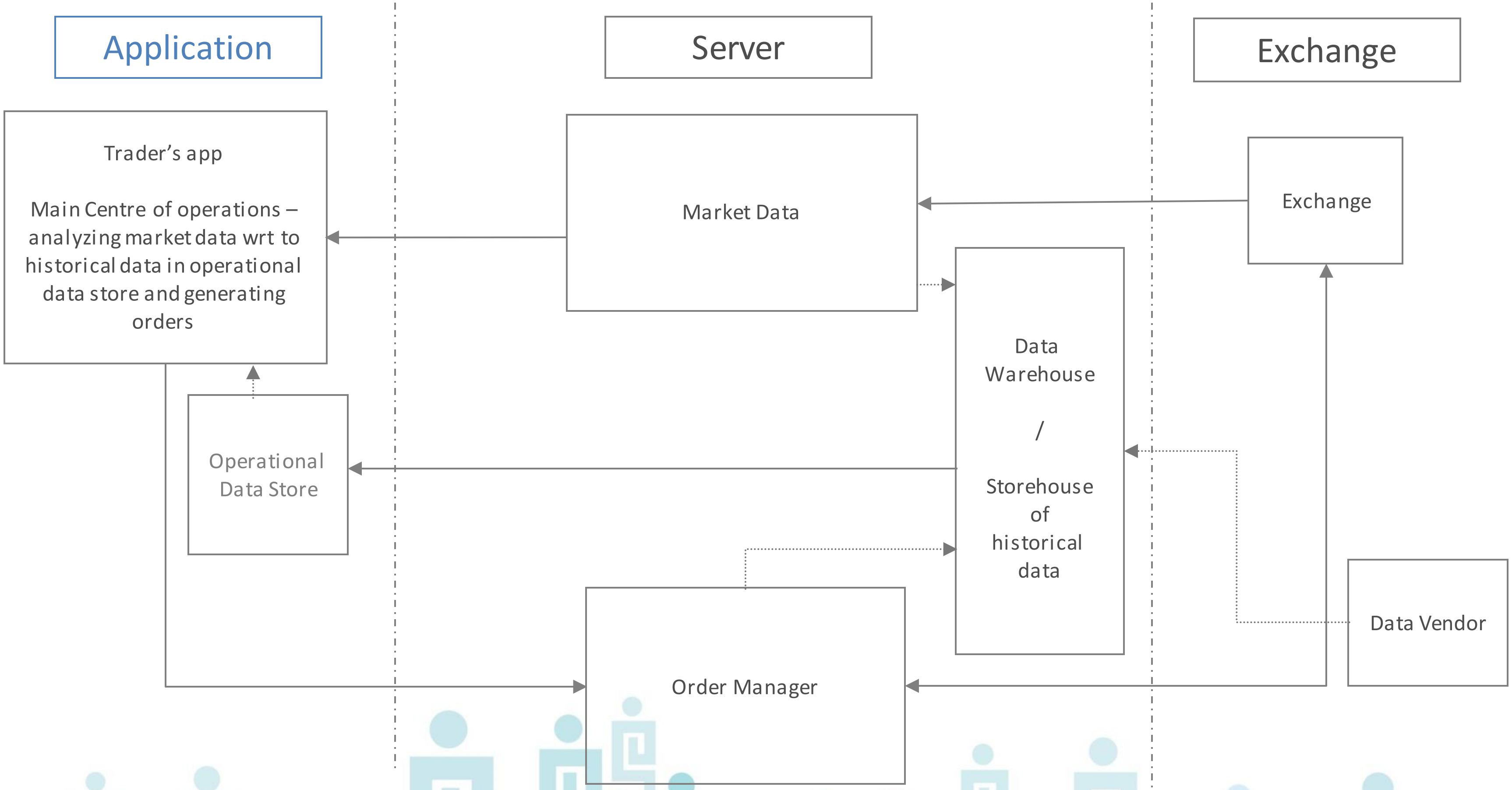
System Architecture of a Traditional Trading System

The server – a central node for understanding exchange data and forwarding to multiple trading applications. And a centralized node for handling all orders to the exchange


Server
Exchange
Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Order Manager
Data Vendor

System Architecture of a Traditional Trading System

The applications in the trader's pc which do all the processing

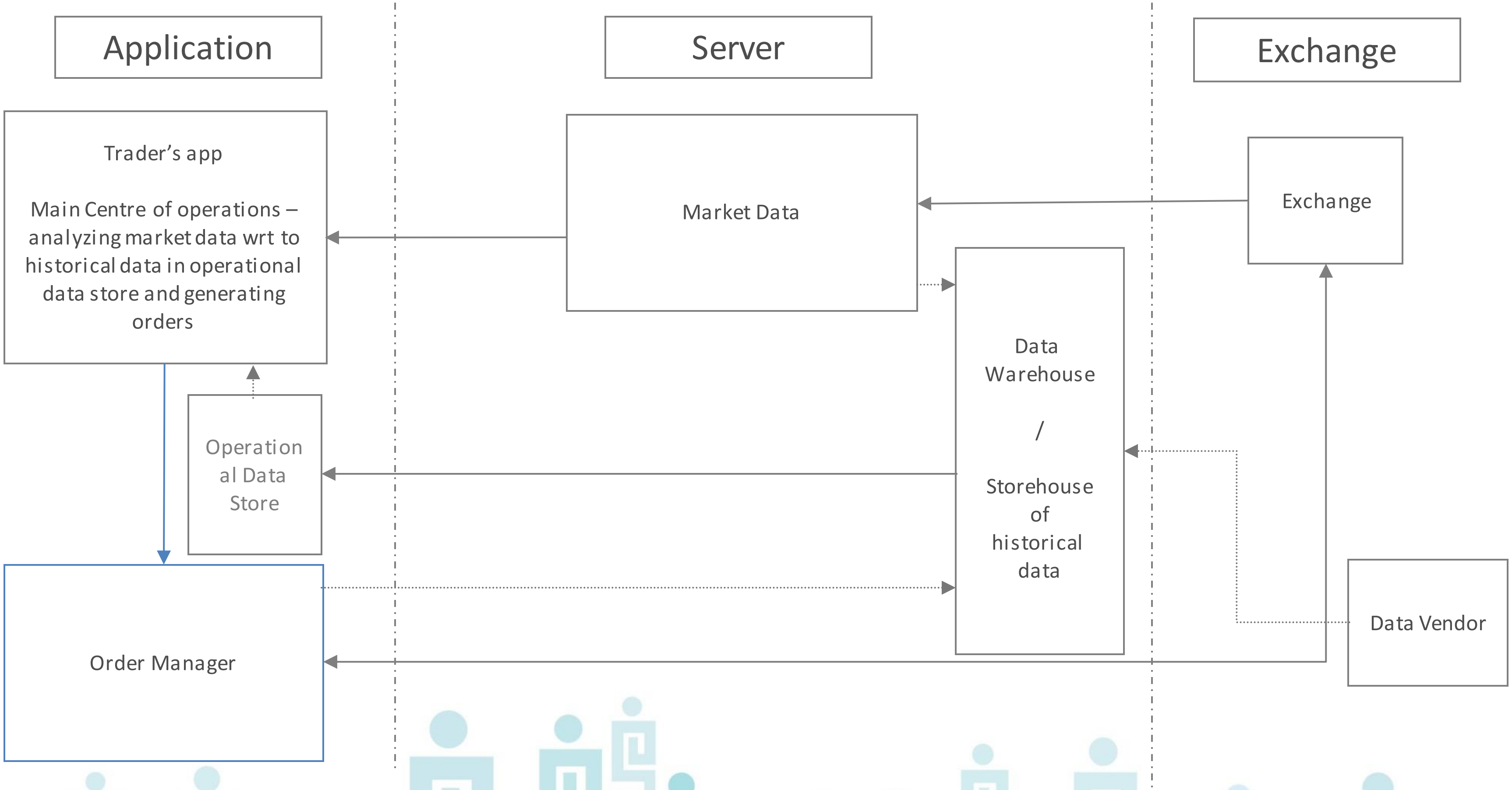


Application
Server
Exchange
Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Order Manager
Data Vendor

System Architecture of a Traditional Trading System

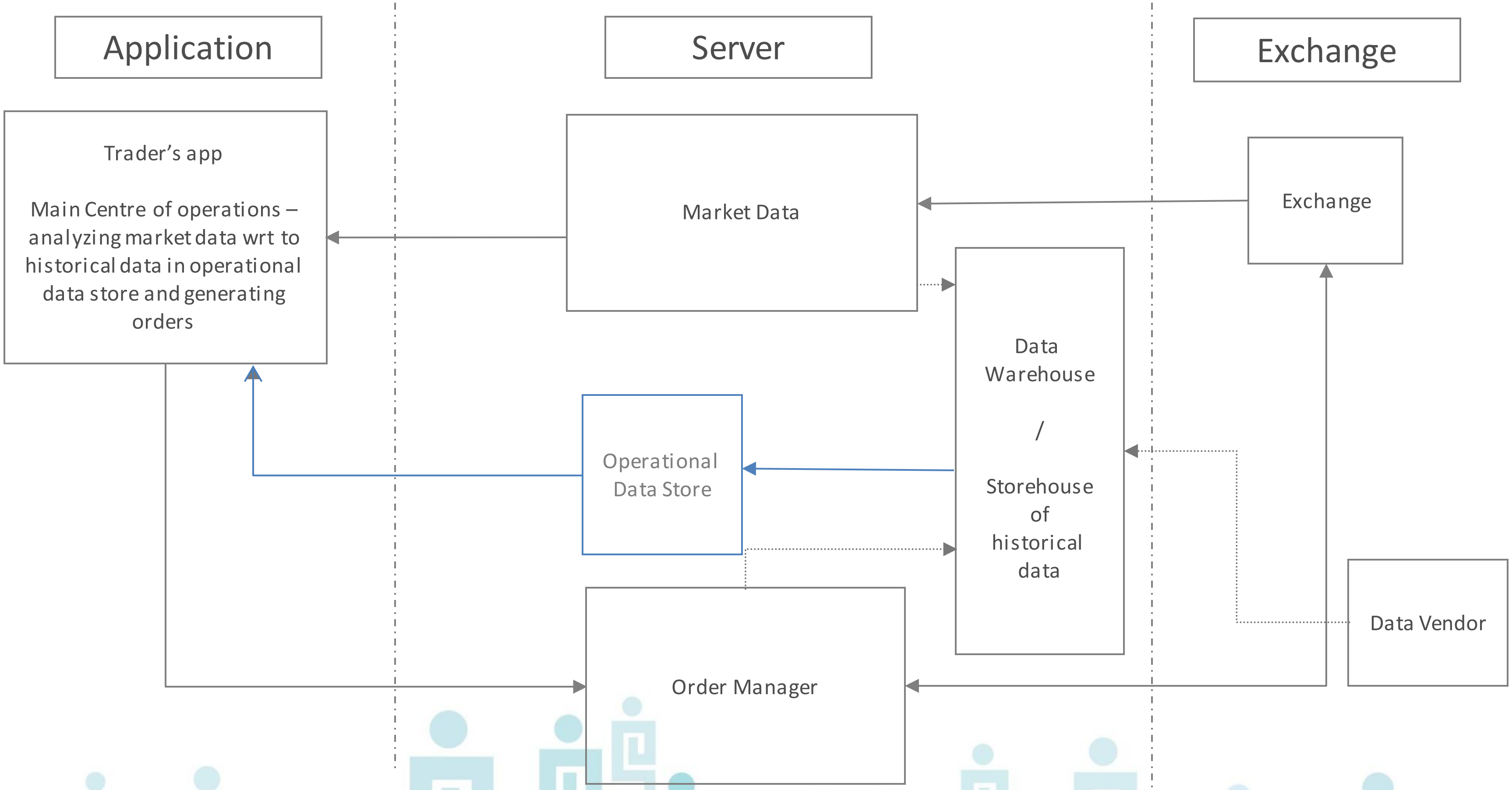
These are not rigid – but the diagram provided is a generic architecture.

The order manager could reside in the trader's application itself


Application
Server
Exchange
Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Order Manager
Data Vendor

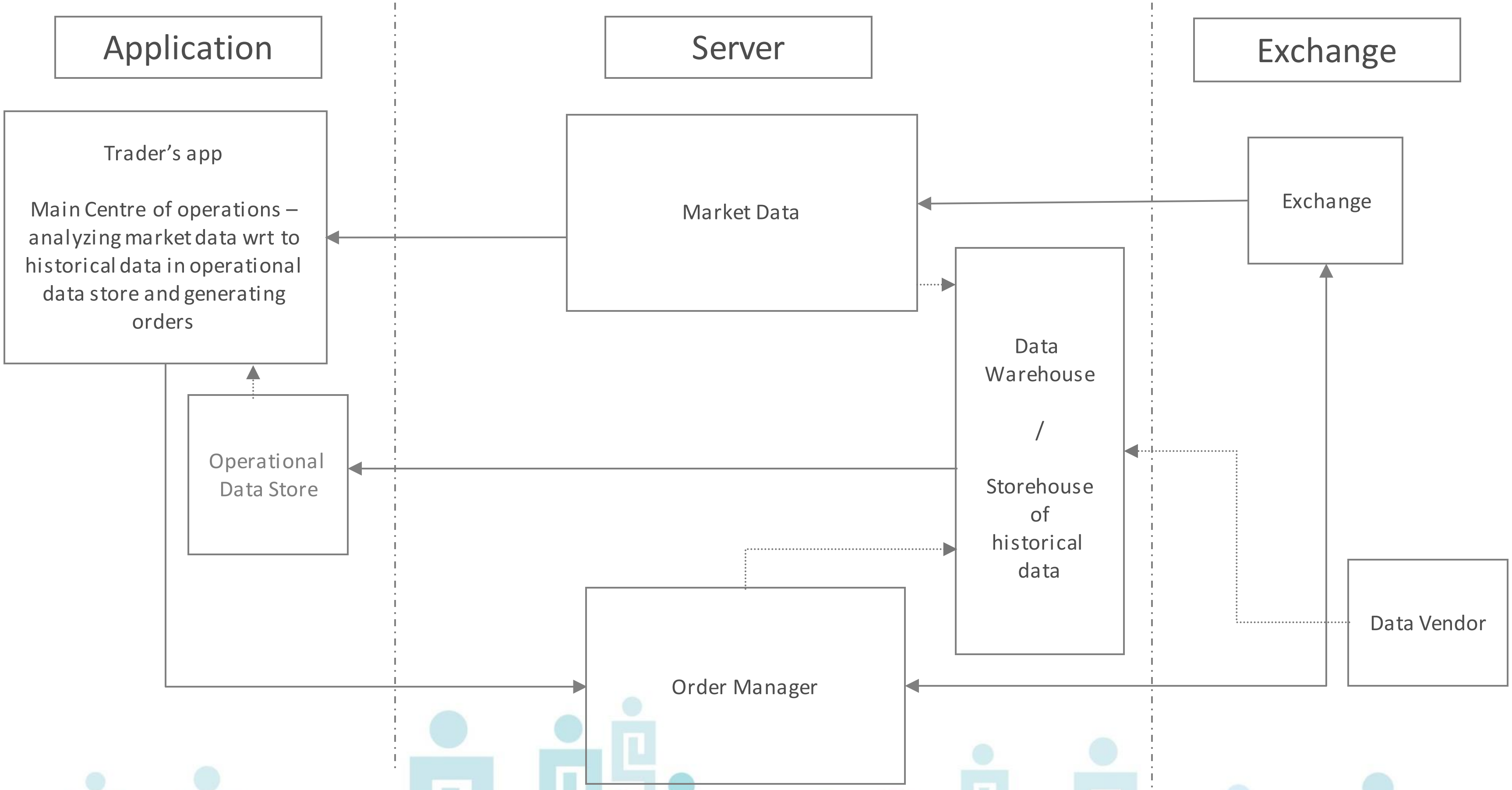
System Architecture of a Traditional Trading System

Or the operational data store could reside in centralized servers themselves


Application
Server
Exchange
Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Data Vendor
Order Manager

System Architecture of a Traditional Trading System

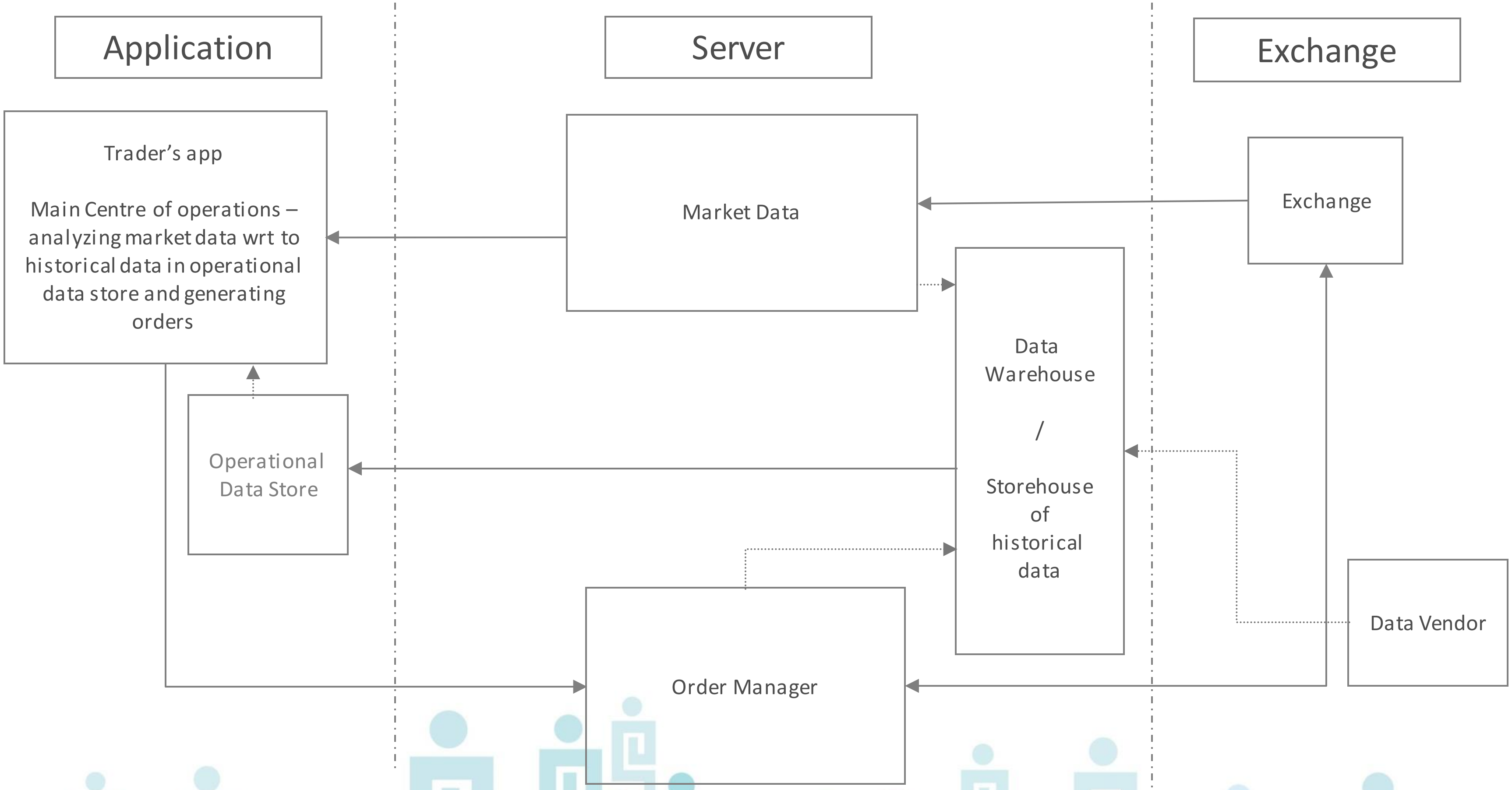
All said and done, the main decision making happens at the application level by the trader himself



Application
Server
Exchange
Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Data Vendor
Order Manager

System Architecture of a Traditional Trading System

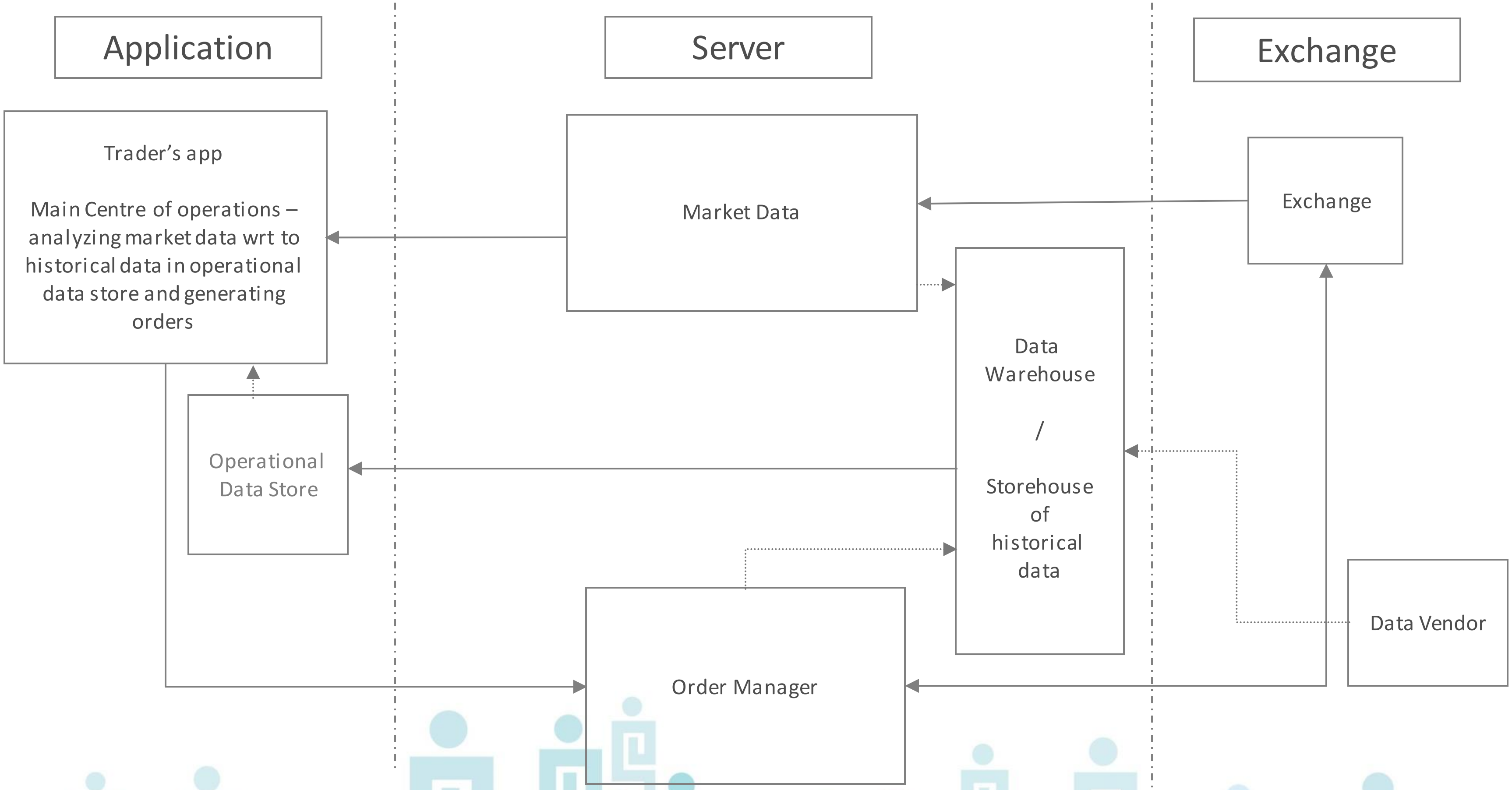
All said and done, the main decision making happens at the application level by the trader himself – who has to manually compare patterns in current exchange data to historical patterns


Application
Server
Exchange
Trader's app

Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Data Vendor
Order Manager

System Architecture of a Traditional Trading System

All said and done, the main decision making happens at the application level by the trader himself – who has to manually compare patterns in current exchange data to historical patterns, and then type his order into the system himself/herself to trade on the exchange


Application
Server
Exchange
Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Data Vendor
Order Manager

Solutions

Human Trader

Solutions

Human Trader

1 Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)

Solutions

Human Trader

1 Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)

2 Best response time is of the order of a few hundred milliseconds (0.101 second)

Solutions

Human Trader

1 Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)

2 Best response time is of the order of a few hundred milliseconds (0.101 second)

3 Can monitor market prices of upto around 50 instruments for pre-defined simple patterns

Solutions

Human Trader

1 Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)

2 Best response time is of the order of a few hundred milliseconds (0.101 second)

3 Can monitor market prices of upto around 50 instruments for pre-defined simple patterns

4 Will not be able to understand and manage risks for portfolios with hundreds / thousands of positions

Solutions

Human Trader

1 Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)

2 Best response time is of the order of a few hundred milliseconds (0.101 second)

3 Can monitor market prices of upto around 50 instruments for pre-defined simple patterns

4 Will not be able to understand and manage risks for portfolios with hundreds / thousands of positions

5 Have to type order details with great precision (and thus stress) to ensure 'typos' don't cause wrong trades

Solutions

Human Trader | **Algorithmic Trading System**

1 Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)

2 Best response time is of the order of a few hundred milliseconds (0.101 second)

3 Can monitor market prices of upto around 50 instruments for pre-defined simple patterns

4 Will not be able to understand and manage risks for portfolios with hundreds / thousands of positions

5 Have to type order details with great precision (and thus stress) to ensure 'typos' don't cause wrong trades

Solutions

Human Trader

Algorithmic Trading System

1 Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)

Have near 100% uptime.
Without fatigue.

2 Best response time is of the order of a few hundred milliseconds (0.101 second)

3 Can monitor market prices of upto around 50 instruments for pre-defined simple patterns

4 Will not be able to understand and manage risks for portfolios with hundreds / thousands of positions

5 Have to type order details with great precision (and thus stress) to ensure ‘typos’ don’t cause wrong trades

Solutions

Human Trader

1 Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)

2 Best response time is of the order of a few hundred milliseconds (0.101 second)

3 Can monitor market prices of upto around 50 instruments for pre-defined simple patterns

4 Will not be able to understand and manage risks for portfolios with hundreds / thousands of positions

5 Have to type order details with great precision (and thus stress) to ensure 'typos' don't cause wrong trades

Algorithmic Trading System

Have near 100% uptime.
Without fatigue.

Can respond to opportunities in microseconds (0.000001 second). Including 'short lived opportunities'

Solutions

Human Trader

1 Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)

2 Best response time is of the order of a few hundred milliseconds (0.101 second)

3 Can monitor market prices of upto around 50 instruments for pre-defined simple patterns

4 Will not be able to understand and manage risks for portfolios with hundreds / thousands of positions

5 Have to type order details with great precision (and thus stress) to ensure ‘typos’ don’t cause wrong trades

Algorithmic Trading System

Have near 100% uptime. Without fatigue.

Can respond to opportunities in microseconds (0.000001 second). Including ‘short lived opportunities’

Can monitor prices of tens of thousands of instruments in parallel (for complex patterns)

Solutions

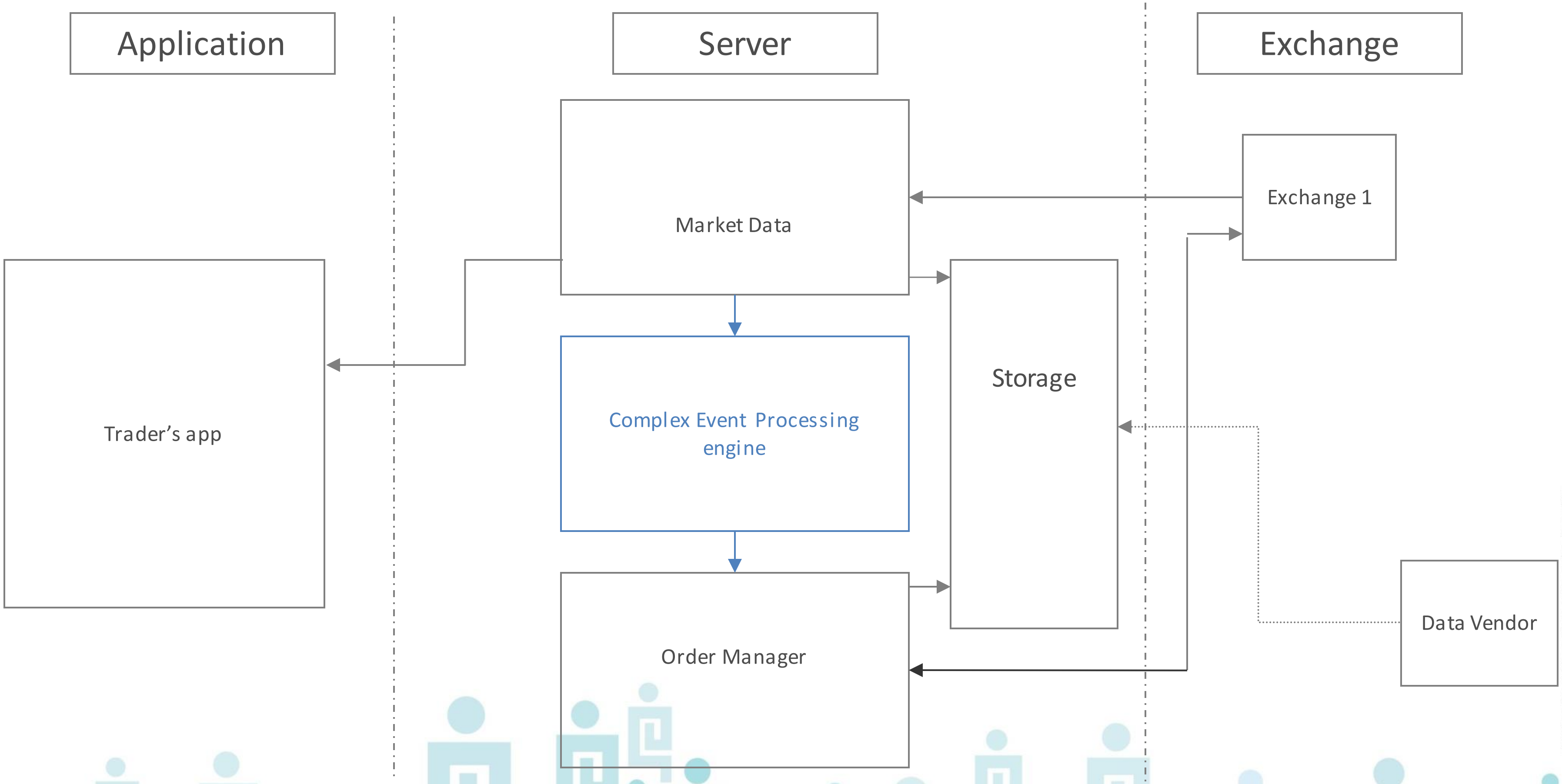
	Human Trader	Algorithmic Trading System
1	Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)	Have near 100% uptime. Without fatigue.
2	Best response time is of the order of a few hundred milliseconds (0.101 second)	Can respond to opportunities in microseconds (0.000001 second). Including 'short lived opportunities'
3	Can monitor market prices of upto around 50 instruments for pre-defined simple patterns	Can monitor prices of tens of thousands of instruments in parallel (for complex patterns)
4	Will not be able to understand and manage risks for portfolios with hundreds / thousands of positions	Can manage portfolios with positions in thousands of instruments in parallel.
5	Have to type order details with great precision (and thus stress) to ensure 'typos' don't cause wrong trades	

Solutions

	Human Trader	Algorithmic Trading System
1	Have to be at their trading desk (and not on vacation; or lunch / toilet breaks)	Have near 100% uptime. Without fatigue.
2	Best response time is of the order of a few hundred milliseconds (0.101 second)	Can respond to opportunities in microseconds (0.000001 second). Including 'short lived opportunities'
3	Can monitor market prices of upto around 50 instruments for pre-defined simple patterns	Can monitor prices of tens of thousands of instruments in parallel (for complex patterns)
4	Will not be able to understand and manage risks for portfolios with hundreds / thousands of positions	Can manage portfolios with positions in thousands of instruments in parallel.
5	Have to type order details with great precision (and thus stress) to ensure 'typos' don't cause wrong trades	If properly tested, ATS will send millions of logically sound orders in a day without typos

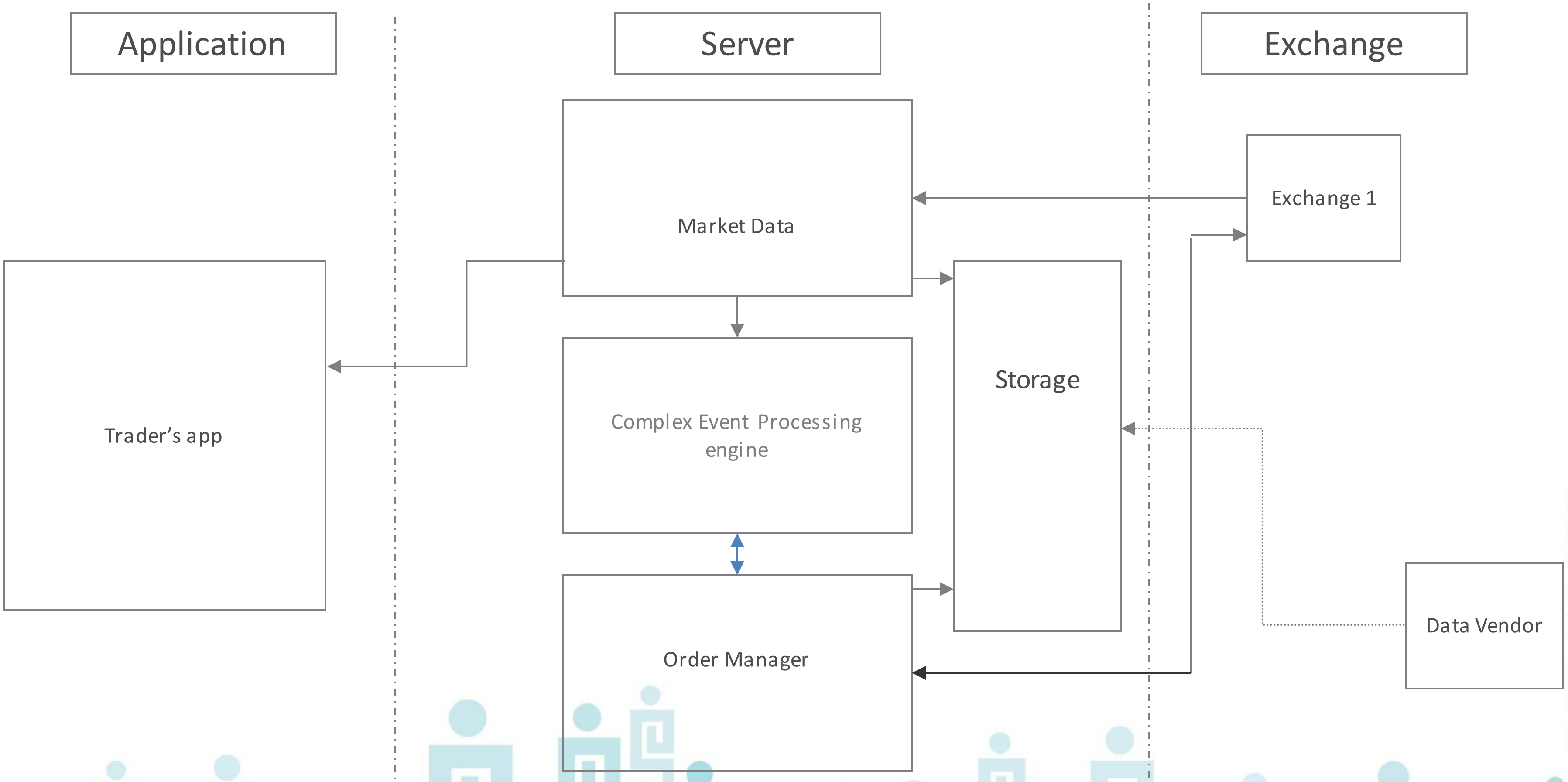
System Architecture of an Automated Trading System

The first step was to automate the monitoring of prices, and automate the trading decisions, and automate the executing of orders – this is handled in the CEP



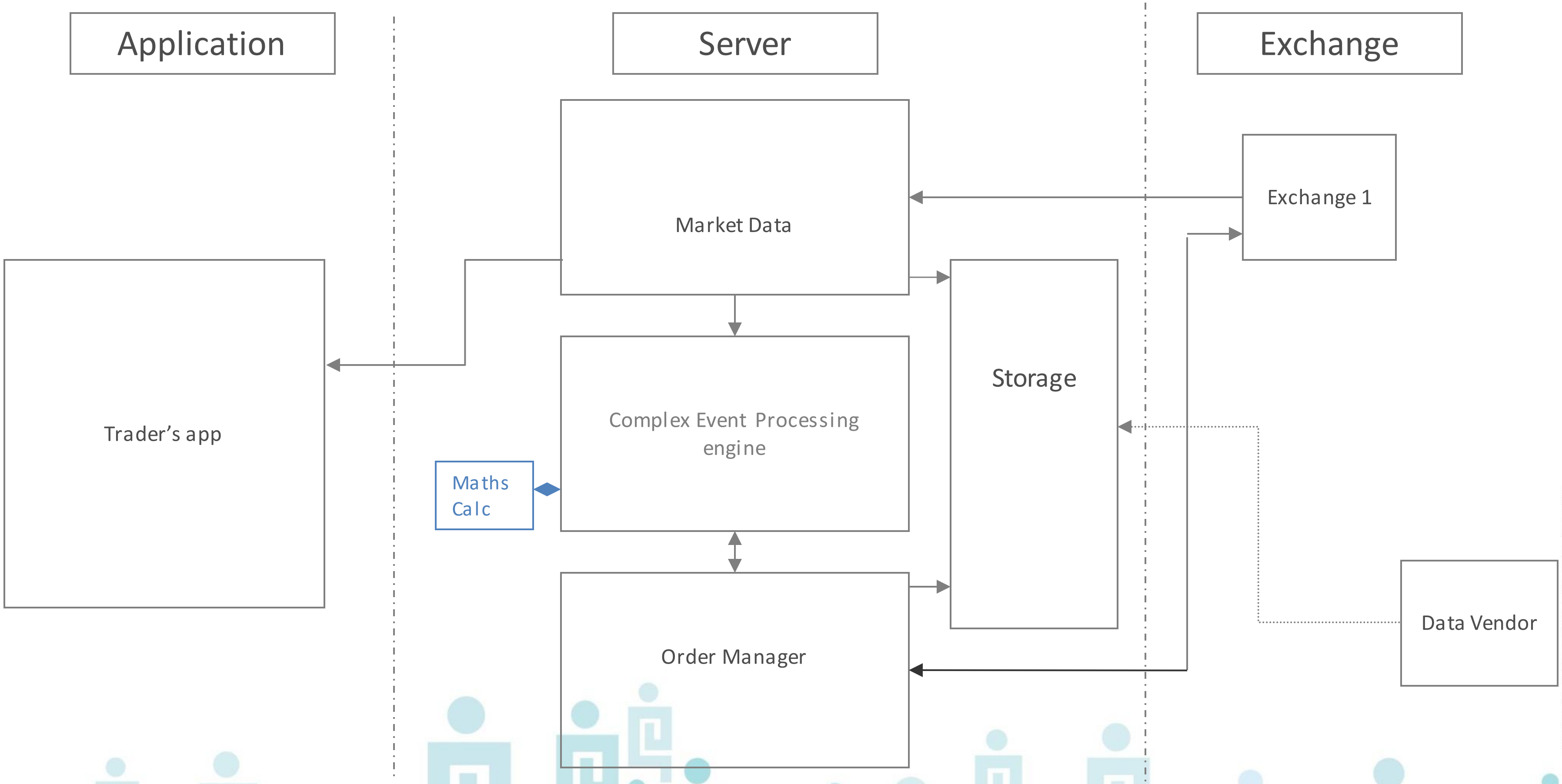
System Architecture of an Automated Trading System

The CEP gets a feedback about orders, current positions and executions. It can use the same information for making trading decisions.



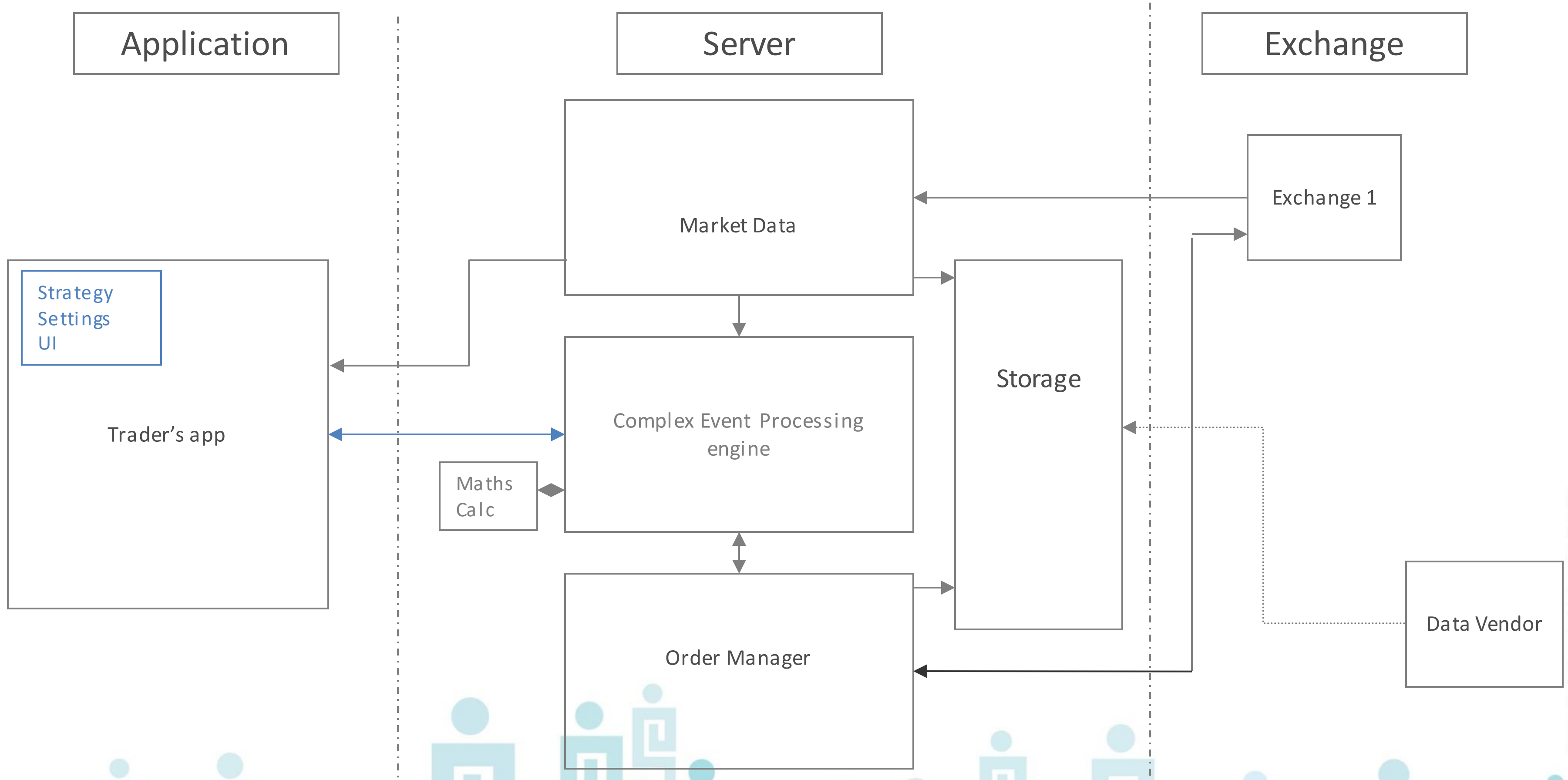
System Architecture of an Automated Trading System

Complex mathematical operations are handled in a dedicated calculation block in the server block (e.g. Options Greeks calculations)



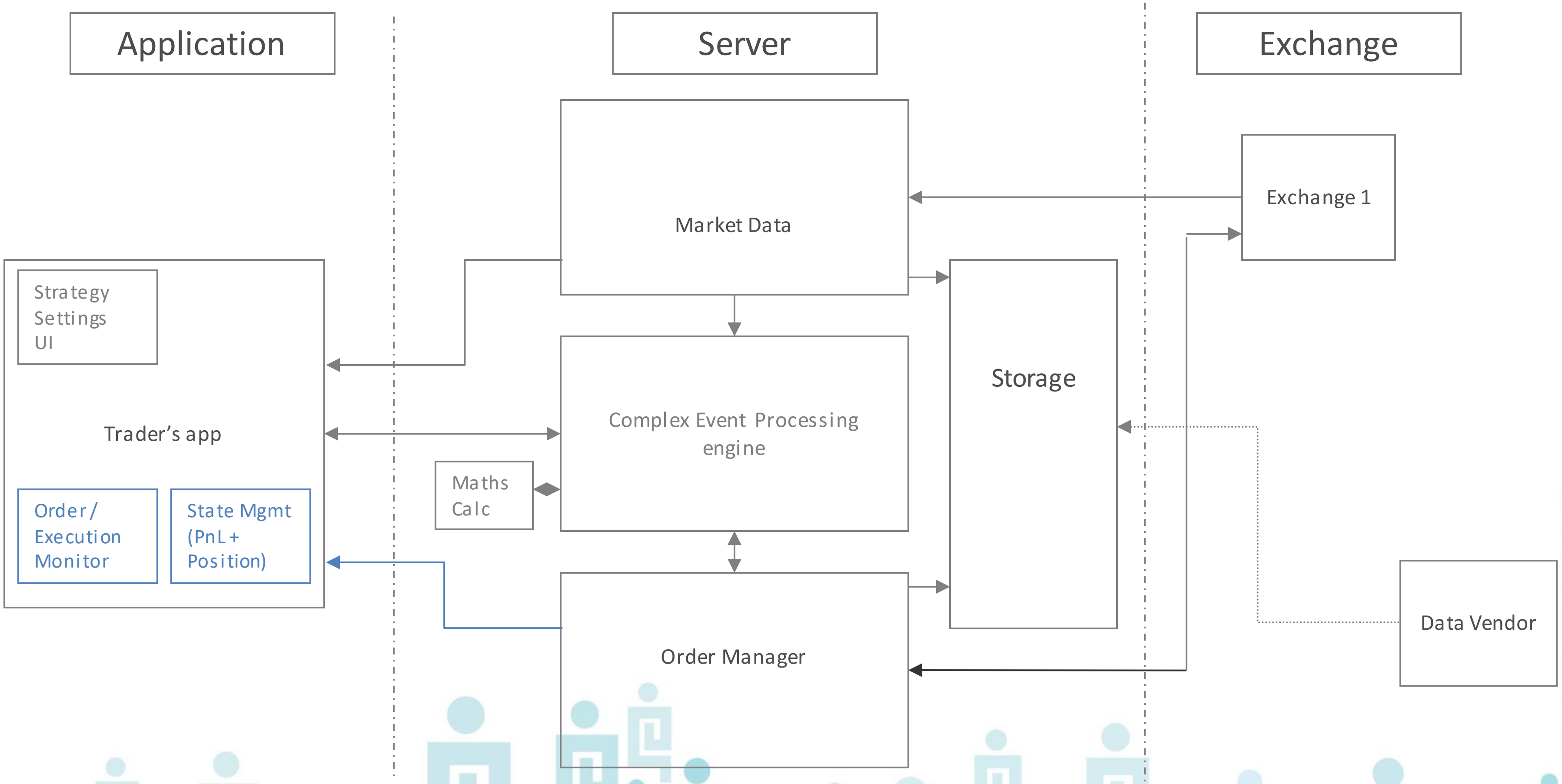
System Architecture of an Automated Trading System

The role of the application layer has reduced drastically – (i) an input screen for strategy settings



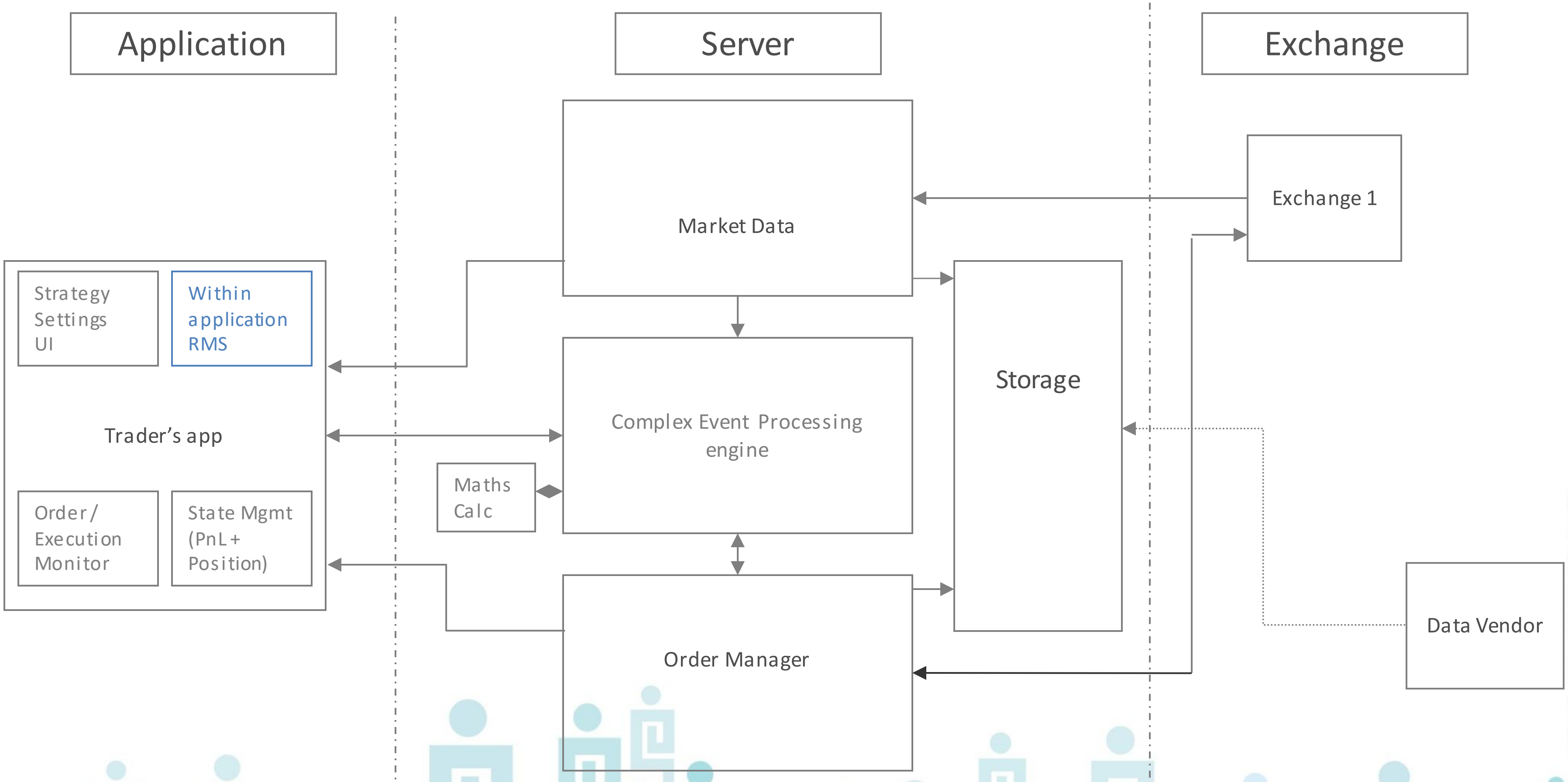
System Architecture of an Automated Trading System

The role of the application layer has reduced drastically – (ii) monitor of system state, i.e. orders, executions and positions



System Architecture of an Automated Trading System

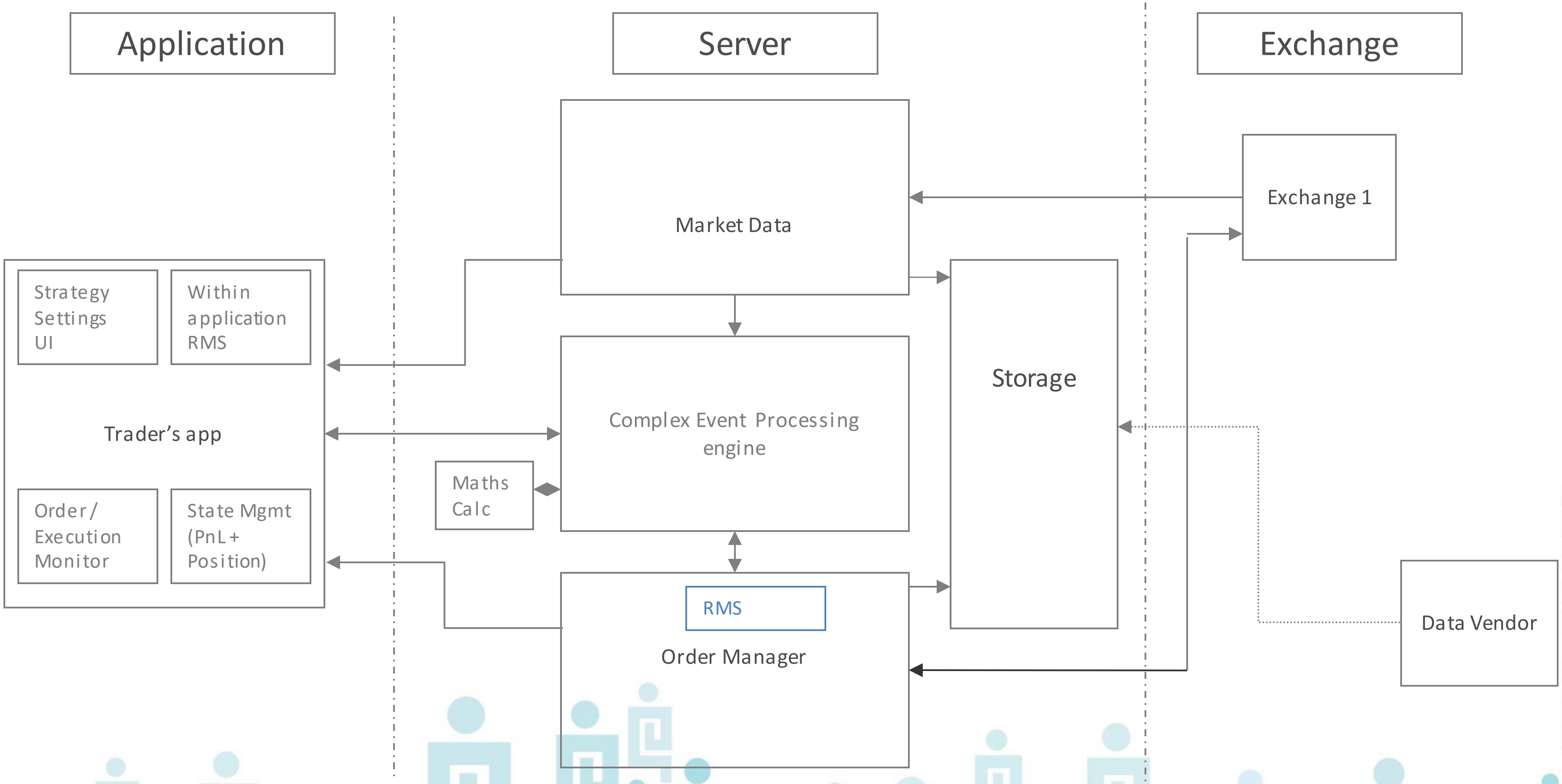
The role of the application layer has reduced drastically – (iii) preliminary fat finger RMS checker (RMS = Risk Management System)



© Quantinsti Quantitative Learning Pvt. Ltd.

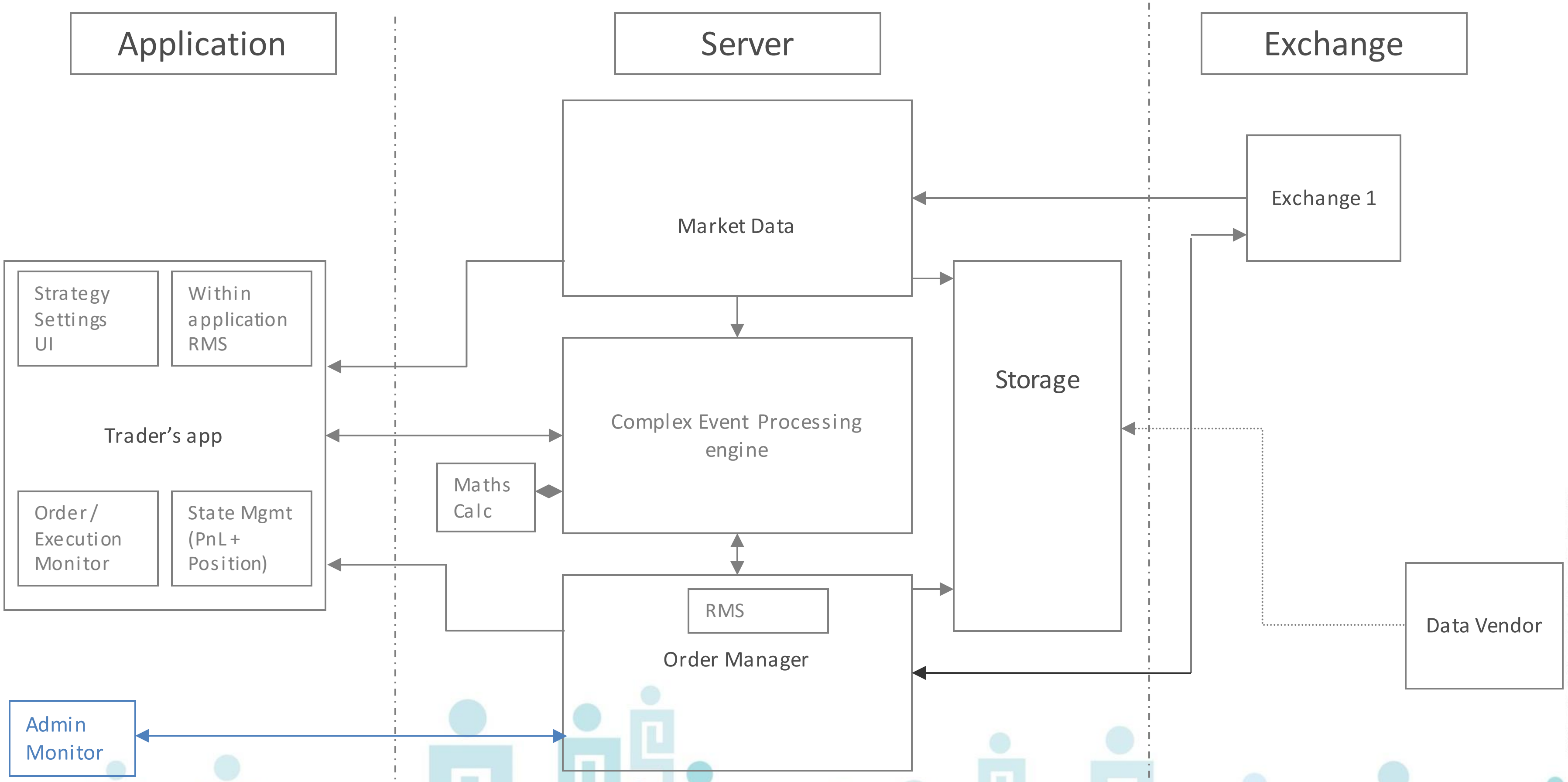
System Architecture of an Automated Trading System

RMS is now automated and checked by the OMS before an order is generated



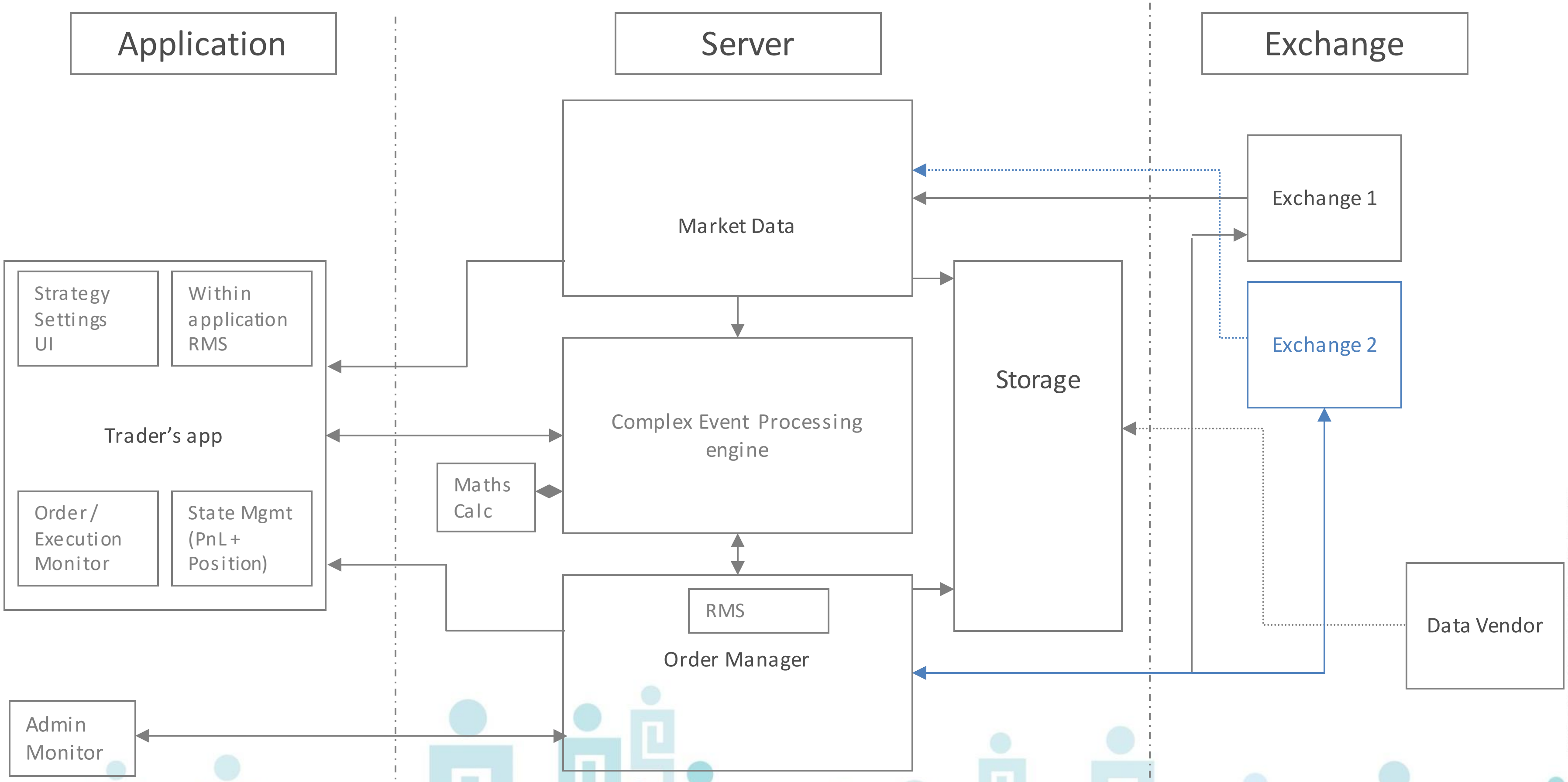
System Architecture of an Automated Trading System

Because RMS is automated, a second level of monitoring is necessary – an overall global position monitor



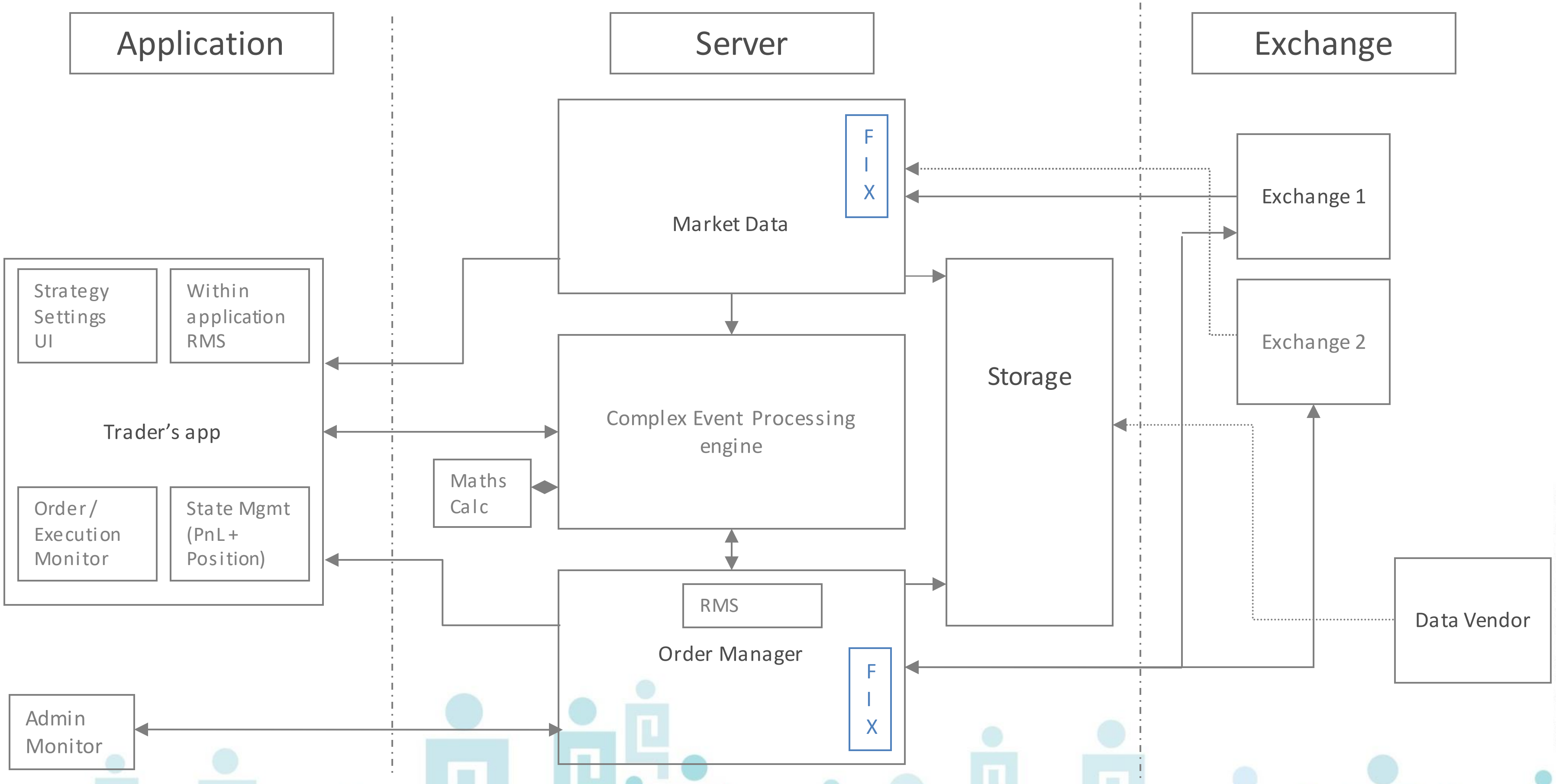
System Architecture of an Automated Trading System

Since scaling up is now possible, the systems are now connected to multiple exchanges



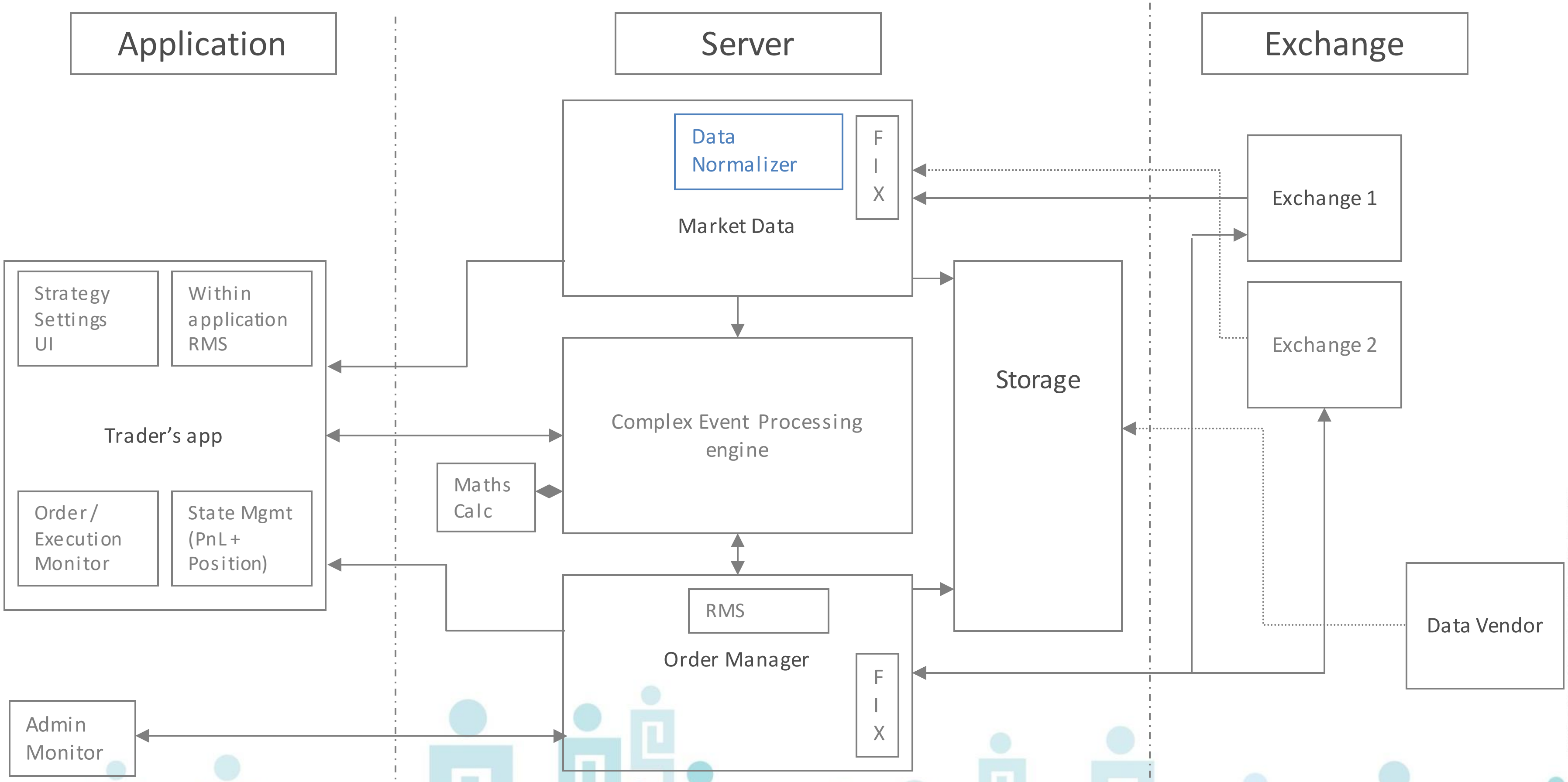
System Architecture of an Automated Trading System

To make it easier to connect to new exchanges, standardized protocols like FIX became the norm



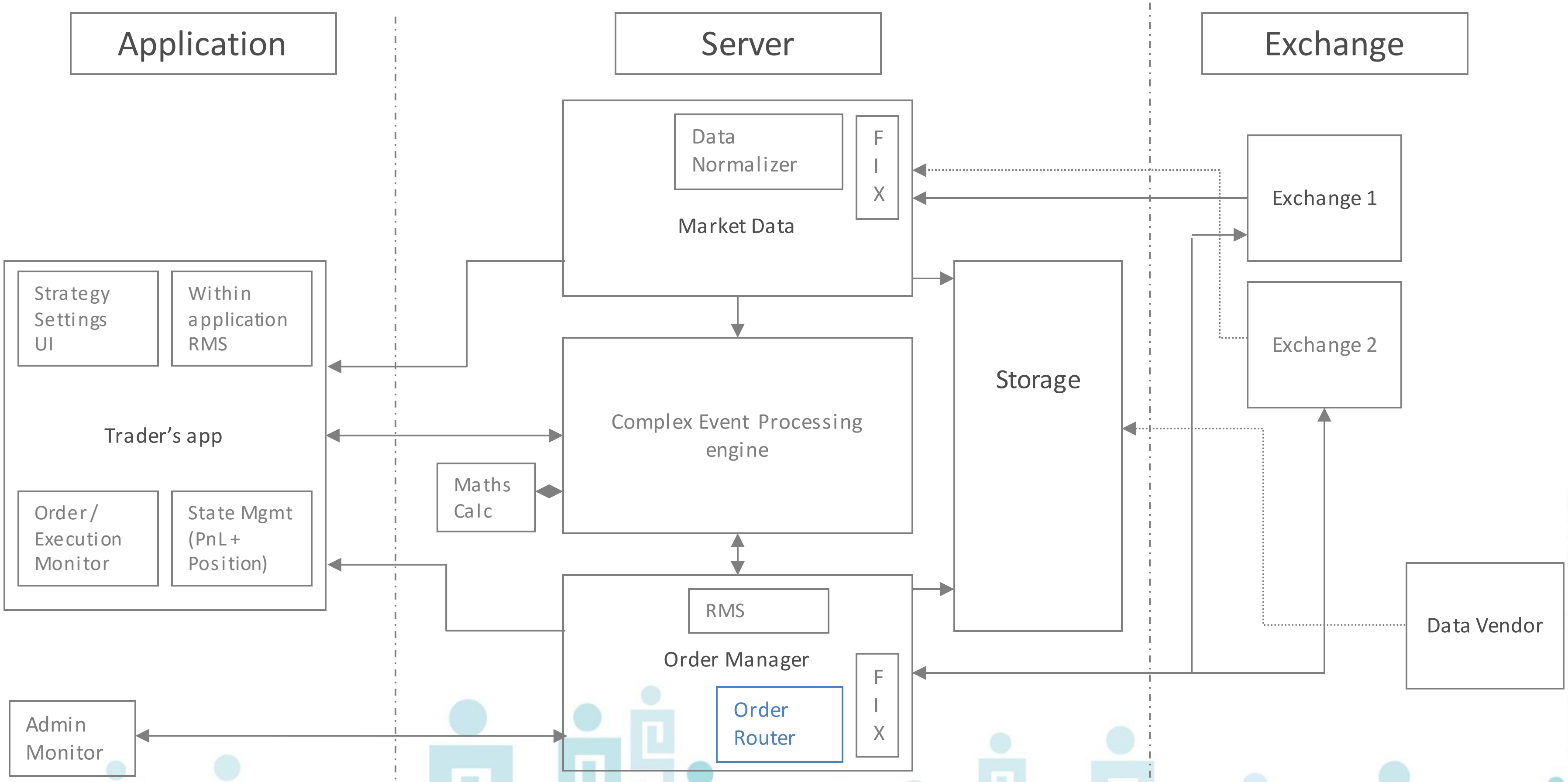
System Architecture of an Automated Trading System

This also necessitated adding data normalizer block in the market data adaptors to convert data from multiple exchanges into a standard format



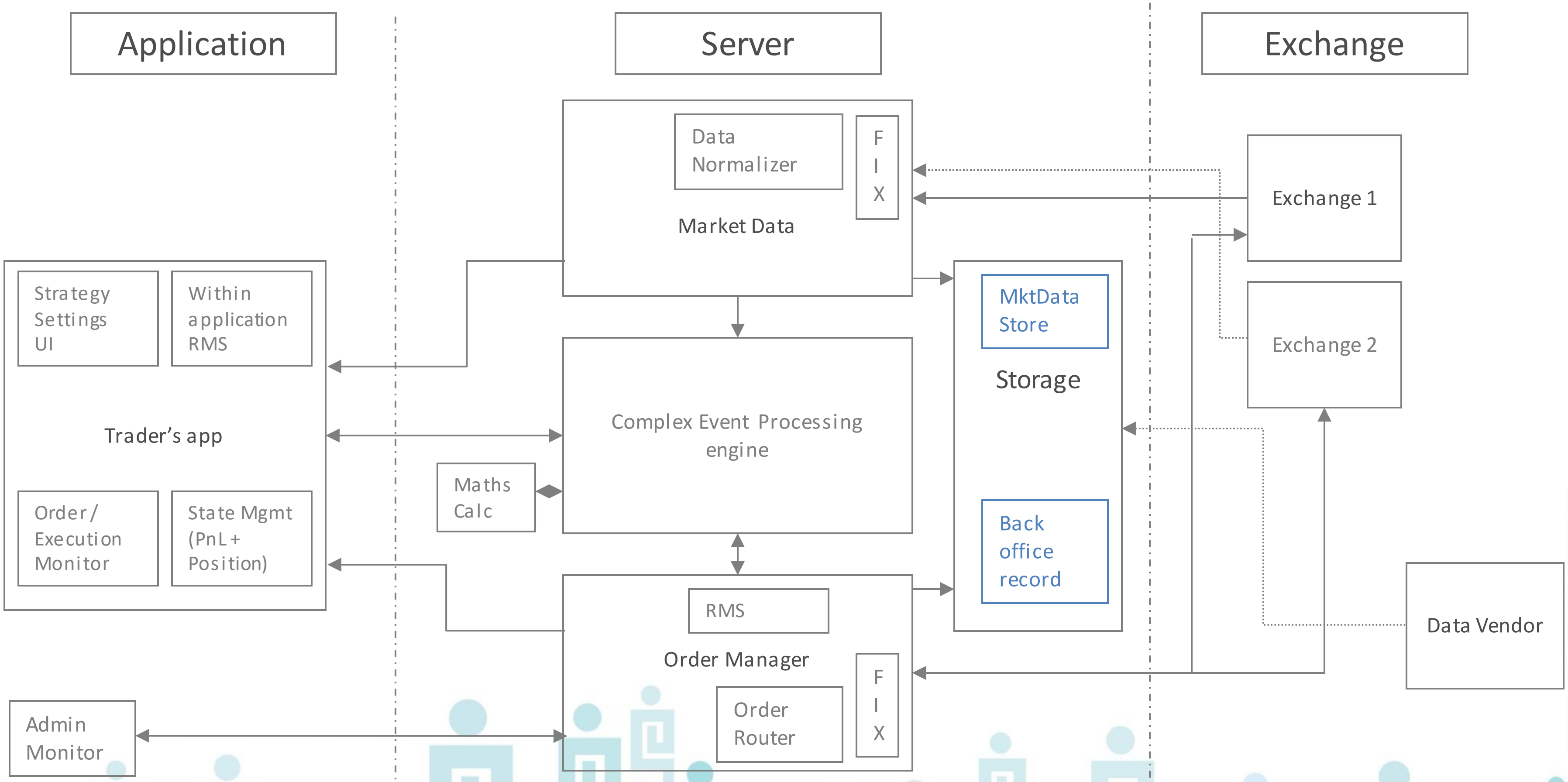
System Architecture of an Automated Trading System

Moreover, an Order Router had to be added to the OMS to route orders from the same OMS to multiple exchanges



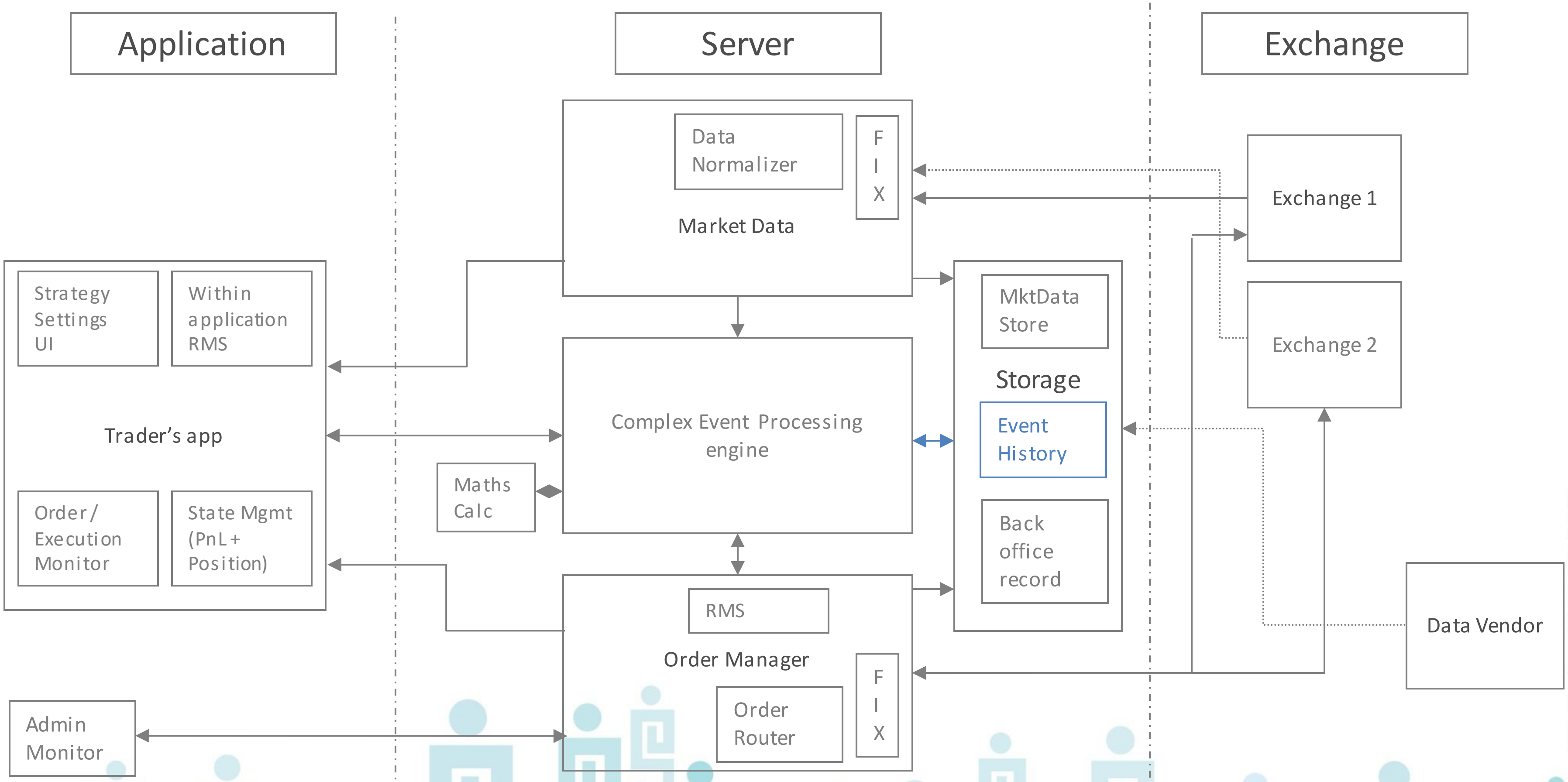
System Architecture of an Automated Trading System

Regulatory requirements have complicated storage requirements – requiring storage of trade information (in addition to market data that firms were storing for in-house use)



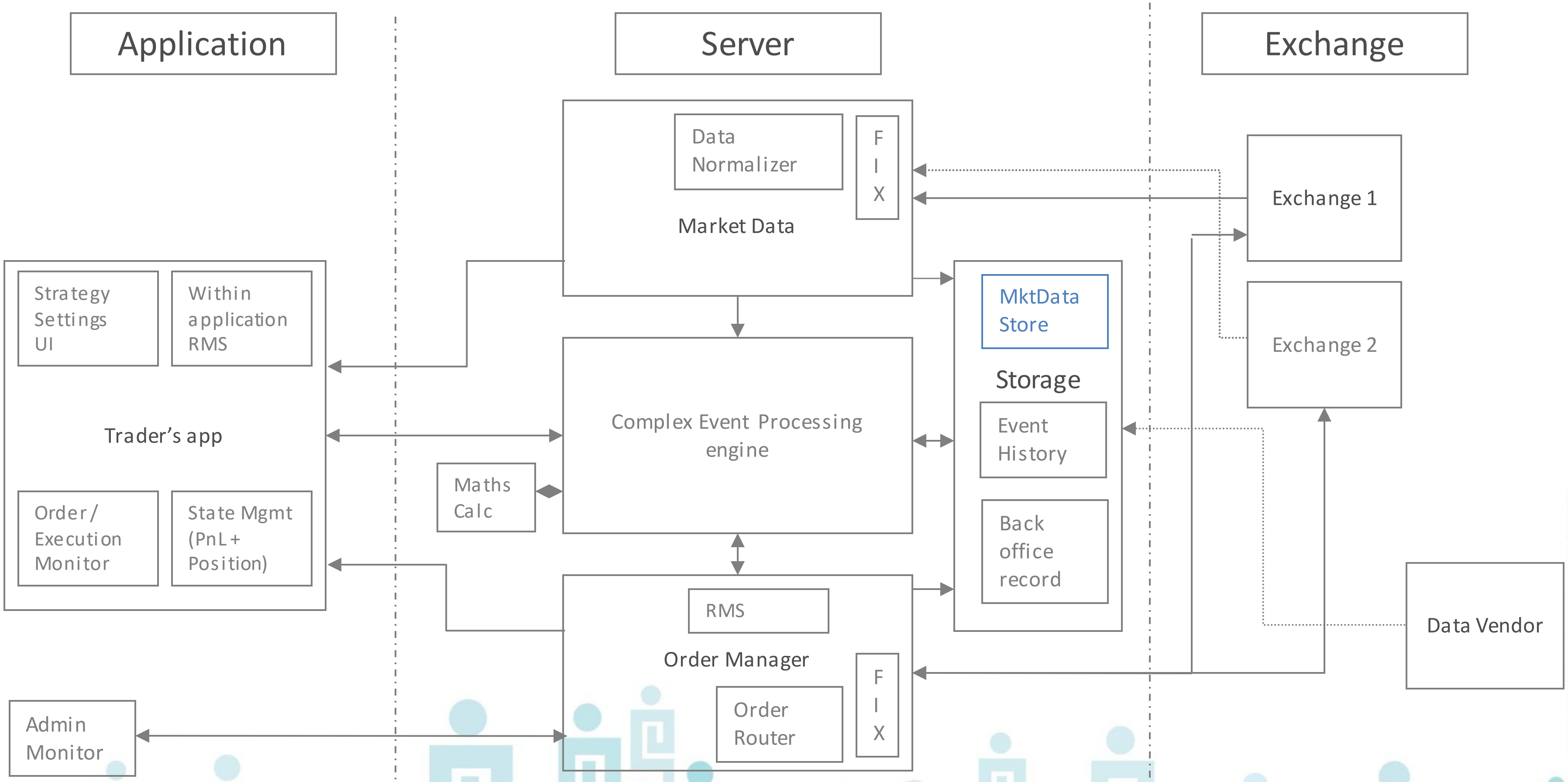
System Architecture of an Automated Trading System

Moreover, the CEP engine has its own storage requirements of event history to identify future opportunities (without doing entire re-calculations)



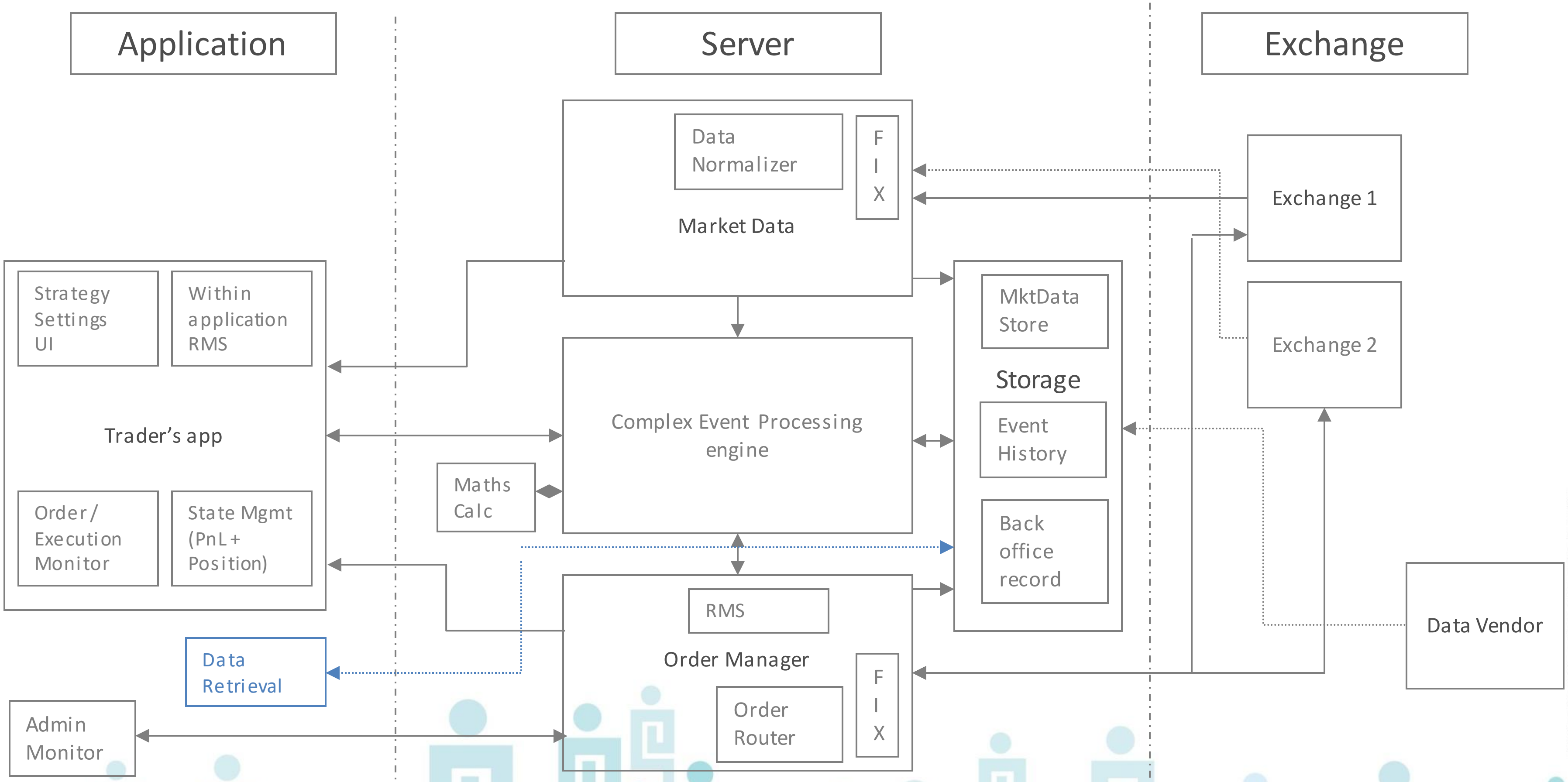
System Architecture of an Automated Trading System

The amount of data in the market data has also gone up drastically – because market participants now respond to micro events and to more events



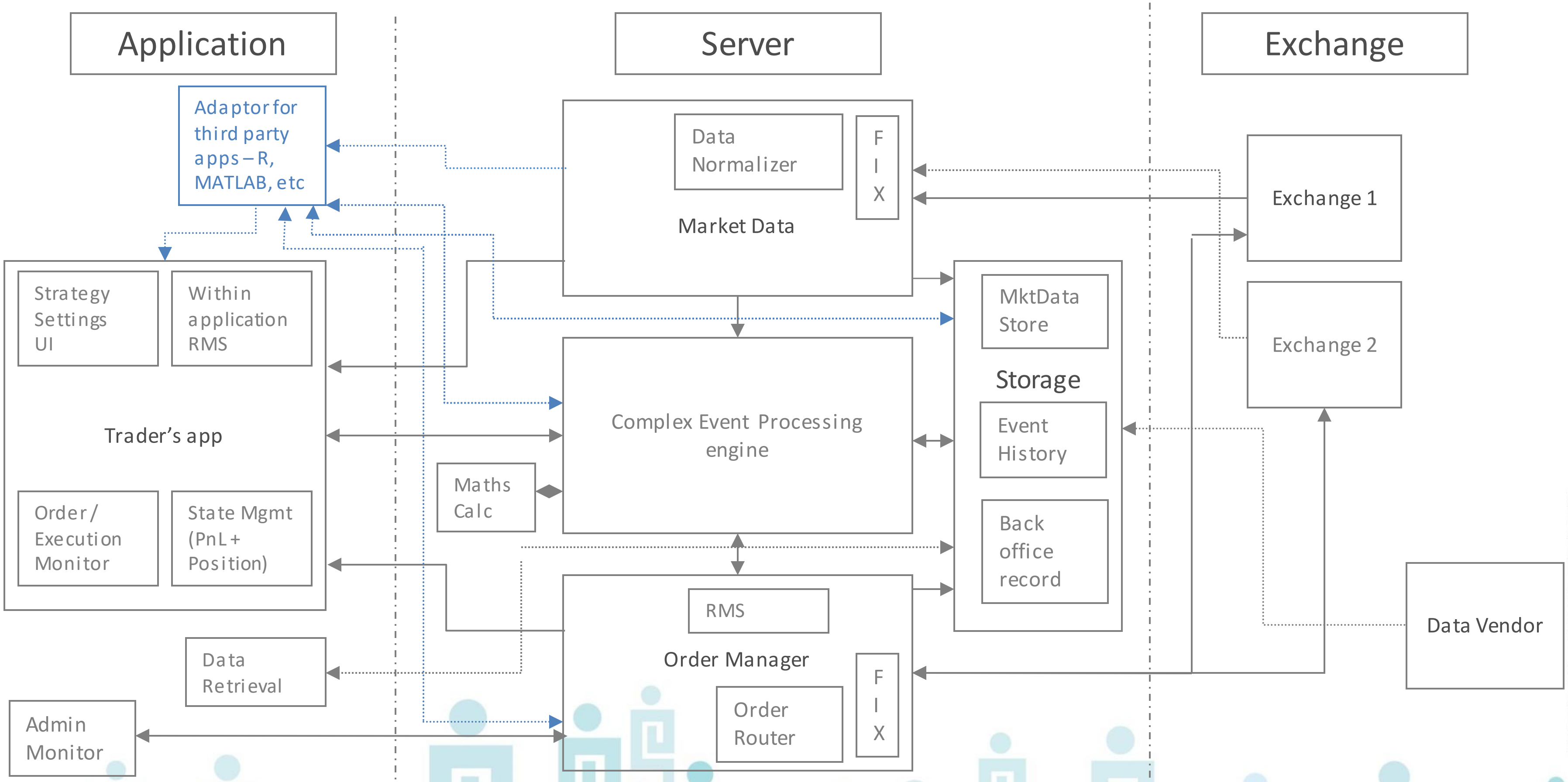
System Architecture of an Automated Trading System

With increase in complexity of the data, the sophistication of the data tools have gone up.



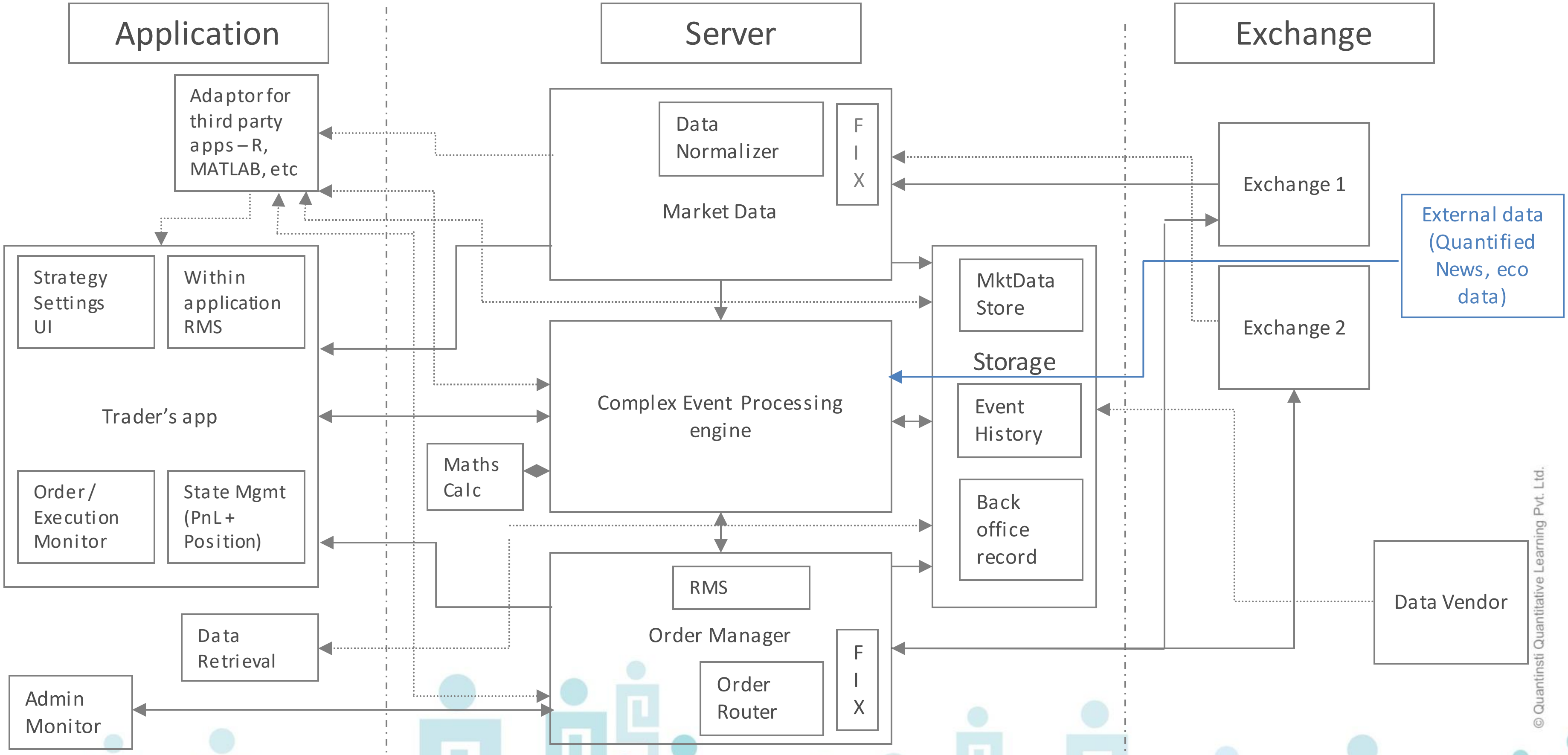
System Architecture of an Automated Trading System

Third party data analytical applications have to be tightly integrated with all the blocks



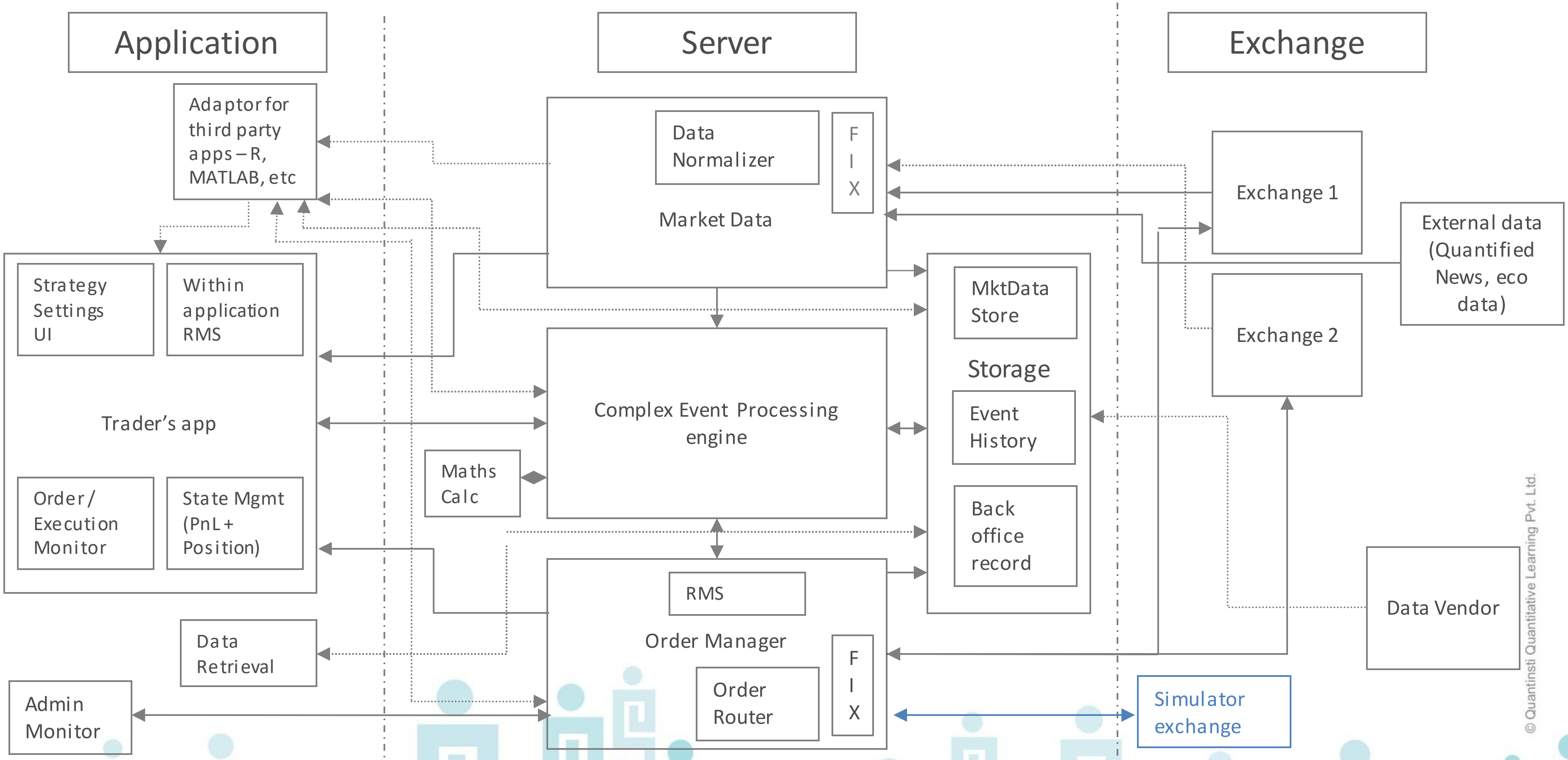
System Architecture of an Automated Trading System

Inputs could also be diverse and not just limited to exchange market data. These could be quantified news scores, or economic/earnings data in standardized format. The external data may be integrated with the market data server or may directly be consumed by the CEP engine.



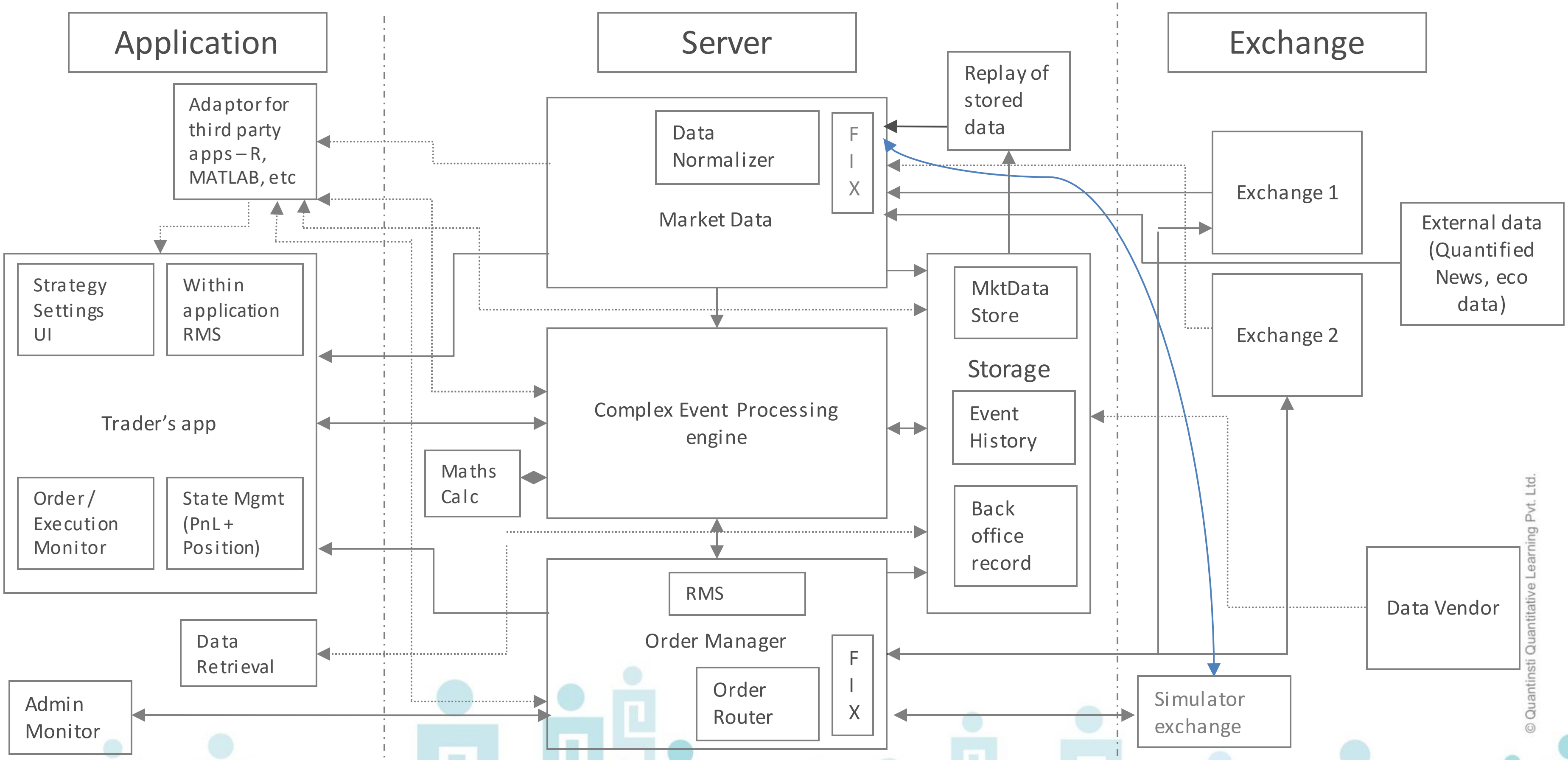
System Architecture of an Automated Trading System

To validate the correctness of implementation an algorithmic trading strategy, simulators were added which would simulate the behavior of exchanges on receiving market data



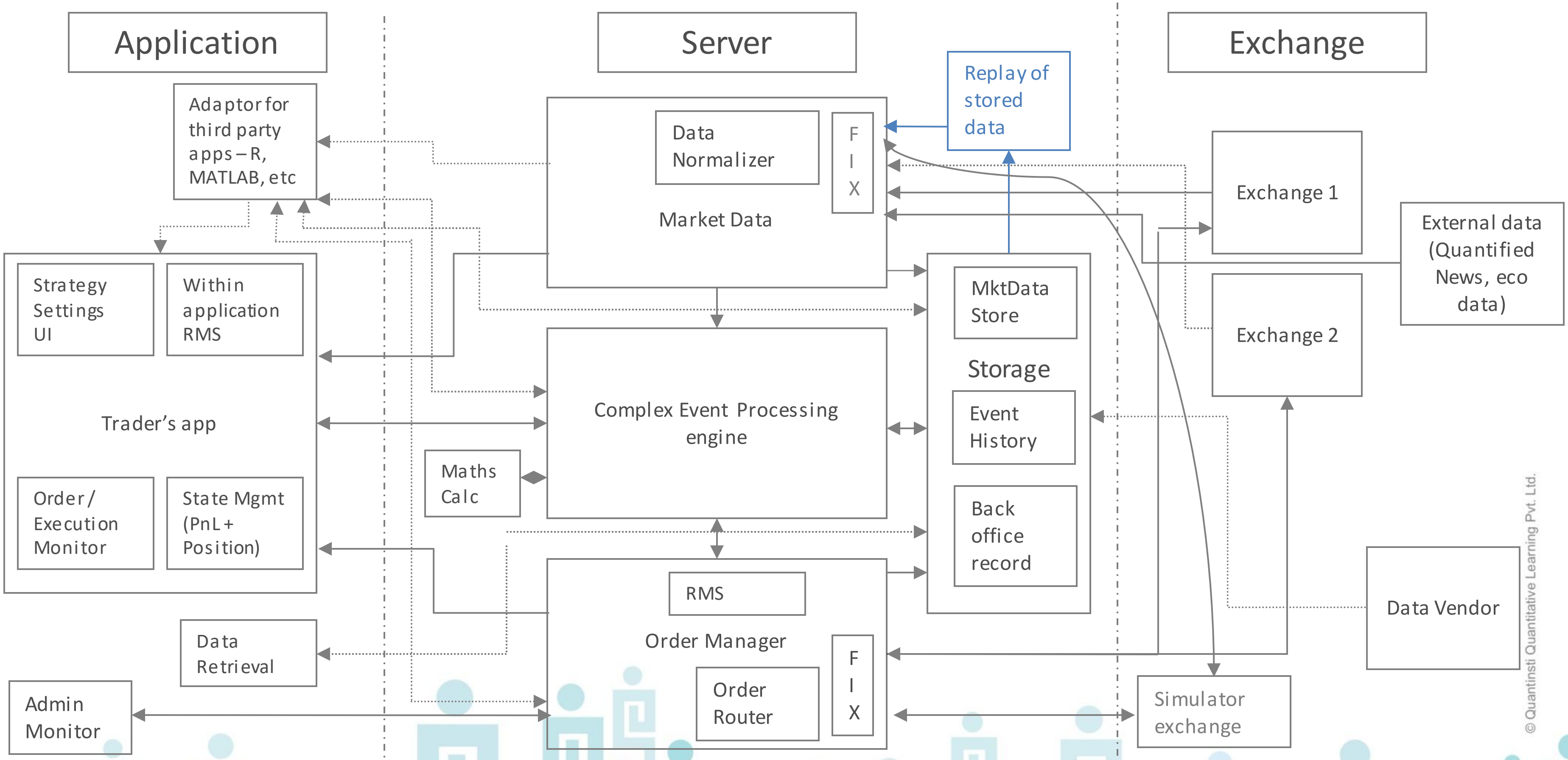
System Architecture of an Automated Trading System

Simulators should also be able to see the current state of the market data to be able to simulate the exchange behavior most accurately.



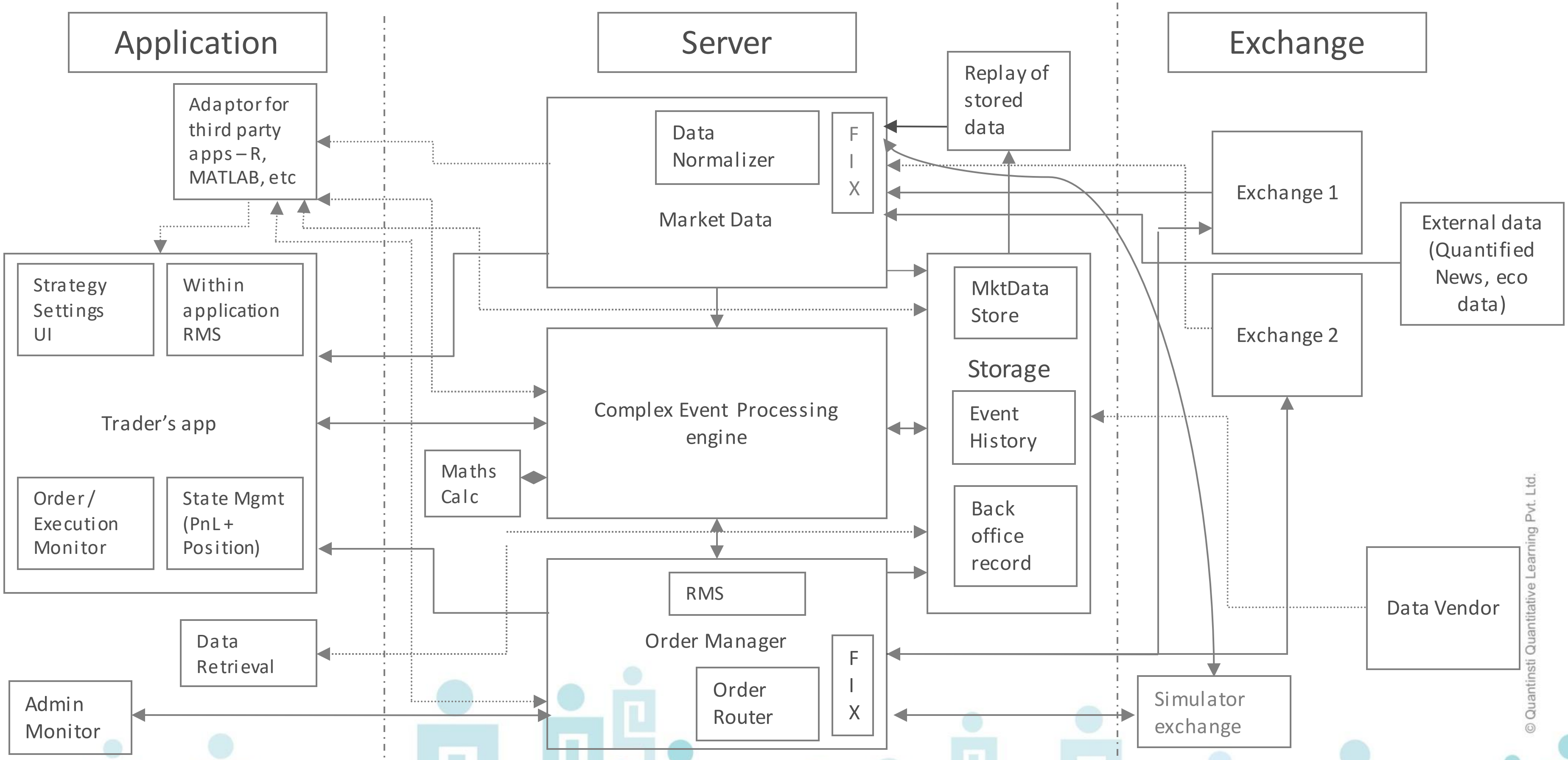
System Architecture of an Automated Trading System

To increase productivity, and to provide the ability to test against any historical scenario – the ability to feed (a.k.a. replay) historical data back to the system was added



System Architecture of an Automated Trading System

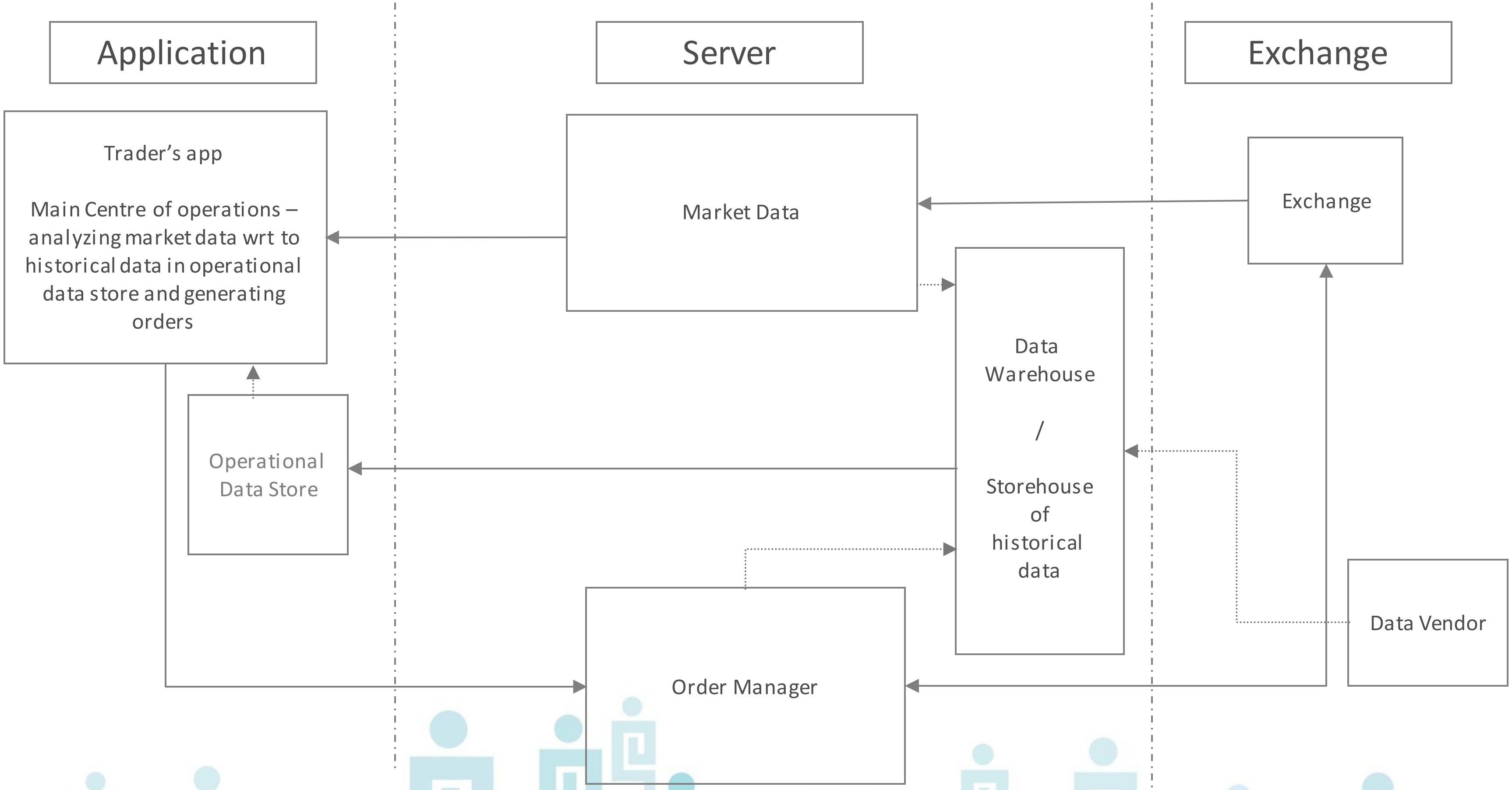
This is thus where we have ended up now.



© Quantinsti Quantitative Learning Pvt. Ltd.

System Architecture of a Traditional Trading System

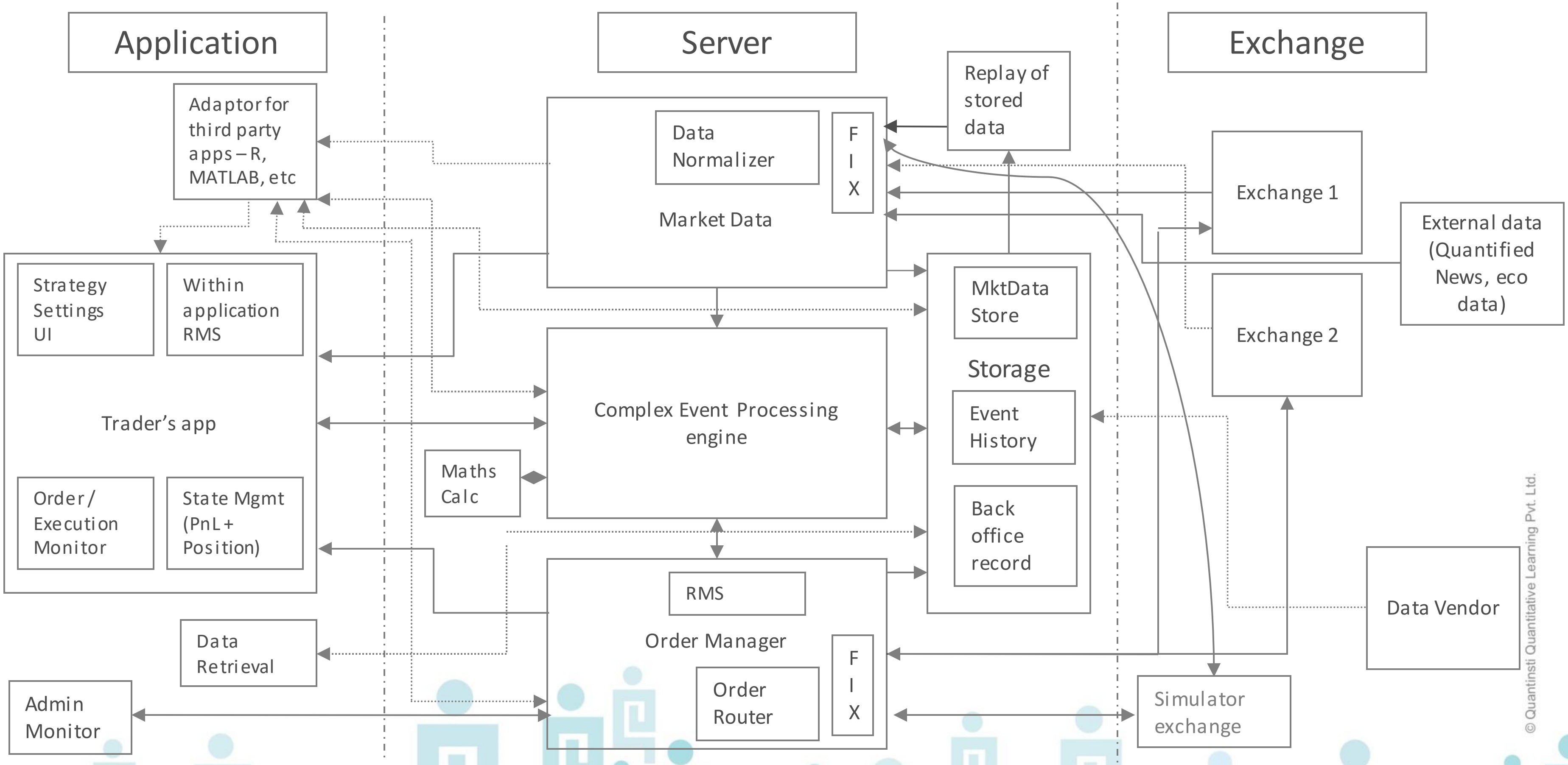
This is what we have evolved from



Application
Server
Exchange
Trader's app
Main Centre of operations – analyzing market data wrt to historical data in operational data store and generating orders
Market Data
Exchange
Data Warehouse / Storehouse of historical data
Operational Data Store
Data Vendor
Order Manager

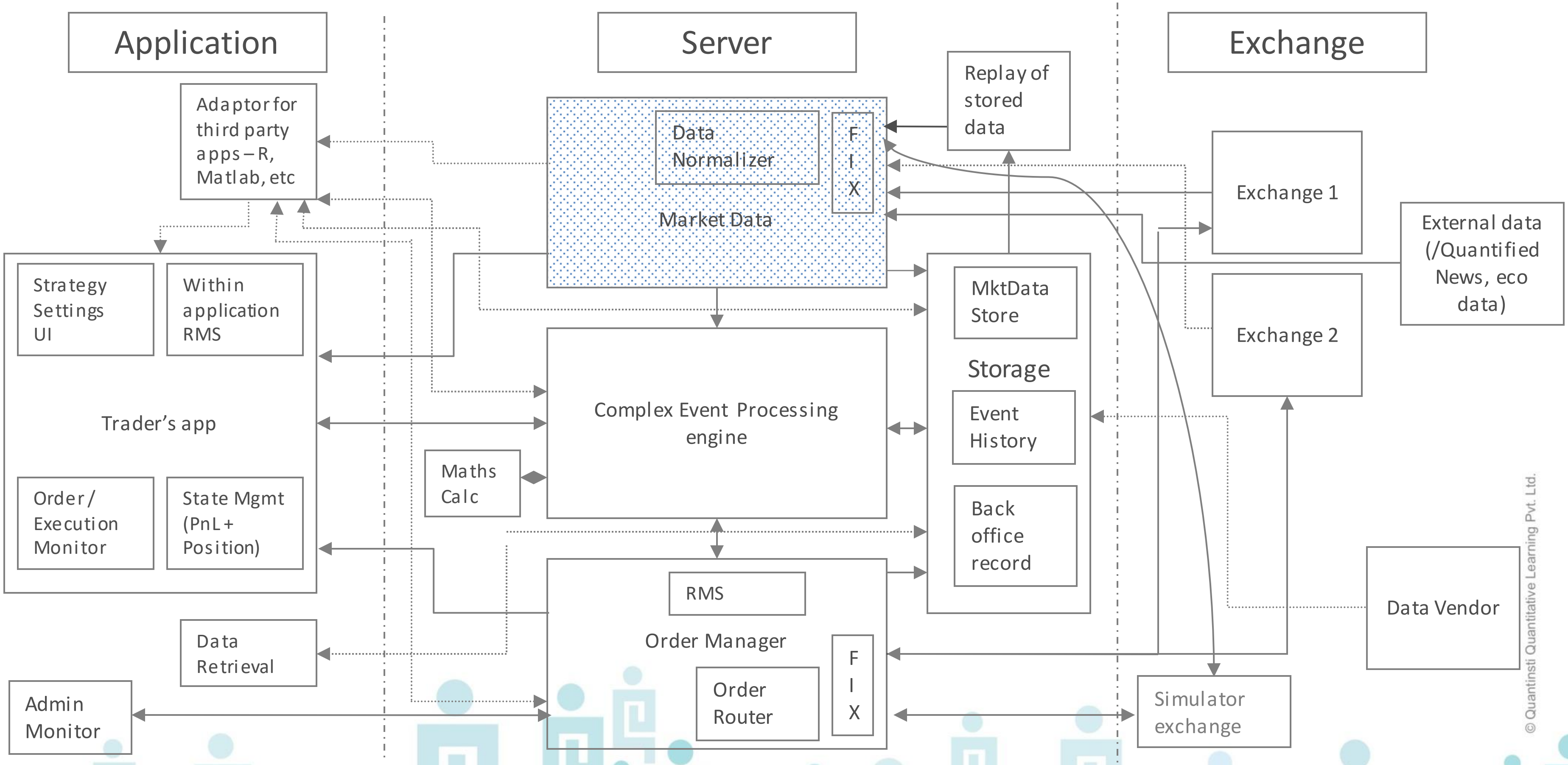
System Architecture of an Automated Trading System

The complexity is often more than this diagram can depict. Often entire blocks are implemented entirely in hardware (FPGA, ASICs) instead of being coded in high level application languages



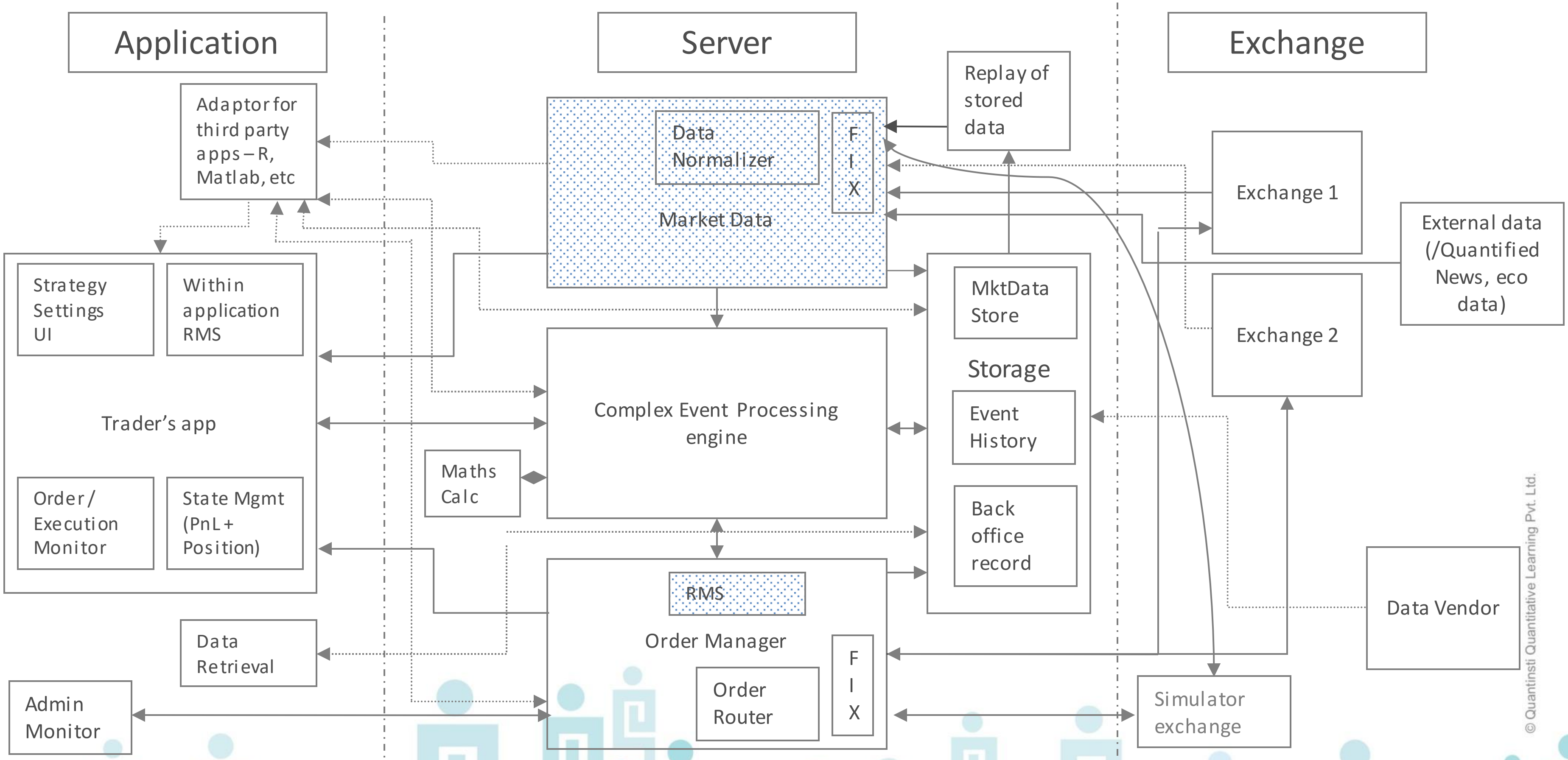
System Architecture of an Automated Trading System

FPGA implementations are done / attempted in the following blocks – (i) Market Data Adaptor



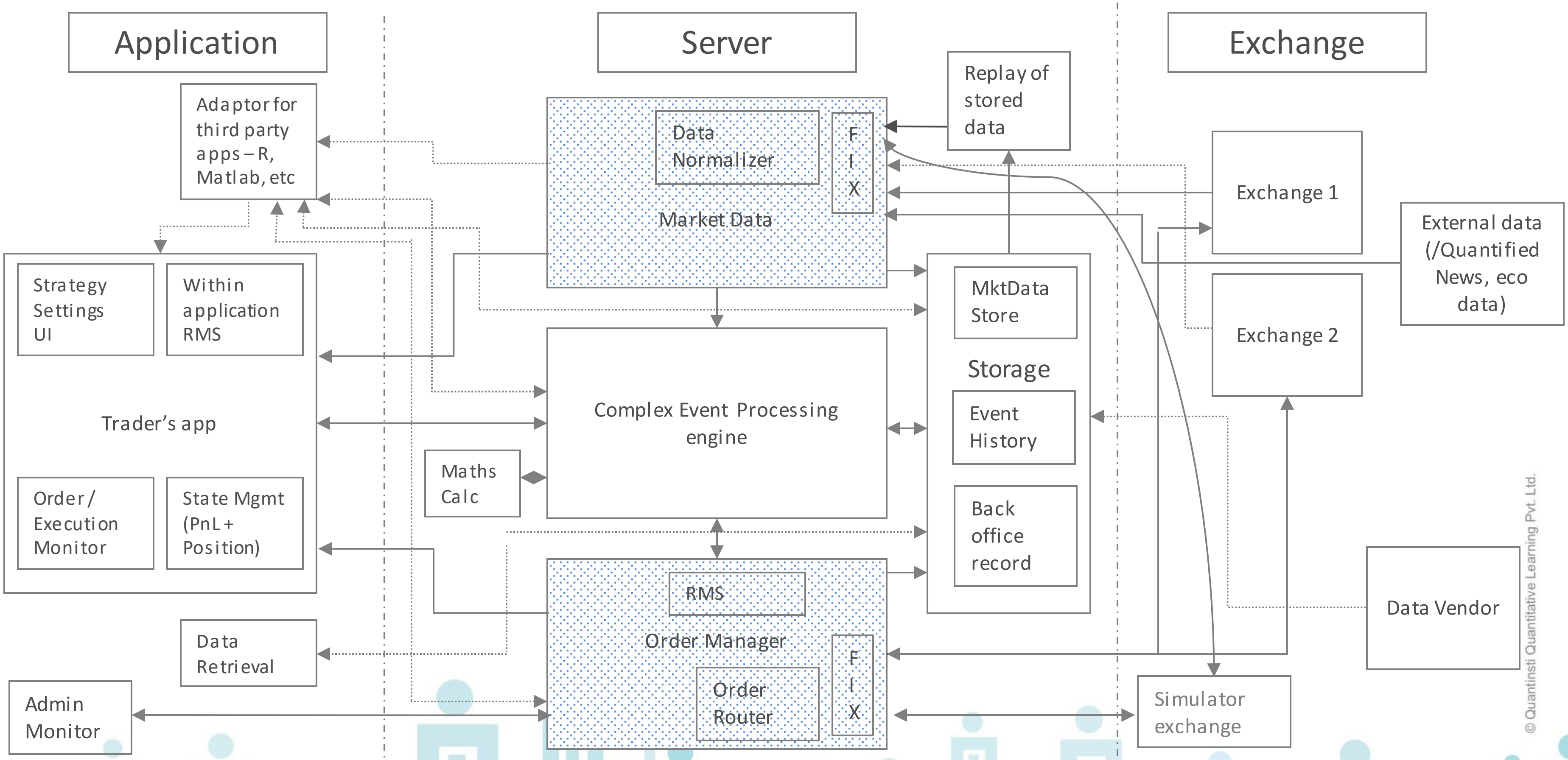
System Architecture of an Automated Trading System

FPGA implementations are done / attempted in the following blocks – (ii) RMS checks in OMS



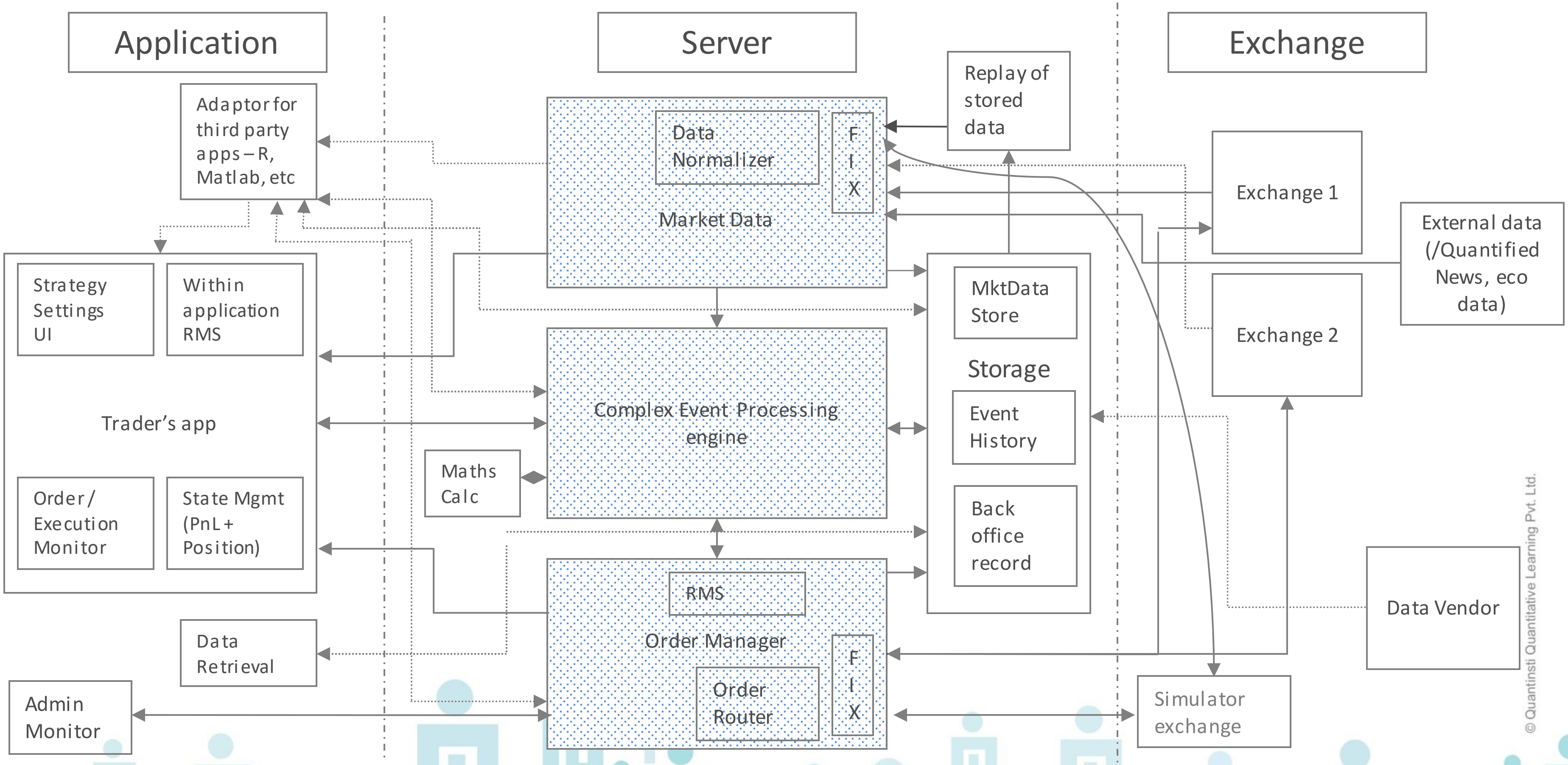
System Architecture of an Automated Trading System

FPGA implementations are done / attempted in the following blocks – (iii) the entire OMS in some cases !



System Architecture of an Automated Trading System

FPGA implementations are done / attempted in the following blocks – (iv) extremely simple trading strategies which are very very very latency sensitive



© Quantinsti Quantitative Learning Pvt. Ltd.

Market Data

The market data that the exchange shares with market participants typically contains the following basic set of information (and more):

5th Best Ask Price	Cumulative Quantity at 5th Best Ask Price
4th Best Ask Price	Cumulative Quantity at 4th Best Ask Price
3rd Best Ask Price	Cumulative Quantity at 3rd Best Ask Price
2nd Best Ask Price	Cumulative Quantity at 2nd Best Ask Price
1st Best Ask Price	Cumulative Quantity at 1st Best Ask Price
1st Best Bid Price	Cumulative Quantity at 1st Best Bid Price
2nd Best Bid Price	Cumulative Quantity at 2nd Best Bid Price
3rd Best Bid Price	Cumulative Quantity at 3rd Best Bid Price
4th Best Bid Price	Cumulative Quantity at 4th Best Bid Price
5th Best Bid Price	Cumulative Quantity at 5th Best Bid Price

This information is often provided in one out of 3 broadly popular ways …

Market Data

Market Data Method 1: Snapshot market data

The snapshot of the top ‘n’ buy and sell prices in each instrument are provided to all market participants every ‘n’ timeframe

Snapshot data packet:

5th Best Ask Price	Cumulative Quantity at 5th Best Ask Price
4th Best Ask Price	Cumulative Quantity at 4th Best Ask Price
3rd Best Ask Price	Cumulative Quantity at 3rd Best Ask Price
2nd Best Ask Price	Cumulative Quantity at 2nd Best Ask Price
1st Best Ask Price	Cumulative Quantity at 1st Best Ask Price
1st Best Bid Price	Cumulative Quantity at 1st Best Bid Price
2nd Best Bid Price	Cumulative Quantity at 2nd Best Bid Price
3rd Best Bid Price	Cumulative Quantity at 3rd Best Bid Price
4th Best Bid Price	Cumulative Quantity at 4th Best Bid Price
5th Best Bid Price	Cumulative Quantity at 5th Best Bid Price

The market participants are thus totally in the dark between two snapshots.

Market Data

Market Data Method 2: Tick By Tick Data

Each and every tick that happens at the exchange is provided to the market participants. The market participants have to then construct the order book from these basic sets of data.

TBT data packets:

New Buy Order	Quantity 100, Price 150.50
Cancel Buy Order	Cancel existing buy order for Quantity 100, Price 151
Modify Buy Order	Modify existing buy order for of (Quantity 100, Price 150.5) to (Quantity 100, Price 150)

Market Data

Market Data Method 3: Snapshot TBT

The snapshot of the top 'n' buy and sell prices in each instrument are provided to the market participants the moment any of these prices (within the top 'n' levels) change

Snapshot TBT data packet:

5th Best Ask Price	Cumulative Quantity at 5th Best Ask Price
4th Best Ask Price	Cumulative Quantity at 4th Best Ask Price
3rd Best Ask Price	Cumulative Quantity at 3rd Best Ask Price
2nd Best Ask Price	Cumulative Quantity at 2nd Best Ask Price
1st Best Ask Price	Cumulative Quantity at 1st Best Ask Price
1st Best Bid Price	Cumulative Quantity at 1st Best Bid Price
2nd Best Bid Price	Cumulative Quantity at 2nd Best Bid Price
3rd Best Bid Price	Cumulative Quantity at 3rd Best Bid Price
4th Best Bid Price	Cumulative Quantity at 4th Best Bid Price
5th Best Bid Price	Cumulative Quantity at 5th Best Bid Price

Market participants don't have to maintain the order book, and also get information the moment it changes

Market Data

In case of snapshot data, typically various financial instruments are grouped together. Market participants can subscribe to snapshot data for various groups, and get snapshot data for all instruments in that group.

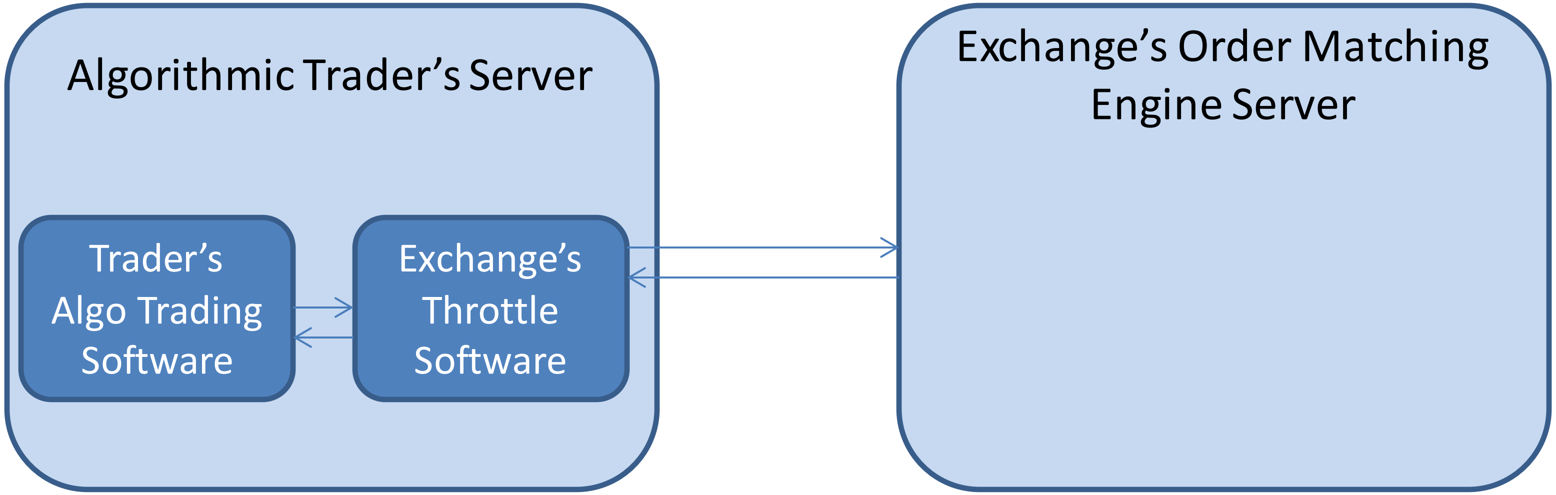
Snapshot market data usually has the lowest volume of data. And is useful for market participants who do not have the need for micro-timeframe information.

In case of Tick By Tick data (TBT data) and Snapshot TBT, the amount of data is huge. Typically exchanges allow market participants to listen to TBT (or Snapshot TBT) for specific financial instruments only (even within a group). This reduces the data load on the exchange infrastructure as well.

Since in case of TBT or Snapshot TBT, information is provided immediately after the occurrence of the event - therefore arbitrageurs, market makers, and high frequency traders use such data.

Order Sending

In some exchanges, they force participants to send orders to a piece of software provided by the exchange but residing on the trading member's servers. Such pieces of software perform load smoothening at the trader's end and ensures that a smooth stream of information reaches the exchange.



Algorithmic Trader's Server
Trader's Algo Trading Software
Exchange's Throttle Software
Exchange's Order Matching Engine Server

However, this design is getting obsolete – and exchanges force traders to do flow management at their end themselves. If algorithmic trading software violate flow management metrics, the exchange disconnect them.

Order Sending

In terms of order sending, trading systems have to mainly handle the following limits:

i) Number of Orders sent per second. (higher order rates (but only up to a certain limit) can be received on paying more)

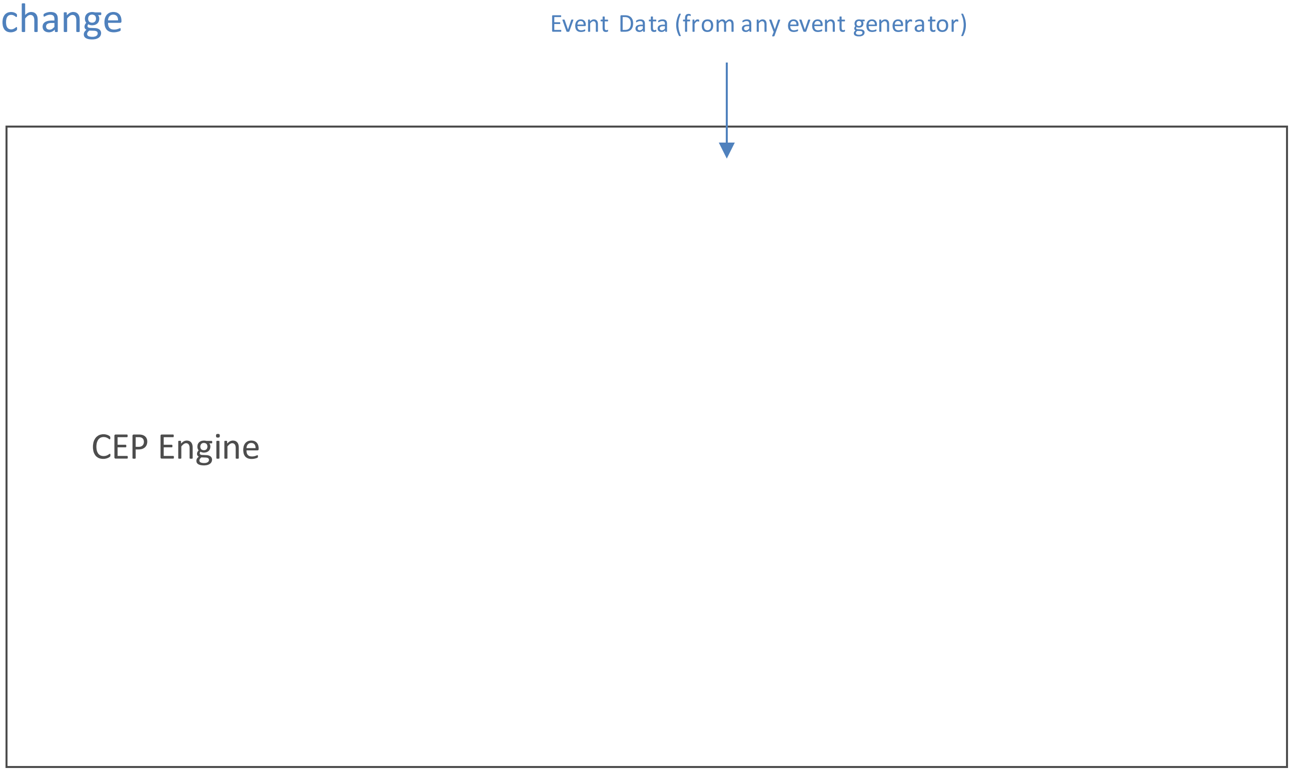
ii) Ratio of Orders sent to Trades done. (This ratio depends upon the liquidity category of the instrument in which orders have been sent – often this ratio is applied in a lax way for illiquid instruments. This ratio is also sometimes calculated only if the orders are sent at untradeable prices – i.e. prices which are far away from the top of the order book)

The Complex Event Processing (CEP) Module

The CEP is the heart of the system – which listens to market data and determines what actions to take based on algorithm settings

The Complex Event Processing (CEP) Module

The Input to the CEP is 'Event Data'. This could be (i) market data or (ii) change of strategy settings in the application. This could also be (iii) order reports (execution or acknowledgement) from the exchange

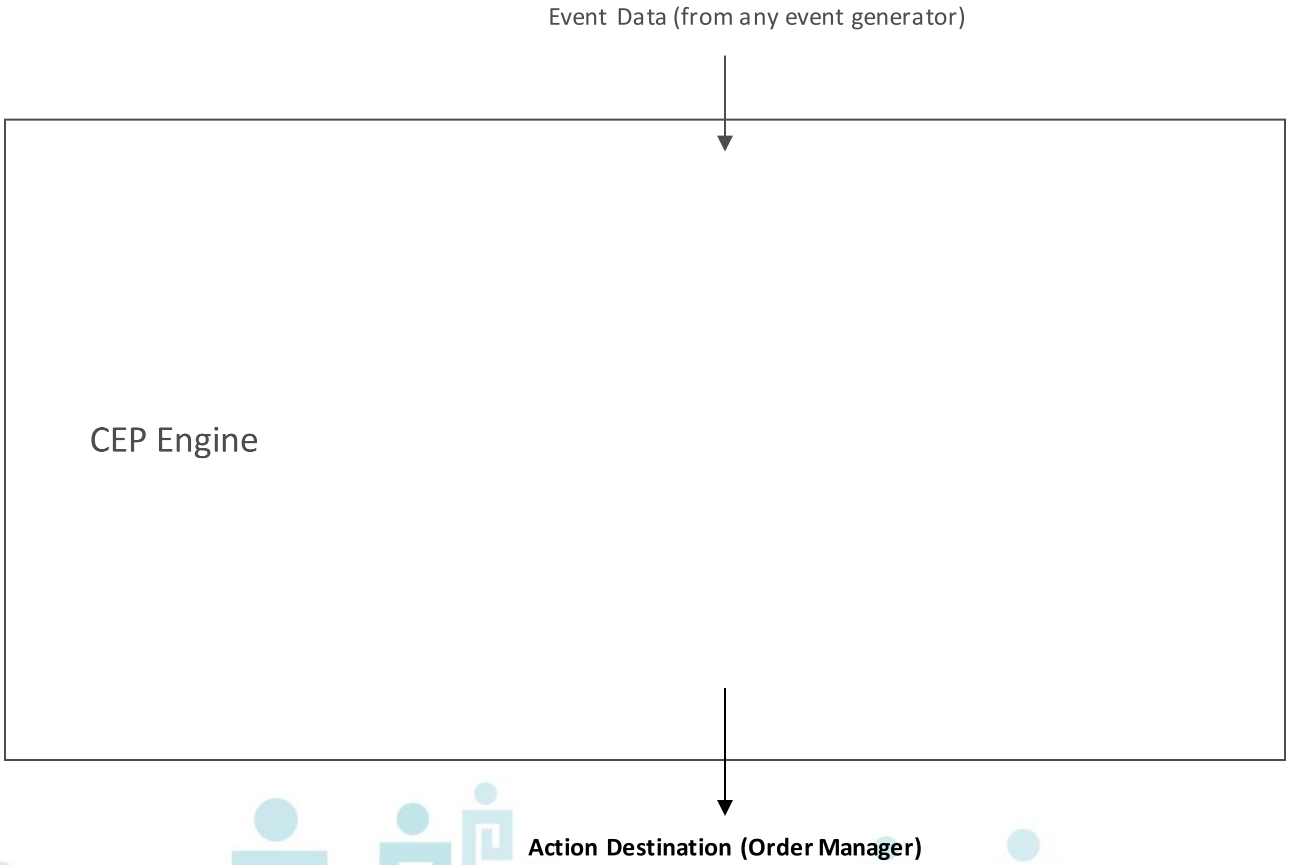


Event Data (from any event generator)

CEP Engine

The Complex Event Processing (CEP) Module

The Output of the CEP module is (i) an action (sending an order request to the Order Manager)



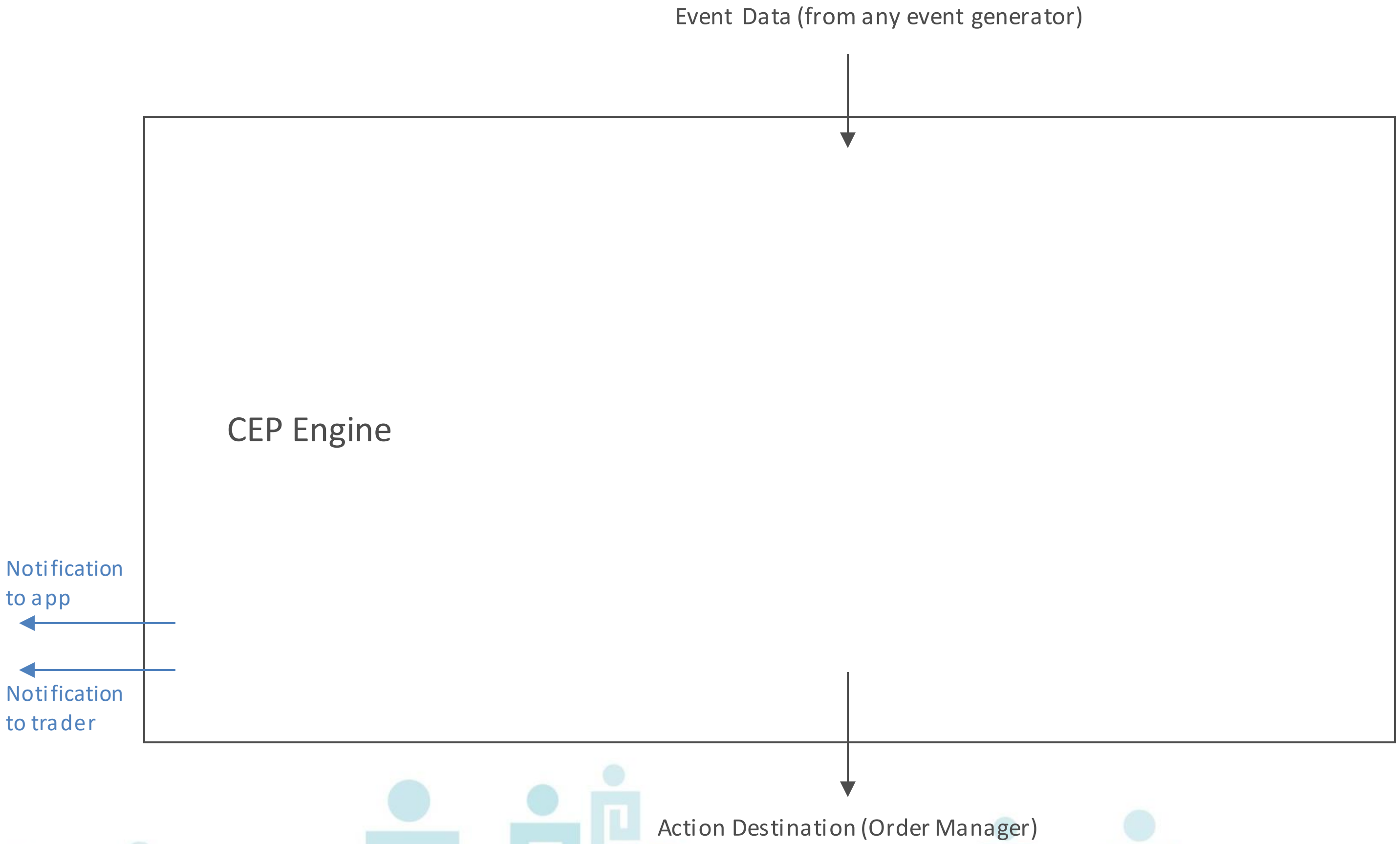
Event Data (from any event generator)

CEP Engine

Action Destination (Order Manager)

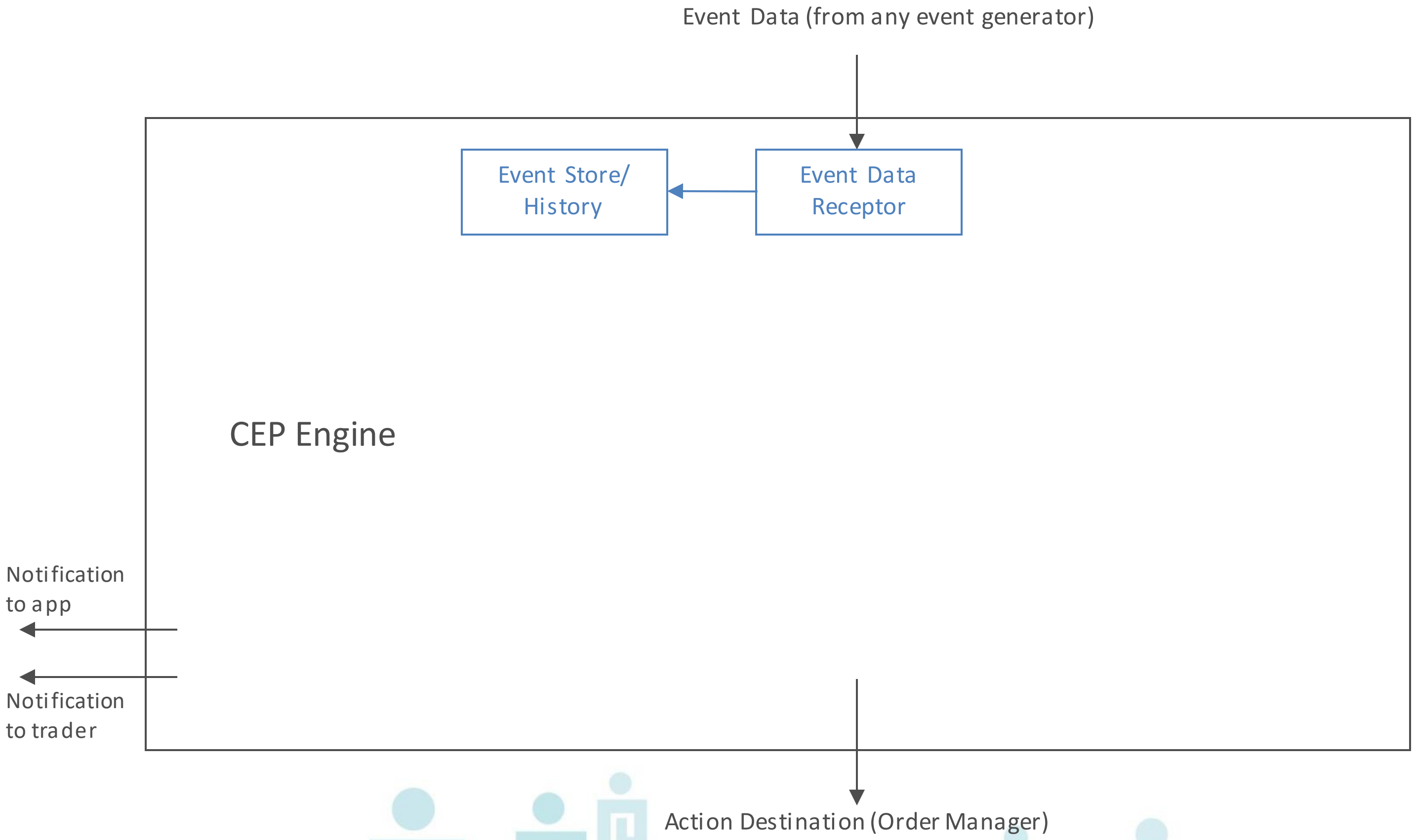
The Complex Event Processing (CEP) Module

… and (ii) sending notifications to both the trader and the application


Event Data (from any event generator)
CEP Engine
Notification to app
Notification to trader
Action Destination (Order Manager)

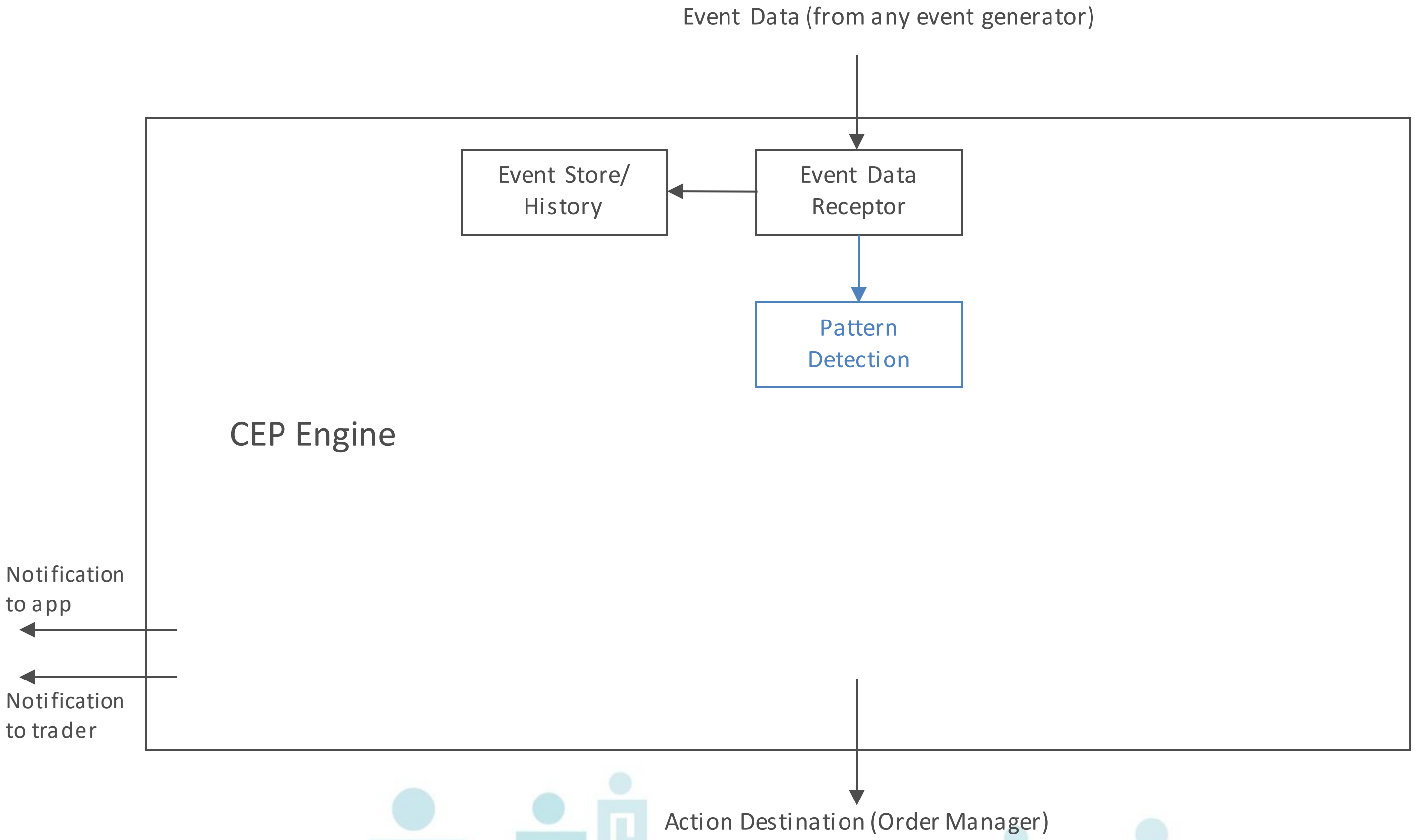
The Complex Event Processing (CEP) Module

Within the CEP black box, the first task is to decode the event data in a receptor, as well as store it in event-history



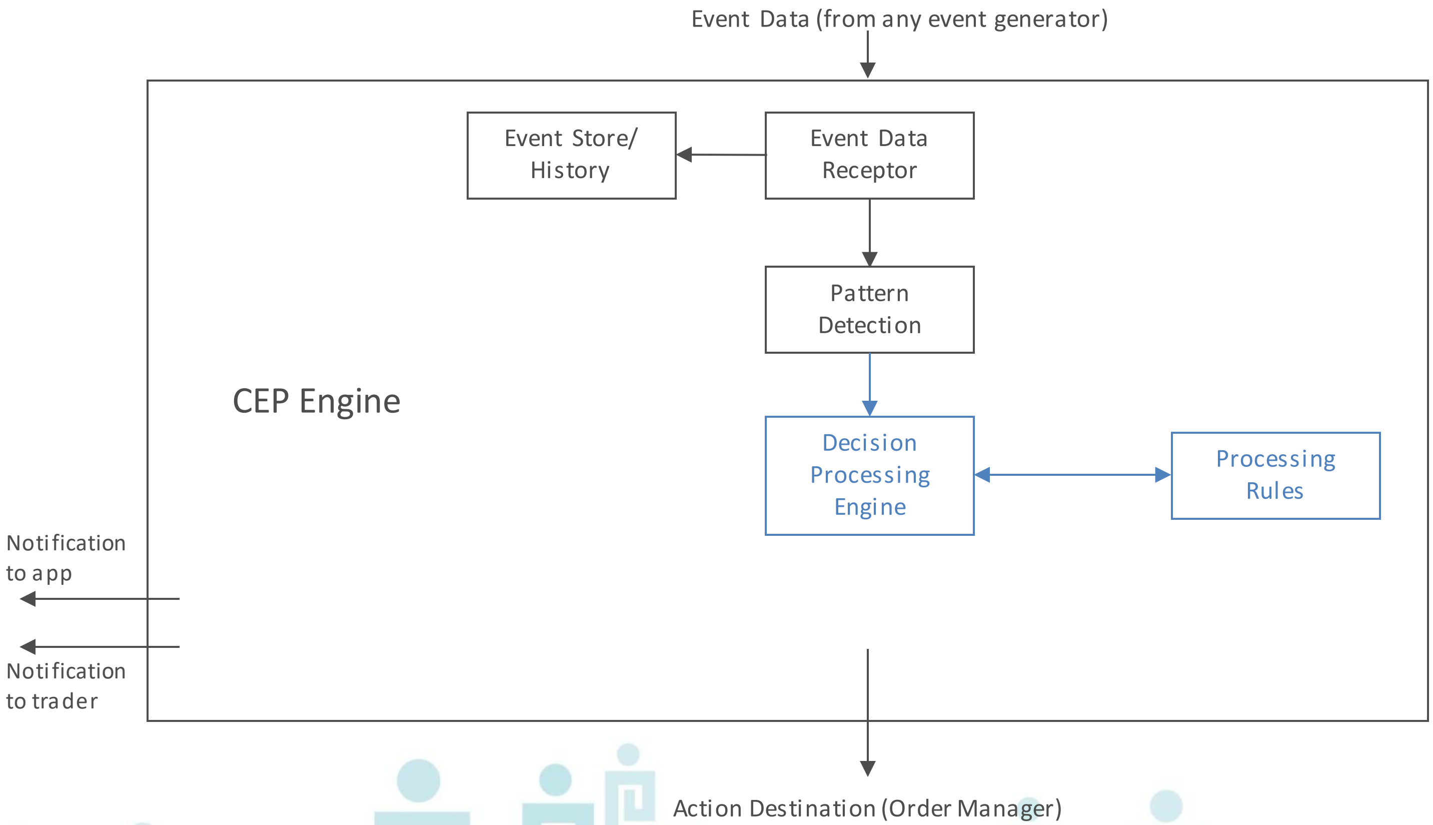
The Complex Event Processing (CEP) Module

The next block checks the event for known patterns



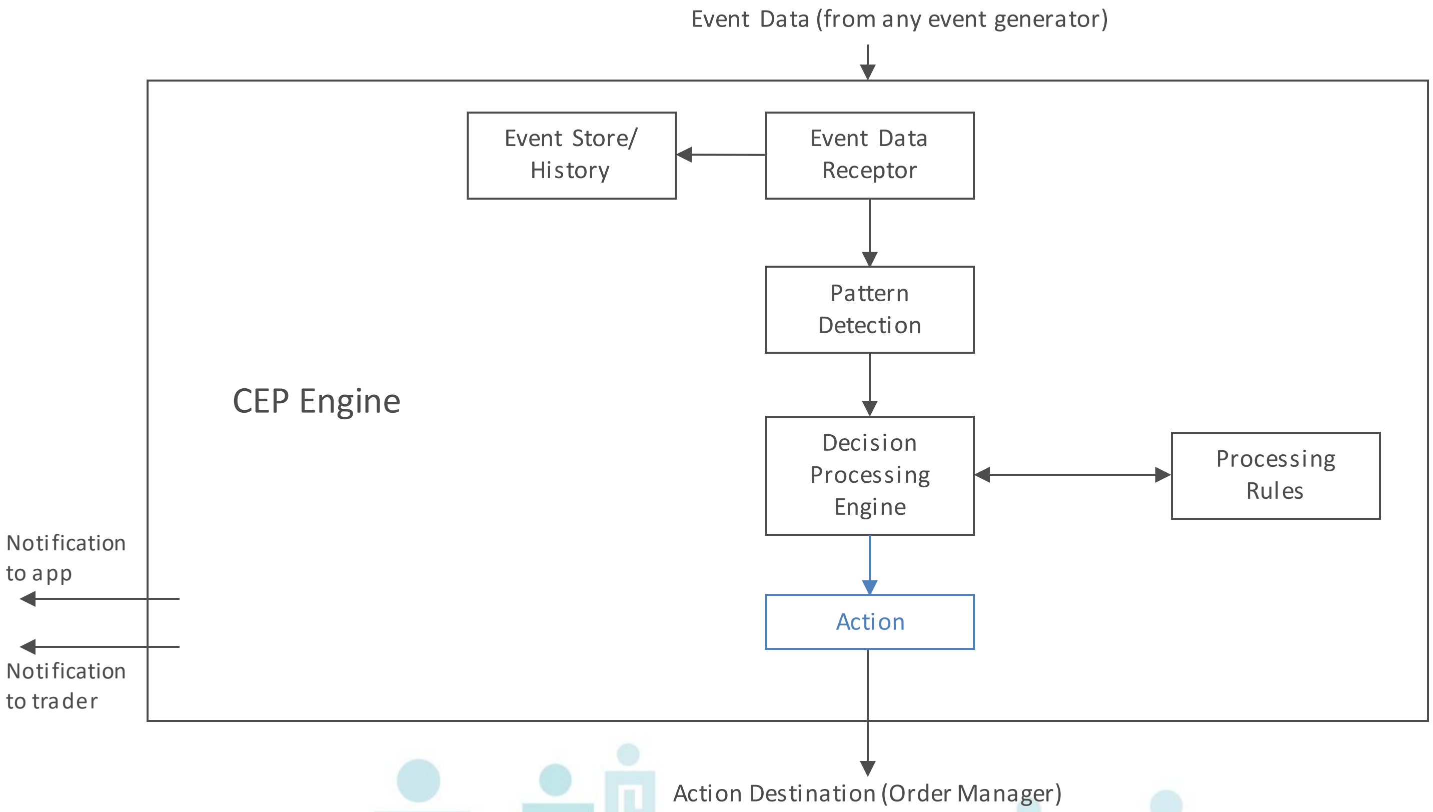
The Complex Event Processing (CEP) Module

Based on Decision Processing Rules, a Decision Processing Engine determines what to do in case a pattern is recognized



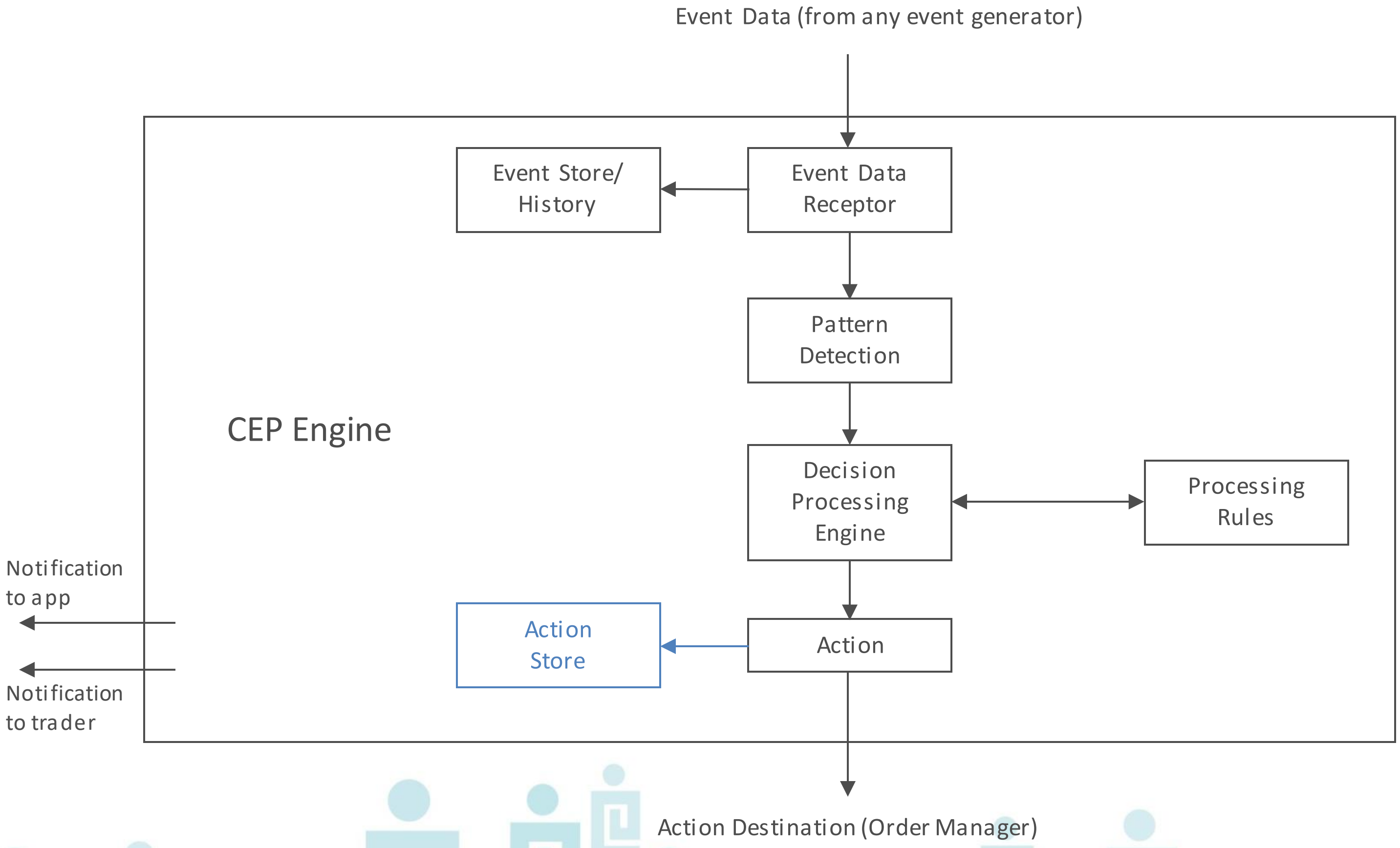
The Complex Event Processing (CEP) Module

If the Decision Processing Engine determines that an action has to be done, then it is conveyed to the action destination outside the CEP



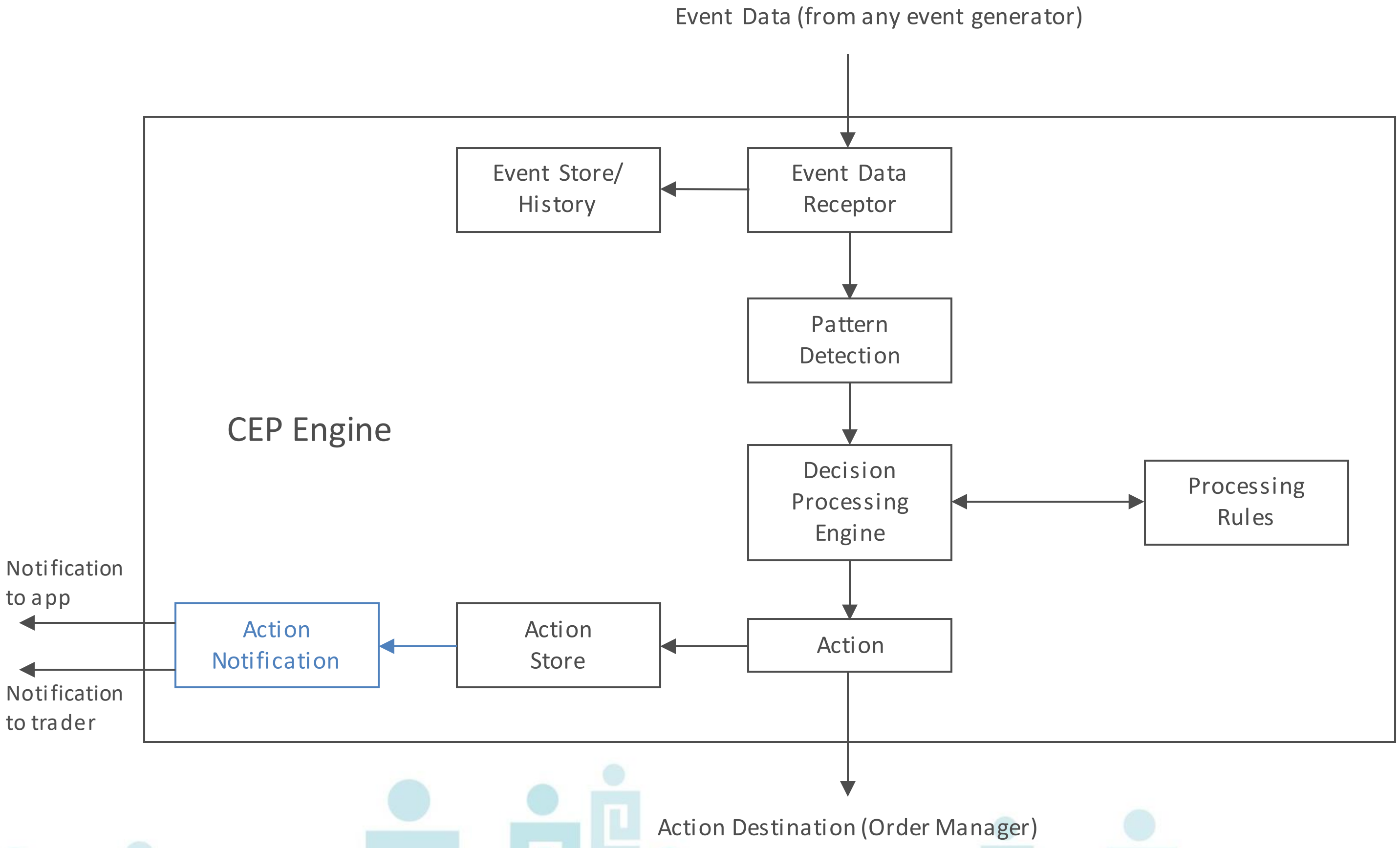
The Complex Event Processing (CEP) Module

For regulatory and for analysis purposes, this information is stored in an action-store



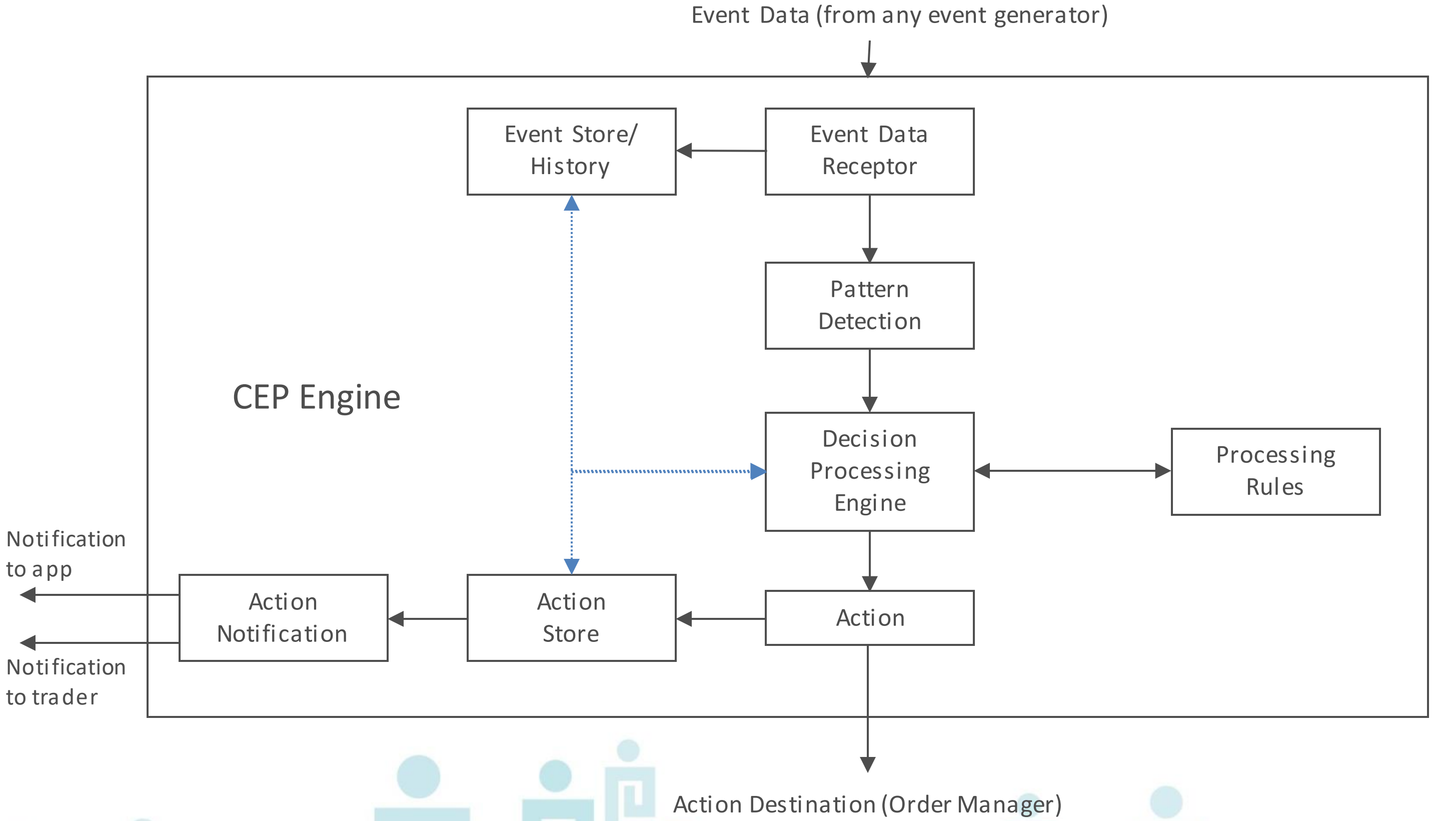
The Complex Event Processing (CEP) Module

Information about the action is then conveyed to traders & application by the Action Notification block



The Complex Event Processing (CEP) Module

In a boot-strapping CEP, the Decision Processing Engine could consult the event store and action store to formulate processing rules in the fly

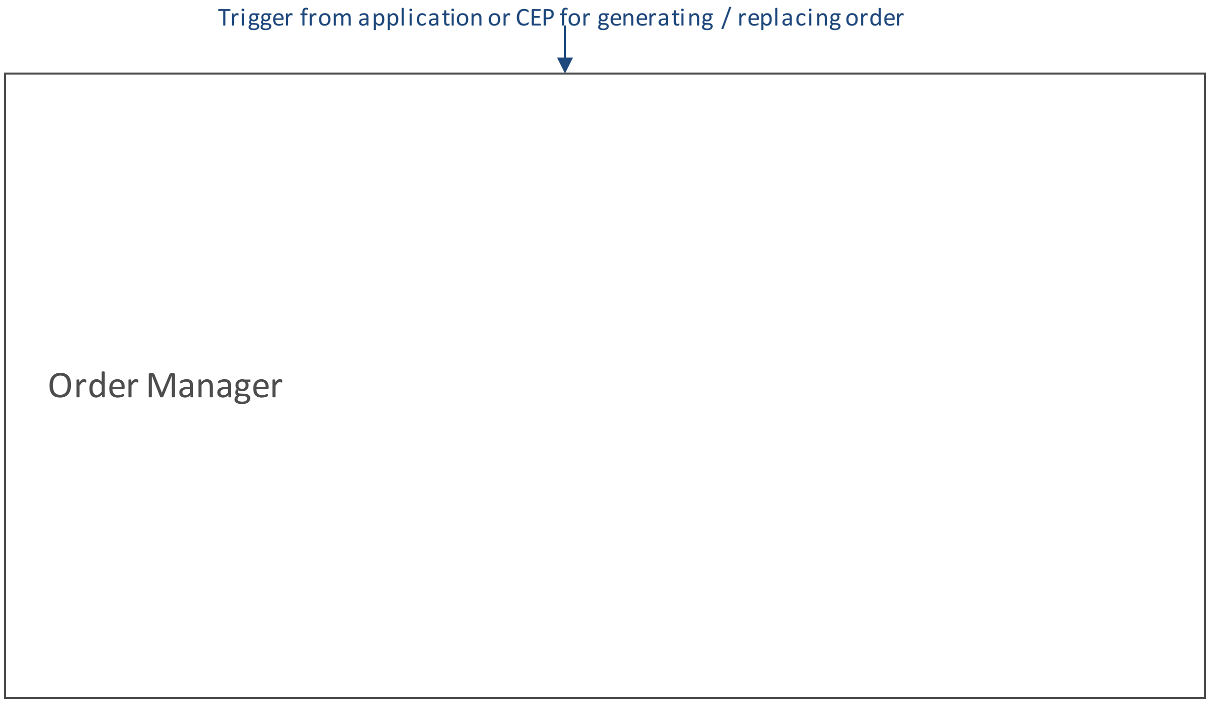


The Order Manager

The Order Manager generates and manages Orders sent from the system to multiple destinations. Moreover it also perform RMS in real time before sending an order

The Order Manager

The input to the OM is the signal from the CEP block

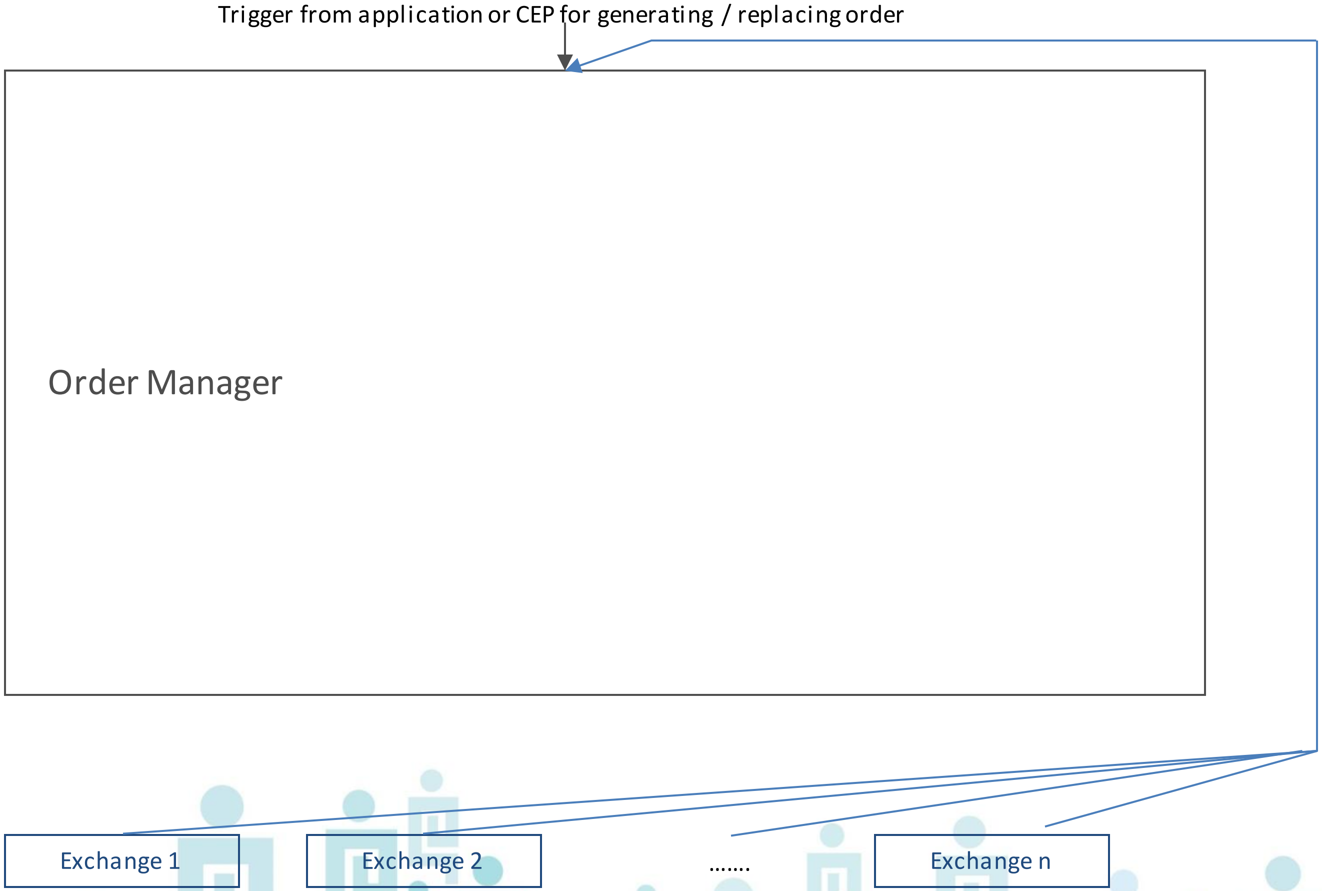


Trigger from application or CEP for generating / replacing order

Order Manager

The Order Manager

Another input to the OM is order acknowledgements and execution reports from the exchanges



Trigger from application or CEP for generating / replacing order

Order Manager

Exchange 1

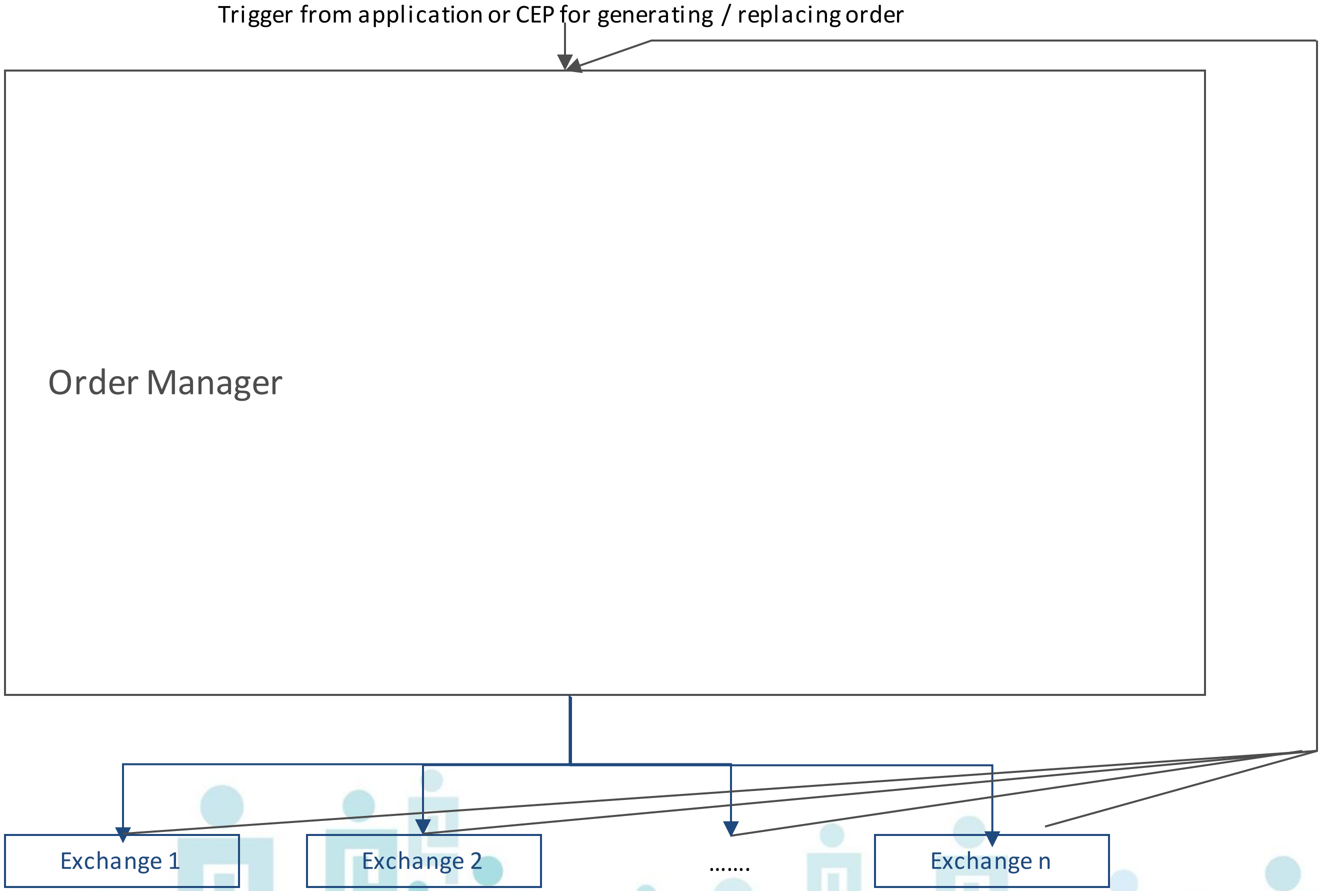
Exchange 2

…….

Exchange n

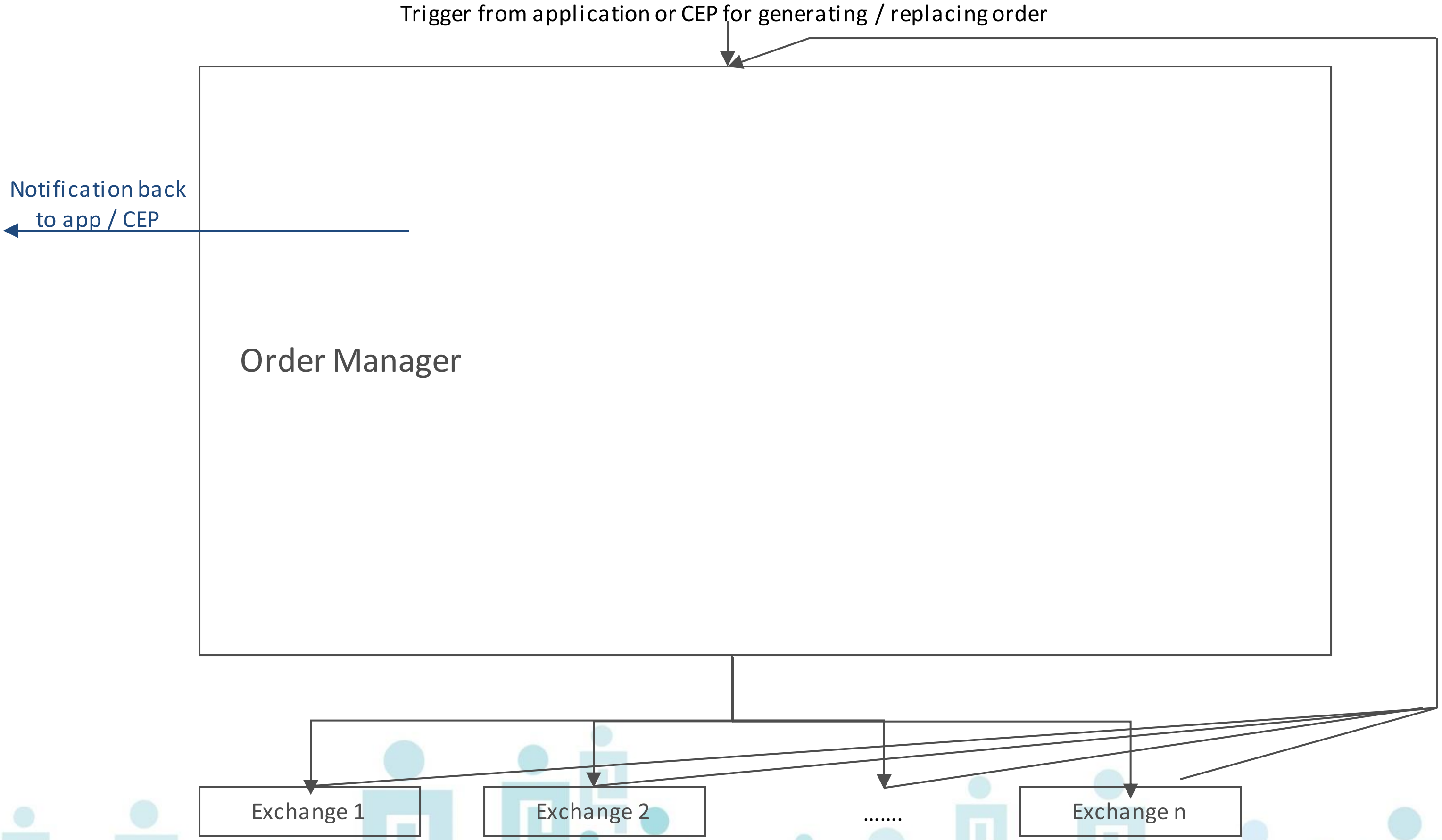
The Order Manager

The output of the OM is (i) orders routed to exchanges / other destinations


Trigger from application or CEP for generating / replacing order
Order Manager
Exchange 1
Exchange 2
…….
Exchange n

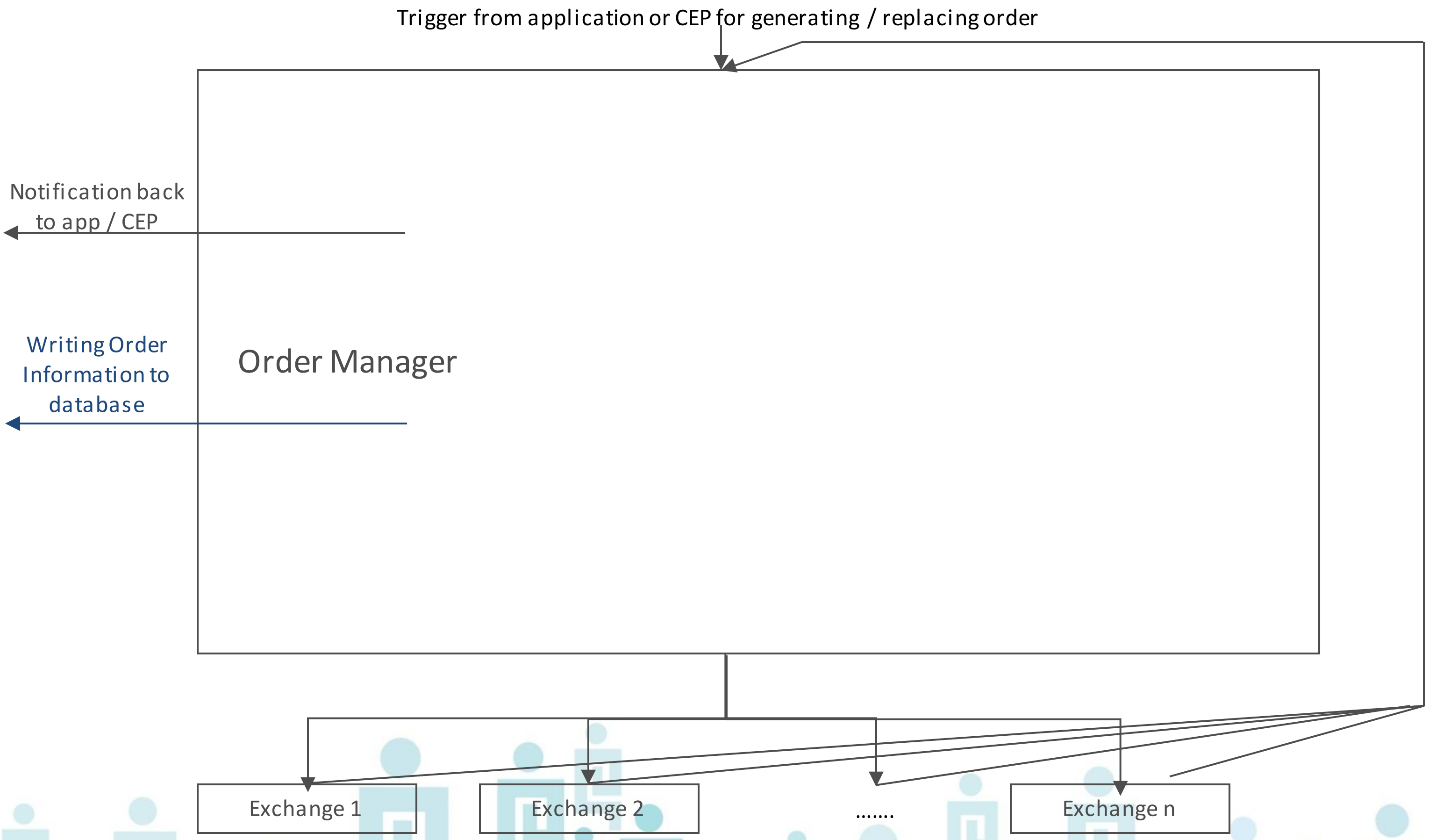
The Order Manager

… (ii) notifications back to the application


Trigger from application or CEP for generating / replacing order
Notification back to app / CEP
Order Manager
Exchange 1
Exchange 2
…….
Exchange n

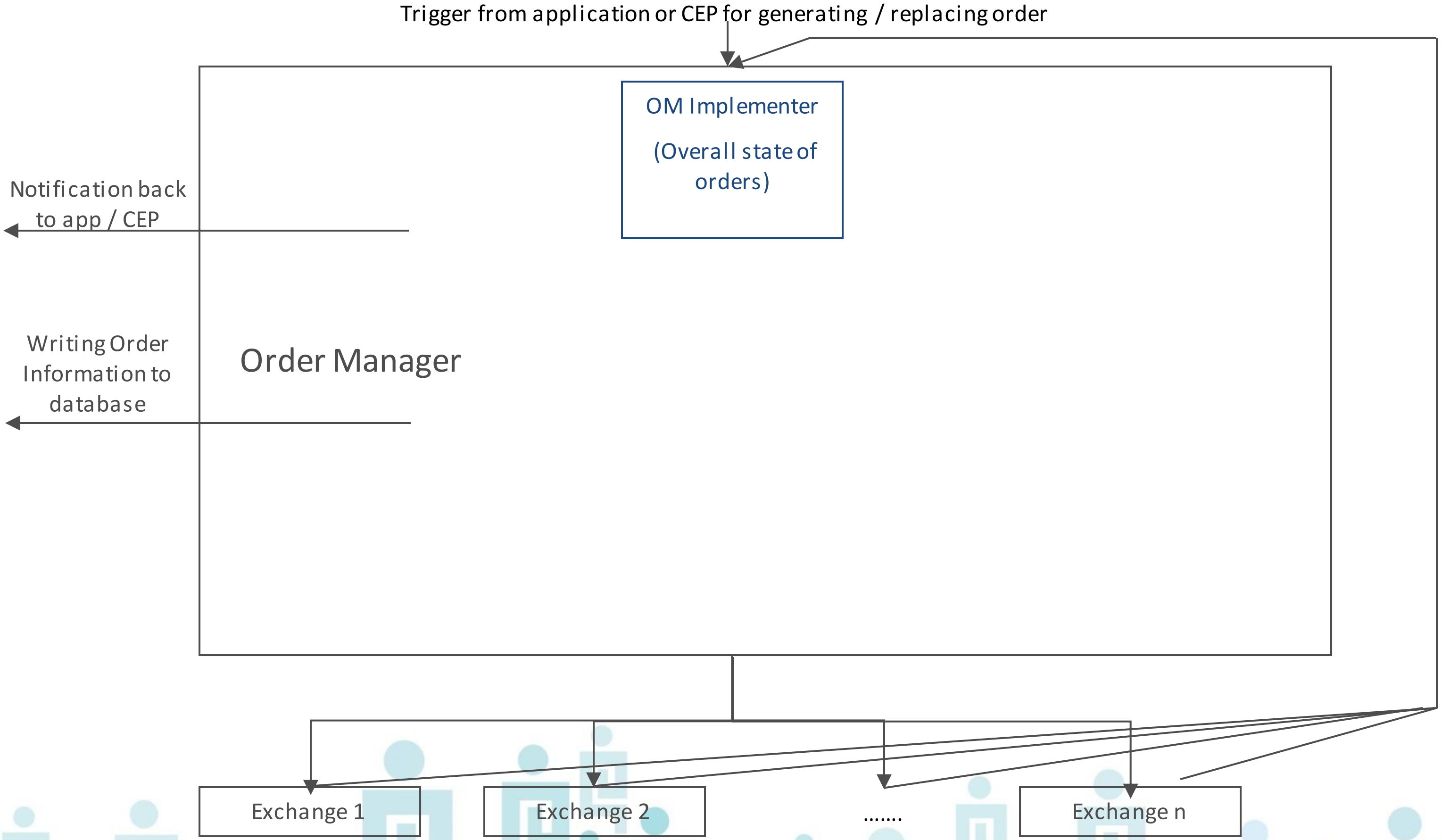
The Order Manager

… and (iii) writing order information into a database


Trigger from application or CEP for generating / replacing order
Notification back to app / CEP
Writing Order Information to database
Order Manager
Exchange 1
Exchange 2
…….
Exchange n

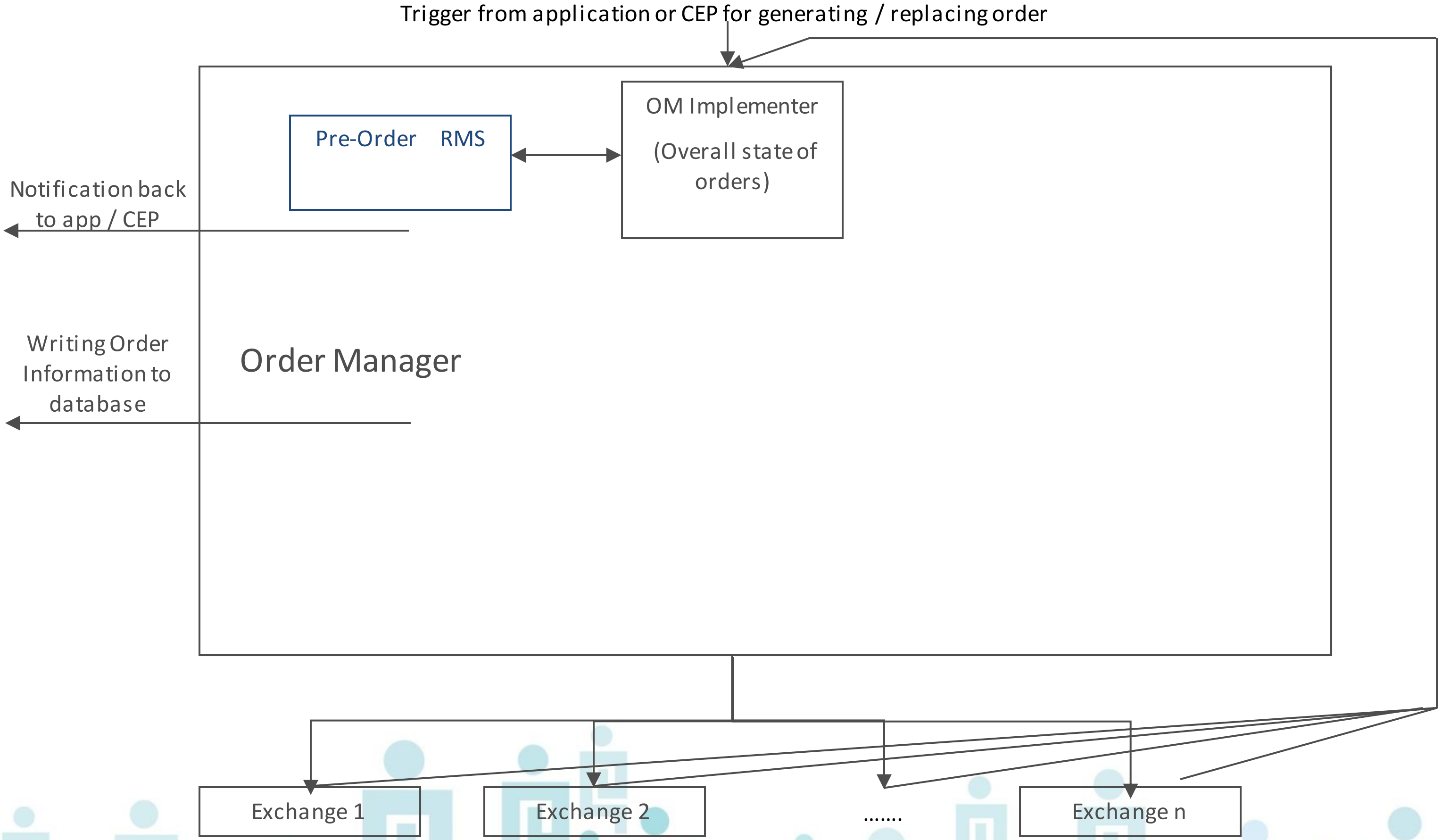
The Order Manager

The trigger is handled by the OM implementer which maintains an overall state of orders


Trigger from application or CEP for generating / replacing order
OM Implementer
(Overall state of orders)
Notification back to app / CEP
Writing Order Information to database
Order Manager
Exchange 1
Exchange 2
…….
Exchange n

The Order Manager

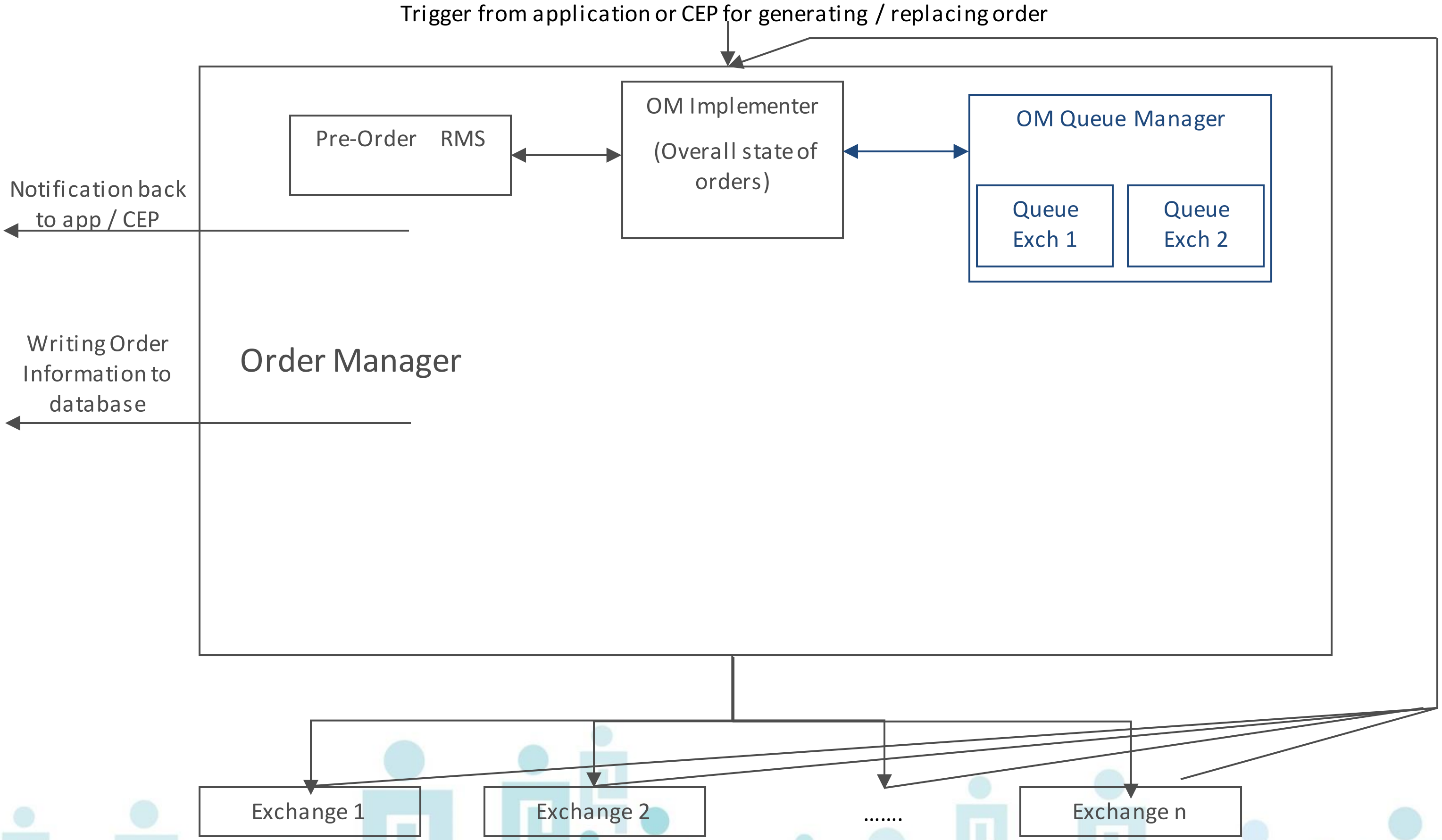
It does the Pre-Order RMS (max order size, net portfolio position, max trade value, etc)


Trigger from application or CEP for generating / replacing order
Pre-Order RMS
OM Implementer
(Overall state of orders)
Notification back to app / CEP
Writing Order Information to database
Order Manager
Exchange 1
Exchange 2
…….
Exchange n

© Quantinsti Quantitative Learning Pvt. Ltd.

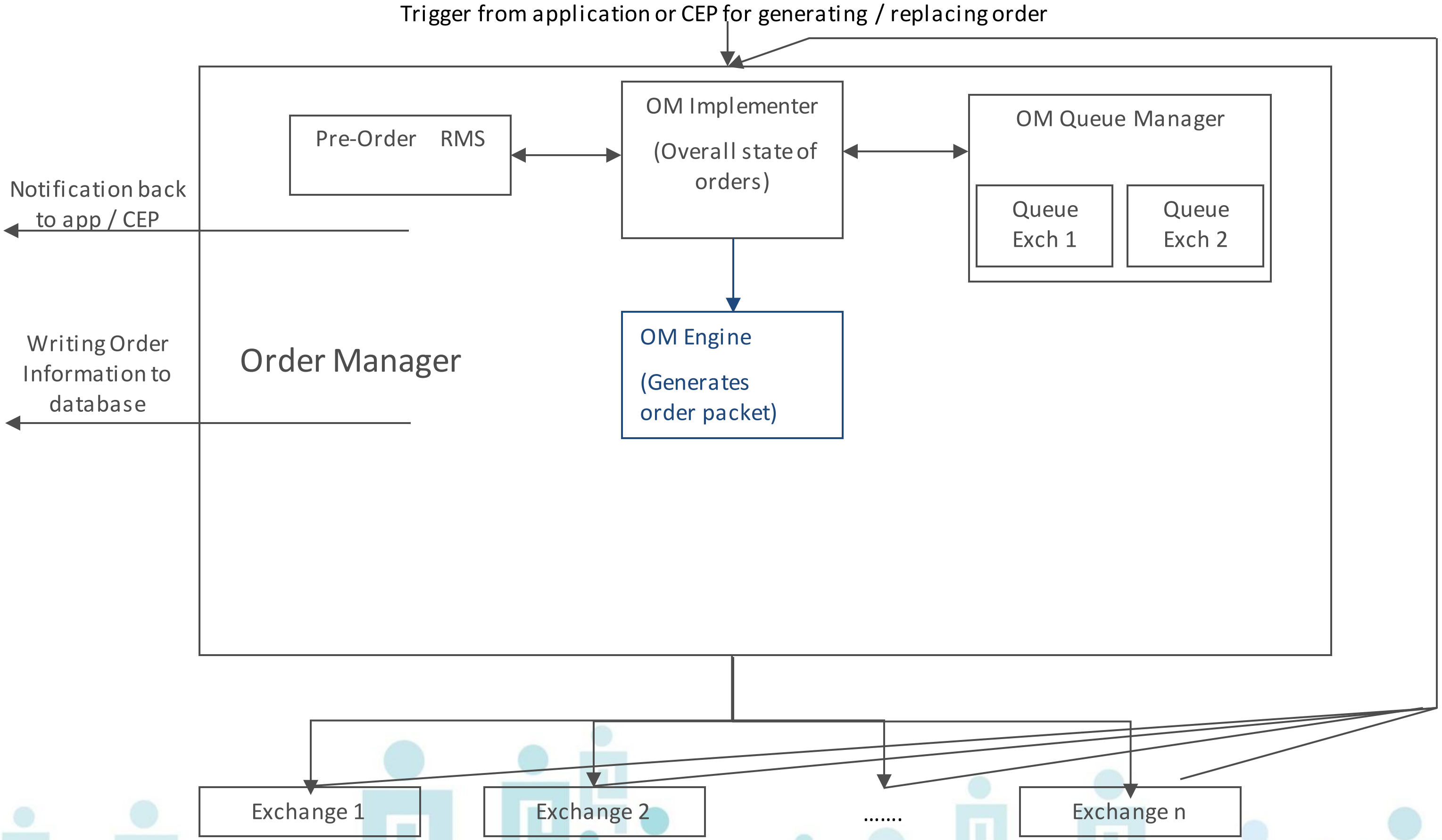
The Order Manager

… and then checks the OM Queue for each destination


Trigger from application or CEP for generating / replacing order
OM Implementer
(Overall state of orders)
Pre-Order RMS
OM Queue Manager
Queue Exch 1
Queue Exch 2
Notification back to app / CEP
Writing Order Information to database
Order Manager
Exchange 1
Exchange 2
…….
Exchange n

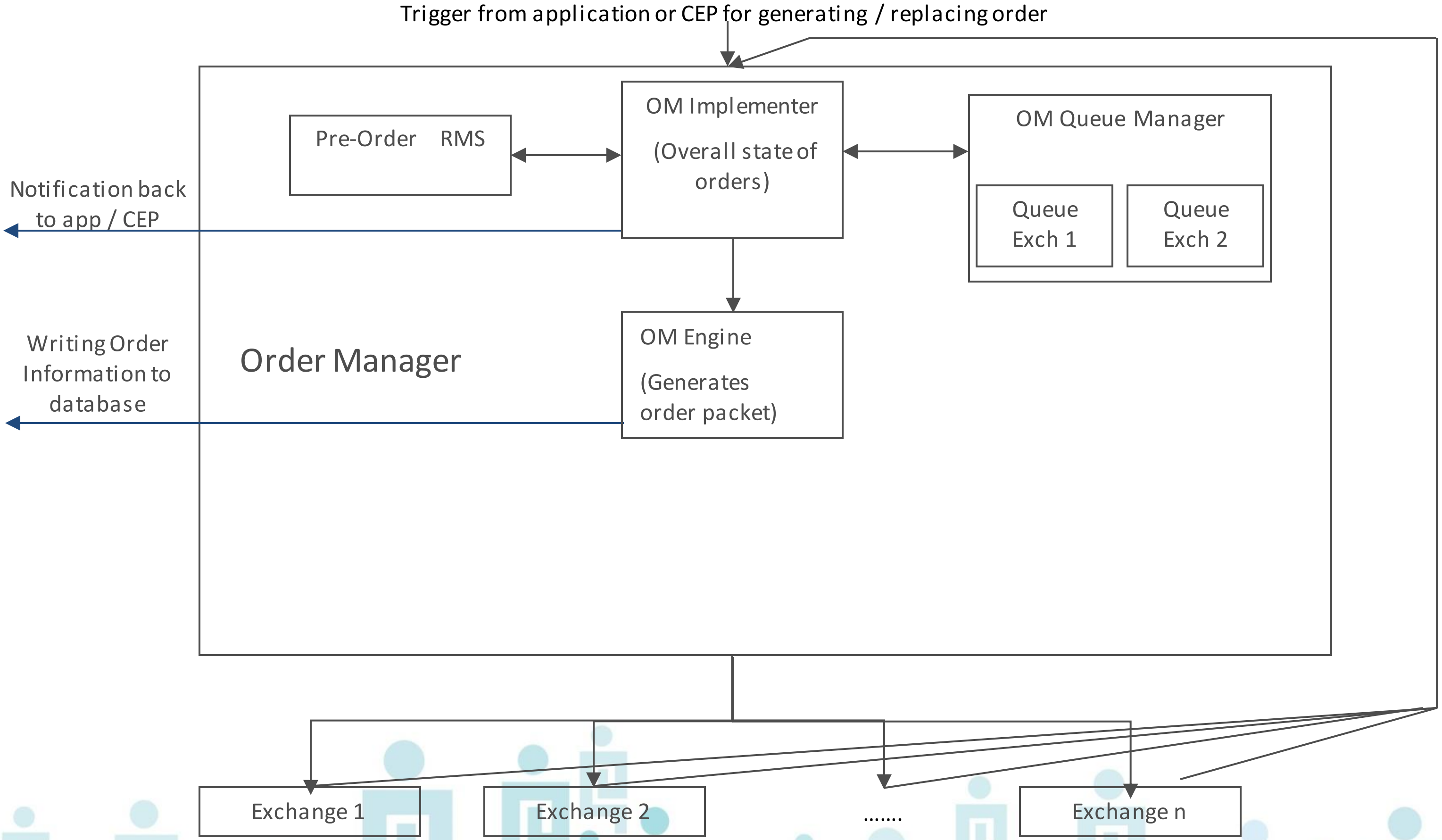
The Order Manager

If the queue is free, it then orders the OM engine to prepare a packet


Trigger from application or CEP for generating / replacing order
Pre-Order RMS
OM Implementer
(Overall state of orders)
OM Queue Manager
Queue Exch 1
Queue Exch 2
Notification back to app / CEP
Writing Order Information to database
Order Manager
OM Engine
(Generates order packet)
Exchange 1
Exchange 2
....
Exchange n

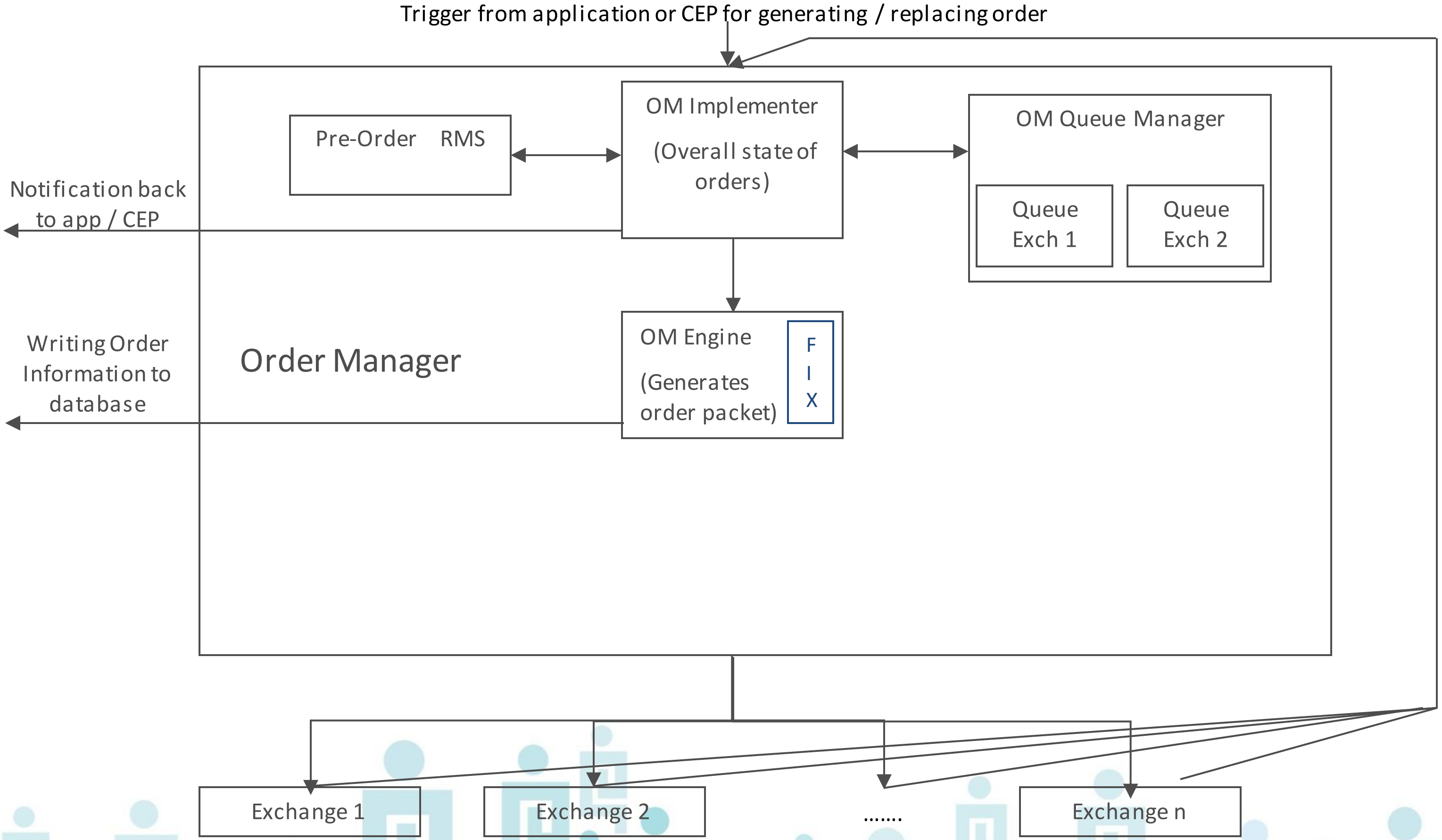
The Order Manager

This information is conveyed to the app/CEP and is also noted into the database


Trigger from application or CEP for generating / replacing order
Pre-Order RMS
OM Implementer
(Overall state of orders)
OM Queue Manager
Queue Exch 1
Queue Exch 2
Notification back to app / CEP
OM Engine
(Generates order packet)
Writing Order Information to database
Order Manager
Exchange 1
Exchange 2
....
Exchange n

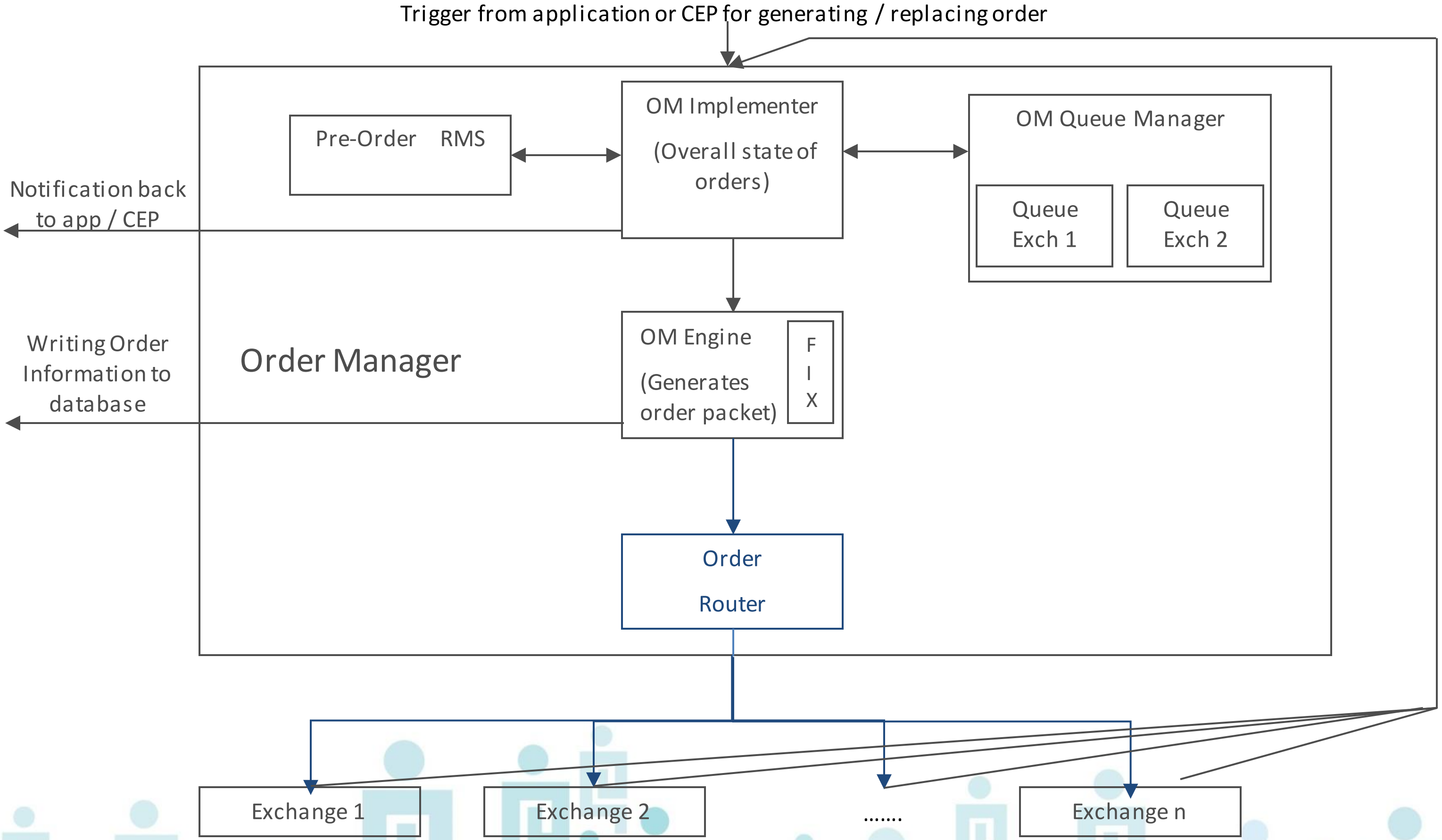
The Order Manager

For FIX protocol destinations, the orders are generated in FIX format


Trigger from application or CEP for generating / replacing order
Pre-Order RMS
OM Implementer
(Overall state of orders)
OM Queue Manager
Queue Exch 1
Queue Exch 2
Notification back to app / CEP
OM Engine
(Generates order packet)
F I X
Writing Order Information to database
Order Manager
Exchange 1
Exchange 2
…….
Exchange n

The Order Manager

Next the Order Router determines the destination of the order, and forwards the message to the correct line



The Future

What’s going on?

- Computing: Approaching limit of transistor density, i.e. computing is hitting the deadlock

The Future

What’s going on?

- Computing: Approaching limit of transistor density, i.e. computing is hitting the deadlock
- Speed: Approaching speed of light

The Future

What's going on?

- Computing: Approaching limit of transistor density, i.e. computing is hitting the deadlock
- Speed: Approaching speed of light

⚠ We are almost hitting the boundaries of the laws of physics. ⚠

Thank You