

Product



- [Product](#)
- [Market Data Nexus](#)
- [CEP](#)
- [Smart Order Routing](#)
- [Strategy Engines](#)
- [DARE](#)
- [Photon](#)
- [System Administration](#)
- [Database Access Layer and Persistence](#)
- [Threading Model](#)
- [High Availability and Failover](#)

PRODUCT

Marketcetera Module Framework

The Marketcetera platform is comprised of a series of modules that can be snapped together in different combinations like a series of molecules combining to create new substances.

Marketcetera modules perform logically discrete functions, interacting through well-defined interfaces. Modules are independent pieces of functionality that may be packaged separately or even supplied by different vendors. A module can emit data and / or receive data and / or request data flows. The Module Framework provides an operating environment for these modules to exist and exchange data.

This modular design has significant advantages for our customers. Most importantly, it allows us to very quickly create affordable customized deployments for our customers. Unlike monolithic applications (where the smallest component is the whole), Marketcetera wrote several smaller modules separately so that, when composed together, they construct the executable application program. This makes our modular approach far more reusable than a traditional monolithic design, since all of these modules may then be reused without change in other projects, saving you time and money.

This also facilitates breaking down of projects into several smaller projects so that your implementation will be more easily assembled by large teams. No team members are creating the whole system, or even need to know about the system as a whole, they can focus just on the assigned smaller task. Compared to monolithic applications, our modular system is also easier to maintain as it is designed to allow repair or replacement of parts of the application without requiring wholesale replacement.

Here is a typical buy-side workflow that illustrates how some of our key modules interact. The Market Data Nexus is where the signal comes in to the platform. You can connect to multiple market data sources at the same time. The market data is consumed by the various plug-in modules, such as the UI and Strategy Engine modules. Orders are subsequently created either by a running strategy or by manually entered by a user. Those orders are passed to an instance of DARE, our order routing module, which is responsible for maintaining connections to brokers or other FIX gateways (there can be more than one), and receiving execution reports back from exchanges. DARE will typically run in a client-based datacentre or be collocated with the broker. As the executions flow back from the liquidity destination, their information is presented in the UI module.

The Marketcetera Module framework provides multiple APIs to integrate with your other trading systems such as black box signal generation, real-time trade drop-copy and event dissemination. This modular architecture gives you total control over your trading platform at a fraction of the cost of proprietary commercial software or in-house solutions.

Supported Standards

- Eclipse RCP (Rich Client Platform): Photon User Interface
- Message Bus: Apache ActiveMQ message broker for low latency messaging with high availability and failover
- Security: JAAS (Java Authentication and Authorization Service)
- Persistence: JPA (Java Persistence API)
- Data Storage: RDBMS
- System Monitoring: JMX (Java Management Extensions)
- LBrokerage Connections: QuickFIX/J for FIX (Financial Information eXchange) connections