



Skyler C3 Algorithmic Container Solution

Accelerate your algorithm execution with Skyler's C3 Algorithmic Container Solution.

Key Features:

- Embed custom algorithms and indicator calculations
- Utilize stream processing capabilities
- Access specialized trade, quote, order, last value, and time series caches
- Execute and parameterize logic as part of the data flow or query execution
- Embed third party libraries (e.g. Matlab)
- Execute rapid prototyping via Python scripting
- Use compression to expand data set
- Query system using C3 SQL
- Easily integrate into existing trading infrastructure

Latency: < 1 milli second

Throughput: > 200,000 messages/sec

OS: 32bit, 64bit
Linux RH or
Windows 2003

API server : C++, Python

API client : C++, Java

Skyler introduces the C3 Algorithmic Container Solution. The solution allows brokers and hedge funds to rapidly deploy new and more sophisticated trading algorithms and execute them with unprecedented speed and microsecond latency. The system provides an in-memory framework to embed custom indicator calculations or algorithms on streaming and cached market data. The Skyler C3 Algorithmic Container Solution tackles the market data challenge, allowing the trader to focus on greater trading efficiency of the algorithm.

The benefits for the customer include:

- Significantly faster execution of algorithms
- The ability to deploy novel strategies that leverage a multitude of different real-time market data types across multiple venues
- The ability to deploy new trading strategies more quickly

In addition, Skyler's customers can offer highly parameterized and configurable custom algorithmic offerings to provide a trading edge to its customers.

The Need

The increasing need to remain competitive in the securities industry is forcing companies to re-evaluate their trading infrastructure. Particularly for algorithmic trading, the resurgence of market data coupled with the need for faster, customized, and smarter algorithms to guide trading, call for more powerful trading applications. Current technology is simply not fast or flexible enough to handle the sophistication that is required by algorithmic trading.

The question becomes how to design an algorithmic trading system that is truly useful, meaning it can execute existing and future algorithms without compromises, is general enough to address multiple trading needs and open to allow for highly specific and custom business logic. And perhaps most importantly, the system must seamlessly cope with today's market data challenges.

The answer lies in Skyler's new C3 Algorithmic Container Solution. The Skyler solution deals with a multitude of data types, provides various complementary data processing technologies, and specifically targets challenges of existing algorithmic containers systems. These challenges include the need to embed and customize proprietary algorithms and indicator calculations as part of the trading process. Furthermore, they often don't provide the ability to execute this custom logic in an ad hoc manner with sub millisecond latency. In addition, today's trading applications lack a comprehensive but simple interface to Level I and Level II data types and the required current and time series views. As a result, traders that implement their proprietary algorithms expend unnecessary energy caching, reordering or even aggregating market data, instead of focusing on the trading logic. Thus, the search is not only for faster systems and more sophisticated calculations, but also more standard interfaces to streaming and cached data sources that can be utilized in the execution stages of novel trading strategies.

Skyler C3 Algorithmic Trading Solution

The Skyler C3 Algorithmic Trading Solution takes on these new challenges of the market. It features the ability to:

- Run analytics over dedicated trade and quote data stores
- Run analytics over dedicated order book caches
- Run analytics over other streaming or cached market data
- Cache data using a last value or time series mechanism
- Use standard calculations or build in custom analytics

- Embed custom indicator calculations or algorithms
- Execute business logic as part of the data flow or in an ad hoc manner
- Use Python scripting for fast deployment and rapid prototyping
- Embed third party libraries such as Matlab to execute custom algorithms
- Compress cached data to store more in memory
- Run as standalone server or as embedded shared library
- Use standard adapters and feed handlers to existing Financial Services standards
- Scale on the inbound and outbound side to accommodate for growing data input and distribution requirements

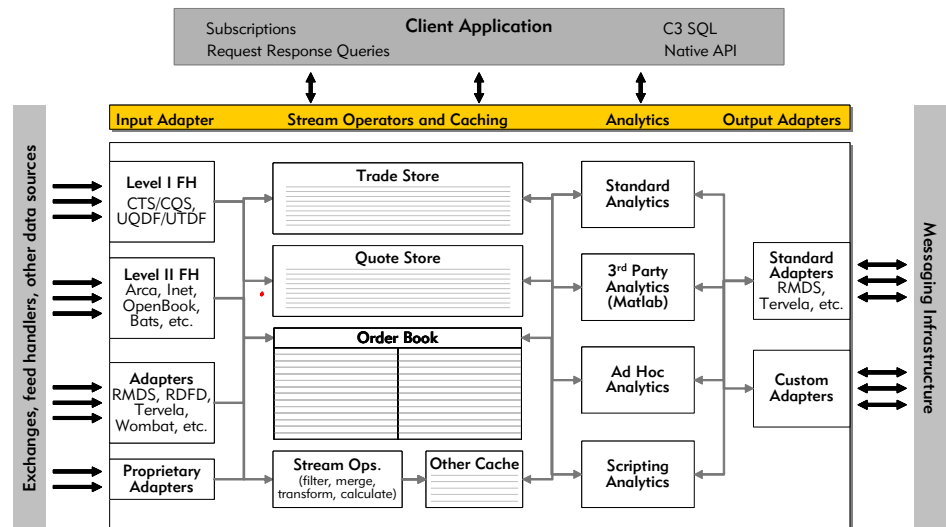


Figure: System diagram of sample algorithmic container solutions

Framework Overview

The Skyler C3 Algorithmic Container works as a very fast in-memory platform for custom algorithms and functions. It integrates with trading systems and market data feeds, provides standard market data analysis and caching, and manages distribution of the events. The remaining task for the users is embedding their core algorithm. In the following description, the main components and functions of the system are outlined.

Input Adapters

The Skyler C3 Algorithmic Container solution is equipped with a number of feed handlers and adapters to existing feed handling infrastructure. C3 can take in the main North American Level I and Level II direct feeds from the exchanges and take charge of the feed handling. The C3 feed handlers include:

- Level I trades and quotes: SIAC CTS/CQS, NASDAQ UQDF/UTDF, and others
- Level II orders: ARCA, NASDAQ TotalView, BATS, NYSE OpenBook, EDGX, and others

Skyler C3 is also open to interface to existing or proprietary feed handling infrastructures including Wombat, Reuters RFD, InfoDyne, and others.

Skyler is also open to take the raw or normalized messages from messaging infrastructure including RMDS, Tervela, 29West, Solace, and others.

Once the data are in the system the user can perform multiple actions:

Stream Processing

Stream processing capabilities can be leveraged to derive value added data, merge data streams, conduct filtering, transformation, or data reduction operations, or apply other logic to the data. Skyler provides a number of predefined stream operators as well as the ability to embed custom stream operators into the data flow.



Caching

In addition, the Skyler C3 Algorithmic Container provides a number of pre-built caching capabilities:

- **Trade Data Store:** The trade data cache stores the trades from the different feeds according to the original exchange timestamp, therefore ensuring that any kind of analytics over the trade data is not influenced by fluctuations in the speed of the data delivery. In addition, the trade data store handles correction messages and other out of bound messages delivered by the exchange. It is optimized to support analytics over the trade data on a per-venue or a cross venue basis. These analytics can include aggregation mechanism such as a VWAP or others. An analytics module can request a specific data window out of the trade data store or simply subscribe to a trade data stream to trigger calculations.
- **Quote Store:** The quote store captures all incoming quotes from the different venues and prepares them for custom calculations or analytics.
- **Order Book:** The order book cache is an extremely fast data processing entity that deals with the different aggregation types of the order book. It maintains the full depth of the order book in real time and supports per venue and cross venue aggregation. The different aggregation mechanisms include no aggregation, so every individual order can be seen, price aggregation on a per venue basis as well as price aggregation across all venues. The order book cache also supports different states accommodating for stale data as well as venue specific flags. In addition, the C3 order book cache can publish both snapshots and updates, thereby preventing the client application from having to rebuild the book from scratch in the event of a lost connection. The order book cache is critical in supporting price discovery in algorithmic trading. The order book cache aggregates orders and sends order book snapshots out in less than 10 microseconds. Performance tests have shown sustained update rates of over 200,000 messages per second on pedestrian hardware. Due to the multithreaded nature of the application increased performance will be achieved with additional and faster CPUs.
- **Last Value Cache:** In addition to the highly specialized trade data, quote, and order book stores the Skyler C3 system also supports a general purpose last value cache that can be updated immediately. This very simple but extremely fast cache supports update rates of over 1 million messages per second.
- **Time Series Cache:** The C3 system also supports a general time series store that focuses on extremely fast appending of time series data. This cache is designed to support very high stream input rates of over 1 million messages per second while supporting fast query rates.

Compression

All of the above are in-memory caching mechanisms only. To further increase the number of records that can be stored, Skyler is developing a proprietary compression mechanism that supports high compression ratios and fast query response.

Real-time Custom Algorithms

With Skyler's Algorithmic Container Solutions, traders are the decision makers. The Solution provides the user with a number of different mechanisms to embed custom business logic, algorithms, or indicator calculations.

The user has three implementation options:

- The custom business logic as stream operators - as part of the data flow allowing the custom logic to react to every message in the data flow. Examples include: standard filtering, aggregation, or transformation operation.
- The custom business logic as modules that subscribe to other modules including any of the caches or stream operators. Examples include custom event detection algorithms that first analyze the order book and if a specific event happens then request the related trade data to finally trigger an order.
- The custom business logic as modules that are configured, parameterized, and executed as part of the query execution. These post processing mechanisms include custom

indicator calculations that have a number of variables and are triggered individually by different traders and with different parameters.

The user has a number of options when developing and using these custom algorithmic modules. The primary mechanism uses Skyler's SDK and a set of Skyler provided base classes to code the algorithm. The base classes provide the user with hooks into the different caching and stream processing entities so the user can focus on the logic of the algorithm. Currently, the Skyler C3 SDK supports C++ to embed custom modules.

Skyler also offers rapid prototyping capabilities that allow the user to use Python scripting language to write the algorithmic logic. This Python script can be embedded as a module in the Skyler C3 Algorithmic Container and allows for a very rapid deployment and high flexibility in testing new algorithms.

In addition, Skyler allows users to embed third party libraries into custom modules. This could include a C++ Matlab library that runs the algorithm defined by the trader.

Querying, Publishing, and Retrieving Results

Once the data flow and data model are defined, the user can query the system and retrieve the events, calculated values, or triggers to issue an order. Results can be published via messaging infrastructure or returned to the client that queries the C3 Algorithmic Container using the C3 Client side API. Skyler C3 currently provides a number of adapters to standard messaging infrastructures such as RMDS, Tervela, and others. The client side API is available in C++ and Java and also supports Skyler C3 SQL to facilitate querying as well as data management. The system supports both push and pull mechanisms to return the results and allows for request-response as well as subscription type queries.

Integration into Trading Infrastructure

The Skyler C3 Algorithmic Container can run as a standalone application or can be embedded into trading applications as a third party library. The system is released on 32bit and 64bit Windows and Linux and is a multi-threaded application allowing it to take advantage of the resources available, both the number of CPU's and speed of the CPUs provided.

Using the off-the-shelf adapters provided by Skyler, ease of integration is ensured. The Skyler C3 SDK and API allow you to build custom adapters, in case the system needs to be integrated with proprietary trading infrastructure, In addition, Skyler provides integration services support if desired.

Conclusion

The Skyler C3 Algorithmic Container Solution removes the latency bottleneck of today's EMS and algorithm execution systems. The solution outperforms traditional systems in speed and flexibility, and provides ease of use, and a rich feature sets. It is focused primarily on providing deeper insights into market data to allow traders to make the most informed decisions in trading.

About Skyler

Skyler is an enterprise software company delivering ultra low latency trading solutions to the financial services industry. The Skyler C3 Solutions source liquidity, analyze tick data, trigger smarter trading decisions, and achieve higher fill rates at better prices. Skyler's customers benefit from faster insights into activities in the market, the ability to deploy innovative new trading strategies, and faster access to the market and therefore high trading efficiency. The Skyler C3 Liquidity Discovery, Tick Analytics, Algorithmic Container, Order Routing, and Compliance Solutions are focused on specific customer needs and emphasize flexibility, ease of use, and simplicity of integration. They combine powerful, low-latency in-memory data processing technologies with market specific business logic modules.

To test drive the Skyler C3 solutions, contact 925.689.9814 or contact@skylertech.com. Please visit us at our website www.skylertech.com.