CAPLIN

Caplin Xaqua 1.0

Overview

November 2009

CONFIDENTIAL

Contents

1 Preface

1.1 What this document contains

This document gives a business and technical overview of Caplin Xaqua.

It contains:

- An explanation of what Caplin Xaqua is and what you can use it for.
- An explanation of the business benefits of using Caplin Xaqua.
- An explanation of the Caplin Xaqua architecture.
- A description of each of the main Caplin Xaqua components.
- Descriptions of the non-functional aspects of Caplin Xaqua, such as resilience, system monitoring and management, and customization.

1.2 Who should read this document

This document is intended for:

- E-commerce Managers
- Business Managers
- Technical Managers
- Enterprise Architects and System Architects
- Software Developers

About Caplin document formats

This document is supplied in three formats:

- Portable document format (*.PDF* file), which you can read on-line using a suitable PDF reader such as Adobe Reader®. This version of the document is formatted as a printable manual; you can print it from the PDF reader.
- Web pages (*.HTML* files), which you can read on-line using a web browser. To read the web version of the document navigate to the *HTMLDoc_m_n* folder and open the file *index.html*.
- Microsoft HTML Help (*.CHM* file), which is an HTML format contained in a single file.
  To read a *.CHM* file just open it – no web browser is needed.

For the best reading experience

On the machine where your browser or PDF reader runs, install the following Microsoft Windows® fonts: Arial, Courier New, Times New Roman, Tahoma. You must have a suitable Microsoft license to use these fonts.

Restrictions on viewing .CHM files

You can only read *.CHM* files from Microsoft Windows.

Microsoft Windows security restrictions may prevent you from viewing the content of *.CHM* files that are located on network drives. To fix this either copy the file to a local hard drive on your PC (for example the Desktop), or ask your System Administrator to grant access to the file across the network. For more information see the Microsoft knowledge base article at http://support.microsoft.com/kb/896054/.

1.3 Related documents

- **Caplin Trader Overview**

  A business and technical overview of Caplin Trader. It explains what Caplin Trader is and how it works with Caplin Xaqua.

- **Caplin DataSource Overview**

  A technical overview of Caplin DataSource.

1.4 Typographical conventions

The following typographical conventions are used to identify particular elements within the text.

Type	*Uses*
XYZ Product Overview	Document name
◆	Information bullet point

1.5 Feedback

Customer feedback can only improve the quality of our product documentation, and we would welcome any comments, criticisms or suggestions you may have regarding this document.

Please email your feedback to documentation@caplin.com.

1.6 Acknowledgments

Adobe® Reader is a registered trademark of Adobe Systems Incorporated in the United States and/or other countries.

Adobe Flex is a trademark of Adobe Systems Incorporated in the United States and/or other countries.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Silverlight is a trademark of Microsoft Corporation in the United States and other countries.

Sun, *Solaris and Java*, are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. or other countries.

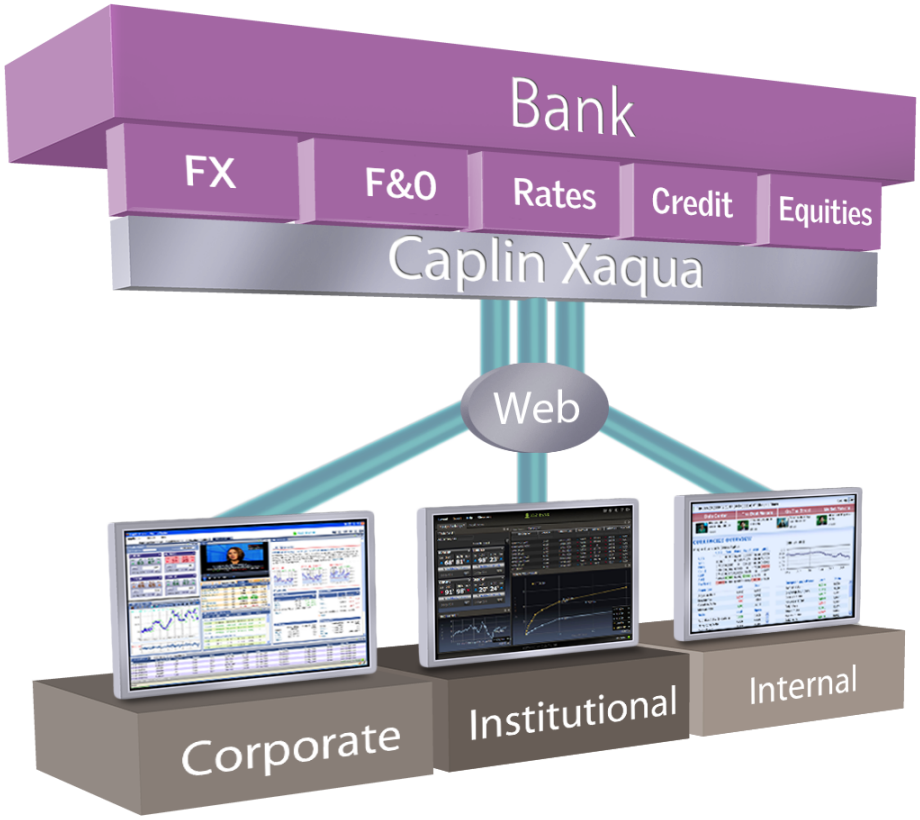
Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Lua is free software from the Pontifical Catholic University of Rio de Janeiro. Lua 5.0 license Copyright © 1994-2007 Lua.org, PUC-Rio.

2 What is Caplin Xaqua?

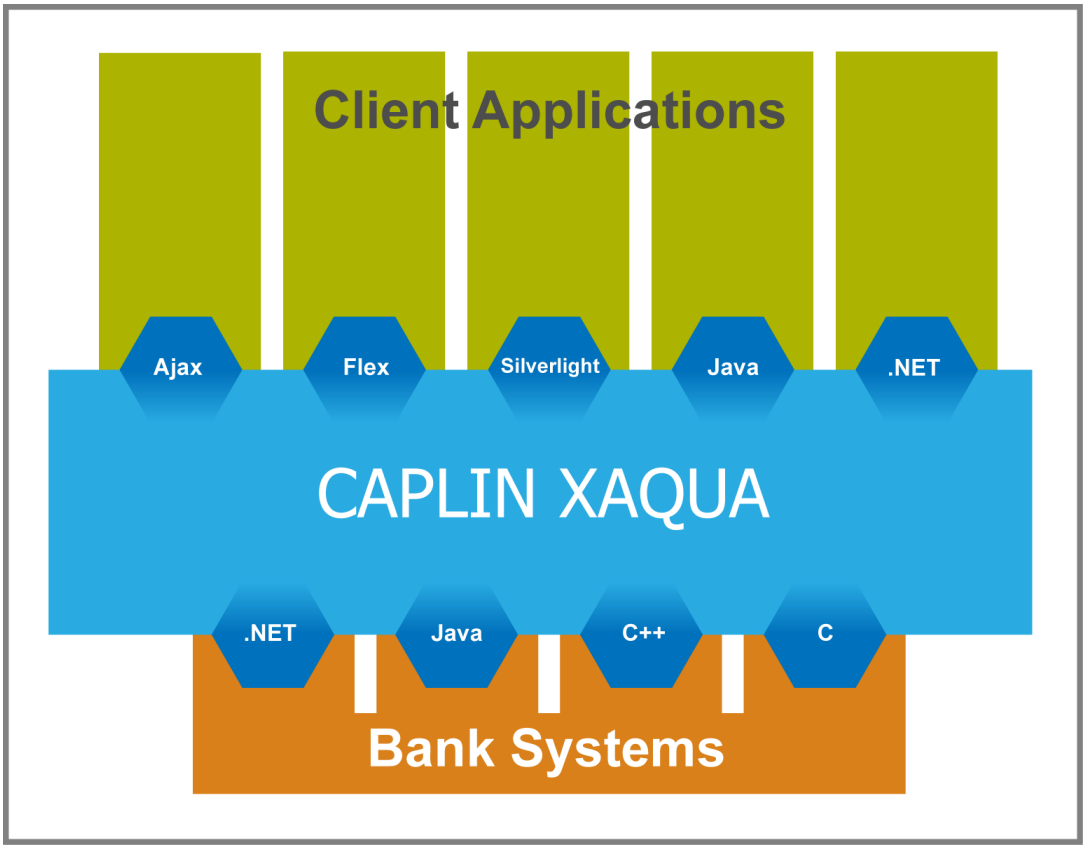
Caplin Xaqua is a single-dealer platform that enables banks to deliver multi-product trading direct to client desktops. It uses third generation 25 financial Web technology, and provides an abstraction layer allowing multiple trading systems and data sources to be connected to multiple client applications.

Caplin Xaqua offers very rapid time to market, low cost, and low risk compared to implementations that are developed from scratch.



A universal single dealer platform

Caplin Xaqua provides a comprehensive set of APIs to support connection to banks' internal systems via C, C++, Java™, and .NET, and to client applications implemented in Ajax, Adobe Flex™, Microsoft Silverlight™, Java, and .NET.



Caplin Xaqua integrates client applications with bank systems

Caplin Xaqua not only integrates clients with bank systems but also ensures that changes in the bank systems do not impact client applications, and vice versa. It does this through an “anti-corruption layer” implemented using Abstract Financial Objects. This layer normalizes all trading, pricing, and permissioning data; the client is decoupled from the bank’s providing systems, while still being built on their models and workflows.

Caplin Xaqua incorporates financial domain-specific features, such as:

- TradePipes – a mechanism for easily implementing trade flows.
- The ability to route, manage, map, and permission market data.
- The application of user entitlements and permissioning in real time.

Caplin Xaqua is built on Caplin’s massively scalable, ultra-low latency, Internet technology. It manages all aspects of real-time data streaming, both to the client and back to the bank's systems. It provides resilience, failover, load-balancing, and latency management. These features ensure that the end user has a high performance trading application suited to their needs, while the bank can rely on continuous connectivity that ensures the most up-to-date prices are available for each trade.

Caplin Xaqua supports a wide range of client-side technologies, including Rich Internet Application (RIA) technologies such as Ajax, Microsoft Silverlight, and Adobe Flex, as well as technologies more suitable for building installed desktop applications such as Java and .NET. Any combination of client and server-side technologies can be used across any Caplin Xaqua applications, accessing the same normalized models and data. This allows you to choose the best technology for a particular job, without having to compromise on the supporting infrastructure or limiting the future choice of technologies.

The anti-corruption layer and Abstract Financial Objects

An anti-corruption layer is a software layer between two systems that allows them to be integrated without explicit knowledge of each other's implementations. The systems typically share the same domain (subsystem) but are implemented according to different specific design models. The anti-corruption layer protects each system from changes made in the other system, and allows one system to be replaced by another if necessary.

Caplin Xaqua provides an extensive anti-corruption layer that separates client applications from bank systems, so the client applications are much less affected by changes to the bank systems, and vice versa.

The anti-corruption layer comprises a set of Abstract Financial Objects that encapsulate the behavior of a particular domain within a trading system. These domains include trade execution, pricing, and permissioning; for more information see Caplin Xaqua subsystems 13. The Abstract Financial Objects provide generalized behaviour; banks can configure this behavior to match their own workflows or processes in each of the domains.

An important feature of Caplin Xaqua's Abstract Financial Objects is that they implement financial data in abstract, normalized formats that are independent of the data representations used in the bank systems. The data types supported include pricing data, trading data, and permissioning data.

TradePipes

TradePipes are a type of Abstract Financial Object. They are Caplin Xaqua's representation of the conversation between a client application and the bank's trading system when a deal (trade) takes place. A single user could have multiple TradePipes open simultaneously to a trading system, or even to multiple trading systems. In a system with many users concurrently trading, Caplin Xaqua will use many TradePipes to manage the deals.

The lifetime of a TradePipe depends on the type of deal; for example:

- A one-click deal would open a TradePipe for at most a few seconds.
- An RFQ deal would open a TradePipe for up to several minutes, as the user requests successive quotes and then closes the deal.
- A resting order (“sell my dollars only when the price reaches 1.60 dollars to the pound sterling”) may use a TradePipe that could be open for a whole trading session, or even across many sessions.

2.1 What can I use Caplin Xaqua for?

Caplin Xaqua enables you to put advanced trading capabilities and real-time information in front of any user anywhere.

You can use it to deliver your current e-commerce offerings via a Rich Internet Application (RIA), without introducing client connectivity problems. Typical deployments of Caplin Xaqua focus on fixed income, forex, equities, and many of their derivatives, as well as cross-asset trading.

Caplin Xaqua can also be used solely for market data distribution; it can connect to sources of market data and distribute the data to a portal in real time. Using Caplin Xaqua components such as Caplin Liberator and Caplin Transformer, you can manage financial market data, add value to it, and distribute it across any network, rapidly and securely, to wherever it is required.

2.2 Business benefits

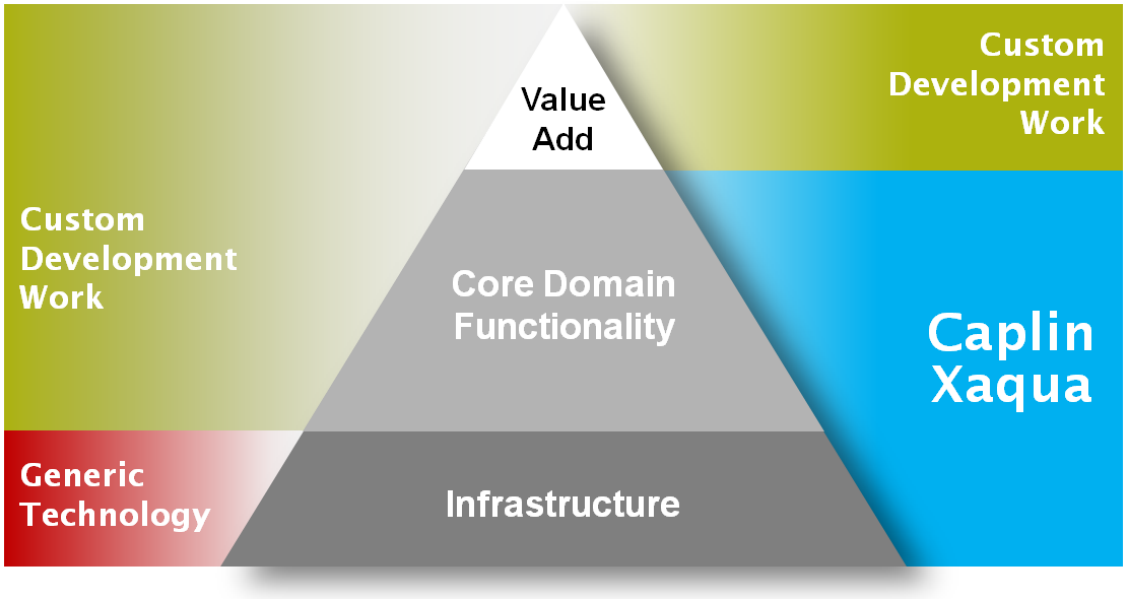
Benefits of single-dealer platforms

The business benefits of implementing a single-dealer platform include:

- Improved deal flow due to targeted offerings and ownership of the user desktop.
- Improved margins through custom pricing.
- Improved customer retention through differentiation and added value.
- Reduced transaction fees compared to multi-dealer portals.
- Avoids competing only on price.

Benefits of Caplin Xaqua's third generation technology

- Caplin Xaqua's third_generation_technology 25 means you are freed to focus the bank's development effort on adding value, as the following diagram shows:



Caplin Xaqua frees up your development effort

So if you implement your trading or market data system using first or second generation generic technology, you must devote resources to developing core business domain functionality.

Caplin Xaqua's third generation technology includes this core domain functionality, so you can focus your scarce development resources on the aspects of your system that add value and differentiate your offering from the competition. You will achieve a faster time to market, while minimizing risk.

Benefits of Caplin Xaqua

Using Caplin Xaqua to implement a single-dealer platform results in:

- Reduced initial time to market.

  As little as three or four months, compared to years for some build-it-yourself systems.

- Lower costs.

  The single-dealer platform is built using a small development team in a short period of time. This makes it affordable if you do not have an enormous in-house development team or a massive budget.

- Lower risk.

  Several major banks have already used this product so you, and we, can be sure that the project will be successful.

- Avoidance of technology lock-in.

  The choice of client-side technology is entirely up to you, and you can use one or more different technologies – that is what Caplin Xaqua is designed for.

- Improved differentiation from competitors.

  Because you do not have to commit your development resources to building the same supporting domain-specific infrastructure as your competitors, you can concentrate your development effort on adding value to your users, and provide them with the information and services that make your organization unique.

- Quick and easy expansion.

  Adding a new asset class is as easy as interfacing the server-side of Caplin Xaqua to the appropriate trading system or systems (we already have off-the-shelf interfaces for some systems and are adding to this library all the time) The interface uses XML configuration to describe the trade workflow(s) for the new asset class.

- Rapid time to market for new products and workflows.

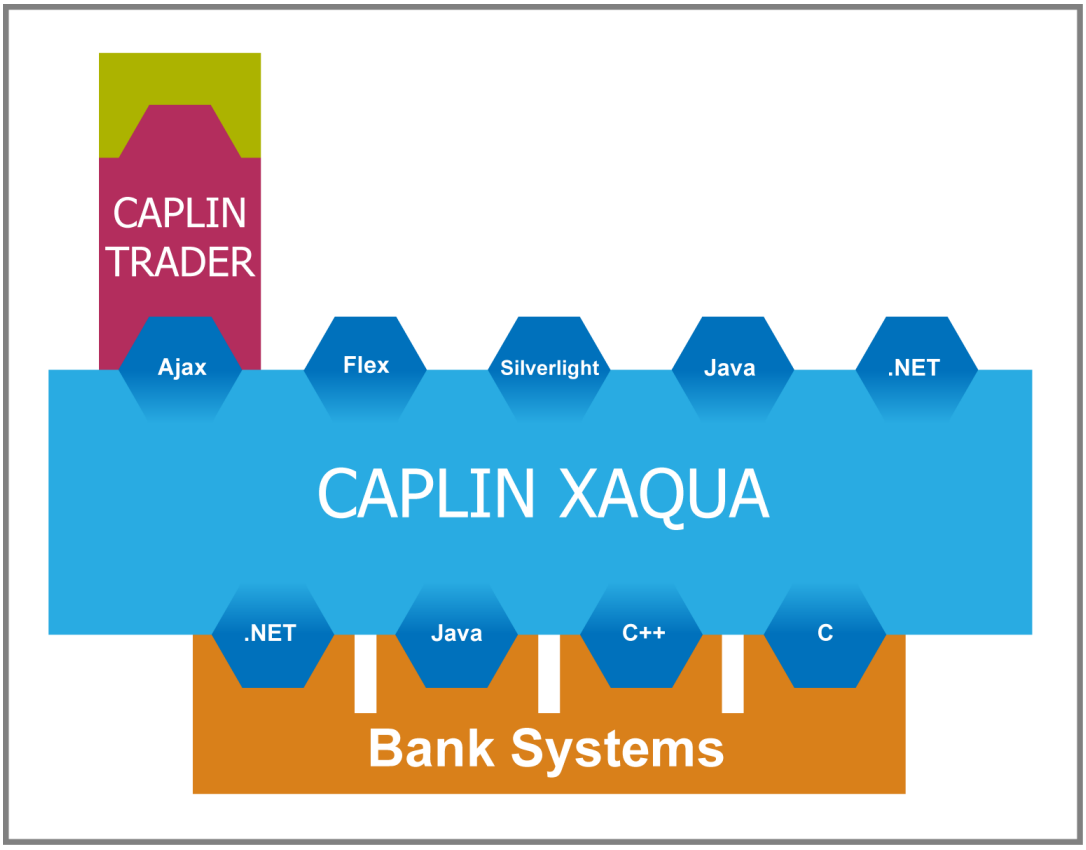
  Even quicker than adding a whole new asset class – just add a trading workflow to the workflow library using XML configuration, and adjust the trading ticket as required.

2.3 What is Caplin Trader?

Caplin Trader is a Caplin Xaqua client application that provides a framework and comprehensive set of components for constructing browser-based trading applications implemented in Ajax. It greatly reduces the time to market when building such an application, leaving the implementer free to concentrate on providing unique features and added value.

Caplin Xaqua allows you to develop client applications using standard client-side technologies such as Ajax, Adobe Flex, Microsoft Silverlight, Java, and .NET. However, if you wish to deploy a trading client that executes in standard Web browsers using Ajax technology, you will find it convenient and cost-effective to base your solution on Caplin Trader.

For more information about Caplin Trader, see the **Caplin Trader Overview**.



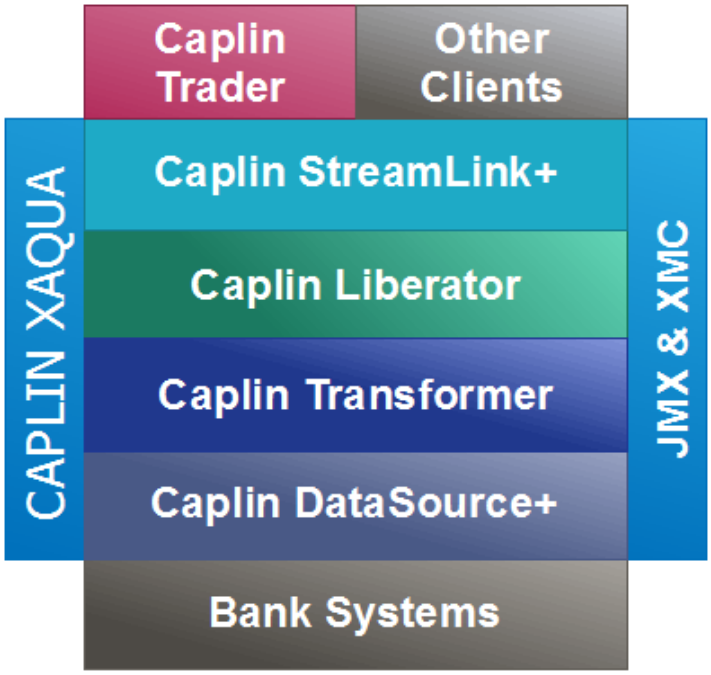
Caplin Trader and Caplin Xaqua

A typical Caplin Trader screen

3 The Caplin Xaqua architecture

The following sections explain the architecture of Caplin Xaqua – its major components and subsystems.

3.1 The Caplin Xaqua technology stack



Caplin Xaqua technology stack

Caplin StreamLink+

Caplin StreamLink+ is the interface between client applications and Caplin Xaqua. Its APIs provide access to the Caplin Xaqua's financial domain functionality, allowing a client application to interact with the bank's systems, while decoupling it from the implementation specific features of those systems.

Caplin StreamLink+ APIs are available in Ajax, Adobe Flex, Microsoft Silverlight, Microsoft .NET, and Java, so you can implement client applications in your preferred technology and still reap the benefits of using Caplin Xaqua.

Caplin Liberator

Caplin Liberator is a real-time financial internet hub that delivers trade messages and market data to and from subscribers over any network, tunneling automatically through proxy servers and firewalls. Its ultra-high-performance publishing engine can send millions of updates per second at low latency from a single sever. It quickly and securely routes both price data and trade messages between client applications and bank systems.

Caplin Transformer

Caplin Transformer is an event-driven real-time business rules engine. Standard modules provide functions such as caching, data normalization, filtering of grid like displays on behalf of client applications, and calculation of FX cross rates. Transformer's software development kits (SDKs) for Java, C and Lua make it easy to construct custom modules to fit specific business needs, and modules can be pipelined to accomplish complex tasks.

Caplin DataSource+

Caplin DataSource+ is the interface between Caplin Xaqua and the bank's systems. It plays a key role in implementing Caplin Xaqua's anti-corruption layer; for example, its domain specific APIs help cushion the rest of Caplin Xaqua and the client applications from changes in the bank's systems.

Caplin DataSource+ is used to implement DataSource adapters that transform price data, permissioning data, and trade messages from the bank's systems, into generic DataSource messages that can be understood by other Xaqua components. DataSource adapters also convert outgoing DataSource messages into the formats required by the bank's systems.

Caplin DataSource+ APIs are available in Java, C++, and .Net.

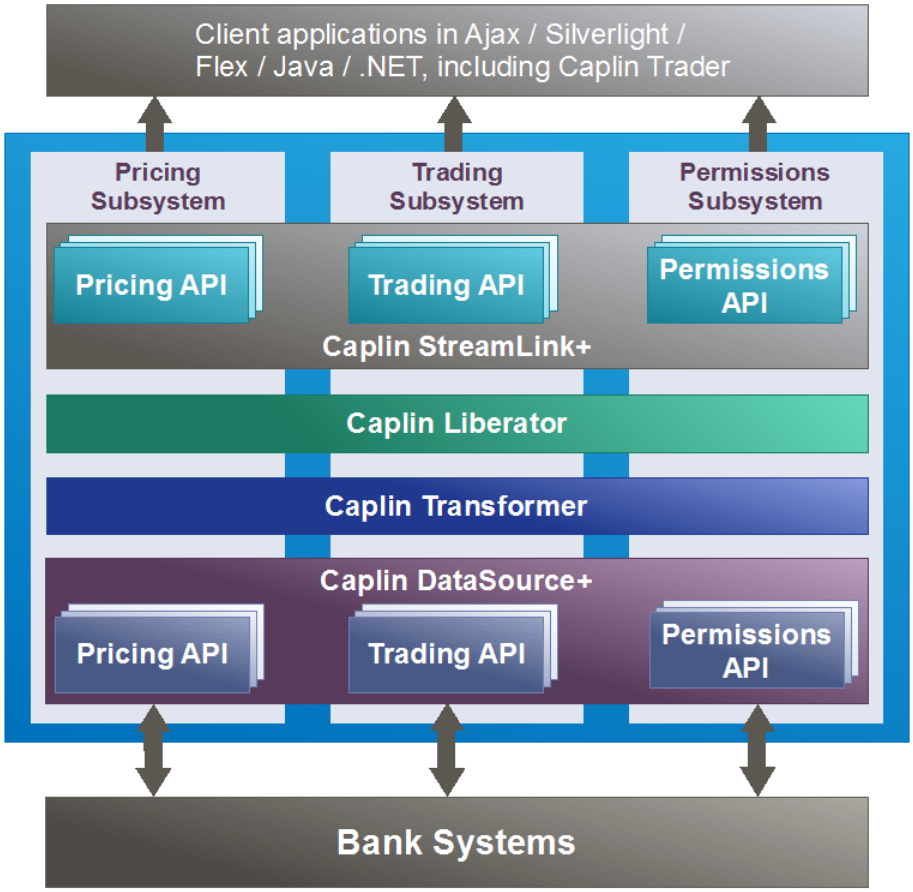
JMX & XMC

Applications built with Caplin’s DataSource+ APIs can be enabled for monitoring and management, including Caplin Liberator, Caplin Transformer, and all standard DataSource adapters. Java Management Extensions (JMX) provide an API, based on Java MBeans, for building monitoring and management capabilities into Caplin Xaqua components.

The Caplin Xaqua Management Console (XMC) is a Java application that communicates with Caplin Xaqua components via JMX, and allows you to monitor and control these components through a GUI interface.

3.2 Caplin Xaqua subsystems

Caplin Xaqua implements core financial domain models through its subsystems, as shown in the following diagram. Each subsystem is implemented as a set of Abstract Financial Objects.



Caplin Xaqua Architecture showing subsystems

Pricing Subsystem

The Pricing Subsystem obtains instrument prices in real time from multiple disparate sources and normalizes them so they are suitable for display by client applications and can be easily filtered and sorted. It implements a standard but flexible domain model for obtaining, processing, and displaying price information and other market data.

The **Pricing API** in **Caplin StreamLink+** allows client applications to subscribe to financial instruments, and receive continuously updated pricing data and other information about the instruments in real time. Clients can subscribe to individual instruments, or more usually, to sets of instruments (for example Major FX currencies, US Treasury Bonds). Using this API, clients do not need to know where the subscribed data comes from or what format it originally took; these details are handled by Caplin Xaqua through its intelligent flow control and Abstract Financial Objects.

Caplin Liberator is at the heart of the Pricing Subsystem. It handles the clients' connections to Caplin Xaqua, manages the subscriptions to prices and other data, and streams the data updates to the clients in real-time.

If required, business rules (implemented as **Caplin Transformer modules**) can be used to normalize received price data (in addition any normalization done through the Pricing API), and create derived data such as price spreads, cross rates, and volume weighted average prices.

The **Pricing API** in **Caplin DataSource+** is used to implement **Caplin DataSource adapters** ("Pricing DataSources") that interface with the sources of price information, such as the bank's pricing systems and external data feeds. These adapters convert the raw price data into standard formats suitable for the other Caplin Xaqua components and client applications, and provide the ability to group and segment data using containers. This means that clients do not themselves have to handle the disparities in data formats across the various price sources.

Trading Subsystem

The Trading Subsystem allows users of client applications to trade the financial instruments displayed on their screens. It handles the product workflow – trade transactions and associated message flows – between the client and the bank's trading systems.

Trading activity is implemented through TradePipes. A TradePipe implements a model of the workflow for a particular type of trade, such as a one-click deal or a Request For Quote (RFQ). The model is specified in XML as a state machine; at run time the TradePipe automatically synchronizes the workflow between the client and bank ends of the TradePipe, exchanging the correct trade messages and handling any error conditions that may occur.

You can easily configure Caplin Xaqua to handle a new trade workflow just by defining the trade model in XML to match the workflow in the relevant bank trading system. It then remains for you to modify the client applications to present the user with the trade tickets, other UI controls, or API access, that allow the user to trade using the new workflow.

The **Trading API in Caplin StreamLink+** handles the interface between TradePipes and the client application. For example, this is where you specify the Caplin Xaqua trade model that is associated with a particular trading control in the UI (such as a particular type of trade ticket), or expose the model to a custom API. The library code underlying the API is responsible for loading and applying the trade model at the client end of the TradePipe.

Caplin Liberator is the router for TradePipes, passing trade messages between clients and the relevant Trading DataSources.

The **Trading API** in **Caplin DataSource+** is used to implement **Caplin DataSource adapters** ("Trading DataSources") that interface with the bank's trading systems and liquidity services. A Trading DataSource handles the bank's trading system end of the TradePipe, applying the trade model associated with the TradePipe, and coordinating the workflow with the bank trading system to which it is attached. This coordination includes converting the trade messages received from clients into a format that the particular bank trading system will understand, and converting trade messages returned from the trading system into Caplin Xaqua's normalized format, before passing them on to clients.

Permissions Subsystem

The Permissions Subsystem manages authentication of users, their access to data, and their trading capabilities. It implements a generic permissions model that embodies a number of concepts:

- Users, including user groups and accounts that allow an inheritance tree of permissions to be created.
- Permissioning data, identifying both the users that are authorized to interact with Caplin Liberator and client applications, and the financial products that these users are permitted to view and trade.
- Products – for example, “All FI Instruments”.
- Mappings, which can assign Users and Products to spread groups, amount tiers, or customized pricing.
- Actions, defining the interaction that a user can have with a product – for example, “RFQ trading”.
- Permissions that determine whether an action on a product will be allowed or denied.
- Rules that link permissions to user interactions.

This model is flexible, so you can quickly integrate Caplin Xaqua with your existing permissioning systems, or even use it to implement a new permissioning regime that meets your exact business requirements.

The **Permissions API** in **Caplin StreamLink+** handles the interface between the Permissions Subsystem and the client application. Using this API, the behavior of the client application can be dynamically modified to match the permissions of the user who is currently logged in to Caplin Trader Client. An example of this is to display information that the user is allowed to view, and to hide information that the user is not allowed to view. This information could be anything from pricing data for currency-pairs, to menu items, data grids, trade tiles, and tenors.

Caplin Liberator forwards permissioning information to clients via the Caplin StreamLink+ Permissions API. Caplin Liberator's Permissioning Authorization module locally manages access to cached data, according to the permissions granted to Liberator users.

Caplin Liberator enforces security by requiring end-users to log in. In a bank's typical security regime, users are also authenticated when they sign on to the client application. It is much more convenient for users if they can sign on just once, with the client application handling the additional Liberator login in the background. Caplin Xaqua allows you to do this in a highly secure manner through the Caplin KeyMaster component, which integrates Caplin Liberator with the bank's existing single sign-on system.

The **Permissions API** in **Caplin DataSource+** is used to implement **Caplin DataSource adapters** (“Permissioning DataSources”) that implement the permissioning model. The Permissioning DataSource supplies Caplin Liberator's Permissioning Auth Module with permissioning data, providing updates in real time as the data is changed. A Permissions DataSource can be implemented as the master permissioning system, or it can interface to the bank's existing permissioning system(s). Caplin Xaqua supports multiple Permissioning DataSources, which makes it easier to retrieve permission information from multiple asset-class permissioning systems.

3.3 Software environment

Caplin Xaqua components are designed to run in a variety of software environments.

- DataSource+ adapters can be implemented in several programming languages product (.NET, Java, C++, C) and run on platforms appropriate to those languages.
- StreamLink+ can be used in client applications that run in several different software environments: Ajax, Adobe Flex, Microsoft Silverlight, Microsoft .NET, and Java.
- Caplin Transformer and Caplin Liberator run under the Linux® and Sun® Solaris™ operating systems.

Note: For detailed information on software environment requirements (operating system versions, and so on), please refer to the Caplin technical documentation for the individual Caplin Xaqua components.

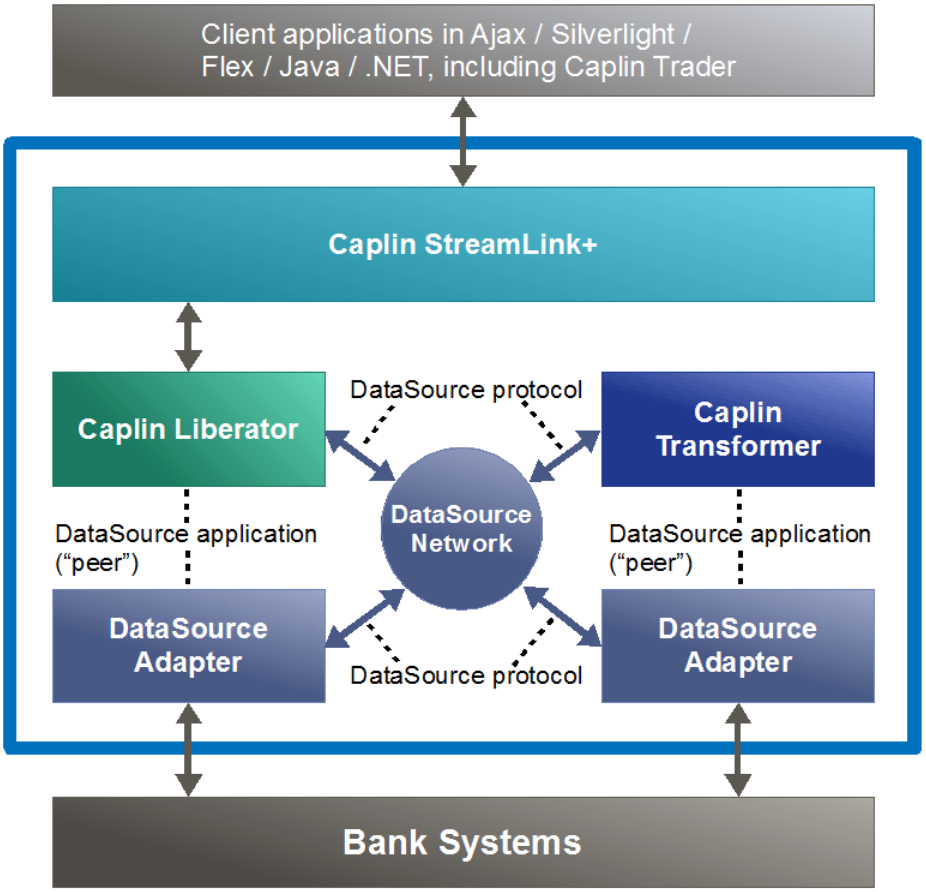
4 More about the Caplin Xaqua components

The following sections contain additional information about the Caplin Xaqua components.

4.1 DataSource+

Caplin DataSource+ is the interface between Caplin Xaqua and the bank systems. It consists of APIs and their underlying code libraries that support the domain specific (subsystem) interfaces described in Caplin Xaqua subsystems 13 – the Pricing, Trading, and Permissions subsystems. The business domain knowledge built into the Abstract Financial Objects used by these APIs allows you to quickly and easily integrate Caplin Xaqua with your existing bank systems, and develop additional trading functionality for your customers. As an integral part of Caplin Xaqua's anti-corruption layer, Caplin DataSource+ also cushions the rest of Caplin Xaqua and the client applications from changes in the bank systems.

At a lower level, DataSource+ also supports the “DataSource” communications infrastructure that links Caplin Xaqua components to each other and to the bank's systems. This is shown in the following diagram:



DataSource communications infrastructure within the Caplin Xaqua Architecture

The term 'DataSource' is used in several related contexts:

- The **DataSource network** is the messaging network used within Caplin Xaqua.
- The **DataSource protocol** is the protocol used by this messaging network.

  This is a bidirectional protocol specifically designed to support real-time financial messaging, including the transmission of market data and financial trading messages. The protocol supports both broadcast and subscription models of message propagation.
- **DataSource applications** (**DataSource "peers")** are applications that can communicate using the DataSource protocol. They include Caplin Liberator and Caplin Transformer, and DataSource adapters.
- **DataSource adapters** act as the interface between the bank's systems and Caplin Xaqua.
- The **DataSource+ APIs** are used to write custom DataSource adapters, such as the Pricing DataSources, Trading DataSources, and Permissioning DataSources.

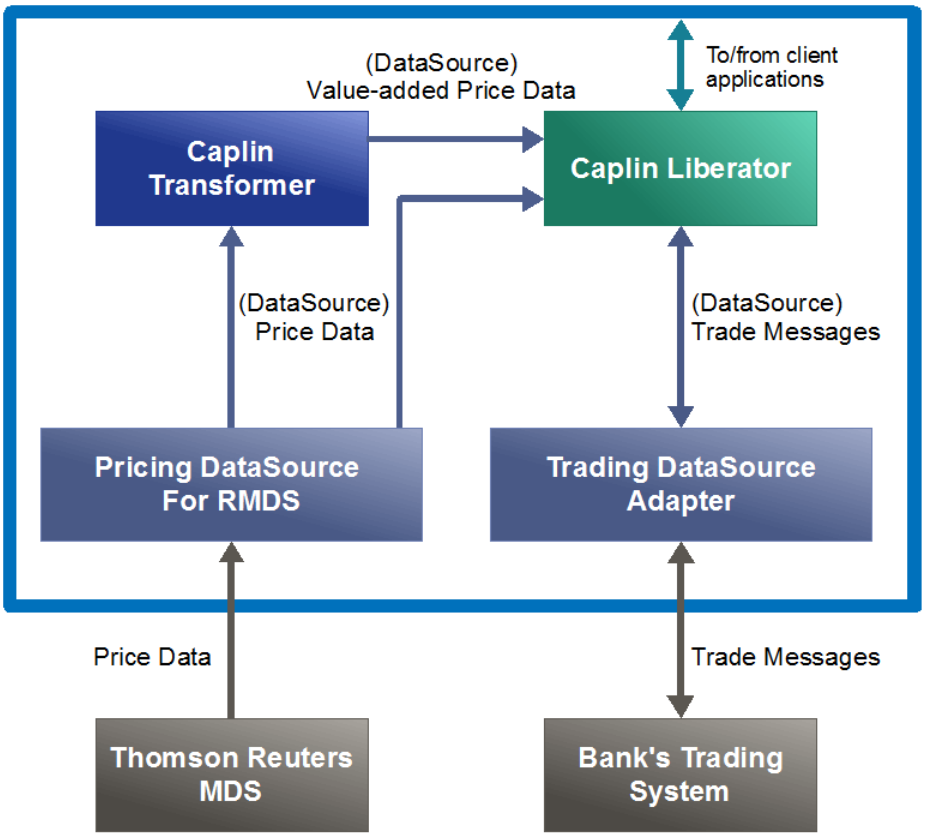
For more technical information about DataSource concepts and implementation see the **Caplin DataSource Overview**.

DataSource adapters

Caplin Xaqua components such as Caplin Liberator and Caplin Transformer normally need to communicate with external sources and sinks of data that are located in the bank's systems; they do this via DataSource adapters.

A DataSource adapter transforms incoming data into DataSource messages that can be understood by other Caplin Xaqua components, and converts outgoing DataSource messages into the formats required by the other systems. This is illustrated in the following diagram, which shows the back end of a trading system with two DataSource adapters. The DataSource for RMDS adapter obtains indicative price information from a Thomson Reuters MDS feed and passes it to Caplin Transformer. A **custom Trading DataSource adapter**, implemented using the DataSource+ Trading API, passes trade messages between Caplin Liberator and the bank's trading system.

Note: In Caplin product documentation DataSource adapters are often just called "DataSources".



DataSource adapters in an single-dealer platform implementation

Off-the-shelf DataSource adapters

Caplin can supply a number of off-the-shelf DataSource adapters for accessing existing market data distribution systems, vendor feeds, messaging middleware, and databases. These include:

- DataSource for Thomson Reuters Market Data System (RMDS)
- DataSource for TIB/RV
- DataSource for Databases

The **DataSource for Databases** adapter provides a mechanism for moving data between a JDBC™ compliant database and Caplin Xaqua. It includes a comprehensive XML-based configuration facility, allowing you to implement fully functioned DataSource applications that access databases, with no need to write code.

DataSource Failover

To ensure continuity of service, a DataSource application can be configured so that it is connected to two or more DataSource peers. When it loses its connection to the first peer the DataSource application automatically fails over to use the second peer. The failover capability is provided through a DataSource facility called data services. You can configure any number of alternate failover DataSource peers in priority order.

For more information see the **Caplin DataSource Overview**.

4.2 Caplin Liberator features

Caplin Liberator is a real-time financial internet hub that delivers trade messages and market data to and from subscribers over any network, tunneling automatically through proxy servers and firewalls.

Client applications communicate with Caplin Liberator via StreamLink+, enabling them to send and receive data updates, and trade messages across TradePipes, in real time with low latency. Liberator in turn uses the DataSource protocol to exchange data and trade messages with other components of Caplin Xaqua, such as Caplin Transformer and the DataSource adapters 18 that act as gateways to other systems.

Caplin Liberator contains a high performance publishing engine capable of delivering millions of updates per second from a single server. It also provides standard Web server functionality to clients using HTTP and HTTPS connections.

Data and subscription management

Caplin Liberator's data and subscription management features are a key part of Caplin Xaqua's anti-corruption layer and data normalization capability. Caplin Liberator supports a wide variety of data types and formats, including structured records, trade messages, pages, news headlines, news stories, time series, alerts, and chat.

On subscribing to a data object managed by Caplin Liberator, a client application will receive an initial image of that object followed by an update whenever it changes. Subscriptions can be restricted to just part of an object (such as specific fields in records), and clients can specify filters so as to receive only updates matching certain criteria.

Caplin Liberator organizes data into groups and hierarchies, using the concept of a “directory” object to which a client can subscribe so as automatically to receive all its contents. In addition, “container” objects allow clients to subscribe to dynamically managed collections of objects through references. This facility is particularly useful for subscription to volatile lists, such as index constituents, order books, and most-actives, and for handling grid-like displays of financial instruments.

Caplin Liberator offers sophisticated caching capabilities. As well as automatic last-value caching for all objects, it can (where required) cache multiple levels of records, which is useful for “level 2” quote data, time-series replay, and intraday news histories. It can be configured to purge the cache at pre-set times, on a per-object basis.

“Container paging” allows client applications that subscribe to long lists of records to specify a “window” of content that is currently visible on the screen, so the application is only sent updates for data that it is actually displaying.

Data mapping

Caplin Liberator obtains the objects requested by clients by subscribing in turn to appropriate objects from sources such as Caplin Transformer and DataSource adapters. Powerful data mapping capabilities allow generic object names requested by clients to be mapped to user-specific object names during this process. This capability is important for business features such as price tiering.

Resilience

Caplin Liberator is designed for high availability and resilience in the face of network or server failures.

- Continuous heartbeats between clients and Liberator, and between Liberator and DataSources, guarantee that connection loss is handled properly, even when the operating system does not close the connection.
- Caplin Liberator will fail over to an alternate DataSource if an existing DataSource becomes unavailable. It uses a highly configurable failover strategy.
- Clustered Liberators are grouped into failover configurations, so if a Liberator becomes unavailable its clients will automatically fail over to an alternate Liberator.
- Global cache synchronization ensures that, in the event of such a failover, the alternate Liberator already has the required data in cache, minimizing failover time and preventing overload.

4.3 Caplin KeyMaster

Caplin KeyMaster is software that integrates Caplin Xaqua with any existing authentication system, so that end users or web applications do not need to explicitly log in to Caplin Liberator in addition to their normal login procedure. It implements a secure method of user authentication by means of a user credentials token that is digitally signed using public key encryption.

KeyMaster works in conjunction with other Caplin Xaqua components (Liberator Auth modules and StreamLink+) and third party software/customer developed software, including single sign-on and user permissions systems, web applications, and web application servers.

4.4 System monitoring and management

Caplin Xaqua contains full enterprise-wide monitoring and management capability.

Monitoring facilities

Applications built with Caplin's DataSource+ APIs can be enabled for monitoring and management, including Caplin Liberator, Caplin Transformer, and all standard DataSource adapters. Every DataSource application provides a minimum standard set of information to the monitoring system, including process information, connection and peer information, and access to log messages. Caplin Liberator and Caplin Transformer also provide additional component specific information.

When writing your own DataSource applications, you can use standard DataSource+ API functions to easily add new items for monitoring and management .

Caplin Xaqua's monitoring and management capabilities are provided by a plug-in monitoring module, which is loaded at runtime by the DataSource process. Currently there are two monitoring modules available: SOCKMON provides socket-based monitoring and management using a simple command language, and JMX provides Java MBean-based monitoring and management for Java clients using a built-in JMX Server. Both of these modules can be used to integrate existing monitoring systems with Caplin Xaqua.

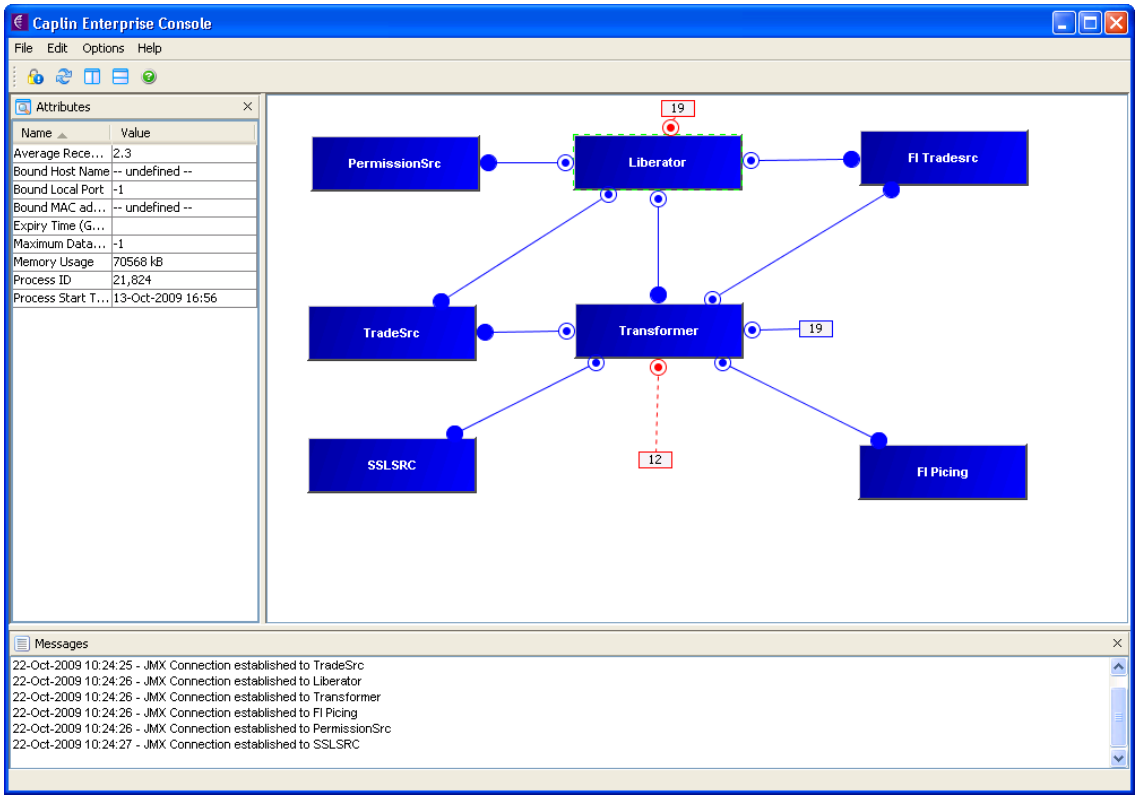
You can monitor and manage Caplin Xaqua using the **Caplin Xaqua Management Console**, which communicates with Caplin Xaqua components via JMX. Its GUI interface allows you to easily monitor Caplin Xaqua from a high-level view, and also to drill-down to individual components and specific items within components.

Caplin Xaqua Management Console (XMC)

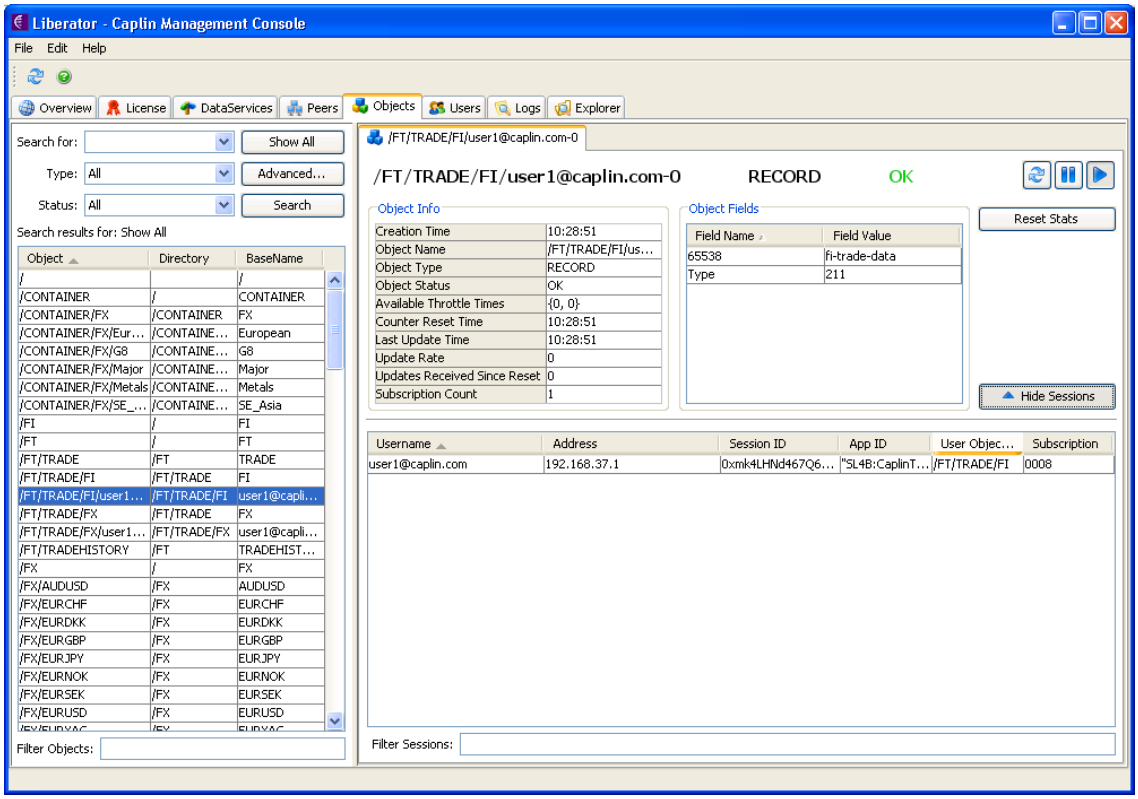
The Caplin Xaqua Management Console (XMC) is a Java application that communicates with Caplin Xaqua components via JMX, and allows you to monitor and control these components through a GUI interface.

The XMC has a core set of screens that provide an overview of all system activity. You can drill down into the status of server components by opening individual tabbed console windows for each server. These tabs are configurable through XML files; this allows you to customize the console by adding your own tabs. You can also implement custom views in Java, using the provided SDK.

Additionally the XMC provides a generic JMX explorer that can be used to view all the JMX MBeans exposed by the individual components. This facility can be used by power users or Caplin engineers to help diagnose problems in live systems.



Caplin Xaqua Management Console – Network Overview screen



Caplin Xaqua Management Console –
detail screen showing Liberator information

Logging

Caplin Xaqua components such as Caplin Liberator, Caplin Transformer, and off-the-shelf DataSource adapters, all produce log files containing comprehensive information about their run-time activities, for both operational and diagnostic use. All deals and trades are fully audit logged at each point in the system, ensuring full traceability of the whole deal life cycle.

The level of detail that is logged is in most cases configurable. Many log files are cyclic, so the log file is automatically closed at a defined (configurable) time, and a new log is then opened.

5 Customization

Typical deployments of Caplin Xaqua often need to be customized in some way. For example, you may need to integrate Caplin Xaqua components with your existing systems, and you may also need to implement behaviour that is specific to your organization's business requirements.

Most Caplin Xaqua components are highly configurable. Some configuration enables components to integrate and communicate with each other; for example, the configuration of DataSource peers. Other configuration changes the behaviour of the component; for example, you can define new Trade Models in XML that will be executed by a Trading DataSource adapter and by client applications.

Many Caplin Xaqua components have associated SDKs, with APIs that enable you to modify and extend the components' behaviour. For example DataSource+ APIs, such as the Trading API, allow you implement custom DataSource adapters that will interface with the bank's existing systems and third-party systems – see Caplin Xaqua subsystems 13.

6 What is a third generation platform?

Caplin Xaqua is a *third generation* platform for delivering multi-product trading. To understand what this means, and what its advantages are, we need to consider how the third generation differs from its predecessors – first and second generation platforms.

First generation financial trading platforms are basically messaging systems that have no domain awareness. They merely pass messages containing name–value pairs between the bank systems and client applications, often by means of a streaming web server. This leaves the client applications to implement all the business logic that interprets and acts on these messages, so that significant time and cost must be spent on developing and maintaining the client applications.

Second generation platforms improve on the first generation, in that they are object based. The client applications and bank systems operate on, and communicate through, software objects. A client typically holds copies or representations of objects that originate on the bank system, and/or accesses objects on the bank system through remote object calls (this technique is often termed "object remoting"). Unfortunately, this approach tends to tightly couple the client application to the bank systems; for example, if anything changes on the server side the object has to be modified or the client application will break. This coupling makes it more difficult and costly to develop alternative client applications and to hook existing client applications into alternative/additional bank systems.

Third generation financial platforms, such as Caplin Xaqua, are much more than just message passing and object management frameworks. They incorporate knowledge and functionality relating to the core financial business domain. They also implement an anti-corruption layer that decouples client implementations from the bank systems. These features mean that:

- It is much easier and less costly to develop and maintain client applications.
- It is much easier to integrate alternative client applications with the bank systems.
- Clients can be implemented in many different technologies (Ajax, Microsoft Silverlight, Adobe Flex, and so on), without impacting the bank systems.
- Changes to the bank systems have much less impact on client applications; for example a particular data feed could be entirely replaced by a different one that supplies the same business data, and there would be no impact on the clients.
- The bank can focus its resources on adding value and delivering its own services to its users through a single-dealer portal.

The key point is that the third generation of financial web technology is about financial applications and the core domain functionality needed to implement them, rather than being about generic technology that can be used in a variety of different vertical segments.

7 Glossary of terms and acronyms

This section contains a glossary of terms, acronyms, and abbreviations, relating to Caplin Xaqua and the technologies it uses.

Term	Definition
Abstract Financial Object	Within Caplin Xaqua, Abstract Financial Objects encapsulate the behaviour of a particular domain (subsystem) within a trading system. These domains include trade execution, market data and price information, and user entitlements (permissioning). An object is “abstract” in that it represents a generalized model of a domain, which can then be configured by a bank to accurately reflect the way that a particular organization manages and executes its data and trade flow. Caplin Xaqua's **anti-corruption layer** is implemented as a set of Abstract Financial Objects. Also see “The anti-corruption layer and Abstract Financial Objects 6” in What is Caplin Xaqua? 4
Adobe Flex	A software development kit and integrated development environment from Adobe Systems Inc. for creating **Rich Internet Applications** based on the Adobe Flash platform. Flex uses ActionScript code and **XML**-based user interface descriptions (MXML) and compiles them into binary Flash files (SWF files).
Ajax	Asynchronous **JavaScript** and **XML** A combination of Web technologies used to implement interactive Web clients.
Anti-corruption layer	“Anti-corruption layer” is a term used in **domain-driven design** to describe an isolation layer between systems that allows the systems to be integrated without explicit knowledge of each other's implementations. Caplin Xaqua provides an “anti-corruption layer” that *decouples* the bank's trading, pricing, and permissioning systems from the operation of the client applications that use those systems. As a result, the client applications are much less affected by changes to the bank's systems, and vice versa. The anti-corruption layer in Caplin Xaqua is implemented as a set of **Abstract Financial Objects**. Also see “The anti-corruption layer 6 and Abstract Financial Objects” in What is Caplin Xaqua? 4
Authentication	The process of verifying the identity of a user, for example by checking a user name and password that the user supplied when attempting to log in.
Authorization	An alternative term for **permissioning**.
Auth module	A software module that handles **authentication** and **permissioning** within Caplin Liberator
API	Application Programming Interface.
DataSource	Synonym for **DataSource application**.
DataSource adapter	A **DataSource application** that integrates with an external (non-

Term	Definition
	Caplin) system, exchanging data and/or messages with that system.
DataSource application	A Caplin Xaqua application that uses the DataSource+ APIs to communicate with other Caplin Xaqua applications via the **DataSource protocol**. Also known as a **DataSource peer**.
DataSource peer	See **DataSource application**.
Domain-driven design	An approach to software design that models the domain that the software is to represent. It is primarily focussed on the characteristics of that domain, rather than the technology used to implement the software. For more information, see the Wikipedia article on domain-driven design and the Domain-Driven Design Community site http://domaindrivendesign.org/.
Failover	A technique to support software resilience whereby, when an application loses the service provided by a server, it reconnects (fails over) to an alternative server in order to minimize the interruption to the service.
Flex	See **Adobe Flex**.
JavaScript	A scripting language used in Web browsers.
JMX	Java Management Extensions A Java technology for application and network management.
Microsoft Silverlight	Silverlight is Microsoft's framework for developing and deploying Rich Internet Applications.
Permissioning	The process of determining the access rights that a user has to resources, such as data and functionality provided by the software. Also known as **authorization**.
RIA	See **Rich Internet Application**.
Rich Internet Application	A web application, such as Caplin Trader, that has the features and functions of a desktop application, but which does not need to be installed on the client platform. Rich Internet Applications typically run in web browsers.
SDK	Software Development Kit.
Single sign-on	A user authentication process in which a user supplies just one set of user credentials (such as a user name and password). The user can then access multiple applications and systems without being prompted for credentials again.
Silverlight	See **Microsoft Silverlight**.
Trade model	A Trade Model represents the workflow when trading a financial instrument. Examples are Request for Quote (**RFQ**) and Executable Streaming Price (**ESP**). Trade Models consist of a number of states and transitions between the those states.
TradePipe	TradePipes are a type of **Abstract Financial Object**. They are Caplin Xaqua's representation of the conversation between a client application and the bank's trading system when a deal (trade) takes place. Also see "TradePipes 6" in What is Caplin Xaqua? 4
UI	User interface.

Term	Definition
XMC	Caplin Xaqua Management Console. In older Caplin documentation, the XMC is called the EMC (Enterprise Management Console). Also see System monitoring and management 21.
XML	Extensible Markup Language.

Contact Us

Caplin Systems Ltd
Triton Court
14 Finsbury Square
London EC2A 1BR
Telephone: +44 20 7826 9600
Fax: +44 20 7826 9610
www.caplin.com