

## Caplin Platform Overview

### 4.4

October 2009 ■ Confidential



# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Preface.....</b>                          | <b>1</b>  |
| 1.1      | What this document contains.....             | 1         |
| 1.2      | Who should read this document.....           | 1         |
| 1.3      | Related documents.....                       | 1         |
| 1.4      | Feedback.....                                | 1         |
| 1.5      | Acknowledgments.....                         | 2         |
| <b>2</b> | <b>What is the Caplin Platform?.....</b>     | <b>3</b>  |
| 2.1      | What can I use the Caplin Platform for?..... | 3         |
| 2.2      | Business benefits.....                       | 4         |
| <b>3</b> | <b>Platform architecture.....</b>            | <b>5</b>  |
| 3.1      | Online trading architecture.....             | 5         |
| 3.2      | Market data distribution architecture.....   | 7         |
| <b>4</b> | <b>Software environment.....</b>             | <b>8</b>  |
| <b>5</b> | <b>Platform components.....</b>              | <b>9</b>  |
| 5.1      | DataSource.....                              | 9         |
|          | DataSource adapters .....                    | 10        |
|          | Off-the-shelf adapters .....                 | 11        |
|          | DataSource SDKs .....                        | 11        |
| 5.2      | RTTP and StreamLink.....                     | 11        |
| 5.3      | Liberator.....                               | 12        |
|          | Data and subscription management .....       | 12        |
|          | Data mapping .....                           | 13        |
|          | User management .....                        | 13        |
|          | Performance and scalability .....            | 13        |
|          | Resilience .....                             | 14        |
| 5.4      | Transformer.....                             | 14        |
| 5.5      | PriceMaster.....                             | 15        |
| 5.6      | KeyMaster.....                               | 15        |
| 5.7      | Enterprise Management Console.....           | 16        |
| <b>6</b> | <b>Platform resilience.....</b>              | <b>17</b> |
| 6.1      | DataSource Failover.....                     | 17        |
| 6.2      | Client failover to alternate Liberator.....  | 18        |

---

|    |                                       |    |
|----|---------------------------------------|----|
| 7  | Security.....                         | 19 |
| 8  | System Monitoring and Management..... | 20 |
| 9  | Customization.....                    | 21 |
| 10 | Glossary of terms and acronyms.....   | 22 |

# 1 Preface

## 1.1 What this document contains

**Note:** This document is now replaced by the **Caplin Xaqua Overview**, first issued in October 2009.

This document gives a business and technical overview of the Caplin Platform.

It contains:

- ◆ An overview of the uses of the Caplin Platform.
- ◆ Business benefits of using the Platform.
- ◆ An explanation of the Platform architecture.
- ◆ A description of each of the main Caplin Platform components.
- ◆ Descriptions of non-functional aspects of the platform, such as resilience, security, system monitoring and management, and customization.

## 1.2 Who should read this document

This document is intended for:

- ◆ Technical Managers
- ◆ Enterprise Architects and System Architects
- ◆ System Administrators
- ◆ Operators
- ◆ Software Developers

## 1.3 Related documents

- ◆ **Caplin Trader Overview**
- ◆ **Caplin DataSource Overview**
- ◆ **Caplin StreamLink Overview**

## 1.4 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 thoughts to [documentation@caplin.com](mailto:documentation@caplin.com).

## 1.5 Acknowledgments

*Windows*, *Internet Explorer*, and Microsoft *Excel* are registered trademarks of Microsoft Corporation in the United States and other countries.

*Sun*, *Solaris*, *Java*, *JavaScript*, *JDBC*, and *JMX* 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.

*Firefox* is a registered trademark of the Mozilla Foundation.

*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 the Caplin Platform?

**Note:** This document is now replaced by the **Caplin Xaqua Overview**, first issued in October 2009.

The Caplin Platform is a suite of software products for on-line financial trading and Web delivery of real-time market data.

Designed from the ground up for financial services applications, the Caplin Platform provides all the performance, security, and reliability that users demand for high-end trading. It is designed to offer the same quality of service over the Internet as traditional trading applications achieve over private circuits.

The Caplin Platform provides low-latency bi-directional messaging over internal networks, external links, and the Internet. It supports Ajax, Java™, and Windows® clients, as well as Caplin Trader Client, a highly configurable trading front end based on Ajax technology. A wide range of adapters and SDKs enable integration with any existing trading platforms, market data platforms, and common messaging platforms. The Platform provides extensive support for data management functions such as price tiering, order routing, and spread calculation.

The Caplin Platform integrates with, and supplements, your existing e-commerce trading infrastructure, allowing you to create and assimilate new tradable products and instruments. It gives you the ability to publish these to the market via traditional methods, and in addition provides a direct Internet based dealing user interface for your online portal. This is achieved by using Caplin's off-the-shelf data adapters and toolkits, to build or connect to pricing and trading systems, as well as connecting to required market data and exchange feeds.

### 2.1 What can I use the Caplin Platform for?

The Caplin Platform enables you to put advanced trading capabilities and real-time streaming information in front of any user anywhere. You can use it to add Web 2.0 functionality to your current eCommerce offerings, and to solve virtually all your client connectivity problems.

Typical deployments of the Caplin Platform focus on fixed income, forex, equities, and many of their derivatives, as well as cross-asset trading.

The Platform can also be used solely for market data distribution; it can connect to sources of market data and distribute the data to a portal. Using Platform 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

The following table shows the benefits that the Caplin Platform can bring to your business.

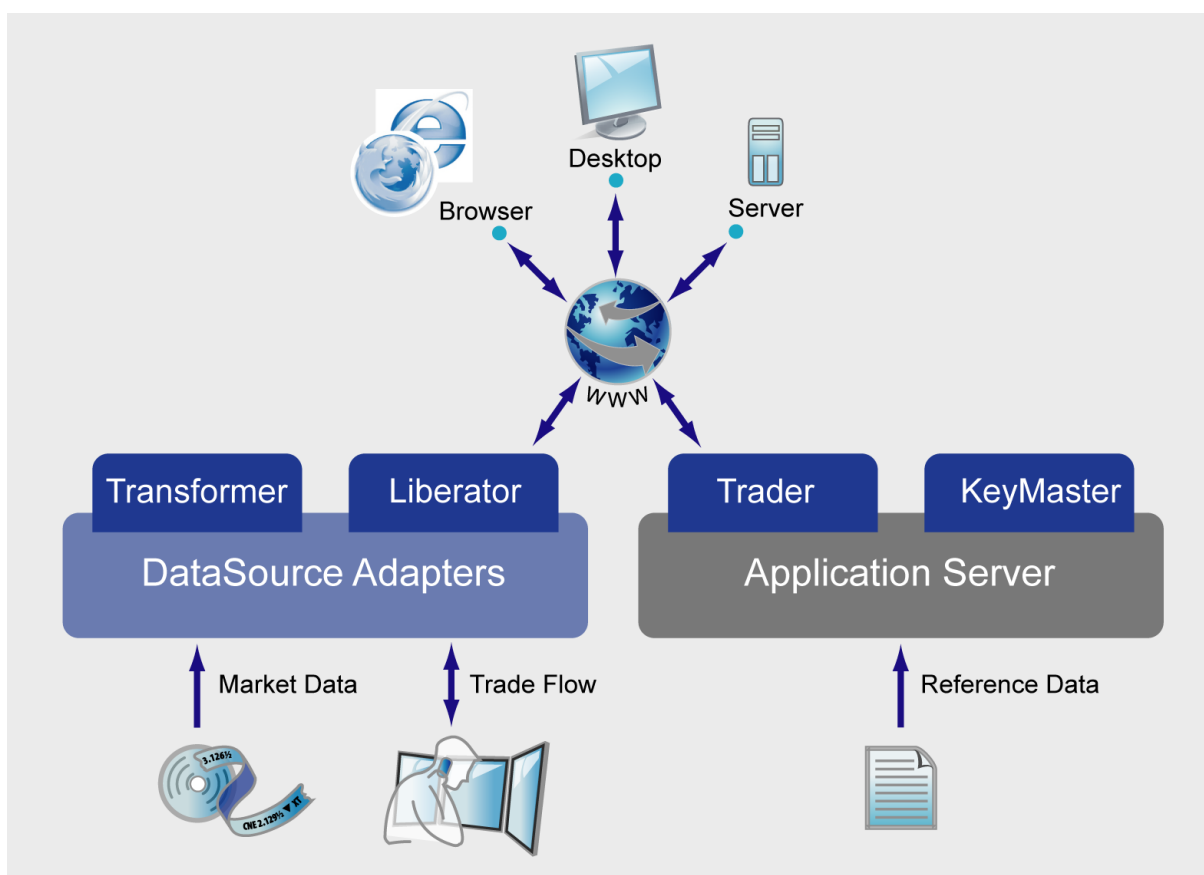
| Benefit   | Because  |
|---|--|
| Rapidly develop online financial applications.  | We provide all the tools and components you need.  |
| Easily support cross-product trading.   | Our flexible trading and data models support a wide range of asset classes.                                      |
| Roll out applications quickly and cheaply to users everywhere.                          | We support universal browser deployment with no plugins or security issues.                                      |
| Provide a highly flexible and extensible user interface.                                | Caplin Trader Client delivers this out of the box.   |
| Support unlimited numbers of users without adding latency or degrading service quality. | Caplin Liberator supports 10,000 users per server, and sophisticated clustering technology scales without limit. |
| Seamlessly integrate with existing trading, market data and reference data systems.     | Caplin DataSource products and development kits solve this problem.  |
| Easily implement advanced user pricing models.  | Caplin Transformer allows you to manage even the most complex price spreading and user tiering.                  |
| Provides a highly secure connection to your clients.                                    | The entire Caplin platform complies with rigorous security policies.   |

### 3 Platform architecture

The Caplin Platform is very flexible and can be used to implement a number of different financial business solutions. The following diagrams give a high level view of how the Platform is architected when it is deployed in two typical situations: online trading and distribution of market data. Each diagram shows the major platform components used and their relationships.

The main Caplin Platform components, including ones that are not shown in these simplified architecture diagrams, are described in the [Platform components](#) section.

#### 3.1 Online trading architecture



Caplin Platform Architecture for Online Trading

#### Communications

Caplin Platform components communicate with each other using two different protocols. The **DataSource** protocol is a high volume peer-to-peer protocol for communicating between server components. **RTTP** is a massively scalable Web protocol for communicating with very large numbers of clients. Caplin Liberator acts as the bridge between these two protocols.



## Trader

**Trader** is **Caplin Trader Client**, an out-of-the-box Web trading application that enables you to build multi-product financial trading portals. It can be hosted on any J2EE compliant application server. You can also integrate your own trading client application with the Caplin Platform using Caplin's **StreamLink** SDKs.

For more information on Caplin Trader Client see the document **Caplin Trader Overview**.

## Transformer

Caplin Transformer is a real-time business rules engine. It uses the DataSource protocol to subscribe to real-time data from other components of the Caplin Platform, such as DataSource adapters. It then uses Transformer Modules to analyze the data and apply business rules to it, before publishing derived or added-value data in real time back to components such as Liberator.

Transformer is useful for generating price spreads, cross rates, VWAP, indices, and so on. Standard modules provide functions such as caching, data normalization, calculation of derived data, time series storage and replay, event driven or timed publishing, and interfacing with databases and generic protocols such as ODBC, XML, and message queues.

Transformer's SDKs for Java, C and Lua make it easy to construct custom modules to fit specific needs. Modules can be pipelined to accomplish complex tasks.

## Liberator

Caplin Liberator is a bidirectional streaming push server 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.

Liberator supports a wide variety of data types and formats which it organizes into static Directories and dynamic Containers, with auto-subscription and multi-level caching. It communicates with other components of the Caplin Platform using the DataSource protocol.

Liberator securely authenticates users, checks the permitted level of data access, and maps requests onto internal items on a per-user basis. Dynamic permissioning and single sign on integration are fully supported.

There is extensive support for clustering, load balancing, and failover. Performance and bandwidth are optimized using multithreading, throttling/conflation, bursting, paging, and global caching.

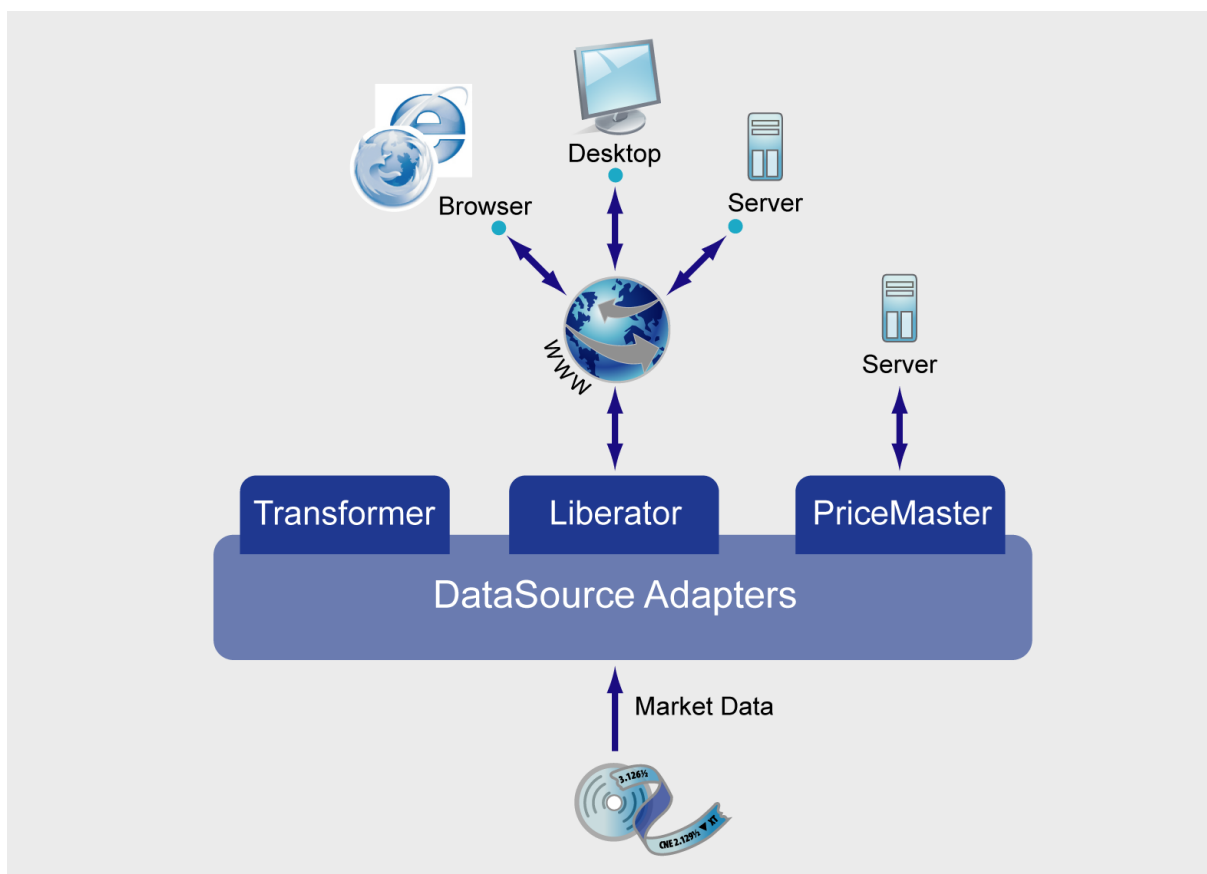
## DataSource adapters

DataSource adapters obtain market data from external market data distribution systems and vendor feeds, and send this data via the DataSource network to the Transformer and Liberator components. They also pass trade flow messages to and from external trading systems.

## KeyMaster

Caplin KeyMaster integrates the Caplin Platform with your existing authentication software, so that end users have access to all the functionality of the trading platform through a single sign on. KeyMaster can be hosted on any J2EE compliant application server.

### 3.2 Market data distribution architecture



#### Caplin Platform Architecture for Market Data Distribution

The architecture for market data distribution uses Transformer, Liberator, and DataSource adapter components in the same way as the online trading architecture, except that it does not need to handle trade message flows.

#### PriceMaster

The additional Caplin PriceMaster component is a data contribution engine for managing, converting, permissioning, and distributing real-time market data. Data received via DataSource can be published to quote vendors, trading hubs, data aggregators, and other firms, as well as to Liberator for distribution via the Internet. PriceMaster supports symbol and field mapping, page/record conversion, and alerting. It is controlled and configured through a powerful graphic interface.

## 4 Software environment

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

- ◆ The server components of the Caplin Platform (Transformer, Liberator, PriceMaster, DataSource adapters) run under the Linux® and Sun® Solaris™ operating systems.
- ◆ Clients using StreamLink for Browsers run in Firefox® and Microsoft Internet Explorer® web browsers.
- ◆ Clients using StreamLink for Java run in Java-enabled software environments.

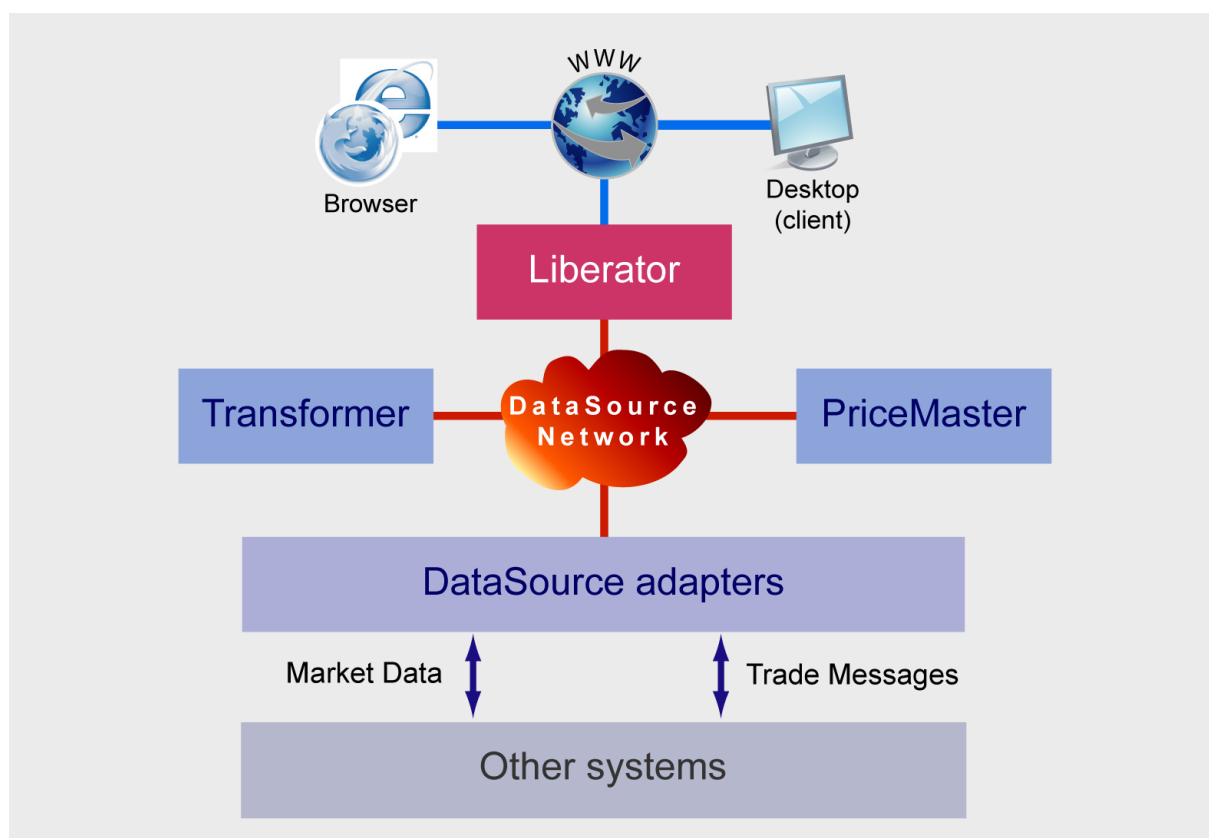
**Note:** For detailed information on software environment requirements (operating system versions, and so on), please refer to the Caplin technical documentation for the individual Platform components.

## 5 Platform components

The following sections describe the main Caplin Platform components in more detail, explaining what they do and how they relate to other platform components and external systems.

### 5.1 DataSource

DataSource is the communications infrastructure that links Caplin server components to each other and to non-Caplin systems. This is shown in the following diagram:



**DataSource within the Caplin Platform Architecture**

The term 'DataSource' is applied to several related items:

- ◆ The **DataSource network**, a messaging network used within the Caplin Platform.
- ◆ The **DataSource protocol**, 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.
- ◆ **DataSource applications (DataSource peers)**, applications that can communicate using the DataSource protocol, including Caplin components such as Liberator, Transformer, and PriceMaster.
- ◆ **DataSource adapters** that act as the interface between internal systems and the Caplin Platform.

- ◆ **DataSource SDKs**, used to write custom DataSource adapters.

DataSource supports both broadcast and subscription models of message propagation.

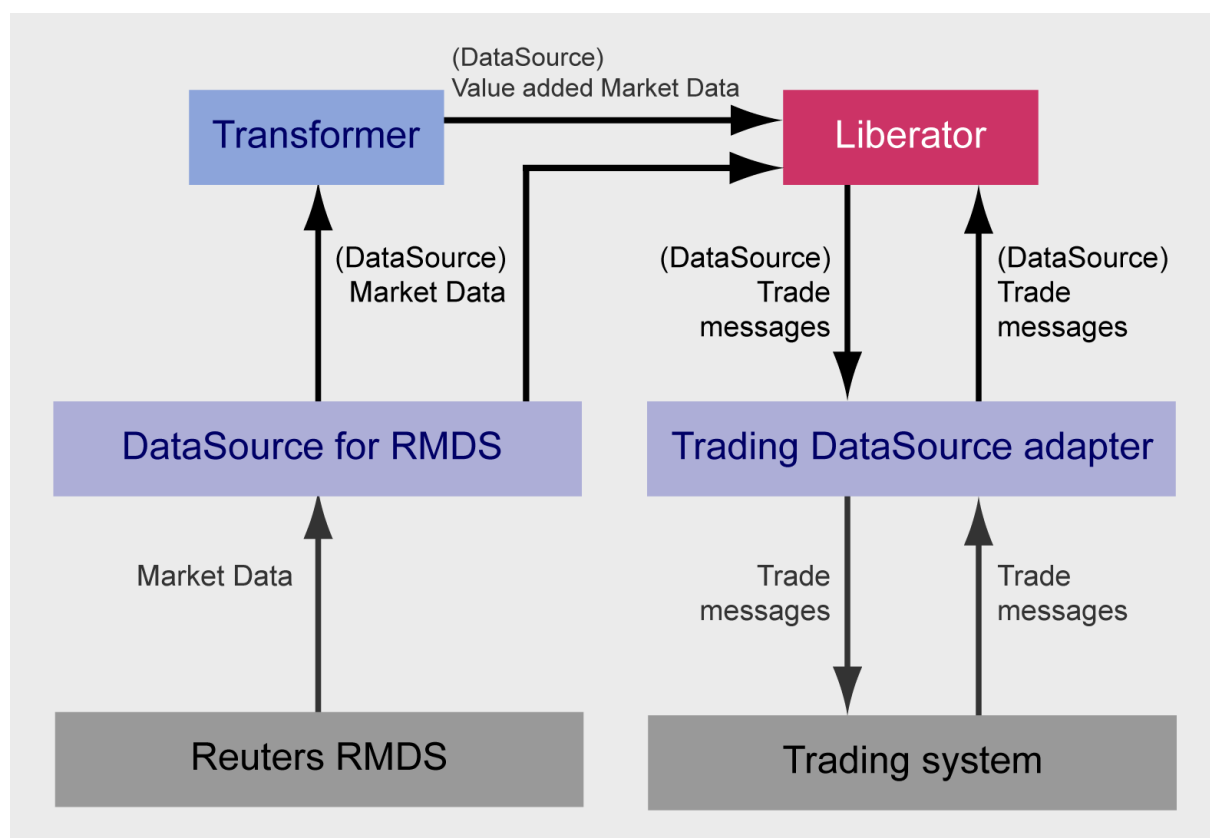
For more technical information about DataSource see the **Caplin DataSource Overview**.

## DataSource adapters

Caplin server components such as Liberator, Transformer, and PriceMaster normally need to communicate with external sources and sinks of data. They do this via DataSource adapters.

A DataSource adapter transforms incoming data into DataSource messages that can be understood by other Caplin 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 Reuters RMDS feed and passes it to Transformer. A custom Trading DataSource adapter passes trade messages between Liberator and an external trading system.

**Note:** In Caplin product documentation DataSource adapters are often just called “DataSources”.



**DataSource adapters in an example trading system**

## Off-the-shelf 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 Reuters RMDS
- ◆ DataSource for Reuters Triarch
- ◆ DataSource for TIB/RV
- ◆ DataSource for Databases

The **DataSource for Databases** adapter provides a mechanism for moving data between a JDBC™ compliant database and the Caplin Platform. It includes a comprehensive XML-based configuration facility, allowing you to implement fully functioned DataSource applications without needing to write code.

## DataSource SDKs

You may have sources of market data originating from your own internal systems or from data feeds for which there are no off-the-shelf DataSource adapters. You may also want to feed data, that has been modified or enhanced by Caplin Transformer, back into your internal systems, or exchange trade messages between Platform components and your order management system.

You can meet all these requirements by implementing custom DataSource adapters. Caplin provide both Java and C/C++ SDKs for rapid development of custom adapters.

## 5.2 RTTP and StreamLink

Caplin Liberator exchanges real-time financial data and trade messages with clients using Caplin's Real Time Text Protocol, **RTTP**. This is a text-based protocol that can be encapsulated in HTTP or HTTPS or operate directly over TCP/IP. It automatically tunnels to clients through firewalls and proxy servers, with no special port requirements. A client's connection to Liberator is a Persistent Virtual Connection that is self-healing and survives transient disconnects across unreliable internet connections, with no action required by the client.

The RTTP protocol is designed to ensure that communication between a Liberator and its clients utilizes bandwidth extremely efficiently.

The **StreamLink** libraries connect client applications to Liberator via the RTTP protocol. They provide an object oriented API, on top of the RTTP protocol, that provides access to RTTP functionality.

Using the appropriate StreamLink SDK, you can build applications in Ajax, JavaScript™, HTML, and Java. The following table shows which StreamLink SDK is used for which client technology.

| Client application technology | StreamLink SDK          |
|-------------------------------|-------------------------|
| Ajax/JavaScript/HTML          | StreamLink for Browsers |
| Java                          | StreamLink for Java     |

## StreamLink features

- ◆ The StreamLink SDKs provide subscription, data contribution, channel control, and metrical functions. They have a rich callback API for sending and receiving trade, data, and status messages.
- ◆ All StreamLink SDKs use the RTTP protocol and can access data from Liberator.
- ◆ StreamLink SDKs enable access to the full range of functionality of the RTTP transport mechanism, including tunneling, data health checking, advanced flow control and persistent virtual connection.
- ◆ A client typically has a single connection to Liberator, and a single cache, regardless of the number of windows, frames, or display modules.
- ◆ StreamLink automates the functions that most applications require. It provides default processing of responses from Liberator that are easy for a developer to extend. You only need to write code to perform custom processing in specific instances – everything else can be handled by StreamLink.

## Ajax support through StreamLink for Browsers

StreamLink for Browsers allows you to implement real-time streaming data display and message handling from within a Web browser. StreamLink for Browsers is browser-neutral and provides a powerful object-oriented JavaScript interface with no downloads, plugins, ActiveX controls or Java code. It supports multiple subscribers, multiple frames, and multiple windows per browser.

It can be used as a simple HTML extension called RTML, or to build complex Ajax-based rich Internet applications using its object-oriented API.

## 5.3 Liberator

Caplin Liberator is a bidirectional streaming push server designed to deliver market data and trade messages over any network that supports Web traffic.

Client applications communicate with Liberator via the StreamLink family of products, enabling them to send and receive data updates and trade messages in real time with low latency. Liberator in turn uses the DataSource protocol to exchange data and trade messages with other components of the Caplin Platform, such as [Transformer](#)<sup>[14]</sup>, [PriceMaster](#)<sup>[15]</sup>, and the [DataSource adapters](#)<sup>[10]</sup> that act as gateways to other systems.

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

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 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.

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.

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 (see [Type 2 records](#) <sup>[23]</sup>), time-series replay (see [Type 3 records](#) <sup>[23]</sup>), and intraday news histories. It can be configured to purge the cache at pre-set times, on a per-object basis.

## Data mapping

Liberator obtains the objects requested by clients by subscribing in turn to appropriate objects from sources such as 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 applications such as price tiering, and is implemented via an Auth module (see the section on [User management](#) <sup>[13]</sup>).

## User management

Liberator handles user authentication and permissioning through components known as Auth modules. The basic functions of an Auth module are, to authenticate users who attempt to log on to Liberator, to check the user's permitted level of access to a requested object (read-only, read-write, or no access), and to map client requests on to internal object names as required.

Liberator is supplied with three standard Auth modules (cfgauth, openauth, and XMLauth), offering different degrees of flexibility. You can also create your own custom Auth modules using the Caplin Liberator Auth module SDK or the Java Auth SDK, and use “[permission](#)” objects <sup>[19]</sup> to dynamically change configuration and permission settings in response to commands from an external user management system.

The cfgauth and XMLauth modules provide support for authentication in single sign on environments, using the [KeyMaster](#) <sup>[15]</sup> single sign on manager, and you can implement custom extensions to single sign on authentication by writing a custom Auth module.

## Performance and scalability

A single Liberator instance running on a 2 x dual-core 2.4GHz AMD Opteron server can support 10,000 concurrent users, each receiving 100 messages per second (where each message is a typical record update containing five fields of market data), with peak end-to-end latency of no more than 250 milliseconds.

Liberator has a number of design features that enable it to achieve this level of performance:

- ◆ Highly optimized low-level coding for message handling and I/O.
- ◆ Multi-threading for fast response and scalability on hardware platforms with multiple processors and multiple processor cores.
- ◆ Support for load sharing across server instances.
- ◆ A “service list” mechanism to allow efficient load balancing across servers.
- ◆ Cluster management to share information about license counts, users connections, data, and subscriptions.
- ◆ Global DataSource synchronization to avoid objects being requested via multiple paths in load-balanced configurations.
- ◆ Configurable “throttling” (data conflation) restricts the rate at which object updates are sent to clients, reducing network usage levels, client loading and Liberator loading.
- ◆ Configurable “bursting” (batching of messages) optimizes network bandwidth utilization and can improve message latency.
- ◆ “Container paging” allows display applications that subscribe to long lists of records to specify a “window” of content that is currently visible on the screen, so the application avoids receiving updates that are not being displayed.



## Resilience

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.
- ◆ 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.

Also see the section on [Platform resilience](#) <sup>[17]</sup>.

## 5.4 Transformer

Caplin Transformer is an event-driven real time business rules engine.

Transformer uses the DataSource protocol to subscribe to real-time data from other components of the Caplin Platform such as [DataSource adapters](#) <sup>[10]</sup>. It analyzes the data, applies business rules to it, and publishes derived or value-added data in real time, back to components such as [Liberator](#) <sup>[12]</sup>.

### Standard functions

Transformer's built-in capabilities include:

- ◆ Data subscription, publishing, and caching.
- ◆ Normalization of data from multiple sources into a uniform format, for processing elsewhere in the Caplin Platform and by client applications.
- ◆ Creation of derived data such as price spreads, cross rates, and volume weighted average price.
- ◆ Creation and storage of time series data, with replay on demand.
- ◆ Event-driven or timed publishing of data.
- ◆ Storage and retrieval in external databases.
- ◆ Sending data via generic outputs, such as ODBC, XML, or message queues.
- ◆ Generating email alerts.

### Transformer modules

Transformer's data analysis and business processing is implemented through Transformer Modules, which are loaded automatically on start-up. Transformer Development Kits are available for Java, C, and the Lua scripting language. These enable you to construct modules to fit your specific needs.

Transformer allows you to pipeline modules to accomplish complex tasks, with management of inter-module communication.

## 5.5 PriceMaster

Caplin PriceMaster is a data contribution engine for managing, converting, permissioning, and distributing real-time market data. Data received via [DataSource](#) <sup>[9]</sup> can be published to quote vendors, trading hubs, data aggregators, and other firms, as well as to [Liberator](#) <sup>[12]</sup> for onward distribution over the Internet. PriceMaster supports symbol mapping, field mapping, page shredding, and page construction.

PriceMaster is built on Caplin [Transformer](#) <sup>[14]</sup> and benefits from Transformer's reliability, resilience, flexibility, and modularity. It can easily be extended to provide custom functionality.

Control and configuration are provided through a powerful graphical interface, which is implemented as a set of custom views for the [Enterprise Management Console](#) <sup>[16]</sup>.

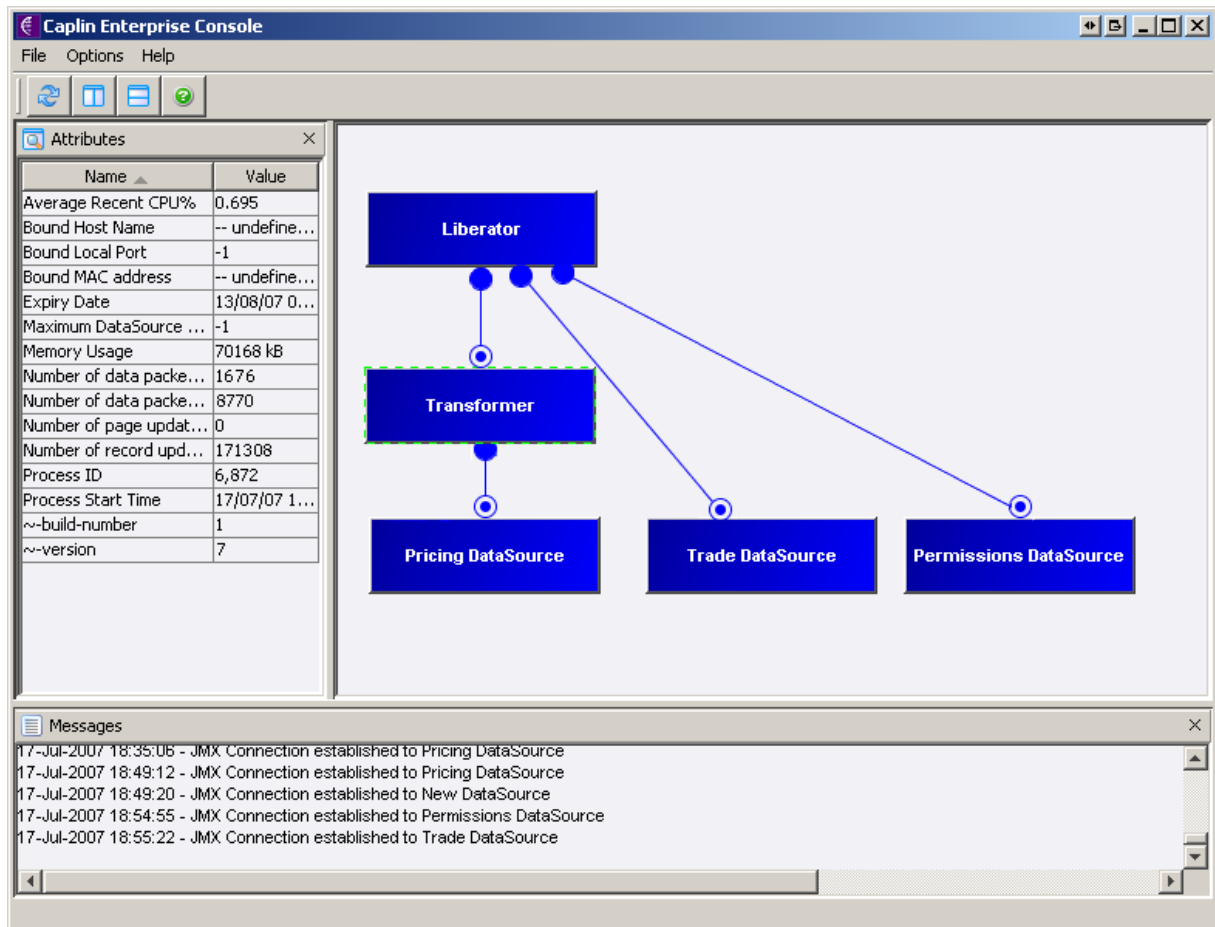
## 5.6 KeyMaster

Caplin KeyMaster is software that integrates the Caplin Platform with any existing authentication system, so that end users or web applications do not need to explicitly log in to 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 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.

## 5.7 Enterprise Management Console

All the key components of the Caplin Platform, including Liberator, Transformer, PriceMaster and DataSources, can report their status via the JMX management interface. The Caplin Enterprise Management Console (EMC) is a Java application that communicates with Platform components via JMX, and allows you to monitor and control these components through a GUI interface, as shown in the following example screen.



**Enterprise Management Console - Network Overview screen**

The EMC 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 EMC 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.

Also see the section on [System Monitoring and Management](#).

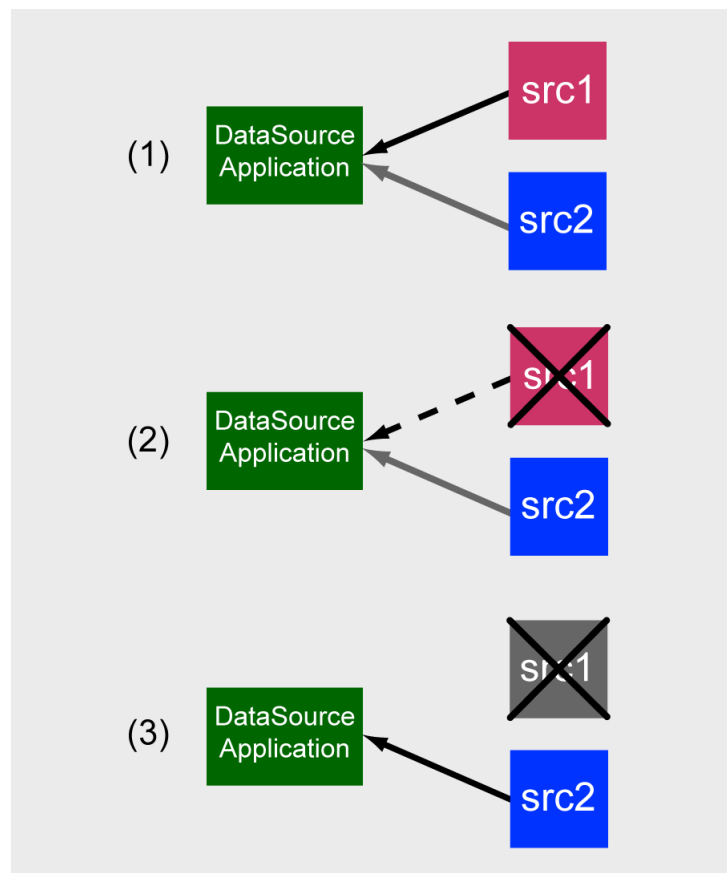
## 6 Platform resilience

The Caplin Platform is engineered to be resilient in the face of hardware, software and communication errors.

### 6.1 DataSource Failover

A DataSource application may lose the connection to one of its DataSource peers; this could be because of a network fault, or because the peer has failed. The DataSource peer can determine that the peer connection is lost by detecting loss of the socket, DataSource message timeout, or by detecting loss of heartbeat messages across the connection.

To ensure continuity of service, the DataSource can be configured so that it is connected to two or more peers, as shown in the following diagram.



**DataSource failover example**

When it loses its connection to the first peer (src1 in the diagram) the DataSource application automatically fails over to use the second peer (src2). 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**.

## 6.2 Client failover to alternate Liberator

A client application may lose the connection to its Liberator server; this could be because of a network fault, or because the Liberator has failed.


The client can determine that the Liberator connection is down by detecting loss of the socket, through message response timeout, or by detecting that heartbeat messages between the client and Liberator have been lost. In the case where the connection loss is transient, RTTP's Persistent Virtual Connection feature provides rapid and seamless recovery, with no action required by the client.

When recovery of the Persistent Virtual Connection fails, the client will try to reconnect the Liberator session by issuing a new HTTP request and logging into Liberator again. Liberator can then send the client any pending update messages.

If this reconnection attempt fails, the client can connect to an alternate Liberator instead. A failover algorithm, configured in the client, determines which Liberator the client should fail over to. The available algorithms are flexible and allow various configurations, including backup and redundant servers; they involve trying one or more alternative servers in a specified order.

## 7 Security

### Authentication and permissioning

User authentication and permissioning is comprehensively handled by the Auth module within Caplin Liberator. See [User Management](#) .

### Permissions objects

Custom Auth modules can be sent real-time configuration and permission information from DataSource peers via permissions objects. Permissions objects provide a means for a DataSource, or an external system connected to a DataSource, to communicate with an Auth module and dynamically change configuration and permission settings.

A StreamLink enabled client can also subscribe to particular permissions objects, and receive updates to them from Liberator through the standard update mechanism. The client can then use the permission information to control the way the application behaves. For example, a trading application could disable the quick trade button for instruments that the user is not permitted to trade.

### Single sign on

You can integrate the Caplin Platform with your existing single sign on system using Caplin [KeyMaster](#) .

## 8 System Monitoring and Management


The Caplin Platform contains full enterprise-wide monitoring and management capability.

### Monitoring facilities

Applications built with Caplin's DataSource SDK can be enabled for monitoring and management, including Liberator, 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. Liberator, PriceMaster, and Transformer also provide additional component specific information.

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

The Platform'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; 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 the Caplin Platform.

You can also monitor and manage the Caplin Platform using the [Caplin Enterprise Management Console](#) .

### Logging

Caplin Platform components such as Liberator, 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.

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.

## 9 Customization

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

Most Platform components are highly configurable. Some configuration is to enable components to integrate and communicate with each other, for example configuration of DataSource peers. Other configuration changes the behaviour of the component; for example, you can create your own permissioning structures and control entitlement to objects held on Liberator, by defining XML-based configuration in the XML Auth module.

Many Platform components have associated SDKs, with APIs that enable you to modify and extend the components' behaviour. For example DataSource SDKs allow you implement custom DataSource adapters that will send data to and from your own back-end systems.



## 10 Glossary of terms and acronyms

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

| Term                       | Definition  |
|----------------------------|---|
| <b>Ajax</b>                | <u>A</u> syncronous <u>J</u> avaScript and <u>X</u> ML<br>A combination of Web technologies used to implement interactive Web clients.  |
| <b>API</b>                 | <u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface   |
| <b>Auth module</b>         | The software module that handles <b>authentication</b> and <b>permissioning</b> within Caplin Liberator.  |
| <b>Authentication</b>      | 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.   |
| <b>Authorization</b>       | The process of determining the access rights that a user has to resources, such as data and functionality provided the software.<br>Also known as <b>permissioning</b> .  |
| <b>Bursting</b>            | A technique used by Caplin Liberator to improve performance by writing user output in defined “bursts”, or “batches”. See <a href="#">Liberator performance and scalability</a> <sup>13</sup> .   |
| <b>Conflation</b>          | See <b>Throttling</b> .   |
| <b>DataSource</b>          | Synonym for <b>DataSource adapter</b> .   |
| <b>DataSource adapter</b>  | A <b>DataSource peer</b> that obtains data from, and/or sends data to, an external (non-Caplin) system.   |
| <b>DataSource network</b>  | See the section on <a href="#">DataSource</a> <sup>9</sup> .  |
| <b>DataSource peer</b>     | An application that can communicate with another application using the <b>DataSource protocol</b> .   |
| <b>DataSource protocol</b> | The Caplin protocol that transmits financial data across the <b>DataSource network</b> . See the section on <a href="#">DataSource</a> <sup>9</sup> .   |
| <b>EMC</b>                 | <u>E</u> nterprise <u>M</u> anagement <u>C</u> onsole<br>A Caplin Platform component for monitoring and controlling other components through a GUI interface.   |
| <b>Failover</b>            | 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.               |
| <b>JMX</b>                 | <u>J</u> ava <u>M</u> anagement <u>E</u> xtensions<br>A Java technology for application and network management.   |
| <b>KeyMaster</b>           | Caplin KeyMaster integrates the Caplin Platform with any existing <b>authentication</b> system, so that end users or web applications do not need to explicitly log in to <b>Liberator</b> in addition to their normal login procedure. |
| <b>Liberator</b>           | Caplin Liberator is a bidirectional streaming push server designed to deliver market data and trade messages over any network that supports Web traffic.  |
| <b>Peer</b>                | See <b>DataSource peer</b> .  |
| <b>Permissioning</b>       | An alternative term for <b>authorization</b> .  |

| Term                         | Definition  |
|------------------------------|---|
| <b>PriceMaster</b>           | Caplin PriceMaster is a data contribution engine for managing, converting, <b>permissioning</b> , and distributing real-time market data.   |
| <b>Public key encryption</b> | A method of sending encrypted information between two parties without the need for them to exchange a key for encrypting and decrypting the information. Rather than using a single key it uses two related keys – a public key and a private key.  |
| <b>RTTP</b>                  | <u>Real Time Text Protocol</u><br>Caplin's object-oriented, real-time, protocol for the distribution of financial data and trade messages over internet-protocol networks between client applications and Caplin Liberator.<br>See the section on <a href="#">RTTP and StreamLink</a> <sup>11</sup> . |
| <b>SDK</b>                   | <u>Software Development Kit</u>   |
| <b>Single sign on</b>        | 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.  |
| <b>StreamLink</b>            | Caplin StreamLink is a family of <b>SDKs</b> that allows developers to add <b>RTTP</b> streaming capability to client applications.<br>See the section on <a href="#">RTTP and StreamLink</a> <sup>11</sup> .   |
| <b>Throttling</b>            | A technique used by Caplin Liberator to improve performance by restricting the rate at which object updates are sent to a client. Also called conflation.<br>See <a href="#">Liberator performance and scalability</a> <sup>13</sup> .  |
| <b>Transformer</b>           | Caplin Transformer is an event-driven real time business rules engine.  |
| <b>Type 2 record</b>         | A record type that is structured to hold "level 2" quote data. Level 2 quote data enables several price quotes per symbol (coming from different market makers or traders) to be available at the same time.  |
| <b>Type 3 record</b>         | A record type that is structured to store update history by keeping all updates rather than overwriting the symbol/field pair. Commonly used for holding and viewing daily trade activity.  |



*The information contained in this publication is subject to UK, US and international copyright laws and treaties and all rights are reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the written authorization of an Officer of Caplin Systems Limited.*

*Various Caplin technologies described in this document are the subject of patent applications. All trademarks, company names, logos and service marks/names ("Marks") displayed in this publication are the property of Caplin or other third parties and may be registered trademarks. You are not permitted to use any Mark without the prior written consent of Caplin or the owner of that Mark.*

*This publication is provided "as is" without warranty of any kind, either express or implied, including, but not limited to, warranties of merchantability, fitness for a particular purpose, or non-infringement.*

*This publication could include technical inaccuracies or typographical errors and is subject to change without notice. Changes are periodically added to the information herein; these changes will be incorporated in new editions of this publication. Caplin Systems Limited may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.*

## Contact Us

Caplin Systems Ltd.  
Triton Court  
14 Finsbury Square  
London EC2A 1BR  
UK

Telephone: +44 20 7826 9600  
Fax: +44 20 7826 9610

**[www.caplin.com](http://www.caplin.com)**