

CAPLIN

Caplin Xaqua 1.0

Best Practices For Deploying Caplin Xaqua

November 2010

CONFIDENTIAL

Contents

1	Preface.....	1
1.1	What this document contains.....	1
	About Caplin document formats	1
1.2	Who should read this document.....	1
1.3	Related documents.....	2
1.4	Typographical conventions.....	3
1.5	Feedback.....	3
1.6	Acknowledgments.....	4
2	Deployment architectures.....	5
2.1	Security model.....	6
2.2	Failover legs.....	7
2.3	Single leg deployment (no failover).....	7
2.4	Multi-leg deployment and resilience	10
2.5	Cross-site deployment.....	11
2.6	Deployment across a WAN.....	12
3	Failover scenarios	14
3.1	Liberator failover.....	14
3.2	Transformer failover.....	15
3.3	DataSource failover	16
4	Authenticating client sessions.....	17
5	Deploying the Xaqua Management Console (XMC).....	19
6	Deployment requirements.....	20
6.1	Operating systems.....	20
6.2	Supported hardware.....	20
6.3	Hardware recommendations.....	21
7	Appendix A: Default port allocations.....	24
7.1	Default Liberator ports.....	24
7.2	Default Transformer ports.....	25
7.3	Default DataSource adapter ports.....	26
8	Appendix B: Reverse proxies and Liberator.....	27
9	Glossary of terms and acronyms.....	30

1 Preface

1.1 What this document contains

This document gives recommendations on how to deploy Caplin Xaqua in a typical live environment.

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/>.

1.2 Who should read this document

This document is intended for Architects, System Administrators, and Developers who are planning for the deployment of Caplin Xaqua in a live environment.

1.3 Related documents

- ◆ **Caplin Xaqua Overview**

A business and technical overview of Caplin Xaqua.

- ◆ **Caplin DataSource Overview**

A technical overview of Caplin DataSource.

- ◆ **StreamLink 5.0 Overview**

A technical overview of Caplin StreamLink.

- ◆ **Caplin Trader Overview**

A business and technical overview of Caplin Trader.

This is of interest to readers deploying Caplin Xaqua when the Caplin Xaqua client applications have been developed using the Caplin Trader framework.

- ◆ **Caplin Liberator Administration Guide**

Describes the Caplin Liberator server and its place within Caplin Xaqua. Explains how to install, configure, and manage Liberator.

- ◆ **Caplin Transformer Administration Guide**

Gives an overview of Caplin's Transformer product and describes how to install, configure, and manage it.

- ◆ **KeyMaster Overview**

An overview of Caplin KeyMaster, including the product architecture and an explanation of how KeyMaster authenticates users in a single sign-on environment.

- ◆ **Caplin Xaqua: Monitoring and Management Overview**

Describes the Caplin Monitoring and Management solution and its place within Caplin Xaqua.

- ◆ **Caplin Xaqua: Getting Started With The XMC**

Explains how to configure the Caplin Xaqua Management Console (XMC).

- ◆ **Benchmarking Caplin Liberator**

Details the results of a set of performance benchmark tests carried out on Caplin Liberator 4.4.

The information provided in this report can assist customers in production capacity planning when deploying Liberator.

1.4 Typographical conventions

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

Type	Uses
aMethod	Function or method name
<i>aParameter</i>	Parameter or variable name
<i>/AFolder/Afile.txt</i>	File names, folders and directories
<div>Some code;</div>	Program output and code examples
The value=10 attribute is...	Code fragment in line with normal text
Some text in a dialog box	Dialog box output
Something typed in	User input – things you type at the computer keyboard
XYZ Product Overview	Document name
◆	Information bullet point
■	Action bullet point – an action you should perform

Note: Important Notes are enclosed within a box like this.
Please pay particular attention to these points to ensure proper configuration and operation of the solution.

Tip: Useful information is enclosed within a box like this.
Use these points to find out where to get more help on a topic.

Information about the applicability of a section is enclosed in a box like this.
For example: "This section only applies to version 1.3 of the product."

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.

Visit our feedback web page at <https://support.caplin.com/documentfeedback/>.

1.6 Acknowledgments

Adobe® Reader is a registered 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.

Sun, Solaris, Java, 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.

The following are trademarks, service marks or registered trademarks of Intel Corporation: *Intel®*, *Xeon®*.

AMD Opteron is a trademark of Advanced Micro Devices.

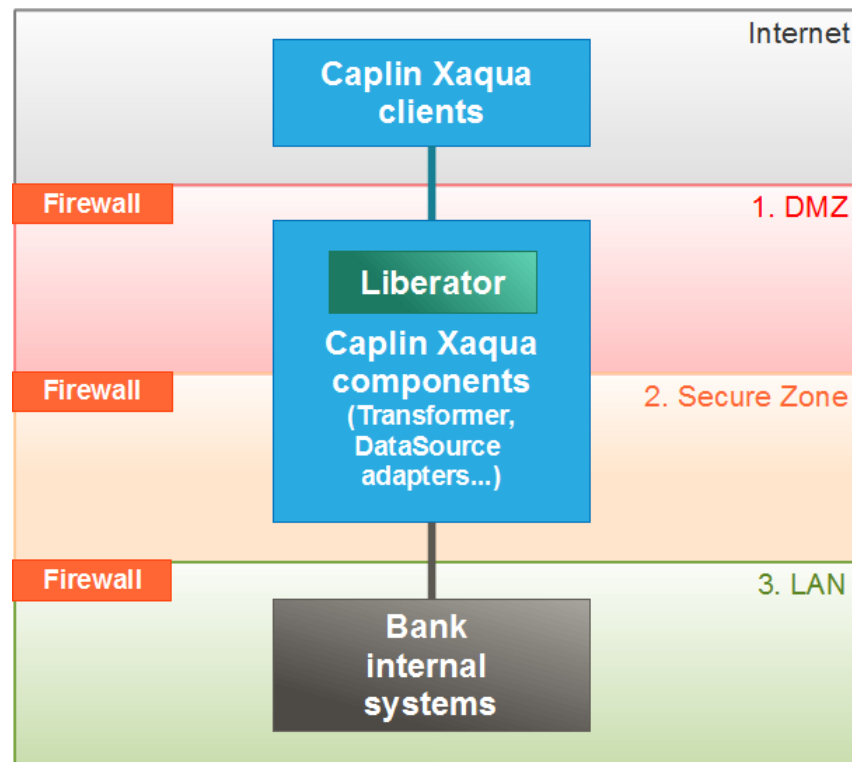
2 Deployment architectures

Caplin Xaqua can be deployed in several different architectures to support various security and resilience requirements. The following sections describe each of these architectures, explaining where the individual Caplin Xaqua components are situated within them.

- ◆ [Security Model](#) 
- ◆ [Failover legs](#) 
- ◆ [Single leg deployment \(no failover\)](#) 
- ◆ [Multi-leg deployment and resilience](#) 
- ◆ [Cross-site deployment](#) 

2.1 Security model

Because end-users can access the Bank's data feeds and trading systems from the Internet, the Caplin Xaqua components and Bank systems should be allocated to networks that are secured behind appropriate firewalls. The following diagram shows a standard 3 tier security model, based on a security zoning of firewalls and networks. The zones have increasing levels of security from the top to the bottom of the diagram.



Security Zones

- ◆ The **Internet** is the publicly accessible zone and is *insecure*. The Caplin Xaqua client applications, such as applications implemented using Caplin Trader, typically execute in this zone.
- ◆ **Zone 1** is the "Demilitarized Zone" (**DMZ**), housing the Bank's servers that interface with the Internet, and Caplin Xaqua's Liberator servers. This zone sits behind a firewall, but the servers are addressable from the Internet Zone.
- ◆ **Zone 2** is the **Secure Zone**. This sits behind a further firewall and the servers in it are on a separate sub-network to servers in the DMZ. This zone contains the rest of the Caplin Xaqua components. It also contains the application server(s) that serve the Bank's client portal and Caplin Xaqua clients.
- ◆ **Zone 3** is the most secure of the zones; it is on a separate network (**LAN**), only accessible from the Secure Zone through another firewall. This is where the Bank's core internal systems that interact with Caplin Xaqua are located.

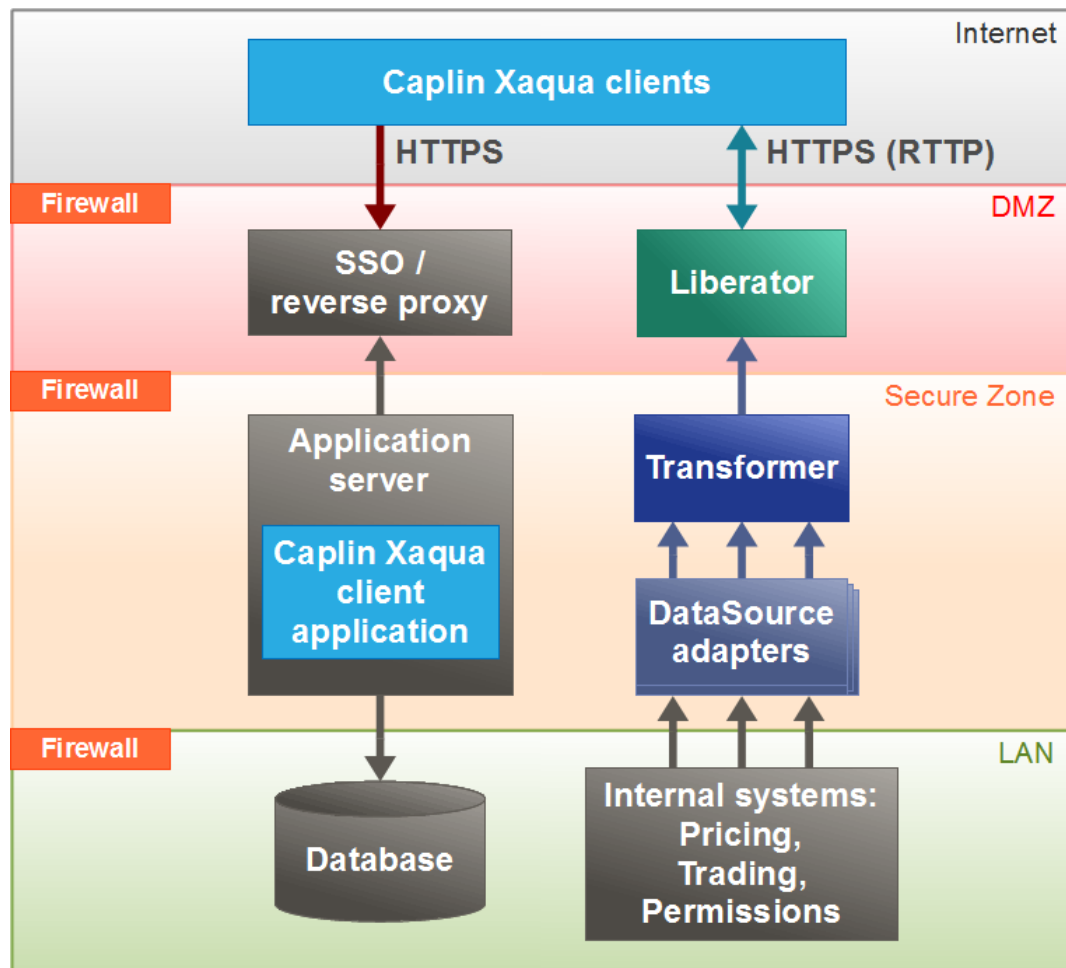
2.2 Failover legs

Caplin Xaqua is typically deployed with multiple component instances to provide resilience against hardware, software, and network failures. To help achieve this, the hardware and software components can be arranged in processing units called “failover legs”.

In normal operation, all the components in a single failover leg – typically Caplin Liberator, Caplin Transformer, DataSource adapters, and the Bank’s internal systems – work together to provide the system's functionality. If a component fails (or a connection to it, or the machine on which it runs), the operations provided by that component are taken up by an alternative copy of the component running in a different failover leg. This transfer of operation is called “failover”.

2.3 Single leg deployment (no failover)

The following diagram shows a single leg deployment of a Caplin Xaqua installation. This is the simplest configuration and does not provide failover if a component, machine, or network connection should fail.



Single leg deployment (no failover)

Assuming the **Caplin Xaqua client** executes in a browser, it is deployed on an application server behind the Bank's existing single sign-on (SSO) system (see [Authenticating client sessions](#)^[17]). Clients access the server through HTTPS, and after the user has signed on, the Caplin Xaqua client application is downloaded to the browser from the application server. The application may use a database for persistent storage of data, such as user-modified layouts.

The Caplin Xaqua server components (Liberator, Transformer, and DataSource adapters) interact with the Caplin Xaqua client in the browser; streaming price information to the application in real-time, exchanging trade messages, and supplying permission information.

Security zones

The diagram shows how components can be deployed to conform to the security model previously described (see [Security model](#)^[6]).

- ◆ Being internet facing, **Caplin Liberator** is deployed behind an internet firewall within a **DMZ**.
- ◆ **Caplin Transformer** and the **DataSource adapters** reside in the **Secure Zone**.
- ◆ The database that is accessed by the application server, and the Bank's internal systems for pricing, trading, granting user permissions, and so on, reside in a highly secure **LAN** environment behind an additional firewall.

The arrows show the direction in which connections between components are initiated. The directions are typically determined by the security model. In this example, as the Liberator is in the **DMZ**, it is not allowed to initiate the connection to the Transformer in the **Secure Zone**, as this would violate the security regime; the connection attempt would be denied by the firewall. Instead, the Transformer initiates the connection, which the Liberator listens for.

Similarly, the DataSource adapters in the **Secure Zone** are not allowed to initiate connections to the Bank's highly secure internal systems in the **LAN** zone; instead they listen for connections from the internal systems.

Tip: When two Caplin Xaqua components communicate with each other, to meet your particular security restrictions, you can configure which component initiates the connection.

Tip: Reverse proxies
To ensure low-latency and high volume streaming throughput, it is recommended that you do not install a reverse proxy between Liberator and the clients.
See [Appendix B: Reverse proxies and Liberator](#)^[27].

Connections

When the Caplin Xaqua client executes in a browser, Liberator listens on the HTTPS port for client traffic (using the RTTP protocol).

Connections between Liberator, Transformer and DataSources use the DataSource protocol, based on TCP/IP. For more information, see the **DataSource Overview**.

Connections between DataSources and the internal systems may use the protocols of standard market data platforms (such as TIB or RMDS), or they can use other proprietary protocols.

The connection between the application server and the database is typically through JDBC™.

Tip: For maximum security, the Caplin Xaqua clients should connect to the application server and the Liberator via HTTPS, *not* via HTTP.

Port Usage

Port usage is fully configurable within Caplin Xaqua, and within most application servers and databases. For more information, see [Appendix A – Default Port Allocations](#)^[24].

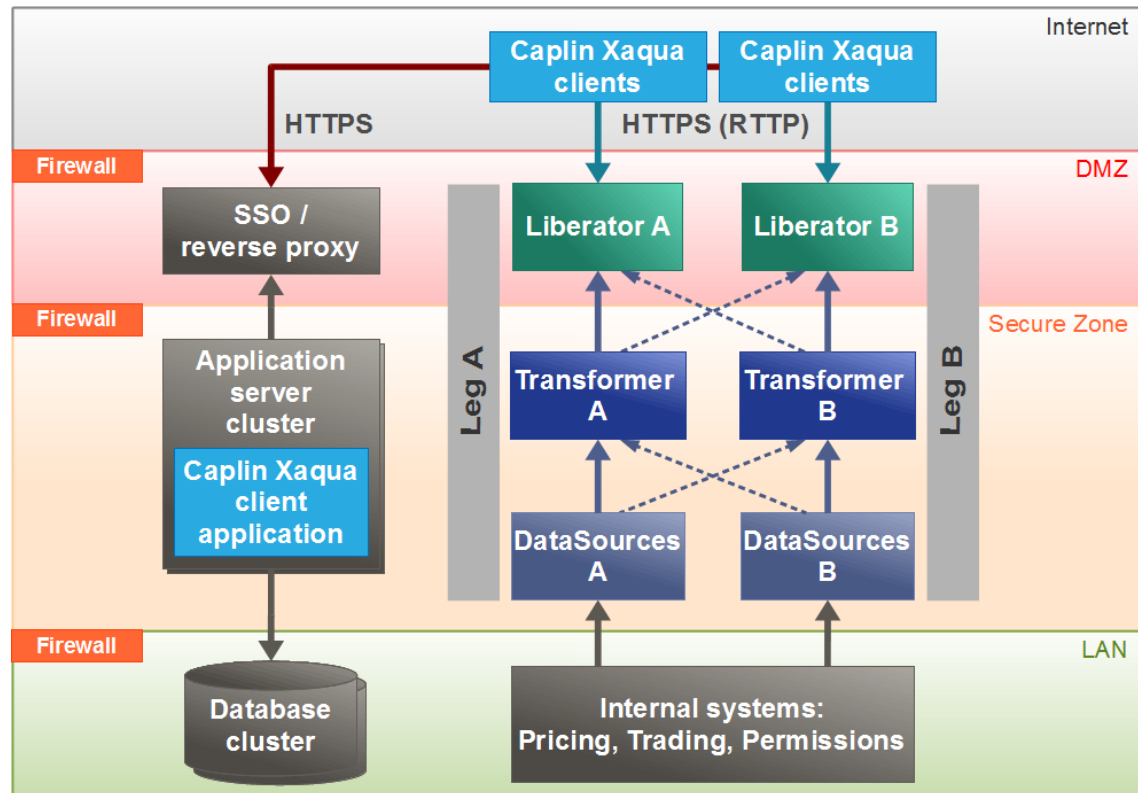
DNS configuration

When the Caplin Xaqua client is browser-based, the Liberator URL and the application server's URL must share a common domain.

For example, if Liberator is hosted at `liberator.mydomain.com`, the application server may be hosted at `appserver.mydomain.com` but not at `appserver.myotherdomain.com`.

2.4 Multi-leg deployment and resilience

To ensure resilient operation, Caplin Xaqua should be deployed using multiple component instances across multiple failover legs, as shown in the following diagram.



Multi-leg deployment for resilience

The diagram shows a single geographical site running Caplin Xaqua and associated software, within the security regime described in the [Security model](#)^[6] section.

At startup, Legs A and B are independent of each other, and are used to balance the transaction load. Each leg is configured the same way and is connected to the same internal systems for pricing, trading, and permissioning.

If a Caplin Xaqua component detects a significant reduction in quality of service during operation, it automatically instigates failover to an alternate instance of the component – the corresponding instance in the other leg. For example, if Liberator A fails, the clients connected to it fail over to Liberator B.

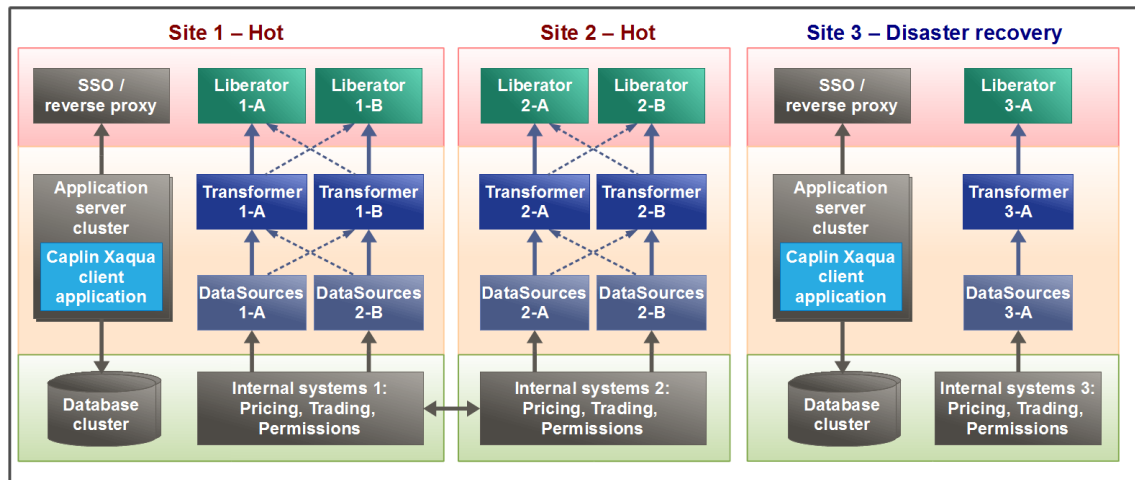
Failover connections are configured for each component instance. To meet the requirements of the security regime (Liberators cannot initiate connections to Transformers) but still allow failover, and to reduce the time taken to fail over, the connections between pairs of failover components are made in advance when the components start up, rather than during a failover event. The dotted arrows show connections that are made in advance but are not active (transmitting data) until a failover occurs. For example, in the diagram, when Transformer A starts up, it connects to the failover Liberator B in addition to Liberator A (the Liberator it normally talks to).

For more detail on how failover works, see [Failover scenarios](#)^[14].

2.5 Cross-site deployment

For maximum resilience, both Caplin Xaqua and the application server components should be deployed across multiple sites.

The following diagrams illustrate an example deployment across 3 geographically distinct sites. Sites 1 and 2 are “hot” sites – that is, they are live installations. Site 3 is for disaster recovery purposes; if the whole of Site 1 or Site 2 should become unavailable, the software at Site 3 is started up and the client connections that would have been handled by the now unavailable site are routed to Site 3 instead.



Deployment across two hot sites and a disaster recovery site

Tip: To reduce transmission latency between clients and the Liberator, Caplin recommends establishing a hot site within each center of geographic client density.

For example Site 1 could be located in London, and would handle connections and transactions from clients in Europe, and Site 2 could be located in New York, to handle connections and transactions originating in North America.

See [Deployment across a WAN](#) ^[12].

Note that each hot site has two failover legs to handle failure of individual components within the site (see [Multi-leg deployment and resilience](#) ^[10]).

Disaster recovery sites are set up in exactly the same way as hot sites, and may also use multiple failover legs as required. You may want to set up more than one disaster recovery site.

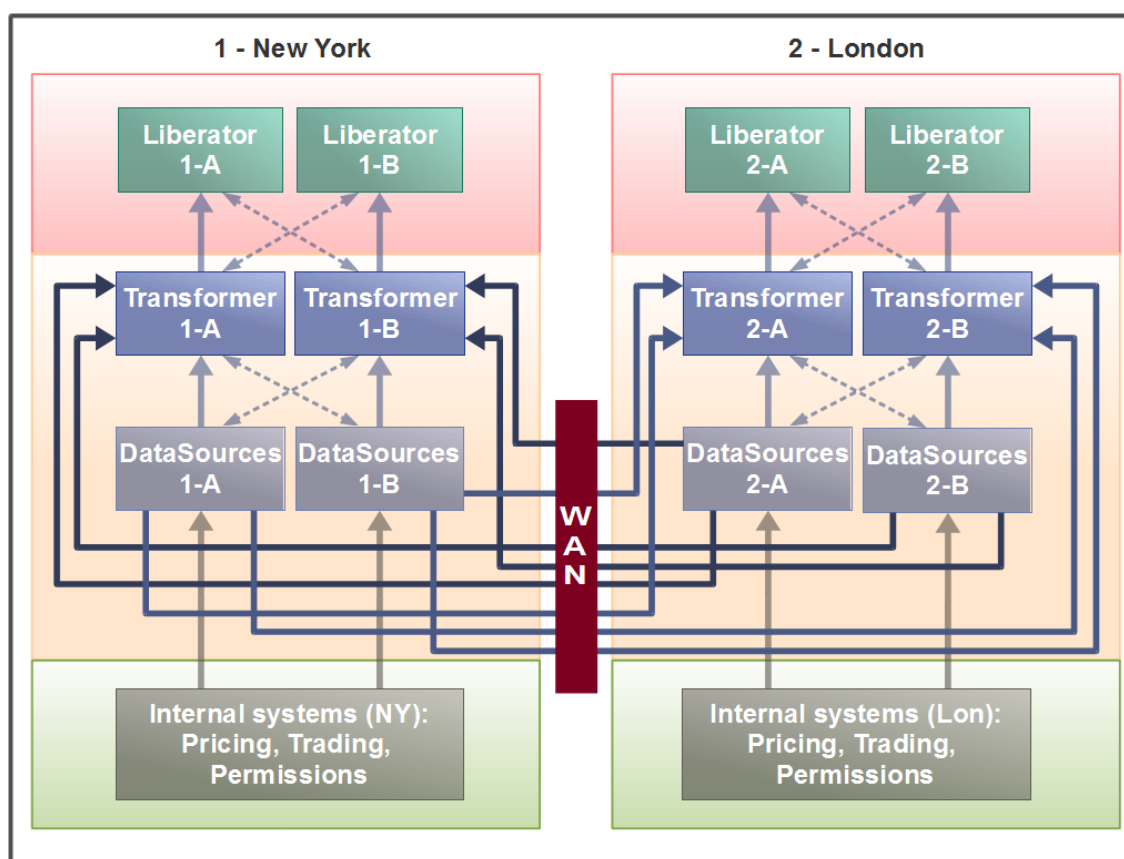
Although there are no technical differences between a disaster recovery site and a hot site, the software license fees for the disaster recovery site may be discounted provided the software is only operational when one of the hot sites is not available to end-users.

The failover from a hot site to a disaster recovery site is typically initiated manually.

2.6 Deployment across a WAN

You may need to deploy your Caplin Xaqua installation across several sites and share data across those sites. For example, you could have a site in New York serving North American FI customers, and a site in London serving European FI customers. The Bank's pricing system in New York supplies US Bond prices, and the pricing system in London supplies European bond prices. Customers in both regions wish to see and trade both US and European bonds, so pricing information must be exchanged between the two sites over a relatively slow wide area network (WAN).

The following diagram shows an obvious, but not recommended, way to do this. In this deployment, every Pricing DataSource is connected to every Transformer, so that every Transformer can receive prices for all instruments.



**Two sites connected by a WAN
(not recommended)**

However, this approach is not recommended for the following reasons:

- ◆ It wastes communication bandwidth.

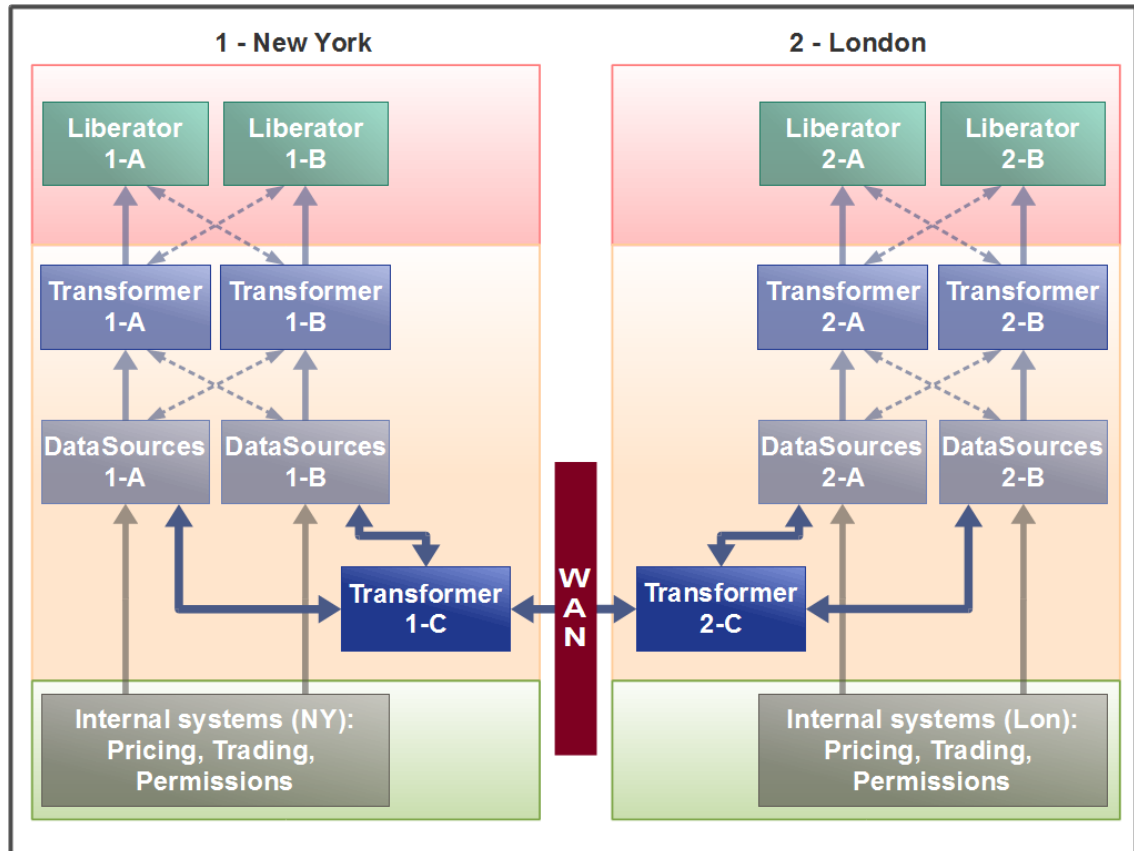
Assume the Transformers at a site are both active and so share the connection and traffic load. Each such Transformer subscribes to instruments in isolation from the other Transformer, even when some or most of the subscriptions are in common. When such subscriptions are for instruments whose prices are supplied from the site in the other regional location, the price updates are sent across the WAN *twice*, once for each subscribing Transformer. This wastes bandwidth on the slowest communication link in the network, and can result in end-users experiencing high latency on updates.

- ◆ The connection topology is complex.
- ◆ The sites are too interdependent.

Changes in one site, such as adding another pricing DataSource adapter, affect the other site.

These problems become more acute as more regional locations are added. For example, adding a third site (Tokyo) increases the number of Transformer to DataSource connections from 8 to 36.

The following diagram shows a recommended approach to connecting regional sites:



**Two sites connected by a WAN
(recommended)**

Each site has an additional Transformer: '1-C' in New York and '2-C' in London. The sole function of this Transformer is to route subscription requests and price data updates for non-local instruments to the appropriate remote site, where they are received by the corresponding Transformer at the remote site and passed on to the relevant DataSource. This arrangement has several advantages:

- ◆ Traffic on the WAN is much reduced, and scales more linearly as further regional sites are added.
- ◆ The connection topology is much simpler.
- ◆ Local changes can be made at one site without affecting the other regional sites.
- ◆ Consequently, *it is much easier to add another regional site.*

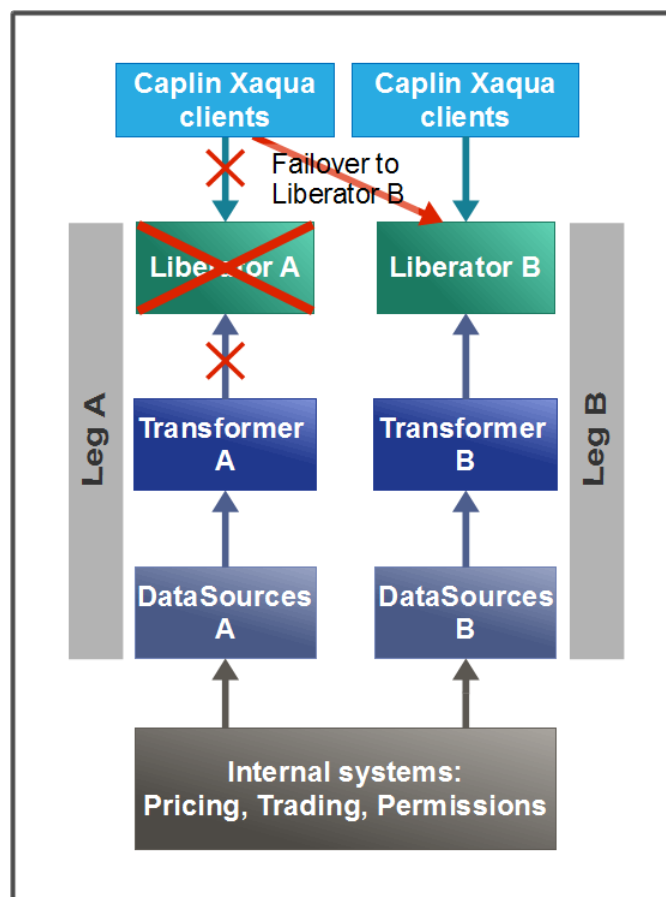
3 Failover scenarios

When a component fails, Caplin Xaqua will react to maintain quality of service on each failover leg. The following scenarios show how failover is handled for each component type.

- ◆ [Liberator failover](#) ¹⁴
- ◆ [Transformer failover](#) ¹⁵
- ◆ [DataSource failover](#) ¹⁶

3.1 Liberator failover

On loss of a Liberator instance, or loss of connection to the Liberator, Caplin Xaqua clients automatically connect to an alternate Liberator instance. The failover information is configured within the StreamLink library built into the client. When the client connects to the alternate Liberator, all previous instrument subscriptions and trade channels are restored.

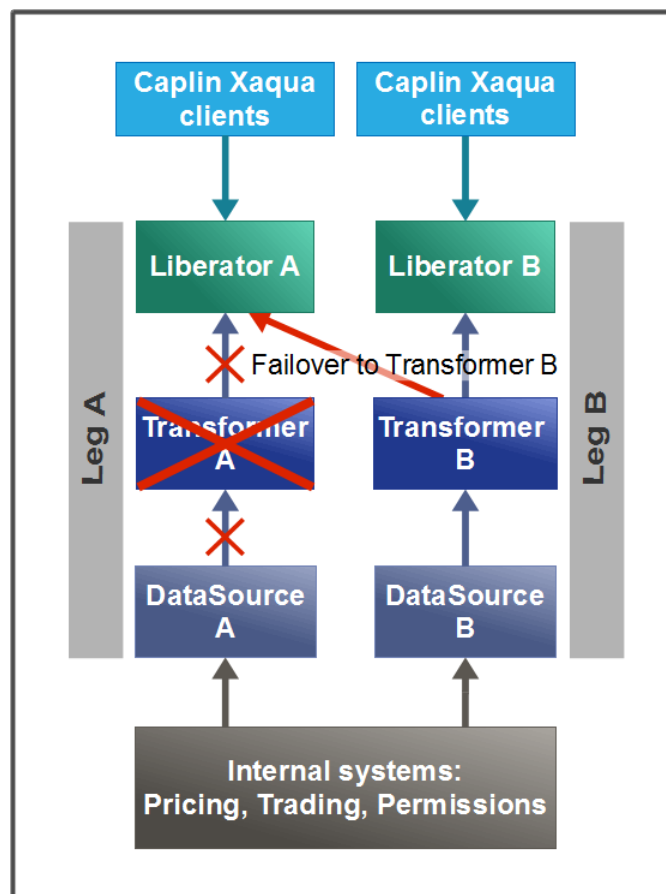


Liberator failover

The end-user does not have to explicitly log in to the alternate Liberator; this happens automatically in the background using a KeyMaster token hosted by the application server. For more information see [Authenticating client sessions](#)^[17].

3.2 Transformer failover

On loss of a Transformer instance, or loss of connection to the Transformer, the connected Liberators use failover connections to maintain quality of service. Clients remain connected to the same Liberator instance, so the failover is not apparent to them.



Transformer failover

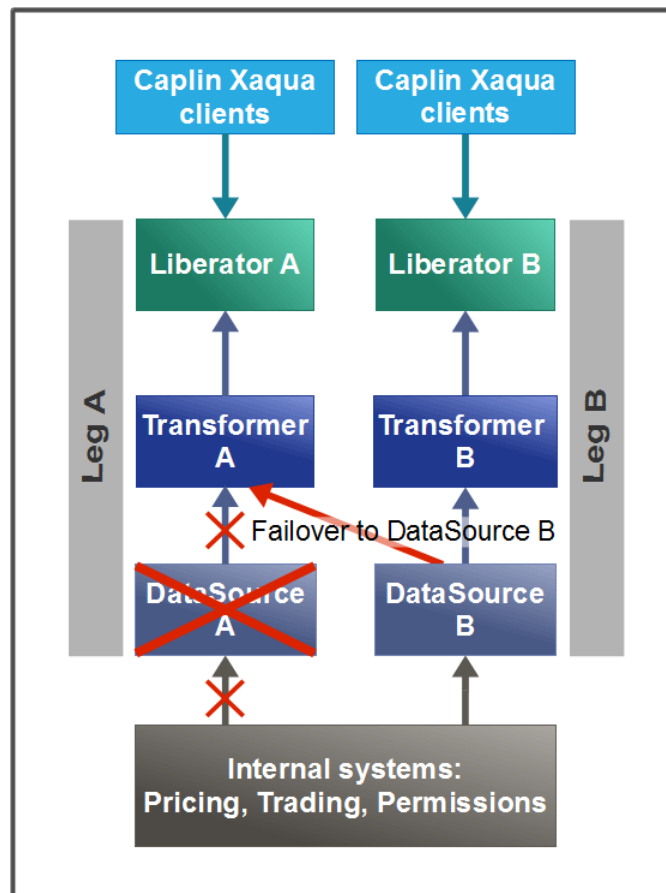
A Liberator's subscriptions to streaming data are automatically reinstated when the DataSource application that supplies this data (Transformer A in this example) fails over.

For example if Liberator A subscribes to 5,000 streamed instruments served by Transformer A, and this Transformer becomes unavailable, Liberator A automatically resubscribes to these instruments on Transformer B.

For simplicity of presentation, the diagram implies that before the failover, Liberator A has only subscribed to instruments from Transformer A, and after failover it requests all those instruments again from Transformer B. In reality, Liberators and Transformers are typically set up to balance subscriptions across all the DataSource adapters, which reduces the number of subscriptions that have to be switched over when a failover occurs.

3.3 DataSource failover

On loss of a DataSource instance, or loss of connection to the DataSource, the connected Transformers use failover connections to maintain quality of service. Clients remain connected to the same Liberator instance, so the failover is not apparent to them.



DataSource failover

A Transformer's subscriptions to streaming data are automatically reinstated when the DataSource application that supplies this data (DataSource adapter A in this example) fails over.

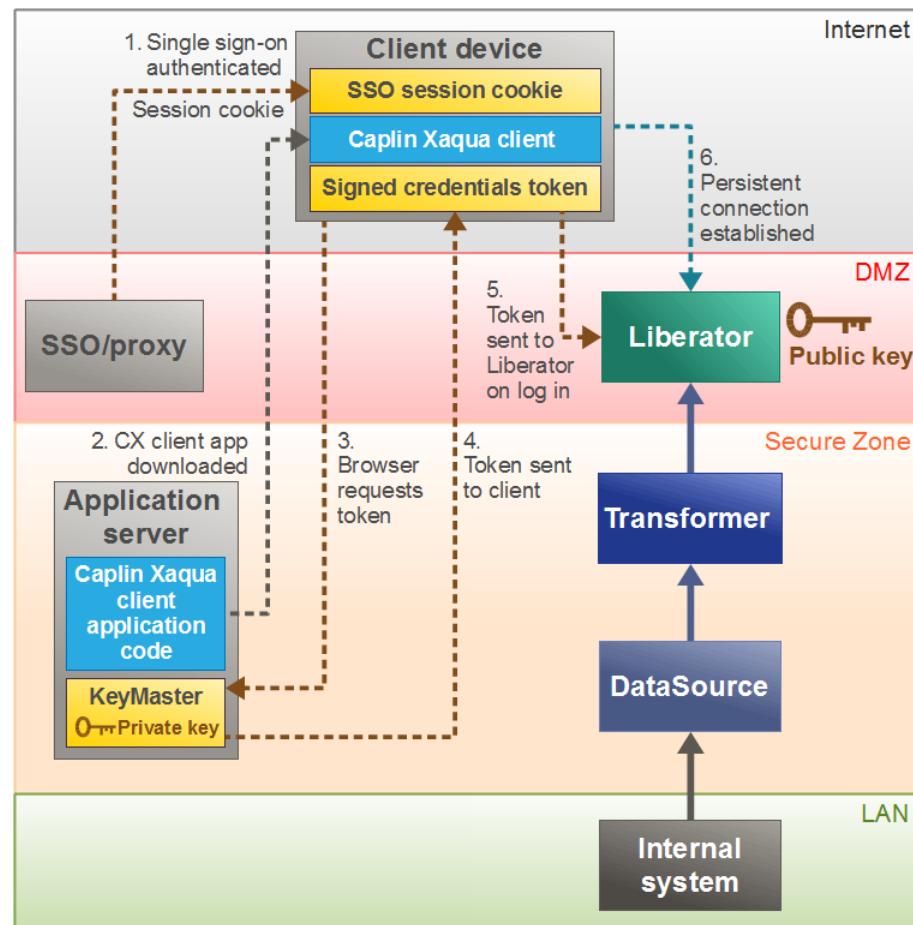
For example if Transformer A subscribes to 5,000 streamed instruments served by DataSource A, and this DataSource becomes unavailable, Transformer A automatically resubscribes to these instruments on DataSource B.

For simplicity of presentation, the diagram implies that before the failover, Transformer A has only subscribed to instruments on DataSource A, and after failover it requests all those instruments again from DataSource B. In reality, Liberators and Transformers are typically set up to balance subscriptions across all the DataSource adapters, which reduces the number of subscriptions that have to be switched over when a failover occurs.

4 Authenticating client sessions

Client sessions should be authenticated through the Bank's existing single sign-on (SSO) system. Once an end-user has signed on, the Caplin Xaqua client logs the user on to the Liberator (via the StreamLink library built into the application) This can be done automatically using Caplin KeyMaster. Caplin KeyMaster is software that integrates Caplin Liberator with an existing SSO system, so that end-users do not have to explicitly log in to the Liberator server in addition to logging in to the Bank's single sign-on server.

The following diagram shows the role of each component in handling sign-on and authentication of client sessions, and shows the main steps of the authentication process, assuming that KeyMaster is being used. It assumes the Caplin Xaqua client runs in a browser.



**Authenticating a client session
using Caplin KeyMaster**

The end-user signs on to the Bank's portal site. The sign-on process is handled by the Bank's existing SSO system, which sends a session cookie back to the browser.

1. The end-user navigates to the Caplin Xaqua client URL protected by the SSO, and downloads the Caplin Xaqua client application.
2. The Caplin Xaqua client application requests a user credentials token from the Caplin KeyMaster running on the application server. This request is secured by the SSO.
3. KeyMaster generates a signed user credentials token using a private key and sends this to the client application.
4. The browser uses the token to log into Liberator.
5. Liberator use the public key to verify the signature in the token. If the verification succeeds, the Liberator sets up a persistent connection to the client application.

Note that KeyMaster tokens may only be used once and expire after a configured timeout.

Impact of Liberator failover

If Liberator fails over, repeated authentication against the SSO is not required; instead, the Caplin Xaqua client requests a new KeyMaster token [steps 2-5] before logging into the alternate Liberator instance.

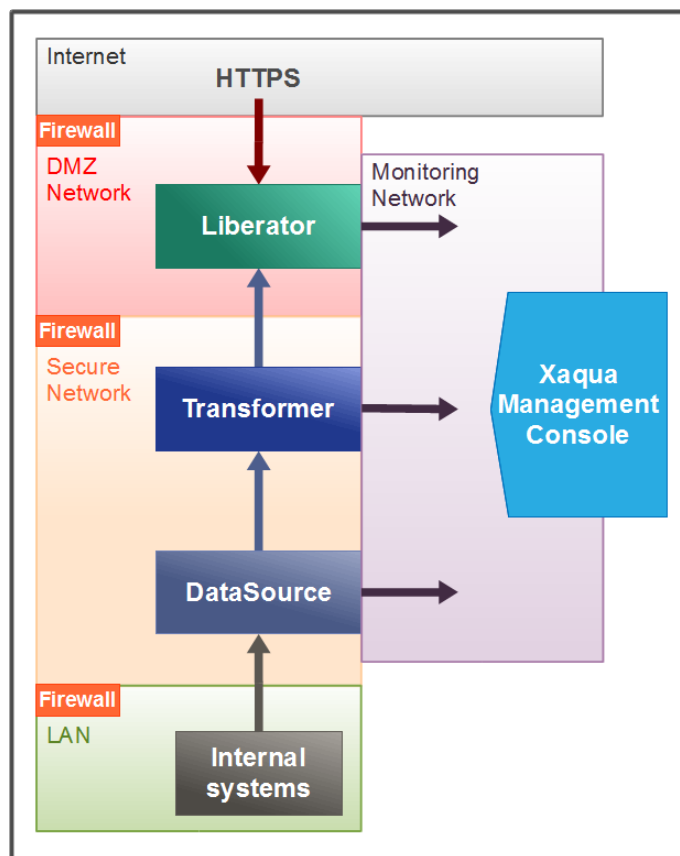
Tip: For more information on client authentication using KeyMaster, see the **KeyMaster Overview**.

5 Deploying the Xaqua Management Console (XMC)

The Xaqua Management Console (XMC) allows you to monitor the behavior and performance of all Caplin Xaqua components including Caplin Liberator, Caplin Transformer, and DataSources. The XMC establishes separate JMX™ connections to each Caplin Xaqua component. Each server component listens on an RMI registry port and an RMI client port. The JMX runtime uses a static JMX service URL to perform a JNDI lookup of the JMX connector stub in the RMI registry.

Authentication of XMC sessions follows the same principles as client session authentication (see [Authenticating client sessions](#)^[17]). First the XMC requests a KeyMaster Credentials token from the application server, then it uses the token for secure login to Liberator.

The security implications of connecting a monitoring device directly into the live business data networks should be considered. Caplin Xaqua allows you to run your monitoring application on a completely different network, should this be a security requirement; this is shown in the following diagram:



Separate network for JMX monitoring

In the diagram, the Caplin Xaqua components and internal systems are on separate networks and communicate across firewalls. The *monitoring connections* (RMI Registry and RMI Client) for Liberator, Transformer, and the DataSource adapter connections, are to a Monitoring Network which is completely separate from the other networks. The Xaqua Management Console is also connected to the Monitoring Network.

6 Deployment requirements

These sections list the software and hardware requirements for deploying Caplin Xaqua.

6.1 Operating systems

Caplin Xaqua components run on the following operating systems:

Caplin Xaqua component	Operating systems
Liberator 4.x Transformer 4.x	Red Hat Enterprise Linux® 4 or 5 Sun® Solaris™ (SPARC) 8 or 10 Java™ Runtime 1.4.2, 5.0 or 6.
Java DataSources	Java Runtime 1.4.2, 5.0 or 6 on Windows, Linux or Solaris.

6.2 Supported hardware

Caplin Xaqua components run on the following hardware:

Caplin Xaqua component	Runs on hardware:
Liberator 4.x / Transformer 4.x	x86 (Linux only): 32 or 64 bit Sun SPARC (UltraSPARC IV, IV+, T2): 32 or 64 bit

6.3 Hardware recommendations

Caplin makes hardware specification recommendations based on the expected load profile and latency requirements. In order to make an initial sizing of Caplin Xaqua component instances and hardware specifications, you should identify the following load profile metrics:

Metric	Variable
The number of concurrent users	n
The number of groups of instruments that can be subscribed to (for example FX majors could be one group, FX Synthetic instruments could be another group, so that $g_{max}=2$)	g_{max}
The total number of instruments available for subscription (within a particular group of instruments)	i_g
The number of subscriptions per user (by instrument group)	s_g
The update frequency of an instrument (by instrument group)	u_g
The number of dynamic fields (per instrument group)	f_g
The average field size in characters (bytes) (per instrument group)	b_g

Liberator bandwidth requirements

The following formula gives an estimation of Liberator output bandwidth; that is, the raw data rate from Liberator to clients, not including protocol overheads:

$$\text{Output bandwidth (bits/s)} = 8n \sum_{g=1}^{g_{max}} s_g u_g f_g b_g$$

Similarly, the following formula gives an estimation of Liberator input bandwidth; that is, the raw data rate from DataSources to Liberator:

$$\text{Input bandwidth (bits/s)} = 8 \sum_{g=1}^{g_{max}} i_g u_g f_g b_g$$

Tip: For latency measurements at particular load profiles, please see the document **Benchmarking Caplin Liberator**, or contact your Caplin Representative.

Worked example

The following example illustrates how to calculate the raw bandwidth requirements for Liberator, given two sets of instruments – FX Major currency pairs (for example GBPUSD or EURUSD), and FX Synthetic instruments (for example MXNJPY):

Metric	Example
Number of concurrent users	n=15,000 users
Number of groups of instruments that can be subscribed to	gmax=2
Number of instruments in total available for subscription	i _{FXMajors} = 50 FX major currency pairs i _{FXSynthetics} = 1000 FX synthetic instruments
Number of subscriptions per user (by instrument group)	S _{FXMajors} = 20 FX major subscriptions per user S _{FXSynthetics} = 10 FX synthetics subscriptions per user
Update frequency of an instrument (by instrument group)	U _{FXMajors} = 4 FX major updates per second U _{FXSynthetics} = 2 FX synthetics updates per second
Number of dynamic fields (per instrument group)	f _{FXMajors} = 5 fields (bid, ask, bidsize, asksize, update-time) f _{FXSynthetics} = 5 fields
Average field size (per instrument group)	b _{FXMajors} = 10 characters per field (e.g. "12345.6789") b _{FXSynthetics} = 10 characters per field

An estimation of Liberator output bandwidth (towards clients) is given by:

$$\text{Output bandwidth} = 8n \sum_{g=1}^{gmax} s_g u_g f_g b_g = 8 \times 15000 \times ((20 \times 4 \times 5 \times 10) + (10 \times 2 \times 5 \times 10)) = 600 \text{ Mbit/s}$$

An estimation of Liberator input bandwidth (from DataSources) is given by:

$$\text{Input bandwidth} = 8 \sum_{g=1}^{gmax} i_g u_g f_g b_g = 8 \times ((50 \times 4 \times 5 \times 10) + (1000 \times 2 \times 5 \times 10)) = 880 \text{ Kbit/s}$$

A mid-ranged server machine typically supports output bandwidths of between 1 and 2 Gigabits/sec before hardware resources significantly affect the latency of transmission. The calculations for this example indicate that a single Liberator instance running on such a server should be capable of servicing the load profile (although Caplin always recommends multiple instances for resilience).

Note: The formulae given here allow you to obtain an initial estimate of the number of Liberators required for the deployed Caplin Xaqua installation. However, to obtain a firmer estimate, you should quantify actual bandwidth requirements and latency tolerances by load testing and/or by referring to the document **Benchmarking Caplin Liberator**.

Typical recommended Liberator server hardware specifications

Hardware component	Recommended type / quantity
Processor (x86)	2x Intel® Xeon® 3000, 5000, 7000 (quad-core) or: 2x 4P AMD Opteron™ (quad-core) For example: 2x E7420 Xeon
Processor (Sun SPARC)	SPARC64 VII or UltraSPARC T2 For example: M3000 2.75 GHz VII+
RAM	8 GB
Network	3 x Gigabit Ethernet card
Storage	150 Gigabytes minimum.

7 Appendix A: Default port allocations

The ports through which Caplin Xaqua components communicate with each other, and with external entities, are defined in configuration files. The following sections show the port settings:

- ◆ When Caplin Xaqua is shipped as part of a Caplin Trader install kit.
- ◆ When Caplin Xaqua is shipped for use with a Caplin Xaqua client that is not based on Caplin Trader.

7.1 Default Liberator ports

The following table shows the default port allocations used by Liberator, and the configuration items you can set to change these allocations.

When Liberator is shipped as part of a Caplin Trader install kit, the default port allocations are explicitly defined in the configuration files (column **A** in the table). When Liberator is supplied for use with other client technologies, the port allocations are not defined in the shipped configuration files, so Liberator uses its inbuilt default values (column **B** in the table), unless you explicitly change them by adding entries to the configuration files.

For more information on the configuration items, see the **Caplin Liberator Administration Guide**.
For information about the ports used for connections to the Xaqua Management Console, see **Caplin Xaqua: Getting Started With The XMC**.

For connections from	A) Caplin Xaqua configured for Caplin Trader: Liberator listens on port	B) Caplin Xaqua configured for other Caplin Xaqua clients: Liberator listens on port	Configuration item
HTTP clients ¹	50180	8080	http-port (set in <i>rtttd.conf</i>)
HTTPS clients ²	50181	4443	https-port (set in <i>rtttd.conf</i>)
Client socket connections ³	50182	15000	direct-port (set in <i>rtttd.conf</i>)
Client socket connections using SSL ^{3,4}	not used	15001	directssl-port (set in <i>rtttd.conf</i>)
Transformer & DataSources ⁵	50100	0	datasrc-port (set in <i>rtttd.conf</i>)
Xaqua Management Console (RMI Registry)	50120	1099	rmi-registry-port (set in <i>jmx.conf</i>)
Xaqua Management Console (RMI Client)	50130	1100	rmi.client.port (set in <i>java.conf</i>)

Notes:

1. HTTP: Port 50180/8080 is typically assigned only if Liberator is downstream from a hardware SSL wrapper, otherwise you should normally change the HTTP port to 80 (this allows traffic to pass through an unmodified firewall and be accessible to clients).
2. In live operation, connections from HTTPS clients are typically allocated to port 443 (this allows traffic to pass through an unmodified firewall and be accessible to clients), unless the Liberator is deployed behind port translation hardware.
3. This port is typically used in Caplin Xaqua deployments where the client is not browser-based. It is not normally used in Caplin Trader projects.
4. To enable client socket connections using SSL, you must set the configuration item **directssl-enable**.
5. The Caplin Xaqua default of 0 means that no DataSource applications can connect to Liberator. To allow such connections, set **datasrc-port** to a unique non-zero value.

7.2 Default Transformer ports

The following table shows the default port allocations used by Transformer, and the configuration items you can set to change these allocations.

When Transformer is shipped as part of a Caplin Trader install kit, the default port allocations are explicitly defined in the configuration files with the values shown in column **A** of the table. When Transformer is supplied for use with other client technologies, the port allocations defined in the shipped configuration files are different – see column **B** of the table.

For more information on the configuration items, see the **Caplin Transformer Administration Guide**.
For information about the ports used for connections to the Xaqua Management Console, see **Caplin Xaqua: Getting Started With The XMC**.

For connections from	A) Caplin Xaqua configured for Caplin Trader: Transformer listens on port	B) Caplin Xaqua configured for other Caplin Xaqua clients: Transformer listens on port	Configuration item
DataSources	50101	25010	datasrc-port (set in <i>transformer.conf</i>)
UDP command interface port ¹	50161	10010	udp-port (set in <i>transformer.conf</i>)
Xaqua Management Console (RMI Registry)	50121	1099	rmi-registry-port (set in <i>jmx.conf</i>)
Xaqua Management Console (RMI Client)	50131	1100	rmi.client.port (set in <i>java.conf</i>)

Notes:

1. This port is not normally used in Caplin Trader projects.

7.3 Default DataSource adapter ports

The following table shows the default port allocations that are used by DataSource adapters for monitoring connections, and the configuration items you can set to change these allocations.

When a DataSource adapter is shipped as part of a Caplin Trader install kit, the default port allocations are explicitly defined in the configuration files with the values shown in column **A** of the table. When the DataSource adapter is supplied for use with other client technologies, the port allocations defined in the shipped configuration files are different – see column **B** of the table.

For information about the ports used for connections to the Xaqua Management Console, see **Caplin Xaqua: Getting Started With The XMC**.

For connections from	A) Caplin Xaqua configured for Caplin Trader: DataSource listens on port	B) Caplin Xaqua configured for other Caplin Xaqua clients: DataSource listens on port	Configuration item
Xaqua Management Console (RMI Registry) ¹	50122, 50123, 50124, 50125, 50126, 50127	1099	rmi-registry-port (set in <i>jmx.conf</i>)
Xaqua Management Console (RMI Client) ¹	50132, 50133, 50134, 50135, 50136, 50137	46000	rmi.client.port (set in the jvm-options standard startup option for the JVM)

Notes:

1. Each DataSource adapter should be configured to listen for XMC RMI Registry and RMI Client connections on unique ports. Column A shows the list of ports used by the DataSource adapters provided with the Caplin Trader install kit. Column B shows the *default* port settings for a DataSource adapter supplied for use with client technologies other than Caplin Trader; you will need to change these for individual DataSource adapters to make them unique.

8 Appendix B: Reverse proxies and Liberator

Your organization may have security policies that route incoming network traffic (typically from the Internet) to a server located in a **DMZ**. Such a “reverse proxy” server typically acts as a firewall, hiding the details of the Web servers from the Internet clients. It may also be used as a load balancer to distribute the incoming traffic across the available web servers. Although it is possible to use a reverse proxy in front of a set of Liberator servers, this is not recommended, for the following reasons:

- ◆ Performance

Liberator is designed to stream fast moving data to a large number of concurrent clients. Reverse proxies are designed for more traditional HTTP requests and do not generally scale well for the kind of traffic Liberator has to deal with.

- ◆ Streaming support

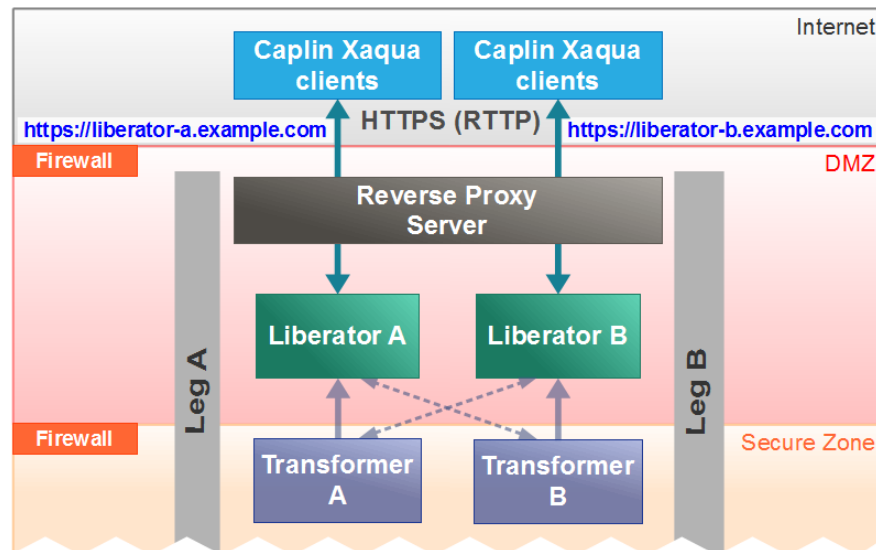
When reverse proxies are used as load balancers they can prevent real time streaming of data. A proxy will usually try to employ some 'sticky' logic to make sure that once a client has initiated communication with a particular web server via the reverse proxy, it continues to communicate only with this server. In a general purpose reverse proxy server this logic is rarely perfect, but when the servers behind the proxy are Liberators, it must be so, to ensure that each Caplin Xaqua client always receives streaming updates from the Liberator to which it is connected.

- ◆ Security

Liberator is penetration tested so putting a reverse proxy between Liberator and the client does not necessarily enhance security.

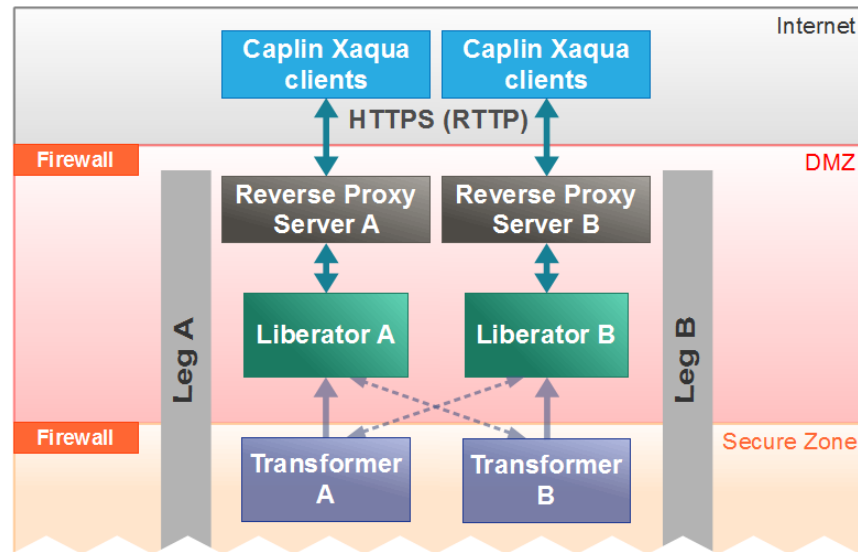
Tip: If your security policy stipulates that reverse proxy servers must be used, then it is recommended that you configure your Caplin Xaqua installation in one of the following ways, to overcome the performance and streaming issues described above.

- Configure the reverse proxy server so that it recognizes each Liberator as having a different IP address / hostname, rather than treating all Liberators as having the same virtual IP address / hostname. This allows the StreamLink library in the Caplin Xaqua client to address each Liberator separately. See the following diagram, where the two Liberators have virtual IP addresses liberator-a.example.com and liberator-b.example.com respectively:



Reverse proxy recognizing separate
Liberator IP addresses

- Alternatively, install a reverse proxy server for each Liberator, as shown in the following diagram. Each proxy server has a different IP address, which is the virtual IP address of the single Liberator behind it.



Reverse proxy for each Liberator

9 Glossary of terms and acronyms

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

Term	Definition
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
Caplin KeyMaster	Caplin KeyMaster is software that integrates Caplin Xaqua with any existing web-based 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 is usually integrated with a single sign-on system.
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.
Caplin Transformer	Caplin Transformer is an event-driven real-time business rules engine.
Caplin Trader	A web application framework for financial trading. An application constructed with Caplin Trader is a Caplin Xaqua client application. Caplin Trader was formerly called "Caplin Trader Client".
Caplin Xaqua	A framework for building single-dealer platforms that enables banks to deliver multi-product trading direct to client desktops.
Caplin Xaqua client	A client desktop or web application that interfaces with Caplin Xaqua to deliver multi-product trading to end-users. The application can be implemented in any technology that is supported by Caplin Xaqua; for example Ajax, Microsoft .NET, Microsoft Silverlight™, Adobe Flex™, and Java™. Also see Caplin Trader .
DMZ	<u>D</u> emilitarized <u>Z</u> one A physical or logical network that contains and exposes an organization's external services to a larger untrusted network, usually the Internet. Definition from Wikipedia contributors, "DMZ (computing)", <i>Wikipedia, The Free Encyclopedia</i> , http://en.wikipedia.org/wiki/DMZ_(computing) (accessed Sept 2010).
DNS	<u>D</u> omain <u>N</u> ame <u>S</u> ystem (or <u>S</u> ervice, or <u>S</u> erver) The Internet system that translate names into IP addresses.
DataSource	DataSource is the internal communications infrastructure used by Caplin Xaqua 's server components such as Caplin Liberator , and DataSource adapters . DataSource is also used as a synonym for DataSource application when the context is obvious.

Term	Definition
DataSource application	<p>A Caplin Xaqua application that uses the Caplin DataSource APIs to communicate with other Caplin Xaqua applications through the DataSource protocol.</p> <p>Also known as a DataSource peer.</p>
DataSource adapter	<p>A DataSource application that integrates with an external (non-Caplin) system, exchanging data and/or messages with that system.</p>
DataSource peer	<p>See DataSource application.</p>
Failover	<p>The transfer of operation from a failed hardware or software component to an alternative copy of the component, ensuring uninterrupted provision of service.</p>
Hot site	<p>A site that hosts a live (operational) installation of Caplin Xaqua.</p>
JDBC	<p><u>J</u>ava <u>D</u>atabase <u>C</u>onnectivity</p> <p>A Sun Microsystems standard defining how Java applications access data in a database.</p>
JMX	<p><u>J</u>ava <u>M</u>anagement <u>E</u>xtensions</p> <p>A facility within the Java Platform, Standard Edition (J2SE) that provides a standard way of monitoring and managing resources across a network.</p>
JNDI	<p><u>J</u>ava <u>N</u>aming and <u>D</u>irectory <u>I</u>nterface</p> <p>A Java API that enables Java applications to access multiple naming and directory services.</p>
LAN	<p><u>L</u>ocal <u>A</u>rea <u>N</u>etwork</p> <p>A communication network that connects together computers and other devices across a small geographic area, such as within a building or other single site.</p> <p>Also see WAN.</p>
RMI	<p><u>R</u>emote <u>M</u>ethod <u>I</u>nvocation</p> <p>A Java API that enables Java objects to communicate remotely with other Java objects.</p>
RTTP	<p><u>R</u>eal <u>T</u>ime <u>T</u>ext <u>P</u>rotocol.</p> <p>Caplin's protocol for streaming real-time financial data from Caplin Liberator servers to client applications, and for transmitting trade messages between clients and Caplin Liberator in both directions.</p>
SSO	<p>See Single sign-on.</p>
Single sign-on	<p>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.</p>
StreamLink	<p>The StreamLink libraries connect client applications to Liberator through the RTTP protocol. They provide an object oriented API that gives access to RTTP functionality.</p>
User credentials	<p>Information used to authenticate a user; for example, a user name and password.</p>
User credentials token	<p>A data structure, containing user credentials, that is passed from one application to another in order to authenticate the user.</p>

Term	Definition
WAN	<u>Wide Area Network</u> A computer network that spans a wide geographical area (for example, between cities, countries, or continents), usually connecting together two or more LANs .
XMC	See <u>Xaqua Management Console</u> .
Xaqua Management Console	A Java application that communicates with Caplin Xaqua components through JMX, and allows you to monitor and control these components through a GUI. Formerly called the EMC (Enterprise Management Console).

Contact Us

Caplin Systems Ltd
Cutlers Court
115 Houndsditch
London EC3A 7BR
Telephone: +44 20 7826 9600
Fax: +44 20 7826 9610
www.caplin.com

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.

This publication may contain links to third-party web sites; Caplin Systems Limited is not responsible for the content of such sites.