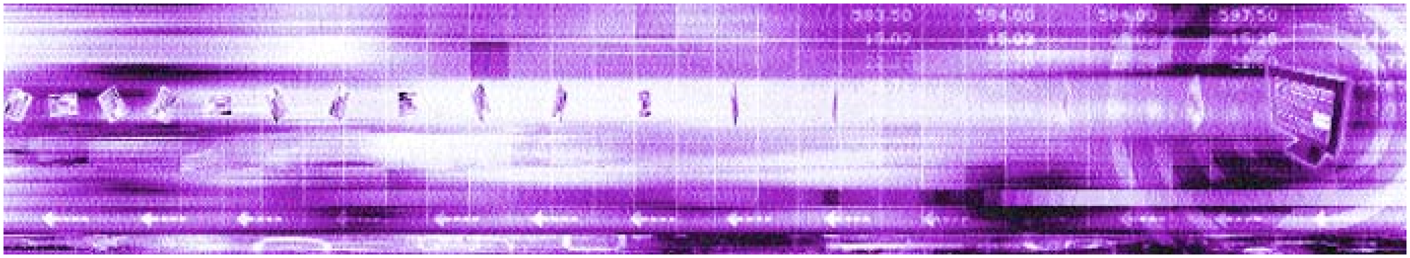


CAPLIN AUTH SDK 4.4

Overview

April 2007

Preface 3

What this document contains 3

Who should read this document 3

Typographical conventions 3

Acronyms and glossary. 4

Feedback 5

Overview. 6

What is an Auth Module? 6

The Caplin Platform architecture. 7

How Auth Modules Work. 9

Functions of an Auth Module 11

Standard Auth Modules 12

Openauth 12

Config Auth 12

XML Auth 12

Writing your own Auth Module. 13

Auth Module structure. 13

Creating an Auth Module in C. 13

Creating a Java Auth Module 13

Example Auth Module created using an SDK. 14

1 Preface

1.1 What this document contains

This document gives an overview of the Auth Modules which come with Caplin Liberator and of how you can use the Auth Module SDK to create your own modules for controlling user authentication and object permissioning.

1.2 Who should read this document

This document is written for administrators wanting to get an idea of the different ready made Auth Modules which come with Caplin Liberator, and for developers needing to create custom Auth Modules for use with a Caplin Liberator. In the introduction to the Auth SDK this document describes functions and processes requiring a knowledge of programming in C and Java.

1.3 Typographical conventions

This document uses the following typographical conventions to identify particular elements within the text.

Type	Use
Arial Bold	Function names and methods. Other sections and chapters within this document.
Arial Italic	Parameter names and other variables.
Times Italic	File names, folders and directories.
`Courier`	Program output and code examples.
❖	Information bullet point.
■	Instruction.

1.4 Acronyms and glossary

Auth Module	An application that performs authentication and authorization functions.
Authentication	Permitting a particular user to login to Caplin Liberator in order to access streaming data. Also known as entitlement.
Authorization	Permitting particular data to be viewed.
Caplin Liberator	A suite of software applications and components for publishing real-time information using RTTP protocol over IP networks. Caplin Liberator collects real-time data from one or more sources and redistributes it to suitably permissioned RTTP subscribers.
DACS	Reuters's system for administering market data and permissioning trading floor, enterprise and Internet users to data, transactions, and other services.
DTD	Document Type Definition, a specification that accompanies an XML document and identifies what the tags are that separate paragraphs and how each is to be processed.
Field	A data element of an object, identified by a field name or field number and with a data value of a string.
Object	There are several types of RTTP object: Directory, Page, Record, News headline, News story and Chat object. Each type is identified by a three digit number.
Parameter	A data element of an object, identified by a field name or field number and with a data value of a string. Same as "field".
Parser	A program that receives input in the form of program instructions, markup tags or some other interface and breaks them up into parts (for example, objects, methods and their attributes) that can then be managed by other programs.

RTTP	RTTP (Real Time Text Protocol) is a web protocol developed by Caplin Systems Ltd that implements advanced real-time streaming for almost all types of textual information, including logical records, news and free-format pages.
Symbols	The letters used to uniquely identify a financial instrument (e.g. the symbol for Microsoft's common stock on the Nasdaq market is MSFT). Many symbol naming conventions (symbologies) exist, but each is applied consistently in each market, and one symbology is used across all North American equity markets.
UDP	UDP (User Datagram Protocol) is a communications protocol that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol.
XML	XML (Extensible Markup Language) contains markup symbols to describe the contents of a page or file. An XML file can be processed purely as data by a program or it can be stored with similar data on another computer or displayed. XML is "extensible" because, unlike HTML, the markup symbols are unlimited and self-defining.
XML tags	The sequence of characters or other symbols that you insert at certain places in a file to indicate how the file should look when it is printed or displayed or to describe the document's logical structure. Tags that have other tags within them are called parent tags; those within are called child tags.

1.5 Feedback

Customer feedback can only improve the quality of Caplin 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.

2 Overview

2.1 What is an Auth Module?

Caplin Liberator supports a modular system for handling authentication of users and entitlement of objects. This allows users to be authenticated, objects to have permissions loaded, read and write permissions for a user to be checked and object name mappings to be performed.

- ❖ Authentication is the process of determining whether someone is who they say they are. In networks such as the Internet, authentication is commonly done through the use of logon passwords: knowledge of the password is assumed to guarantee that the user is authentic. The user must know and use the declared password.

- ❖ Authorization is the process of giving someone permission to do or have something. A system administrator defines which users are allowed access to which files. Authorization is sometimes seen as both the preliminary setting up of permissions by a system administrator and the actual checking of the permission values that have been set up when a user is getting access.

For details on how to create your own Auth Modules, refer to the companion Javadoc and Doxygen documents **Caplin Liberator Auth Module SDK Developer's Guide** for Java or C respectively.

2.2 The Caplin Platform architecture

Figure 2-1 below shows a detailed illustration of the Caplin Platform architecture, including all available products and the protocols they use to communicate.

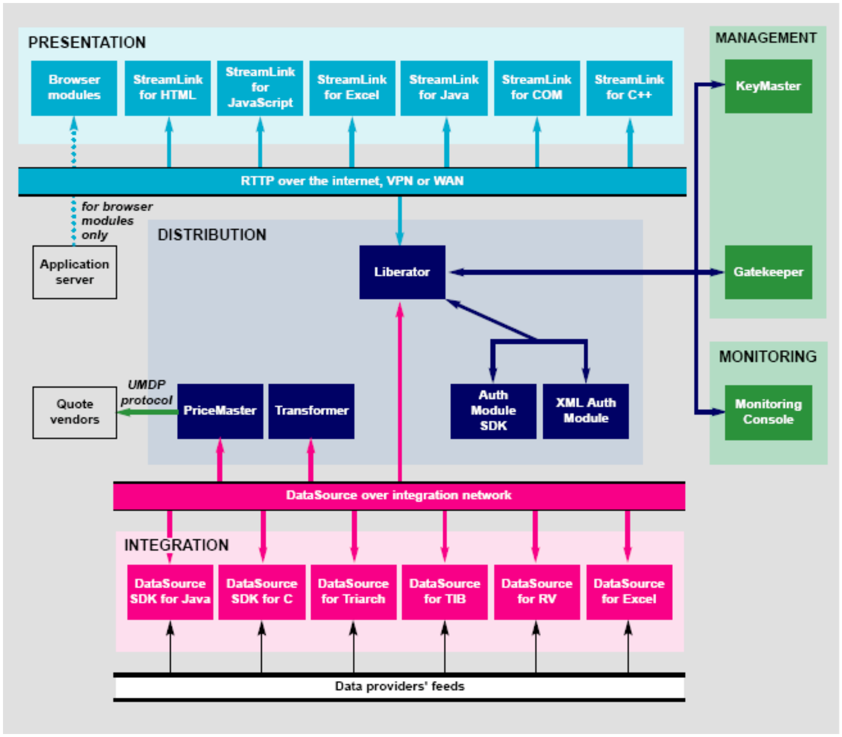


Figure 2-1: Caplin's architecture

Figure 2-2 below shows a simplified Caplin platform diagram and highlights an Auth Module and its place in the platform.

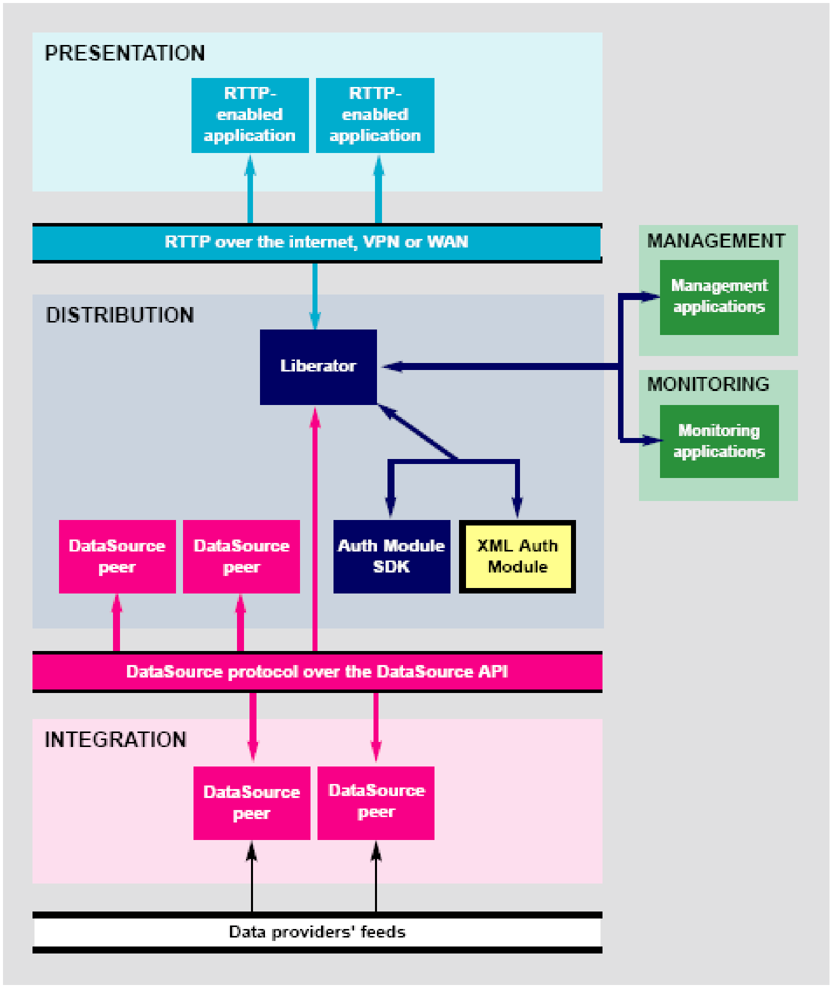
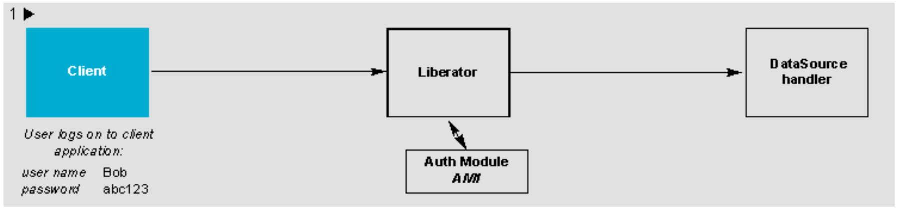
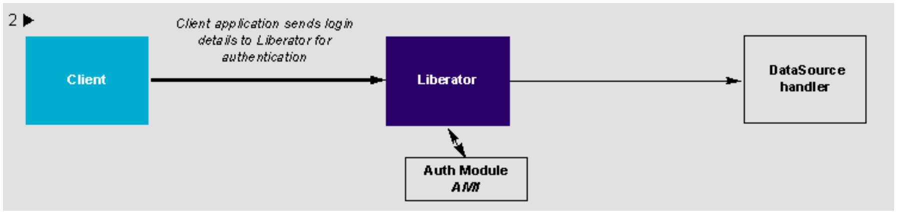


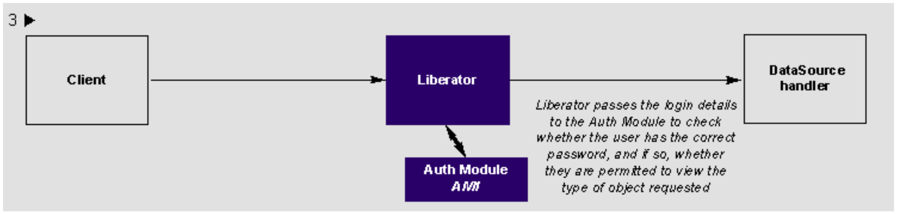
Figure 2-2: Auth Module in Caplin's architecture

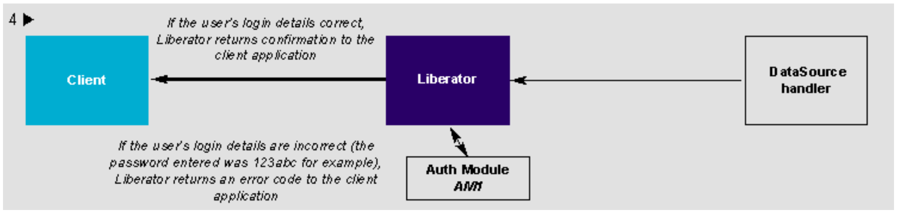
2.3 How Auth Modules Work

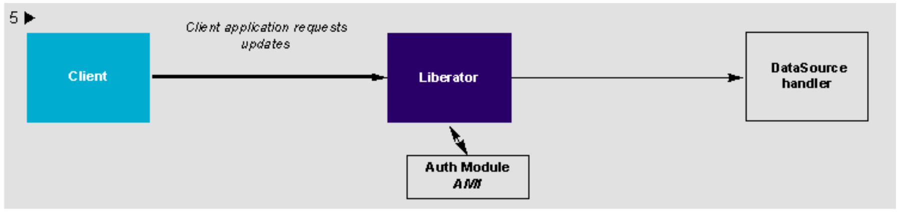
The diagrams below give a step by step illustration of how Auth Modules are used to perform permissioning and entitlement.

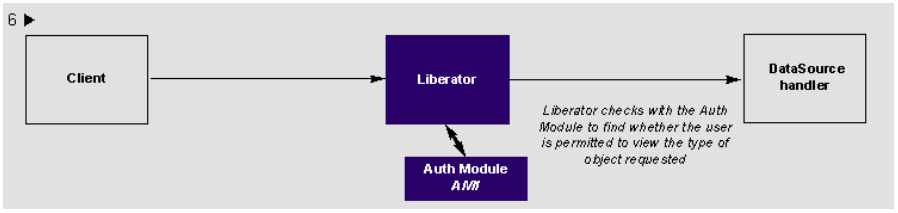


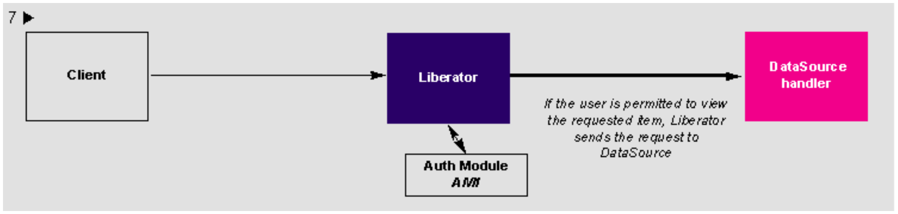

2 ▶
Client
Client application sends login details to Liberator for authentication
Liberator
DataSource handler
Auth Module AMI

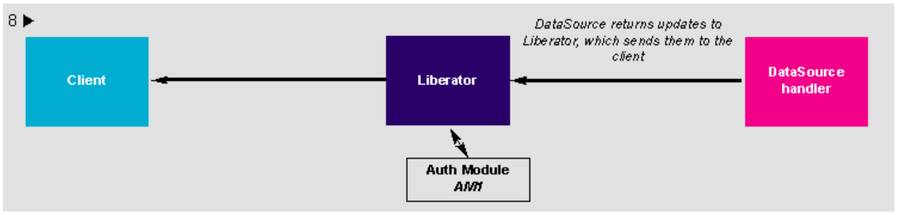

3 ▶
Client
Liberator
DataSource handler
Auth Module AMI
Liberator passes the login details to the Auth Module to check whether the user has the correct password, and if so, whether they are permitted to view the type of object requested


4 ▶
If the user's login details correct, Liberator returns confirmation to the client application
Client
Liberator
DataSource handler
If the user's login details are incorrect (the password entered was 123abc for example), Liberator returns an error code to the client application
Auth Module AMI


5 ▶
Client application requests updates
Client
Liberator
DataSource handler
Auth Module AMI


6 ▶
Client
Liberator
DataSource handler
Liberator checks with the Auth Module to find whether the user is permitted to view the type of object requested
Auth Module AMI


7 ▶
Client
Liberator
DataSource handler
If the user is permitted to view the requested item, Liberator sends the request to DataSource
Auth Module AMI


8
DataSource returns updates to Liberator, which sends them to the client
Client
Liberator
DataSource handler
Auth Module AM1

2.4 Functions of an Auth Module

The primary function of an Auth Module is to check the user on initial log-on, and to check whether that user has permission to read from or write to a given object.

Its functions include:

- Reject or invalidate users - for instance reject a user when they try to log in with the same details as a user who is already logged on.
- Integrate with Caplin KeyMaster to support single sign-on capability.
- Allow object-based permissioning. This means only allowing access to an object if it contains a certain value in one of its fields (as opposed to allowing access based on the symbol name of the object itself).
- Check the permissioning for each update.
- Return a range of user defined codes. This allows an Auth Module and a custom client application to provide custom messages for denying access.
- Map a user-requested object name to a different name in the server (and therefore at upstream DataSource(s) too). This can be useful for providing different data under the same symbol name to different users or groups of users, for example in order to provide preferential currency spreads to certain customers, or customized data for particular logins.
- Implement session management – actions such as ejection and re-validation of Liberator sessions are supported.

3 Standard Auth Modules

Caplin Liberator is equipped with three standard Auth Modules: xmlauth, openauth and cfgauth.

These modules and their implementation are described in more detail in the companion document **Caplin Liberator Administration Guide** (in the chapter “Authentication and Entitlement”), and also in the **XMLAuth Administration Guide**.

3.1 Openauth

This is the simplest Auth Module possible and can be used for systems where no authentication or authorization is needed.

Openauth will allow any username to enter the system and with any password. It can also specify whether all users have either or both read and write access to any object in the system.

3.2 Config Auth

This module allows the number of users and the types of objects they can read to be configured. It is intended for relatively low numbers of users where the usernames and other details do not need to be changed often.

3.3 XML Auth

This module enables programmers and system administrators to use XML to create their own permissioning structures and control entitlement to objects held on Caplin Liberator.

As XMLauth is more complex than the other standard modules, there is an accompanying document **XML Auth Admin Guide** which must be referred to for instructions on how to use this module.

4 Writing your own Auth Module

You can create your own Auth Modules for Caplin Liberator using one of the the Auth Module SDKs. The **Liberator Authentication SDK** is a library of C functions that you can use to develop an Auth Module. If you prefer to write the custom Auth Module in Java, you can use the classes provided in the **Liberator Java Authentication SDK**.

4.1 Auth Module structure

An Auth Module must implement two main features :

❖ The initialization function.

  This is an entry point into the module from the core server code.

❖ A set of C functions or Java methods that implement the functionality of the auth module.

This is all that is needed to set up and initialize the module. You may need to do some of your own initialization of resources within the initialization function. From this point it is left to the C functions or Java methods to handle their various tasks.

4.2 Creating an Auth Module in C

You create a custom C Auth Module by implementing a set of C functions. The API is callback-based. Calls are made on the various functions as and when an Auth action occurs in the Liberator. Every callback function must return an authentication code representing the result of the Auth call.

For more detailed information on how use the intialization function and implement the functions in a C Auth Module, see the **Liberator Authentication SDK** documentation.

4.3 Creating a Java Auth Module

You create a custom Java Auth Module by implementing the ***Authenticator*** interface. There is a default implementation adaptor (***AuthenticatorAdaptor)*** that can be extended if only a few custom methods are needed. The API is callback-based. Calls are made on the various ***Authenticator*** methods as and when an Auth action occurs in the Liberator. Every callback function must return an ***AuthenticationResult*** instance representing the result of the Auth call.

For more detailed information on how to on how use the intialization function and implement the the ***Authenticator*** interface, see the **Liberator Java Authentication SDK** documentation.

4.4 Example Auth Module created using an SDK

There is an Auth Module, called demoauth in the *doc/examples* directory of the Caplin Liberator installation. This is a basic implementation of an Auth Module using the functions and definitions described in this section.

Note: *Most of the possible features are included, but many are contained in preprocessor blocks, so by default they are not compiled.*

Contact Us

Triton Court
14 Finsbury Square
London EC2A 1BR
UK
Telephone: +44 20 7826 9600
Fax: +44 20 7826 9610

www.caplin.com

info@caplin.com