

BIC BITES

*BIC Bites are a series of short, high level, papers on topics of interest to BIC Members.
If there is a topic you would like to see as a BIC Bite, please contact info@bic.org.uk*

Comparison of two BIC standards: BIC Realtime for Libraries and the Library Communications Framework (LCF)

Background

The aim of this comparison is to clear up misconceptions about the content of each of these standards and how they relate to one another.

The following links lead to more detailed information on each of these standards:

- [BIC Realtime for Libraries](#)
- [Library Communications Framework \(LCF\)](#)

What are these standards for?

These two standards primarily address two sections of the library supply chain that are normally distinct:

- **BIC Realtime for Libraries** is concerned with the supply of books (including both physical and digital books) by publishers and supply chain intermediaries, such as distributors and wholesalers, to libraries.
- **LCF** is concerned with communication between kiosks or other terminal devices or applications and library management systems, to support self-service functions such as the lending of content items to library patrons.

Some areas of possible overlap have emerged, largely from a desire to extend the application of LCF into new areas, but still within a library lending context, such as the lending of e-books to library patrons by a publisher or intermediary that operates an e-book lending service on behalf of libraries, as well as supplying physical and digital products in the conventional way.

How do the standards compare in terms of technology?

Both BIC Realtime for Libraries and LCF have adopted a similar technical approach:

- Both standards are based upon the application of web service technologies and client-server communication design principles.
- Both standards define methods of communication between devices/applications in which the basic method of communication is the exchange of a pair of messages between client and server applications, referred to as “request” and “response” messages.
- Both standards specify how these request and response messages can be expressed in XML.
- BIC Realtime for Libraries already specifies JSON as an alternative to XML for messaging, and LCF plans to add JSON support in due course.

However, at a more detailed level, different technical approaches have been adopted, based upon the preferences of the principal contributors to the development of each standard.



BIC Realtime for Libraries:

- The message formats are based closely upon work done earlier on BIC Realtime for trade supply, which in its turn borrowed significantly from EDItEUR's EDItX supply chain communication standards.
- In line with the approach taken in developing BIC Realtime for trade supply, the BIC Realtime for Libraries APIs can be implemented in one of two ways, the choice being agreed between supply chain partners: either based upon formal, machine-readable web service definitions (WSDLs) provided as part of the standard and using HTTP POST communication methods for requests; or a simpler approach using HTTP GET requests.
- BIC Realtime for Libraries is concerned with business-to-business communications, so leaves some important details to supply chain partner agreement, such as communication security.

Library Communications Framework (LCF)

- LCF was originally conceived as defining a purely abstract framework for communication between self-service terminals and library management systems, largely based upon the earlier, proprietary 3M SIP2 technology.
- It was decided early on that the web service implementation should be based upon a set of widely-respected web service design principles known as REST, leading to a much more extensive use of the built-in features of the basic web protocol HTTP (such as HTTP response codes); there are no WSDL descriptions in LCF;
- LCF is increasingly concerned with communications over the public internet between terminal applications (which may be running on a patron's mobile phone) and library management systems, including handling payments, demanding a much greater level of engagement with issues of data privacy and data security than in BIC Realtime for Libraries.

Is there any overlap between the transactions supported by the two standards?

No, there is no overlap in functionality between the two standards.

Implementing both BIC Realtime for Libraries and LCF in a library management system – what are the issues?

There is nothing to prevent implementation of both BIC Realtime for Libraries and LCF in a library management system (LMS), but the differences in technical approach of the two standards present some challenges. LCF provides implementers with a much more comprehensive specification, leaving fewer options down to negotiation between implementers; BIC Realtime for Libraries, by contrast, specifies message payloads in detail, but leaves the choice of transport and security protocols much more up to trading partner agreement.

In the case of BIC Realtime for Libraries, the LMS implementer will need to agree details with supply chain partners – publishers' distributors or wholesalers – as to the precise method(s) of web service communication not fixed by the BIC standard. The supply chain partners may normally prefer to provide WSDL descriptions of their web service endpoints, in which case the LMS implementers will need to decide whether to use these in building the necessary client applications to incorporate into the LMS or develop their client applications without using the XSDL description, which is likely to involve more extensive testing with supply chain partners. Details such as communication security methods will also need to be agreed.

By contrast, implementers of LCF will find more of the details specified by the standard, so less needs to be discussed and agreed between LMS implementers and terminal application developers.

©Book Industry Communication 2021 All Rights Reserved

Written by Francis Cave, BIC Consultant.

Last updated 12th October 2021



the book industry's supply chain organisation