

# LOCKSS

James Currall

June 2004

## Abstract

For researchers, librarians, and publishers who are concerned that the digital material that has become the record of academic output will prove as ephemeral as the rest of the web, the LOCKSS approach is a promising solution. It has the capability to deliver a number of things:

Providing future generations of researchers with access to current literature for research, teaching, and learning.

Ensuring that current and future librarians have an inexpensive, robust mechanism, which they control, to provide their communities with long-term access to essential literature.

Providing current and future publishers with an assurance that their journals' editorial values and brands will be available only to authorised and authenticated readers.

## Contents

<b>1 Introduction</b>	<b>1</b>
<b>2 What is LOCKSS?</b>	<b>2</b>
<b>3 Why should a library be interested?</b>	<b>2</b>
3.1 Continuing access . . . . .	2
3.2 Digital preservation . . . . .	3
<b>4 How does it work?</b>	<b>3</b>
<b>5 What resources are required?</b>	<b>3</b>
<b>6 The Issues</b>	<b>4</b>
6.1 Critical Mass of UK Participants . . . . .	4
6.2 Plug-ins for UK Journals . . . . .	4
6.3 Collection Development . . . . .	4
<b>7 The LOCKSS Alliance</b>	<b>5</b>
<b>8 Technical Details</b>	<b>5</b>
<b>9 Acknowledgements</b>	<b>5</b>

## 1 Introduction

The concept behind the LOCKSS system is based on simple rules. Acquire lots of copies. Scatter them around the world so that it is easy to find some of them and hard to find all of them. Lend or copy your copies when other libraries need them. And collaborate only with competent and trusted libraries. In other words, in a similar way to the way books and libraries have operated since the invention of moveable type. A comparable system might have saved much of the world's literature lost in the fire that destroyed the Library of Alexandria in 415 AD!

After several years of testing in approximately 80 libraries around the world, LOCKSS became a production system in April 2004. This means that essentially the concept has been proved from a technical perspective and the emphasis has shifted decidedly into the content domain.

## 2 What is LOCKSS?

For centuries libraries and publishers have had clear roles: publishers provided information; libraries acquired it and kept it safe for reader access. There is no fundamental reason for the online environment to force institutions to abandon these roles although the advent of e-journals is threatening to do so.

The LOCKSS model capitalises on the traditional roles of libraries and publishers. LOCKSS creates low-cost, persistent digital "caches" of authoritative versions of web-delivered content such as e-journals. The LOCKSS software enables institutions to collect this material locally and store, preserve, and archive content thus safeguarding their community's access to that content. The LOCKSS model enforces the publisher's access control systems and, for many publishers, does no harm to their business models.

Accuracy and completeness of LOCKSS caches are assured through a peer-to-peer (library-to-library) polling and reputation system, which is both robust and secure. LOCKSS replicas cooperate to detect and repair preservation failures. LOCKSS is designed to run on inexpensive hardware and to require almost no technical administration. The software has been under development since 1999 and is distributed as open source.

The LOCKSS Program has as its mission to build tools and to provide support to:

**Libraries** so they can easily and affordably create, preserve, and archive local electronic collections

- Own rather than lease electronic information
- Retain traditional custodial role of scholarly information
- Provide continuing and perpetual access to their local community

**Publishers** so they can easily and affordably provide content to the libraries for preservation and archiving

- With minimal risk to their business model or to their publishing platforms ensure perpetual access to their materials
- Fulfill librarians' requirements that publishers guarantee both continuing (day to day) and perpetual (long-term) access to content sold.

## 3 Why should a library be interested?

There are two issues that the LOCKSS approach addresses:-

- continuing access to digital resources
- preservation of digital resources

### 3.1 Continuing access

Currently, libraries can generally only lease access to e-journal content from publishers. The material is held (in one place) on the publishers' server. There are two problems here:-

1. There is a single point of failure either technically (which might lead to short-term loss of access) or if the publisher decides that it can no longer offer the title in question for economic or other reasons (which might lead to long-term loss of access).
2. If a library decides to discontinue subscription, whilst the publisher may be contractually obliged to provide continuing access to the material published during the period that the library had a subscription, publishers often grant perpetual access to their journals by providing a CD-ROM version, rather than continuing access to their web delivered version. This is not very convenient for either libraries or their patrons.

In addition there are a large number of e-journals published not by commercial publishers, but by one or more academic 'enthusiasts' in a discipline, which are at risk if those involved move institution, simply lose interest or haven't the time to devote to it anymore. The LOCKSS project team about a dozen US institutions are particularly worried about this type of material and are working to preserve high risk, 'born digital' humanities electronic journals (<http://lockss.stanford.edu/humanities.htm>). They feel that it is important to have key UK libraries join this group. The number of 'very important' humanities titles are staggering and there are probably a considerable number that are more important to the UK and European scholarly community than to those in the US and need focussed attention from UK and European libraries.

### **3.2 Digital preservation**

LOCKSS does not provide a comprehensive solution to digital preservation, but it does address the problem of technical or other failures corrupting the material. To do this it requires that there are at least 4-6 copies of the material on different caches and so it is only suitable for 'published' and not 'single unique copy' type materials.

## **4 How does it work?**

LOCKSS creates low-cost, persistent digital "caches" of e-journal content at institutions that:

1. subscribe to that content; and
2. actively choose to preserve it.

LOCKSS uses the caching technology of the web to collect pages of journals as they are published, allowing libraries to take physical custody of selected electronic titles they purchase. Unlike normal caches, however, pages in these caches are never flushed.

It works by getting libraries to install a piece of software on a low-specification PC with a large hard disk, turning it into a cache for web pages. The program then collects the content of various journals that the library in question has subscribed to. If the system detects that one of its copies is damaged or missing, it asks the original publisher, or the cache of another library, to send it a fresh copy.

But who decides which copy is the correct one? For this purpose, LOCKSS employs another simple idea: that of the opinion poll. The caches vote at intervals on material by comparing digests (unique signatures computed from a given file). The caches on the losing side repair their copies in collaboration with those on the winning side.

Such polling makes it difficult to trick the system. To change an article deliberately, attackers would have to subvert the majority of the caches and do so for a long time. The architecture also makes it unlikely that the system will become obsolete. If a cache runs out of storage space, for instance, libraries simply buy and boot up a new PC with a larger hard disk, which will then automatically acquire the appropriate material from the publisher and other caches. Also as the LOCKSS software is open source, it is free to evolve over the years as technology changes.

## **5 What resources are required?**

The cost of entry to LOCKSS is very low and there are two elements:

- Of the order of £700 for PC hardware, if a new one has to be purchased, but in Glasgow we started with a machine that was no longer suitable as a desktop machine and added a larger hard disk for about £50.
- Someone to follow a fairly straight-forward set of instructions to set up the cache and carry out cache maintenance (for which little technical expertise is required) and which takes a few hours per month (often less).

## **6 The Issues**

### **6.1 Critical Mass of UK Participants**

The LOCKSS beta testing has suggested that to be viable, a piece of content should be held on a minimum of about six LOCKSS caches. At the moment there are only of this order UK participants. If any specifically UK material is to be cached using LOCKSS, either all participants would have to subscribe to and hold it (irrespective of collection policies) or the number of UK LOCKSS caches needs to be substantially increased.

CURL should have a role here to encourage libraries to become participants, particularly if they have significant e-journal holdings.

### **6.2 Plug-ins for UK Journals**

In order for LOCKSS to work with a particular publishers web site, aside from the publishers permission to cache the material, a Plug-in must be configured to enable the correct material to be captured and avoid the special traps that publishers put in to stop ordinary folk from hoovering up their material.

Whilst this is not a hugely difficult job, it requires skills that currently are only held by the LOCKSS team at Stanford. If UK specific material is to be brought into LOCKSS caching, skills in plug-in writing need to be developed in the UK and plug-ins need to be developed for the Journals that the UK community considers important.

Without this, we are always going to be following a US agenda (simply because there are a lot more LOCKSS players in the US).

This looks like a job for JISC funding and the role for CURL might be to lobby JISC for this and, as a community, identify priorities for plug-in development.

### **6.3 Collection Development**

Although it is theoretically possible to cache all the e-journals that libraries take via the mechanisms that LOCKSS offers, there are obvious limitations. What happens with LOCKSS, as far as material that is important to UK libraries is concerned, depends on the e-collection strategy of the UK library community.

Specifically the issues are:-

1. Even with the current number of e-journals that a library subscribes to, the disc space requirements are potentially huge and choices will have to be made. All libraries do not have to cache all titles (subject to a minimum critical mass). Can CURL negotiate with publishers so that cached material can be shared between consortia of libraries that all take a particular title?
2. Plug-ins will never be developed at a rate commensurate with the increase in number of e-journals available. Some sort of community prioritising will have to take place - what role can/should CURL have in this?
3. It is probably the journals that are not published by large commercial publishers that are most at risk and therefore needful of LOCKSS protection, such as journals in the humanities and social sciences that are published by academics on a shoestring. What role can/should CURL have in the identification and preservation of these?
4. Some publishers may be reluctant to allow their material to be cached using LOCKSS or to provide some of the information that would be needed to write plug-ins. What role can/should CURL have in bringing pressure to bear here?
5. Some publishers may be willing to be involved in the writing of plug-ins for their journals if they perceive LOCKSS as a thoroughly good thing. What role can/should CURL have in bringing publishers on-side in this way?

## 7 The LOCKSS Alliance

Thus far, LOCKSS development has been funded by a variety of organisation. For the future, funding has to come from the library community. For this purpose, the LOCKSS Alliance has been set up, as a subscription organisation for libraries that care about the long-term survival of their digital assets. The CURL libraries could join the LOCKSS Alliance as a group and get a vastly discounted membership rate. For details about the Alliance in general, see <http://lockss.stanford.edu/alliancelibraries.htm>. In particular, "LOCKSS is in a nascent stage of organisational development" and by taking this step, CURL would be in a good position to influence its future direction.

The Alliance needs the community's support to carry LOCKSS forward. Without support this effort will go away. While no-one can promise LOCKSS will be a success, it shows great potential.

There are few actions libraries can take now to preserve their social role as memory organisations; to preserve digital scholarly information for future generations. The risks of going forward now with LOCKSS are few. The risks of doing nothing are extremely high.

## 8 Technical Details

Technical overview LOCKSS creates low-cost, persistent digital "caches" of authoritative versions of http-delivered content. All file formats delivered through HTTP are included (html, jpg, gif, wav, pdf, etc.). The LOCKSS software enables institutions to build local collections; to locally collect, store, preserve, and archive authorized content, thus safeguarding their community's access to that content.

LOCKSS caches collect the content as it is published. For paid e-journals, a library must participate at point of subscription or renewal to benefit from the system. Once a library activates an e-journal titles preservation in the LOCKSS system, that titles newly published content is continuously collected and preserved. The process is halted only when a library turns off content collection or if a publisher turns off permission. The LOCKSS model enforces the publishers access control systems and, for commercial publishers, does no harm to their business models. The current version of LOCKSS software is restricted to electronic journals and to date has been applied primarily to paid-subscription-based scientific journals.

Accuracy and completeness of LOCKSS caches are assured through a peer-to-peer polling and reputation system (operated through LOCKSS' communication protocol), which is both robust and secure. LOCKSS replicas cooperate to detect and repair preservation failures. LOCKSS is designed to run on inexpensive hardware and to require almost no technical administration. The software has been under development since 1999 and is distributed as open source through <http://www.sourceforge.net/>.

LOCKSS has flexible, extensible, three-layer architecture.

1. Infrastructure platform: The platform layer is the distribution and execution environment. For security and maintainability reasons it is based on a CD-R that boots the OpenBSD operating system.
2. Daemon: This layer of software runs the LOCKSS system and provides: damage detection and repair, a web proxy, a web crawler, and a web administrative interface.
3. Journal plug-ins: This layer of software adapts the generic LOCKSS system to specific journals of interest. A set of re-usable components for plug-ins and a testing framework has been implemented to assist the open source community in developing plug-ins for journals.

## 9 Acknowledgements

Some of the material used in this document to describe the LOCKSS system has been drawn from materials written by the LOCKSS team at Stanford. In particular the author would like to thank Vicky Reich and David Rosenthal.