Flexible Bit Preservation on a National Basis

Bolette A. Jurik, The State and University Library of Denmark, Aarhus; Anders B. Nielsen, The National Archives of Denmark, Copenhagen, Eld M. O. Zierau, The Royal Library of Demark, Copenhagen

Abstract

In this paper we present the results from The Danish National Bit Repository project. The project aim was establishment of a system that can offer flexible and sustainable bit preservation solutions to Danish cultural heritage institutions. Here the bit preservation solutions must include support of bit safety as well as other requirements like e.g. confidentiality and availability.

The Danish National Bit Repository is motivated by the need to investigate and handle bit preservation for digital cultural heritage. Digital preservation relies on the integrity of the bits which digital material consists of, and it is with this focus that the project was initiated.

This paper summarizes the requirements for a general system to offer bit preservation to cultural heritage institutions. On this basis the paper describes the resulting flexible system which can support such requirements. The paper will explain principles and design, as well as how both design and implementations can be used by any institution or company with requirements to bit preservation.

Introduction

Bit preservation is defined as the required activities to ensure that the bit-streams remain intact and readable [7]. Bit preservation is crucial for all digital preservation. Any bit damage of digital material can hinder correct interpretation by software and ultimately result in total loss of the digital material [15]. Important contributors of bit preservation are: number of redundant replicas of data items, their independence, and recurrent integrity checks.

Attention to bit preservation has increased in the last decade. In many places bit preservation has been seen as a solved issue, and storage solutions have been the basis for bit preservation. Recent initiatives have focused on the risks that are involved in bit preservation, and pointed at limitations in current practices using storage solutions [3]. Examples of bit repositories that take a risk-based approach to ensure bit safety are: LOCKSS (based on on-line peers) [17], DuraCloud (based on storage offered by the cloud) [4], and The Bit Repository [13] described in this paper.

There is an increasing awareness that independence between data replicas is crucial for data safety, also on the organizational level [6]. This fact along with wishes for shared solutions in order to keep costs at a minimum is the background for defining bit preservation solutions in a community as is the case for Private LOCKSS Networks (PLN) [12][16][17] and The Bit Repository described in this paper [5].

This paper will present the results from The Danish National Bit Repository project concerned with establishment of system that can offer flexible bit preservation solutions to Danish cultural heritage institutions. The project is founded by The Danish National Archives, The Danish Royal Library and The Danish State & University Library. The results of the project are threefold:

- A *general design* of the design which will be denoted as The Bit Repository
- A *reference implementation* of The Bit Repository
- An *instantiation* of the implementation, named the Danish Instantiation

The *general design* is aimed at creating a sustainable bit repository, independent of storage platforms, and providing a wide range of bit preservation solutions for digital collections. This includes inexpensive solutions with low bit safety and expensive solutions with high bit safety. For each collection of data the number of data replicas can be defined, as well as their independence and cross-storage integrity check frequency. The preservation solutions can furthermore be defined to meet other requirements like high confidentiality and high availability of data. We describe in detail the *general design* of the bit repository including the clients and service layer, the coordination layer and the pillar layer. Further, we describe the *reference implementation* and finally the Danish *Instantiation* which are currently under development.

This paper is called “Flexible Bit Preservation on a National Basis”, since it describes the findings and developments made in connection with development of the Danish National Bit Repository. The Bit Repository architecture and design however generally facilitates differentiated solutions to bit preservation, and is sustainable with respect to the changing storage technologies. The architecture and design and implementation are not restricted to exist on a *national* basis, but the described instantiation is a Danish national instantiation.

Before the actual description of the threefold result, this paper will describe the background and motivation for the results, by describing the feasibility study for the Danish National Bit Repository, which revealed the need for this flexible solution.

Feasibility Study

In 2008-2009, the Danish Ministry of Culture initiated a feasibility study which was to investigate the feasibility of a joint bit repository to offer bit preservation solutions to Danish digital cultural heritage institutions. The project participants were The Danish National Archives, The Royal Library, and The State & University Library, who are also stakeholders of the bit repository now being developed.

The aim of the bit repository was to find a common system that would provide secure large scale means of ingesting, storing, auditing, and accessing bits. Basic assumption for this study was that is ensured by having several replicas of data and basically depends on:

- Number of replicas of data
- Independence between replicas of data
- Frequency of integrity checks [3]

Where any bit preservation solution can be described as illustrated in figure 1 (even back-ups). The solutions all include

 ©2012 Society for Imaging Science and Technology

replication of the data and some sort of coordination for access and ingest and possibly among replica units [7]. The units on which data is placed is here called pillars and consist of a specific media in a technical and organizational environment.

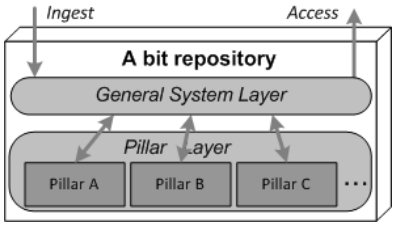

Figure 1. General view of a bit repository

The difference between different bit preservation solutions is the way that the replicas are placed on the pillars, and how the general system layer is implemented. One example is LOCKSS, where LOCKSS caches in a Private LOCKSS Network would be the pillars, and the general system layer will cover the communication protocol and network that enables LOCKSS caches to communicate.

Quite early in the feasibility study it became apparent that a (shared) bit repository is a (shared) repository which must include functions from all OAIS functional entities, and thus should be regarded as an OAIS repository [10]. As mentioned, this (shared) bit repository only includes bit preservation, while participating institutions retain responsibilities related to the logic of the content in their institution repository. Thus the (shared) bit repository becomes an OAIS bit repository within an Institution repository which may also be regarded as an OAIS repository [5].

Another important outcome from the feasibility study was that the bit repository had to facilitate solutions that covered much more than just bit preservation. An analysis of the stakeholders' requirements showed that additional requirements to bit safety differed for different data collections according to the mission of the owner and purpose of the data collection [5]. For instance some data collections had special requirements in order to fulfill national and international legislation such as copyright acts, archive acts and personal data protection acts. Other examples were requirements for specific types of media, e.g. offline Write Once such as DVD in order to better fulfill confidentiality requirements or requirements of low costs. Yet others required fast online access and possibilities of mass processing (processing masses of data, e.g. characterization of a full archive) [2]. Furthermore, the analysis showed that requirements for bit safety varied for different data collections in accordance with their perceived value. For instance digitally born e-books need a higher bit safety than digital copies of printed books, because the latter may be re-digitized, if the digital copies are damaged or lost. Therefore a lower bit safety for the collection of digitized books can be acceptable. The mentioned requirements to the bit repository can be classified as information security requirements such as availability, confidentiality and integrity (as specified in the ISO27000 series [11]), as well as requirements related to economy. Here integrity of bits is covered by bit safety.

The reason why the feasibility study pointed at new development of a bit repository is that none of the known existing systems could provide the basis for meeting all the above mentioned additional requirements to bit safety. For instance LOCKSS is based on on-line caches, which excludes the possibility of a pillar based on off-line DVDs which is required for the Danish National Archives in order to meet confidentiality requirements. Although the DuraCloud platform can be used for other pillars than cloud based pillars, it is too weak on authorization and authentication to serve confidentiality issues, and therefore not a suitable choice either. On the basis of the conclusions from the feasibility study, the development of the so-called Bit Repository began in 2010.

The Bit Repository

The basis for the Danish National Bit Repository project is a general architecture which is designed to support differentiated bit preservation requirements which includes bit safety (integrity) as well as confidentiality and availability.

It is important to note the general design principles are generic and implementation independent, i.e. independent on the technical platforms which an implementation can be based upon. This general architecture and design principles will be described in the section "General Architecture of the Bit Repository".

The current development is of course building on specific platforms. However, as will be described, the development work aims at a flexible bit repository which can easily be extended and changed along with the technical evolution. This is described in section "Reference Architecture and Implementation".

Finally, there is the actual instantiation of the implementation which will be used as the Danish National Bit Repository (from hereon called the Danish Instantiation). However, much more than technology is needed in order to operate an actual bit repository. Therefore the section "The Danish Instantiation of the Implementation" will include discussion of the organizational aspects as well as the concrete instantiation.

General Architecture of the Bit Repository

In order to allow for different services on different pillars, the architecture is independent of the types of pillars, but has the possibility to exploit the special services that a specific pillar can offer. Taking this into account along with the basic dependencies of bit safety, the architecture ended up as illustrated in figure 2.

In this architecture the general layer is split into a coordination layer and a client & service layer. The coordination layer only has the function to coordinate exchange of information between clients, services and pillars. The client & service layer contains client components that are started upon client request, and service components that are continuously active. The client components handle on demand service functions such as ingest and access. The service components cover monitoring functions and cross operations as e.g. cross pillar integrity check and run continually.

Combinations of different pillars with special characteristics enable a variety of service within this architecture. The level of bit safety will rely on the choice of pillars (and thus pillar characteristics) for replicas. Independence between replicas can for instance be made regarding the media type; choosing one replica to be placed on a pillar with an optical media type (e.g. DVD) and another replica on a pillar with magnetic media (e.g. Server).

There can also be made independence with respect to e.g. organization or geography.

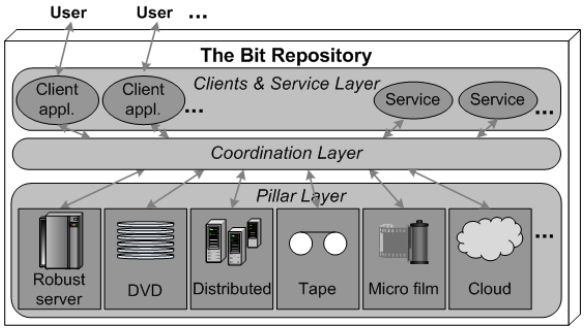

Figure 2. Architecture of The Bit Repository with examples of pillars

An additional advantage of this architecture is that it is independent of the types of platforms and the number of platforms. This enables shifting and/or adding pillars as new storage technologies appears.

Possible services regarding availability of data will rely on access characteristics for a pillar. For instance, there are better possibilities for fast access to replicas if they are placed on a pillar with distributed hardware architecture than if placed on DVDs.

However, the bit repository is not just defined by pillar characteristics. It must rely on design principles that ensure that all information security aspects can be supported. The following sections give a more detailed description of the general design of each layer in order to obtain this goal.

The Pillar Layer – general design

The design principles for pillars in the pillar layer are that they must be as independent and self-contained as possible. This independence must exist on several levels.

Firstly, each pillar must store authoritative and complete knowledge of the data placed inside it. No pillar must depend on e.g. a master index placed somewhere else in the bit repository, and thus rely on external information that the pillar itself does not have. It is therefore also required that any pillar at any stage can reply on requests of its contents for a data collection as well as checksums for the individual data items. The operations that the pillars must support are:

- *Put file*: ingest data
- *Get file*: access data
- *Delete file*: delete data (if allowed)
- *Replace file*: replace data (usually as part of repair)
- *Get Checksum*: of specified data item
- *Get File IDs*: for a specified set of data items

In addition there are operations for audit trails, status and alarms. Also single processing and mass processing operation may be offered by some pillars.

Secondly, each pillar must not depend on knowledge of other pillars in the bit repository. Each pillar will require knowledge of how it will communicate and what communication it must react upon, but it must not depend on specific knowledge of the coordination layer or a specific client or service instance.

Thirdly, a pillar must be independent of how clients, services and coordination layer work.

Fourthly, the software developed for each pillar should be developed independently, in order to avoid that the same software error can harm several pillars.

Furthermore, it is required that the pillars are internal for the bit repository in the sense that they cannot function in other contexts which can manipulate data in the pillars. This is required in order to keep the Bit Repository in control of changes to bits, which must only occur as part of restoring bit integrity.

All pillars must rely only on asynchronous communication, and they must be robust in the sense that absence of response or wrong responses must not result in major delay or crashes.

The Coordination Layer – general design

The main design principle for the coordination layer is that its only function is to coordinate exchange of information between clients, services and pillars. The layer must be as thin as possible in the sense that it does not depend on persistent information, and that it does not rely on special features of the components, e.g. the platform type of pillars.

The type of information that can be exchanged is either messages or data items from a data collection.

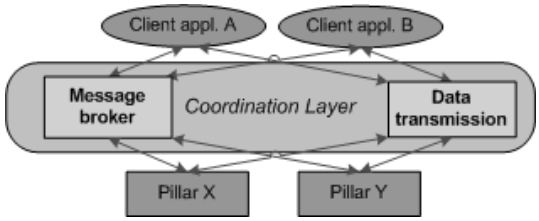

Figure 3. Information passed through coordination layer

Figure 3 illustrates how this can be divided into message exchange (via message brokering) and data exchange (via data transmission).

The messages are the basis for the actual coordination, and it is the protocol for these messages that is the backbone of the system. The protocol supports communication with the bit repository and within it. Any storage facility that can hold replicas of data can be integrated into a bit repository using the protocol. The protocol supports communication without assumptions of timing aspects in the communication, i.e. on-line disc servers as well as off-line media like DVDs and microfilms can be used as platforms in the bit repository. This allows for changes of the physical platforms based only on the assumptions that data can be ingested and accessed.

The Clients & Service Layer – general design

The clients handling user related service functions can only have knowledge of the minimal elements from settings defining the collection of data that that they services. The settings define the context in which a data collection must be treated by the bit repository (e.g. which replicas on which pillars and frequency of cross pillar integrity checks). Clients must not depend on specific knowledge of the pillars except from being able to identify itself as a client with a legal request to a pillar on which the data is placed.

©2012 Society for Imaging Science and Technology

The reason is that this would make it harder to make flexible solutions where conditions for replicas can change, e.g. by increasing or decreasing number of replicas placed on different pillars.

Furthermore, the operation of the bit repository must be independent of the clients, i.e. an absence of clients must only have impact on the service they deliver for the specific data collection they belong to (e.g. access). The operations that can be offered on this level is the same as the operations of the pillars, i.e. Put File, Get File, Get Checksum etc.

The client and service also offers services which cover monitoring functions and cross operations. The services to be covered are:

- *Audit trails*: continuous collection and securing of audit trails, and making these available
- *Monitoring*: continuous monitoring of coordination layer
- *Alarms*: listens for any alarms and take action
- *Integrity*: regularly collecting checksums from the various pillars, and ensuring that the data are consistent across pillars

Checking integrity of data in the Bit Repository is one of the key elements of ensuring bit preservation. In case of inconsistency, integrity functionality also ensures processes for raising alarms and restoring consistency.

Services will to some degree need a minimum of information on the components in the system, e.g. which pillars, clients, services and coordination layer instances are expected to be present. Still, the principle is that the services must rely on as little information as possible in order to fulfill their function.

In order to facilitate the asynchronous communication for these operations, the protocol has a specific pattern for messages used in each operation. As illustrated in figure 4 for the 'Get File' operation, there is a set of messages representing the operation apart from the actual data transmission. This set consists of two parts; the first two messages are used to identify the pillars to request data from are identified, and the remaining are used to perform the actual 'get operation'. Note that the 'operation part' allows for many responses before completion. This is to facilitate that an operation on off-line media like DVDs can have to go through several steps before the final completion of operation.

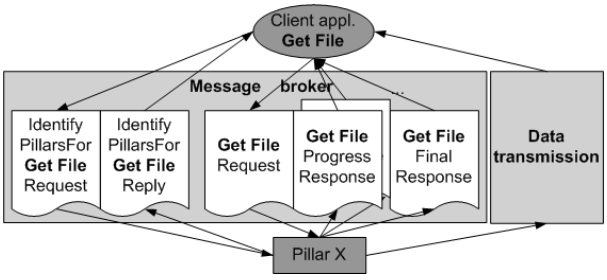

Figure 4. Messages for the Get File operation

The messages are designed to support a principle of giving the pillar the initiative of all data transmissions needed for operations. For instance, it is a pillar that upload a file to a client in connection of a client get request, and it is the relevant pillars that download a file in connection with a put operation.

There are also other types of messages for audit trails, status and alarm services, where some of them differ from this pattern, e.g. for alarms which must be sent without identification of recipient.

Reference Architecture and Implementation

Various choices have been made in order to implement the general design. Except from the design of the protocol, it is common for all the implementation choices that they at a later stage can be exchanged, e.g. the system is designed to be as independent of the choice of message broker as possible. The following sections describe some of the important choices made for each layer. The bit repository components for general implementation are available as open source [13].

The Pillar Layer – reference implementation

For the general design there has been made a fully functional reference implementation of a pillar which works on a normal file system. A reference implementation is an implementation that can be used as an actual implementation (in this case for a pillar), or the reference implementation can be used as documentation for how to make your own implementation. The reference pillar is coded in Java.

In a concrete instantiation of the general bit repository each pillar should be developed separately, in order to ensure as much independence between data replicas as possible. It will therefore not be safe to use the reference implementation for all pillars. Also, it is not likely that the reference implementation can be used for all pillars, since the implementation will depend on characteristics of the pillar like media type, possibility for mass processing etc.

The Coordination Layer – reference implementation

The actual implementation of the coordination layer consists of an XSD definition of XML-based messages, a choice of message bus and a data transmission protocol and principles of use.

The definition of the protocol messages is coded such that it enables a change of a message broker system without the need for changing the messages. The protocol can maintain new versions with a versioning system that respects that not all components of the system can be updated at the same time. This allows new functions to be added to the protocol. The XSD's along with example XML messages can be found on the Bit Repository web site [13]. A simplified list of the GetFileRequest message elements is given below.

- ***GetFileRequest*** - Get File Request msg
- ***Core*** - Mandatory part of a message
  - ***CorrelationID*** - Unique communication id.
  - ***BRCollectionID*** - Unique BR collection id
  - ***To*** - Name of 'To'queue/topic
  - ***ReplyTo*** - Name of 'from' queue/topic
- ***FileID*** - Unique id. for data
- ***FilePart*** - Spec. of part of data
- ***FileAddr*** - Delivery addr. (eg. URL)
- *VersionAttributes*

The choice of message broker for the actual implementation is ActiveMQ [1]. The actual implementation of the data transmission is https [9]. Both ActiveMQ and https uses TLS [14]

to provide communication security. An overview of how the different layers are implemented in the general implementation is given in figure 5.

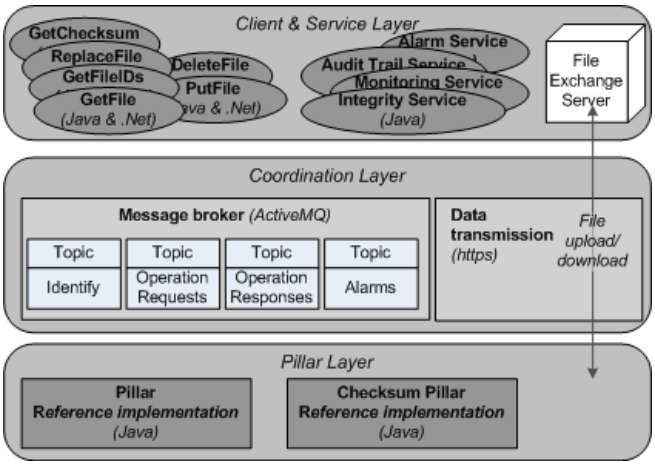

Figure 5. General implementation of layers

This figure also shows the message broker topics chosen for the general implementation and the file server used for data transmission. Multiple instances of the coordination layer can be parallelized in order to make it scalable and in order to avoid a single-point-of-failure.

The Client & Service Layer – reference implementation

All clients and services will be available as reference clients and services written in Java. The clients and services are illustrated in figure 5. The reference clients and services will have access to a file exchange server in cases where this is needed in order to perform data transmission to/from pillars.

There is still analysis to be done before the single processing and mass processing clients can be written. Mass processing is very platform dependent [2] which only allows for a very general interface for a protocol. Furthermore this interface must be supported by other procedures in order to ensure that mass processing cannot harm the data in the bit repository.

The Danish Instantiation of the Implementation

The Danish instantiation of the Bit Repository will use the reference implementation, and will go into pilot operation in 2012.

Technical issues

As a starting point the Danish Instantiation will have four of the pillars depicted in figure 2: A disk pillar and a tape pillar at The Danish State & University Library, a distributed server park pillar at The Danish Royal Library, and a DVD pillar at The Danish National Archives. All the pillar software will be developed individually by the institutions that have the pillar.

The coordination layer will be based on several instances of coordinated ActiveMQ message broker instances and there will be more https servers for the data transmission.

As a starting point it is the Java reference services that will run as services. It is up to the users of the bit repository whether they will use reference clients, or whether they will develop their own clients. If they develop their own, these will of course have to respect protocol policies and can only access information via the protocol.

Organizational issues

There are also a number of organizational issues that need to be taken care of around the Danish Instantiation. For instance, agreements on operation tasks related to coordination layer are necessary in order to ensure that the clients can communicate with the pillars.

Furthermore, the arrangement of running services needs an agreement. It is possible that any of the bit repository users can perform services, e.g. integrity and audit trail services are related to specific collections. Also alarm and monitoring services can be collection dependent, but in order to ensure a minimum of security, there must be a minimum of service for monitoring that system components are alive and that general alarms are send if the security of the system is endangered.

Active bit preservation is essential for true bit preservation in the Bit Repository. Thus, having an agreement for an integrity service should be considered essential. Such a service will need agreements of intervals and actions and hosting.

In order to ensure a sustainable system, there are tasks of maintenance with respect to the coordination layer, as well as the underlying protocol. These will need some overall organizational coordination as well.

Last, but not least comes the question of how to ensure that media migrations in the individual pillars cannot harm bit safety, e.g. by having simultaneous media migrations.

Discussion

The main purpose of the Danish National Bit Repository project is sustainable bit preservation. The sustainability is mainly based on using mutually independent pillars, each having its own storage technology and its own organization.

Sustainability is also dependent on economic feasibility. Specializing on a storage technology in each organization brings economies to scale, especially for small organizations. Nevertheless, the project does so far not aid in estimating the cost or level of bit preservation safety it can provide.

There are many considerations on which pillars to include in order to cover the solutions needed for a user, and there is still a challenge to choose the right solution for specific material. “Evaluation of Bit Preservation Strategies” [8] provides a method that can help in such decisions, but further research and experience from a system in production is needed in order to make this method fully operational. The evaluation method aims to cover the different types of information security requirements, and thus cover contradicting requirements. For instance bit safety and confidentiality can be contradicting regarding the number of distributed replicas of data; increasing the number can increase bit safety, but also makes the data more vulnerable for leaks. Also availability and opportunity to do mass processing are considerations that can be part of the evaluation of an optimal bit repository solution [7].

In order to make an evaluation, there is also a question to which degree it is possible to find measures as basis for an evaluation. For instance probabilities for single bit errors and even total media failure for specific media are not fully known. This

©2012 Society for Imaging Science and Technology

also counts for use of e.g. RAID technologies combined with types of optical disc based media. Such measures can be crucial for evaluation of frequency and coverage of integrity checks on individual pillars. More research and experience is needed in order to provide such measure, although we acknowledge that the rapid change of media technology can make it hard to keep pace with the information needed.

A noteworthy factor that can affect sustainability and bit safety is that due to the very limited amount of producers of storage equipment and especially storage media, single point of failures can still occur. The recent flooding in Thailand causing huge delays in the supply of hard disc drives is such an example.

The independence of the organizations operating a pillar is another important part of the sustainability. This independence is important to avoid the single point of failure due to a monolithic organizational breakdown. Nevertheless, the three organizations operating the pillars in the Danish Instantiation are all funded by the Danish Ministry of Culture, and therefore all vulnerable in case of a general decrease in funds. Due to the national origin of the material this will often be the case, especially for small nations.

Conclusion

The Danish National Bit Repository project is to provide large scale, flexible bit preservation solutions for cultural heritage institutions. The flexibility of the solution also makes it usable for other institutions or companies with requirements to bit preservation. An actual Bit Repository instantiation ensures bit safety by using multiple, independent pillars and frequent cross-pillar integrity checking. The Bit Repository solution exists within a full repository where aspects of functional preservation are left to the external surrounding repositories..

The architecture and reference implementations for The Bit Repository are designed to be flexible and sustainable. It is flexible as it is possible to take into account that different materials need different degrees of bit safety, confidentiality, availability and costs, and facilitates tailored solutions. This architecture and reference implementation is sustainable, as it can be used in different instantiations, extended to different use scenarios and updated along with technological changes.

The Danish instantiation of The Bit Repository is the joint bit preservation solution offered by The Danish National Archives, The Danish Royal Library and The Danish State & University Library, and it will cover bit preservation for the diverse Danish digital cultural heritage in Danish cultural institutions.

References

[1] ActiveMQ, available at http://activemq.apache.org/ (retrieved March 2012)

[2] B. A. Jurik, E. M. O. Zierau, Different Mass Processing Services in a Bit Repository, Proceedings of the Fourth Workshop on Very Large Digital Libraries, Berlin, Germany, pg. 11-18 (2011)

[3] D. S. H. Rosenthal, Bit Preservation a Solved Problem?, The International Journal of Digital Curation, vol. 5, no. 1 (2010)

[4] DuraCloud, available at http://www.duracloud.org/, (retrieved March 2012)

[5] E. Zierau, U.B. Kejser, Cross Institutional Cooperation on a Shared Bit Repository, Journal of the World Digital Libraries, vol. 3, issue 1, pp. 11-21, Publisher: TERI Press, New Delhi (2010)

[6] D. S. H. Rosenthal, T. Robertson, T. Lipkis, V. Reich, S. Morabito, Requirements for Digital Preservation Systems, A Bottom-Up Approach, D-Lib Magazine, vol. 11, no. 11 (2005)

[7] E. M. O. Zierau, A Holistic Approach to Bit Preservation, Doctoral Dissertation, Copenhagen University, available at http://www.diku.dk/research/phd-studiet/phd/thesis_20111215.pdf, retrieved March 2012 (2011)

[8] E. Zierau, U. B. Kejser, H. Kulovits, Evaluation of Bit Preservation Strategies, Proceedings of the 7th International Conference on Preservation of Digital Objects, Vienna, Austria, pg. 161-169 (2010)

[9] HTTP Over TLS, available at http://tools.ietf.org/html/rfc2818, (retrieved March 2012)

[10] ISO 14721:2003, Space data and information transfer systems – Open archival information system – Reference model, available via: http://www.iso.org/iso/iso_catalogue.htm (retrieved December 2009) (2003).

[11] ISO/IEC. 27001:2005 Information technology – Security techniques Information security management systems – Requirements, available via http://www.iso.org/iso/iso_catalogue.htm (retrieved December 2009) (2005-2010)

[12] Private LOCKSS Networks description, available at http://lockss.stanford.edu/lockss/Private_LOCKSS_Networks, (retrieved March 2012)

[13] The Bit Repository, available at https://sbforge.org/display/BITMAG/The+Bit+Repository+project (retrieved March 2012)

[14] The TLS Protocol, version 1.0, available at http://www.ietf.org/rfc/rfc2246.txt (retrieved March 2012)

[15] V. Heydegger: Just One Bit in a Million: On the Effects of Data Corruption in Files, Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries, pp. 315-326, Agosti, M., Borbinha, J. , Kapidakis, S., Papatheodorou, C., Tsakonas, G. (eds.) LNCS, vol. 5714, Springer, Heidelberg (2009)

[16] V. Reich, D. S. H. Rosenthal, Distributed Digital Preservation: Private LOCKSS Networks as Business, Social, and Technical Frameworks, Library Trends, vol. 57, no. 3, pg. 461-475 (2009)

[17] V. Reich, , D. S. H. Rosenthal, LOCKSS: A Permanent Web Publishing and Access System, D-Lib Magazine, vol. 7, no. 6 (2001)

Author Biography

Eld Zierau received her M.Sc.in computer science from University of Aarhus (1989) and her PhD in digital preservation from University of Copenhagen (2011). Since 1989 she has worked in various industries with similar challenges to long term preservation today. She joined the Royal Library in May 2007, with involvement in the PLANETS project, the web-archiving project, and in enhancement of a digital object management system, and the Danish National Bit Repository project.

Bolette Jurik. received her M.Sc.in computer science from University of Aarhus (2002) and her PhD in computer science from BRICS Int'l Ph.D School, University of Aarhus (2005). Bolette has worked at the State and University Library since 2005 and since 2008 with digital preservation. She has been involved in the ECDL2008 Organising Committee, TPDL Steering Committee, PLANETS EU project, Danish National Bit Repository and is currently working within the SCAPE EU project.

Anders Bo Nielsen holds a master in economics from the University of Copenhagen (economic history and IT as specialty) from 1997 and a master in IT (software development as specialty) from the IT University of Copenhagen from 2007. He is a principal consultant at the Department of Appraisal and Transfer at the Danish National Archives, where he has been employed since 1997.