PART I – PROJECT DETAILS AND ACCESS

Project Number:	**100939**
Project Title:	**RASSC (Retention and Access Services in Supply Chains)**
Deliverable Type: (PU/PP/RE/CO)*	**PU**

Deliverable Number:	**D2.2**
Contractual Date of Delivery:	**PM10**
Actual Date of Delivery:	**PM11**
Title of deliverable:	**Cost models and cost modelling tools**
Work-Package:	**WP2**
Nature of the Deliverable: (R, P, D, O)**	**R**
Version:	**V1.0**

Abstract:

This report describes a cost model and associated tools for estimating the long-term costs of operating a trusted digital repository service for aerospace data. The report reviews work already done by the digital preservation community on techniques for cost modelling, analyses the requirements for cost modelling for long-term retention and re-use of aerospace data (design, manufacturing and IVHM), and then describes both a simple empirical cost model and more sophisticated cost simulation tool and how they can be applied.

Author:	Matthew Addis, mja@it-innovation.soton.ac.uk
Responsible Party:	IT Innovation Centre
Address:	Gamma House, Enterprise Road, Southampton, SO16 7NS

*Type: PU - public; PP- Restricted to program participants; RE – Restricted to group specified by consortium; CO – Confidential, only for members of the consortium
**Nature: R = Report; P = Prototype; D = Demonstrator; O = Other

Contents

1 Introduction and scope of cost modelling in RASSC

RASSC (Retention and Access Services in Supply Chains)[1] is a UK RTD project supported by the Technology Strategy Board. Project partners are BAE Systems, Eurostep, Ovation Data Services and the University of Southampton IT Innovation Centre.

RASSC has developed models, tools and demonstrations of long-term data retention and access in aerospace supply chains. This market is yet to embrace the huge advantages that shared services can bring to long-term data retention, shared access and commercialisation of retained assets. Furthermore, a move from purely in-house working to the use of outsourced retention and access services provides major opportunities for dedicated service providers to deliver new storage and asset management services into supply chains.

Scenarios investigated in the project include:

- Sharing of Integrated Vehicle Health Monitoring (IVHM) data, for example to enable better maintenance schedules and optimisation of spares inventory as part of MRO, and more widely to help prime contractors transition to the delivery of capabilities instead of products.

- Long-term Data Retention (LTDR) repositories to support design reuse, certification based on electronic data, risk reduction of product retirement, and lower cost of compliance and litigation readiness.

- Information management as a sustained third-party service in order to reduce costs, gain access to expertise not available in-house, free-up internal resources, and transfer of responsibility.

All three rely on the ability to establish and operate a Trusted Digital Repository (TDR) which can sustain and provide access to digital assets over their entire lifetime (e.g. the service life of an aircraft). There are many challenges in creating and operating a TDR, which include:

- Obsolescence at all levels (hardware, software, people) means continuity of information and services over the life of an aircraft will requires a programme of managed migrations and validations.

- Maintaining usability over long-periods of time requires structured capture of a wide range of contextual information at the initial point of data retention to ensure the data, remains understandable. This includes the need for standard formats and submission agreements to reduce life cycle costs.

- Specialist skills and resources are needed to plan, build, operate and manage a TDR. A shared, trusted repository service is a way of provide these specialist skills across a supply chain.

The cost model described in this report is for the long-term operation of such a digital repository service for aerospace data.

[1] www.rassc.org

The scope of the cost model is the repository functionality and repository infrastructure services as shown below. Repository construction is not included in the cost model.

Terminology used in this report follows ISO 14721: Open Archive Information System (OAIS)[2], ISO 16363: Audit and Certification of Trusted Digital Repositories (TDR)[3] and EN9300 series: Long Term Archiving and Retrieval (LOTAR[4]).

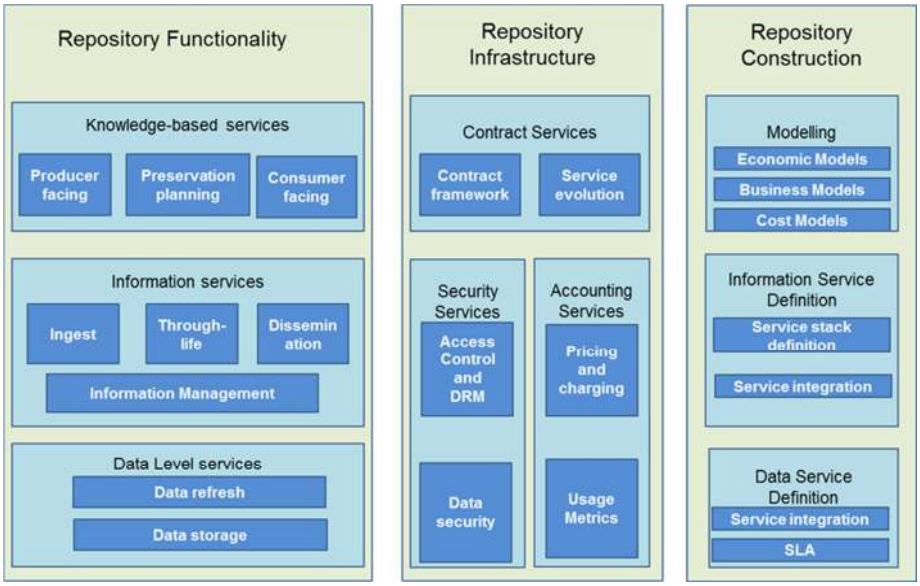

Figure 1 RASSC Service Model (copyright BAE Systems)

1.1 In scope

The scope of the cost model is for running the repository, including internal functions for preservation as well as providing services for ingest (receiving and processing Submission Information Packages - SIPs) and services for access (creating and delivering Dissemination Information Packages - DIPs).

The model includes the costs of operating, maintaining and upgrading the repository over-time, for example refreshes or migrations of the hardware, software, people and processes involved.

1.2 Not in scope

The cost model does not include:

- The costs associated with finding, aggregating, formatting and delivering data to the repository service, i.e. creating a SIP.
- The costs associated with the identification, request, receipt and use of data extracted from the service, i.e. deciding what DIP to request and then using it.
- The initial cost associated with the planning, design, procurement and deployment of the repository.

[2] http://www.iso.org/iso/catalogue_detail.htm?csnumber=24683
[3] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=56510
[4] http://www.lotar-international.org/home.html

2 Cost modelling approaches

Long-term cost modelling is an active area of research and development. Many institutions are interested in the Total Cost of Preservation (TCP) over time for their assets.

Existing cost modelling approaches can be split, roughly speaking, into three main classes:

1. Empirical models based on the preservation lifecycle where costs are estimated for each of the functions at the different stages of the lifecycle. Each stage is broken down into smaller and smaller functions until specific cost estimates can be calculated.

2. Cost estimates based on previously incurred costs of similar preservation projects or activities. Data collected from past experience is extrapolated or interpolated to predict future costs.

3. Simulations of the operation of a repository based on the services provided, processes followed and resources used. Cost data is calculated, collected and aggregated as the simulation progresses.

2.1 Lifecycle cost models.

The Digital Preservation Coalition defines[5] preservation as "Digital Preservation Refers to the series of managed activities necessary to ensure continued access to digital materials for as long as necessary." Preservation means enabling access, i.e. ensuring that data can be correctly interpreted and used by a designated community. Preservation involves activities across the complete content lifecycle of and hence many cost modelling approaches analyse the costs associated with each stage in the lifecycle.

Examples of this approach include:

- LIFE model developed by the British Library[6].
- KRDS model for research data developed by Neil Beagrie[7].
- California Digital Library (CDL) Total Cost Preservation (TCP) model[8].
- Danish National Archives (DNA) Cost Model Digital Preservation (CMDP)[9].

The Digital Curation Centre (DCC) lifecycle model provides a useful checkpoint when deciding what is in or out of scope of a cost model since it highlights the broad range of activities involved in preservation beyond retention of digital objects, e.g. planning, community watch and metadata.

[5] http://www.dpconline.org/advice/preservationhandbook/introduction/definitions-and-concepts
[6] http://www.life.ac.uk/2/documentation.shtml
[7] http://beagrie.com/krds-i2s2.php
[8] http://wiki.ucop.edu/display/Curation/Cost+Modeling
[9] http://www.ijdc.net/index.php/ijdc/article/viewFile/177/246

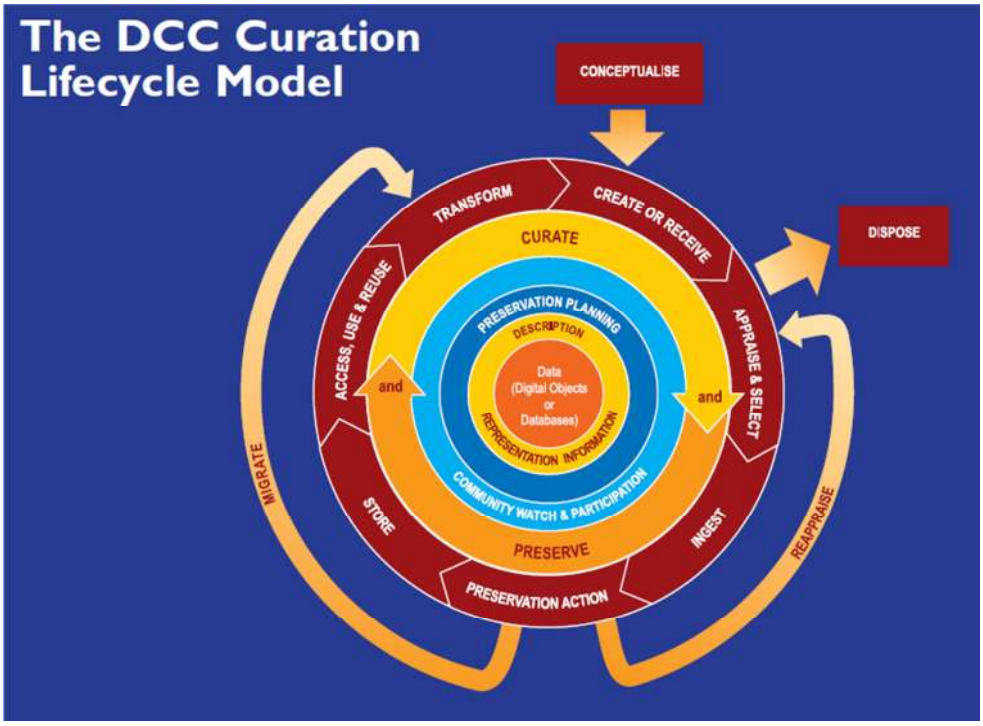

Figure 2 DCC Lifecycle model (reproduced from the DCC website http://www.dcc.ac.uk/lifecycle-model/)

ISO16363 Trusted Digital Repositories (evolution of TRAC: trusted repositories audit criteria) is also worth a mention. Although not a cost model or lifecycle per se, ISO16363 does specify a set of criteria for assessing the trustworthiness of a repository and provides some guidelines on how to meet those criteria. In this sense it provides an additional checklist on what will be needed within a repository and hence where costs will be generated.

2.1.1 LIFE

A general approach to cost modelling is the work of the LIFE project[10] coordinated by the British library. This has developed a detailed lifecycle for digital preservation and then developed methods to estimate the costs of each stage in the lifecycle over time.

The basic elements of the cost model are:

$$L_T = Aq + I_T + M_T + Ac_T + S_T + P_T$$

Figure 3 LIFE cost model (reproduced from How much does it cost? The LIFE Project - Costing Models for Digital Curation and Preservation[11])

Where:

- L_T = Total cost
- Aq = Acquisition cost
- I = Ingest cost

[10] http://www.life.ac.uk/
[11] http://liber.library.uu.nl/publish/issues/2007-3_4/index.html?000210

- M = metadata cost
- Ac = Access cost
- S = Storage cost
- P = preservation cost.

The subscript T means that costs have to be calculated over the lifetime of the items being preserved.

A valuable output from LIFE are the series of case studies[12] (web archiving, e-Journals, newspapers etc.) from the partners involved that include detailed spreadsheets implementing the LIFE model and provide real-world worked examples of what the costs of preservation really are. The LIFE approach is founded on considering cost over time, e.g. for 5,10 or 20 year periods, with the result that the LIFE model and examples include activities such as migration and time varying costs such as storage.

Acquisition	Ingest	Metadata	Access	Storage	Preservation
Selection (Aq1)	Quality Assurance (I1)	Characteri-sation M1)	Reference Linking (Ac1)	Bit-stream Storage Costs (S1)	Technology Watch (P1)
IPR (Aq2)	Deposit (I2)	Descriptive (M2)	User Support (Ac2)		Preservation Tool Cost (P2)
Licensing (Aq3)	Holdings Update (I3)	Administrative (M3)	Access Mechanism (Ac3)		Preservation Metadata (P3)
Ordering & Invoicing (Aq4)					Preservation Action (P4)
Obtaining (Aq5)					Quality Assurance (P5)
Check-in (Aq6)					

Figure 4 Breakdown of cost elements in the Life model (reproduced from How much does it cost? The LIFE Project - Costing Models for Digital Curation and Preservation)

[12] http://www.life.ac.uk/2/documentation.shtml

2.1.2 California Digital Library (CDL) Total Cost of Preservation (TCP) model

The CDL approach is to take a OAIS and service oriented view of the cost modelling problem[13]. The elements of the model are shown below.

Preservation activities are embodied in an archival

1. ***System***; composed of various
2. ***Services*** supporting necessary and desirable functions; running on
3. ***Servers***; designed, deployed, maintained, enhanced, and utilized by
4. ***Staff***; in support of content
5. ***Producers***; who use
6. ***Workflows*** to submit instances of
7. ***Content Types***; which occupy
8. ***Storage***; and are subject to ongoing
9. ***Monitoring***; and periodic
10. ***Interventions***; all subject to appropriate managerial
11. ***Oversight***.

Figure 5 CDL breakdown of preservation activities reproduced from Total Cost of Preservation (TCP): Cost Modeling for Sustainable Services

The model considers the need to support Producers who submit content to be preserved, but not Consumers who subsequently need to access and use that content. Access can be a major if not dominant cost in digital preservation.

$$TCP = A + n \cdot P + m \cdot W + \ell \cdot C + k \cdot S + j \cdot M + i \cdot V + O$$

TCP	Total cost of preservation for all *Producers*.
A	Fixed cost of the baseline archival *System*.
n	Number of content *Producers*.
P	Unit cost of supporting a *Producer*.
m	Number of submission *Workflows*.
W	Unit cost of supporting a *Workflow*.
ℓ	Number of *Content Types*.
C	Unit cost of supporting a *Content Type*.
k	Number of units of preservation *Storage*.
S	Unit cost of *Storage*.
j	Number of preservation *Monitoring* activities.
M	Unit cost of a *Monitoring* activity.
i	Number of preservation *Interventions*.
V	Unit cost of an *Intervention*.
O	Fixed cost of administrative and managerial *Oversight*.

Figure 6 Cost model parameters reproduced from Total Cost of Preservation (TCP): Cost Modeling for Sustainable Services.

[13] https://wiki.ucop.edu/download/attachments/163610649/TCP-total-cost-of-preservation.pdf?version=5&modificationDate=1336402730000

The model divides costs into fixed costs and recurring costs where costs can be one off or recurring. This allows the model to include one-off capital expenditures, e.g. large items of equipment that are needed to establish a service, as well as unit costs where the total cost of the service is proportional to usage, e.g. units of storage.

To calculate the cost per Producer, the approach is to apportion fixed costs equally across all n Producers and then add the unit costs incurred by the specific Producer. This results in a Pay As You Go (PAYG) cost per Producer for a period of time, e.g. a year.

By then applying a discount factor d to the PAYG cost G, the long-term total cost is calculated over T periods.

$$G(T,d) = G \cdot \frac{1-(1-d)^T}{d}$$

Finally, the model calculates the Paid Up Price (otherwise known as an endowment) that is paid by the Producer in order to cover the total cost. This includes the interest earned by the cash whilst it is drawn down to pay the costs of the service.

$$F(T,d,r) = G \cdot \frac{e}{r} \cdot \frac{(1+e)^T-(1-d)^T}{(1+e)^T \cdot (e+d)}$$

r is the nominal annual percentage rate (APR) of investment return and e is the effective annual rate including monthly compound interest.

$$e = \left(1+\frac{r}{12}\right)^{12} - 1$$

This then leads to the ability to compare PAYG (including discounts) with Paid-Up pricing.

Figure 7 Total cost over as a function of time. Reproduced from Total Cost of Preservation (TCP): Cost Modeling for Sustainable Services

2.1.3 KRDS (Keeping Research Data Safe)

The KRDS cost model takes a lifecycle approach (see stages in table below taken from KRDS user guide) to breakdown costs into different categories. The recommendation is then to build a spreadsheet model of the total cost, inc. annual discounting for estimating long term costs. In this respect, the KRDS model is similar to LIFE and other lifecycle/spreadsheet approaches.

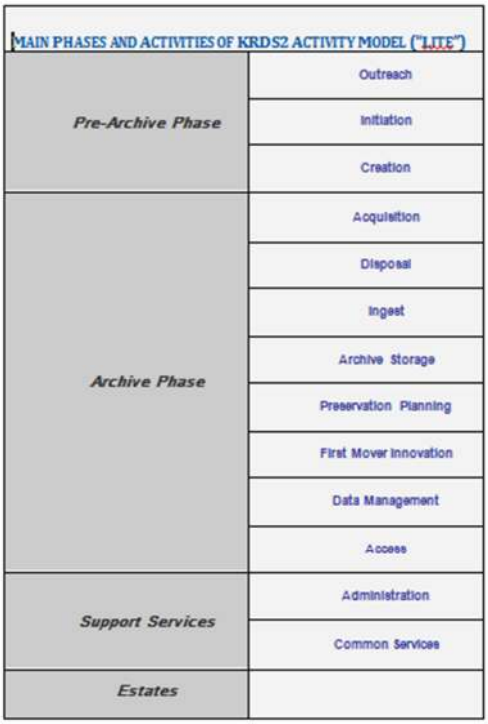

Figure 8 Main phases/activities in the KRDS model (reproduced from KRDS2 activity model http://www.beagrie.com/krds.php)

More interesting in KRDS are tools/guidelines for doing a corresponding value and benefits analysis[14] This considers internal and external beneficiaries, direct and indirect benefits and whether the benefits accrue in the short or long term. Examples are included of generic benefits for research data. By considering tangible and intangible benefits, the model is not dissimilar from earlier work done by eSpida[15] that used a Kaplan and Norton balanced scorecard approach.

[14] http://www.beagrie.com/intro_benefits%20analysis%20toolkit_0711.pdf
[15] http://www.gla.ac.uk/services/library/espida/

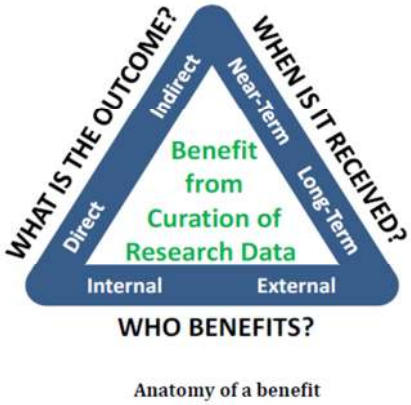

Figure 9 KRDS model of benefits of research data curation. Reproduced from KRDS toolkit guide http://www.beagrie.com/krds.php

2.1.4 Danish National Archive (DNA) Cost Model Digital Preservation (CMDP)

Like the TCP model from CDL model, the CMDP from DNA also bases itself on OAIS. Perhaps the most interesting part of the model is how it treats the costs that arise from the different formats that need to be handled (documents, images etc.) as shown in Figure 10.

*Format Interpretation (pw) =number of pages * time per page (min) * complexity (L, M, H)*

Format	Specifications and other relevant documentation	No. of pages	Complexity	Quality
TXT	ISO 10646 ISO 646	20 15	L L	H H
PDF/A 1.0	PDF/A (ISO 19005-1) PDF 1.4 (ISO 32000-1)	29 700	L H	M M
TIFF 6.0 LZW	TIFF 6.0 Baseline LZW (ISO 12639:2004)	121	M	H
GML 3.X	ISO 19136 2007 ISO 19100-serie (Open GIS) 19103 19104 19107 19108 19109 19111 19123 (understanding of xml, xml schema and Xlink assumed)	380 67 102 166 48 71 78 65	H	H

Table 7 Examples of how different formats' documentations (no. of pages, complexity and quality) have been evaluated as basis for calculating the Format Interpretation factor.

Figure 10 Variables used in establishing a measure of effort associated with the handling and preserving of different file formats. Cost in this sense is a 'Format Interpretation' factor that is subsequently used to estimate the effort involved in different activities involving a format, e.g. writing or supporting software to read or migrate that format. Reproduced from CMDP project report http://www.costmodelfordigitalpreservation.dk/

This is then used to calculate the cost of different preservation activities, e.g. format migration which includes development/testing of format conversion tools and use computational resources to do conversion.

For example, the migration cost in person weeks of effort is calculated to be:

Migration Cost (pw): Format Interpretation + Software Provision + Migration Processing

The DNA then compared the predictions made by their model with costs of actual preservation projects done in the past. This is shown in Figure 11 for one of the projects. The difference is significant in most areas, with the model typically estimating costs that are higher or lower than reality by 50% or more. This shows that accurate cost prediction is not easy. Indeed, getting within 50% should be considered a good result.

	Case 1		CMDP		CMDP - Case 1	
	pw	%	pw	%	Δ pw	%
IP Designs	**44**	**12**	**50**	**24**	**6**	**12**
A (1968-1998)	29	66	20	40	-9	-31
B (1999-2000)	15	34	16	32	1	6
C (2001-2004)	0	0	14	28	14	n.a.
B & C	15	34	30	60	15	50
Migration Plans	**150**	**42**	**39**	**19**	**-111**	**-74**
A (1968-1998)	105	70	15	38	-90	-86
B (1999-2000)	30	20	14	36	-16	-53
C (2001-2004)	15	10	10	26	-5	-33
B & C	45	30	24	62	-21	-47
Prototypes (Software Provision)	**164**	**46**	**116**	**57**	**-48**	**-29**
A (1968-1998)	101	62	48	41	-53	-52
B (1999-2000)	50	30	36	31	-14	-28
C (2001-2004)	12	7	32	28	20	62,5
B & C	62	38	68	59	6	9
Migration Package (total)	**358**	**100**	**205**	**100**	-153	**-43**

Figure 11 Comparison of actual costs (expressed as person weeks of effort) for a preservation project (Case 1) with the predicted costs from the CMDP model. Reproduced from CMDP project report http://www.costmodelfordigitalpreservation.dk/

2.2 Cost models based on historical data

The most relevant example here is the NASA Cost Estimation Toolkit (CET)[16]. This is used to calculate mission costs and has limited detail on the long-term retention and access to data from a mission. CET is a useful reminder that probably the best way to estimate future costs is from past experience – but only if there is enough data to hand – which in turn requires a proactive effort to ensuring the right data is collected from the outset. It is unlikely that this data is available for RASSC and therefore the CET type approach is ruled-out. However, in the long-term as a RASSC service becomes established and operated, this approach should become increasingly viable.

2.3 Cost models based on simulation

This is the approach currently taken by David Rosenthal (Stanford University) for his long-term cost modelling work, much of which is described on his blog[17]. David's current focus is on the effects of uncertainty or variability of the inputs to a cost model (e.g. interest rates) on long-term costs through use of Monte Carlo techniques[18].

[16] http://opensource.gsfc.nasa.gov/projects/CET/CET.php
[17] http://blog.dshr.org
[18] http://blog.dshr.org/2011/09/modeling-economics-of-long-term-storage.htm

A simulation approach has also been taken by IT Innovation to cost modelling in other application domains, for example, audiovisual (AV) preservation. Our approach is to model preservation processes and their associated costs using a Discrete Event Simulation with a stochastic approach to event generation (e.g. ingest or access workloads, data corruption).

The main benefits of Monte Carlo or stochastic approaches are to generate a probability distribution for costs over time. This allows actuarial analysis of long-term costs, e.g. what is the probability that costs will not go above a given limit. For example, this can be important when considering 'endowment models' where the question is what one-off sum of money needs to be invested today to secure the long-term preservation of data, e.g. over decade or century timescales.

3 Requirements for RASSC cost model

A set of services have been defined for the RASSC repository. These are based on LOTAR, which in turn makes heavy use of the OAIS model. Therefore, the most obvious approach is to base a cost model on the functional areas of OAIS in a similar way to that done by the Danish National Archive.

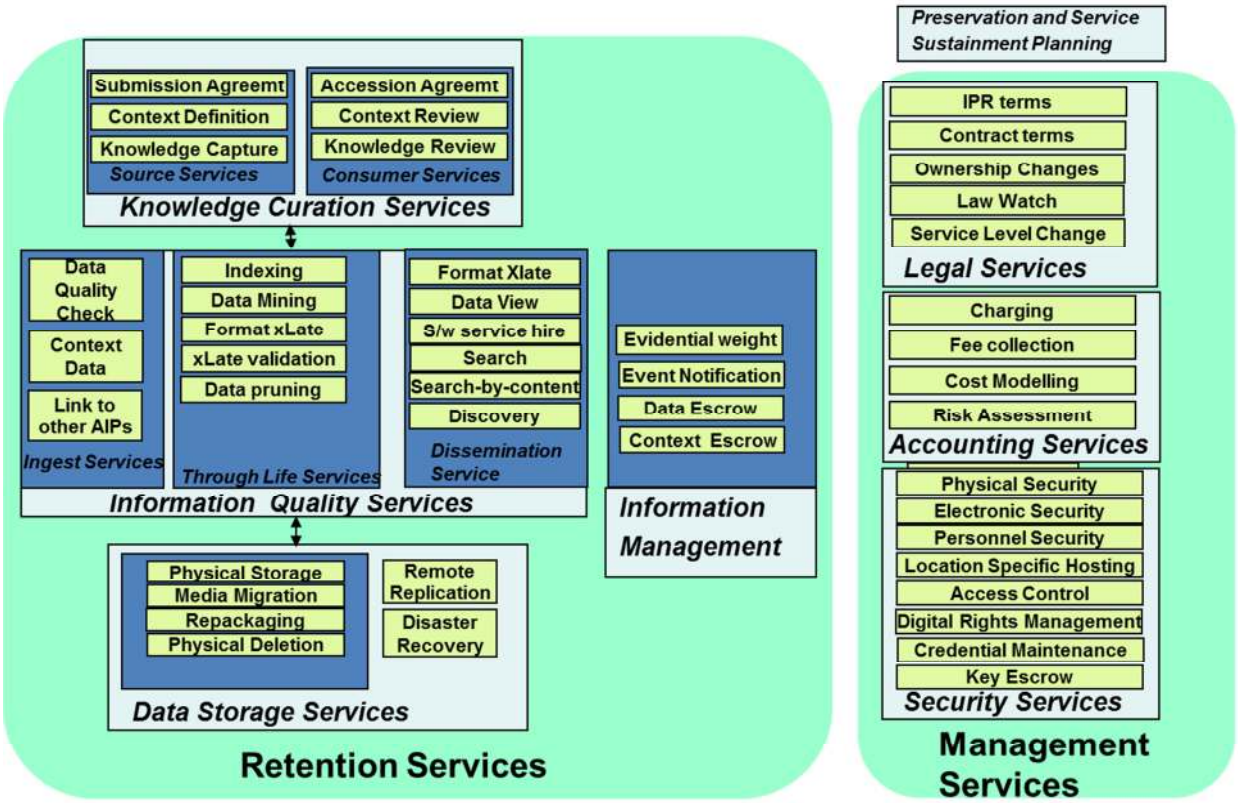

Figure 12 Detailed RASSC Service Model (copyright BAE Systems)

The use case deliverable (D4.1a) highlights some specific requirements of the repository that have an impact on the cost model, or at the least the major cost areas within the cost model.

- The repository will be used to hold multiple types of data, e.g. design data and IVHM data.

- Each type of data has its own characteristics (retention period, format, volume, ingest and access pattern, reason for retention etc.).

- Data submitted to the repository must be quality checked.

- Two types of design data that are requested from the repository – exact data and 'accurate enough' data. How this data is generated is not specified, e.g. on demand or up-front. The model needs to support these two different designated communities and how they are serviced.

- Evidential weight is an important part of asserting the integrity and authenticity of design data used for certification. Therefore, the repository needs to capture/create 'supporting evidence' in addition to the actual 'evidence' submitted in SIPs. The supporting evidence gives evidential weight to the evidence, e.g. it can be shown to not have changed.

One area that merits further explanation isthe role of the repository w.r.t. software used to interpret/use data held by the repository.

The long-term sustainment of software (e.g. CAD tools) is not a basic function of the repository. It could however be provided as an additional repository service if required to service a particular designated community. The approach of sustaining the original software can be tackled using a 'museum' approach, i.e. everything required to run the software is maintained including underlying operating system, computer hardware etc. or it can be tackled using an emulation approach whereby a more modern infrastructure is made to emulate the behaviour of the original stack which in turn allows the original software application to be run.

The act of migrating data between versions of a software application used to read it, e.g. a CAD tool, is however typically seen as a basic function of the repository since this is the usual strategy employed to maintain the ability to use data. Again there are some subtleties here that depend on institutional practice. For example, at BAE Systems, the Design Office (DO) approach is that when viewing a model of an aircraft part, then the original model is used in newer versions of the software, but if the model itself needs to be changed, e.g. because the part is being modified, then the part is then re-modelled in the newer version of the software (not converted) – at which point a new model has been created and also needs to be preserved. The reason for this approach is that the consequential effects of an error in data migration may be substantial, so that any migrated model must be revalidated, which is expensive.

This migration strategy will also change over time, e.g. as longer-lived and open formats emerge. For example, for CAD models the conversion from CATIA4 to CATIA5 is known to be problematic due to a fundamental change in CATIA's modelling approach. However, there is now a move to a STEP format for 3D model data, which not only improves model exchange between tools or organisations, but also provides a much more open and longer-lived data format that means there should be no need to change the model description when using subsequent versions of CATIA (or at least until they decide to overhaul the way their software works again).

In any event, checks are still to ensure that key characteristics (significant properties in ISO16363) have been correctly maintained when newer versions of a software application are used. For example, for a 3D CAD model of a part, LOTAR defines the key characteristic to be

the part's shape, and ways of detecting changes to the shape include changes to the volume, centroid and area. Changing a model so all three characteristics are unchanged will only be possible in an extremely limited number of ways and hence this simple approach will trap the vast majority of errors in misinterpretation of a model in a newer software version). It is in this way that LOTAR sets requirements for data sustainment in terms of the outcomes of sustainment. These include the migration to new platforms and software without loss of information. No loss of information is defined by maintaining the key characteristics of the data being sustained

This still leaves the issue of what happens if newer versions of the software fail to correctly interpret older models. The choice are to re-model (which has a significant cost), resolve the issue with the software vendor (if they are willing and able to do so) or maintain the ability to use the older software (which means keeping the stack alive and having to deal with software licensing issues). In general there is no single solution to this problem, so it is a case of planning based on the best strategy at the time and then monitoring the situation and adjusting the plan if necessary.

4 RASSC cost model

4.1 Options

There are many approaches that can be taken when building a cost model.

- Take a lifecycle based approach and associate costs with each stage
- Use the TCP model from CDL and adapt/extend (e.g. to include access)
- Use ISO16363 and associate costs with the functions required to meet the criteria
- Use the functional services in the RASSC stack and associate costs with each one

4.2 Simple empirical model

Our model follows the approach of CMDP in using OAIS as the basis of a cost model and CDL in separating out the costs associated with producers, content and workflows (but in our case with an extension to include consumers). We also follow the CMDP approach of needing to consider separately the costs of each content type that needs to be preserved.

A diagrammatic representation of the model is shown below.

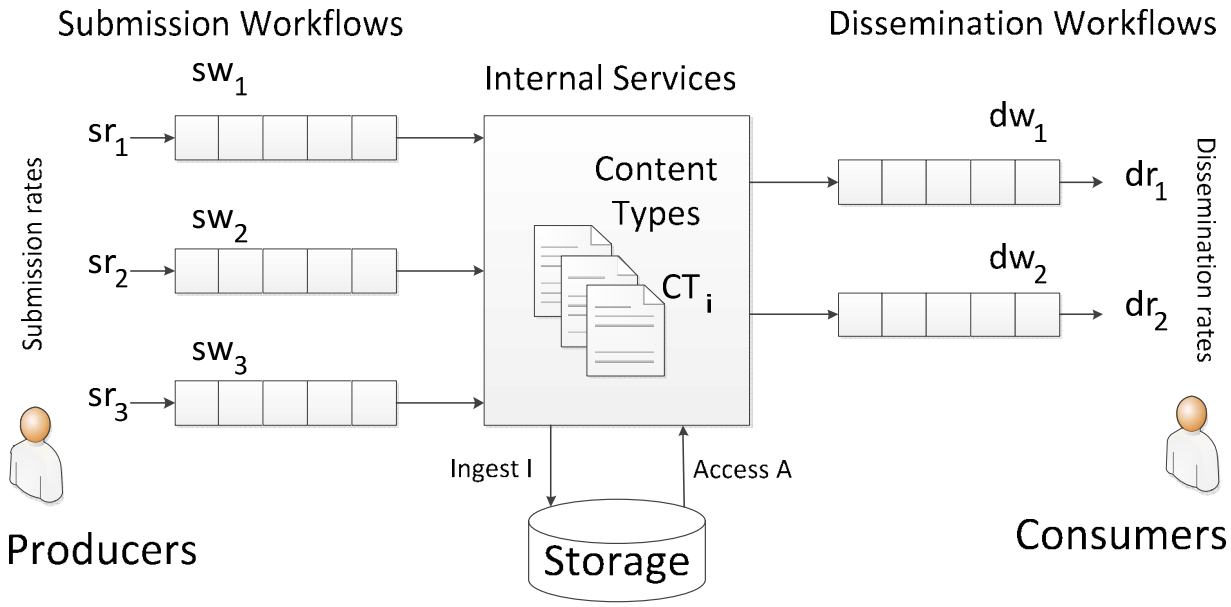

Figure 13 RASSC cost model

Producers submit SIPs to the repository. SIPs are received and processed using a submission workflow. There can be more than one workflow depending on the type of content, e.g. because different validation checks are required or different operations are needed to assemble AIPs from one or more SIPs.

Consumers request DIPs from the repository. DIPs are generated and delivered using a dissemination workflow. The can be more than one workflow depending on the type of content or the need to do various types of extraction and processing of AIPs in order to create a DIP.

Internal Services are responsible for the long-term preservation of the AIPs and do so for each Content Type that needs to be supported. The AIPs are held in a storage system that has ingest, access and storage costs.

4.3 Model Parameters

Submission Parameters

$sw_i = submission\ workflow\ i$

$sr_i = rate\ of\ SIP\ submissions\ to\ sw_i$

$U_{sw_i} = Unit\ cost\ of\ a\ SIP\ submission\ to\ sw_i$

$F_{sw_i} = Fixed\ cost\ of\ sw_i$

$n_p = Number\ of\ producers$

$F_p = Fixed\ cost\ of\ supporting\ producers$

$U_p = Unit\ cost\ of\ adding\ a\ producer\ to\ the\ system$

Dissemination Parameters

$dw_i = dissemination\ workflow\ i$

$dr_i = rate\ of\ DIPs\ delivered\ through\ dw_i$

$U_{dw_i} = Unit\ cost\ of\ a\ DIP\ dissemination\ through\ dw_i$

$F_{dw_i} = Fixed\ cost\ of\ dw_i$

$n_c = Number\ of\ consumers$

$F_c = Fixed\ cost\ of\ supporting\ consumers$

$U_c = Unit\ cost\ of\ adding\ a\ consumer\ to\ the\ system$

Storage and Content Type Parameters

$U_I = Unit\ cost\ of\ ingest\ of\ data\ into\ storage$

$U_A = Unit\ cost\ of\ access\ to\ data\ in\ storage$

$F_I = Fixed\ cost\ of\ ingest\ of\ data\ into\ storage$

$F_A = Fixed\ cost\ of\ access\ to\ data\ in\ storage$

$IV_i = Storage\ ingest\ data\ volume\ resulting\ from\ a\ SIP\ submission\ to\ sw_i$

$AV_i = Storage\ access\ data\ volume\ needed\ to\ produce\ a\ DIP\ for\ dissemination\ through\ dw_i$

$F_S = Fixed\ cost\ of\ storage$

$U_S = Unit\ cost\ of\ storage$

$n_{CT} = Number\ of\ different\ content\ types$

$F_{CT} = Fixed\ cost\ of\ supporting\ content\ types$

$U_{CT} = Unit\ cost\ of\ adding\ a\ Content\ Type\ to\ the\ system$

General Parameters

$T = duration\ of\ the\ cost\ projection$

4.4 Cost model

Submission costs

$$C_{sw} = Submission\ Workflow\ cost = \sum_i sr_i . U_{sw_i} . T + \sum_i F_{sw_i}$$

$$C_p = Producer\ support\ cost = F_p + U_p . n_p$$

Dissemination costs

$$C_{dw} = Dissemination\ Workflow\ cost = \sum_i dr_i . U_{dw_i} . T + \sum_i F_{dw_i}$$

$$C_c = Consumer\ support\ cost = F_c + U_c . n_c$$

Costs of Storage and preservation of Content Types

$$C_{CT} = Content\ Type\ cost = F_{CT} + U_{CT} . n_{CT}$$

$$C_{SI} = Storage\ ingest\ cost = \sum_i sr_i . IV_i . U_I . T + F_I$$

$$C_{SA} = Storage\ access\ cost = \sum_i dr_i . AV_i . U_A . T + F_A$$

$$C_S = Storage\ cost = F_S + \sum_i sr_i . IV_i . U_S . T$$

Total cost of preservation

$$TCP = Total\ preservation\ cost\ over\ time\ T = C_{sw} + C_{dw} + C_p + C_c + C_{CT} + C_{SI} + C_{SA} + C_S$$

5 Example cost areas

Some examples of generic areas where cost is incurred are given below.

Cost area	Example costs
Submission	Submission agreement. SIP submission. SIP validation. Chain of custody from producer to repository. Collecting supporting evidence. Knowledge capture.
Dissemination	Search and navigation. Building DIPs. Clearing rights. Enforcing security for access. Access agreements. Data mining and indexing. Data processing services.
Content Types	Assembling and validating AIP. Technology watch. Format migrations. Validation that key characteristics are maintained. Support for software used to create/use each content types. Mapping between SIP workflows and Content Types. Maintaining evidential weight. Retention policies.
Support for Producers and Consumers	User management infrastructure. Authentication of new users. Security infrastructure. Training. Documentation. Accounting and billing for usage. Contracts.
Storage	Ingest and access of data – i/o costs (network, performance of storage). Storage and bit level preservation. DR, replication, media/software/hardware refresh and migration. Secure deletion.

A very simplified example of how to use the cost model is shown below. This is based on the IVHM scenario. The scenario includes flight data being added to the repository and then the repository supporting analysis services for the flight data as part of dissemination. The analysis services are modelled as supported by a utility compute infrastructure (IaaS model) so that costs are only incurred for the processing that needs to be done.

The scenario doesn't include all the other types of information that need to be collected along with the flight data, e.g. configuration and stores of the aircraft, maintenance schedule and any previous problems, details of the parts and assemblies etc.

Submission Parameters	
$sr_i = rate\ of\ SIP\ submissions\ to\ sw_i$	number of aircraft flights each day that generate IVHM data submissions.
$U_{sw_i} = Unit\ cost\ of\ a\ SIP\ submission\ to\ sw_i$	processing and validation of IVHM data from a flight, e.g. operator time and error handling
$F_{sw_i} = Fixed\ cost\ of\ sw_i$	licensing costs of Share-a-space, server for running software, development and testing

n_p = *Number of producers*	number of flight technicians logging IVHM data
F_p = *Fixed cost of supporting producers*	development of a user guide
U_p = *Unit cost of adding a producer to the system*	setting up security, e.g. access permissions, training of producers

Dissemination Parameters	
dr_i = *rate of DIPs delivered through* dw_i	requests for fatigue analysis service
U_{dw_i} = *Unit cost of a DIP dissemination through* dw_i	cpu cycles and temp storage used to do an analysis run (based on a utility model)
F_{dw_i} = *Fixed cost of* dw_i	development and test of service
n_c = *Number of consumers*	number of engineers doing fatigue analysis
F_c = *Fixed cost of supporting consumers*	user guide
U_c = *Unit cost of adding a consumer to the system*	security, training

Storage and preservation of Content Types	
U_I = *Unit cost of ingest of data into storage*	cost per GB of data transferred into storage
U_A = *Unit cost of access to data in storage*	cost per GB of data transferred out of storage
F_I = *Fixed cost of ingest of data into storage*	network infrastructure
F_A = *Fixed cost of access to data in storage*	network infrastructure
IV_i = *Storage ingest data volume resulting from a SIP*	volume of data for each flight
AV_i = *Storage access data volume needed to produce a DIP*	volume of data for a specific part/assembly that needs fatigue analysis for one or more flights.
F_S = *Fixed cost of storage*	Tape library
U_S = *Unit cost of storage*	Cost per TB on data tape media
n_{CT} = *Number of different content types*	One: flight data
F_{CT} = *Fixed cost of supporting content types*	Digital Asset Management tool
U_{CT} = *Unit cost of adding a Content Type to the system*	Establishing key characteristics, migration and validation tools.

6 TCP over time

The empirical model for RASSC assumes that the costs over time period T are constant. This will not be true where T is significant, e.g. a year or more.

There are various approaches to modifying the model to include the way costs will vary with time, for example.

- Make each cost element a function of time and then sum over a set of time periods. For example a Unit cost U becomes U=U(t).

- Apply a discount factor to each cost element (e.g. each year) and sum over a set of time periods (e.g. 20 years). This is a Discounted Cash Flow (DCF) approach.

The CDL TCP model uses a DCF approach. It assumes that the discount d can be applied again in successive periods, i.e. the unit cost after j periods is $U_j = (1-d)^j U_o$ where U_o is the initial unit cost. This then allows the costs to be summed-up 'forever' resulting in a converging series.

DCF with a fixed discount is attractive as it makes the calculation simple and generates a single 'forever cost'. However, it does include major assumptions on how costs behave over time. For example, in the CDL model the assumption is that the discount is the same for each period and is the same for all elements of the cost.

The approaches of DCF v.s. making each cost element a function of time are of course the same thing, it's just a case of whether to make the discount an explicit factor or to roll it into the cost element function.

The approach we propose for RASSC is to make cost elements a function of time to give the most flexibility. This allows different discounts to be applied to different cost elements.

7 Detailed simulation of costs

The cost modelling approaches described already in this report allow for the following aspects of cost to be explored:

- How Total Cost of Ownership (TCO) of storage, including trends for the costs in each of the components (e.g. media, servers, power, space, cooling, maintenance).
- How TCO varies with time because of the active nature of digital preservation (e.g. media refreshes, server migrations, proactive measures to ensure data safety e.g. scrubbing).
- How data volumes and rates of data ingest and access over time affect costs, i.e. the impact of different usage levels and storage capacity.
- The role economic parameters are also important for long-term budgeting, e.g. the need to consider interest rate trends when using discounted cash-flow techniques.

However, there is a great deal of uncertainty around many of the parameters used for cost modelling, for example: ingest rates, cost trends, and technology roadmaps.

Furthermore, the people, processes and systems used to implement a trusted digital repository cannot be assumed to be 100% reliable – there is a probability of failures or errors in all parts of the overall system. These need to be mitigated (or tolerated) and the costs and residual risks also included in the model.

To address these aspects, especially the stochastic nature of many of the parameters of the model, a simulation based approach is needed. This can be used to do many runs of the same model across the range of values of the input parameters, for example using a Monte Carlo approach.

In this section, we describe such an approach and show the type of results it can produce. The approach necessitates more complex and detailed models than described in Section 4. This takes time and effort to develop and in RASSC we have focussed on just one part of the problem: long term storage and access using commodity IT storage technologies.

The interactive simulation tool developed has been created by IT Innovation in conjunction with the EC supported PrestoPRIME project[19]. The uses tool is called iModel and used a discrete event simulation approach. The simulation contains one or more storage systems, each of which is modelled as providing a set of services (e.g. ingest, access, checksum validation). Each service uses one or more resources (e.g. copying data, checking integrity). Requests to use a service are added to a queue for that service (e.g. queue of files to be ingested) where each request is then taken from the queue for processing if sufficient resources are available.

During the simulation, time ticks away and events are generated (e.g. random corruption of files in a storage system, requests to access a file, new files to be added to the archive). These events can trigger actions, e.g. a copy/repair process might be triggered if a file access event identifies that a copy of a file is corrupted. These actions then are added to the relevant service queues (e.g. file access queue for access events, file copy queue used as part of a repair process or scheduled file migration).

[19] http://www.prestoprime.org/

A storage system will process items in the queues for its services according to how much resource it has available (e.g. serving access requests sequentially or in parallel). The available capacity of the resources used by each service determines how many items in the queue for that service will be processed for each tick of the clock. If there is insufficient resource then not all items in a queue will be dealt with and the unprocessed items remain in the queue and are carried over to the next tick of the clock.

For a simulation of more than one storage system, a series of interactions are defined between storage systems, for example replicating files. In this way, the services for the storage systems become coupled. For example, if storage system 1 is used for ingest of files and the policy is to replicate those files to storage system 2 and storage system 3 before ingest is considered successful, then the rate at which items will be processed on the ingest queue is dependent on the copy resources available to create replicas of the file on the other storage systems. A set of template configurations are provided that correspond to common patterns for real world storage configurations, e.g. mirrored servers, HSM, online primary storage server plus deep archive for disaster recovery.

The core of the simulation is a relatively simple one – a set of services with queues and resources, a set of event generators and a set of template configurations for how storage systems are connected together.

On top of the core simulation is the user interface that allows the user to set parameters, interact with the simulation, and view results. This is where specific UI features are used, e.g. sliders, radio buttons, auto scaling graphs, easy tabbing between storage systems – all of which are designed to make the tool easy to use and tailored to the problem of cost and loss simulation.

The cost model used by the simulation is based on the premise that use of resources by each service will incur a cost (e.g. ingest, access and storage all have a cost). The cost is accumulated as the clock ticks. By attaching costs to resources, the different costs for each storage system can be accounted, e.g. resources used for copying files, checking their integrity, performing local repair or providing access. This allows the simulation to be easily extended if needed by simply adding further resources and costs. For example, should the model need to include the costs of archive activities such as cataloguing or rights clearance then these can be added to the ingest service.

The tool is implemented in Java and is available online[20] as open source (LGPL license).
Existing simulation frameworks were considered (e.g. Simul8[21], iGrafx[22], SimEvents[23], PRISM[24]). Whilst some are able to cover the core of the simulation, they all have difficulties when it comes to building custom user interfaces, using non-standard probability distributions or queue disciplines, and allowing user interaction and changes to the settings during simulation. These factors would make the tool hard to develop on one of these platforms and in particular hard to extend to include more complex functionality. There is

[20] http://prestoprime.it-innovation.soton.ac.uk
[21] Simul8 Simulation Software http://www.simul8.com/
[22] iGrafx process simulation and analysis tool http://www.igrafx.de/
[23] SimEvents discrete event simulation engine (part of Simulink from Mathworks) http://www.mathworks.co.uk/products/simevents/index.html
[24] PRISM probabalistic model checker http://www.prismmodelchecker.org/

also the major problem that these frameworks are mostly commercial and expensive to license which would significantly limit the ability to provide the tool to the community to use for free.

7.1 Model

Whilst the architecture of the simulation is relatively simple, the tool is provided with functionality that aims to model realistic corruption and storage system management processes. For this purpose, a detailed data storage model has been developed where archived assets are represented as file objects that include such properties as name, size, and corruption details. Each asset can have more than one file (replica) representing it within the system. Consequently, each storage system contains a list of such items and is responsible for their storage and integrity management.

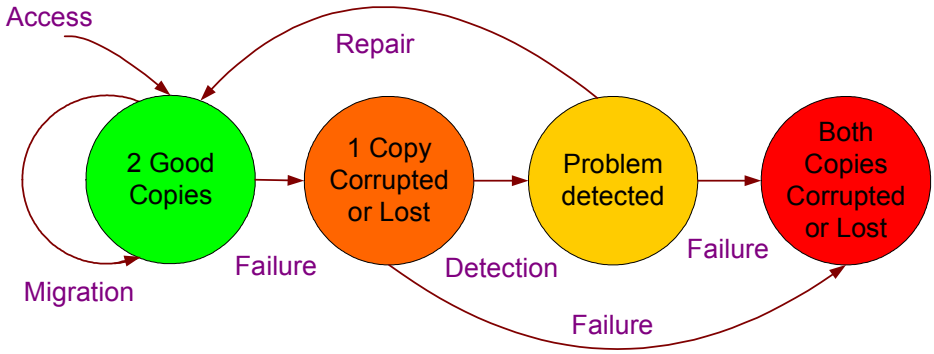

Figure 14 Conceptual model of storage, access, migration and integrity management

Figure 14 presents a simple conceptual model for analysing cost and risk for data storage and access. With reference to Figure 14, the bedrock of data safety is to keep multiple copies of each data object (green circle), e.g. by using different technologies in different locations, and ideally operated by different people. This guards against major risks, e.g. by enabling disaster recovery, but also guards against unanticipated problems with individual technologies and processes, i.e. it ensures eggs are not 'all in one basket' at any level. The diagram shows the simplest version of this approach: keep two independent copies of each data object. Each copy is stored in a storage system of some description. One or more of these storage systems is used to serve requests (access) for data objects already in the systems, or to receive new data objects (ingest).

For each storage system used to hold a copy of each data object there is the need to regularly migrate each component of the technology stack (hardware, operating system, management software, formats etc.) to address technical obsolescence, media degradation, and to provide improved capacity or performance. At the same time as data is being stored, accessed or migrated there is always the chance that one of the copies is damaged or lost resulting from a failure in the corresponding system used to store it (orange circle). This can be modelled as a probabilistic process where risks (e.g. data corruption) are represented as probabilities of transitioning between the states. But only after the corruption is detected (yellow circle) can any action can be taken, e.g. to repair or replace the damaged or lost copy by using the remaining good copy. If at any time something happens to the second copy (the only remaining good copy), then there is a risk that both copies are permanently lost or damaged (red) – i.e. the data object is lost.

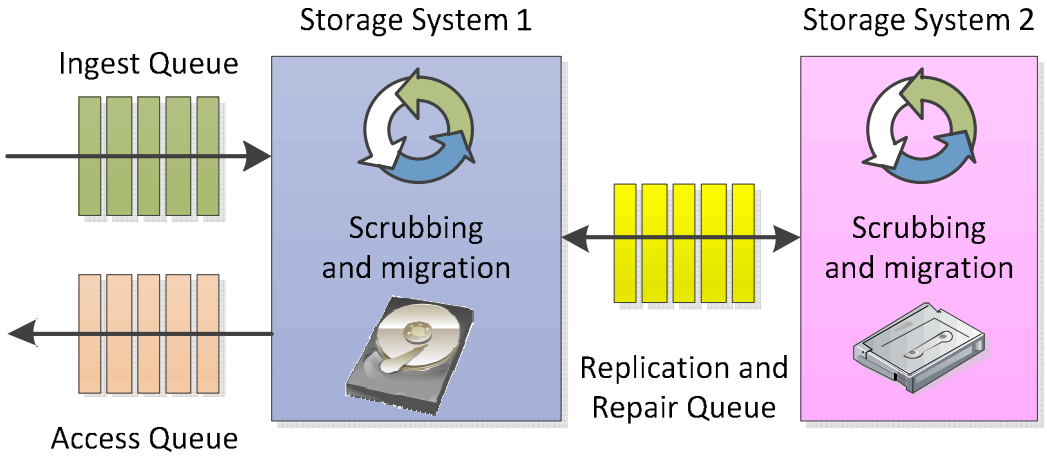

Figure 15 Example storage system configuration

Figure **15** shows an example storage configuration that might underpin the two copy model shown in Figure 14. The configuration consists of two storage systems (on each of which one copy of the data objects are held). Storage System 1 (SS1) is used to handle ingest and access requests and Storage System 2 (SS2) is used to provide a second safety copy of the data objects. Data objects are replicated from SS1 to SS2. In this configuration, SS2 provides a disaster recovery capability as well as a source of 'good' data to repair any corruptions or loss in Storage System 1 (and vice versa). The ingest, access, replication and repair activities are modelled as queues. Within each Storage System, internal processes check data integrity or perform local migrations, e.g. media or file formats.

Each storage system has a cost associated with its operation, as do all state transitions in Figure 14. In this way, the model allows both the risks and costs to be combined into a single model.

This is of course a simple model and does not consider the case that a corrupted copy can be repaired without needing to resort to accessing the other copy, for example by concealing errors rather than repairing them. Likewise if both copies are damaged, there may be cases where it is possible to use fragments of each to reconstruct a new good copy. This would be a transition from the red circle back to the green circle. It is possible to add these new transitions to the diagram as a refinement. These new transitions would also have new costs associated with them, e.g. the use of an operator or tool to do repair instead of a simple file copy of known good file to replace a known corrupted file. Likewise the model can easily be extended to include more than two copies of each data object. In the simulation tool we have developed, we can model up to 3 separate storage systems each of which can hold 1 or more copies of each data object.

During the simulation files become corrupted as a result of latent (silent) corruption or access corruption (e.g. during file read/write or access). For each corruption type it is possible to define a number of corruption events that are probabilistically triggered by the simulation on a per-tick basis (with a tick typically representing a real-world increment of 1 day). For example, a possible corruption event might be specified as: corruption of a 1Kbyte block with a probability of occurrence of 1 in 10^{10} blocks over 12 months. Assuming a Poisson distribution, this rate is then converted into the probability of corruption on a per-tick basis.

The current model assumes that corruptions are randomly distributed across the data being stored. When corruption takes place, it starts at a randomly selected location within the storage system and damages the data size that was specified by the corruption event blocks described earlier, files are modeled as containing critical section (the size of which can be configured) and a non-critical section. When corruption hits the critical section, then file is considered as not repairable and this triggers a repair process by using one of the other replicas of the file.

On the top of corruption processes, the operation of each storage system includes ingesting new files, providing access to existing files, migrating files and storage, and managing integrity. Since all of these system-level activities consist of more than a single atomic action (e.g. access to an asset includes file integrity check, its potential repair and transcoding before it becomes accessed), they are defined as workflows consisting of series of actions that are, in the end, realized through the execution of storage system services.

Some workflows are relatively simple, for example checking the integrity of a file by reading it, generating a checksum and then comparing that checksum with a reference checksum (Figure 16).

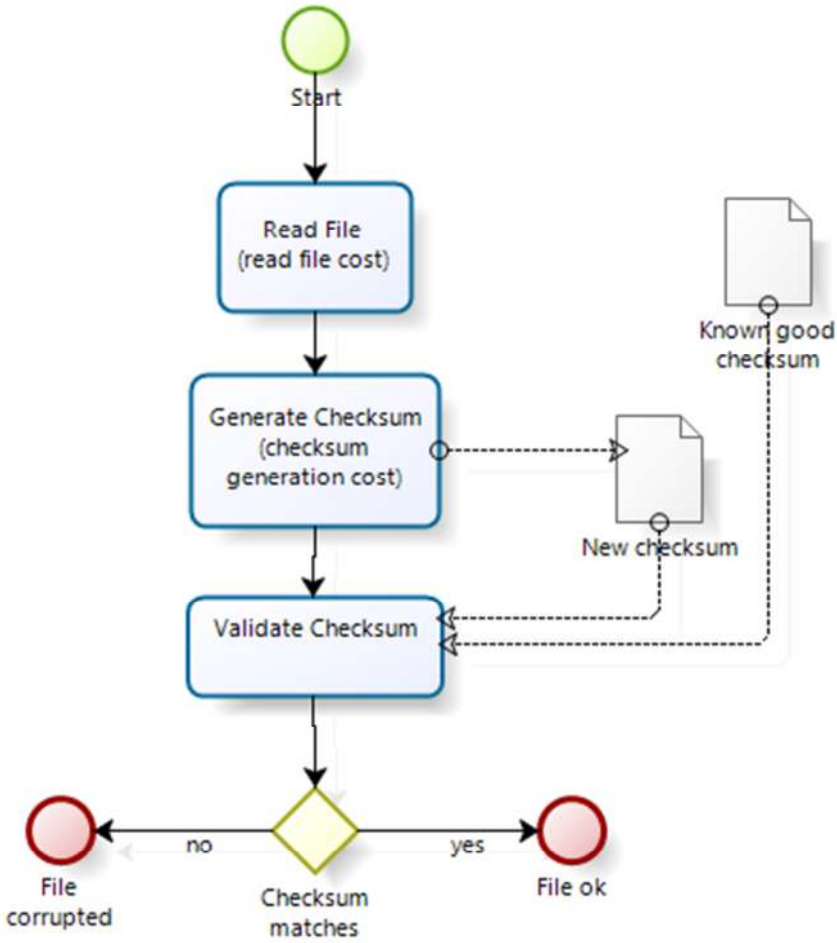

Figure 16 Validate checksum workflow (BPMN notation)

These simple workflows are then incorporated into other processes, for example copying a file which involves a read operation, a write operation, and a check that no integrity has been lost in the process (Figure 17).

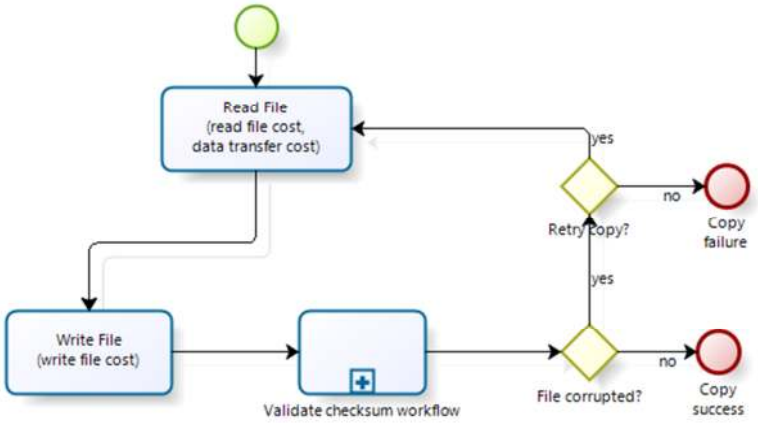

Figure 17 Copy file workflow

Further workflows can then be built up that have increasing complexity, for example the integrity check and repair workflow (Figure 18) that verifies the integrity of the file, and if integrity is lost then performs necessary repair through a copy operation of a known good replica or if unsuccessful then an attempt to do a local repair, and workflow for file format migration (Figure 19 - called transcoding in this case due to origins in work on file format migration for audiovisual content).

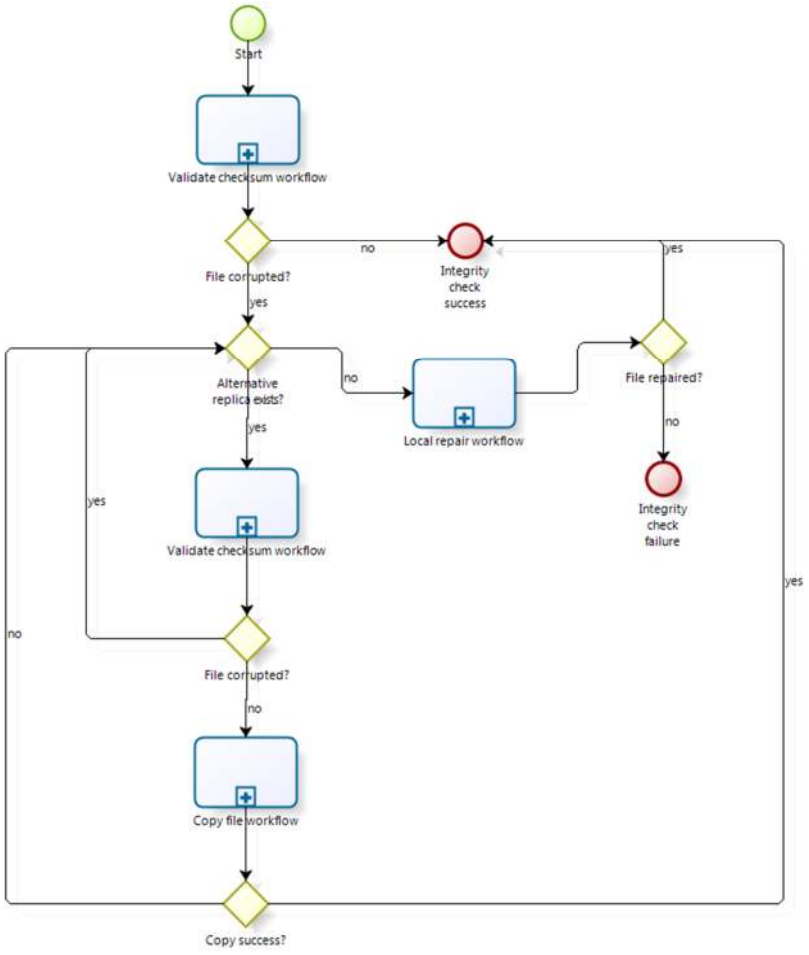

Figure 18 Integrity management workflow

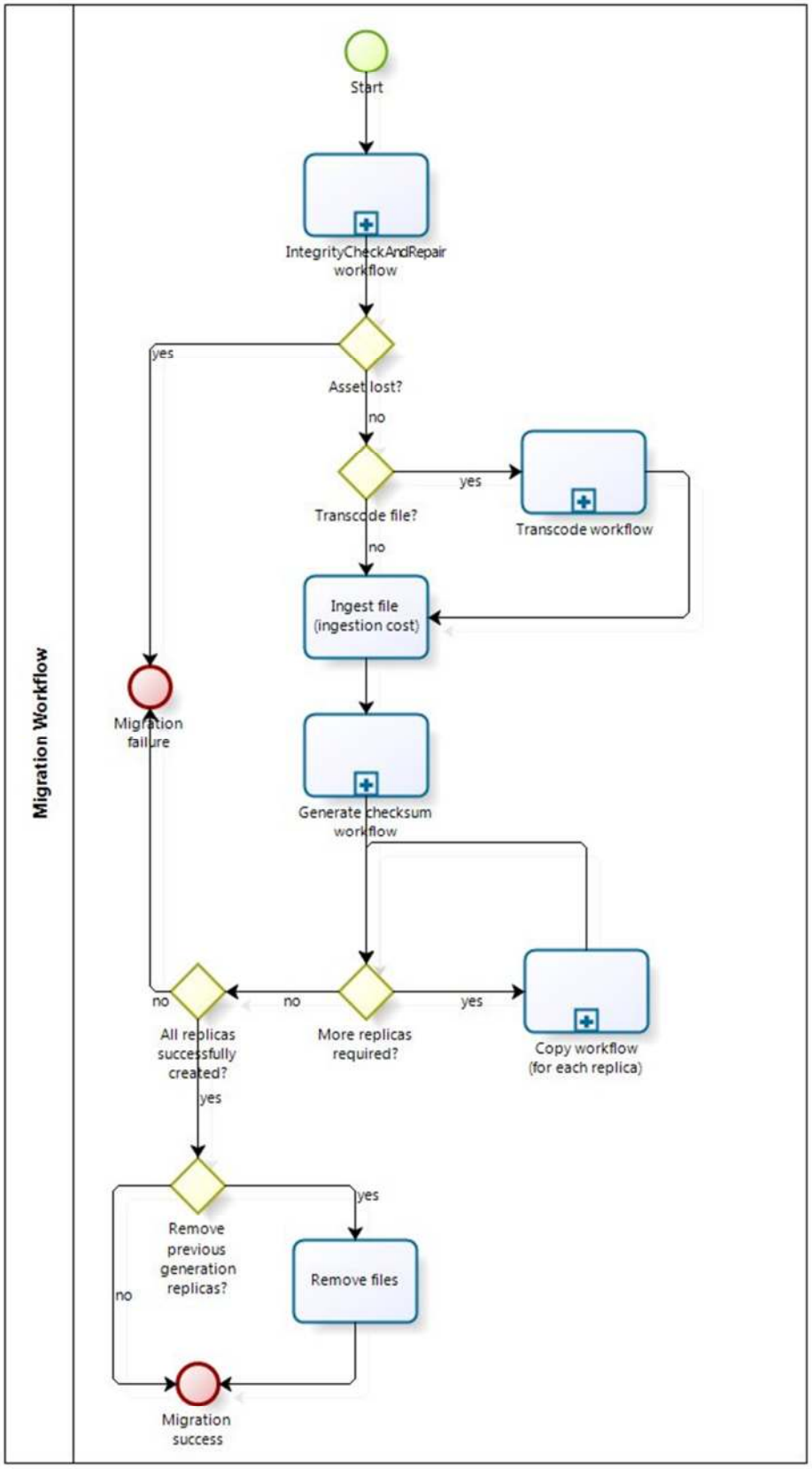

Figure 19 Migration workflow

The ability to break down and express each management activity in the form of a workflow (and its actions) provides several advantages. Firstly, the system-level functionality is broken

down into atomic parts that are realizable through the execution of different services. Since services are resource constrained, this allows the simulation to be used to understand of the impact of under-provisioning of resources. Secondly, since the tool is customized to process workflows, the same system-level functionality can be realized using different set of actions (hence workflows) allowing for a broad customization to real-world archive system examples and their simulation.

Since most of the storage system operations are dependent on the execution of services, another important feature of the simulation tool is the ability to model the impact of resource constraints on the efficiency of the system. This can be achieved by setting up a specific allocation of resources to individual services that remains unchanged during the simulation or to allow different services to consume shared pool of resources. In the latter case, the model provides two simple resource allocation algorithms (round robin and greedy) that control the access to limited resources. These algorithms can be further extended to address more realistic allocation strategies aiming to optimize usage of resources without system performance loss.

7.2 Examples

7.2.1 Case study: the cost of risk of loss

The following case study presents an example of the use of the simulation tool to perform repeated simulations and to visualize the results set in order to estimate the long-term costs and risks of archiving digital content.

In order to make investment decisions, an archive requires basic actuarial information for competing preservation strategies. The results should indicate a cost curve for the percentage of loss of archive content over a significant period, including the probability of loss, the size of the uncertainties, and a cost function to show how the probability of loss varies against increase – or decrease – in investment. The following describes an approach using discrete event simulation to provide quantified costs, risks and uncertainties for long-term storage of file-based assets using IT storage technology.

The interactive storage simulation tool described above allows a user to manipulate a storage model in order to observe the effects of changing the storage strategy on cost and on the risk of loss of assets. The tool can also be used to batch process a number of parameterized configurations in order to explore the space of possible storage strategies. Given the results of this, it is possible to compare directly the effect of, for example, keeping a greater number of replicas of each asset while scrubbing the files less frequently.

The storage simulation tool uses a stochastic simulation; each time it is run using the same initial configuration the results may vary in terms of the number of files lost and the total running cost. By repeatedly running the tool using the same configuration, it is possible to generate a probability distribution of asset loss. Figure 20 is a probability distribution function (PDF) that was generated by sampling the model 1000 times, each time simulating 10 years of preservation activity, and indicates that we would expect to lose around 0.3% of the archive over 10 years (or around 75 assets for an archive of 25,000 assets).

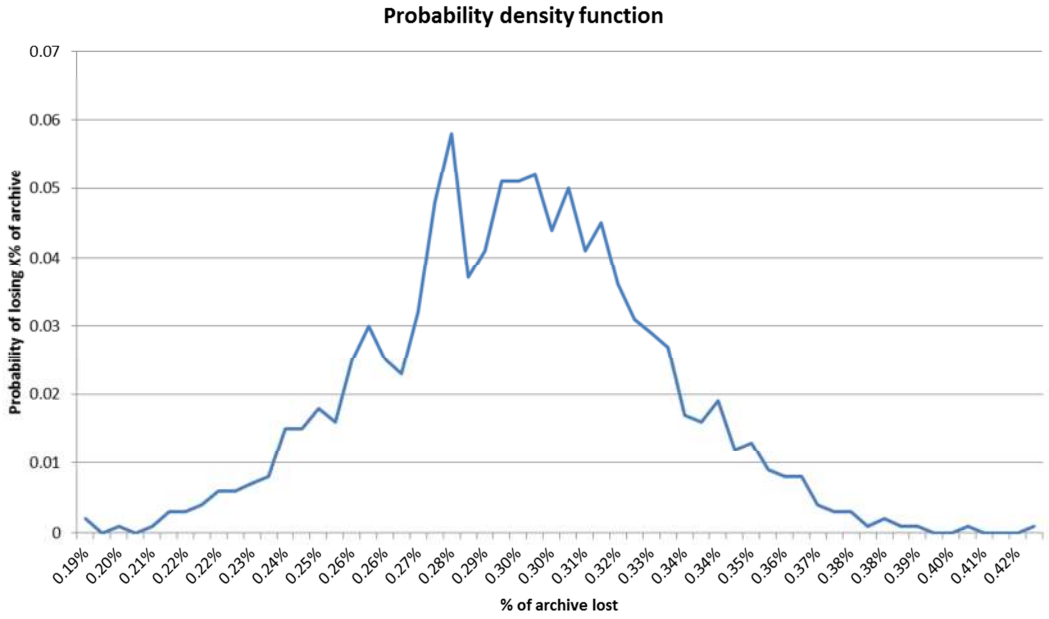

Figure 20 Probability density function

The PDF indicates the probability of losing exactly a given percentage of the archive. By cumulatively summing the probabilities up to a given percentage of archive loss, we can generate a cumulative distribution function (CDF) of the PDF. The CDF gives the probability of losing this amount of the archive or less. However, the probability of losing a given amount of the archive or more is the information that is of greater interest to the AV archive community. This can be found by taking the complement of the CDF, shown in Figure 21. The probability of losing more than a very small amount of the archive is high, while losing more than large amounts of the archive is low.

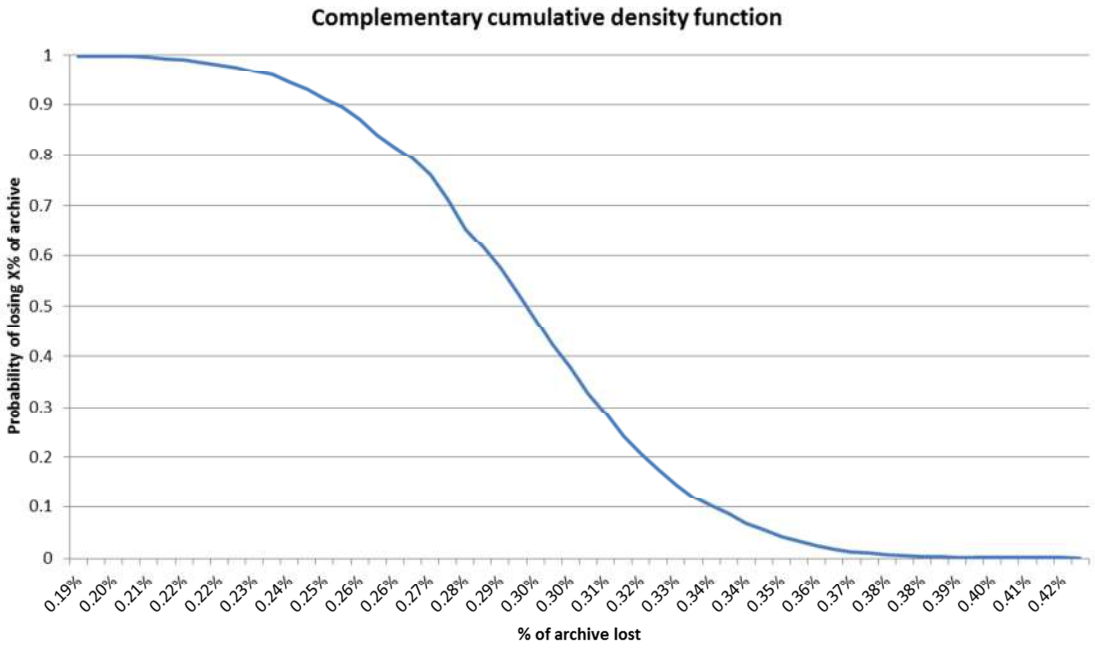

Figure 21 Complementary cumulative density function

Figure 22 shows a two-dimensional representation of the multi-dimensional data produced by the process outlined above. This shows a single storage system where the number of additional copies of a file stored in that system and frequency of integrity checking (scrubbing) have an impact on both the cost and the risk of file loss. The figure illustrates the risk and cost landscape for the loss of more than a specified percentage of the archive's assets (the acceptable maximum level of loss). The boundary between adjacent coloured bands represents configurations of equal cost. The white contour lines are lines of equal risk of loss. Each intersection of values from the X and Y axes represents a storage simulation that was executed (multiple times) using the storage simulation tool. The intervening values are interpolated.

The type of visualization described here helps AV archive decision makers to identify the optimal storage strategy given their constraints. Firstly, given a fixed budget, it enables them to select the storage strategy with the lowest probability of asset loss. For example, for a given budget of 50 million (Euros), Figure 22 indicates that the strategy with the lowest probability of loss of more than 0.1% of the archive over 10 years is to keep 3 additional copies of each asset and to check the integrity of the files every 10 months. In this case, it is not cost efficient to increase the frequency of scrubbing, as it will cost more but is unlikely to deliver any benefit in terms of data safety. Similarly, on the assumption that the risk of losing 0.1% of the archive over 10 years with a probability of 1 in 5 is considered acceptable, then the strategy with the lowest cost is to keep 3 additional copies of each asset and to check the integrity of the files every 12 months.

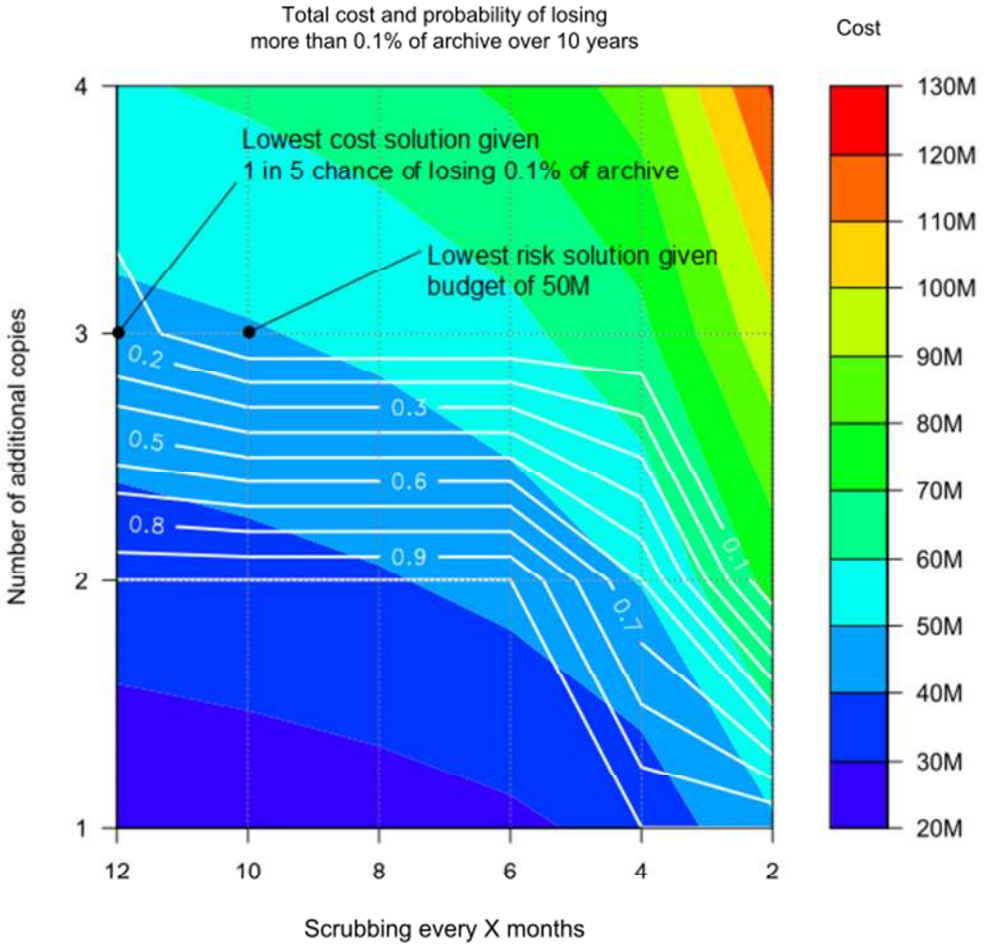

Figure 22 Cost of risk map

7.2.2 Case study: the forever cost of storage

In this section we present an example of long-term cost modelling. A simulation of the cumulative cost of running a storage system over a 15 year period is shown in Figure 23. The simulation is for two storage systems with one copy of the data on each. The first storage system models a HDD server using a simple JBOD (Just a Bunch of Disks) approach with a 3 year migration cycle, costs that halve every 3 years, and annual data scrubbing. The second storage system models an offline second copy of the data on data tape stored on shelves with a 4 year migration cycle (for example 2 generations on the LTO roadmap), and the cost of storage halving every 2 years (but this only gets sampled every 4 years during a migration). For both storage systems the capex cost is amortized over the full life of the storage system. The reduction in gradient of the cost curve every three years reflects the fall in cost of storage and is dominated by the cost of the HDD servers. The vertical jumps correspond to scrubbing or migration operations. They have been artificially compressed in time to make them more visible whereas in reality these tasks might take many months to complete. The different periodicities of scrubbing and migration are reflected in the different height of the jumps, with date tape migrations having a significant effect every 4 years as it is modelled as a manual process. The cost curve can be seen to asymptotically approach a limit that is approximately 5 times the cost of the system at the end of year 1. In other words the 'forever' cost of storage is approximately 5 times the year 1 cost and includes all necessary migrations and data integrity management activities. Approximately 40% of the long-term TCO is scrubbing and migration, which includes integrity repairs of data corruption (mostly occurring in the HDD system due to the JBOD configuration) and the labour costs of handling data tapes that arise from a 'store on shelves' model. If more resilient and automated storage systems had been used (e.g. RAID HDD array and data tape library) the fraction of TCO for scrubbing and data tape migration would have been lower, but of course the cost of storage itself higher. The important point is that the cost of all activities required for long-term data retention need to be included in a cost model to get a full picture of the long-term costs. For simplicity no data is added or removed from the archive and no data access takes place. These can of course be added to the simulation.

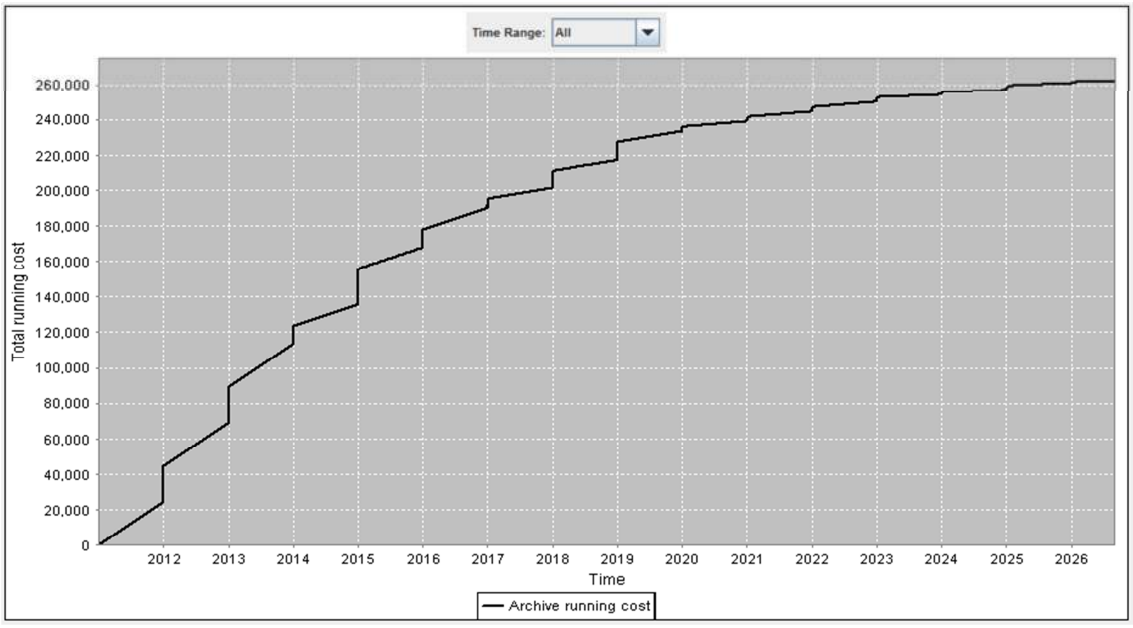

Figure 23 Simulation of the cumulative cost of a 2 copy storage strategy over a 15 year period