Journal of Data, Information and Management (2022) 4:107–117
https://doi.org/10.1007/s42488-022-00068-4

ORIGINAL ARTICLE

A system for organizing, collecting, and presenting open-source intelligence

Saiful Khan[1] · **David Wallom**[1]

Received: 31 March 2022 / Accepted: 14 April 2022 / Published online: 8 June 2022

Abstract
Open-source intelligence is a rapidly expanding area of the security and intelligence industry, involving the collection of internet located open data from various sources, turning that data into actionable intelligence, which is reused where possible and relevant. While creating or processing the raw input data capturing and managing the corresponding provenance information, e.g., workflow, state, raw evidence, reports, and summaries, that simplifies its retrieval and reuse is essential. In comparison, scientific workflows and tools that support them are routinely used in the majority of academic research disciplines, managing diverse sets of data resources and their provenance. Based on the techniques established within the academic community, we have developed a system for managing this open-source intelligence data and associated provenance information. This will enhance the efficiency of retrieving stored data products and reusing them to support intelligence-led security decision-making. The open-source intelligence company partnered within this project has an operational envelope that includes collecting and analyzing personal subject information. Therefore, they must understand the scope of their data holdings appropriately, especially in light of obligations under the General Data Protection Regulation. The system developed allows for tracking requests for intelligence products, ownership of the collection, analysis and generation of intelligence briefs, and tracking the delivery of those final products to the customer for future billing. This adds further layers of efficiency to operations and hence reduces the costs of producing intelligence products.

Keywords Information management · Business intelligence · Data quality · Data provenance · Knowledge-base construction · Workflow management

1 Introduction

The research lifecycle is used to describe the process of research, development, planning, implementation, publication, reuse, and is used as a reference where different tools and capabilities are built or available to support this process (Demchenko et al. 2013) . Within the intelligence community, there is also a well known and well-understood cycle describing the process of intelligence capture and utilization, the intelligence cycle, direction, collection, analysis, dissemination (UK Ministry of Defence 2011). It is clear that these cycles are in many ways interchangeable. Therefore a company operating within the security and intelligence domain looking to make step changes in its operational and information management capabilities could reasonably consider the best practice within the academic research domain and hence the development or availability of tools from this domain.

Within academic research, there is a developing consensus on how research can be improved through the paradigm of open data. A set of principles has increasingly described this, that data should be FAIR (Wilkinson et al. 2016), with one of the stated goals being to enhance the reusability of data. This is an acronym for the following;

1. *Findable* – Data and supplementary materials have sufficiently rich metadata and a unique and persistent identifier.
2. *Accessible* – Metadata and data are understandable to humans and machines. Data is deposited in a trusted repository.

✉ Saiful Khan
saiful.khan@eng.ox.ac.uk

David Wallom
david.wallom@oerc.ox.ac.uk

[1] Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK

3. *Interoperable* – Metadata use a formal, accessible, shared, and broadly applicable language for knowledge representation.
4. *Reusable* – Data and collections have a clear usage licenses and provide accurate information on provenance.

This guiding principle to enhance reuse means that the FAIR principles are equally at home in the commercial sector as the academic one, particularly where reuse and understanding of data holdings can bring about competitive business advantage.

A crucial part of achieving these benefits is understanding that it is essential to start when good data management practice is the norm. We must also remember that this applies to all types of data, not only the static and archived but also the active and in current use. An even higher priority is ensuring that metadata associated with the raw data is captured and curated and the point of creation of the data object to ensure that the reason, method and ownership of data is recorded, ensuring management of provenance. Tools and techniques to do this have emerged in many different disciplines in terms of activity tracking, task allocation, and data management.

For the management of data and the recording of operations on it, electronic lab books have been developed within the wet lab-based sciences. These may be commercial off the shelf products or open-source tools (Trefethen et al. 2012), developed as a method of capturing within the research environment the raw data generated within an experiment, as well as the protocols that were used to generate and then analyze the data to convert it into actionable information. However, these are not generally configured to support the allocation of work; just the recording of work that has occurred. It is therefore essential to consider this functionality.

Within the IT industry, issue ticket systems (e.g., Trac 2021, JIRA (Atlassian 2021)) have been established for some time as a method to allow the management of issues, allocation of resources to work on those issues and verification of activities on those items. There have been many different use cases where these systems have been used to facilitate tracking. However, until now, there has not been an example where the functionality of an issue ticket system is extended to support the complete management of a company's business process from initiation and recording of a customer order, the capture of activities to fulfill the request, the management of the data used within the order, any analysis performed on that data and the collected output product within the single system.

Within the partner business for this project, Horus Security Consultancy, the traditional intelligence cycle is followed, but alongside this is the management of the day-to-day responsibility of fulfilling a particular customer's requirements. Therefore a mix of an electronic lab book system and an issue ticketing system is necessary. Within this paper in Section 2 we discuss the current standard practices for data management within the company as a set of requirements and benchmarks in capability we must exceed, Section 3. This is then followed in Sections 4 and 5 with a discussion of the design and performance of the tool that we developed. In Sections 6, we discuss how this has facilitated operational enhancement and business process change, particularly in the relationship to how work is shared between the different geographically distributed offices.

2 Standard practices with data

The partner company's operational workflow is divided between multiple groups, with the core teams being Intelligence, Investigations, and Screening. These teams are however united by a common workflow, which typically has five main stages:

1. Request submitted by client,
2. Initial collection of raw data by researchers,
3. Subsequent analysis and preparation of reports by analysts,
4. Dissemination to client, and
5. Generation of invoice.

The **Intelligence** team's focus is on 24/7 research and raw intelligence collection. The targets for the data collection include a diverse array of geographical contexts, situations, and groups that a client may be interested in. For each target set, data collection involves navigation across social media, blogs and news websites, and an array of tools are used for searching and regular monitoring of specific websites. The relevant content is inspected and harvested manually, and saved as evidence on a network filesystem. Complex bespoke file naming conventions are used at multiple stages in the workflow to keep track of document dates, source, provenance, and other attributes. Spreadsheets are maintained to facilitate collaborative work and to store the provenance information.

The output from the data collection step include flash 'alerts' and daily summary reports of salient updates. These initial products are then passed to the analysts, and also transformed into reporting tables which contain free-form text and various tags, labels, and attributes (e.g. times, types, themes) from a predefined list. The reporting tables are open-ended and incremental. The main outputs from the analysts are daily and weekly digest documents, bespoke statistical work, and longer threat assessments. All reports are emailed to the relevant clients as a PDF. They are saved in local directories using complex bespoke file naming

conventions. Overall a highly manual and specialized configuration that has no ability to enable versioning of data contained within or record who has the current ownership of any one piece of the information management process leading to the possibility of clashes without more manual process instigated.

The **Investigations** team conducts more focused research projects, centered on products and companies as clients. As with intelligence, the search seeds for data collection are open source websites and emerging links that point to a specific target individual or group, including material contained within the so called dark-web. All identified evidence is inspected and harvested manually, and persisted as PDF documents which are saved locally. The initial output is similarly a client specific daily report and emergency email reports. These are followed by more in-depth analytical and investigatory reports. All output is published as PDF or as emails.

The **Screening** team conducts pre-employment checks such as fact checking activities including general searches, due diligence, financial and criminal history checks, and information gathering around potential derogatory indicators, such as unprofessional conduct.

The team uses various tools, again centered around open source intelligence collection. A bespoke in house developed software platform (HOSS) is used to (i) capture information and log new screening requests from clients, (ii) catalog supporting evidence (e.g. photos, forms) for a case, (iii) keep track of individual screening cases, (iv) maintain statistics across the team, (v) auto generate and auto fill some parts of the reports, and (vi) to securely publish the results to the client. HOSS involves multiple SQL databases, is hosted on Microsoft Azure, and exposes a number of simple user interfaces for internal analyst facing and external client facing purposes.

Regarding information searches, there are small number of fixed, predefined search criteria, whilst the majority of activities with each screening case involve unique criteria for the individual to be screened. The inbound work queue for the screening cases to be completed is maintained manually in an MS Excel spreadsheet. All evidence and information is saved as PDF files and saved in local directories using complex bespoke file naming conventions to keep track of document dates, provenance, and other attributes.

2.1 Business practices driven by data usage methods

Horus's open source intelligence gathering, analysis, and dissemination workflow is a manual process. There is a considerable amount of duplication of both data collection and report generation.

Data and evidence related to a single task is often collected by multiple researchers. The various spreadsheets used for process management within the group are broadly similar. Spreadsheets limit this task scheduling to a sequential process as parallel collaboration is rendered difficult by tasking using a shared spreadsheet. Even though data is searchable and locatable via the filesystem search index, there is a significant chance that PDFs will be mislabeled and misplaced in a complex and cluttered directory structure. This often leads to duplication and difficulty locating existing data holdings and indirectly data loss.

Just as importantly, there is no means to capture the workflow and process data generated during the life-cycle of a task. This has important implications for GDPR compliance.

2.2 Areas for improvement

Workflow analysis points to a number of areas where improvements will easily lead to increased efficiency, more frequent reuse of collected raw data, enhanced security and privacy, and better ability to comply with regulation such as GDPR. Fully automating the entire workflow is not possible due to the absolute precision that clients demand and the level of human input required for analysis and insight. Improvements in workflow and process should also fit the current workflow, processes, and general operational restrictions and dependencies.

Workflow management – controls and supports operational processes through the automation and management of company workflow by capturing and analyzing associated data and provenance in its lifecycle. Horus is powered by locally hosted Windows servers and desktops. None of the parts of the workflow use any centralized database or services to capture and store the workflow data. However, many of the repetitive and manual aspects of the workflow can be streamlined and optimized. The repetitive nature of data collection can be minimized and reuse of data is possible by effective collaboration through a centralized tasking system and searchable task database.

Document management and search – is an automated or semi-automated way of managing, securing, searching, and reusing documents. Horus uses a fileserver to manually store and organize documents in folders, and Windows search to retrieve the documents. Complex bespoke file naming conventions are used at multiple stages in the workflow to keep track of document dates, provenance, and other attributes. Access to the file server is restricted through Microsoft Active Directory and individually assigned privileges. Windows search supports full-text search limited to certain file types. Search results are often not relevant as filesystem search does not rank the relevance of files and does not support filtering.

3 Related work

Relational databases are the most popular data management system used to store structured data in a relational data model. However, a large volume of data being generated are semi-structured and unstructured, e.g., workflow, provenance, documents, files, and meta-data. As a result, there have been numerous attempts, challenges, and solutions discussed in the literature (e.g., Rejeb et al. 2022) to create a different system to model, store and efficiently search such unstructured data.

3.1 Workflow management

Systematically captured and stored workflow data can support mechanisms to ensure the quality and validity of data that can be examined or reviewed in the context of its source and transformations over time (Khan et al. 2016; Kruschwitz and Hull 2017). Workflow data allows users to track provenance information such as evolution. Workflow management systems formalize and structure complex and distributed organization specific processes and data.

There are many different generic workflow tools available, though these would require significant customization to be useful in the scenario that we describe. Therefore we consider a subset of these tools, issue trackers, nominally used in help desks and other task tracking applications. Systems such as these, e.g., Atlassian Jira can model both control-flow and data-flow. Control-flow describes and models the state of tasks, data-flow describes and models the information storage and exchange between tasks (Van Der Aalst et al. 2016). A significant bottleneck of such a system is scalability and searchability, e.g., storing many files and documents and searching inside those documents.

3.2 Document management

This section will describe the primary data models, databases, and content management systems used for managing files and documents.

Relational Data Model – The relational data model uses predefined tables and schema to define the tables' structure and relationships between them. A table defines named columns and the type of data that can be stored in each column. The relational data model's central concept is that similar kinds of data is stored in the rows of the same table with a unique primary key. A primary key and a foreign key are used to group the instances where two tables are related. A Data Definition Language (DDL) is used to define the tables Data Modification Language (DML) (e.g., SQL) to store and manipulate data. There are hundreds of commercial and open-source relational databases available.

The relational model is simple, highly scalable and supports ACID (Hogan 2018) properties. However, in a relational model, the records are stored in a predefined schema or format; changing or updating that schema is very expensive. In particular, there must be an understanding of the effect on existing data within the table. At present, in addition to structured data a semi-structured, unstructured and files are being generated.

Document-oriented Data Model – Document-orientated databases are another most suitable for document management. Such databases use a document-orientated data model that follows no internal structure, i.e., the fields and relations do not exist as predefined concepts. It acquires the type of information from the data itself. All data attributes for an object are placed in a single document (or JSON-like object) and are stored as a single entry. A relationship between the two entities is created using references. Typically all related information is stored together, and it allows every instance of data to be different from any other. Hence, creating a data model that is flexible to future changes. A popular document-oriented databases is MongoDB.

A workflow management system and a document-oriented database can model and manage complex workflow, provenance, and files. However, there are certain limitations to storing files containing a large amount of unstructured data and searching through these data. For example, MongoDB is a document-oriented database that does not support storing files that exceed the size limit of 16 MB and provide a limited full-text search capability. Therefore, specialized file systems such as MongoDB GridFS are used to store large files, and a specialized search engine is required to support the full-text search of those files.

3.3 Enterprise search engine

Commercial (e.g., Microsoft SharePoint, Asite system 2021, Bentley's eB Insight 2021) and open-source (e.g., Alfresco 2021) tools are commonly used in enterprises for document management and search. Such document databases provide limited full-text search capabilities. Enterprise search engines are specialized for indexing and searching over large-scale structured, semi-structured, and unstructured data and documents. Many open-source and commercial enterprise search engines are available, and most of them primarily use Apache Lucene or Lucene-like inverted index under the hood.

The system uses a workflow management system to support workflow and provenance data, a document database accompanying a filesystem for managing large-scale documents and files and their metadata, and an enterprise search engine to search and reuse.

4 Designing a hub for information management

The main design criteria for developing a system are:

1. Ability to capture and store workflow and provenance data,
2. Store large scale files and documents,
3. Support free-text search and reuse of quality data,
4. Integrate all the components or sub-systems to enable automatic control-flow and data-flow between the sub-systems, and
5. Support a seamless workflow and operation.

It was aimed that we would use as many open source tools as possible, minimizing the number of proprietary components and developed a number of microservices to implement missing functionality and integrate components of the system. Figure 1 illustrates an abstraction of the entire architecture of the system. The system consists of seven major components as shown in Fig. 1:

1. User Interfaces (UIs),
2. API Gateway and Microservices,
3. Company Workflow Management (CWM),
4. Document Management System (DMS),
5. Enterprise Search Engine (ESE),
6. Monitoring and Analytics, and
7. Backup and Replication,

4.1 User interfaces

There are three major UIs, shown in Fig. 1, for workflow management, document management and monitoring data visualization. The Fig. 2 illustrates an example workflow using the most significant parts of the UIs. Analysts use the CWM UI to file an issue ticket and client information and relevant data and provenance. The DMS UI is mainly used for managing and searching files and documents. IT administrator mainly uses the monitoring and analytics to monitor the system's health and usage.

4.2 API gateway and microservices

In a microservices architecture, clients usually require access to multiple functionalities from more than one microservice. An API gateway or an intermediate indirection is a service that provides a single-entry point for certain groups of microservices. It acts as a reverse proxy to handle routing requests and provides additional cross-cutting features such as authentication, SSL termination, and cache. We implemented the API gateway using NGINX (server and load balancer) creating an interface between the UIs, external services and the microservices.

The microservice layer implements multiple RESTful API services to read, write, update and delete data. The APIs are guarded by an identity provider service that keeps

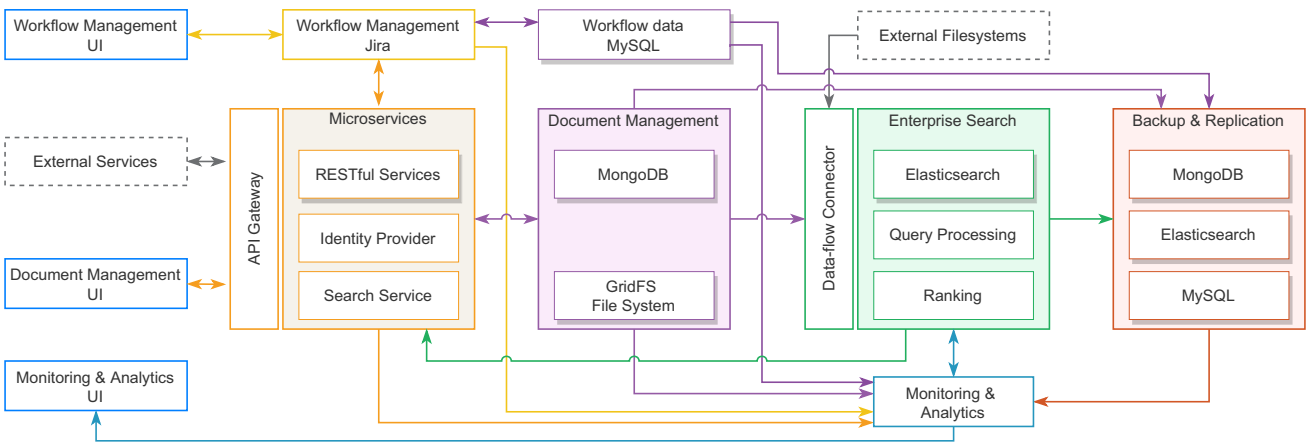


Fig. 1 The architecture of the system. Analysts interact with the system using three user interfaces, e.g., workflow management, document management and search, and monitoring and analytics. Each UI is supported by back-end services, e.g., company workflow management for supporting workflow and provenance, back-end microservices for document management system. Services use databases to maintain data persistently, e.g., workflow data is saved to MySQL; files and documents in MongoDB and GridFS. A data-flow connector agent periodically scans the MongoDB database operation-log for creating and updating the enterprise search engine index implemented using Elasticsearch. The Elasticsearch also stores the analytics and monitoring log, external network file system data to make them searchable. All data is replicated and backups are created. An API Gateway allows other services to consume data from the system

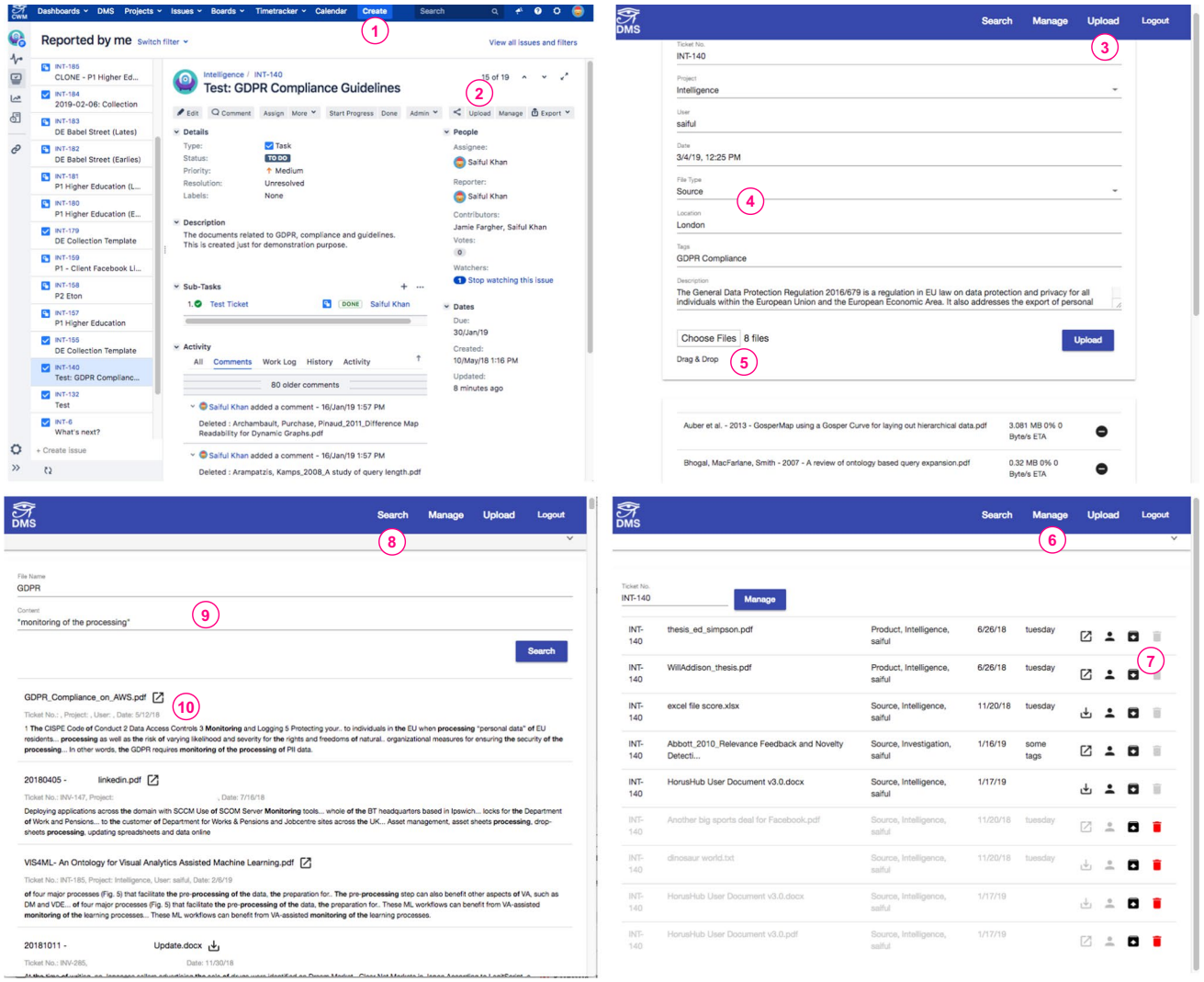


Fig. 2 An example workflow. An analyst creates (1) an issue ticket at CWM. The analysts collect data and create summary reports related to the ticket and upload (2). An upload interface (3) allows the analyst to add additional metadata (4) before uploading (5) the files and documents to the DMS in a batch. The uploaded data and reports can be managed via a DMS management interface (6). The management interface provides different actions (7) (e.g., archive, delete, set as private) that can be performed on the uploaded data. The analysts use the search interface (8) of the DMS to search (9) files and documents uploaded to the DMS. The search results (10) are presented intuitively, e.g., the metadata are shown, the matching texts are highlighted to help the analysts to filter the relevant results quickly

the authentication and authorization policies. For example, different types of data should have different rules for how it should be processed and managed to comply with GDPR (Van Loenen et al. 2016). Essential or more general data can be open within the company. In contrast, sensitive or personally identifiable data require more security and relevant user access. A Rule-based Authentication Control (RBAC) design pattern manages and enacts predefined policies created by a system administrator. The system uses single sign-on (SSO) to authenticate users and to inherit the predefined access policy of relevance to the specific user.

4.3 Company workflow management

Domain-specific workflow management is a well-researched discipline with a well-established set of principles and software. In order to support the five workflow states described in Section 2 and to capture related data and provenance, the CWM implements the following functionalities:

1. *control-flow* for management of activities,
2. *resource* for modeling groups, projects, roles, and authorizations,

3. *data or artifact* for modeling decisions, data creation, forms, documents,
4. *time* for management of duration's, deadlines, time spent on tasks for invoice and appraisals,
5. *function* for describing activities and related applications,
6. *extensibility* for enhancing the functionalities through software patches or plugins, and
7. *interoperability* to connect with external tools and services via REST APIs.

We customized Atlassian Jira (2021) to model the above mentioned functionalities. Among these, the most relevant is that although Jira supports file attachment, it has certain limitations or missing functionality such as capturing file metadata, managing large files, and indexing contents. We developed a Jira plugin to allow data and document flow between CWM and a specially designed DMS (described in Section 4.4) to solve these challenges. The plugin provides options via a UI widget to connect to the document management system for uploading files. Each ticket created in CWM is assigned a unique operational identifier. The unique operational identifier assigned to the ticket is used to link related information stored in different subsystems of the system.

4.4 Document management system

We designed a document management system to handle diverse, fast-growing, and large files and documents. We choose MongoDB, a NoSQL database, as a core database for the DMS. To store large documents and achieve horizontal scalability, we must decide whether to store the binary data in a separate repository or filesystem or with the metadata. Therefore, we used GridFS to support storing and retrieving files that exceed the document size limit of 16 MB. When files related to a task of CWM are uploaded, they are transferred to the DMS using an encrypted token. The DMS writes the metadata to MongoDB collection and files to GridFS collection as binary data.

MongoDB, however, does not support the full-text search for searching inside stored files. Therefore a dedicated document search engine is developed to support full-text search described in the next section.

4.5 Enterprise search engine

We deployed an Elasticsearch ESE to facilitate free-text search across all primary services within system, such as CWM, DMS, and external file-systems. We implemented a data-flow connector to push data from the primary services into the ESE and make them searchable. The data-flow connector internally uses two tools Elasticsearch (a) ingest-attachment and (b) FS Crawler.

The ingest-attachment plugin extract content from different types of files and documents. This plugin internally uses the Apache Tika content analysis tool to extract file content and submit the parsed content to the search engine. The FS Crawler indexes the files and documents archived in the external file-systems. While (b) is configured to scheduled crawling and refresh of the index according to predefined time intervals, (a) does that in real-time to support effective collaboration among analysts and to reduce duplication during data collection.

The search results are ranked based on search context. A search interface provides an additional option to filter search results. When analysts look for files, their knowledge about various properties of files and contexts plays an important role. The search queries are enriched to expand the search coverage, and the retrieved files are ranked based on their relevance. When search happens regularly, we can capture knowledge (search query and relevance feedback) and improve the search results' relevance. A detailed description of a knowledge management system work can be found in (Khan 2015).

4.6 Monitoring and analytics

The use of microservices requires changing the approach to software management, specifically how an organization handles monitoring infrastructure, applications, and data. It is also essential to capture and understand microservices performance, scalability, security and troubleshoot any problems. Microservices generate events (e.g., access log, errors, debug information, and so on), and those logs provide all the information needed for monitoring, maintenance and debugging of the application. We implemented a transport layer in the microservices to push the log information to the search engine via Elasticsearch Logstash.

We deployed Elasticsearch Metricbeat to periodically collect metrics from the operating systems, databases, servers and send them to the ESE. The events and analytics are visualized and monitored using dashboards built by Elasticsearch Kibana.

4.7 Backup and replication

We backup data at predefined time intervals to ensure compliance and granular recovery of the data if needed. We also synchronously replicate databases in a secondary server to ensure quick resumption of operation in any database or machine outage. MySQL and MongoDB both provide built-in mechanisms to design replication of these databases. We configured replication functions to replicate both workflow data uploaded to the MySQL database and the files stored in MongoDB and GridFS.

5 Results and discussion

We analysed the last ten years data collected and stored at the company. We found that the data collected from open source websites are mainly stored in text, PDF, Microsoft Word, and image format. The final reports are produced in Microsoft Word and PDF format. A small number of audio and video files are collected. We also found that 4% files are less than 1KB, 80% files are between 1KB-1MB, 15% files are between 1-60MB, and the remaining 1% files are greater than 60MB.

The files are accessed only when the analysts require to prepare a new report or edit an existing report. Therefore, the ingress or upload operation is more frequent and expensive than egress.

In the following section, we will report performance, measures the system's ability to provide a specific response time to complete an ingress operation. After that, we report scalability, measures the system's ability to increase performance by adding additional resources.

5.1 Performance

Based on the analysis, we set up our test environment and the parameters; for example, we used a size between 1-100MB to test the ingress operation's performance. We used three physical servers to generate parallel requests. We evaluated two matrices to measure the performance:

1. *Response time vs. uploaded file size* – captures the time taken by the API service to upload files of size 1-100MB and
2. *Response time vs. percentage of completed requests* – captures the time taken by the API service to complete percentage of ingress operation.

The Fig. 3(a) shows that the response time of the the ingress operation increases linearly. The average response time and the median response time overlaps, confirming the response time follows a normal distribution. The Fig. 3(b) shows that almost all the request completes within a specific time frame, that is, 99% requests complete at the same time, and around 1% files take slightly more time to complete.

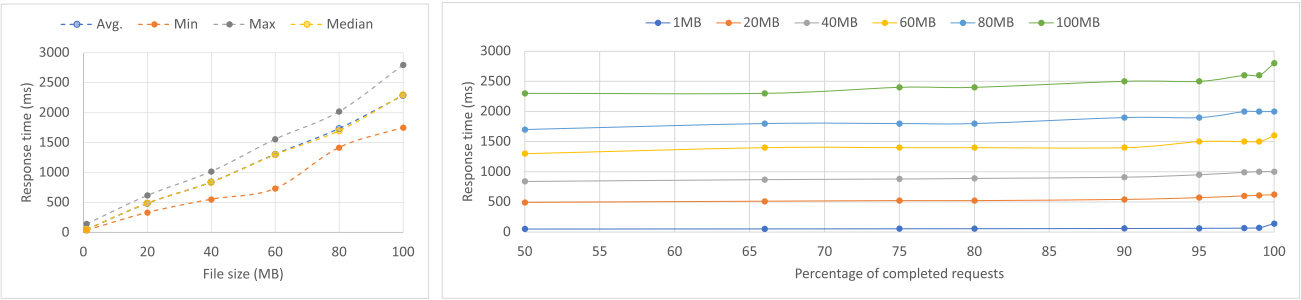
5.2 Scalability

In this experiment, we used three physical machines; at each machine, we created parallel clients, and each client generated parallel requests to the ingress API. Each iteration runs for an hour, and we aggregated the results of all iterations. Figure 4(a) shows ten iterations. For example, we created 50 parallel clients in one machine during the 1st iteration, and during the 10th iteration, we created 166, 166, and 168 parallel clients from three servers.

Figure 4(b) illustrates the total number of parallel requests sent to the APIs during the hour-long test. Figure 4(c) shows the number of requests sent per second. We used a fixed 5MB file in this study. We observed that when the number of request per second increases beyond ˜55, the traffic shaping policy kicks off to limit the traffic. For that reason, when client size goes above ˜400, there is a sudden drop in total requests and the rate at which requests are generated. We test two parameters to measure the scalability:

1. *Response time vs. number of clients* – captures the time taken by the API to complete the increases operations and
2. *Response time vs. percentage of completed requests* – captures the time taken by the API to complete percentage of increases operations.

The results, figure , show that the response time increases linearly. We created not more than 200 parallel clients at each server, figure . The response time improves when we start to distribute the clients among multiple physical machines.



(a) (b)

Fig. 3 Performance of the system: (**a**) Response time vs. uploaded file size, (**b**) Response time vs. percentage of completed operations

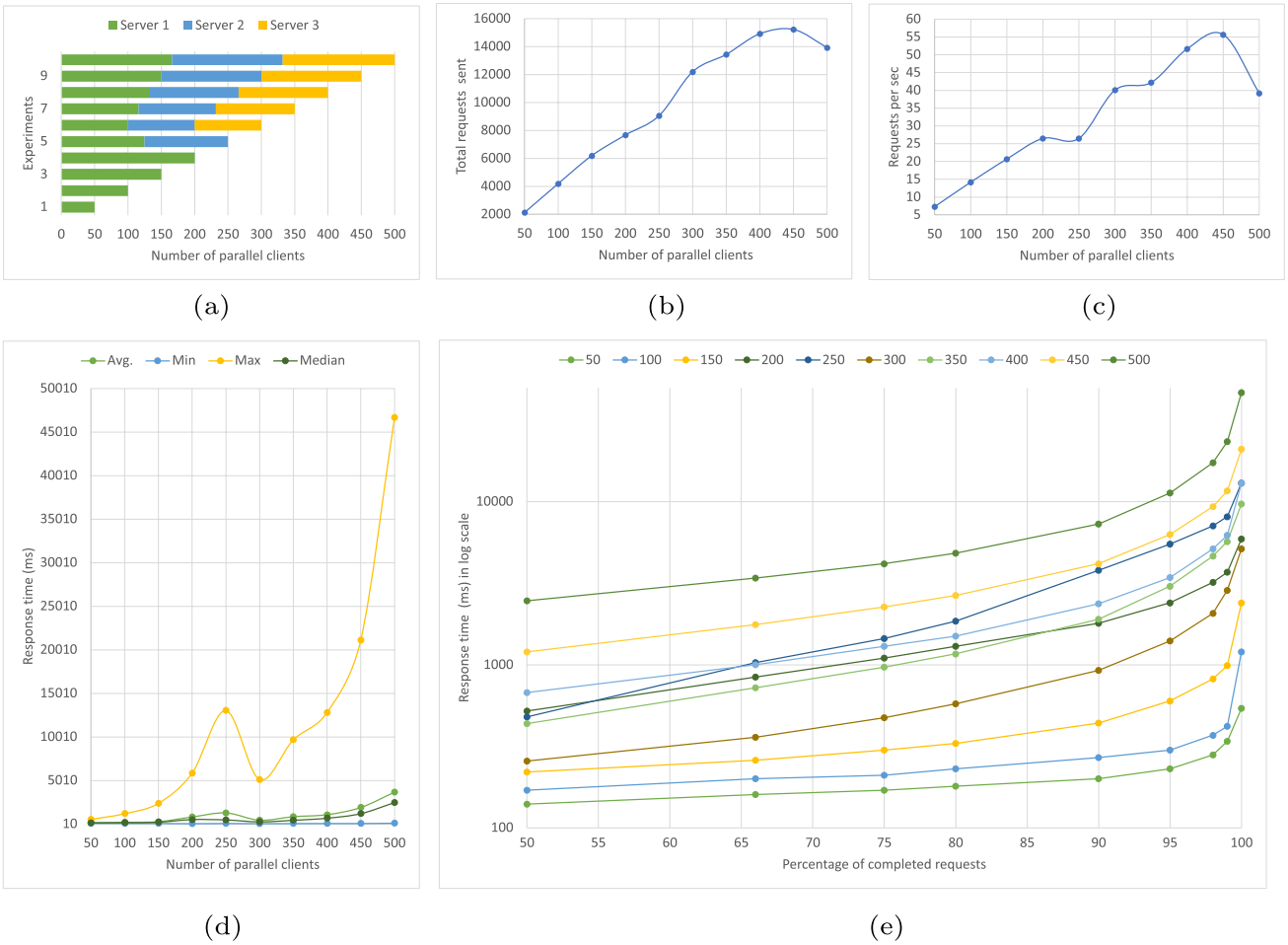


Fig. 4 Scalability of the system: (**a**) We generated a varying number of parallel clients from different machines, (**b**) Total number of requests generated, (**c**) Requests generated per second, (**d**) Response time vs. number of clients, and (**e**) Response time vs. percentage of completed requests

The average response time and the median response time overlaps which confirms that the response time is normally distributed.

In Fig. 3(b) we can see that almost all the request completes with in a specific time frame, that is, 99% requests complete in same time and around 1% files takes slightly more time to complete.

The results, Fig. 4(d), show that the response time increases linearly; the average response time and the median response time overlaps, confirming that the response time is normally distributed. Figure 4(d) also shows that the response time appears to improve when distributing the clients among multiple physical machines rather than generating from a single client. For that reason, we created not more than 200 parallel clients at each server (Fig. 4(a)).

In Fig. 3(e) we can see that almost all the request completes within a specific time frame, that is, 99% requests complete in the same time and around 1% files takes slightly more time to complete.

5.3 Discussion

We deployed the microservices to a Ubuntu Linux virtual machine configured with 16GB RAM and an 8-core Intel Xeon 2.20 GHz CPU. A cluster manager manages the microservice instances, and a load balancer distributes the load uniformly among the available eight instances.

In this paper, we reported a formal study where we used files between 1-100 MB. Although the analysts rarely collect very large files, e.g., executable, high definition audio and video files, ISO images, we also did tests to confirm if the system can handle such files.

The main objective of this empirical analysis was to verify if the system scales linearly, and the results confirm that the Hub's response time increases with the file size and number of parallel requests.

Moreover, the microservices are state-less application and are implemented using Node.js. Therefore, as the usage increases, we can create as many instances of the

microservices needed to meet that demand. We can do so very quickly using containers, e.g., Docker, and container orchestrator, e.g., Kubernetes.

6 User evaluation and methods used for providing feedback

When the CWM was first introduced, the analysts struggled to see how this would replace their manual workflow processes, consisting of an excel spreadsheet and numerous emails. We modeled and implemented custom dataflow and workflow process for each team; after that, the users could see the benefit of using the automated system. Based on analysts feedback, we made several additions and enhancements to replace the existing practices.

It would also take some time and effort to change the habits and mindset of the analysts. Therefore, the analysts started using the system parallel to the existing process, which highlighted several issues or missing functionality that would need to be added to the hub. Each time an analyst identified a missing functionality, they would work with us and discuss the issue. We would then suggest some possible solutions, and we would agree on the most suitable for implementation. Once we enhanced the system to a satisfactory level and the analysts had been using the system for a test period, we reviewed the various stages of the process from start to finish. We found that many enhancements had left some obsolete functions superseded but more appropriate addons.

In the second phase of the development, we integrated the DMS and ESE into the CWM. Again, we demonstrated this to the analysts. We went through the current process and then discussed the various enhancements that would require. There was some back and forward and parallel operation to iron out issues and shortfalls. We closely worked with the analysts and made all the enhancements required to roll out the system to all the analysts.

The hub's development and functionality continue to develop, with the next phase being the accounts department's inclusion to speed up invoicing and the ability to implement a customer-facing interface to receive customers' request, deliver products, and disseminate.

6.1 Changing company business practice and managing change

Our system have been gradually implemented across the company's different branches, with the investigations team acting as a trial group. By necessity, the transition to the Hub has been gradual, with training requirements and legacy issues demanding careful resolution. This, however, quickly offset by the efficiency gains brought by the company-wide introduction of the system and the changing working practices this brought. Within teams, the Hub allows for better allocation of resources and management of workloads. It has also enhanced the company's ability to operate on a 24/7 basis, ensuring shift workers are both clearly tasked and fully accountable for their outputs.

Looking at the company as a whole, the Hub system significantly contributed to how business is managed. Crucially, by streamlining the workflow process, the finance team can issue invoices more quickly, which better reflects the work conducted. From an HR perspective, the CWM makes it easier to account for individual staff members' outputs and balance resources more effectively between the teams.

For analysts working on the company's various teams, a key challenge has always been to make data as recoverable and as reusable as possible. The system meets these requirements, ensuring analytical value that can be straightforwardly added to data collection and without duplication of effort.

Data entering the Hub has the potential to include personal data. The Hub facilitates the easy adoption of measures to support the data protection regulation stated in GDPR, for example, by implementing the logical and operational safeguards for securing data, including solid privacy controls.

6.2 Implications of the introduction of GDPR

Article 5 of the GDPR (Information Commissioner's Office (UK) 2021) set out guidelines for how personal data should be collected, processed, updated, rectified, managed, secured, distributed and achieved. The system plays a critical role in addressing the important aspects of the GDPR, e.g.,

1. Store in a form that permits retrieve and identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed.
2. Allows analysts to delete or rectify any inaccurate personal data.
3. Ensures appropriate security and privacy or personal data, including protection against unauthorized or unlawful processing and accidental loss, destruction or damage, using appropriate technical or organizational measures.

7 Conclusion

Within this paper we have noted how similar the research and intelligence processes are, and have demonstrated how it should therefore be possible to reuse or integrate different tools meant for the research domain to the intelligence one. As such we consider also how the business process for what

was a small company can be improved through ensuring that the reasons for data collection and other associated metadata are collected and associated correctly with the raw data. This includes the provenance chain through the recording of a chain of ownership and access to the raw data and hence who has contributed to the final customer delivered product.

our system since its early phase of release has already showed that it is performing the required functionality of activity tracking. The integration of the DMS and the search engine the system's users are now able to allocate work across international time zones within the company, tracking the work that has been done previously, plan future activities within the team, and search for documents and files using free text. Over the last two years further integration with other in-house and third party tools are done such that the system became the foundation of the company's data management strategy and hence growth strategy.

Acknowledgements The authors would like to thank Innovate UK and Horus Security Consultancy, Oxford, for sponsoring this Knowledge Transfer Partnership (KTP) project. The authors would also like to thank colleagues from the Horus Security Consultancy for their feedback and evaluation of the system.

References

Alfresco (2021) Alfresco. https://www.alfresco.com Accessed 2021-04-01

Asite (2021) Asite. https://www.asite.com Accessed 2021-04-01

Atlassian (2021) Jira Software. https://www.atlassian.com/software/jira Accessed 2021-04-01

Bentley (2021) Bentley Eb. https://www.bentley.com Accessed 2021-04-01

Demchenko Y, Grosso P, De Laat C, Membrey P (2013) Addressing big data issues in scientific data infrastructure. In: Proc. of the 2013 int. conf. on collaboration technologies and systems, pp 48–55

Hogan R (2018) A practical guide to database design. CRC Press

Information Commissioner's Office (UK) (2021) Guide to the UK general data protection regulation (UK GDPR). https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr Accessed 2021-04-01

Khan S (2015) Visualization assisted enterprise search engine. PhD thesis, Department of Engineering Science, University of Oxford. https://ora.ox.ac.uk/objects/uuid:d1790b99-c30e-487b-b87e-98d4e3a8b2bb

Khan S, Kanturska U, Waters T, Eaton J, Banares-Alcantara R, Chen M (2016) Ontology-assisted provenance visualization for supporting enterprise search of engineering and business files. Adv Eng Inform 30(2):244–257

Kruschwitz U, Hull C (2017) Searching the enterprise. Found Trends Inf Retr 11(1):1–142

Rejeb A, Keogh JG, Rejeb K (2022) Big data in the food supply chain: a literature review. J Data Inf Manag:1–15

Trac (2021) The Trac ticket system. https://trac.edgewall.org/wiki/TracTickets Accessed 2021-04-01

Trefethen A, De Roure D, Newman D, Wallom D, Emptage N, Lakhoo R (2012) NeuroHub: A research information environment for neuroscientists 4

UK Ministry of Defence (2011) JDP 2-00 (3Rd Edition) understanding and intelligence support to joint operations technical report

Van Der Aalst WMP, La Rosa M, Santoro FM (2016) Business process management: Don't forget to improve the process!. Bus Inf Syst Eng 58(1):1–6

Van Loenen B, Kulk S, Ploeger H (2016) Data protection legislation: A very hungry caterpillar. The case of mapping data in the European Union. Gov Inf Q 33(2):338–345

Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, Bouwman J, Brookes AJ, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo CT, Finkers R, Gonzalez-Beltran A, Gray AJG, Groth P, Goble C, Grethe JS, Heringa J, t Hoen PAC, Hooft R, Kuhn T, Kok R, Kok J, Lusher SJ, Martone ME, Mons A, Packer AL, Persson B, Rocca-Serra P, Roos M, van Schaik R, Sansone SA, Schultes E, Sengstag T, Slater T, Strawn G, Swertz MA, Thompson M, Van Der Lei J, Van Mulligen E, Velterop J, Waagmeester A, Wittenburg P, Wolstencroft K, Zhao J, Mons B (2016) The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 3:1–9