Forensic Science International: Digital Investigation 35 (2020) 301021

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

Digital forensics as a service: Stepping up the game

H.M.A. van Beek*, J. van den Bos, A. Boztas, E.J. van Eijk, R. Schramp, M. Ugen

Netherlands Forensic Institute, The Hague, the Netherlands

ARTICLE INFO

Article history:
Received 10 January 2020
Received in revised form
25 June 2020
Accepted 25 June 2020

Keywords:
Digital forensics
Digital forensics as a service
DFaaS
Hansken

ABSTRACT

After providing Digital Forensics as a Service (DFaaS) implementations to law enforcement agencies for close to a decade, we present our view from an inside-out perspective. We share the lessons learned from an organizational, operational and development perspective in a forensic and legal context. We conclude with our vision on how to bring the DFaaS concept to the next level for both investigative and innovative purposes.

1. Introduction

The ubiquitous use of information and communication technology in society at large is providing its users with real time access to all kinds of resources (e.g., travel schedules, high school grades and banking applications). This also attracts users with a more nefarious inclination, in order to commit old crimes by digital means (e.g., use encrypted text messages for communication with accomplices) or to commit totally new cybercrimes (e.g., hacking or blackmail using ransomware). This leads to a growing need for digital forensic results by all stakeholders, including case investigators, cross-case analysts, prosecutors, lawyers and judges. This raises high demands on digital forensic knowledge, tools and expertise in a rapidly changing technological context, applicable in the pace of operational judiciary systems.

We approached the scalability issues of digital forensic processes by implementing Digital Forensics as a Service (DFaaS) (van Baar et al., 2014). Since 2010, we provide this service to law enforcement organizations in the Netherlands (including ourselves), first implemented as XIRAF (Bhoedjang et al., 2012), later as HANSKEN (van Beek et al., 2015). Since 2019, we make HANSKEN available to governmental (law enforcement) organizations and science institutes outside the Netherlands.

From all the lessons learned in the last decade, we conclude that an open and extensible platform is paramount to efficiently and effectively cooperate and share knowledge.

We summarize the DFaaS concepts in Section 2. In Section 3, we present the progress we made on HANSKEN together with our experience with providing and using the platform. We share our lessons learned while servicing several Dutch law enforcement agencies in Section 4. Next to the organizational impact, we provide an operational and development perspective in this forensic and legal context. We present our vision on the future of the DFaaS concept for both investigative and innovative purposes in Section 5. Finally, we discuss related work in Section 6 and summarize our final conclusions in Section 7 by presenting the path we followed to implement DFaaS in The Netherlands.

All links presented in this paper were last visited on March 31, 2020.

2. DFaaS concepts

In 2014, we analyzed the traditional digital-forensics investigation process and introduced Digital Forensics as a Service (van Baar et al., 2014). As stated in the paper introducing our implementation of the concepts (van Beek et al., 2015), the main goal is to provide a service that processes high volumes of digital material in a forensic context and gives easy and secure access to the results.

In (van Beek et al., 2015) we list the main forensic drivers that led to the DFaaS concepts: (1) *minimize case lead time*, to make the digital evidence available to investigation teams as soon as possible; (2) *maximize coverage*, in order to understand as much from the seized digital material as possible; and (3) *efficiently mobilize people*, to let dedicated people work on specialized tasks for which they are educated and equipped.

* Corresponding author.
E-mail address: harm.van.beek@nfi.nl (H.M.A. van Beek).

https://doi.org/10.1016/j.fsidi.2020.301021

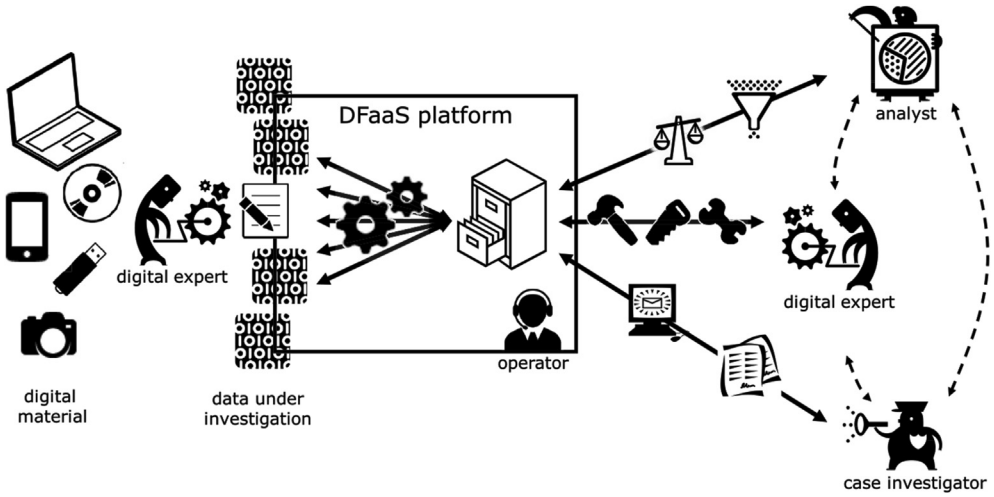

Fig. 1. Digital forensics as a service.

We explained how a centralized digital forensic platform helps in organizing the handling of the digital evidence, based on the adopted process model presented in Fig. 1.

On the right, there are case investigators and analysts that have questions related to the digital evidence. First, digital experts acquire digital material from devices (i.e., physical or logical disk images and digital forensic reports). This base material is uploaded to the central platform and processed using a configurable set of tools. The results of applying these tools to the data under investigation are stored in a centralized database. Next, these results can be queried, filtered and aggregated using multiple methods that utilize the platform's API. A role we added later to this figure is the *platform operator*, responsible for the inner workings of the platform.

In DFaaS, we have a clear notion of digital traces. A *trace* represents a digital artifact and consists of (a link to its) data and corresponding meta data. While processing digital material, traces are identified and meta data is extracted. We call this process the *extraction* process (the gears in Fig. 1).

The meta data extracted by the platform must adhere to a predefined *trace model*, specifying the structures of a trace. This model forms the base for the structured trace database (the cabinet in Fig. 1) and is key for sharing and reusing knowledge. The model forces traces to be normalized, making it possible to easily query, filter and aggregate the traces.

Next to normalization, the *origin* of traces and their properties are contained in the model. This origin can be based on the structure of the data (*extracted*) or on an artificial intelligence tool (*mined*). Also, a user can upload or extend the trace via the platform's API (*user added*) or add annotations to a trace (*annotated*). Finally, properties can relate to the actual processing of the data, capturing the provenance of the trace (*processed*).

3. Progress on Hansken

In 2015, we presented Hansken (van Beek et al., 2015) as our next generation implementation of the DFaaS concept (van Baar et al., 2014). Hansken, named after a famous 17th-century elephant,[1] consists of several interconnected components, following a service-oriented architecture. Hansken is designed and implemented based on security, privacy and transparency principles, among others (van Beek et al., 2015).

Fig. 2 contains an overview of all implemented concepts and their functions. Hansken consists of three layers: the back-end tools encapsulating the forensic knowledge, the centralized DFaaS platform and the front-end for utilizing the platform in criminal investigations and scientific research.

Hansken provides several well-documented application programming interfaces (APIs) based on a decomposition of the investigative process (van Beek, 2018). This makes it easier to develop components for specific functions in the process. Hansken provides an API for connecting tools to extract traces and end-user APIs for operations, investigations, user preferences and key management. A (Python) scripting library wraps the REST APIs and facilitates operating the platform or investigating cases in an automated manner.

While querying and filtering the trace store via the platform API, typically a subset of the traces is returned. Adjacent to the traces themselves, statistical information on the result set can be delivered by the platform, called aggregations.

The Hansken trace model's descriptions are translated into several languages to support multilingual user interfaces, among others for populating lists and defining filters.

3.1. Example cases

Hansken is used in over 1000 cases, among which:

- cases with data from over 1000 devices;
- cases with over 100 terabytes of raw material;
- cases with over 100 million identified traces;
- over 100 cases available concurrently.

With Hansken it was possible to investigate these cases using one index with acceptable response times (a few seconds per query).

In 2016, the Dutch Prosecution Office seized several mail servers in Canada that were used for secure (PGP) communication using specially adapted BlackBerry phones. They stated[2] that the 3.6 million encrypted messages found on these servers will provide evidence for several criminal cases, including murder, armed

[1] http://www.elephanthansken.com/about/

[2] https://www.om.nl/actueel/nieuws/2017/03/09/versleutelde-berichten-schat-aan-criminele-informatie (in Dutch).

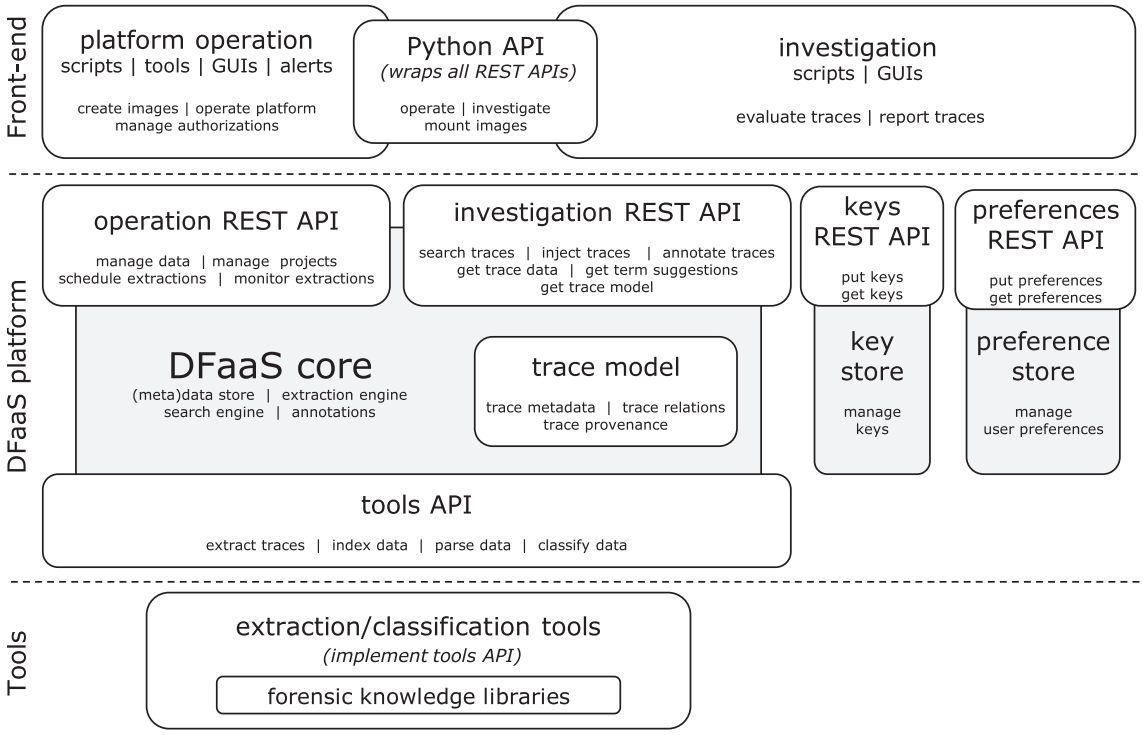

Fig. 2. Hansken concepts.

robbery, drugs, money laundering and other forms of organized crime.

Detailed questions were posed by defence attorneys about the use of Hansken. They are centered mainly on the issue of quality control: How was the data entered into Hansken? Was the data edited? How accurate is Hansken? The court concluded that Hansken was not used to collect evidence, but rather to search through and provide insight into already available evidence. As a result, it has no influence on which information is collected.

The court in Canada set a condition for the use of the data: not all data could be used in all cases. The Public Prosecution Service must specify in advance, to the court, what information it intends to examine. After receiving the court's approval, this subset of the data is filtered out from the rest, with the help of Hansken (see Section 3.3.4), and made accessible to case investigators.

With a court ruling (Court of Amsterdam, 2018) in 2018 that the 3.6 million encrypted messages from the mail servers in Canada were lawfully searched, Hansken withstood a thorough judicial review.

3.2. Development and operations

Developing an entire platform including both a custom back-end and multiple front-ends is a large undertaking for any organization, but specifically for one that traditionally focuses on completing individual reports and not on producing software or supporting an operational digital service-based environment. In order to develop and operate Hansken we lean on leveraging existing technology, applying industry best practices and employing specialists in areas besides digital forensics.

3.2.1. Technology

We did not build Hansken from scratch. A guiding principle is that we do not want to reinvent wheels, but adopt and adapt existing third-party technologies to the digital-forensic context. Table 1 summarizes the main technologies Hansken builds upon and where in our platform we apply them. We also list the most prominent tools we use during the development and deployment of the platform.

3.2.2. Development

The development of a platform such as Hansken entails collaborating with many stakeholders and integrating the expertise of many diverse specialists. In order to keep the associated processes simple and manageable, we have applied agile principles from the start of the initial development. Currently, our processes can be characterized as SCRUM and SAFe, with many development practices from eXtreme Programming. The goal of these practices is always to introduce feedback loops that are short and automated where possible.

The current main development organization consists of seven teams of approximately five members each, including many different roles and specializations such as developer, forensic specialist, architect, operator, tester and system engineer. This allows development to take place on different parts of the platform concurrently, such as automated cluster deployment scripts (developed and maintained by system engineers), forensic data-analysis tools (by forensic data analysts and software developers) and user interface improvements (by software developers and interface designers). This in coordination with cross-cutting concerns such as architecture and design, quality assurance, build and deployment automation and many others, of which the developers and maintainers are embedded across teams to foster knowledge transfer and collaboration.

For each change, over 40,000 automated tests are executed. If one fails a change is not accepted. Such a comprehensive test suite is the result of a systematic approach over the years. A dedicated group of testers continuously maintains and improves the test suite during each release cycle. First, for each functional addition or change, a set of synthetic tests is assembled to verify the

functionality. Second, whenever a bug is reported, it is formalized and documented as an automated test case, in order to assist developers to reproduce it and to add it to the test set for regression. Third, a set of large and realistic integration tests that mimic a realistic input (such as the contents of a hard drive or mobile device) is run to automate acceptance testing.

See Lesson 4.3 for more information on our lessons learned while working agile in a governmental context.

3.2.3. Operations

Deploying and maintaining such a large platform requires expert knowledge. Dedicated system engineers maintain the hardware and deployment scripts. They take part in the DevOps team, making communication lines short towards the developers and forensic specialists. Due to the agile approach, new releases of the platform are available every few weeks. Deployment of these new releases is automated extensively, in order to reduce any dependency on manual tasks and guarantee uniformity and reproducibility.

Several Dutch law enforcement agencies have set up local implementations of Hansken. They have made the platform available as a service to their case investigators and digital experts. We support the operation of those implementations with guidance and use this experience to further improve the automation of our testing and deployment tools.

Centralizing digital forensic analyses and data handling must be orchestrated (i.e., the load on the platforms must be prioritized, scheduled and monitored). These operator tasks are currently fulfilled by trained and skilled operators, in close cooperation with digital experts, investigative teams and prosecutors. Adjacent, they support end users with Hansken-related questions for their specific investigations.

These operators are not digital experts, instead they support the DFaaS process. Their role in the process however is of great importance, as they fulfill key functions around handling and processing supplied data in a forensically sound manner, communicating problems and performing as a helpdesk to users.

3.2.4. Backups and archives

The goal of an operational platform is to be used for investigative purposes, not as an evidence archive. When an investigation is completed, its case data must be archived and removed from the platform, of course in consultation with the investigative teams. In this way, case data on the platform can be better managed and the resource and capacity of the platform can be freed up and used for other cases.

Nevertheless, backing up and archiving of data is an important part of digital forensic investigations. In case of an emergency, a backup can be used to quickly restore data and can ensure that the investigation continues. Additionally, case-related evidence must be archived for reproducibility of the investigations.

There may always be circumstances that require an investigation to be carried out or re-evaluated, typically upon request of (higher) courts or defense attorneys. Hansken is developed in such a way that it makes it easy for backing up and archiving of the user-added traces and annotations. Archiving of the material uploaded to Hansken is currently not part of its implementation.

3.3. Automated digital forensics

As in all areas of society, digitization is also a large component of modern forensics. Digital forensics, which has always dealt with digital evidence, takes a natural lead towards increasing automation. This means that instead of researching, developing and applying knowledge, methods and tools in the area of digital forensics itself, we also focus on expressing these finding as software tools (and fundamentally, not just how to do things, but how to do them in software).

In this sense, Hansken is a necessary tool to express and apply progress in service-based, automated digital forensics. Even though our development philosophy of not reinventing wheels applies here as well, the area of automated digital forensics is still in its infancy and has shown to require much more custom methods and tools.

3.3.1. Automated trace extraction

Hansken orchestrates a fixed process for a large set of specific tools to extract traces from data entered into the system. The first step entails determining the type of the data. This process uses provided metadata, but also attempts to directly recognize the data itself. Such recognition steps can be as simple as looking for a file type-specific marker at the start of a stream, up to a more complicated approach to find specific types of tables in a database format to determine not only the basic type of database (e.g., SQLite), but also the application it is related to (e.g., WhatsApp messages).

Based on this identification, a set of tools is selected to attempt parsing of the data. Parsing identifies two main types of data well-known in computer science parsing: terminal and non-terminal trace properties. Terminal types, or properties, refer to or describe a final piece of information, such as location, size, date, name or type of something and are directly recorded and stored. Non-terminal types, or containers, refer to data that needs to be recursively processed again, as it was encapsulated by the current data. Common examples of this are files in a file system, payloads in a protocol message, compressed files in an archive, thumbnail images in a picture, mail folders and attachments in a messaging database and so on.

Top-level data entered into Hansken tends to consist of large containers such as volumes and file systems, but can also be large databases, compressed archives or network captures. In practice many recursive steps are needed to find all the properties, as is shown in Table 2, where a picture's camera type was found after processing an image, a volume, a file system, many folders, a large email archive, several email folders, many emails with attachments and the resulting picture inside one of those attachments.

The implemented platform is transparent with respect to this data processing: during the processing stage, for each property of each trace, the platform records the sources in the original material and the tools that were subsequently applied to retrieve the property. A single piece of information may end up scattered across multiple devices and locations and requiring several transformations (e.g., related to fragmented file systems, compression or out-of-order protocol messaging) to retrieve from its original sources.

Table 2 provides an example for a picture in a compressed file container attached to an email message.

3.3.2. Failures and error handling

Many things can cause traces to not be found by a manual or automated investigation. From a forensic point of view, there is a difference between whether data was not examined, whether it was examined and nothing was found or whether an examination was started but not completed for some technical reason. To provide insight into the automated processes leading to the reported traces, each trace is stored not only with its properties and relation to transformation and locations, but also with information about how it was found.

This includes information on the tool and version used to examine the information as well as whether it executed correctly. All this information, along with error messages and other logging

information is stored in each trace. If in some cases this information is too large (e.g., error messages that provide large outputs as debug information) it is still stored, but in a separate *failed traces*-log and referenced from the trace.

Apart from this direct coupling of output and run-time information, Hansken also counts the amount of tools that are run to examine a specific input in order to have an additional soft error-checking mechanism. In cases where a smaller amount of outputs are returned, an entry is created in the logs to note that some of the tools may not be executing correctly.

An important observation, especially when assembling a large-scale platform in order to aggregate different types and implementations of forensic knowledge, is that the quality of the output is directly related to the lowest quality component or tool used by the platform. While the impact of unreliable tools may be limited if they are used to process rare or generally uninteresting information, having a parser for a top-level format such as a common file system or network protocol that is of low quality will have a hugely negative impact on anything that eventually is or is not found. This is an important reason for us to implement many of these top-level forensic knowledge components ourselves, in order to have direct control over properties such as logging, error reporting, algorithm selection and software engineering trade-offs.

3.3.3. Provenance

We designed the platform with security, privacy and transparency in mind, i.e. zero trust (security), you can only access what you need (privacy) and all activity is visible (transparency). Those principles serve as a basis for the forensic and legal demands.

The provenance of traces is paramount for them to be accepted in court. This provenance consists of the chain of evidence (the relation of the traces to the original material) and the chain of custody (the detailed process that led to the trace).

Hansken gives as much insight in its internal workings as possible. The following information is available via the Hansken APIs:

For each image:

- start time and operator who uploaded the image;
- unlimited properties for storing provenance information;
- hash totals of original image;
- calculated hash totals after upload.

For each extraction:

- start time, duration and operator who started the extraction;
- overall statistics (e.g., number of traces extracted and number of bytes read);
- statistics per tool:
  - number of invocations, including profile (e.g., time spent and bytes read);
  - number of created traces;
  - number of processed traces;
  - number of failures, including a summary of the failures;
  - unused tools (i.e., tools that were not invoked).

For each trace:

- parent trace (source);
- tool that created the trace, including version details;
- tools that processed the trace, including version details and invocation order;
- for each property: the tool that created it, including version details;
- tools that failed processing the trace;
- calculated hash totals;
- relation to the original image data.

The use of micro services and well-defined programming interfaces (API) in the platform design, also makes it possible to easily monitor and track the activity and use of the platform.

3.3.4. Investigation scoping

In several cases, there are limitations on the use of digital evidence.

First of all, legal and privacy protections can restrict access. This typically *white lists* traces, i.e. it states what traces are allowed to take part in an investigation (e.g., emails in a specific account, pictures in a specific folder, documents created by a specific author, or traces within a specific time period). Seizing only this material is not always possible (e.g., due to time restrictions or since the material needs to be analyzed first). A simple solution would be to create a mandatory filter that implements the restriction and apply this filter to all queries on the trace collection in the DFaaS platform. This, however, is not sufficient for all functionality provided by Hansken (e.g., for text suggestions based on words found in the data). Such functions must adhere to these restrictions as well. In such cases, a so-called *partial clone* of the index is created, based on the filter that implements the restriction. Case investigators are authorized to access only this partial index.

Secondly, law can limit investigative teams to gain insight in communication or documentation related to medical or legal files, so-called privileged communication. This typically *black lists* traces, i.e. it states what traces are *not* allowed to take part in an investigation. To support investigative teams with handling this material in a legal way, the Hansken platform is extended with specific functionality. During the trace extraction process, filters can be applied to identify and mark suspected traces. These traces are *not* available to the investigators. This is never a perfect solution, there will be false positives as well as false negatives. Human intervention is needed to evaluate these markings. If traces are not marked, but identified to contain privileged communication while investigating, the API and GUIs provide functions to hide them for the investigators. These functions support the processes defined and approved by the Prosecution Office. In multiple case investigations, they are successfully applied, reported and discussed in court (Court of Amsterdam, 2018).

4. Lessons learned

To develop, operate and use this service in the Netherlands, multiple law enforcement agencies bundle their capacity, work together and share knowledge. We learned (and are still learning) many things during this process, having several setups of the platform support over a thousand cases, processing petabytes of data.

We are aware of the advantages we have as a governmental organization. By law, the NFI has a role to support the legal justice system. We have short lines of communication with digital experts and case investigators at law enforcement agencies, prosecutors and judges. For case-specific functions, developers can get access to data that is under investigation.

Implementing Digital Forensics as a Service has many advantages, but of course also has several limitations. We experienced that implementing Digital Forensics as a Service in an organization is complex.

4.1. Lesson: The DFaaS business case is hard to make

Centralizing the data processing gives insight into otherwise hidden costs. These operational costs (labor, hardware and

software) used to be spread over organizations, teams and individuals. This makes it hard to measure these costs and thus to compare them with the costs for setting up and maintaining a centralized platform. As a result, it is hard to make a clear business case (i.e., to identify the actual profits expressed in money, time and labor).

Although the costs for centralizing are significant, organizations experience that the benefits are worth it. Especially, since the handling of the digital evidence is clear and comprehensive. People involved with criminal investigations spend more time on their expertise and better focus on their role in the investigative process. This has led to a significant change in the way digital experts and case investigators cooperate in handling digital evidence. Instead of acting as bulk data processors and data transfer hubs, digital experts can now focus on interpreting and evaluating relevant data and traces identified by case investigators.

This all sounds like an easy business case, but making the transition from a traditional way of working to a centralized one is complex. Setting up a DFaaS platform with new hardware and software is not the challenge. This can be done in a separate and new environment that can be maintained and operated by a small team. In The Netherlands, these teams contain typically four to six people per platform implementation. The organization, however, needs to be transformed by introducing a new way of working and cooperating.

4.2. Lesson: All users' needs must be taken into account

For making these process changes work in practice, all people involved have to break out of their customary and comfortable situations. Such far-reaching changes inevitably generate resentment and resistance. It is key for an organization to support their people by making this change as easy as possible. To convince them to embrace such a change, they first of all must experience the benefits. Their needs and requirements must be taken into account when developing, deploying and operating a DFaaS platform. This especially holds for the digital experts, who used to play the pivotal role in the process. Now, they have to release control of parts of this process and depend on a remote system to support their work.

A wrong or incomplete implementation can raise resistance, undermines the adoption of the concepts, directly reducing the benefits. Even as implementations improve, this can have negative effects on the long term. Availability of the platform is key for it to be accepted. The platform must be available in the right place at the right time. For example, end-users should not switch computers to access the platform and it must be easy to add investigative results to their case reports.

4.3. Lesson: Working agile in a broad governmental context is hard

In general, governments are organized in a bureaucratic way. To minimize risks, each adoption must follow a detailed and fine grained paper trail. The benefits must be secured in advance to get commitment on the investment. As a result, decision making is a slow process, especially in a large scale and complex legal context.

The rapid change in the course and direction of technology, however, causes digital forensic investigations to continuously work with moving targets. The classic waterfall approach cannot be used to deliver a platform that adheres to the ever-changing demands on digital-forensic investigations. So, tailoring an agile process that takes into account all stakeholders' needs and contributions is a task that takes time and tenacity.

We have been working on Hansken using an agile (Beck et al., 2001) approach from day one. Changes are welcomed and can be incorporated quickly. The backlog is organized in epics and features. During three-weekly cycles, the product owner and software engineers sit together with the business owners (representing the involved law-enforcement agencies) to prioritize and schedule the work, enabling quick adaptation to changing needs. Time is reserved for high-priority bug fixing and case-specific support. Each sprint finishes with a demonstration of new or improved features and in most cases with a release. We recently finished our hundredth sprint.

Although an agile approach supports short release times, there

Table 1
Hansken's main third-party dependencies.

Technology	Use
Java	The Hansken core platform and extraction tools are coded in Java.
Apache Zookeeper	Hansken implements a service-oriented architecture, where Zookeeper is used for maintaining configuration information, naming, and providing distributed synchronization of the Hansken services. https://zookeeper.apache.org
Ceph, Hadoop HDFS	The data store supports several underlying (distributed) file systems, among others Ceph and HDFS. https://ceph.io, https://hadoop.apache.org
Hadoop MapReduce	Distribution of the extraction process is implemented on Hadoop MapReduce. https://hadoop.apache.org
Kafka	Progress of the extraction processes can be monitored using Kafka. https://kafka.apache.org
Elasticsearch	The trace meta data and keyword indices are stored in Elasticsearch, which also serves as core for the query, filter and aggregation engine. https://www.elastic.co/products/elasticsearch
Cassandra	Additional data, like user preferences and project administration is stored in Cassandra databases. Furthermore, it is used for caching specific trace data and storing data related to user-added traces. https://cassandra.apache.org
Knockout, AngularJS	The current graphical user interface are based on Knockout (Tactical UI) and AngularJS (Expert UI). https://knockoutjs.com, https://angular.io
NGINX	These interfaces (websites) and additional documentation are hosted using NGINX. https://www.nginx.com
OpenStreetMap	An offline OpenStreetMap setup provides graphical map data. https://www.openstreetmap.org
GroupDocs.Viewer	Presentation of trace contents takes place by transforming source data into HTML5 using (commercially licensed) GroupDocs.Viewer. https://github.com/groupdocs-viewer
fusepy, FUSE	The Python API (wrapping the REST API) support the mounting of disk images in Hansken, using fusepy and FUSE. https://github.com/fusepy, https://fuse.sourceforge.net
Keycloak	For the authentication of users, Hansken builds on Keycloak, which can be configured to use single sign-on with several well-known identity providers. https://www.keycloak.org
OpenLDAP	User authorizations can be registered in OpenLDAP. https://www.openldap.org
Grafana	Grafana is used for monitoring all these services and tooling. https://grafana.com
Atlassian suite	We use the Atlassian suite to plan, track, test, release and deploy the Hansken software. https://www.atlassian.com
Ansible Kubernetes	Using Ansible, automated deployment can be done on configurable hardware. Alternatively, Hansken can be containerized and deployed using Kubernetes. These deployments include the underlying 3rd-party systems. https://www.ansible.com, https://kubernetes.io

are typically more change requests than there is time to refine and implement them. As a negative side-effect, some requests might never be prioritized and thus never get implemented. Also, scheduling new features based on benefits to the end-users is difficult when multiple organizations are involved with somewhat differing views on the perceived benefits. Especially, since these organizations have to justify their input (either in funding or manpower) in a traditional way.

4.4. Lesson: Continuous development can frustrate forensic investigations

The key benefit of an agile approach and centralized deployment is that new features become available as soon as they are implemented and tested, typically once every few days or weeks. This makes it possible to adhere to the high demands on digital forensic software, but might frustrate forensic investigations. Since DFaaS implementations depend on many underlying technologies (see Table 1 on page 4), also updates or upgrades of these systems must be taken into account. Compatibility with existing hardware and underlying software is crucial for the DevOps process to function. Apart from important security patches that must be applied, such changes might cause significant changes for operational platforms (including major improvements).

A recent example is the upgrade in Hansken from Elasticsearch version 2 to version 7. Since the API of this technology changed, this had serious impact on Hansken. Next to upgrading the software itself, the roll out of a new version of Elasticsearch required temporary addition of hardware. Some cases had to be migrated which led to extensive communication with investigative teams.

Software components must be properly tested before committing to a new version and full integration tests must pass before releasing new code for production. Even though we thoroughly test updates and upgrades, we experience that setup errors, incompatibilities or performance issues can negatively influence an operational platform. This impact may not always be experienced immediately after installation.

Case investigators not only use the platform for getting access to the material, they also administer the investigation in it. Hansken allows users to mark evidence items for review and offers the possibility to annotate traces. One of the limitations of this functionality is that after reprocessing the data, it is not always possible to link previously created notes to the newly extracted traces (e.g., because the old traces were not valid and replaced with correct ones, or they are extended with new properties, updating their representations).

The issue of practitioners having to (partially) redo their investigation is not unique to DFaaS. However, the centralized nature as well as the continuous development process both exacerbate this problem by synchronizing updates and deployment across all users and cases. From a technical point-of-view this also brings many advantages: all users use the same version of all tools, without requiring action on their part.

4.5. Lesson: A monolithic platform will never support all case-specific needs

Every day, new devices come to market, new apps are released and many apps are updated. Each new criminal investigation shows new digital material that is not yet supported by forensic tools.

It is impossible to keep pace with all these developments. As a result, a DFaaS platform might not fulfill the often urgent needs for specific case investigations (e.g., digital currencies, cloud data, but also sudden large changes such as a new Apple File System). In those cases, other means are found to process the digital evidence (e.g., open source or commercial tools and case-specific scripts).

Supporting all these new means to process evidence by a DFaaS platform is only feasible if it provides solutions to embed or connect external methods and tools.

4.6. Lesson: A DFaaS platform does not replace a DF expert

DFaaS brings digital evidence in reach of laymen in the field of digital forensics. The amount of tactical information that is directly available in the digital material is huge. Even without detailed forensic evaluation that is needed for a profound understanding of the material, case investigators can use this tactical information for investigative purposes. However, their different educational backgrounds often lead to less critical reviews of the interpretation of

Table 2
Example trace nesting.

image: copy-of-disk.raw
data.raw.size: 2,199,023,255,552
data.raw.hashes.sha1: 02ee 55fa...
→**volume**, **filesystem**: NTFS
→**folder**: /
→**folder**: users
→**folder**: Jack
→**file**, **email archive**: my-emails.pst
→**email folder**: inbox
→**email**: received
→**attachment**, **file archive**: pictures.zip
→**file**, **picture**: NFI-building.jpg
file.name: NFI-building.jpg
file.createdOn: 2020-01-18 16:07:12 Z
file.owner: Jack
picture.camera: iPhone7
picture.width: 2048
picture.height: 1536
data.raw.size: 2,320,429
data.raw.hashes.sha1: 8cd9 364e...

forensic artefacts. As a result, missing or incorrectly labeled information is not recognized as such, making users jump to conclusions (Hamilton, 2013).

For a more in-depth review with independent tools, the user depends on a second-line digital forensics expert. To avoid confirmation bias and missing analyses of alternatives, personnel must be trained to critically review the results and involve digital forensics experts.

The risks discussed have always existed in digital forensics, as software has always been used to perform many tasks. What a DFaaS approach in this situation does is amplify the potential for problems due to its large-scale nature, but this actually highlights the importance of professional culture and education.

4.7. Lesson: DFaaS must serve all stakeholders in a crime case

The use of Hansken for forensic investigations has been discussed in Dutch courts on multiple occasions (Court of Amsterdam, 2018; Court of Gelderland, 2019). NFI-employed court experts answered hundreds of question on the internals, reliability and practical use of the platforms, explaining among others the effects of bugs and the completeness and specificity of the results.

This led to discussions in courts on applying the principle of *equality of arms* to all digital evidence (i.e., all parties involved in a crime case must have access to digital evidence for investigation and evaluation purposes). Of course, this depends on the type and context of a case and use and origin of the material. Giving defense attorneys/suspects direct access to digital evidence via DFaaS implementations can answer this demand. Access might need to be limited, for example because the digital material contains (seized) digital currencies or illegal content like child sexual abuse material (CSAM). Also, activity in the DFaaS platforms by both parties (LEAs/prosecutors and defense attorneys/suspects) should not be available to the other party (e.g., the executed queries, applied filters and added annotations). We are currently collecting requirements for extending Hansken in this context.

5. Vision on digital forensics as a service

We conclude with our vision on the DFaaS concepts, making implementations future proof for investigative and innovative purposes. This vision guides our efforts on improving the Hansken platform.

An open and extensible platform is paramount to efficiently and effectively cooperate and share knowledge. To enable all experts and investigators to focus on their interests and expertise, optimally utilizing their knowledge and skills, implementation of the DFaaS concepts should aim at job focus and cooperation. Focus should be on centralizing data sharing and automation of the trace extraction process (data structure analysis) for investigative purposes. This is part of the knowledge needed for handling digital evidence in a criminal investigation.

5.1. Vision: DFaaS platforms support standardization

Historically, the various forensic sciences have been using standards for testing and calibration laboratories as their own certification standard. For instance, the UK Metropolitan Police became certified against ISO 17025 (ISO 17025, 2017) in 2015.[3] Raising use of this standard within the forensic domain, shows inadequacies of this standard. Work has started on formulating a standard dedicated to forensic science: ISO 21043 (O 21043-1:2018: Forensi, 2018; O 21043-2:2018: Forensi, 2018). In the IT security area, separate standards formulate how to collect and preserve digital evidence (O 27037:2012: Informati, 2012) and how to analyze and interpret digital evidence (O 27042:2015: Informati, 2015).

Using a standard platform for the storage and analysis of digital evidence, economy of scale can be achieved for getting compliance to these standards. Also, other forms of compliance are mandated, such as privacy and security norms.

Furthermore, to make and keep DFaaS implementations future proof, they must support international standards for storing forensic data and representing digital evidence.

5.1.1. Forensic data storage

Since most digital forensic investigations highly depend on commercial tools, their storage formats have become de facto standards. Examples are Guidance's EnCase Image File Formats (E01, Ex01) for physical (exact) copies of data stored in seized data carriers, AccessData's AD1 and EnCase Logical (L01, Lx01) for logical images, and Cellebrite's UFED reports and MSAB's XRY reports for evidence extracted from mobile devices.

To interchange evidence and interconnect tools, tool-independent international standards are introduced that gain more interest. Non-commercial formats are adopted by both commercial and non-commercial tools, especially Advanced Forensics File Format 4 (AFF4) (Cohen and Schatz, 2010; Schatz and Cohen, 2017) for physical images and AFF4-L (Schatz, 2019) for logical images.

While designing and implementing Hansken in 2013, we developed the NFI Forensic Image Format that adheres to the complex combination of design principles: data must be compressed, encrypted, but also seekable. The format supports both physical and logical images. We are currently researching if and how we can replace our implementation with the open AFF4 and AFF4-L formats, in order to improve interoperability and free ourselves from maintenance work.

5.1.2. Evidence representation

Digital evidence can be of interest to other investigations, possibly taking part in investigations in other organizations or jurisdictions. Representation of such evidence and its provenance is key for the receiving party to use it as legal evidence. Several alternatives to represent digital evidence are available, among others Digital Forensic XML (DFXML) (Garfinkel, 2009, 2012) and Digital Forensic Analysis eXpression (DFAX) (Casey et al., 2015), Casey et al. (2018) provides a good summary.

Since 2016, a community-wide ontology for representing digital evidence is under development, called Cyber-investigation Analysis Standard Expression (CASE) (Casey et al., 2017). A growing community[4] of academic, governmental, law enforcement, for-profit and non-profit organizations cooperate in this initiative. We plan to map the Hansken trace model to CASE.

5.2. Vision: DFaaS is about sharing DF knowledge

Traditional forensic desktop tools incorporate many tasks: acquisition, extraction, storage, and presentation. Instead, a DFaaS implementation unbundles these tasks, in order to connect and integrate knowledge from many different tools. This also allows many different parties to concurrently build software components for the platform. This requires a standardized way to integrate with

[3] https://www.ukas.com/news/metropolitan-police

[4] https://caseontology.org/

external components (e.g., to communicate context, execute without UI and return results). Our goal is not to reinvent wheels, but aggregate existing knowledge into a single knowledge base.

Centralization helps in sharing, but only if knowledge developed by others can be used and thus embedded in or connected to the platforms. This should not be limited to knowledge developed and maintained by law enforcement agencies.

As an example, many investigations benefit from integrating external data sources, such as Call Detail Records from mobile phone providers. This allows querying and filtering of such information directly in the context of a specific case, within the legal boundaries of the investigation.

5.2.1. Simplify tool integration

Hansken contains hundreds of tools for extracting and indexing data from digital media, which are generally implemented in Java. Example libraries are Firefli (file type recognition), Snorkel (file system structures), Tapir (wiretap data structures), Traces (file and database structures) and Jaguar (compressed file archives).

There are several commercial or open source tools and scripts from different communities that make it possible to analyze different types of digital material. Hansken is designed in such a way that it is possible to add the result of such third-party tools, implemented in other languages. Examples of such integrations are Tesseract[5] for finding text in graphics (OCR) and Apache Tika[6] for extracting text from many data structures.

For external tools that cannot integrate directly through an API, we have developed custom components to allow outputs (such as reports or logs) to be parsed and the results added to the Hansken trace store. Examples are Cellebrite Extraction Reports and XRY Reports.

The previously mentioned AFF4 and CASE formats provide tooling (in Python) to automate the handling of data stored and evidence expressed. This allows them to easily integrate through an API. Before any such standard-based data sharing is possible, Hansken internal objects must be mapped to these standards.

5.2.2. Simplify tool development

Although we have a clear and well-defined *tools API* for Hansken, developing a tool requires a lot of knowledge on how it is used in the platform, what inputs are provided and what outputs are required. To better support tool developers, a Tool Development Kit should be provided. Such a kit should provide tool/container templates, stub the platform and can be used to run, debug and test tools based on local data.

In addition, for technical contributors that lack software engineering skills, more options can be provided to lower the bar even further. These solutions range from comprehensive examples, tool templates or even tools designed specifically to minimize software engineering efforts, such as user interfaces (van den Bos and van der Storm, 2013) or domain specific languages (van den Bos and van der Storm, 2011) to describe digital traces that can automatically be converted into tools.

5.2.3. Share investigative components

A DFaaS implementation automates a large part of the trace extraction process and offers the results in a user-friendly and interactive way. The forensic analyses not automatically supported by the platform or existing tools, can be developed outside the platform (e.g., using scripts, tools or a domain-specific language (DSL) like Nugget (Stelly and Roussev, 2018; Stelly, 2019)). The knowledge captured in these scripts might be case-specific, but can also be valuable to other cases.

To share such annotated scripts (including performed research or other references), the DFaaS platform should be extensible with a reporting service, offering a *reporting API*. This API should support uploading and executing of scripts that generate reports.

5.3. Vision: Separate DF knowledge from software engineering

Developing forensically sound tools to recover traces is technically challenging and time-consuming work. In order to maximize productivity of domain experts in this field, concerns around forensic aspects, reverse engineering and software engineering should be separated.

This is achieved in Hansken through the use of our own domain-specific language (DSL) called Metal,[7] which allows binary data structures to be described declaratively. This means that the forensic expert only needs to describe the structure of a file format, protocol or memory layout without concern for run-time performance, memory allocation, scalability or any other engineering aspect.

Example structures that are declared in Metal are file systems (XFS), container files (ZIP), executable files (EXE), multimedia files (JPEG), memory layouts (BSD) and network protocols (TCP/IP).

These advantages compound as file formats often evolve and descriptions need to be adapted and modified continuously. New algorithms and improved implementations are automatically used on old and new descriptions. Additionally, using a single and implementation-independent approach to describing data structures means that complex operations involving nested data (e.g., in ZIP containers), fragmented data (e.g., in many file systems and network captures), pointers (e.g., in memory dumps) and any other form of data dependencies can be combined to recover data from exceptionally complex data streams without requiring extra effort.

Ideally, these descriptions can be viewed and edited at the top-level user interface by forensic experts in order to express any knowledge about a specific case and immediately change the underlying analysis tools. These changes can then be annotated and reviewed in order to automatically grow the database of low-level forensic knowledge.

5.4. Vision: DFaaS implementations must give insight in the context of traces

Singular traces rarely tell the whole story, digital forensic investigations often rely on a collection of traces and the relations between them. Since all traces are normalized and available via a well-defined API, the DFaaS concepts support evidence evaluation to a great extent. The current implementations already supports the querying, filtering and aggregation of traces. This, however, is not sufficient. Several evaluations can be automated to a great extent, especially those that are straightforward or well documented.

5.4.1. Correlate traces

Digital experts can manually find relations between particular traces (e.g., the target of a link file). Many of these correlations lend themselves well to automation as they can be resolved as a hit or miss, expecting either a single file to match the particulars stated in the link file or none.

Other relations between traces are less clear, like pictures that

[5] https://tesseract-ocr.github.io/

[6] https://tika.apache.org/

[7] https://github.com/parsingdata/metal

visually look like one another. Such correlations that convey a level of uncertainty lend themselves well to the use of artificial intelligence techniques. The result can take many forms, from a single number describing an entire data set to a matrix representing relations from every trace to many or all other traces. Preferably, such results are captured inside the platform, making them available to all its users.

For both 'hard' and 'soft' relations, DFaaS platforms need a way of serializing the relation, be it a one-to-one relation, grouping of traces, scoring of likeness or a custom relation between traces as defined by an end-user. Relation and correlation support should be modeled on established technologies (e.g., the Cypher or Gremlin graph query languages pioneered by the Neo4j graph database[8] and the Apache TinkerPop project[9] respectively). Supporting both a graph structure and the tree structure inherent to the current trace model, investigators could leverage a more rich query interface to either get a better understanding of the particulars of a data set or arrive at that level of understanding more efficiently than they otherwise could.

Artificial intelligence techniques often involve applying a large model, trained on known data. The platform should be equipped with a user-editable database of models that can be used during the extraction process. The same use case applies to other extraction processes that need 'large' resources for their task, for example a tool that looks up a file's hash in a database to mark it as a known interesting trace.

5.4.2. Automated evidence evaluation

By giving clear insight in all details of traces and making it possible to combine and aggregate those details, (case-specific) evidence evaluation is possible. Typical use cases are forensic reports in which digital traces are evaluated in the context of a case. A case investigator can for example start a report by identifying and linking a specific trace or query, including questions to a digital expert. The expert can finalize (and sign) the report by providing answers to the questions.

Next to case-specific evaluations, common analyses take place in many cases. For example, a report on clock evaluation, visualizing relevant traces with links to background literature and explanations on how to interpret the reported data. They capture forensic analyses in a user-friendly way, making it easily accessible and reusable across cases. Such evaluations can already be captured and reused outside the platform, for example by using scripts (see Section 5.2.3).

We propose to extend any DFaaS platform with dynamic reporting functionality to capture such (automated) reporting of digital evidence or (semi-automated) evidence evaluation. These reports can be based on templates that incorporate rich text, links to remote (scientific) resources and (aggregated) trace collections. Inclusion of such trace collections can be simplified by wrapping the REST APIs with a client-side visualization library.

This library can subsequently serve as a basis for a widget-oriented configurable web interface. Reports can then be created by dragging and dropping configurable widgets and serve as templates.

Also, an existing knowledge base system (like a wiki) can be extended with visualization widgets based on DFaaS platform queries. An advantage of this approach is that version control and user tracking is already implemented, automatically providing the chain of custody for such reports.

5.4.3. Support experiments

When evaluating digital evidence, the main questions to answer are what activity took place that caused the evidence. These activities typically have to be linked to or addressed to the suspects. Digital forensic evidence evaluation, however, is like shooting a moving target: Once evidence is evaluated for one case, it hardly ever can be directly used for other cases because the context of the evidence (hardware or software) differs or because of legal protections afforded the defendant. So, digital experts often have to execute experiments over and over to draw forensically sound conclusions. Such experiments can be supported by extending DFaaS platforms with functions to execute experiments in containerized environments. The experiments themselves can be captured and replayed using techniques for behavior-driven development (BDD) (Chelimsky et al., 2012) and automated testing of user interfaces.

Storing and sharing results of experiments via DFaaS platforms (i.e., the traces that relate to the execution of experiments) makes it possible to better evaluate the behavior of hardware and software.

5.4.4. Share system-independent data

Data that shows up during investigations but is not directly related to the behavior of the systems under investigation, can be automatically collected and provided via the DFaaS platform APIs. Typical examples of such reference data are traces that originate from remote resources like internet sites and can change over time (e.g., URL parameters, cookie parameters and security certificates).

5.5. Vision: A single DFaaS platform is not always sufficient

Within one organization, one platform implementation might not suffice (e.g., because storage is distributed over several physical locations, operations is distributed over multiple units, or the technology does not scale to the organization's needs). For such organizations, multiple platforms can grow into a cluster by interconnecting them, for example by proxying multiple clusters via a single point-of-entry. Based on properties inside the requests, it can be decided to handle the request at the local platform, or forward it to a remote implementation, transparent for the end-user. In this way and by sharing identity providers, it can implement a single sign-on for multiple platforms. Additional advantages are the collection of cross-platform statistics (use, users, performance, extraction), a single point of operation and investigation and cross-platform case analyses using scripting APIs.

5.6. Vision: The DF community needs to join forces

We are setting up a program to improve and extend the currently operational DFaaS platform Hansken. To support criminal investigations, the platform is available to law enforcement agencies. This includes the forensic tools and libraries[10] that implement the data structures and trace extractions. For research and development, the platform is available to forensic science institutes, universities (of applied science) and relevant joint research projects.

The platform is only one of the resources to share. The DFaaS Community also requires a joint communication platform and code repository. These make it possible to share and discuss user experiences like how-to's and do's-and-don't, but also share documentation, like best practice manuals and checklists. As mentioned before, code examples to share are operational scripts, remote

[8] https://neo4j.com/
[9] https://tinkerpop.apache.org/

[10] The NFI forensic libraries are licensed by the Netherlands Forensic Institute. Parts of these libraries are available to law enforcement agencies only.

connectors (API implementations), visualization components, as well as forensic tools and platform improvements and extensions.

The Netherlands Forensic Institute leads this DFaaS program focusing on 'platform scaling' and 'criminal justice chain implementation', named "OK Hansken",[11] under the supervision and instructions of a steering committee representing Dutch law enforcement and judicial organizations, funded by the Dutch Ministry of Justice and Security.

6. Related work

This paper completes the trilogy of papers on Digital Forensics as a Service by the NFI. In the first paper, 'DFaaS: a game changer' (van Baar et al., 2014), we introduced the DFaaS concepts and the way they impact the application of digital expertise in criminal investigations. In the second paper, 'DFaaS: game on' (van Beek et al., 2015), we explained the design principles on which the DFaaS implementation Hansken is built. Our vision is also based on the lessons learned while working on the predecessor of Hansken, named xiraf (Bhoedjang et al., 2012).

Next to our implementations, others have been working on large setups for combining digital forensic tools, for example Google Rapid Response Framework (GRR) (Cohen et al., 2011) and Scalable Realtime Forensics (SCARF) (Stelly and Roussev, 2017).

The DFaaS implementation Hansken has been evaluated before. The National Cybercrime Centre of the Norwegian National Criminal Investigation Service (Kripos) published an exhaustive evaluation report (National Cybercrime, 2018),[12] focusing on the technical testing of Hansken. Recently, Borhaug (2019) evaluated the DFaaS implementation in the Netherlands, providing insight in the current state and recommendations on how to advance the Hansken implementation. In 2020, a case study is presented (Bas Seyyar and Geradts, 2020) on the privacy impact of large-scale digital forensic investigations, based on the use of Hansken.

A lot of papers are published on the challenges and future of digital forensics. Here, we shortly summarize the main publications on these topics, viz. the growing amount, diversity and complexity of data and the growing need for forensic analysis of this data. In 2009, Beebe et al. (2009) summarized digital forensic science, providing an overview of research topics that were both addressed and unaddressed. Garfinkel (2010) presented in 2010 that the Golden Age of computer forensics was quickly coming to an end. He gave a nice overview of the challenges he foresaw in 2010. To identify future research challenges, Quick and Choo (2014) compared several studies (among which DFaaS) on the impacts of increasing volume of digital forensic data in 2014.

Casey et al. (2018) summarize the evolution of expressing and exchanging cyber-investigation information in a standardized form. Karie et al. (2018) discuss knowledge management as a strategic asset in digital forensic investigations. The results of this paper align with the key findings by Luciano et al. (2018) in 2018 and Casey et al. (2019) in 2019: the DF community must share knowledge, collaborate, simplify the adoption of tooling to the ever-changing digital material and work on standardizing the field. Verma et al. (2019) present an automated, privacy preserving and efficient digital forensic framework that leverages machine learning for evidence prediction and privacy evaluation. Montasari and Hill (2019) presented an overview of the challenges and paradigms in the near future of digital forensics. Srinivasan and Ferrese (2019) give a comprehensive overview of distributed (cloud) solutions for providing forensics as a service, including xiraf and Hansken. They concluded that the key factors that limit current solutions are the read speed while acquiring data.

7. Conclusions

We have been providing the DFaaS (van Baar et al., 2014) platform implementations xiraf (Bhoedjang et al., 2012) and Hansken (van Beek et al., 2015) to Dutch law enforcement agencies for close to a decade. This conclusion presents the path we followed to implement DFaaS in The Netherlands.

7.1. Introduce a new way of working and cooperating

Governmental organizations are typically organized in a bureaucratic way. Their focus on minimizing risks results in slow decision making. In contrast, the digital forensic field evolves rapidly, asking for a flexible and adoptive organization. A wrong or incomplete implementation undermines the adoption of the concepts and reduces the benefits. Therefore, the needs and requirements of all people involved must be taken into account when implementing a DFaaS platform. Because of the equality-of-arms principles, this group is not limited to law enforcement agents, but also includes prosecutors, lawyers and judges. All people involved have to break out of their customary processes, which inevitably generates resentment and resistance.

Also, a DFaaS implementation always requires a combination of software engineering and digital forensic knowledge. These two concerns should be separated to the greatest extent. Especially since both parts, software engineering and digital forensic knowledge, evolve at their own pace and regularly in many different directions.

We tailored an agile process that takes into account all stakeholders' needs and contributions, a task that took time and tenacity.

7.2. Implement a DFaaS platform

Both the costs for and benefits of implementing the DFaaS concepts are significant. When investigating digital material using a DFaaS platform, the biggest benefit we experience is the availability of digital evidence to case investigators. The tactical information that is directly available in the digital material is huge. Even without detailed forensic evaluation that is needed for a profound understanding of the material, case investigators can use this tactical information for investigative purposes. Of course, cooperation with digital experts is needed to explain the digital evidence in detail, and to sort out the origin of traces and their relations. Otherwise, they might jump to unwanted conclusions. Advanced training and good user interfaces help in reducing this risk. Also, the forensic concepts implemented in the platform help. The provenance is available and attached to the traces, making the reports acceptable in court.

We experience that although the advantages of using a DFaaS implementation are clear, the business case is hard to make. A DFaaS gives detailed insight in the operational costs for handling digital evidence, which are hard to compare to the current costs scattered over organizations, teams and individuals.

Deploying a DFaaS platform cannot be compared to installing local software. As in any forensic investigation, upgrades and updates of tools can both boost and frustrate investigations that highly depend on the availability of the platform. They not only use the platform for getting access to the material, but also administer the investigation in it, amongst others by annotating the evidence. Therefore, specialists are needed for setting up the infrastructure and base operating systems and installing, configuring and

[11] https://hansken.org/

[12] This evaluation report by Kripos (National Cybercrime, 2018) is available upon request via the corresponding author of this paper.

maintaining the platform software.

Since such a platform consists of several complex components, it is key to automate the deployments and upgrades, based on solid documentation. The same holds for operating the platform. We experienced that deploying and operating such platforms is a full-time job for a team of specialized professionals. In The Netherlands, these teams contain typically four to six people per Hansken implementation.

7.3. Keep improving

The DFaaS concepts implemented in Hansken mainly focus on extracting traces from digital material in a forensically sound way, and making them available to investigative teams within their legal context.

Singular traces rarely tell the whole story. Digital forensic investigations often rely on a collection of traces and the relations between them. Nowadays, artificial intelligence is helping investigators to quickly process large sets of traces by automating classification. DFaaS platforms can and should support such functions, as long as case investigators can identify results from such tools (i.e., which classification was obtained in which way).

A great benefit of a DFaaS platform is that all digital evidence is available in one collection. This makes it possible to automate parts of the evidence evaluation, especially those that are straightforward or well-documented, like clock validation. Such evaluations can be case-specific, but might also be generic and applicable to many cases. By extending the platform with dynamic reporting functionality, such knowledge can be captured too, supporting automated reporting.

7.4. Cooperate and share

As mentioned, DFaaS is all about sharing digital forensic knowledge.

We continuously enhance the implementation, based on the stakeholders' needs. Driven by the principle that we do not want to reinvent wheels, we adopt and adapt third-party technology to the digital-forensic context. Since new technology comes to market every day and thus shows up in criminal investigations, it is impossible for a monolithic platform to support all case-specific needs. DFaaS focuses on unbundling the digital forensic tasks. As such, knowledge developed by others can be easier embedded in and connected to the platform. This is accelerated by supporting international standards for storing data and representing evidence.

To simplify tool integration, APIs should be based on widely supported standards and made publicly available. To better support and promote the centralization of knowledge in DFaaS platforms, a Tool Development Kit should be provided. Also, the platform must provide functionality to share knowledge for investigative or operational purposes at any level, for example scripts to automate specific tasks.

To maximize the sharing of digital forensic knowledge and to bundle innovation capacity, We need to join forces and build a wide community of law enforcement agencies, forensic science institutes, universities (of applied science) and relevant joint research projects. Members of the community should cooperate by sharing forensic knowledge and jointly advance the DFaaS concepts and implementations. To make this possible, the Netherlands Forensic Institute leads a DFaaS program focusing on 'platform scaling' and 'criminal justice chain implementation', named "OK Hansken", under the supervision and instructions of a steering committee representing Dutch law enforcement and judicial organizations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank all colleagues that made and make it possible to provide Digital Forensics as a Service by putting energy in xiraf and Hansken since 2005. This involves co-workers including hired staff, among others at the Dutch National Police, the Dutch Tax Fraud Investigation Service, the Dutch Prosecutors Office and the Netherlands Forensic Institute. We also thank the peer reviewers for their valuable comments and very constructive suggestions.

References

van Baar, R., van Beek, H., van Eijk, E., 2014. Digital forensics as a service: a game changer. Digit. Invest. 11 (Suppl. 1), S54–S62. https://doi.org/10.1016/j.diin.2014.03.007 proceedings of the First Annual DFRWS Europe.

Bas Seyyar, M., Geradts, Z., 2020. Privacy impact assessment in large-scale digital forensic investigations. Forensic Sci. Int. Digit. Invest. 200906. https://doi.org/10.1016/j.fsidi.2020.200906. http://www.sciencedirect.com/science/article/pii/S2666281720300263.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D., 2001. Manifesto for Agile Software Development. http://www.agilemanifesto.org/.

Beebe, N., 2009. Digital forensic research: the good, the bad and the unaddressed. In: Peterson, G., Shenoi, S. (Eds.), Advances in Digital Forensics V. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 17–36.

van Beek, H., 2018. A forensic visual aid: traces versus knowledge. Sci. Justice 58 (6), 425–432. https://doi.org/10.1016/j.scijus.2018.08.006.

van Beek, H., van Eijk, E., van Baar, R., Ugen, M., Bodde, J., Siemelink, A., 2015. Digital forensics as a service: game on. Digit. Invest. 15, 20–38. https://doi.org/10.1016/j.diin.2015.07.004 special Issue: Big Data and Intelligent Data Analysis.

Bhoedjang, R., van Ballegooij, A., van Beek, H., van Schie, J., Dillema, F., van Baar, R., Ouwendijk, F., Streppel, M., 2012. Engineering an online computer forensic service. Digit. Invest. 9 (2), 96–108. https://doi.org/10.1016/j.diin.2012.10.001.

Borhaug, T.S., 2019. The Paradox of Automation in Digital Forensics. Master's thesis. NTNU.

van den Bos, J., van der Storm, T., 2011. Bringing domain-specific languages to digital forensics. In: 33rd International Conference on Software Engineering (ICSE'11). ACM, pp. 671–680.

van den Bos, J., van der Storm, T., 2013. TRINITY: an IDE for the matrix. In: 29th IEEE International Conference on Software Maintenance (ICSM'13). IEEE, pp. 520–523.

Casey, E., Back, G., Barnum, S., 2015. Leveraging cybox to standardize representation and exchange of digital forensic information. Digit. Invest. 12, S102–S110. https://doi.org/10.1016/j.diin.2015.01.014 dFRWS 2015 Europe. http://www.sciencedirect.com/science/article/pii/S1742287615000158.

Casey, E., Barnum, S., Griffith, R., Snyder, J., van Beek, H., Nelson, A., 2017. Advancing coordinated cyber-investigations and tool interoperability using a community developed specification language. Digit. Invest. 22, 14–45. https://doi.org/10.1016/j.diin.2017.08.002. http://www.sciencedirect.com/science/article/pii/S1742287617301007.

Casey, E., Barnum, S., Griffith, R., Snyder, J., van Beek, H., Nelson, A., 2018. The evolution of expressing and exchanging cyber-investigation information in a standardized form. In: Handling and Exchanging Electronic Evidence Across Europe. Springer, Cham, pp. 43–58. https://doi.org/10.1007/978-3-319-74872-6_4.

Casey, E., Ribaux, O., Roux, C., 2019. The kodak syndrome: risks and opportunities created by decentralization of forensic capabilities. J. Forensic Sci. 64 (1), 127–136. https://doi.org/10.1111/1556-4029.13849 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/1556-4029.13849. https://onlinelibrary.wiley.com/doi/abs/10.1111/1556-4029.13849.

Chelimsky, D., Astels, D., Dennis, Z., 2012. The RSpec Book.

Cohen, M., Schatz, B., 2010. Hash based disk imaging using aff4. Digit. Invest. 7, S121–S128. https://doi.org/10.1016/j.diin.2010.05.015 the Proceedings of the Tenth Annual DFRWS Conference. http://www.sciencedirect.com/science/article/pii/S1742287610000423.

Cohen, M., Bilby, D., Caronni, G., 2011. Distributed forensics and incident response in the enterprise. Digit. Invest. 8, S101–S110. https://doi.org/10.1016/j.diin.2011.05.012 the Proceedings of the Eleventh Annual DFRWS Conference. http://www.sciencedirect.com/science/article/pii/S1742287611000363.

Court of Amsterdam, 2018. Judgment of 19 April 2018, Tandem II, ECLI:NL:RBAMS:2018:2504. http://deeplink.rechtspraak.nl/uitspraak?id=ECLI:NL:RBAMS:2018:2504.

Court of Gelderland, 2019. Judgment of 26 June 2019, Bosnië, Subs Brandberg,

IJshamer, Maan, ECLI:NL:RBGEL:2019:2832. http://deeplink.rechtspraak.nl/uitspraak?id=ECLI:NL:RBGEL:2019:2832.

Garfinkel, S.L., 2009. Automating disk forensic processing with sleuthkit, xml and python. In: 2009 Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering, pp. 73–84. https://doi.org/10.1109/SADFE.2009.12.

Garfinkel, S.L., 2010. Digital forensics research: the next 10 years. Digit. Invest. 7, S64–S73. https://doi.org/10.1016/j.diin.2010.05.009 the Proceedings of the Tenth Annual DFRWS Conference. http://www.sciencedirect.com/science/article/pii/S1742287610000368.

Garfinkel, S., 2012. Digital forensics xml and the dfxml toolset. Digit. Invest. 8 (3), 161–174. https://doi.org/10.1016/j.diin.2011.11.002. http://www.sciencedirect.com/science/article/pii/S1742287611000910.

Hamilton, C., 2013. Communicating for Results: A Guide for Business and the Professions. Cengage Learning, Boston.

Karie, N.M., Kebande, V.R., Swaziland, K., 2018. Knowledge management as a strategic asset in digital forensic investigations. Int. J. Cyber-Secur. Digital Forensics 7 (1), 10–20.

Luciano, L., Baggili, I., Topor, M., Casey, P., Breitinger, F., 2018. Digital forensics in the next five years. In: Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018. ACM, New York, NY, USA, pp. 46:1–46:14. https://doi.org/10.1145/3230833.3232813 doi:10.1145/3230833.3232813.

Montasari, R., Hill, R., 2019. Next-generation digital forensics: challenges and future paradigms. In: 2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3), pp. 205–212. https://doi.org/10.1109/ICGS3.2019.8688020.

NCIS National Cybercrime Centre, 2018. Technical Testing of Hansken. Tech. rep.. Kripos, Norway. available upon request via the corresponding author of this paper.

ISO 17025:2017: General Requirements for the Competence of Testing and Calibration Laboratories, Standard, Nov. 2017. International Organization for Standardization, Geneva, CH.

ISO 21043-1:2018: Forensic Sciences – Part 1: Terms and Definitions, Standard, Aug. 2018. International Organization for Standardization, Geneva, CH.

ISO 21043-2:2018: Forensic Sciences – Part 2: Recognition, Recording, Collecting, Transport and Storage of Items, Standard, Aug. 2018. International Organization for Standardization, Geneva, CH.

ISO 27037:2012: Information Security – Security Techniques – Guidelines for Identification, Collection, Acquisition and Preservation of Digital Evidence, Standard, Oct. 2012. International Organization for Standardization, Geneva, CH.

ISO 27042 2015: Information Security – Security Techniques – Guidelines for the Analysis and Interpretation of Digital Evidence, Standard, Jun. 2015. International Organization for Standardization, Geneva, CH.

Quick, D., Choo, K.-K.R., 2014. Impacts of increasing volume of digital forensic data: a survey and future research challenges. Digit. Invest. 11 (4), 273–294. https://doi.org/10.1016/j.diin.2014.09.002.

Schatz, B.L., 2019. Aff4-l: a scalable open logical evidence container. Digit. Invest. 29, S143–S149. https://doi.org/10.1016/j.diin.2019.04.016. http://www.sciencedirect.com/science/article/pii/S1742287619301653.

Schatz, B., Cohen, M., 2017. AFF4 Standard v1.0. Github. https://github.com/aff4/Standard/blob/master/AFF4StandardSpecification-v1.0.pdf.

Srinivasan, A., Ferrese, F., 2019. Forensics-as-a-service (Faas) in the State-Of-The-Art Cloud, Security, Privacy, and Digital Forensics in the Cloud, p. 321.

Stelly, C.D., 2019. A Domain Specific Language for Digital Forensics and Incident Response Analysis. Ph.D. thesis. University of New Orleans.

Stelly, C., Roussev, V., 2017. Scarf: a container-based approach to cloud-scale digital forensic processing. Digit. Invest. 22, S39–S47. https://doi.org/10.1016/j.diin.2017.06.008. http://www.sciencedirect.com/science/article/pii/S1742287617301950.

Stelly, C., Roussev, V., 2018. Nugget: a digital forensics language. Digit. Invest. 24, S38–S47. https://doi.org/10.1016/j.diin.2018.01.006.

Verma, R., Govindaraj Dr, J., Chhabra, S., Gupta, G., 2019. Df 2.0: an automated, privacy preserving, and efficient digital forensic framework that leverages machine learning for evidence prediction and privacy evaluation. J. Digit. Forensics Secur. Law 14 (2), 3.