OSSDIP: Open Source Secure Data Infrastructure and Processes Supporting Data Visiting

Martin Weise[1✉], Filip Kovacevic[1], Nikolas Popper[1], and Andreas Rauber[1]

[1]TU Wien, Austria
✉martin.weise@tuwien.ac.at

September 6, 2022

Abstract

Meeting the conflicting goals of protecting and maintaining control over sensitive data while also allowing access by third parties constitutes a significant challenge. Secure data infrastructures support data visiting in a highly controlled and monitored environment which, if properly set-up and operated, provide high security guarantees through a combination of technical, legal and procedural mechanisms. To ease the process of deploying such a secure data infrastructure, we present a detailed documentation of the architecture and processes of such an infrastructure and provide a pre-configured reference implementation based entirely on open source software that can be flexibly configured to meet differing security requirements and deployment scenarios.

We combine mechanisms for data visiting on secured infrastructure components with optional components of data anonymization and fingerprinting, covered by extensive logging and monitoring functions and embedded in defined processes and contractual frameworks. The set-up is based upon the experience of operating such a secure infrastructure in the medical domain for almost ten years, addressing the emerging need to make such a solution available to a larger set of stakeholders. We show that our system significantly enhances data visiting, offers a higher level of data isolation and present our open source reference implementation thereof.

Keywords— Data Visiting, Data Leakage, FAIR Data, Isolated Nodes, Secure Data Infrastructure

1 Introduction

In an increasing number of settings, both researchers in academia as well as stakeholders in industry need to safeguard access to highly sensitive data, e.g. due to privacy requirements or commercial sensitivity of data, while still wanting to make it accessible to third parties for research purposes or to assist with specific analytical tasks. This challenge has become acute during the early days of the COVID-19 pandemic when evidence-based decision making required the analysis and sometimes integration of highly sensitive (due to privacy or commercial reasons) information such as health data, social science data, movement data from telecommunications operators, or supply-chain logistics data from the retail sector. But even outside this exceptional situation, academia–industry collaborations as well as industry-to-industry cooperations frequently are hindered by the conflicting needs to keep the data secret that the other party should process or analyze. Homomorphic encryption, while ensuring that the data is kept hidden from the analyst, does not sufficiently support the exploratory and interactive types of analyses required in many settings, which frequently require an analyst to actually see the data and interpret instance-level characteristics and attribute semantics.

While data sharing is being proclaimed as the future in open science, many settings do not allow for such approaches be it due to privacy concerns, confidentiality agreements, or inherent risks to core business activities in case data were extracted and became available to e.g. companies. Data visiting, on the other hand, is an approach where data stays under the control of the owner and allows the consumers (e.g. analysts or machine learning algorithms) to come to the data to work with it. Closely monitoring the processes and interaction with data during these visits allows to put a certain level of safe-guards in place to prevent accidental data leakage or intentional data breaches. However, most research infrastructures provide data visiting support only on a very rudimentary level, sometimes even failing to prevent the (even unintended) export of parts of data via simple file transfer mechanisms when analysis results containing parts of the data are downloaded.

We define and document a system architecture and provide a modular pre-packaged configuration of components constituting the core of such a secure data infrastructure as reference implementation. Selecting only well-known and tested open source components allows us to minimize risks of data-theft while at the same time focusing on the main research challenge: design a secure system and identifying compulsory and optional components, as well as processes needed to operate it. Contractual obligations that need to be defined are discussed as part of the infrastructure set-up to complement the technical measures. The infrastructure is initially configured to run as a trusted third-party environment, with a subset of this configuration being suitable for deployment within a data owner's environment. It is the result of a second iteration of the initial handcrafted solution using the experience gathered during the set-up and operation of a trusted third party data platform in the health sector operating as a joint initiative of our institutional IT services, a national research center on IT security and technical expertise of the people involved, findings of internal audits and user feedback. This paper presents the architecture of the resulting secure data infrastructure, discusses design decisions and touches upon the processes complementing the system set-up. Additionally, it presents a condensed summary of the risk factors involved and ideas for modular additions to enhance the level of security in the system. We do not cover general aspects of IT infrastructure security which obviously need to be considered in any IT infrastructure operation, but rather focus on avoiding data leakage occurring most likely from ignorance and accidental disclosure. We also would like to stress, that in terms of data leakage, e.g. unauthorized exfiltration of data from the system, the focus is less on a researcher maliciously trying to establish a covert channel to steal data, but rather to prevent accidental data loss, the circumvention of data pseudonymization safeguards via data linkage on its use for unauthorized purposes. This observation is also shared by other infrastructures providing research access to sensitive data, blaming a combination of enthusiasm and ignorance for attempts to bypass output checks or using data in unauthorized contexts (Dood, 2020).

The remainder of the paper is structured as follows: Sec. 2 describes approaches and similar infrastructures that enable secure data management, Sec. 3 reviews the levels of control, the selected virtualization approach to create isolated environments and organizational measures before introducing the core infrastructure components. We present our controls for safeguarding the infrastructure, Sec. 4 which consists of roles (e.g. Data Owner, Data Provider and Analyst) and controlled access, data segmentation, network segmentation, process automation and monitoring. The standard processes needed to transparently operate important tasks in the secure data infrastructure are presented in Sec. 5. We further discuss the limitations and possible weaknesses of our approach and give an overview of possible extensions in Sec. 6.

2 Background

Specifying the technical and organizational boundaries of systems that enable governments, academia and businesses to use highly sensitive data is an ongoing field of research. The need to provide secured compute services in a cloud setting, with clear segmentation of the underlying network has been recognized for a long time (Hao et al., 2010). Using open source tools, techniques and procedures a secure container infrastructure can be created with moderate effort given the right guidance. We elaborate on necessary principles for trusted data infrastructures in Sec. 2.1 and introduce similar data visiting infrastructures in Sec. 2.2. In Sec. 2.3, we align these with secure enclaves.

2.1 Principles for Trusted Data Infrastructures

Practical guidelines (Akula, 2019) ease the process of constructing a secure environment by addressing a wide range of security dimensions with and – as one of the key contributions of this paper – reference implementations lowering the entrance barrier for providing secure compute platform environments. There are many data management solutions that address effective decision making in the context of preventing unintended disclosure of sensitive information. The "fives safes" dimensions (Desai, Ritchie, and Welpton, 2016) maximize the usage of detailed public records while at the same time protecting personal rights of individuals. Splitting decisions into five dimensions allows a data management solution to protect the overall confidentiality but gives enough flexibility to tailor some specific dimensions to stronger security than others. In the following, we give a short overview on the dimensions since we address them through technical enforcement and organizational processes in the main body (c.f. Sec. 5) of this paper: (i) *safe projects* address management decisions regarding appropriateness of the usage of the data through auditability and review processes, (ii) *safe people* identifies individuals that access the sensitive data and require them to sign legally binding terms of use, (iii) *safe data* ensures appropriate data de-identification and access capabilities with respect to the research questions formulated, (iv) *safe settings* addresses the necessity of security and transparency to achieve trust with the public and data owners, (v) *safe outputs* ensures only approved, aggregated research results can be exported out of the system.

Taking this into consideration, the UK Health Data Research Alliance describes "a strategy to build public trust and meet changing health data science needs" in their green paper on trusted research environments (TREs) (United Kingdom Health Data Research Alliance, 2020). The original requirement for *safe setting* is extended to address outsourcing of computing infrastructure to other parties and maintaining inaccessibility to the sensitive data by these. The environment that holds the sensitive data must implement a barrier to the outside world. Further, they extend the original framework by allowing a *safe return* of the results produced from the processing of data back into the trusted research environment through a mapping mechanism. It must ensure that in case of de-identified subsets of data, the re-identification always perfectly maps back to the individuals where the data originated from not to poison the original data set, but to enhance the data set with e.g. research artifacts.

2.2 Data Visiting Infrastructures

A number of discipline-specific and national data infrastructures have emerged during the past few years, that implement and serve as inspirations and best-practice guidelines for the concepts listed above. Due to data leakage concerns, research data centers may enable researchers local data access only. In a connected, global community of researchers and practitioners this in almost all cases, results in unacceptable data visiting conditions for sensitive data.

The `Kadi4Mat` research data infrastructure (Brandt et al., 2021) provides tools and workflows as a service for analysts in material science. It uses workflows that the analyst creates to provide an automated data processing pipeline covering analysis, visualization or transport within one of multiple process engines. The concept of a process engine is described as an executor for a specific task on behalf of the user within the workflow. This allows the analyst to have a reproducible result and creates metadata along the way that is collected and recorded. Their web-based application follows a standard client-server architecture and a *PostgreSQL*[1] database to make the metadata available in their repository. Contrary to the system presented, our approach uses temporary virtual machines that are isolated from each other, as well as an air-gapped data node holding sensitive data, also improving the usability of the data since researchers are provided with instant feedback instead of job submission-systems. To the best of our knowledge, we were not able to determine how `Kadi4Mat` protects sensitive data in the infrastructure.

The `RemoteNEPS` (Skopek, Koberg, and Blossfeld, 2016) system allows remote data access through web technology in a secure environment. The National Education Panel Study (NEPS) hosts a secure data infrastructure capable of handling 50 user simultaneously at a machine cluster consisting of 72 physical cores and 1.344 TB random access memory. Their technical approach is to use a web browser with *Java*[2] plugin as client software to connect with the secure data infrastructure through a remote desktop server and allows no Internet access within the desktop session (Barkow et al., 2011). They extend conventional approaches like remote execution or job-submission systems e.g. `Kadi4Mat` that have input- and possibly output queues, since the output is immediately present on the screen of the user. Their infrastructure uses Active Directory Services[3] and biometric key-stroke authentication for each new login attempt. The analyst can use commercial data science- and text processing tools, as well as editors in the Windows Desktop[4] environment. During publication, their system was in production for four years and served more than 200 users. Our approach similarly provides the analyst with a remote environment and data science tools, but only considers open-source operating systems and -software with intent to make it available to as many institutions as possible, without licensing processes. After configuration, our infrastructure also supports commercial software. Also our approach allows approved connections to the open Internet to e.g. allow usage of proprietary software that depends on license server connections.

The `DEXHELPP` infrastructure (Popper et al., 2017) has been operational in Austria for almost 10 years. To facilitate research, it creates a secure and controlled environment where data owners can deposit their data, after which analysts can perform their analysis and experiments within that environment without the need to transfer the data outside of the system. Data providers can specify fine-grained access rights to individuals or groups of analysts, to entire data sets or just specific subsets thereof, e.g. limiting the number of records, or excluding specific details of records. The access of analysts to these sources is accurately recorded, which allows for auditing and inspection of the intended usage of the data. One important aspect of the system is the trade-off between the controlled environment and the choice and offer of modeling and programming tools available to the analysts. `DEXHELPP` tackles this by providing the analysts with a wealth of commonly used tools, the requirements for which were elicited by observing current practices. Further, the server environment offers a fast computing environment, with special hardware such as graphical computing (GPU) available on demand and is also a suitable environment (on a dedicated research server where the data is held in an encrypted vault) to merge and link data sets from different sources, which otherwise

[1] "PostgreSQL". [Online]. URL: `https://www.postgresql.org/`, accessed 2021-12-22

[2] "Java". [Online]. URL: `https://www.java.com/`, accessed: 2021-12-22

[3] "Active Directory Services". [Online]. URL: `https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/dd578336(v=ws.10)`, accessed 2021-08-02

[4] Windows OS. [Online]. URL: `https://www.microsoft.com/en-us/windows`, accessed 2021-08-03

would not be released. Compared to RemoteNEPS and Kadi4Mat, it offers an open source infrastructure with isolated desktop where the analyst authenticate against the researcher environment with a time-based one-time password instead of a biometric method.

Similar infrastructures have recently been set-up in different countries particular in the health sector, such as e.g. SAIL Databank (Jones et al., 2014), ePouta (Palmgren et al., 2019), French Health Data Hub (Cuggia and Combes, 2019), OpenSAFELY (Williamson et al., 2020), etc. The infrastructures referenced provide a system design on an abstract level, offering little technical guidance on the components or configuration. Inspired by these and similar models we present the Open Source Secure Data Infrastructure OSSDIP (Weise and Rauber, 2021), specifically re-using experience gained from DEXHELPP as well as personal communication which several other operators of secure infrastructures. We extend DEXHELPP by distributing core infrastructure components across multiple nodes (c.f. Sec. 3.4) in aim to provide well-defined interfaces and multiple layers of security (c.f. Sec. 3). It follows the same core infrastructure set-up that we document at a fine granular level of detail to allow other institutions to set up a copy within their own premises. It is designed to be able to be hosted by an external third party provider and thus needs to document and establish trust both towards the data owner as well as the analysts wanting to work with the data. Frequently, it will be hosted by the data owner (although it can also be hosted externally), thus collapsing those two roles and easing/eliminating certain process steps.

In the remainder of this paper we only consider the case where the data owner is the infrastructure carrier. Experience shows that the perceived technical burden of getting started with such a technical infrastructure in institutions is often too large to embark on the mission to establish a secured data visiting infrastructure. As a consequence of this, a lot of sensitive data that cannot be shared with research has to be kept inaccessible for large user communities or can only be provided in anonymized and/or highly abstracted levels of aggregation, also reducing its value for research. Presenting a sound system architecture together with straightforward processes and a reference implementation is needed that can be easily deployed for evaluation purposes, as presented in this paper, should ease the adoption of data visiting infrastructures and thus make research data more widely accessible while allowing the data owner to retain full control over their data. Our contribution reduces the technical burden by providing institutions with a automatized deployment of a reference implementation that requires little resources for proof-of-concept deployments, but can be scaled to production deployments through configuration files before deployment.

2.3 Secure Enclaves

Secure enclaves are infrastructures built for secure and confidential computing and follow the principle of a *trusted computing base* (Kostiainen, Dhar, and Capkun, 2020), where the extent to which software and hardware that needs to be trusted is reduced to a minimum level for a particular task. Oftentimes users can run applications and work with sensitive data while ensuring higher security and privacy degrees. In the context of academic research computing (Peisert, 2021), a secure enclave describes a secure computing infrastructure to tackle the problem of data confidentiality through technical, administrative and procedural solutions. Campus secure computing enclave systems are either bought as strategic investment or are the result of in-house efforts to provide such an environment.

The principle of a *trusted computing base* can also be fulfilled by hardware-based *trusted execution environments* (TEEs) that is a hardware element implemented in e.g. the central processing unit of chip manufacturers[5][6][7]. Since these hardware-based TEEs might increase the initial impediment to start using a secure enclave, efforts towards providing hardware-independent implementations thereof are coming forward. Through using simple abstractions (physical memory protection, security monitor, etc.) provided by the hardware while also allowing platform-specific features, Keystone (Lee et al., 2020) can be a open-source secure enclave that allows customization from the hardware manufacturer, hardware operator and the enclave programmer. The Open Enclave SDK[8] aims for a similar goal.

In Table 1 we give an overview on common features of secure enclaves (Peisert, 2021) to later align them with our own methodology. Features on the *physical level* consist of dedicated hardware co-processors or system-wide bus-address filters, to separate secure from memory partitions and provide secure processors with isolated memory containers for applications. A simple mechanism on this level can also be a human-operated "airlock" that requires two trusted operators to connect a privileged storage to the analyst machine. On the *network level*, a secure enclave might support virtual private networking (VPN) for connection to the infrastructure and encrypted data transfer through secure copy or other file transfer protocols. At a *workstation level*, a secure enclave might allow users to connect to the execution environment via remote desktop. On this level further control mechanisms can be deployed, such as

[5] "Arm TrustZone Technology". [Online]. URL: https://developer.arm.com/ip-products/security-ip/trustzone, accessed 2021-12-22

[6] "Intel Software Guard Extensions". [Online]. URL: https://www.intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions.html, accessed 2021-12-22

[7] "AMD Secure Encrypted Virtualization". [Online] URL: https://developer.amd.com/sev/, accessed 2021-12-22

[8] "Open Enclave SDK". [Online]. URL: https://openenclave.io/sdk/, accessed 2021-12-22

Physical Level	Dedicated Hardware co-Processors
	System-wide Bus-Address Filters
	Trusted Execution Environments
	"Airlocks" with Two-Person Rules
Network Level	Virtual Private Networking
	Time-based One-time Passwords
	Encrypted Data Transfer
Workstation Level	Remote Desktop
	Access Control
Data Level	Encryption (at rest)
	Homomorphic Encryption
	Pseudonymization
	Anonymization
	Differential Privacy

Table 1: Secure Enclave Features

recording and storing every user action, managing privileges with access control models and disabling cut/copy/paste operations from the remote machine to the local machine to prevent data flowing off through a tunnel.

Secure enclaves may provide encryption models for data at rest on the *data level*. Major database management systems focus on securing data at rest with homomorphic encryption. This approach still leaves room for the adversary to physically access the data, which is why encryption for data in use in cloud environments has also been proposed (Sidorov and Ng, 2015). Secure enclaves can provide features at data level by adding data de-identification techniques to remove the parts of the data that are deemed most sensitive trough anonymization, pseudonymization and/or differential privacy (Garfinkel et al., 2015). While the former two methods do not change the aggregate of the data set, the latter one does by releasing statistical information about the data. How well-de-identified the data actually is or how hard it is for an adversary to re-identify certain records can be measured by using the k-anonymity (Sweeney, 2002), l-diversity (Machanavajjhala et al., 2007) and t-closeness (Li, Li, and Venkatasubramanian, 2007) criteria.

Having these features in mind, we not aim for protection in the physical level in `OSSDIP`, since we assume a trusted hypervisor and execution environment to be present on each of the nodes either through technical-, organizational- or legal obligations between the organization providing the resources and the organization operating the infrastructure. Network level features are also not the main contribution of `OSSDIP`, although the system is compromised of best-practice software that provides VPN, time-based two-factor authentication and encrypted data transfer. From the workstation level onwards, our system differs from secure enclaves like `Keystone` since `OSSDIP` focuses on preventing accidential data loss through ignorance rather than an malicious insider.

3 System Architecture

We give an overview of the architecture in Sec. 3.1, the prerequisites in Sec. 3.2 and introduce the concept of isolated virtual machines in Sec. 3.3 that support organizational measures presented in Sec. 3.4.

3.1 Overview

Data breaches are a key security issue in modern computing and have a multitude of root causes (Mousa, Karabatak, and Mustafa, 2020). We aim at ensuring ongoing confidentiality through technical and organizational measures and establish integrity with our five controls (see Sec. 4). Availability is provided through deploying the system on commodity server hardware using standard tools to establish a secure connection from the open Internet, and resilience through using only best-practice open source software.

A secure system needs to deploy security controls that target every technical and organizational aspect specific to the setting of secured data visiting. We address the security of processing sensitive data by architectural design and automated decision making on behalf of stakeholders. The provider of the secure data infrastructure in legal terms acts as data processor. The actual ingress of sensitive data is initiated by the data owner (see Sec. 4.1 for detailed role definitions) as is the egress. Our secure data infrastructure stores the sensitive data in a data node that has a strict firewall barrier around it. Only process-approved connections to selected Virtual Machines[9] (VMs) for data

[9] We refer to the various computational nodes as virtual machines as this is the way these are provided in our reference implementation to allow easy deployment of test set-ups. For real life deployment, these will frequently be deployed an individual physical machines to enable proper air-gapping of central nodes, allow the assignment of the Administrator role to different persons to avoid privilege concentration.

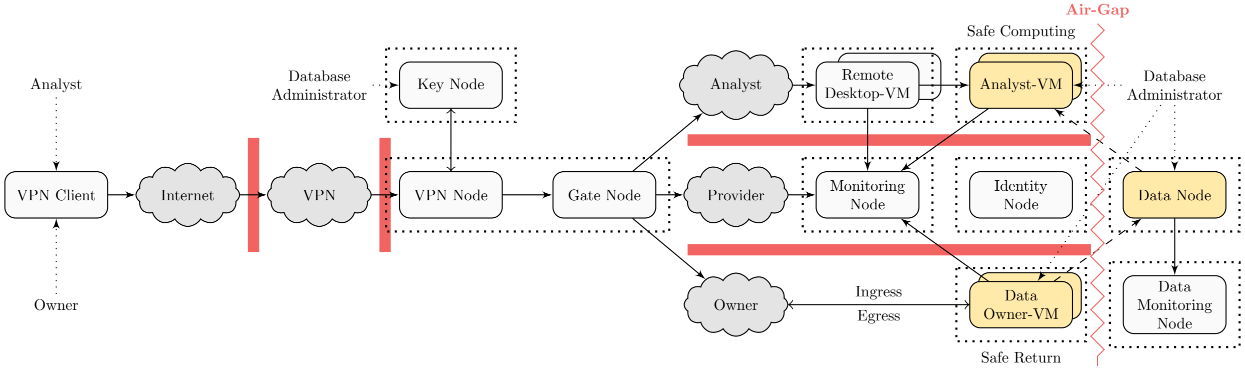


Figure 1: The multiple security layers in our reference implementation. Components in golden color contain sensitive data anytime, red bars are restricted firewall barriers. Dotted boxes denote physical servers on which nodes can be deployed or virtualized (the VPN Node and Gate Node can share a physical server).

import or to provide an isolated copy of the data, as well as for maintenance and monitoring, are allowed to pass this barrier.

The overall concept is centered around the principle of never providing access to the data node where all data is being held. For each individual analysis request, the specific subset of the data required is extracted from the central data store and copied onto a dedicated compute VM (Analyst-VM, through the rest of the paper we capitalize core infrastructure nodes c.f. Sec. 3.4) together with the tools required to perform the analysis. Access to this Analyst-VM is granted to the (single) analyst working on the task at hand - however, never directly, but only via a dedicated Remote Desktop-VM to introduce a media break and avoid any data flowing off via e.g. a tunnel. Thus, an analyst can establish a remote desktop connection to a dedicated VM from which solely a secure shell connection (SSH) to the corresponding Analyst-VM can be established, holding a copy of only the subset of data (possibly finger-printed and aggregated) as well as the tools required for addressing the task at hand. Export of any result files (trained models, figures, charts) is possible only via a dedicated temporary Data Owner-VM via which the approval of data owner is obtained. An overview of this architecture is depicted in Figure 1 and described in further detail below. These Analyst-, Remote Desktop- and Data Owner-VMs are being destroyed after a specific transfer or analysis task is completed.

3.2 Organizational Measures

Similar to the holistic approach to TREs (United Kingdom Health Data Research Alliance, 2020) introduced earlier, we argue that only technical enforcement is not enough to keep sensitive data confidential for a multitude of reasons: (i) establishing awareness that certain operations require more conscious decisions than others and may have unintended consequences attached to it; (ii) accountability which enables a transparent communication of important processes in the system and making actors of the processes responsible for certain steps of the processes; (iii) legally binding terms of use to allow the processing of personal information, non-disclosure agreements, data access agreement, etc. Since we use well-known open source software and provide the Analyst with standard data science tools, we improve the status-quo since little to no additional training should be required to use `OSSDIP`.

3.3 Prerequisites

Below, we provide an in-depth description of the various components of the `OSSDIP` infrastructure and our reference implementation. This implementation is available as supplemental material at the end of this paper and optimized for easy deployment. Specifically, all `OSSDIP` components can be deployed almost fully automatically using Ansible[10] playbooks. To achieve this, all nodes are created as virtual machines on at least two hosts in our reference implementation. For production deployments, at least three dedicated physical servers (Data Node, Monitoring Node and a node containing the rest) should be used. It demonstrates the design decisions required to determine in which way such a data visiting set-up could be provided. A detailed description of the set-up process is provided in Sec. 4.4.

[10] "Ansible is Simple IT Automation". [Online]. URL: `https://www.ansible.com/`, accessed 2021-07-27

3.4 Core Infrastructure Components

In our reference implementation we use a single Virtualization Host that provides the necessary resources for the virtualized components, specifically the five core infrastructure nodes plus dedicated, temporary VMs for data delivery and for checking safe returns or result exports by data owners, and both a Remote Desktop-VM and Analyst-VM combination temporarily per task per analyst.

VPN Node is the endpoint for the analyst and data owner to establish a connection to the secure data infrastructure. We run a standard OpenVPN Access Server[11] implementation with the recommended `AES-256-CBC` cipher without compression on UDP port 1194, that is installed automatically by the set-up playbook.

Gate Node is the firewall that manages the traffic between the networks in the infrastructure. We use the pre-installed *firewalld* software of Rocky Linux[12] for this task. It allows to filter packets based on allow/deny chains that manage the traffic from analyst/data owner to the respective subnets.

Data Node is the central storage that holds sensitive data and can be isolated to a level that only a system administrator is able to access the node for maintenance, but not to make changes to the database engine running on it (c.f. Sec. 4.1). Our reference implementation runs a *MariaDB* server in a virtual machine (Data-VM) that implements the Research Data Alliance (RDA) recommendations for dynamic data citation via temporal tables, c.f. Sec. 4.2. In a production deployment, however, this node should be a dedicated physical machine to not expose the sensitive data to a potentially compromised hypervisor.

Identity Node is a component that offers a directory service to manage Data Owner-VMs and Analyst-VMs. We use OpenLDAP[13] for this task. Internally, it keeps track of all user credentials to access the VPN Node and contact possibilities (e.g. e-mail, telephone number, full name), as well as the metadata that, depending on the configuration, can be publicly exposed via an external node to fulfill transparency requirements and support the FAIR (Findable, Accessible, Interoperable, Reusable) principles for sensitive/closed data (currently under development).

Monitoring Node runs the monitoring endpoint that stores all events that are occurring in the secure data infrastructure. All monitoring activity is collected here and saved in audit trails. This component should (ideally) operate using a hardware-protected write-once storage system, and have a separated access control, but is virtualized in the reference implementation to provide a self-contained starting point using append-only logs[14]. Two separate nodes are used for monitoring to allow the Data Node to remain air-gapped (where the Data Monitoring Node has no network connections except to the Data Node).

Data Owner-VMs are temporary VMs created for the submission of data (ingress) by a data owner into the infrastructure. It is a minimalistic VM supporting only secure copy (SCP) upload of data. Upon completion of the upload the data is shifted to the Data Node by the system administrator after which this VM can be deleted. It is also created for being used for *safe returns* (in terms of TREs (United Kingdom Health Data Research Alliance, 2020)) and data- or result exports as both processes may require clearing from the data owner.

Analyst-VMs are temporary VMs created individually for each analysis and data processing task. Upon creation with a 0ed storage region (in the reference implementation we format the encrypted virtual partitions with 0s as first step in the set-up when creating the file system) they are equipped with a copy of the data subset and the tools required by the Analyst (c.f. Figure 2). Connections are solely possible from the corresponding Remote Desktop-VM and to verified license servers when required by specific tools, as well as for transferring result data to the associated Data Owner-VM for *safe return* and export. This node is the place where the data visiting takes place.

Remote Desktop-VMs are temporary VMs created for a selected Analyst-VM - these only occur in pairs. We use TigerVNC[15] as software implementation that runs inside the Remote Desktop-VM as a process, providing windowing system capabilities for using graphical tools at the Analyst-VM, configured to provide only video connectivity and without any cut-and-paste capability offered by `Xvnc`. The Remote Desktop-VMs sends all interactions and the video stream to the Monitoring Node.

[11] "Change encryption cipher in Access Server". [Online]. URL: `https://openvpn.net/vpn-server-resources/change-encryption-cipher-in-access-server/`, accessed 2021-06-10

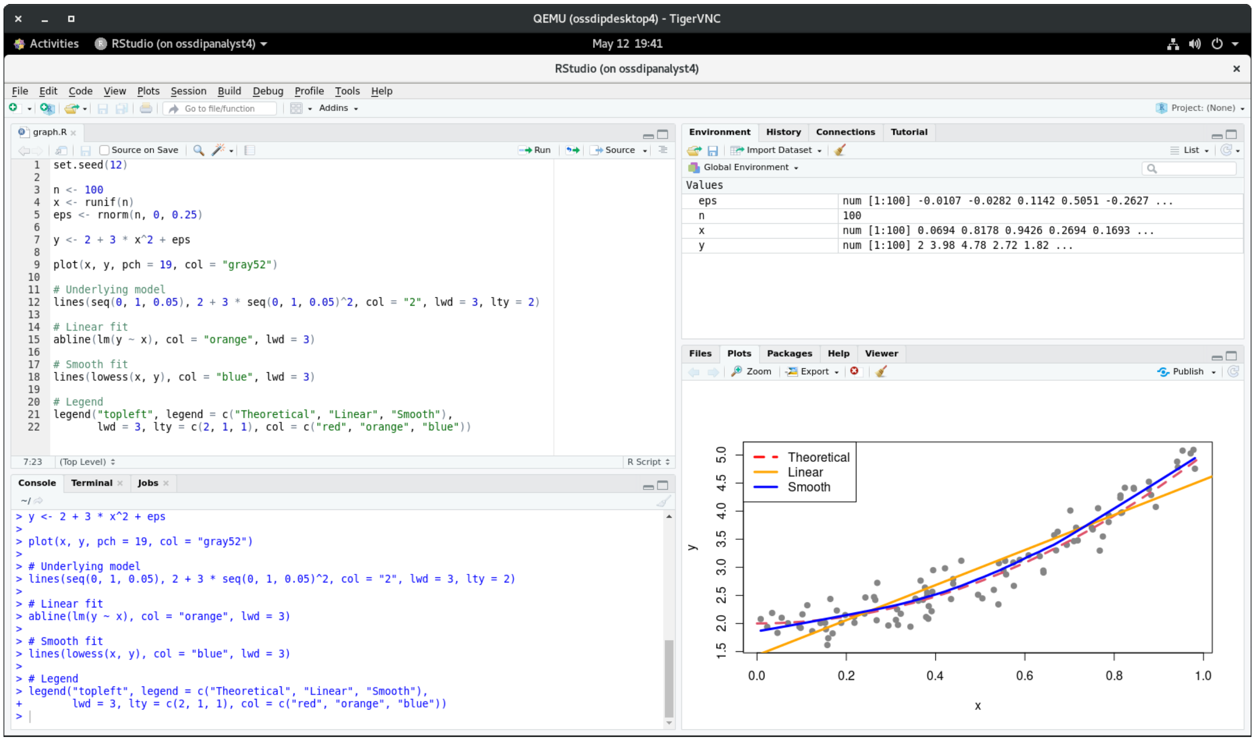
[12] "Rocky Linux". [Online]. URL: `https://rockylinux.org/`, accessed 2021-12-13

[13] "OpenLDAP". [Online]. URL: `https://www.openldap.org/`, accessed 2021-12-13

[14] In the reference implementation we use *rsyslog* that is configured to keep the log append-only, except for log rotation where this restriction is quickly disabled and re-enabled. Since the Data Monitoring Node can only be accessed via SSH in our implementation, we configure the SSH service to set the `CAP_LINUX_IMMUTABLE` flag, so not even root user/process can modify these read-only/append-only attributes

[15] "TigerVNC". [Online]. URL: `https://tigervnc.org/`, accessed 2021-07-27

Key Node is a separated on a dedicated physical server that holds the password to the encryption key of the storage of each node.



```
set.seed(12)

n <- 100
x <- runif(n)
eps <- rnorm(n, 0, 0.25)

y <- 2 + 3 * x^2 + eps

plot(x, y, pch = 19, col = "gray52")

# Underlying model
lines(seq(0, 1, 0.05), 2 + 3 * seq(0, 1, 0.05)^2, col = "2", lwd = 3, lty = 2)

# Linear fit
abline(lm(y ~ x), col = "orange", lwd = 3)

# Smooth fit
lines(lowess(x, y), col = "blue", lwd = 3)

# Legend
legend("topleft", legend = c("Theoretical", "Linear", "Smooth"),
       lwd = 3, lty = c(2, 1, 1), col = c("red", "orange", "blue"))
```

```
> y <- 2 + 3 * x^2 + eps
>
> plot(x, y, pch = 19, col = "gray52")
>
> # Underlying model
> lines(seq(0, 1, 0.05), 2 + 3 * seq(0, 1, 0.05)^2, col = "2", lwd = 3, lty = 2)
>
> # Linear fit
> abline(lm(y ~ x), col = "orange", lwd = 3)
>
> # Smooth fit
> lines(lowess(x, y), col = "blue", lwd = 3)
>
> # Legend
> legend("topleft", legend = c("Theoretical", "Linear", "Smooth"),
+        lwd = 3, lty = c(2, 1, 1), col = c("red", "orange", "blue"))
> |
```

```
Values
  eps    num [1:100] -0.0107 -0.0282 0.1142 0.5051 -0.2627 ...
  n      100
  x      num [1:100] 0.0694 0.8178 0.9426 0.2694 0.1693 ...
  y      num [1:100] 2 3.98 4.78 2.72 1.82 ...
```

Figure 2: The Analyst can visit sensitive data using e.g. RStudio through the windowing system from the Remote Desktop-VM. The screenshot contains sample data for visualization purposes.

4 Secure Data Infrastructure Controls

This section describes the high-level configuration of the technical implementation of our secure data infrastructure, the processes required to operate on the secure data infrastructure are explained in Sec. 5. To prevent data leakage, our approach implements multiple layers that secure the control of the data from the physical location of the server to the pixel displayed on the screen in five controls. Since analysts need to disclose how they are going to use the data through an approval process, the data owner may have interest in checking for the truthfulness of their statement. In the following we explain the five controls (Roles and Controlled Access, Data Segmentation, Network Segmentation, Automation, Monitoring) of our approach that protect the data.

4.1 Roles and Controlled Access

Physical security that restricts access to the server hardware is the first control to protect the sensitive data in the infrastructure, c.f. (Knapp, Denney, and Barner, 2011). We recommend to place the hardware needed into a dedicated locked server rack where only a designated and certified operator can open the lock. Following the four eye principle, a key card for the server room is needed that is held solely by another operator.

Our method uses the role-based access control concept (Ferraiolo and Kuhn, 1992) where individuals are assigned a set of roles that allows them to access previously defined components in the secure data infrastructure. To prevent privilege escalation attacks (Provos, Friedl, and Honeyman, 2003), we equip each role only with a bare minimum of privileges on a need-to-know principle for data access. We argue to limit the number of roles to only five in order to have a clear distinction of requirements and interests to interact with the system.

Data Owner has a strong interest in providing an identified expert access to the data but wants to retain control of the data and specifically reduce the risk of data leakage to an acceptable level. Our approach provides temporary and isolated Data Owner-VMs that allow import of structured data to the database of the Data

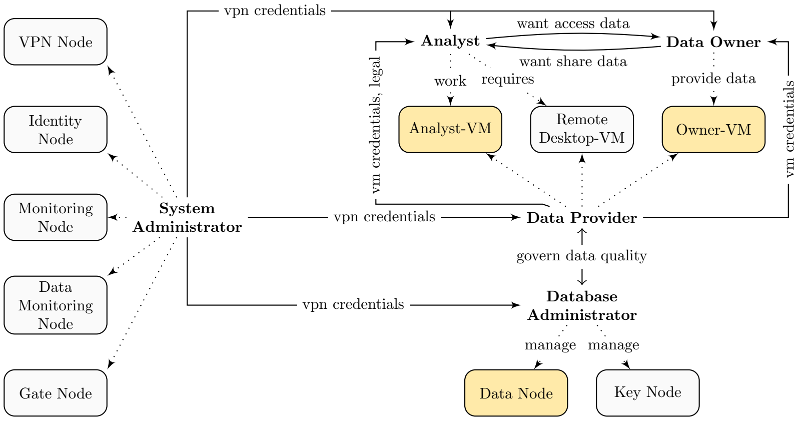


Figure 3: Social architecture of OSSDIP, dotted arrows are tasks that the respective role performs on infrastructure components.

Node after which they are destroyed and consigning the Data Owner full control over who the data is provided to. It furthermore has access to comprehensive logging information providing a full audit trail (all interactions with their data within the infrastructure from data import, via any provisioning steps to data deletion).

Analyst has a clear understanding on what research questions should be answered with it and what data is required. This role can be assigned to experts that need to analyze or process the data, but where sharing the data is not feasible. With the permission of the Data Owner, the Analyst is able to use an according subset of the data in isolated VM. Access is only granted for a limited time period with the possibility to extend it following request and approval processes via the Data Owner. The Analyst is granted access only to own corresponding Analyst-VMs created explicitly for the approved set of analysis or processing goals.

Data Provider is processing the data on behalf of the Data Owner. Entities equipped with this role manage the services and takes care of all data operations (e.g. creating the respective isolated VMs for Data Owner- and Analyst, monitoring user interactions, handle the legal contracts required). In most of the cases, the Data Owner itself decides to provide the services, but can outsource the operation of the infrastructure to a different Carrier role not covered in this paper.

Database Administrator is responsible for maintaining the Data Node. The Database Administrator is nominated by the provider organization and is mutually exclusive with the System Administrator. This exclusivity is required to not enable a change in the central Data Node and subsequently manipulate the logs in the Monitoring Node. This role also has to maintain the Key Node that stores the passphrase for decrypting the encryption key for the node disks for similar reasons.

System Administrator maintains the secure data infrastructure environment, except for the central Data Node. This role also manages the platform where the infrastructure is hosted i.e. OpenStack[16] for our reference implementation. This role may be further sub-divided to assign, e.g. two different individuals, the respective role for the Data Node and the Data Monitoring Node.

Data Owner- and Analyst-VMs, as well as the Data Node, require a two-factor authentication from the role that is using them. Our reference implementation automatically configures them to require a time-based one-time password provided by Google Authenticator[17] for each login. To better understand the roles defined, we visualized their interaction with each other in Figure 3. An important interaction between roles is e.g. the Data Provider equipping the Analyst with VM credentials and the tools needed to analyze the data subset on the Analyst-VM through the Remote Desktop-VM.

4.2 Data Segmentation

To achieve a clear segmentation of data streams for different roles, we implement a control that provides strongly restricted, isolated (virtual) machines for any entity external to the organization. The control provides a query store documenting all queries issued against the database, including execution timestamps, result set hashes and other

[16] "Open Source Cloud Computing Infrastructure". [Online]. URL: `https://www.openstack.org/`, accessed 2021-07-27
[17] "Google Authenticator". [Online]. URL: `https://github.com/google/google-authenticator`, accessed 2021-07-26

metadata to ensure the query results can be reproduced at any time in the future even if the data in the respective tables should change. This will allow the data used to be cited externally using a persistent identifier e.g. DOI (currently relying on an internal mechanism) that, depending on settings, can be set to re-direct to a respective landing page providing externally visible metadata information on the data subset and e.g. contact information for data access requests, supporting FAIRness for sensitive data. Our reference implementation uses a *PostgreSQL* database, implementing time-stamping and versioning via temporal tables (Kulkarni and Michels, 2012).

All connections to this central Data Node are one-way oriented following the recommendation on *safe data* and *safe return* for TREs (United Kingdom Health Data Research Alliance, 2020). These one-way oriented connections are implemented through the use of the low-level netfilter kernel module and custom chains. The Analyst can submit a request for data egress out of the infrastructure. Our approach is that the Analyst places the data to be egressed in a clearly labeled export folder and the System Administrator then transfers the data to the Data Owner-VM. In case of approval, the data then can be re-inserted into the central Data Node or exported by the Data Owner who subsequently can make it available to anybody else. This constitutes an additional control of protecting the sensitive data and results.

For Analyst-VMs additional policies such as prohibition to install barcode-generating packages are implemented in the secure data infrastructure. This massive overhead of processes running in isolated VMs ultimately only allows for visual access to the data to the best of our knowledge. An adversary still is able to take screenshots of e.g. paginated views of data or using code to display encoded representation of data in barcodes. By having a copy of all code being deployed on VM set-up and recording both the Remote Desktop-VM video stream as well as Analyst-VM activities on the Monitoring Node, can be traced activities to detect unusual behavior in task processing. This is complemented by passive security measures such as data fingerprints being embedded upon VM deployment. For future work, we want to implement real-time screen watermarking (Piec and Rauber, 2014) to additionally be able to follow and trace screenshots to the adversary in case of data leakage.

4.3 Network Segmentation

Complementing the control in Sec. 4.2, we separate the central Data Node from the Data Owner- and Analyst-VMs by placing them into their own network subnets as additional security control. The standard netfilter module of the Linux kernel is sufficient for our reference implementation and is used on the Virtualization Host, the Gate Node and the VPN Node. To make administration straightforward and as failsafe as possible with the tools present, we create a new VPN Node inside the infrastructure instead of offing their service at the Virtualization Host. Note that, contrary to the Data Owner-VM, an Analyst-VM does not allow direct external secure shell interaction. For Analyst-VMs the Analyst first connects to the assigned Remote Desktop-VM which then provides secure shell interaction with the respective Analyst-VM.

4.4 Automation

An automation engine like Ansible allows infrastructure operators to control how repetitive processes are executed on the system by using configuration files and scripts. Capabilities to install, configure, update and uninstall parts of the system are supported by the engine and provide a valuable tool to administrators that can customize the system by changing e.g. environment variables only.

The reference implementation of our secure data infrastructure comes pre-configured, with a step-by-step guide to make the initial deployment on an empty host as straightforward as possible for provider organizations. After installation of the operating system with virtualization technology, only the version control system Git[18] and Ansible need to be present as necessary automation dependencies. The set-up starts with executing the playbook (text files with sets of instructions that provide transparent user management, networking configuration). We also use it for safe creation of encryption keys with Linux unified key setup and starting the core infrastructure components (cf. Sec. 3.4). at the Virtualization Host. This allows the automated setup through Ansible within 50-60 minutes using our step-by-step guide.

The regular deployment and destruction of Data Owner- or Analyst-VMs could be vulnerable to human error. That is why we automated these tasks to ensure that managing virtual machines follows a tested standard process. Processes without much human intervention enable a transparent operation of the secure data infrastructure for the Data Provider that especially enables a safe operation from the set-up on-wards.

4.5 Monitoring

Every operation performed on the isolated Data Owner- and Analyst-VMs is monitored closely using explicit contractual agreements between the respective role and the Data Owner. The Monitoring Node is configured to only

[18] "Git". [Online]. URL: `https://git-scm.com/`, accessed 2021-07-31

append to log files and does not allow modifications by the System Administrator. We noticed early on, that only system-level logging alone is not sufficient for later investigation. Although VMs are able to send the events to the central Monitoring Node within the secure data infrastructure, it does not provide enough information to comprehend the performed operations since it only captures the infrastructure's actions.

Therefore, in order to capture the human interaction with the infrastructure, the Analyst must agree to monitoring the shell and remote desktop interaction. This approach allows Data Owners to automatically analyze certain interaction patterns. If an Analyst, for example, tries to extract sensitive data using self-written scripts by displaying scan-able barcodes, this can be detected by scanning the input from the keyboard stream for known barcode names via the logged shell history or checking the recorded video stream of the Remote Desktop-VM.

Since both roles, the Data Owner and the Analyst need to trust the infrastructure as a meeting point for accessing sensitive data, we monitor the Data Provider, Database- and System Administrator roles too. All interactions with the Data Node are logged to the Data Monitoring Node[19], ideally via a dedicated (non-virtualized) server equipped with write-once storage technology.

5 Secure Data Processes

In this section we present the processes that should be transparently communicated with all involved roles to raise awareness of the interactions needed to execute sensitive operations. It covers the basic interactions and omits well-studied standard processes like user identification.

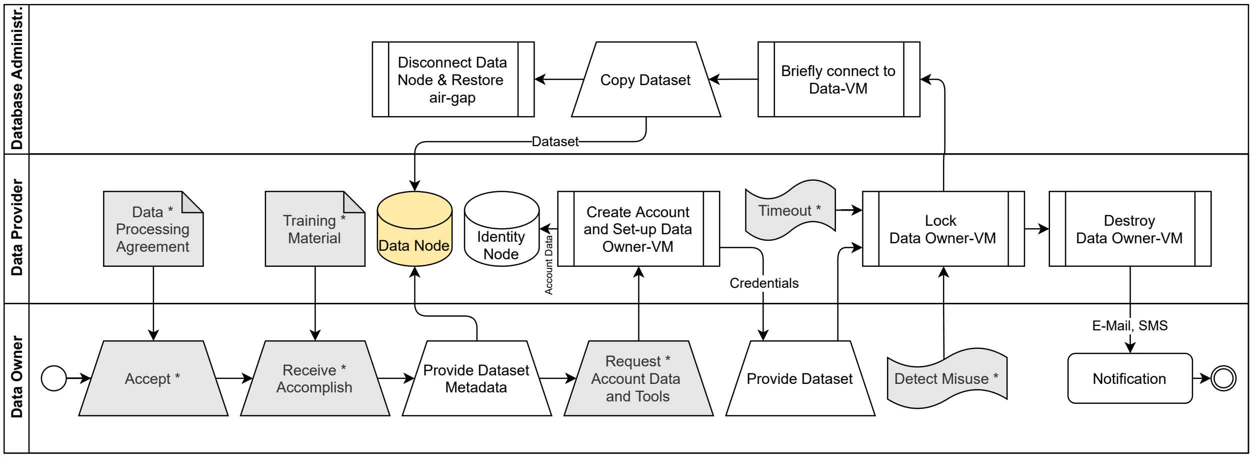


Figure 4: To import data into the infrastructure, the Data Owner must follow the Data Ingress process (steps that are relevant only when the Data Provider is different from the Data Owner are colored gray and marked with an asterisk *). We color the Data Node golden, since it contains sensitive data.

5.1 Data Ingress

Whenever a Data Owner wants to import data into the secure data infrastructure, the data ingress process is started. Initially, the Data Owner must sign an agreement on data delivery ("data processing agreement") and provide meta data of the data set, specifically: (i) list of attributes with respective description and primary key; (ii) number of records e.g. rows; (iii) disclosure of format. Currently only comma-separated values are supported in the reference implementation, providing the separator qualifier, null value encoding, boolean value encoding, date encoding and; (iv) short description of the data. The Data Owner needs to disclose personal information like: (i) first- and lastname (ii) organization (iii) e-mail address and (iv) mobile phone number to send messages at the end of the process to (or receive a call). The Data Owner subsequently receives an account (or re-uses an existing). This step requires manual interaction (will usually be automated using trusted authentication services) and takes a few minutes upon approval before the Data Owner can access the Data Owner-VM. As seen in Figure 4, the secure data infrastructure then automatically creates a new isolated Data Owner-VM, updates the firewall rules to grant access to this machine

[19] This monitoring has to be performed by a dedicated monitoring node to allow the Database Node to remain air-gapped (disconnected) from the rest of the infrastructure most of the time, except for short time windows when data is transferred to a new Analyst-VM from a Data Owner-VM.

for the Data Owner and sends the credentials to access it. The Data Owner can transfer the sensitive data via a double-encrypted channel (VPN and SSH). After confirmation that the data is completely sent (or after a pre-defined timeout) the infrastructure locks the virtual machine, transfers the data from it and securely destroys it by overwriting the strorage with zeroes. The Data Owner then is notified using two different channels that the transfer was successful.

5.2 Data Access

To access the data, an Analyst needs to send a request to the Data Owner containing: (i) personal data that allows identification of the Analyst (e.g. first and last name), (ii) required data (usually a sub set of the available data), and (iii) required tools to analyze the data and optionally prepare the data in a way that it can be re-imported into the Data Node (*safe return* in terms of TREs (United Kingdom Health Data Research Alliance, 2020)) (iv) task and research questions that should be answered with the required data.

After (manual) check of the identity, the Data Owner and additional committees such as boards, review the research questions and grant or reject permission to use the data. When granted, the "data access agreement" between Data Owner and Analyst must be accepted including the conditions of use: (i) prohibition of data download (ii) prohibition of de-anonymization (iii) non-disclosure agreement and (iv) agreement to extensive monitoring. Upon acceptance, a request for an account is issued providing the required information (e-mail address and mobile phone number to send messages).

Subsequently, an Analyst-VM is automatically created with a pre-specified life-time (expiry date) according to the envisaged project duration (with explicit prolongation confirmation required). The requested subset of data at specific aggregation level (potentially adding a fingerprint and/or applying anonymization at specified levels as requested by the Data Owner) is extracted from the Data Node and pushed onto the Analyst-VM together with the requested (and cleared) set of tools for analysis. As the Data Node implements the RDA recommendations for dynamic data citation (Rauber et al., 2015; 2016) recommendations on dynamic data citation, the respective query used to select the subset of data is stored in a query-store with the query execution timestamp and some associated metadata described in (Weise et al., 2021). If the respective subset has to be re-created at a later point in time, the time-stamped query can be re-executed against the time-stamped and versioned database to extract the exact same subset of data, ensuring full reproducibility.

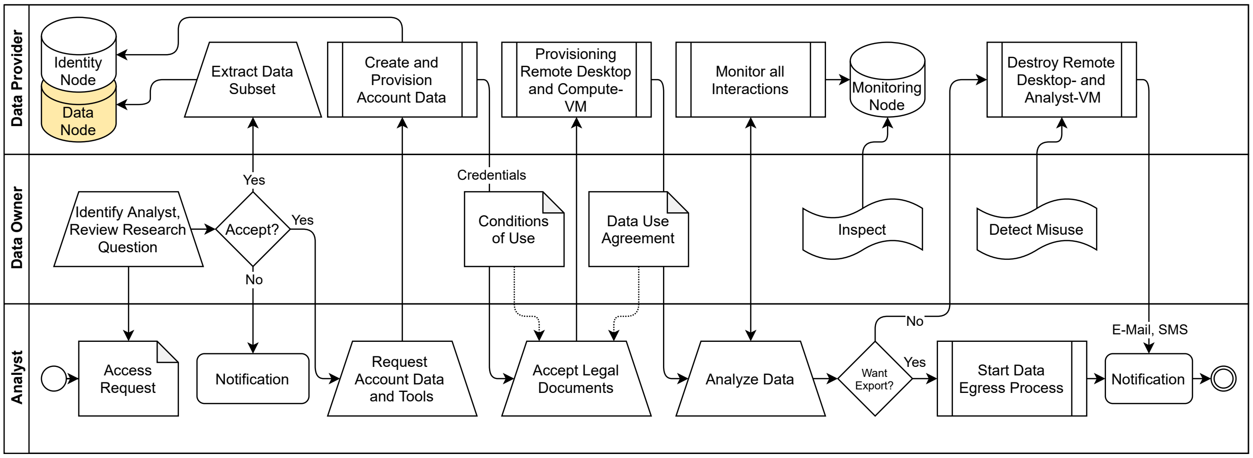


Figure 5: To visit data in the infrastructure, the Analyst must follow the Data Access process. Since the Data Node contains sensitive data, we color it golden.

Furthermore, a dedicated Remote Desktop-VM is created to provide the sole access to the Analyst-VM. Also, the firewall configurations are adapted to allow access by the specified user (Analyst) to the respective Remote Desktop-VM. The Analyst then can analyze the data as long as the time-out is not reached. Additionally, the Data Owner has the right to lock access (via locking the user account of the Analyst to the Analyst-VM) any time to protect the data, and to inspect all activity happening on the Analyst-VM by inspecting its logs.

At the end of the analysis, the Analyst may wish to re-import some data back to the Data Node or to export certain result files to an external machine. This is possible after approval by the Data Owner. Afterwards, Analyst-VM and Remote Desktop-VM are destroyed, the respective firewall entries are removed, and the account credentials disabled.

6 Conclusions and Future Work

The current state of the secure data infrastructure allows a Data Owner to invite experts (e.g. Analyst) to visit sensitive data on a trusted meeting point (Analyst-VM). This ensures that not all layers are able be compromised at the same time. Currently, the set-up of `OSSDIP` requires at least three (optimally nine) physical machines on trusted hardware. We do not standardize the provisioning of required tools for data ingress in the reference implementation. Whenever the Data Owner-VM requires non-standard data science packages, the System Administrator needs to manually configure the VM. This could be resolved through developing a software catalog of audited, trusted data science software along with hardening guidelines to remove communication channels. Our proposed reference architecture is not limited to private clouds but can also be offered as Cloud-as-a-Service by commercial cloud providers. However, additional security measures would need to be implemented since commercial cloud providers follow different trust models than research institutions, the scenario of a compromised hypervisor should also be considered in this case. As a first consequence, the attestation of the Data Node becomes a problem. A commercial provider would need to prove to every user that the Data Node is truly disconnected from the Internet and only accessible by the database administrator for approved processes. While the Data Monitoring Node does log connection establishments and actions that are performed on the Data Node, commercial cloud providers would not publish their log files as they would expose themselves to e.g. man-in-the-middle attacks by giving away information about session occurrences for e.g. data ingestion. Such a problem could be met with a collective attestation scheme (Song et al., 2020) that requires an additional attestation node to be present in `OSSDIP`.

As a second consequence of a using commercial CaaS providers to operate `OSSDIP`, a single storage implementation most likely would exceed the initial budget as the required (highly available) storage grows. A likely setup would include multiple physical nodes and multiple partitions so that one node can also hold data of several users. This setup requires additional protection, roles, privileges to access the data, also because data grid concepts to manage distributed and partitioned data would most probably be employed (Navaz, Prabhadevi, and Sangeetha, 2013). Not only physical but also virtual nodes would need to be multiplied. On the physical node that virtualizes the Analyst-VMs, one VM for each analyst would be created which exposes each VM to a potential "bad neighbour VM" problem of stealing critical information through side-channel attacks (Ristenpart et al., 2009; Zhang et al., 2012). Such problems are considered in our system and processes design through dedicated monitoring nodes which aim at capturing such attack attempts by logging every user operation and video-recording the session to the Analyst-VM.

We have presented an architecture and processes for operating a secure data infrastructure that supports secure data visiting. Data Owners can make their data accessible to experts for specific tasks while maintaining full control of how their data is being used and preventing unintentional data leakage. The concept is based on providing access to the subsets of (potentially fingerprinted and/or anonymized) data required for a specific activity via isolated virtual machines that are monitored and accessible only via remote desktop access along with secure processes that ensure no accidental leak of sensitive data.

Furthermore we presented a reference implementation of this infrastructure that is based entirely on well-studied open-source components. The set-up, configuration and core operational processes have been automated to a degree such that it can be easily deployed within a short period of time. Thus our work significantly improves existing approaches in both technical and organizational (through secure data processes) measures. Future versions will include additional modules for applying fingerprints to data following an approach by (Li, Swarup, and Jajodia, 2005; Lafaye et al., 2008) to be able to track the origin of a possible data leak. Similarly, modules that implement privacy-preserving techniques like k-anonymity, l-diversity and t-closeness are currently being integrated into the system. Timestamping and versioning for the central Data Node using temporal tables allow, in combination with the time-stamped query store (inside the Data Node), to re-produce any subset of data. Support for external citation through e.g. DOIs as well as the creation of the according landing pages, is missing in the current version and is a candidate for future implementation (to make data FAIR).

As part of the EOSC-Life project, we aim to implement more tools that that are used within the biomedical domain (e.g. KNIME[20], Jupyter Notebooks[21]). Since sensitive data leaking out of the infrastructure constitutes a significant threat for any secure data infrastructure, there is a strong need for solid, continuous analysis of the monitoring information, both on the visual video stream as well as the structured log files to raise according alarms and automatically trigger suitable actions to reduce the risk of data leakage.

[20] "KNIME". [Online]. URL: `https://www.knime.com/`, accessed 2021-07-30

[21] "Project Jupyter". [Online]. URL: `https://jupyter.org/`, accessed 2021-07-30

Supplemental Material

The entire source code of the system configuration, set-up scripts and the comprehensive documentation are available in the public GitLab repository [22] under Apache 2 license.

Funding Information

This project is supported by the european Union's H2020 programme by the EOSC-Secretariat project under the Co-Creation programme (grant agreement number 831644) and by EOSC-Life (grant agreement number 824087) in cooperation with the Austrian COVID-19 Future Operations Platform.

[22] OSSDIP. [Online]. URL: `https://gitlab.tuwien.ac.at/martin.weise/ossdip`, accessed 2021-07-30

References

Ferraiolo, D. and Kuhn, R., 1992. Role-Based Access Controls. *Proceedings of the 15th National Computer Security Conference*. Baltimore, Maryland, USA, pp.554–563.

Sweeney, L., 2002. K-anonymity: a model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05), pp.557–570.

Provos, N., Friedl, M., and Honeyman, P., 2003. Preventing Privilege Escalation. *Proceedings of the 12th USENIX Security Symposium*, pp.231–241.

Li, Y., Swarup, V., and Jajodia, S., 2005. Fingerprinting Relational Databases: Schemes and Specialties. *Transactions on Dependable and Secure Computing* [Online], 2(1). Available from: `https://doi.org/10.1109/TDSC.2005.12`.

Li, N., Li, T., and Venkatasubramanian, S., 2007. T-closeness: privacy beyond k-anonymity and l-diversity. *2007 ieee 23rd international conference on data engineering*. IEEE, pp.106–115.

Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M., 2007. L-diversity: privacy beyond k-anonymity. *Acm transactions on knowledge discovery from data (tkdd)*, 1(1), 3–es.

Lafaye, J., Gross-Amblard, D., Constantin, C., and Guerrouani, M., 2008. Watermill: An Optimized Fingerprinting System for Databases under Constraints. *IEEE Transactions on Knowledge and Data Engineering* [Online], 20(4). Available from: `https://doi.org/10.1109/TKDE.2007.190713`.

Ristenpart, T., Tromer, E., Shacham, H., and Savage, S., 2009. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. *Proceedings of the 16th acm conference on computer and communications security*, pp.199–212.

Hao, F., Lakshman, T.V., Mukherjee, S., and Song, H., 2010. Secure cloud computing with a virtualized network infrastructure. *Proceedings of the 2nd usenix conference on hot topics in cloud computing*, HotCloud'10. Boston, MA: USENIX Association, p.16.

Barkow, I., Leopold, T., Raab, M., Schiller, D., Wenzig, K., Blossfeld, H.-P., and Rittberger, M., 2011. 20 remoteneps: Data Dissemination in a Collaborative Workspace. *Zeitschrift für Erziehungswissenschaft* [Online], 14(2), pp.315–325. Available from: `https://doi.org/10.1007/s11618-011-0192-5`.

Knapp, K.J., Denney, G.D., and Barner, M.E., 2011. Key Issues in Data Center Security: An Investigation of Government Audit Reports. *Government Information Quarterly* [Online], 28(4), pp.533–541. Available from: `https://doi.org/10.1016/j.giq.2010.10.008`.

Kulkarni, K. and Michels, J.-E., 2012. Temporal Features in SQL:2011 [Online]. 41(3) (), 34–43. Available from: `https://doi.org/10.1145/2380776.2380786`.

Zhang, Y., Juels, A., Reiter, M.K., and Ristenpart, T., 2012. Cross-vm side channels and their use to extract private keys. *Proceedings of the 2012 acm conference on computer and communications security*, pp.305–316.

Navaz, A., Prabhadevi, C, and Sangeetha, V, 2013. Data grid concepts for data security in distributed computing. *Arxiv preprint arxiv:1308.6058*.

Jones, K.H., Ford, D.V., Jones, C., Dsilva, R., Thompson, S., Brooks, C.J., Heaven, M.L., Thayer, D.S., McNerney, C.L., and Lyons, R.A., 2014. A case study of the Secure Anonymous Information Linkage (SAIL) Gateway: A privacy-protecting remote access system for health-related research and evaluation. *Journal of Biomedical Informatics* [Online], 50. Special Issue on Informatics Methods in Medical Privacy, pp.196–204. Available from: `https://doi.org/10.1016/j.jbi.2014.01.003`.

Piec, M. and Rauber, A., 2014. Real-Time Screen Watermarking Using Overlaying Layer. *2014 Ninth International Conference on Availability, Reliability and Security* [Online], pp.561–570. Available from: https://doi.org/10.1109/ARES.2014.83.

Garfinkel, S.L. et al., 2015. De-identification of personal information. *National institute of standards and technology*.

Rauber, A., Asmi, A., Uytvanck, D. van, and Proell, S., 2015. *Data Citation of Evolving Data: Recommendations of the Working Group on Data Citation (WGDC)* [Online]. http://doi.org/10.15497/RDA00016. Available from: https://doi.org/10.15497/RDA00016.

Sidorov, V. and Ng, W.K., 2015. Transparent data encryption for data-in-use and data-at-rest in a cloud-based database-as-a-service solution. *2015 ieee world congress on services*. IEEE, pp.221–228.

Desai, T., Ritchie, F., and Welpton, R., 2016. Five Safes: Designing data access for research [Online]. Economics Working Paper Series 1601. Available from: https://doi.org/10.13140/RG.2.1.3661.1604.

Rauber, A., Asmi, A., Uytvanck, D. van, and Proell, S., 2016. Identification of Reproducible Subsets for Data Citation, Sharing and Re-Use. *Bulletin of the IEEE Technical Committe on Digital Libraries (TCDL)* [Online], 12(1) (1). Available from: https://doi.org/10.5281/zenodo.4048304.

Skopek, J., Koberg, T., and Blossfeld, H.-P., 2016. RemoteNEPS – An Innovative Research Environment. In: H.-P. Blossfeld, J. von Maurice, M. Bayer, and J. Skopek, eds. *Methodological Issues of Longitudinal Surveys: The Example of the National Educational Panel Study* [Online]. Springer, pp.611–626. Available from: https://doi.org/10.1007/978-3-658-11994-2_34.

Popper, N., Endel, F., Mayer, R., Bicher, M., and Glock, B., 2017. Planning Future Health: Developing Big Data and System Modelling Pipelines for Health System Research. *Simulation Notes Europe* [Online], 27 (4), pp.203–208. Available from: https://doi.org/10.11128/sne.27.tn.10396.

Akula, M., 2019. *Defenders' Guide to Container Infrastructure Security*. LISA19. Portland, Oregon, USA: USENIX Association.

Cuggia, M. and Combes, S., 2019. The French Health Data Hub and the German Medical Informatics Initiatives: Two National Projects to Promote Data Sharing in Healthcare. EN. *Yearbook of Medical Informatics* [Online], 28(01). 195, pp.195–202. Available from: https://doi.org/10.1055/s-0039-1677917.

Palmgren, J., Rasmussen, T., Bengtström, M., Kahri, P., Ebbing, M., Henrichsen, B., Nilsson, M., and Høst, G., 2019. *A vision of a Nordic secure digital infrastructure for health data: The Nordic Commons*. (Technical report). Oslo: Nordic Council of Ministers, NordForsk.

Dood, R., 2020. *Remote Access to Official Microdata*. 8th Conference for Social and Economic Data (KSWD). Berlin, Germany.

Kostiainen, K., Dhar, A., and Capkun, S., 2020. Dedicated Security Chips in the Age of Secure Enclaves. *IEEE Security & Privacy* [Online], 18(5), pp.38–46. Available from: https://doi.org/10.1109/MSEC.2020.2990230.

Lee, D., Kohlbrenner, D., Shinde, S., Asanović, K., and Song, D., 2020. Keystone: An Open Framework for Architecting Trusted Execution Environments. *Proceedings of the Fifteenth European Conference on Computer Systems* [Online], EuroSys '20. Heraklion, Greece: Association for Computing Machinery. Available from: https://doi.org/10.1145/3342195.3387532.

Mousa, A., Karabatak, M., and Mustafa, T., 2020. Database security threats and challenges. *8th International Symposium on Digital Forensics and Security (ISDFS)*, pp.1–5.

Song, Y., Shi, W., Qin, B., and Liang, B., 2020. A collective attestation scheme towards cloud system. *Cluster computing*, pp.1–12.

United Kingdom Health Data Research Alliance, 2020. *Trusted Research Environments*. [Online]. URL: https://ukhealthdata.org/projects/aligning-approach-to-trusted-research-environments/. Accessed September 2020. Version 2.0.

Williamson, E., Walker, A.J., Bhaskaran, K., Bacon, S., Bates, C., Morton, C.E., Curtis, H.J., Mehrkar, A., Evans, D., Inglesby, P., Cockburn, J., McDonald, H.I., MacKenna, B., Tomlinson, L., Douglas, I.J., Rentsch, C.T., Mathur, R., Wong, A., Grieve, R., Harrison, D., Forbes, H., Schultze, A., Croker, R., Parry, J., Hester, F., Harper, S., Perera, R., Evans, S., Smeeth, L., Goldacre, B., and Collaborative, T.O., 2020. OpenSAFELY: factors associated with COVID-19-related hospital death in the linked electronic health records of 17 million adult NHS patients. *Medrxiv* [Online]. Available from: https://doi.org/10.1101/2020.05.06.20092999.

Brandt, N., Griem, L., Herrmann, C., Schoof, E., Giovanna, T., Zhao, Y., Zschumme, P., and Selzer, M., 2021. Kadi4Mat: A Research Data Infrastructure for Materials Science. *Data Science Journal* [Online], 20 (1). Available from: https://doi.org/10.5334/dsj-2021-008.

Peisert, S., 2021. *An Examination and Survey of Data Confidentiality Issues and Solutions in Academic Research Computing*. https://escholarship.org/uc/item/7cz7m1ws. Online; accessed 22 December 2021.

Weise, M., Michlits, C., Staudinger, M., Gergely, E., Stytsenko, K., Ganguly, R., and Rauber, A., 2021. FDA-DBRepo: A Data Preservation Repository Supporting FAIR Principles, Data Versioning and Reproducible Queries. *Proceedings of the 17th International Conference on Digital Preservation*. Beijing, China: iPRES, p.34.

Weise, M. and Rauber, A., 2021. A Data-Visiting Infrastructure for Providing Access to Preserved Databases that Cannot be Shared or Made Publicly Accessible. *Proceedings of the 17th International Conference on Digital Preservation*. Beijing, China: iPRES, p.51.