Future Generation Computer Systems 129 (2022) 269–285

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

ImPACT: A networked service architecture for safe sharing of restricted data

Ilya Baldin [a,*], Jeff Chase [d], Jonathan Crabtree [b], Thomas Nechyba [c], Laura Christopherson [a], Michael Stealey [a], Charley Kneifel [e], Victor Orlikowski [e], Rob Carter [e], Erik Scott [a], Akio Sone [b], Don Sizemore [b]

[a] RENCI/UNC Chapel Hill, 100 Europa Dr., Chapel Hill, NC, 27517, USA
[b] Odum Institute/UNC Chapel Hill Davis Library, 2nd Floor, Campus Box 3355, Chapel Hill, NC 27599-3355, USA
[c] Duke Social Science Research Institute, 140 Science Drive, Gross Hall, 2nd Floor, Durham, NC 27708, USA
[d] Duke University Computer Science Department, 308 Research Dr, Durham, NC 27705, USA
[e] Duke University OIT, 334 Blackwell Street, Durham, Suite 1100, NC 27701, USA

ARTICLE INFO

Article history:
Received 18 February 2021
Received in revised form 11 November 2021
Accepted 24 November 2021
Available online 15 December 2021

Keywords:
Privacy-restricted data
Data Use Agreement
Authorization logic

ABSTRACT

In this paper we describe an architecture developed and prototyped in the course of the NSF-funded project called ImPACT—Infrastructure for Privacy-Assured CompuTations. This architecture addresses the common problems that arise from the need to securely store, control access to and process privacy-restricted data in a multi-institutional, multi-stakeholder setting. Specifically the architecture includes several components—a way to publicly advertise a limited set of data attributes without exposing the sensitive data itself; a set of mechanisms for a data owner to specify and automatically enforce complex data-access policies commonly expressed today as Data Use Agreements (DUAs); a way to securely collect digital attestations from multiple stakeholders to satisfy those policies; and a reproducible template to deploy secure processing enclaves in which groups of researchers can analyze the data in a way that complies with data owner policies using the tools of their choice. The paper describes the architecture and its instantiation in a prototype, providing a performance evaluation of several components.

1. Introduction

Scientists working with restricted data face a number of challenges. On the one hand, the number of potential data sources is increasing rapidly. For example, social scientists may obtain data from various levels of government (from federal down to municipal and their different agencies), school districts, law enforcement, private companies, health systems, and many other sources (data owners). These datasets are often sensitive and controlled, e.g., because their exposure could harm individuals represented in the dataset. At the same time, data owners recognize that sharing the data with researchers may yield insights with compelling social benefits. Their desire to obtain these benefits conflicts with their obligation to protect the data.

To resolve this tension, the regulations governing conditions of use for sensitive data are increasingly detailed and sophisticated. In particular, access may require complex compliance procedures encoded in DUAs, whose terms and conditions vary across data owners. Scientists face additional challenges to comply with institutional standards and approvals from Institutional Review Boards (IRBs), as institutions take an increasingly proactive role in protecting sensitive data and ensuring that research activities are safe and ethical. The challenges and frictions are particularly acute for multi-institutional and multi-disciplinary research teams, discouraging collaborative research that could otherwise enhance knowledge and benefit society.

For the past four years, the ImPACT project (Infrastructure for Privacy-Assured CompuTations) has been working to address the challenges around safe sharing of restricted data. The project has developed tools and services that can reduce friction to negotiate sharing of sensitive data and automate compliance. This paper presents the architecture of ImPACT and its prototype implementation, and evaluates key choices in the context of exemplary scenarios. The project is a collaboration of the University of North Carolina at Chapel Hill (UNC) and Duke University, and is funded by the US National Science Foundation.

The ImPACT project seeks to enable sharing where sufficient trust exists under terms set by the data owners in their policies for access and usage. Data owners control which parties and facilities are authorized to participate, and generally are able

* Corresponding author.
E-mail address: ibaldin@renci.org (I. Baldin).

https://doi.org/10.1016/j.future.2021.11.026

to maintain control over their data and restrict its distribution and use as they see fit. While the ImPACT project has multiple elements for privacy assurance, this paper focuses on its architecture for networked data sharing based on rich authorization and owner-specified policy. The approach employs declarative trust metadata describing access policies, approval workflows, usage conditions, user identities, research project affiliations, and/or security properties of the infrastructure used to process the data. In this way, ImPACT creates a foundation to express real-world access policies rigorously, automate compliance checking for conditions of access, and generate a trail of authenticated assertions to support accountability for non-compliant use. By adopting such technologies in practice and policy, institutions can improve efficiency and researcher productivity, protect the data, and enhance their capabilities to manage and oversee research involving sensitive datasets. Section 2 summarizes the motivation, threat model, and trust model, and Section 3 gives an overview of the ImPACT architecture.

Contributions. The main contributions of our work are as follows. We believe that each elements is novel in this context, and we are not aware of any system that combines these elements in a similar way or that addresses the needs that we identify.

First, we develop *two declarative formalisms* (Section 4) that work in concert to express rich data access policies in machine-readable form: graph-based DUA workflows and logical trust. They are applicable for multi-institutional (federated) settings and are based on needs expressed by research partners in the social sciences. We show how they can represent real-world policies in an exemplary data collection: the National Longitudinal Study of Adolescent to Adult Health (Add Health).

Second, we describe *deployable software services* (Section 5) to interpret and apply these declarative data access policies. ImPACT introduces a Notary Service that interprets DUA graphs and traverses them to collect and record authenticated approvals and agreements, and certify completion of each required workflow. The ImPACT prototype incorporates logical trust software we developed in a related project: it enables participants to issue authenticated logical policies, including required DUAs and a range of requirements based on identity, memberships, and attestations or endorsements by authorities trusted for this purpose by the data owner as specified in its policy. We demonstrate a Web file server (Presidio) that checks policy compliance with a logic engine before returning requested data. Experimental results show that the cost of policy traversal and compliance checking are practical for exemplary policies.

Third, we present an *end-to-end architecture for networked data sharing* that integrates these elements with three key groups of services to assist researchers in accessing sensitive data safely in federated settings (per Section 3).

Identity management ImPACT leverages standard services and protocols for federated identity to make it simpler to deploy, use, and manage. We demonstrate the value of the CILogon and COmanage cyberinfrastructure services [1,2] to bridge to established networks for federated identity in a flexible way. (See Section 5.1.)

Data discovery ImPACT integrates with Dataverse [3], a popular federated catalog that assists researchers to discover and access datasets of interest (Section 6). The approach extends Dataverse to allow the data owner to retain control of data access and use.

Cloud infrastructure ImPACT incorporates policy structures governing security properties of the infrastructure that clients use to store and process the data. We anticipate that institutions will increasingly provide secure and compliant hosting for sensitive data. To illustrate, we summarize a representative Data Enclave architecture in use at Duke (Section 7).

The role of ImPACT's trust plane is to defend against unauthorized data access, while enabling expressive policy to govern access and use. The trust plane validates that each restricted dataset traverses only authorized users and components, including compliant processing infrastructure. The policy also qualifies the component instances (e.g., notaries) that are eligible for use with a given dataset. In this way, ImPACT enables flexible deployments with many software instances operated by different principals (e.g., institutions or research consortia), linked to enable access to specific datasets in "trust networks" defined by their policy. ImPACT certifies all policy decisions and establishes accountability for policy violations by formalizing responsibilities and agreements.

2. Motivation and overview

The design of ImPACT is motivated by usage scenarios developed in our discussions with domain scientists and other stakeholders. While the number of interviews did not allow us to extract results of statistical significance, we were able to develop a number of exemplary use cases that illustrate how ImPACT can address commonly identified needs. Most of our use cases are derived from social science, where sensitive data is broadly understood to contain Personally Identifiable Information (PII). ImPACT is also applicable for data that is 'restricted' or 'sensitive' due to other concerns, e.g. proprietary interest.

ImPACT usage scenarios involve the following *principal roles* and their responsibilities among stakeholders:

Data Owners possess one or more restricted datasets. On the one hand, they want the data to be used for research; on the other, they must be sure the data is not misused. They, or their agency or institution, set policies to limit access to data and govern its use to balance privacy requirements of subjects of the data set with societal or commercial needs to take advantage of the information contained in it. These policies may outline what operations may or may not be applied to the data, how the outcomes of research are to be treated and/or security requirements for cyberinfrastructure used to store and analyze the data.

Institutional Governance ensures compliance with ethical use of the data such as protecting the personal identities of subjects. It might be the responsibility of an IRB and/or legal department.

Infrastructure Provider provides compute and storage services to process the restricted data. It could be a campus IT organization, a public cloud provider, or an intermediary organization working with an institution to support research cyberinfrastructure needs [4,5]. The provider is responsible for the security of the environment.

Researcher is affiliated with one or more institutions and research projects. They may be a principal investigator (PI) on a research project, a member of research staff, or a student. They must accept the rules and conditions as set forth by the data owner in DUAs, and obtain any training certifications or IRB approvals for their research as needed. They store and analyze the data using compliant infrastructure hosted by a qualified provider.

We designed the architecture to serve various needs of these stakeholders and to capture and apply compliance policies governing their actions. For example, researchers who generate data in a funded project, the funder may mandate them to make the data discoverable and share it with other researchers under suitable safeguards. Institutions take an increasingly proactive role in data protection to balance potential benefits of access against the potential damage of exposure, including harm to research subjects, reputational risks, and possibly legal liability.

The interviews also revealed that research with sensitive data often involves collaborations not just across schools or departments, but across separate institutions. For instance, *a collaborative project across universities* has gathered longitudinal survey data on student behaviors and attitudes at participating institutions and aggregated these into a single data set to be shared with authorized researchers at those institutions. Such a collaboration involves researchers from multiple institutions with their own IRBs, who use elements of shared secure infrastructure provided by the participating institutions to analyze multiple datasets from different data owners. Additionally, the data owners are often government agencies or enterprises who collaborate with academic researchers to obtain research value and/or useful insights from their data, often by combining it with data available to the researchers from other sources. The data owners are often highly risk-averse and lack mechanisms to manage safe access.

The ImPACT architecture incorporates mechanisms to facilitate data sharing in such scenarios. It provides languages to express data use policies and automated tools to validate policies, collect and share certifications and agreements to negotiate or establish access, check policy compliance, and record trails of accountability. It can serve as a vehicle to balance benefits and risks by providing *fine-grained controls over data access and facilitating complex negotiations* over data use policies.

2.1. Trust model

Here we summarize the trust model for distributed ImPACT deployments as outlined above. The threat model is disclosure of data to an unauthorized party. ImPACT's role is to enable the Data Owner to define the requirements for authorized use, and then to enforce compliance with those requirements before access is granted. The Data Owner may apply policies and conditions to all aspects of accessing restricted data by a user, including processing and sharing of the outputs.

A key premise is that the various participants may be affiliated with different institutions or enterprises: we represent them as authenticated principal identities, each empowered to control their own data, security assertions, and/or policies. Participating services authenticate the user and collect attributes via federated identity management services commonly used in academic settings. Those trust chains are grounded in institutions affiliated with an approved consortium (e.g., InCommon), or other identity providers, and in trust anchors for any cross-institutional Collaborative Organizations (Section 5.1) accepted by the Data Owner.

Deployed instances of the ImPACT components or other participating services authenticate using cryptographic keypairs in the usual fashion for secure Web services, as do Data Owners. The trust logic used in ImPACT enables principals to endorse other principals for specific roles or attributes, signed under their keypairs. Policies include logical rules to specify what endorsements are required and the rooting of trust chains in accepted anchors. In effect, trust logic enables expression of custom PKI trust networks within the declarative policy at the granularity of a dataset. These PKI networks qualify component instances to contribute to authorization and access for the given dataset (e.g., qualifying institutions and IRBs, project consortium roots, cloud/infrastructure deployments, endorsing authorities for accepted notaries). A related paper [6] summarizes the logic and its use to build trust networks in the context of ImPACT.

The Data Owner issues its authorization policy in a declarative form as a set of one or more DUAs and a set of logical rules that constrain the set of allowable participants and require various attestations, certifications, or endorsements from other authorities. The data provider (e.g., a Presidio instance) checks compliance with the policy before releasing the data to any requester. The Data Owner controls its data provider or trusts it to apply the owner's policy correctly. The policy may allow an infrastructure provider to request data on behalf of a user. The data access policy can apply arbitrary checks to these services according to the policy and governance/endorsement structure, and check that they have made all required attestations through a qualifying Notary Service instance.

Note that certain key aspects of trust are outside of the trust model. Once access is authorized, the Data Owner trusts an authorized Researcher to use the data responsibly and to comply with the agreed conditions of its use. The Researcher trusts the Data Owner for the integrity of the data. If a Data Owner's policy approves a component instance (notary, infrastructure provider/enclave, data provider), then the Data Owner trusts the operators of the instance to protect security of its deployment as represented and agreed. ImPACT does not bear on these aspects of trust for the usage scenarios in this paper.

3. Architecture

The ImPACT architecture takes a unified approach to three related problems: data discovery, access negotiations, and secure analysis. These challenges involve trust among multiple parties operating under various institutional agreements. The ImPACT trust architecture follows several principles: decentralized identity and attestation, policy autonomy, and point-of-use enforcement. These principles make it suitable for a wide range of deployment scenarios.

A common approach to these challenges is to centralize security functions in a single system protected by a portal [4,5] that governs all authorization, identity, and account management. Many systems take this approach. Data owners typically surrender control of their data to the portal and outsource access decisions to the staff supporting the portal. Researchers use the portal to locate the data and also to access and analyze it. The portal may include a processing and analysis environment to keep the data within the confines of the portal system.

The key drawback of this approach is that it forces participants to sacrifice autonomy and delegate their data and functions to the single portal. The portal might not provide all of the data or functions of interest to the researcher. It is not practical for a central portal to meet all possible researcher needs.

In contrast, ImPACT takes a *fully distributed* approach to managed sharing: there is no central point of control or failure in the architecture. Instead, ImPACT separates the various functions and concerns into separate component types (software agents), which various parties may deploy locally and link together by mutual consent, as declared in policy. This architecture enables multi-institutional sharing scenarios involving mixed facilities to authorize activity and to store and process the data. For example, researchers might use ImPACT to bring together datasets of different owners and process them on infrastructure operated by another party, such as a secure network enclave operated by an institution or cloud provider. Of course, the identity networks and trust network for any given dataset rely on component instances and trust anchors declared by its policy.

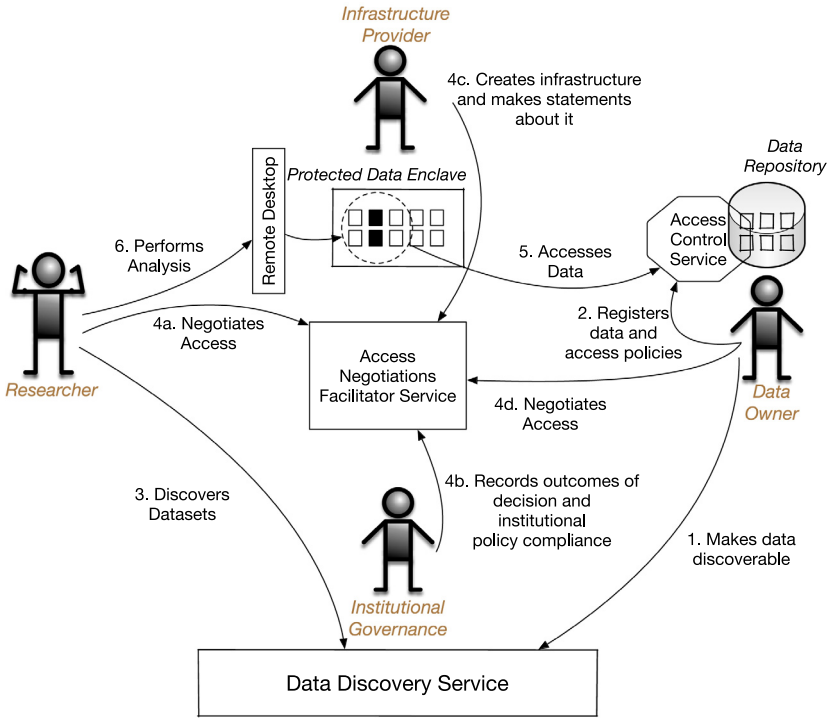


Fig. 1. Participants (principal roles and services) and their interactions in an ImPACT system.

In such distributed scenarios, multiple components or facilities may play a role in enabling safe access to a sensitive dataset. These elements may be controlled by different parties. ImPACT enables sharing when sufficient trust exists under terms set by the data owners in their policies for access and usage. ImPACT enables data owners to control which parties and facilities are authorized to participate, and generally to maintain control over their data and restrict its distribution and use as they see fit.

For example, a data owner might trust a research institution to manage and oversee Data Usage Agreements for its own researchers, a cross-institutional project consortium to qualify researchers for legitimate use, a local storage provider to store the data securely, and a third-party infrastructure provider to maintain a compute enclave meeting specific requirements for safe processing. The ImPACT architecture makes the trust relationships among those component instances explicit and programmable, and validates that all elements comply with trust rules set by the data owners. All parties maintain *autonomy of decisions* governing their own policies and resources, and may delegate specific trust to other parties.

To enable these distributed deployments, ImPACT *factors out trust and identity management* from the components and into a federated trust fabric. Components manage trust by exchanging authenticated statements – assertions and policy rules – and checking for compliance with applicable policy at the point of access or use. The ImPACT prototype built for this architecture combines a logical trust fabric (described below) with standard federated identity management. Researchers use a Web browser to browse datasets and negotiate terms of access authenticated by single sign-on (e.g., InCommon/Shibboleth [1]) for ease of use. The compliance checks combine user identity attributes and certified assertions according to logical policy rules. These choices allow for rich programmable policy and free the ImPACT components from maintaining user accounts or user attributes, leading to more flexible and secure deployments.

3.1. Abstract architecture

Fig. 1 depicts the participants and actions in a typical managed sharing scenario under the ImPACT architecture.

A Data Owner prepares to share a restricted dataset with others under its terms, which allow processing by selected researchers within a protected data enclave on infrastructure that is trusted under the Data Owner's policy. In Step 1, the Data Owner registers the dataset with a *Data Discovery Service* to make it discoverable by other researchers, who can search the service for datasets based on their attributes. Importantly, Data Discovery Service does not provide access to the data, but only to a select set of its attributes and a reference to where the data is stored.

In Step 2, the Data Owner registers its access policies for the dataset with an *Access Control Service*. This federated service collects trust data governing access and applies automated compliance checks at the point of access (e.g., a storage repository). The inputs to those policies are certified statements and attestations from other participants, including the *Access Negotiations Facilitator Service* described below. In the same step the owner informs this service of the details of those policies.

In Step 3, the researcher discovers the data using the Data Discovery Service. Since the data has an access policy associated with it, the Discovery Service redirects the researcher to Access Negotiations Facilitator Service. In Steps 4a, 4b, 4c and 4d the principals involved (Researcher, Institutional Governance, Infrastructure Provider and Data Owner) interact with this service to negotiate access as required by the Data Owner. In particular, a *Data Use Agreement* (DUA) may stipulate that various principals must certify or agree to certain conditions under their authenticated identities. For example, a DUA might require a researcher to accept conditions for allowable use and meet training requirements certified by a project PI, with approval from institutional governance. In ImPACT, access is granted only after all principals have approved their required DUA statements under their authenticated identities. The service records and attests these approvals for later access checks and auditing.

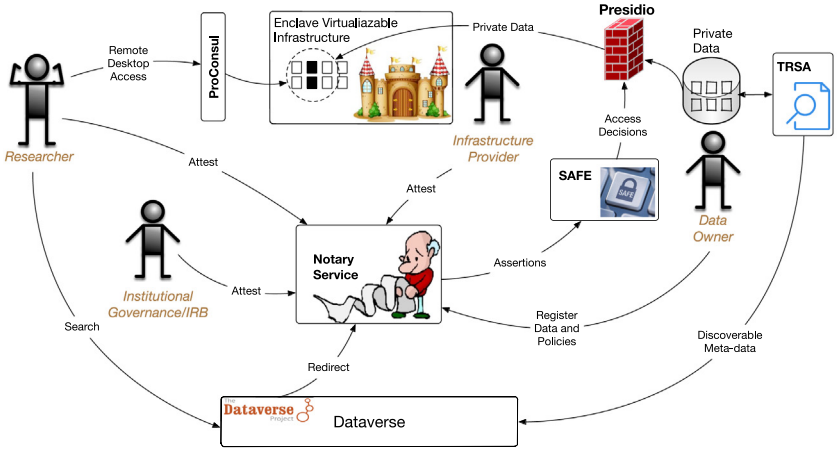


Fig. 2. Core services and interactions in the ImPACT prototype.

In Steps 5 and 6 the researcher obtains access to the data to download it from the Data Owner's approved repository into a Protected Data Enclave for processing. The enclave must comply with security requirements specified by the Data Owner in its access policies. For example, it might limit the tools used for analysis and/or impose specific defenses against data exfiltration, such as remote desktop access with restricted network connectivity.

3.2. Components of the ImPACT prototype

Fig. 2 presents a view of the software components and principal entities in our prototype implementation of the ImPACT architecture. In accordance with the architectural principles, each component may have multiple instances controlled by different entities and supporting their autonomy. We first give an overview of the component functions and their roles in the architecture, then discuss each component in more detail in the following sections.

Dataverse. The role of Data Discovery Service is played by Dataverse [3], a federated data repository system widely used in social sciences and many other domains. Repositories across the globe are leveraging the open source power of Dataverse to publish, share and archive research data. As originally designed, Dataverse ingests each dataset into its repository before making it discoverable. To enable Data Owners to control storage of their data, ImPACT decouples the repository function from the Dataverse discovery service. We extended Dataverse to optionally link datasets by reference (e.g., a URL) instead of storing the data directly. The data remains discoverable via a set of owner-selectable meta-data attributes. The dataset itself resides on storage infrastructure operated by – or otherwise trusted by – the Data Owner.

Trusted remote storage agent. To enable linked datasets in Dataverse, we separated data ingest functions into a library and introduced a new Dataverse component called TRSA or Trusted Remote Storage Agent [7]. The Data Owner runs the TRSA on a machine with direct access to the dataset where it resides, on storage trusted by the Data Owner. The agent examines the dataset *in situ* and harvests metadata attributes for export to a selected Dataverse instance. Researchers can then search metadata catalogs on Dataverse to discover linked datasets of interest. Decoupling ingest functions also supports development of enhanced or customized ingest tooling, and offloading ingest processing from the Dataverse service also improves scalability. Section 6 describes TRSA in more detail.

Notary service. After discovery, the researcher negotiates access to the dataset. A primary goal of ImPACT is to enable rich authorization that automates various elements of policy-based data access including DUAs, which are often managed manually today. To this end, after discovery of an ImPACT-protected dataset, Dataverse directs the researcher to a selected instance of Notary Service [8], a Web-based component implementing the Access Negotiations Facilitator Service from the abstract architecture.

The Notary Service (Section 5.1) manages the approval workflows required by the Data Owner. It relies on the CILogon [2,9] service to allow users to authenticate Web browser sessions with single sign-on. CILogon is a software platform for identity and access management for Web-based application services that support research collaborations. It enables users of these services to authenticate via their institutional identity providers and import their memberships in cross-institutional groups and collaborative organizations certified via COmanage. In this way, the Notary Service obtains (with user approval) a set of certified user identity attributes supplied by the user's home institution and COmanage via CILogon. Notary Service uses these attributes to authorize users and attribute their roles in the approval workflow. Most US research institutions use identity management technologies that are compatible with CILogon, e.g., based on Shibboleth/SAML/InCommon.

Data owners design approval workflows and export them to Notary Service. These workflows, expressed as property graphs, encode the logic of paper DUAs for a given dataset. The Notary Service presents different interfaces to different principal roles—project PIs, staff researchers, data owners, institutional governance or infrastructure providers. Infrastructure Providers register relevant infrastructure elements, researchers create and manage research projects. All principals are presented with opportunities to fill in their portions of a workflow until it is completed. Once the workflow is completed, Notary Service issues a Web token Section 5.1 to the user, which can be used to request access to data from storage.

Presidio access service. Presidio [10] is a simple Web service that exposes a Web API to list and download datasets by URL (Section 5.2). It is configured and managed by the Data Owner, or on their behalf. It runs with direct access to the dataset where it resides on storage trusted by the Data Owner. Before allowing access, Presidio checks for authorization according to the Data Owner's policy, including completion of all required approval workflows in the Notary Service. It uses the Notary Service token supplied in the request and other parameters to fetch relevant trust data and apply a compliance check. Thus Presidio acts as the policy enforcement point for the Access Control Service.

SAFE logical trust. Access control policies are expressed declaratively in machine-readable form using a combination of DUA approval workflows (Section 4.1) and Datalog trust logic (Section 5.3). ImPACT uses the SAFE logical trust platform [11] to represent authorization rules and security assertions issued by various principals. In particular, the Data Owner publishes templated Datalog logic rules for access, and a Notary Service issues assertions to certify completion of approval workflow elements matching those rules by authenticated users. Presidio invokes a SAFE *guard* to check these certifications for compliance with the access policy rules. The access rules may validate other aspects of the policy, including project memberships, trust delegations to the certifying Notary Service, and security attributes of the infrastructure from which a data request originates.

Proconsul data enclave. Finally, the Protected Data Enclave (Section 7) is implemented using a combination of the ProConsul [12] secure remote desktop and on-demand virtual enclave infrastructure for data analysis. Proconsul implements a Shibboleth-authenticated remote desktop providing a 'pane of glass' view of a server enclave created for a given project. As deployed at Duke University, a protected data enclave is a VMware virtual machine cluster on an isolated network, running an approved software stack and approved application packages. Instantiation and tool installation relies on a flexible automated workflow system based on Jenkins [13] or GitLab [14] to build, validate and check policy compliance of application packages automatically. Once logged in to the enclave, the researcher can select from a menu of validated application software to install on demand as individual Singularity [15] packages, all without compromising the security of the enclave.

4. Automating access decisions for restricted data

As discussed in Section 3.2, key steps of Fig. 1 are implemented in our prototype using a set of related components: *Notary Service* which supports interactive DUA negotiations among principals; *Presidio* which allows a data owner to export data for download with rich authorization checks; and *SAFE*, a logical trust platform linking Presidio to other components that have a role in authorization. SAFE includes a logical inference engine which allows us to specify and validate a wide range of data access policies. This section discusses these components in more detail.

4.1. Translating DUAs into machine-readable form

A key goal of ImPACT is to automate the handling of DUA requirements and incorporate them into the access control system. It is common practice today to write DUAs in human-readable forms. We present an approach to encode them in a machine-readable form that exposes individual elements and dependencies as an executable approval workflow to be orchestrated by a Notary Service instance (NS). The NS ingests these machine-readable DUA representations, presents these elements in an easy-to-understand form through its Web interface to users who are authenticated for matching roles, triggers user task prompts when required dependencies are met, and serves as a digital witness attesting that the workflow elements are completed.

Studies conducted on DUAs reveal that they exhibit common structures with a set of clauses specifying different requirements [16]. Perhaps the most visually telling is the presentation of DUA requirements developed by researchers from University of Michigan Institute for Social Research (ISR) who normalized these requirements across over 60 different datasets and presented them in matrix form [17]. Based on this normalization, it is easy to see that requirements fall into several broad categories ('data use', 'publication', 'security' etc.) and while their wording may vary, they suggest similar conditions. Each such category needs some number of assertions from one or more principals, following a dependency structure in which some assertions may 'gate' other assertions. For example: a data owner asserts they have reviewed a proposal, which then opens up a number of other assertions to be filled in by other principals, but not before that.

Note that the assertion categories are targeted to different principal roles. For example, conditions in the 'publication' category must be accepted and followed by the researchers, while those in the 'security' category are targeted primarily at those who provide the infrastructure for analyzing the datasets (i.e. Infrastructure Provider from Section 2).

With that in mind, we decided to encode each DUA and its approval workflow as a Directed Acyclic Graph (DAG) with properties attached to nodes and edges. The Data Owner creates the workflow template for the DUA before advertising the dataset for access, e.g., via Dataverse. The prototype represents the DUA template using GraphML, prepared using standard property graph editing tools such as yEd.

Each node in the DAG represents a required assertion from some principal role; the edges denote dependencies or other relationships among the nodes. There are two mandatory nodes: 'Start' and 'Stop'. Each node has a unique ID and is tagged with one or more 'principal roles' depending on the type of the principal or principals that must make some assertion.

The specific structure of the DAG depends on the complexity of the DUA and the number of conditions and assertions that it requires. The two simplest corner cases for such workflows are: (a) a maximum fan-out graph where each assertion node links directly to the 'Start' and 'Stop' nodes or (b) a 'pipeline' graph where all assertions are structured as a linear sequence of dependencies that begin at the 'Start' node and end on the 'Stop' node. In general, any node can be a start of a fan-out to descendent nodes that may complete in arbitrary order. Assertions on a node for which one or more nodes serve as prerequisites cannot be attested to until all precursor nodes are satisfied. Additionally, the DAG may contain 'checkpoint nodes' that require an assertion from some principal role, like 'Data Owner'. These checkpoint nodes enable the principal to validate and approve specific steps or justifications entered by others, before proceeding with the workflow.

A DAG may also contain conditional paths. A node type distinguishes a regular node from a conditional node (an equivalent of a 'switch' statement'). Conditional nodes force the workflow down a particular path, ignoring others. As a simple example of a conditional node, consider a question to a researcher: "Will you be using a hosted server or your personal computer to process this data?". This node has two mutually exclusive descendent branches to select from depending on the response.

The Notary Service interprets these graphs to execute the DUA as it interacts with web users matching the various principal roles, as described further below in Section 5.1. The Notary Service presents the DUA conditions to its users according to the DUA's encoding in the workflow DAG template. Additional node properties encode other information that the Notary Service uses to interpret the graph as it executes the DUA. For example, nodes include human-readable text to display when presenting the item to a qualifying principal in a web session. When a user answers a prompt or clicks to accept a condition, the Notary Service records assertions to issue on the user's behalf. The Notary Services considers the DUA workflow satisfied when the 'Stop' node is reached via all valid branches.

There are several reasons to decompose complex DUAs into multiple workflow DAGs. This decoupling enables a modular structure in which common requirements are packaged and certified once and then incorporated into the access policies for

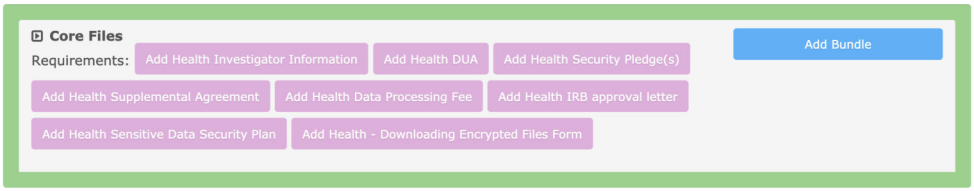


Fig. 3. CPC core policy
Source: https://data.cpc.unc.edu/projects/2/view.

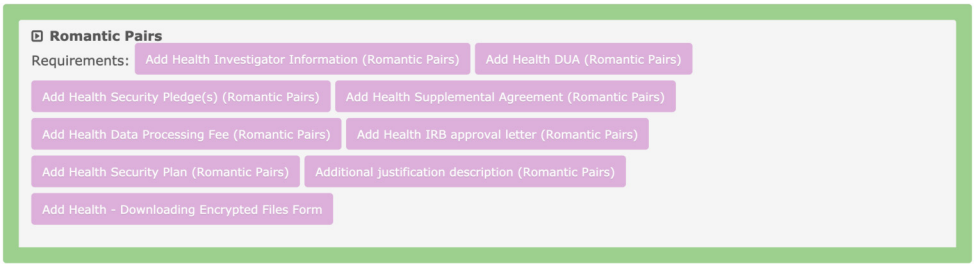


Fig. 4. CPC romantic pairs policy.
Source: https://data.cpc.unc.edu/projects/2/view.

multiple datasets. We also found it convenient to separate DUA clauses with distinct purpose into separate DAGs. For example, many DUAs used in our prototype combine a 'research approval workflow' and an 'infrastructure approval workflow'. The former deals with the research aspects of the proposed analysis on the dataset, and is completed primarily by researchers and institutional governance officers. The latter deals with certifying safeguards and properties of infrastructure used to analyze the data. Research approval workflows are project-specific. If a researcher uses the same dataset for multiple purposes (e.g., in separate projects with separate IRB approval), it may be necessary to complete a DUA workflow separately for each project.

4.2. Exemplary policy scenario: Add health

We illustrate and evaluate our approach and prototype with several representative real-life DUA policies from the National Longitudinal Study of Adolescent to Adult Health (Add Health) [18] conducted by the Carolina Population Center (CPC) at UNC. CPC makes a number of its public and restricted Add Health datasets available to the research community under a variety of policies [19] and procedures, as described on their website and in collections of DUA forms for users to sign manually and file with CPC, typically with a processing fee.

Our purpose was to evaluate the effectiveness of the DUA workflow formalism to encode and implement real-world data access policies. We selected policies for two restricted classes of Add Health datasets: 'Core Files' and 'Romantic Pairs'. We analyzed the process currently employed by the CPC as outlined on their website, expressed the policies using GraphML, and tested them with the Notary Service.

CPC manages a data portal through which access to various dataset policies can be requested. Usually a policy consists of a set of PDF documents for various principal subjects to fill out and sign. Subjects include the PI for the majority of documents, some attestations from every staff member who comes in contact with the data, a letter from IRB approving research etc. Generally there is a base DUA for the type of data being requested, and separate conditions on how the researchers handle (store and process) the data. The sets of forms for the two example sets (Core and Romantic Pairs) are shown in Figs. 3 and 4.

The DAG format is a good fit for describing these policies, leaving a lot of freedom to the Data Owner in the level of detail at which to encode the policy. In the simplest/trivial case the Data Owner may choose to create a graph with 3 nodes - 'Start', 'Agree to the DUA' and 'Stop'. This leaves all of the details contained in the DUA process to be captured outside the workflow, perhaps in a paper form and the Data Owner simply acknowledges to the Notary Service that they have received it and are satisfied. Further along the automation scale, each clause in each of the forms can be represented by a node in a graph thus allowing Notary Service to capture at fine granularity which principal attested to what. Additional information (documents, choices or written statements) can be captured as part of the attestation on individual nodes. For the purpose of this evaluation we chose to capture each form in a separate graph node, to keep the discussion in this section manageable. We have also measured larger and/or more complex synthetic structures (see Section 8.1).

The Core File policy shown in Fig. 3 is encoded in two graphs: a Research workflow and an Infrastructure Approval workflow, shown in Fig. 5(a) and (b). Each node in each graph indicates the type of attestation sought and the principal role expected to provide that attestation (indicated in parentheses and with node color). The difference between the two figures is the 'asynchronous' nature of the second example. In (a) the Data Owner decides that they want to validate the forms, thus nodes coded for Data Owner are present on all branches of both workflows, allowing the Data Owner to interpose on the workflow to inspect and approve the intermediate state. In (b) the Data Owner permits the Notary Service to record the attestations autonomously, so those nodes are absent.

Another interesting aspect is shown in the Infrastructure Approval workflows. Add Health requires the PI to attest/select how the data will be handled, e.g., how it will be stored. Depending on the choice a different type of Security Plan document must be submitted. Notary Service can track the choice and store the appropriate document for future reference (or for inspection by Data Owner).

Fig. 6 shows a swimlane diagram of interactions of different principals with the system using the Core Synchronous policy in Fig. 5a. For consistency the numbering of nodes in the policy workflows is preserved in the corresponding nodes of

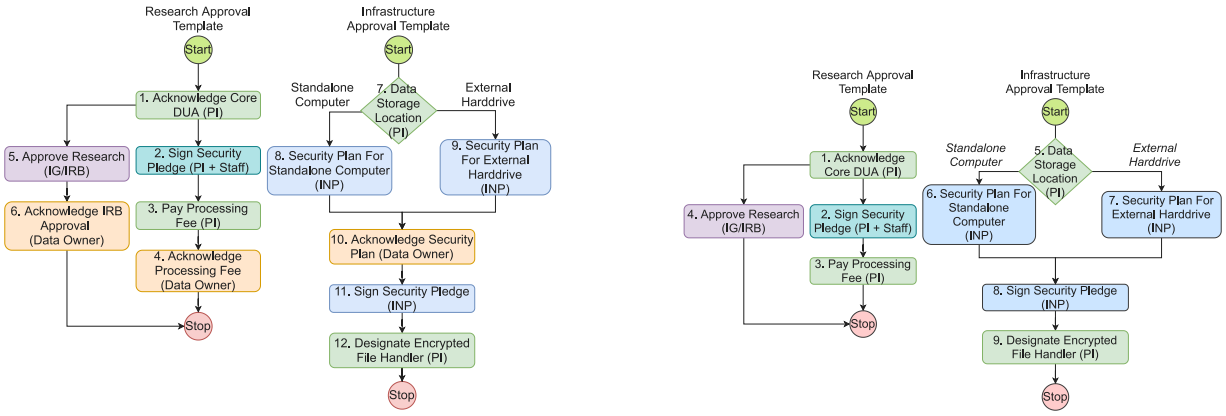


Fig. 5. Add health core files policies.

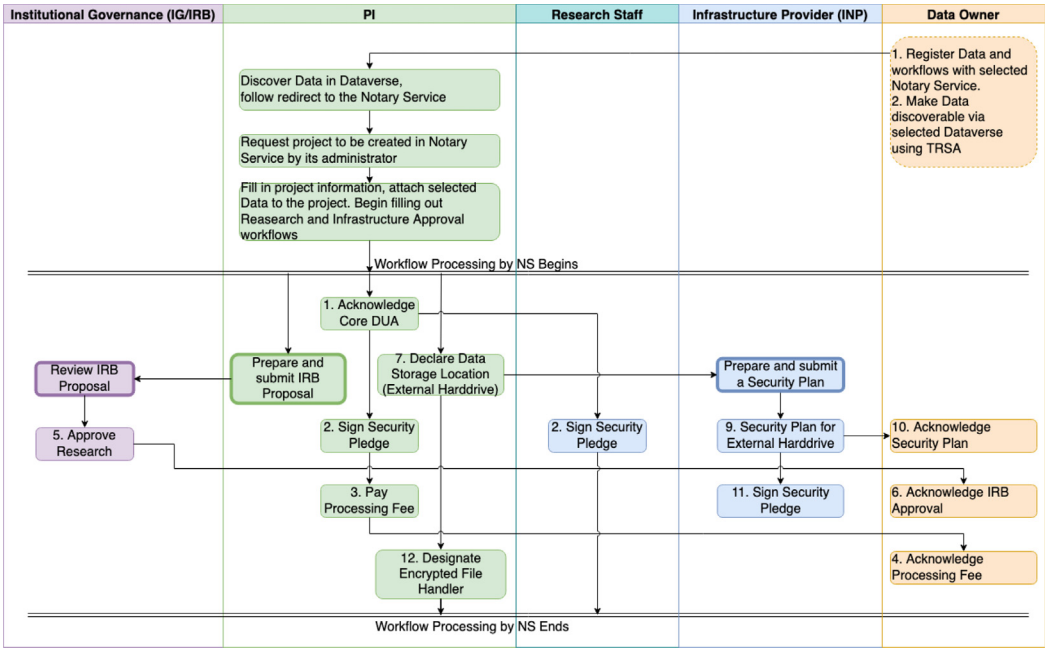


Fig. 6. Swimlane diagram of principal interactions for add health core DUA.

the diagram. Unlike the workflow, the diagram captures actions happening outside Notary Service or even outside the system.

In Fig. 6 the PI discovers the data, is redirected from Dataverse to the Notary Service instance, where they can identify a project as a context for their use of the Data. The PI fills in project information and attaches selected data to the project. This allows the Notary Service to instantiate the workflows registered by the Data Owner for each project and/or for each user. From then on the various principals interact with those workflows attaching attestations, using their verified federated identity. Some steps (shown using nodes with bold borders) may happen outside the system (like e.g. filing the IRB proposal or preparing the security plan). These are a matter of choice for the Data Owner at policy creation time. They may also choose for these documents to be uploaded to the Notary Service for later inspection.

When the workflow processing is completed for both workflows, PIs or research staff working for them can ask the Notary Service to provide an access token for the data. At the time of the request, the Notary Service validates that workflows are properly completed, it then generates SAFE logical statements for the requesting principal confirming that the DUA policies set by the Data Owner are now satisfied. The principal using a token generated by the Notary Service is then redirected to the Presidio instance guarding access to the data. Presidio, using information in the token validates that the DUA is satisfied for the requesting principal by locating and validating appropriate SAFE assertions according to logical policy rules.

The policy for the 'Romantic Pairs' dataset is somewhat more complex compared to the core data: it requires a separate DUA, a Supplemental agreement and a justification, as well as additional security pledges from the IT personnel handling data (not just from researchers) and a designated individual for handling encrypted data (see Fig. 7). In this example we chose to break up a single form (Security Plan) into three separate independent nodes ('Provide a List of Contract Personnel' for the PI, and 'Provide Description of Computer System Storing Sensitive Data' and 'Acknowledge AddHealth Required Security Procedures' for the Infrastructure Provider), as shown in the Infrastructure Approval Template graph. Also we chose these workflows to be 'synchronous' by adding Data Owner assertion nodes in both graphs, which block progress through the approval process until the Data Owner approves the application so that it can proceed.

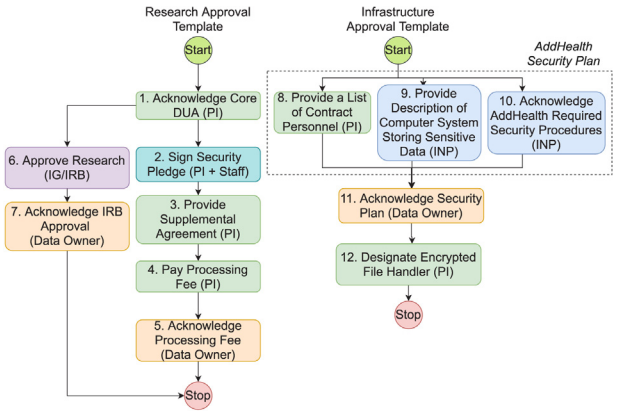


Fig. 7. Research and infrastructure approval workflows for romantic pairs dataset.

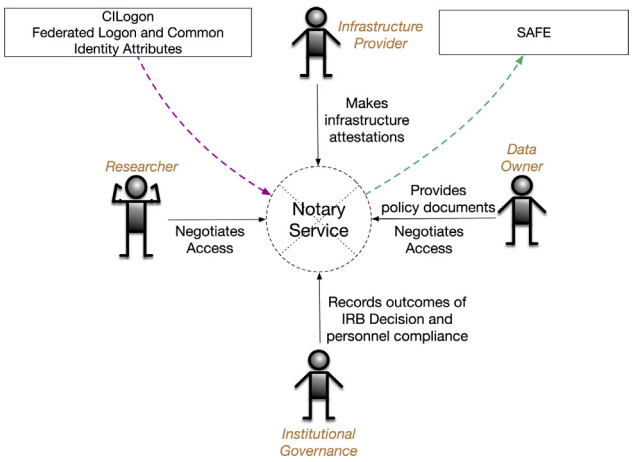


Fig. 8. Notary service interactions with principals.

Together the approach and features we describe here provide for a broad set of options for how to encode DUA policies, how much of the policy outcomes is stored in the Notary Service vs. out-of-band and the level of involvement individual principals (particularly the Data Owner) choose to have with each data access request. This allows to adjust the amount of 'friction' in obtaining access to the data, e.g., based on its sensitivity. Access to less sensitive datasets can be trusted completely to a given Notary Service to simply record the attestations and establish an audit trail if anything goes wrong. More sensitive data sets may require workflows with Data Owner review to approve release to responsible parties.

5. Notary Service and policy checking

5.1. Notary Service

Notary Service (NS) consists of a web service interacting with different client principals – Researchers, representatives of Institutional Governance, Infrastructure Providers – via their User Agents (e.g., Web browsers). It accepts DUA policy descriptions and associated workflow forms from Data Owners. The elements of a DUA are specified as workflow DAGs encoding the different phases and facets of the DUA, as described in Section 4.1. The NS presents views of those documents to other principals within authenticated web sessions, allowing them to accept and/or certify various conditions required in a DUA. The nodes in the DUA workflow graphs are tagged with the type of principal or the principal role that must make the respective attestation within the workflow.

The NS acts as a digital witness by issuing signed attestations that the required approval tasks are complete. The data provider agents guarding access to the data (e.g., Presidio) validate these attestations to make access control decisions before serving the dataset (Section 5.2). These components use SAFE (Section 5.3) to issue, fetch, and validate the attestations according to the Data Owner's policy, which may require multiple distinct approval workflows along with other conditions. Fig. 8 illustrates these interactions.

We envision multiple possible deployment scenarios for an NS instance. For example, an NS may run on behalf of an academic institution, a consortium of institutions, or a consortium of data owners. The dataset access policy establishes the Data Owner's trust in the NS and other conditions of access via logical policy rules in SAFE (Section 5.3). The declarative policy enables any of these scenarios as a choice at deployment time.

Principals authenticate to NS using their institutional credentials via the CILogon [2,9] platform for identity and access management. Users sign on to an institutional identity provider (e.g., Shibboleth), which releases identity attributes to CILogon as a registered service provider approved by the institution. CILogon acts as a bridge for users to authenticate a web session to an NS instance and grant permission to release attributes to an NS instance without any change to the identity provider's policy for attribute release. CILogon eliminates any need for ImPACT infrastructure to maintain user accounts or for institutions to reconfigure their identify services to enable ImPACT. ImPACT can take advantage of any identity services supported by CILogon.

NS uses the identity attributes to infer affiliations and roles that a given principal is empowered to assume in the negotiation or approval process for a DUA. These attributes include institutional (e.g., InCommon) attributes representing institutional roles, supplemented with memberships and roles in other groups. These groups and attributes may be maintained externally to an institution via COmanage, a federated platform to manage cross-institutional Collaborative Organizations with groups and roles. CILogon integrates with COmanage to obtain these user identity attributes and release them to NS with user consent. The NS matches these roles to the roles required for the approvals and certifications in the DUA. In this way the NS user interface presents different views and appropriate task prompts to each user according to the user's role in the process at each stage. Note that a principal may be able to take on different roles depending on the context.

NS is implemented as a Python Django application. It interacts with a CILogon instance using the OIDC protocol to authenticate user sessions and collect their identity attributes. At the back end, NS relies on a relational database and an instance of Neo4j graph database to store its state. Workflow templates are imported into the NS by Data Owners and loaded into Neo4j in the form of GraphML-encoded files. NS verifies that they conform to a set of rules expressed using Cypher language to ensure they have a sane structure.

In order to gain access to data a researcher representing a project on behalf of a given institution adds the dataset of interest into the project defined in NS. NS instantiates a workflow for a specific project/institution/dataset tuple from a registered GraphML-encoded template, creating a new graph object in its Neo4j database. It then generates tasks needed to complete the graph for principals that match roles encoded in the nodes. As eligible users with matching attributes interact with the NS, it

Table 1
Structure of JWT claims in the current NS implementation.

Claim name	Claim description	Type
data-set	SAFE token pointing to the dataset being accessed (generated at the time of dataset registration with NS)	String, private
project-id	Globally unique identifier of the project in Notary Service.	String, private
ns-token	SAFE token of the Notary Service generated from its public key	String, private
ns-name	Human-readable name of the Notary Service	String, private
iss	Issuer of the token. NS FQDN	String, public
sub	OSF DCE rendering of DN attributes from principal's X.509 cert	String, public
exp	Expiration date	Date, public
iat	Issued at date	Date, public
name	Full name of the subject/principal	String, public
ver	Version of the encoding	Private

dispatches tasks to those users, collecting their assertions for the DUA and updating the workflow instance.

Workflow traversal through the web UI is based on the roles of the user principal in a web session. NS analyzes the state of the workflow instance using the state of individual nodes and presents the principal with one or more available assertions needing to be made using one of the roles the principal has assumed. A principal can be blocked from making further assertions by dependencies on other principal,s actions. NS can present a principal with two lists:

- The list of incomplete assertions immediately reachable from the current state of the workflow;
- The list of all assertions reachable for the principal's role that still need to be completed.

If the latter list is empty, the principal has no further actions for this workflow.

NS implements a simple completion-checking algorithm that verifies that all valid branches of a given graph have been completed and 'Stop' node has been reached. Upon completion NS issues a JSON Web Token (JWT) to any principal who is defined to be part of the project. The user's agent passes this token with any data access request to fetch the dataset. For example, on workflow completion NS generates a link that the user can click to fetch the dataset from a Presidio instance, passing the token in the request. The token includes various parameters that Presidio needs to validate access to the data, including the user's current project and the identity of the user's NS, as described below.

Table 1 shows the claims provided by NS in the currently implemented JWT.

5.2. Presidio

Presidio is a data server with an API to browse a repository and download files, enabled for ImPACT authorization. It presents a simple service targeted at browsers, although interactions via e.g. cURL [20] are also possible. We envision that a Data Owner deploys Presidio as a shielded external interface to a data repository. Presidio imports logical policy rules for each tagged dataset from the Data Policy Store indexed by a dataset ID. Thus Presidio may also run as a third-party storage provider for one or more Data Owners who trust it to protect their data. Either configuration allows for autonomy of decisions by the owner with respect to its data access policies.

Presidio is implemented as a Python Flask application and deployed under a Gunicorn [21] uWSGI HTTP server. All client connections are encrypted (by Secure HTTP with TLS) with bidirectional authentication. For a user browser session, the user presents a client X.509 certificate issued by CILogon, and incorporating the user's Distinguished Name issued by their affiliated institution's identity provider. The Notary Service offers a convenience link for a user to obtain a certificate. Typically the user must import the certificate into their browser before contacting Presidio, but they may also use the certificate for programmatic access using cURL or other client user agent.

To check the client's access for a dataset ID, Presidio invokes a standard ImPACT guard script in a trusted SAFE process through a secure channel. The guard fetches, caches, and applies logical policy rules governing access to the requested dataset, issued by the Data Owner. The guard also requires a project identifier (an identifier issued by COmanage) that establishes a context for the request, and an identifier (including public key hash) for the client's selected Notary Service. With these parameter values the Presidio SAFE guard is able to index and fetch all trust metadata elements relevant to the access decision. Section 5.3 describes the guard script and trust metadata flow in more detail.

The client provides these values in the JWT token issued by the Notary Service. The user's browser or other agent passes the token to Presidio in an HTTP request header with the client request. Presidio renders it into a cookie for the client's web session.

Presidio offers two mechanisms for its operator to label stored files and directories with dataset IDs. The integrity of these labels is paramount because SAFE uses them to index and authenticate the policy governing access to the labeled files. For file systems with OS support for user-extensible file attributes, an operator can issue OS-specific commands to annotate the files and/or directories with dataset identifiers. As an alternative mechanism that is OS-independent, Presidio can parse YAML files in the served directories. The YAML markup declares label assignments based on rules to match patterns in the pathnames. Presidio assigns to each file the label from the closest (most specific) ancestral path prefix by either method. A given dataset identifier may cover multiple files and directories: all data objects with the same dataset ID share the same access policy indexed by that ID. Unlabeled files are not served: Presidio reveals no information about files or directories for which it cannot authorize the requester.

5.3. SAFE policy checking

SAFE is a platform for logical trust management for multi-domain distributed systems. SAFE runs as a server process to issue and/or process certificates containing statements in a standard logic language (Datalog [22]) adapted for use as a trust logic following Binder [23] and its successors. Trust logic is a powerful and expressive formalism to represent assertions and rules encoding authorization meta-data and policy. It includes a well-defined and provable inference procedure to validate that a set of statements – encoded in a set of logical certificates – complies with a policy.

Trust logic enables ImPACT to express rich authorization policies involving assertions and rules issued by multiple principals. A defining property of a trust logic is that each statement is attributed to a principal (the speaker), and policy rules may consider the speaker of each statement and its properties as conditions for accepting the statement. To authenticate their statements, SAFE principals (or their programs) encapsulate them in logic certificates—a set of one or more logic statements spoken by the issuer and signed under its keypair, with an expiration date (TTL).

Because the logic is signed, other principals may rely on statements spoken outside of an authenticated session. It is safe to cache and share valid unexpired certificates. These properties are important because they facilitate policies and interactions that involve multiple parties, as is common in our target usage scenarios. Logic statements may endorse the public keys of other principals and delegate specific limited trust to them, supplanting the need for dependence on an external PKI.

The SAFE [11] platform defines a certificate format for signed logic payloads, and a validation engine for policy checks incorporating an off-the-shelf Datalog engine (Styla). A SAFE server process runs on behalf of a principal and runs simple trust scripts approved by the principal. Other software controlled by the principal invokes the script APIs through a secure channel. If the principal invokes scripts that issue certificates, then the SAFE process wields its signing key.

ImPACT adopts the design principle to limit certificate handling to core components and services (e.g., in Fig. 2), which may produce and/or consume certificates. Most users of the system see only familiar identity management systems. They use single sign-on to authenticate their login sessions with core services, which may issue statements to certify their interactions within the session. For example, the Notary Service obtains user identity attributes via CILogon, as described in Section 5.1. In this way, the ImPACT architecture avoids burdening users with the need to manage keypairs and public key cryptography to support the authorization system, except insofar as they use CILogon-issued client certificate to access Presidio. The CILogon certificate is managed automatically, is easy to use, and enables programmatic access to Presidio (e.g., using cURL).

The ImPACT prototype includes SAFE scripts for a Data Owner, a workflow publisher (who may be distinct from the Data Owner who relies on its DUA workflows), Notary Service, and Presidio data provider. The scripts include standard certificate templates with parameterized logic for trust metadata exchanged among these participants in an ImPACT system. These scripts comprise about 200 lines of script/logic code, including exemplary policies and trust structure for a Data Owner.

In the ImPACT scenarios, all component principals that interact with SAFE are issuers of logical assertions and/or policy rules—except Presidio, which acts as an authorizer to check compliance. The components issue statements by invoking script entry points with various string parameters, which the script materializes into their statements. They include attestations issued by the Notary Service (NS), dataset policy rules, infrastructure-related attestations, and certifications that establish the Data Owner's trust in the NS instance and/or Infrastructure Provider.

These issuer scripts post signed logic certificates into a shared certificate store. In this paper we refer to the SAFE store as the Data Policy Store. It is based on a variant of a canonical key–value storage abstraction. Each certificate has a unique index key, or *link*. The link is self-certifying: it is derived from the issuer's identity (a hash of its public key) and a parameterized string label chosen by the issuer. The architecture of the store is out of scope: in our experiments we use an enterprise key–value store (Riak) operated by a trusted party. If the central store fails or is compromised, an attacker can mount a denial-of-service attack, but it cannot subvert the protection system because all certificates are cryptographically signed. For enhanced security a decentralized structure such as a blockchain platform or a collection of encrypted (HTTPS) web servers (operated by the issuers) would serve the purpose.

The scripted key–value store model enables issuers to link their certificates together to form chains and DAGs, and also to link to certificates issued by other parties. Given a logic link, a trust script can retrieve the certificate and its link closure. A SAFE process caches the logic sets that it encounters, indexed by their links. A warm cache of authenticated and unexpired logic minimizes the need for network communication or cryptography in the authorization path for decentralized trust scenarios.

The trust scripts developed for ImPACT make extensive use of certificate linking. Policy packages issued by the Data Owner link to compliance check rules for each workflow DUA required by the policy, and issued by the workflow provider (which may or may not be distinct from the data owner). The Notary Service issues attestations for required workflow elements, linked from a root receipt keyed by the dataset ID and template instance parameters—the Researcher's distinguished name and affiliated project. Linking is also useful to construct federated governance structures. For example, a Notary Service or Infrastructure Provider may link to endorsements from other parties, such as a project owner or consortium root, that the Data Owner's policy may require to establish trust in those components.

To perform an authorization check, a SAFE application invokes a *guard* script. A guard evaluates a collection of certificates: it takes the union of the logic sets in the collection, merges the statements to form a single query context (a logic program), and issues one or more logical queries against it. A guard script specifies how to assemble the context and what query to issue against it. For ImPACT the authorizer is Presidio, which invokes a guard to check access to a dataset ID, passing various string parameters. The guard uses the parameters to generate certificate links from parameterized templates in the script. After assembling the context, it issues a logic query that asks: "Do the statements in these certificates allow me to prove that the subject qualifies for access to the requested dataset according to applicable policy?"

Although Presidio's guard is a standard script with a standard query, it applies a dataset-specific policy issued by the Data Owner. The policy rule package is indexed in the certificate store by the dataset ID and signed under the Data Owner's keypair. Presidio's SAFE engine imports these rules and their closure, validates signatures against the Data Owner's public key, and inserts them into the logic context. All policies use the same query predicate, but the owner's rules define how to evaluate the predicate for that specific dataset. Crucially, the dataset ID is self-certifying: it includes a hash of the owner's public key, enabling the SAFE engine to authenticate the policy. The dataset IDs are known to Presidio and installed by its trusted operator, e.g., acting on behalf of the Data Owner, as described in Section 5.2.

The owner's policy might require specific identity attributes, such as affiliation with an approved institution and project, as well as completion of specific DUA workflows certified by an approved Notary Service. The policy may also designate principals whose assertions are trusted, e.g., via rules that define which institutions and notary services are approved. The logic check validates the trust paths, matches the NS attestations to the requesting user's Distinguished Name, and validates access policy conditions. In particular it verifies that all required NS attestations are present and matched, that the requester matches the workflow attestations and (optionally) possesses specified attributes, affiliations, and/or project or group memberships needed for access. It may also validate that the request originates from qualifying infrastructure.

6. Data discovery with Dataverse and TRSA

Dataverse was a good fit for ImPACT to play the role of the Data Discovery Service, however the ability to extract the metadata details required Dataverse to transfer the file into Dataverse repository in order to process the information there. When data is either sensitive and controlled by restrictive data use agreements the ingest process of Dataverse falls short. To address this shortcoming we designed and implemented a Trusted Remote Storage

Agent (TRSA) that can solve this problem and allow the pre-processing of data to occur in situ under the control of the Data Owner, only publishing the selected metadata within a selected Dataverse instance for discovery while the sensitive or large data remains in the secure remote storage location.

The TRSA is deployed locally in the Data Owner environment and all processing takes place locally before the metadata is pushed to a pre-configured selected Dataverse instance using the published API. Consistent with the principle of autonomy, the metadata collection is done locally by TRSA configured by the owner to perform only certain actions on the datasets and not to disclose any metadata the owner is not willing to share. This allows the Data Owner to fully assume and control the risk of unintended disclosure. While the metadata depends on the type of data being shared, for typical social science data, it is represented by selected column names, types and limited statistics (mean, standard deviation) of those columns.

Dataverse issues DOIs (Digital Object Identifiers) to the registered data objects making them referenceable. The ability to issue DOIs by a given Dataverse instance is contractually bound to a guarantee that the objects remain reachable using a consistent URL as per rules set by IDF (International DOI Federation) [24]. With the introduction of TRSA data storage and discovery become separate responsibilities, the former of the Data Owner and the latter of the entity running the Dataverse instance. This requires Memorandums of Understanding (MOUs) between archives hosting the Dataverse instances and Data Owners using TRSAs wishing to advertise through those Dataverse instances. The MOU requires the owner to ensure that updates to the data collection are noted in the Dataverse system publishing the metadata. This prevents invalid or broken links from persisting in the system.

The architectural design of the ImPACT TRSA is based on a light weight approach that records submissions locally in text format and utilizes Dataverse API calls to track the progress of submissions. The use of Dataverse API by TRSA allows it to acquire a DOI for the publication of the data and publish the selected metadata to a selected Dataverse instance. By leveraging the public Dataverse APIs to publish metadata the tool remains scalable and sustainable for many organizations. As an added benefit, the TRSA also has application for data repositories that house very large data that are impractical to move to Dataverse, even if it is public, thus still allowing the indexing of detailed metadata and discovery of these very large datasets. Its design accommodates scenarios with and without the Notary Service and other ImPACT components, depending on the specifics of the use case.

7. Analyzing private data in a Protected Data Enclave

The Protected Data Enclave (PDE) provides the final capability within a researcher's workflow for operating on restricted data. The purpose of the PDE is to allow researchers to interact with data and analyze it in a manner that is compliant with specific data security obligations imposed as conditions of use, e.g., due to respondent confidentiality concerns, proprietary and other data privacy concerns, regulatory restrictions, and data use agreement or other licensing terms.

In the ImPACT prototype a PDE is based on enclave deployment templates provided by Duke University. A Protected Network (PN) is Duke's name for a technical infrastructure configured for secure storage and processing. A PN is a secure network environment configured and controlled by University IT with suitable defenses such as firewalls, intrusion protection systems, and network access control lists (ACLs). Each PN hosts dedicated virtual compute environments configured for its purpose. Duke uses PNs to store and analyze sensitive data such as HR, business operations, and academic records, as well as restricted research data. The key contribution of the ImPACT project is taking the PN architecture and making it *reproducible and automated* as well as *abstracting* a number of elements, so they can be implemented using different components depending on the hosting organization's deployment choices.

Ideally, a PDE is constructed and deployed on demand for a given project or purpose. By reducing deployment time and cost, a fully automated and reproducible PDE deployment reduces the temptation to add hard-to-manage individual servers or other configuration changes to an existing protected enclave. Our approach uses automation to provision the firewall rules (network and host), the network itself including subnets and ACLs, dynamic virtual machines (VMs), software, and storage. The firewall rules are fairly consistent between different instances of a PDE—they allow administrative systems to access hosts inside of the PDE for patching or services like DNS or LDAP.

Automation of the provisioning of a given virtual machine with base software is relatively easy, but allowing researchers to build more complex software environments while at the same type limiting access to the public internet can be challenging. Within the PDE, we have used container deployment methods for both Singularity containers [15] and Docker containers [25] which rely on continuous integration services (CI) provided by our GitLab [14] server to build container images outside of a PDE, validate and test them for vulnerabilities according to specified policies, and then import them into the PDE. The goal is to restrict access on data transfers into and out of the PDE consistent with DUAs, but still allow researchers access to the tools they need, as well as websites and support communities, to conduct their analyses. Examples of external data sources might be docker build scripts, patches, or repositories that are discipline-specific and scattered across the internet. Other examples are repositories like the comprehensive R Archive network (CRAN). Researchers can develop their methods and toolsets outside of a PDE and then move them into it when ready.

Once the container with tools is built and instantiated within a VM inside of a PDE, the user can run it and have access to the full toolset they built outside the PDE. Thus users can customize the toolsets they use in the enclave for their project without compromising the security of the PDE or taking significant time from support staff to vet their tool choices.

As a strong defense against exfiltration, external access to a PDE is limited to web-based desktop access via Proconsul [12]. With this approach experimenters have a 'glass pane' access to a virtual environment similar to a desktop experience, but with significant amount of controls on data flows in and out of the enclave system. Proconsul controls which researchers have access to which parts of the enclave through a policy set by the IT staff.

Overall, a PDE is a simple construct to allow users to access sensitive data in a secure fashion that minimizes the friction typically associated with operating on a remote service. It can reproduce the desktop experience and also give the researchers access to the same tools they use when doing work on less sensitive data. The key to achieving this is automating as much of the deployment as possible—from building the networks, firewalls, and VMs, to the tools used by the researchers to build, update, and deploy the tools they use.

8. Performance evaluation

All data presented in this section is archived with Zenodo under DOI 10.5281/zenodo.4420281 [26].

8.1. Performance of workflow graph processing in notary service

As described above, Notary Service is a portal implemented using Python Django framework [27] which allows different principals to interact with dataset approval workflows in order for researchers to obtain permission to access data held by the data owner. Data owner creates a workflow and uploads it to the Notary Service, and when a researcher instantiates a project and adds the desired dataset to it, the workflow is automatically instantiated to be filled out. Each node in the workflow graph is a requirement for a statement or assertion from some principal.

From the point of view of processing times, we wanted to evaluate 4 different stages:

1. How long it takes to load a graph into the Neo4j database
2. How long it takes to validate the graph according to Notary Service ruleset
3. How long it takes to check the graph for completeness when it is empty
4. How long it takes to check the graph for completeness when it is completed

We designed several types of synthetic workflow graphs: they have a common Start node with a m child vines of depth n nodes converging on a common Stop node . We built 4 synthetic graphs with mxn 1×2, 3×3, 5×5 and 7×7 rows and columns. The small number of realistic workflow graphs we constructed based on DUA procedures of different providers have a relatively small number of nodes (10–25), so while these test graph structures are not fully representative of real workflow graphs, they allow us to explore corner/worst case performances and scaling in terms of numbers of nodes and depths of individual branches as well as the impact of storing many workflow graphs in a single Neo4j database.

It is worth pointing out that stages 2–4 above are implemented using Neo4j Cypher queries combined with procedural code inside of Notary Service. Stage 1 is executed inside Neo4j and includes no queries, other than the command to load the graph from file into Neo4j. For each of the graphs in this section 1000 iterations were performed to remove sensitivity to other processes happening within the same host. With each graph loaded, the database size grew, thus also allowing us to evaluate performance penalty as more workflow graphs are added to the database.

The test environment was a virtual machine in a VMWare environment: vSphere Client version 6.7.0.30000, VMware Tools: Running, version:10309 (Guest Managed), Compatibility: ESXi 6.0 and later (VM version 11). The VM itself was configured as follows: 4 CPU, 16 GB RAM, 80 GB Disk, CentOS 7, running both the Notary Service and a Neo4j instance (v. 3.5.0/APOC 3.5.0.1) as Docker containers.

Figs. 9 and 10 compare different stage times in absolute terms, as well as time per node (stage time divided by the number of nodes in the graph). Time is shown in seconds, 95% confidence intervals are also plotted. We notice that all stages show similar scaling w.r.t. to the number of nodes in the graph, as one would expect, with larger graphs being able to amortize some any initialization activities happening within Neo4j for each graph. One exception is testing the graph completion of an empty graph, which is nearly linear, i.e. independent of graph size, since the evaluation algorithm exits early if it finds that the workflow graph is incomplete.

To better demonstrate the performance impact of a larger number of graphs loaded into the database, we also show in Fig. 11 the sequence of time measurements for time-to-validate of a 7×7 graph from 1 to 1000. As workflow graphs accumulate in the database we see a jump around 450, where Neo4j

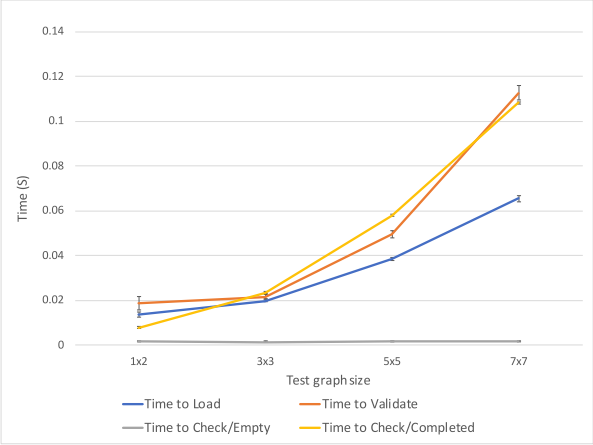


Fig. 9. Notary service stage times.

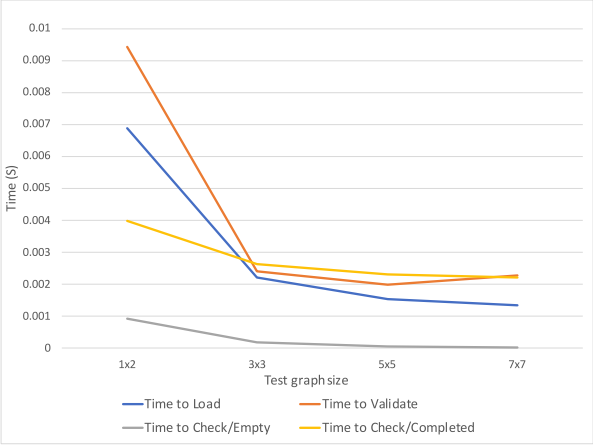


Fig. 10. Notary service stage times Per-Node.

clearly changes its behavior, we suspect due to a change in index memory management. The following indexes were implemented in Neo4j for all tests:

```
ON :Node(GraphID) ONLINE
ON :Node(GraphID, ID) ONLINE
ON :Node(GraphID, ID, Type) ONLINE
ON :Node(GraphID, Type) ONLINE
```

8.2. Performance of Presidio SAFE-based data access verification

As described in Section 5.2, Presidio is a Python gunicorn [21] app proxied by Nginx [28] that allows browsing and downloads of a restricted portion of a filesystem, protecting access to specific data artifacts. Only a principal (a researcher) named in suitable attestations from a qualifying Notary Service – stating that the workflow requirements for this dataset are complete – can be granted access to viewing and downloading the dataset. The app was configured with a single worker in order to produce reliable measurements; otherwise, the random nature of request proxying by nginx across multiple back-end worker processes would have caused high variance in the measured results due to potential imbalances across workers.

Since the download speeds are primarily affected by the choice of protocol in the browser for transferring data (TCP, QUIC, other)

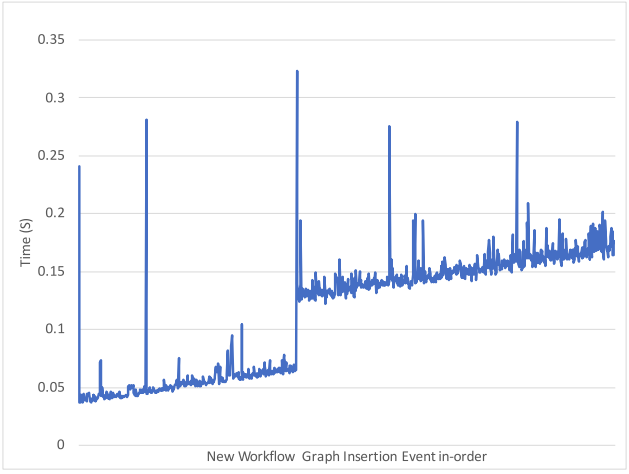


Fig. 11. Notary service time-to-validate sequence.

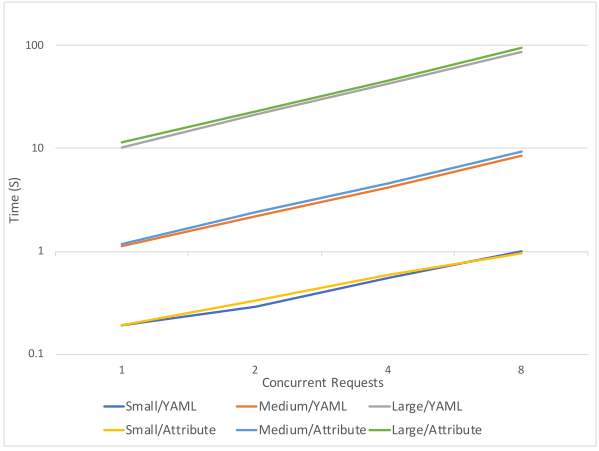


Fig. 12. Presidio performance with YAML and attribute tags and no overrides.

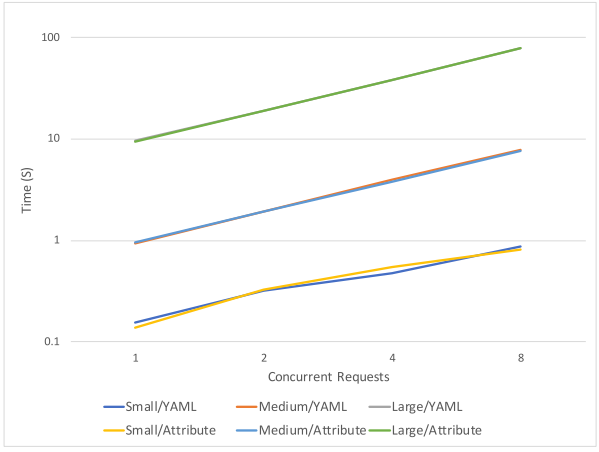


Fig. 13. Presidio performance with YAML and attribute tags and three overrides.

and thus are independent of ImPACT architecture, we chose to evaluate the times needed to list different directory sizes under different configurations. A listing is triggered by a principal who had valid certifications from the Notary Service and thus had the right to list the full directory, Presidio had to evaluate that promise and consult the SAFE policy engine. The SAFE access control decisions were being cached for 2 s in Presidio; SAFE system was also running on same host as Presidio.

The directory sizes we chose to evaluate were *Small* – 500 entries, *Medium* – 5000 entries and *Large* – 50,000 entries. The two configurations tested were using YAML directory configuration files and filesystem attributes to specify the file access policies in the filesystem.

The evaluation environment was a virtual machine with 2 vCPU VM, 8 GB of RAM, pCPUs underlying vCPUs are: Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.30 GHz.

Figs. 12 and 13 demonstrate the results. In Fig. 12 we compare times it takes to list a directory which has no override rules and in Fig. 13 a directory has 3 override rules:

```
version: 1.0
default: <policy reference>
overrides:
   'file_.*0$': notOK
   'file_.*2$': notOK
   'file_.*5$': notOK
```

By using regular expression matching it excludes from listing files whose names end with '0', '2' and '5'. Since file names are in these directories are based on GUIDs, we assume a uniform distribution of last characters in these filenames, thus singling out 30% files with these rules.

Figs. 12 and 13 are plotted in log-scale (log_2 in X axis and log_{10} in Y axis) and compare the performance of Presidio with 1, 2, 4 and 8 concurrent requests. The results show low polynomial $y = C * x^a$, $a \approx 0.8$ scaling with the number of parallel requests to list different size directories. The rule overrides actually slightly shorten the time to list the directory content likely due to the reduced time required to actually render the page with fewer names by the Presidio app. Also of note is a small but consistent difference in performance between YAML and attribute-based filesystem policy configurations, with YAML configuration actually performing slightly better.

9. Related work

The proposed ImPACT architecture is sufficiently novel not to have direct parallels with existing projects. Some of the individual aspects or elements of ImPACT have antecedents that we describe in this section.

To start, in Section 3 we briefly alluded to the fact that a common solution to the problems ImPACT architecture addresses is to centralize all functions – data discovery, data storage and data processing behind a single portal, with the advantage being the relative ease of use by the researchers – the portal becomes a 'one stop shop' for them and the disadvantage being the loss of control by data owners over their data by way of delegating it to the portal and the fact that if a particular dataset is not part of the portal portfolio, there is no easy way to incorporate it into the study. The two premiere systems that take this approach are ICPSR [4,29] in the US and ODISSEI [5] in the Netherlands.

ICPSR is a consortium of more than 750 academic institutions and research organizations, which "maintains a data archive of more than 250,000 files of research in the social and behavioral sciences" [29]. It hosts a number of specialized collections related to e.g. aging, substance abuse and targets researchers primarily from academic institutions. ICPSR grew organically around a number of collections started in 1962. ICPSR Portal hosts data and also allows datasets to be contributed by the researchers. It also has processing capabilities for researchers to analyze the data in-situ inside hosted virtual environments.

ODISSEI (Open Data Infrastructure for Social Science and Economic Innovations) also has a large number of partners and serves the purposes of supporting academic research, specifically in social sciences. Through a partnership with Statistics Netherlands it provides access to a number of longitudinal datasets. ODISSEI has four parts - a *data facility*, which stores sensitive data, an *observatory* which assists in collecting new data, a *laboratory* which develops new methods for analyzing data and a *hub* focusing on outreach and education [30]. The ODISSEI portal allows researcher to pose semantic queries over the metadata allowing for a rich discovery environment. A unique feature, as a processing environment, researchers using ODISSEI data have access to a secure HPC environment called Cartesius hosted by SURF - "a collaborative organization for ICT in Dutch education and research" [31]. Some of the recent work from ODISSEI includes investigating practical applications of techniques that support distributed data processing, like Secure Multi-Party Computations and Differential Privacy which are also aligned with ImPACT long-term goals.

Within the space of data access policies, two efforts stand out that relate to ImPACT architecture: the DataTags project [32] from the team that builds Dataverse repository and Researcher Passport [33] from ICPSR.

DataTags are a way to tag the dataset with its sensitivity/privacy level inside collections. Tags use color names to make them intuitive and easy to remember: e.g., blue means it is open-access/public, then there is green, yellow, orange, red, and crimson which connotes that the data needs the most restriction/protections. Each file in a dataset gets its own tag. Each color of a tag has a specific set of requirements associated with it tha define how data possessing this tag must be handled. For instance, blue requires no access credentials, green requires verifying the user's email address. Yellow requires an application and approval before access is granted. Red and crimson require MFA. Tags may also have requirements to how data is protected at rest and during transmission, i.e. via some level of encryption.

DataTags can be viewed as a way to standardize procedures for operating on restricted data into a small number of well-understood 'bins', thus simplifying interactions between data owners and researchers attempting to gain access to data, simplifying DUAs and standardizing handling of the data. In that sense, DataTags are largely orthogonal to ImPACT, and can be incorporated into e.g. Dataverse instances that support ImPACT Notary Service and TRSA—this has the potential of simplifying some of the workflows that Notary Service operates on, making them more easily verifiable.

Researcher Passport is a form of an online identity which helps repositories gauge their trust in the individual researcher. Additionally it has components that help the repository to assess the risk level of the data it contains. Researcher Passport establishes common characteristics of data users and maintains a record of history of "research experience, data stewardship, and education and training" [33]. It establishes a system of 'points' for e.g. having scientific degrees, being part of funded grants, being a faculty member at a credentialed institution, having security clearances, your publication record. Based on these a researcher gets a numeric score, with the higher number indicating a higher level of trust, which allows data stewards/owners make decisions about sharing their restricted data with the researcher.

A score is also assigned to the particular collections of data, such that the data steward can compare the score of the dataset with the score of the researcher trying to access it before making a decision on whether to grant the access. If an affirmative decision is made, a researcher is issued a 'visa', which provides access to the data. In contrast, rather than a rating system for Researchers, ImPACT grants access based on specific criteria and policies encoded in DUAs, attribute checks, and logical policy rules.

SAFE trust logic enables us to tie together ImPACT's elements of programmable authorization, in which multiple parties in a distributed system may issue policies and required certifications, and various services integrate with standard solutions for federated identity management. Studies of authorization logic have yielded many approaches too numerous to detail here, including recent approaches that are expressive but also complex (e.g., NAL [34]). SAFE embraces direct use of Datalog [22], a rigorously defined and extensively studied general-purpose logic language that is a subset of Prolog, a popular language for logic programming with a standard syntax. Our approach merely adds a modal operator **says** to Datalog, enabling its direct use as a logic of belief and attribution; this idea previously appears in Binder [23], SD3 [35], and SENDLOG [36]. Like all of these systems, SAFE uses signed logic certificates as a transport for authenticated logic. Datalog-with-says is at least as powerful as the XACML web standard, and it enables reasoning from authenticated policy rules and assertions gathered from multiple sources, which is crucial in the federated scenarios characteristic of collaborative science. SAFE adds simple programmable indexing and sharing of logic certificates through a key–value store with scripted linking of related certificates via interpolated string templates. Decentralized authorization based on declarative policy and shared key–value stores are beginning to emerge in complex cloud environments [37].

10. Conclusions

ImPACT is enabled by substantial investments in federated identity management for Web-based services, including single sign-on and standardized identity attributes for research institutions (e.g., InCommon). It is a case study in leveraging distributed identity management for advanced authorization. Here are some observations from our experience:

- **The role of CILogon.** CILogon is a key component enabling us to build out support for secure research collaborations. CILogon allows users to authenticate to ImPACT services with their institutional identities, approve and control release of their attributes to these services, integrate collaborative organizations (via COmanage) as sources of extended attributes for their identities, and obtain cryptographic keypairs enabling them to use their identities with "hands-free" computational tools, all with minimal administrative burden on institutional IT.
- **The role of COmanage project groups.** Data owner policies in ImPACT may require memberships in specific federated groups. ImPACT presumes that researcher actions on restricted data are associated with exactly one such project at any point in time: the user selects the project in the Notary Service before completing a DUA, and requests data in a Presidio session bound to the project. COmanage provides a convenient facility to manage project groups and the authority governing those groups as a basis for access control.
- **The need for role attributes.** ImPACT exposes a need for richer identity attributes representing roles within an organization. For example, in our prototype the Notary Service is unable to map administrative users to DUA tasks by institution-supplied identity attributes. That is because our pilot institutions do not maintain attributes representing a user's staff role or governance authority, and such attributes are not yet standardized. An NS operated on behalf of an institution could obtain them from a separate authority

database (e.g., LDAP) outside of the identity management system. Our prototype represents them as group attributes maintained separately via COmanage. However, these COmanage groups are decoupled from institutional authority and are not endorsed by the institution.

Our prototype faces several limitations and challenges. In particular, it can benefit from further tooling at the front-end user interface. Data owners can benefit from tooling to author declarative policy, incorporate off-the-shelf policy elements, and register datasets and policies automatically. Researchers can benefit from improved tooling to manage their workflow through multiple Web points of contact and determined how to resolve unexpected rejections. Currently data use is limited to download or a desktop interface to an enclave: ImPACT could benefit from tooling to manage access approval for compute jobs launched against restricted data. The ImPACT model lays a foundation for rich policies that grant access based on approved infrastructure and software stacks with required properties, but we leave it to future work to demonstrate and evaluate that capability.

CRediT authorship contribution statement

Ilya Baldin: Conceptualization, Writing – original draft, Project administration, Funding acquisition. **Jeff Chase:** Conceptualization, Writing – original draft, Funding acquisition. **Jonathan Crabtree:** Conceptualization, Writing – original draft, Funding acquisition. **Thomas Nechyba:** Writing – original draft, Project administration, Funding acquisition. **Laura Christopherson:** Investigation, Writing – original draft. **Michael Stealey:** Software, Validation. **Charley Kneifel:** Conceptualization, Writing – original draft, Funding acquisition. **Victor Orlikowski:** Software, Validation. **Rob Carter:** Software. **Erik Scott:** Investigation, Software. **Akio Sone:** Software. **Don Sizemore:** Software.

Declaration of competing interest

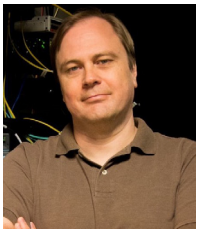
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This material is based upon work supported by the National Science Foundation, USA under Grant No. OAC-1659367.

References

[1] InCommon federation, 2021, https://incommon.org/.
[2] J. Basney, H. Flanagan, T. Fleury, J. Gaynor, S. Koranda, B. Oshrin, CILogon: Enabling federated identity and access management for scientific collaborations, in: Proceedings of Science, Vol. 351, 2019, p. 031. Appeared in International Symposium on Grids and Clouds (ISGC).
[3] M. Crosas, The dataverse network®: an open-source application for sharing, discovering and preserving data, D-Lib Mag. 17 (1) (2011) 2.
[4] ICPSR portal, 2019, https://www.icpsr.umich.edu/icpsrweb/.
[5] ODISSEI (Open Data Infrastructure for Social Science and Economic Innovations), https://odissei-data.nl/.
[6] J.S. Chase, I. Baldin, Federated authorization for managed data sharing: Experiences from the ImPACT project, in: 2021 International Conference on Computer Communications and Networks (ICCCN), IEEE, 2021, pp. 1–10.
[7] TRSA GitHub repository, 2019, https://github.com/OdumInstitute/trsa-web.
[8] Notary Service GitHub repository, 2019, https://github.com/RENCI-NRIG/notary-service.
[9] CI Logon website, 2019, https://www.cilogon.org/.
[10] Presidio GitHub repository, 2019, https://github.com/RENCI-NRIG/impact-presidio.
[11] Q. Cao, V. Thummala, J.S. Chase, Y. Yao, B. Xie, Certificate linking and caching for logical trust, 2017, arXiv:1701.06562.
[12] ProConsul GitHub repository, 2019, https://github.com/carte018/Proconsul.
[13] Jenkins website, 2019, https://jenkins.io/.
[14] Gitlab website, 2019, https://about.gitlab.com/.
[15] G.M. Kurtzer, V. Sochat, M.W. Bauer, Singularity: Scientific containers for mobility of compute, PLoS One 12 (5) (2017) e0177459.
[16] S. Grabus, J. Greenberg, Toward a metadata framework for sharing sensitive and closed data: an analysis of data sharing agreement attributes, in: Research Conference on Metadata and Semantics Research, Springer, 2017, pp. 300–311.
[17] UMich ISR, Data use agreement portal, 2020, https://www.psc.isr.umich.edu/dis/data/restricted/rdc/.
[18] K.M. Harris, The Add Health Study: Design and Accomplishments, Carolina Population Center, University of North Carolina at Chapel Hill, Chapel Hill, 2013, pp. 1–22.
[19] Carolina population center data portal, 2021, https://data.cpc.unc.edu/projects/2/view.
[20] cURL - command line tool and library for transferring data with URLs, 2021, https://curl.se.
[21] Gunicorn - Gunicorn 'Green Unicorn' is a Python WSGI HTTP server for UNIX, 2020, https://gunicorn.org/.
[22] S. Ceri, G. Gottlob, L. Tanca, What you always wanted to know about datalog (and never dared to ask), IEEE Trans. Knowl. Data Eng. 1 (1) (1989) 146–166.
[23] J. DeTreville, Binder, a logic-based security language, in: IEEE Symposium on Security and Privacy, IEEE, 2002, pp. 105–113.
[24] Digital object identifier system, 2021, https://www.doi.org.
[25] Docker website, 2019, https://docker.com.
[26] I. Baldin, M. Stealey, V. Orlikowski, ImPACT Notary Service and Presidio Performance Measurements, Zenodo, 2021, https://zenodo.org/record/4420282.
[27] Django - a high-level Python Web framework that encourages rapid development and clean, pragmatic design, 2020, https://www.djangoproject.com/.
[28] Nginx, 2020, https://www.nginx.com/.
[29] ICPSR: About the organization, 2020, https://www.icpsr.umich.edu/web/pages/about/. Accessed: 2020-12-18.
[30] P. Dykstra, Introducing ODISSEI, 2019, https://odissei-data.nl/en/2019/11/report-community-conference-2019/. Presentation given at the ODISSEI Community Conference.
[31] SURF - collaborative organisation for ICT in Dutch education and research, 2020, https://www.surf.nl/.
[32] L. Sweeney, M. Crosas, M. Bar-Sinai, Sharing sensitive data with confidence: The datatags system, Technol. Sci. (2015).
[33] M.C. Levenstein, A.R.B. Tyler, J. Davidson Bleckman, The Researcher Passport: Improving Data Access and Confidentiality Protection, Technical Report, 2018.
[34] F.B. Schneider, K. Walsh, E.G. Sirer, Nexus authorization logic (NAL): Design rationale and applications, ACM Trans. Inf. Syst. Secur. 14 (1) (2011) 8:1–8:28.
[35] T. Jim, SD3: A trust management system with certified evaluation, in: IEEE Symposium on Security and Privacy, IEEE, 2001, pp. 106–115.
[36] M. Abadi, B.T. Loo, Towards a declarative language and system for secure networking, in: Proceedings of the 3rd USENIX International Workshop on Networking Meets Databases, in: NETDB'07, USENIX Association, Berkeley, CA, USA, 2007, pp. 2:1–2:6.
[37] R. Pang, R. Caceres, M. Burrows, Z. Chen, P. Dave, N. Germer, A. Golynski, K. Graney, N. Kang, L. Kissner, J.L. Korn, A. Parmar, C.D. Richards, M. Wang, Zanzibar: Google's consistent, global authorization system, in: 2019 USENIX Annual Technical Conference (USENIX ATC 19), USENIX Association, Renton, WA, 2019, pp. 33–46.



Ilya Baldin leads RENCI's network research and infrastructure programs. He is a networking researcher with a wide range of interests, including high-speed optical network architectures, cross-layer interactions, novel signaling schemes, and network security. Before coming to RENCI, Baldin was the principal scientist at the Center for Advanced Network Research at the Research Triangle Institute and a network research engineer at the Advanced Network Research group at MCNC, where he was a team member and a leader of a number of federally funded research efforts. He holds Ph.D. and MS degrees in computer science from North Carolina State.

Jeff Chase is a professor and the Interim Director of Graduate Students in the Department of Computer Science Department at Duke University. His research and teaching focus on utility computing, network storage and I/O, distributed systems, operating systems, and large scale network services. He earned his Ph.D. from the University of Washington, Seattle.

Jonathan Crabtree holds a Ph.D. in Information and Library Science from the School of Information & Library Science at UNC Chapel Hill. He is the Director for Research Data Information Systems at the H.W. Odum Institute for Research in Social Science at UNC Chapel Hill, and helps lead the Global Dataverse Community Consortium. His research interests and collaborative activities focus on trusted data repositories and digital preservation, and he brings this expertise to the ImPACT team by overseeing aspects of the work that involve use of Dataverse and ensuring trustworthy access to research data listed in Dataverse.

Thomas Nechyba received his Ph.D. in economics from the University of Rochester in 1994 and has been on the faculty at Duke since 1999 after five years on the faculty at Stanford. He has served as Chair of Duke's Economics Department (2002-09) and more recently as Director of Duke's Social Science Research Institute (SSRI) (2012–19) where he launched a number of collaborations around multi-disciplinary data science, protected data networks and inter-institutional and community partnerships. His research has focused on the economics of primary and secondary education, with particular focus on the intersection of education, housing and local political markets.

Laura Christopherson wears many hats at the Renaissance Computing Institute at UNC—from performing linguistic analysis to improve precision and recall for a literature-based discovery system, to working with scientists to understand and translate needs into system requirements, to collaborating on workforce-development issues for cyberinfrastructure professionals, to mapping the data/research lifecycle in the earth sciences, to designing a publication tracking system for communicating the impact of RENCI projects. She holds a Ph.D. in information science from the University of North Carolina at Chapel Hill.

Michael Stealey is a Distributed Systems Software Engineer at the Renaissance Computing Institute at the University of North Carolina at Chapel Hill.

Charley Kneifel holds a BS from Carnegie Mellon University and a Ph.D. from the State University of New York at Stony Brook, both in chemistry. He is the Senior Technical Director in the Office of Information Technology at Duke University. Dr. Kneifel manages Duke's central technology infrastructure and Software Defined Networking Project. He has coordinated several technology grants at Duke including the National Science Foundation's Data Infrastructure Building Blocks (DIBBS) grant. Prior to working at Duke, Dr. Kneifel was Chief Information Officer at the American Kennel Club for nine years and he held multiple technical positions at NC State University.

Victor Orlikowski is a research software developer and systems administrator for Duke Research Computing. His position is focused on automation—the development of resources to enhance the flexibility of research computing tools so they can best fit researchers' computing requirements. Victor is experienced in networked storage devices and virtual machines. He has been a key member of research teams that have pioneered the tools regularly used by researchers at Duke. Victor has worked at Duke for more than eight years in the Departments of Computer Science, Pratt School of Engineering and OIT. Victor regularly teaches Duke's Introduction to Linux seminars.

Rob Carter holds a BS in electrical engineering and computer science from Duke University. In the 33 years since receiving his degree, he has worked in a variety of capacities, both technical and administrative, within Duke's central IT organization. He is currently a Middleware and Identity Architect for Duke's Office of Information Technology. When he is not working, he spends most of his time walking his two dogs and catering to the five cats he and his wife are employed to serve.

Erik Scott is a computer scientist at the Renaissance Computing Institute at University of North Carolina at Chapel Hill where he builds and integrates software systems for researchers in a range of fields from the social sciences to medicine.

Akio Sone is an Applications Analyst on the Research Data Information Systems team at H.W. Odum Institute for Research in Social Science at the University of North Carolina at Chapel Hill. Before he joined to work on Dataverse-related projects at Odum, he had been a member of the Dataverse developer team at the Institute of Quantitative Social Science at Harvard University.

Don Sizemore is a Systems Specialist for the H.W. Odum Institute for Research in Social Science at the University of North Carolina at Chapel Hill (UNC). He holds an MS in information science from the UNC School of Information and Library Science and a BA in visual communication from the UNC School of Media and Journalism. Don has served the teaching and research missions of UNC for 25 years in capacities ranging from undergraduate computer lab attendant to sole administrator of ibiblio.org. When not at work, he plays pipe organ, piano, and is something of a gardener.