
# ACCSE 2024

The Ninth International Conference on Advances in Computation, Communications and Services

ISBN: 978-1-68558-144-2

April 14 - 18, 2024

Venice, Italy

**ACCSE 2024 Editors**

Pascal Lorenz, University of Haute-Alsace, Colmar, France

# ACCSE 2024

# Forward

The Ninth International Conference on Advances in Computation, Communications and Services (ACCSE 2024), held between April 14th and April 18th, 2024, continued a series of international events targeting the progress made in computation, communication, and services on various areas in terms of theory, practices, novelty, and impact. Current achievements, potential drawbacks, and possible solutions are aspects intended to bring together academia and industry players.

The rapid increase in computation power and affordable memory/storage led to advances in almost all the technology and services domains. The outcome made it possible advances in other emerging areas, like Internet of Things, Cloud Computing, Data Analytics, Smart Cities, Mobility and Cyber-Systems, to enumerate just a few of them.

We take here the opportunity to warmly thank all the members of the ACCSE 2024 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to ACCSE 2024. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the ACCSE 2024 organizing committee for their help in handling the logistics of this event.

We hope that ACCSE 2024 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of computation, communications, and services.

**ACCSE 2024 Chairs**

**ACCSE 2024 Publicity Chairs**
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

# ACCSE 2024
# Committee

## ACCSE 2024 Publicity Chairs

Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

## ACCSE 2024 Technical Program Committee

Kishwar Ahmed, University of South Carolina Beaufort, USA
Muhamad Erza Aminanto, University of Indonesia, Indonesia / NICT, Japan
Chloe Aronoff, University of Michigan, USA
Maxim Bakaev, Novosibirsk State Technical University, Russia
Abdul Basit, State Bank of Pakistan (Central Bank of Pakistan), Pakistan
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Ali Behfarnia, University of Tennessee at Martin, USA
Surendra Bhosale, Veermata Jijabai Technological Institute, India
Freimut Bodendorf, Institute of Information Systems - University of Erlangen-Nuremberg, Germany
An Braeken, Vrije Universiteit Brussel, Belgium
Erik Buchmann, Leipzig University, Germany
Hung Cao, University of New Brunswick, Canada
Seongah Chin, Sungkyul University, Korea
Arun Das, Visa Inc., USA
Erdogan Dogdu, Angelo State University, USA
Mounîm A. El Yacoubi, Telecom SudParis / Institut Polytechnique de Paris, France
Barbara Gili Fivela, University of Salento, Italy
Aviel Glam, Technion - Israel Institue of Technology | RAFAEL - Advanced Defence System Ltd., Israel
Josefa Gómez, University of Alcalá, Spain
Robert C. Green II, Bowling Green State University, USA
António Guilherme Correia, INESC TEC / University of Trás-os-Montes e Alto Douro, Vila Real, Portugal
Béat Hirsbrunner, University of Fribourg, Switzerland
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Mehdi Hosseinzadeh, Washington University in St. Louis, USA
Fu-Hau Hsu, National Central University, Taiwan
Xin Huang, University of Maryland, Baltimore County, USA
Michael Huebner, BTU Cottbus-Senftenberg, Germany
Sergio Ilarri, University of Zaragoza, Spain
Ilias Iliadis, IBM Research - Zurich Laboratory, Switzerland
Kazi Ashik Islam, University of Virginia, USA
Tomayess Issa, Curtin University, Australia
Ajin Joseph, IIT Tirupati, India
Keiichi Kaneko, Tokyo University of Agriculture and Technology, Japan
Yasuko Kawahata, Rikkyo University, Japan
Abbas Khosravi, Deakin University, Australia
Carsten Kleiner, University of Applied Sciences & Arts Hannover, Germany
Yulia Kumar, Kean University, Union , USA

**Copyright Information**

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

**Table of Contents**

# dPIDs - the Emerging Persistent Identification Technology for FAIR and the Digital Era

Andrey Vukolov
*Scientific Computing Group*
*Elettra Sincrotrone Trieste*
Trieste, Italy
0000-0001-6967-3236

Erik van Winkle
*GOFAIR Foundation Fellow*
*DeSci Labs AG*
Bäch, Switzerland
0000-0002-7567-0311

Elizaveta Zhdanova
*Faculty of Fine Arts*
*Valencia Polytechnic University*
Valencia, Spain
0009-0005-3701-0266

Georgios Kourousias
*Scientific Computing Group*
*Elettra Sincrotrone Trieste*
Trieste, Italy
0000-0002-1243-7168

***Abstract*—This paper presents a comprehensive exploration of an emerging technology focused on the persistent identification and sharing of data and metadata, termed "decentralised Persistent Identifier" (dPID). Utilizing the InterPlanetary File System (IPFS) network, dPID provides reproducible persistent identifiers, ensuring that data can be reliably stored and accessed over time. It is designed to incorporate a distributed ledger, Public Key Infrastructure (PKI), decentralized linking, and lookup databases. It also aims to support version control and provenance tracking based on reproducibility. The distributed approach highlights the potential of dPID to align with the principles and guidelines of Findable, Accessible, Interoperable, and Reusable (FAIR) data, positioning it as a valuable component within the open data ecosystem, mitigating in an efficient way the problems, such as link rot and content drift. The proposed technology provides a highly scalable persistent identification and versioning system for shared data that reduces dependencies from social contracts and institution-driven systems. The paper also demonstrates its usability for modern social-oriented identification systems, proposing a use case study for the art industry.**

***Index Terms*—PID, Persistent Identifier, IPFS, Decentralized, dPID, FAIR, Blockchain, Provenance Tracking, Reproducibility**

## I. Introduction

*Persistent Identifiers (PIDs) are not only for documents and data. They can also be used to reference other entities or agents, including people contributing to the research, software, research organizations, physical objects such as samples and instruments, and even abstract concepts such as terms in a controlled vocabulary. As a general rule, whenever something needs to be referenced in a reliable and lasting manner, a persistent identifier should be used* [1].

The modern world is encountering unprecedented challenges in data handling. Growing amounts of data in the realm of academia, including Internet of Things (IoT) sensing, multi-disciplinary works and Artificial Intelligence (AI) form a collective ocean of knowledge, impassible without the collaboration of humans and machines. One point of advocacy for the necessary human-computer symbiosis can be found in the Findable, Accessible, Interoperable, and Reusable (FAIR) principles and subsequent literature around machine-actionability [2], [3]. Policy based recognition in support of these principles has been released over the past 5 years from organizations such as Office of Scientific and Technological Policies (OSTP), European Commission (EC), and National Institutes of Health (NIH) [4]–[6].

The first of the 15 FAIR Principles, Principle F1, states that "Data and metadata must have globally unique, persistent, and resolvable identifiers" [2]. In the upcoming world of FAIR Science, the scientific record strives for the permanence of information online. While permanence is impossible, PIDs and underlying technological infrastructures are essential components of continuous navigation and data retrieval. Existing PID providers such as Digital Object Identifier (DOI), mitigate the problems of persistence using social and technical mechanisms to guarantee consistent mapping and resolution of a given ID to its target resource. This goal of persistent identification and resolution has a standard set of problems.

Ensuring the scalability of social and technical mechanisms in persistent identification technology can be a challenge. Industry 4.0, IoT and rapidly growing open-source development [7] drastically increase the amount of data that should be identified and linked in a strictly deterministic manner. The reusability of metadata is required for the efficiency of data access, meaning that the persistence of metadata and its subsequent linkage to data becomes essential too. Social trust as a mechanism to combat the replication crisis requires added persistence requirements. The high diversity of data storage and transportation techniques, protocols, networks, etc. also reveals new data treatment circumstances where every file should be identified, checked for integrity, versioned, and shared over many possible networks with different address resolution mechanisms. The average scientific project or experiment outcome may contain hundreds of thousands of files (research artifacts), including code, data, reviews, sensemaking statements, etc., both created by the people and captured automatically. Global endeavors in a single scientific domain can produce trillions of individual files of their own accord. The pursuit of permanence for scientific knowledge starts with a problem of scaling persistent identification infrastructure.

Additional challenges surrounding current PID infrastructure include but are not limited to the sovereignty and access control over data [8], the necessity of machine-actionable provenance, client-side reproducibility of existing PIDs due to their generation algorithms being enclosed within the registrar's infrastructure and an exponentially growing amount of

 ISBN: 978-1-68558-144-2

resolution requests due to the formation of PID graphs [9].

With trillions of PIDs likely needed by 2030, the probability of system failure in current federated but centralized architectures lead to system fragility and a potential loss of valuable knowledge [10].

To stay relevant and reliable under the extreme circumstances of science in the digital age, the PID system's intended usage in the modern FAIR-compliant data-driven Internet needs to implement a reproducible, fully automated, scalable, globally unique identifier system, with minimal reliance on social contracts, leading to fewer centralized points of failure and human interactions. In the following sections, we will explore the range of currently available PID systems, delving into their underlying technologies and unique characteristics. Subsequently, we will present a technical proposal for an emerging decentralized PID system, including a concise analysis and discussion on the prospects of its implementation.

Additionally, this paper proposes a potential application of the dPID technology in the realm of fine arts and photography. This use case focuses on hybrid distribution models that encompass both hardcopy and digital formats, addressing the unique challenges of managing, distributing, and verifying artworks across diverse mediums. By leveraging dPID, stakeholders in the fine arts sector could benefit from enhanced provenance tracking, more secure distribution channels, and improved accessibility to art, all while adhering to the FAIR data principles.

The rest of the paper is structured as follows. In Section 2, we explain the current status of the PID implementation landscape, holding European Open Science Cloud Photon and Neutron Data Services (ExPaNDS) project outcomes as a basis. In the first part of Section 3, we give a technical proposal based on one of the possible implementations of a decentralised approach to persistent identification. In the second part of this section, we leverage the technology by giving a technical outline of freshly developed open-source software. Then in the Conclusion section, we summarize the problems we have indicated solved in the paper. This section is also extended with a use case proposal showing the possibilities of dPID in the area of persistent identification and distribution of the artworks.

## II. Current PID Implementations: Landscape and Centralization Status

This section is partially based on works [1] and [11]. It explores the implementation details of the most known and popular PID systems in the context of centralization. The mentioned PID systems: DOI, Open Researcher and Contributor Identifier (ORCID), Handle, Research Organization Registry (ROR), etc., implemented worldwide fall between *centralized* and *federated* architectures [12] implementing the centralised resolution model presented in Figure 1. The decentralised architectures are now only emerging, so none of the existing systems could be named decentralised.

The model in Figure 1 implements a low number of high-risk singular points of failure, (as it is based on the entity called **Global PID Registry**), despite the existence of both primary and secondary authorities. From the client's point of view, they should be called **centralized**. Every existing point of failure here may lead to the failure of the entire facility or domain. If the PID is not reproducible and the generation schema is closed, the centralised PID governance authorities are the only provenance holders of the underlying record, metadata, and addressed data [13]. This self-assignment trap magnifies when record tracking lags behind the growing number of maintained records. Issues such as link rot, content drift, artifact fragmentation and inconsistent resolution [14] in these systems can be traced back to human interactions, leading to lacking persistence because they do not provide lookup table redundancy [15], [16], reproducibility, and proper caching. However, while the extreme circumstances of the modern data-driven world do not affect all domains, centralized authorities such as person-oriented ORCID and organization-oriented ROR have proven trustworthy over time, at scales of 1-10M of PIDs. For their social-driven data flows and scalability cases, they are simple, efficient, and can be considered fit-to-purpose.

As a reply towards the challenges arising, the different flavours of **federated** approach have appeared. It combines the existing conservative model of centralized governance authority with a federated social-driven network of formally independent PID registrars. Each registrar manages the underlying system of centralized prefix-based lookup tables with a federated network of resolvers based on Hypertext Transfer Protocol (HTTP) redirection. They act from the governance point of view, as independent registries with independent databases replicated at the discretion of the centralized authority. Each registrar maintains its own provenance authority and ensures the persistence of associated PIDs alone. To control the resources distributed across the federation of infrastructure providers in maintaining persistent registry infrastructure, an external council or legal entity is still needed, leading to a non-profit governance ogranization overseeing the system. Through this structure, the federated PID infrastructure aims to preserve the administrative efficiency of the centralized approach while reaching redundancy and distributed curation of decentralized approaches. The most popular PID systems in the world, DOI and Handle.net, use the federated approach. It allows expansion of the computational abilities of the system, primarily in the aspect of data replication.

To obtain interoperability and cross-resolution, especially in federated architectures, the third-party aggregation approach, in practice, is the only way. The UNIRESOLVER initiative [17] is the most known solution for implementing it. It utilizes Worldwide Web Consortium (W3C) Decentralized Identifiers (DID) specification [18]–[20] to perform a bidirectional lookup but without internal implementation of metadata versioning and reproducibility tracking, so the records resolved via UNIRESOLVER could not be considered truly immutable and the associated resolution cannot be considered deterministic on the periods on which the PIDs should persist. However, UNIRESOLVER demonstrates the possibilities of cross-resolution and request flow balancing in the federated

 ISBN: 978-1-68558-144-2

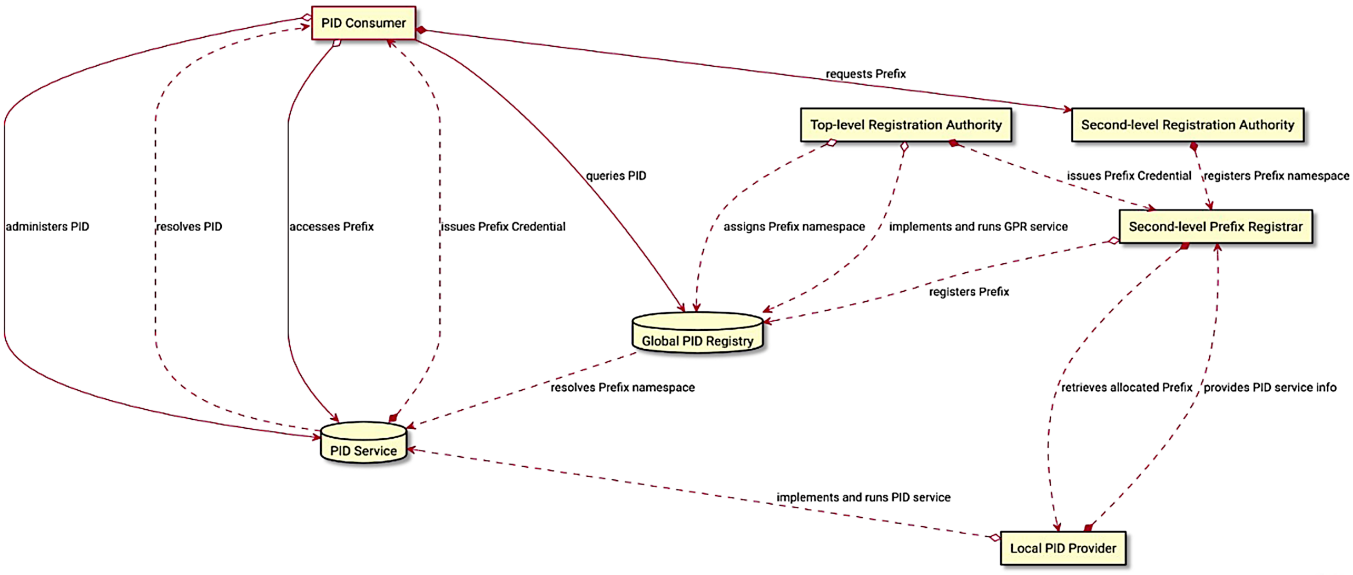


Fig. 1. Traditional PID administration and implementation schema.

architecture without rebuilding the existing provenance chains.

In Table I, a brief list of currently implemented worldwide PID providers is presented, with their underlying technologies and/or PID type specifications and dedicated entity types. The table is taken from [1] with minor additions. In Table I:

- DAG — Directed Acyclic Graph.
- IGSN — International Generalized Sample Number.
- SWHID — Software Heritage Identifier.
- RAiD — Research Activity Identifier.

As it is easy to see from the entries presented in Table I, only one of the PID providers implemented worldwide uses a PID with client-side reproducibility — SWHID. Thus, it should be considered the only one that has already implemented the provable immutability and inline versioning, but under the centralized governance. The federated DOI system holds leadership in dissemination; it is used as the underlying provider by most listed PID providers.

## III. dPID: Adding Decentralization to Data Identification Pipelines

The core idea behind decentralized data identification technologies is that the client should be asking a fundamentally different question when resolving the metadata from the PID. Instead of asking a single point: ”What is the content stored at this location?”, they should be asking a network-based decentralized swarm: “Can you tell me how to find the content with this hash?”. This approach is one of the natural continuations of an idea of the PID Graph [9], proposing the technical mechanisms as the primary mode of guaranteeing persistence, replacing the social mechanisms used in current systems. In a decentralized PID system, **the social persistence of a PID is as strong as the technical prevalence of the network nodes, providing self-describing addressing, resolution, and consistent data**. One of the possible workflow schemas implementing the decentralized approach is presented in Figure 2.

As it could be easily deducted from the block names in Figure 2, most of the elements of the PID system indicated are the distributed ledgers and databases where the data are not owned or stored on the side of the single participant or institution: the lingering in this schema requires a mathematically proven integrity of the internal metadata stored in the system. Also, the presented schema should not be considered as a reference: the decentralised approach makes the implementation schema fuzzy and yields its structure in favour of the workflow.

The ideal implementation of the PID system necessitates uniformity in backend technology and storage models across all resolvers and PID generators. In such a setup, users would need to install a specific software, possibly on a local machine, and register a public key. This process would grant them access to a resolution endpoint, equip them with a PID generator that supports namespace propagation right from the start, and provide a viewer for the provenance chain. It demands data pipeline technologies that can handle storage, identification, and delivery in a manner that is as type-agnostic and mathematically secure as possible. Fortunately, there exists a category of technologies that fulfil these requirements. Broadly, these can be characterized as decentralized redundant storage systems, with BitTorrent being one of the most well-known examples. Such technologies rely on hash-based, mathematically-driven content addressing, and utilize Distributed Hash Tables (DHT) for the delivery of binary data objects. DHT, in particular, incorporates identification directly within its addressing stack, making these technologies an apparent choice for the described purposes [18].

IPFS, the decentralized storage network [21] stands out by allowing data to be stored under a single identifier that is immutable and persistent, based on its binary representation.

 ISBN: 978-1-68558-144-2

TABLE I
INTERNATIONAL PID PROVIDERS, THEIR UNDERLYING TECHNOLOGIES AND ENTITY TYPES.

| Provider | Technology | Entities | Centralization |
|---|---|---|---|
| DataCite | DOI | General purpose | Federated |
| Crossref | DOI | Publications, funders | Federated |
| ePIC | Handle | Metadata in all plaintext schemas | Federated |
| IGSN | Handle | Experimental samples | Federated |
| ORCID | Bespoke (custom) | Persons | Centralized |
| FigShare PID | DOI | Research artifacts | Federated |
| Zenodo | DOI | Publications, digital research artifacts | Federated |
| EUDAT B2SHARE | Handle, DOI | Datasets, digital research artifacts | Federated |
| FAIRshare | DOI | Datasets, policies, standards | Federated |
| SWHID | SHA1-based Merkle DAG | Software artifacts, versioned source code | Federated, decentralized |
| ROR | Bespoke (custom) | Research institutions | Centralized |
| RAiD | Handle (custom, prefixed) | Funders, organizations, persons, instruments, datasets | Centralized |

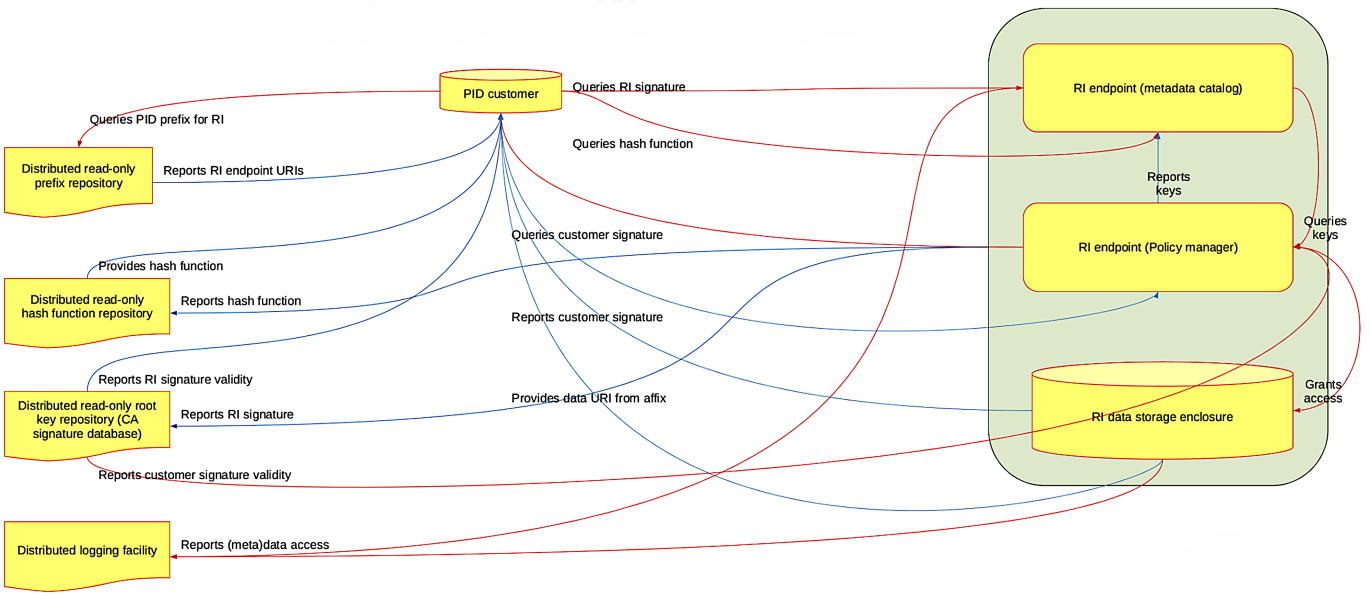


Fig. 2. One of the possible decentralised PID workflow implementation schemas.

This system employs Content Identifiers (CIDs), which are derived from the outputs of unidirectional cryptographic hash functions [22]. These IDs incorporate a prefix-based structure, with the specification detailing how they should be resolved, ensuring a comprehensive description within the ID's structure itself. CIDs can be seen as production-ready PIDs that are resistant to content drift and link rot.

The IPFS storage model is designed to support a cache-on-read retrieval approach, where data, along with its metadata, is recorded into DAGs using various storage tree builders. This ensures that data and metadata are automatically cached, with the option for explicit declaration of caching for specific data on a given network node. The persistence levels for data stored in IPFS, as determined by the CID's state declared in the DAG, are outlined in Table II. In this context, a "pinned" state indicates that a specific CID has been explicitly marked for caching on at least one node within the IPFS network. The term "Accessible" means that the requested data is available for download, and "Findable" means that the data will become available and propagating through the DAG at the moment when at least one node knowing the given CID has appeared in the network.

TABLE II
POSSIBLE AVAILABILITY STATES OF THE DATA ENTRY IN THE IPFS NETWORK.

| IPFS DAG / *Local data* | Pinned | Wanted | Available | Discarded |
|---|---|---|---|---|
| *Stored* | Persistent | Accessible | Persistent | Accessible |
| *Requested* | Persistent | Accessible | Persistent | Findable |
| *Idle* | Persistent | Accessible | Accessible | Accessible |
| *Discarded* | Findable | Findable | Findable | Unavailable |

IPFS also incorporates the InterPlanetary Linked Data (IPLD) concept, utilizing a hash-based addressing model that facilitates the construction of decentralized storage trees [23]. This model enables a persistent linkage between stored items

 ISBN: 978-1-68558-144-2

and their versions, ensuring that each piece of data can be uniquely identified and accessed over time. Given the features that have already been implemented, IPFS emerges as a suitable choice for implementing the PID system, thanks to its robust framework for storing and linking data in a decentralized manner.

### A. dPID: Technical Proposal

To establish a production-grade PID system, integrating IPFS with additional components that facilitate persistent linkage, version control, and contribution tracking is essential [24]. This paper introduces dPID, a decentralized data identification and sharing system designed to enhance the management and curation of research artifacts and FAIR Data Objects. dPID represents a pilot project aimed at offering a reliable solution for persistent storage and data curation [25].

At its core, dPID leverages a sophisticated integration of the IPFS and IPLD technologies alongside the Sidetree Protocol [26]. Sidetree acts as a protocol and Application Programming Interface (API) layer that can operate atop any data addressing system, enabling users to generate lookup databases and customize identities secured by PKI. Utilizing Sidetree, dPID adopts the standardized DID schema standardized by W3C, which facilitates the creation of universally unique identifiers that are resilient and verifiable [20].

dPID provides access to data that is both machine-actionable and human-readable, featuring a web interface built upon the JavaScript Object Notation Linked Data (JSON-LD) specification [27] and supported by an open-source API. The software suite responsible for resolving and minting dPIDs serves as a uniform kernel for every installation, ensuring that the system is completely open-source and operable on dedicated systems as-is. This setup not only enables efficient minting and resolution processes but also guarantees a public resolver functionally equivalent to any resolver within the decentralized network.

Enhancing the foundational persistence offered by IPFS, dPID introduces features like high throughput, strong consistency across the network, decentralized indexing, user-friendly URLs, and the incorporation of a Turing-complete blockchain. This blockchain component autonomously records the root CID of an IPLD data structure, ensuring metadata redundancy and immutability. Consequently, dPID promises reliable persistence for stored FAIR data objects and research artifacts, making it a comprehensive solution for decentralized data management and sharing. dPIDs provide:

- Verifiable ownership with ORCID-based person identification and incremental contribution record.
- Open network participation and metadata redundancy through peer-to-peer nature of IPFS.
- Compliance with FAIR principles via FAIR Data Object specification compatibility.
- “Vendor lock-in” removed in the context of data due to the removal of the singular provenance holder of the scientific record.
- Data integrity persistence with DHT, as it was described above.

### B. dPID Nodes: Brief Technical Outline

dPID Nodes is an open source [28] software suite written in TypeScript and published under MIT license [29]. It acts over IPFS HTTP API that should be provided by an API server. The API server currently used is Kubo - the open-source reference implementation of the IPFS node that runs in the background provides CID resolution and retrieves the data from the decentralised network. dPID considers the identified data as a collection of versioned IPLD entries following Research Object Crate (RO-CRATE) specification [30]. The simplified example of internal linkage is presented in Figure 3. From the perspective of the end user, these systems can be likened to folders that store research artifacts in a format-agnostic manner, with authorship and provenance details facilitated through integration with ORCID. The storage entries are catalogued in a distributed key-value store, each uniquely accessible via a dedicated persistent identifier that leverages underlying IPFS CIDs for addressing. The links between CIDs in Figure 3 illustrate the internal linked data structure that implements versioning and the history of changes. According to IPFS specification, every CID included in this schema is immutable, so the dPID metadata actually formulates a versioned repository for every FAIR Data Object it addresses.

As outlined in the documentation [25], [31], dPID ensures deterministic resolution of PIDs to the internal CIDs of IPFS and their associated content through a DAG. This process allows the content to be immediately cached on the local database of the node it is accessed from. Building upon the IPFS framework, dPID Nodes additionally employ Ceramic [32], a decentralized event streaming protocol, to create a graph-based distributed lookup database. This integration facilitates advanced data addressing using a combination of CID and DID.

When dPID Nodes installation is queried through an HTTP API, it returns JSON-LD object [27]. These objects can then be resolved by the web frontend, presenting the data in a human-readable format. This mechanism ensures that end users can easily access and interpret the stored research artifacts, benefiting from a seamless integration of decentralized storage technologies and modern web standards for data representation and access. The system in its current state should be considered as a pre-beta pilot version, work in progress on the deployment process and features list.

### C. Challenges on the Security

dPID is designed to be secure. Security is among the top priorities and there is high confidence in the current implementation due to the secure nature of its individual components. Openness and strong cryptography with options for transparent upgradability are already in place considering an openly shared source code. It is beyond the scope of this paper to analyse in depth the strengths and weaknesses of the technological choices in the scope of security. Nevertheless, it hints that they

 ISBN: 978-1-68558-144-2

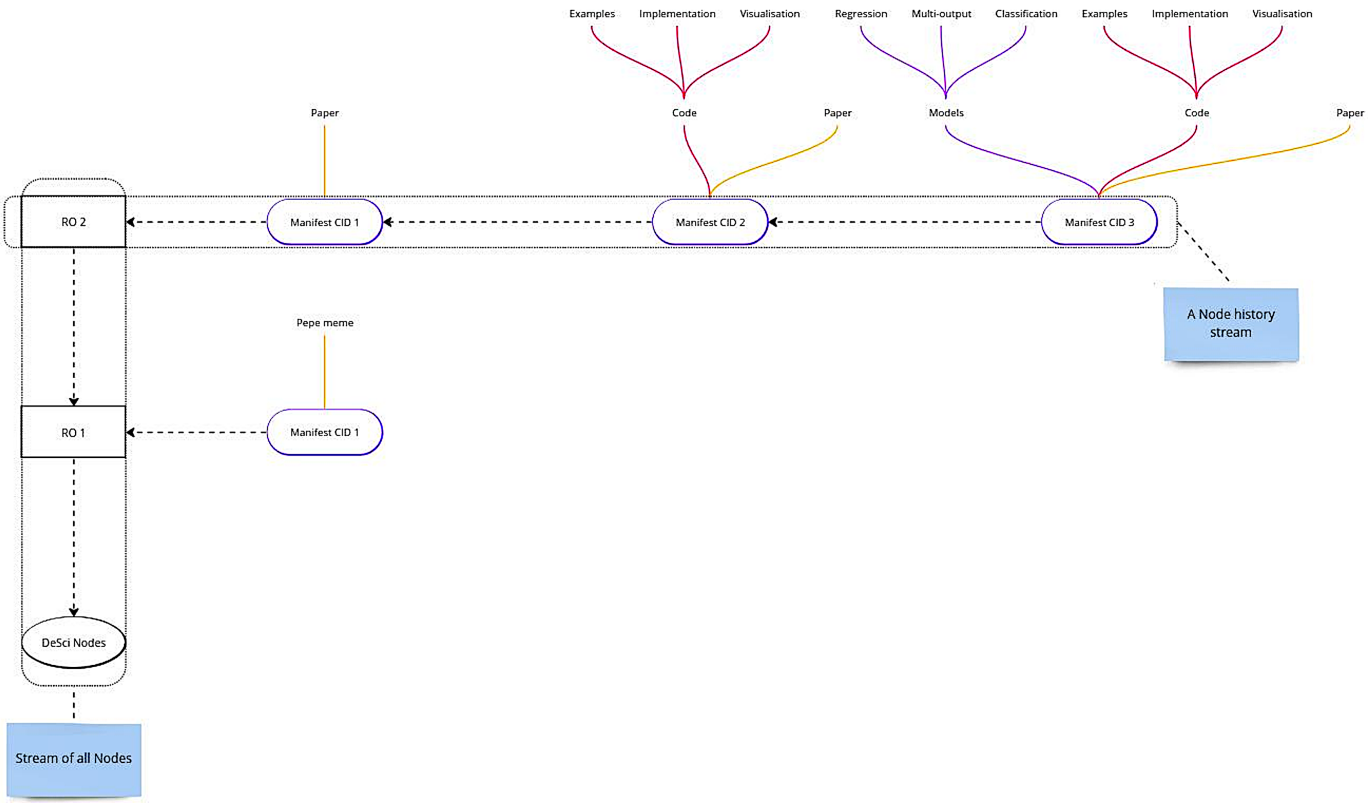


Fig. 3. Simplified internal bitstream storage diagram in dPID.

are taken into consideration. Those choices of technologies also aim at feasible disaster recovery and the robustness of the system. A recent incident demonstrated the readiness of the system and here it is briefly mentioned. Elettra Sincrotrone Trieste, like similar large facilities, had a security incident of unknown nature but with limited and contained damages. An isolated series of ransomware attacks required swift actions and temporary restriction of experimental and non-essential services to the operation of the facility. Such a service was that of the dPID gateway which albeit not compromised, it had to get offline. The underlying technology of dPID allowed for the rapid migration to a new server, outside the facility and hosted in a different country. The migration was rapid and easy (around 2 days including overrun and migration of the underlying CIDs and IPLD tracks), allowing for the continuation of the tests. Without focusing on the individual components that allowed this, the takeaway message is the decentralised nature of dPID which is in contrast to all other established, and especially, the centralised PID solutions.

## IV. Roadmap for Further Development

The dPID initiative for the future strives to:

- Provide a set of convenience libraries with a comprehensive API and viable examples letting software developers adopt the technology.
- Propose a deployment pipeline feasible for different use cases (OS packages, bundles, automated source builders, Docker scripts, etc.).
- Develop comprehensive documentation for end users, publishers and administrators letting them adopt and use the technology in the most configurable and flexible way.
- Propose social-oriented mechanisms such as social attestation, conflict moderation, and access control.
- Formal validation of the security and robustness models of dPID.
- Explore and adopt multiple data-oriented use cases such as migration, storage sharing, and consortium mechanisms.

## V. Conclusion and Use Case Proposal

A description of the dPID system and the problems it solves is provided above. The Nodes web interface is openly published and used to distribute research artifacts published with open licenses. The experimental resolution endpoint lives on the website [33]. For the installation and usage complexity, there are only qualitative estimations that exist now. However, because every participant of the dPID network should share completely the same software and all elements of the system, including the PID resolution point based on the website, the deployment complexity of the system should be estimated as approximately equal to the complexity of deployment

 ISBN: 978-1-68558-144-2 

and debugging of the standard organizational website in the cloud. The deployment pipeline uses highly standardised and documented solutions, such as Docker, so the process can be effectively controlled from the developers' side, and volunteer support through Github is also available.

The research introduces a compelling use case for dPID as a foundational infrastructure for the identification and hybrid (hardcopy + digital) distribution of artworks, incorporating provenance tracking and the engagement of social institutions. This initiative aims to facilitate the management of the provenance chain for photographic artworks, among others, that are distributed in hard copies, as Non-Fungible Tokens (NFTs), and in digital formats. The system is designed to link the author's personal PID with the authorized digital copy of an artwork and the CIDs of copies authorized for distribution, accompanied by provenance chain documents validated by social institutions.

This model is particularly suited for electronic use, incorporating linked data that includes technical details to identify unauthorized digital distribution and support implicitly legal promotion mechanisms, such as search engines. Also, it defines the basic identification procedures and unified initial provenance for the hardcopy distribution of the digital artworks, with an option to extend the practices to the material artworks. Currently, the project is in the stage of defining its workflow model, with the metadata model already established. It is being explored as a potential application for dPID deployment, aiming for social recognition and validation.

Through this approach, dPID could offer a robust solution for artists and institutions to securely manage and distribute artworks. By integrating digital and hardcopy formats with a comprehensive provenance chain, the project seeks to enhance the trust and verifiability of artworks' distribution and ownership, leveraging decentralized technologies for greater transparency and security in the art world.

Due to the integrated attestation mechanisms, for external users, outcomes of the dPID project have approximately the same place as DOIs. Especially for the artworks, dPID extends the usual application area of the authenticity certificates, making them recognisable worldwide using any accessible resolution point held by any dPID network participant. From the author's point of view dPID has an outcome as a versioned, reproducible metadata handler for his authored data (digital photographs, articles, research objects, etc.). Acting locally, it simplifies minting and extends the functionality of systems like DOI, with out-of-the-box immutability.

## ACKNOWLEDGMENT

Authors acknowledge **Linda Simeone** and **Andrey Kurin** for their contribution in the development of the artwork identification use case, especially in the social and legal areas.

## REFERENCES

[1] V. Bunakov, R. Krahl, B. Matthews, N. Vizcaino, and A. Vukolov, "Advanced infrastructure for PIDs in Photon and Neutron RIs," Mar. 2022.

[2] M. D. Wilkinson *et al.*, "The FAIR guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[3] T. Bozada *et al.*, "Sysrev: A FAIR platform for data curation and systematic evidence review," *Frontiers in Artificial Intelligence*, vol. 4, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frai.2021.685298

[4] B. M. Kuehn, "NIH Data Sharing," *JAMA*, vol. 298, no. 20, pp. 2361–2361, 11 2007. [Online]. Available: https://doi.org/10.1001/jama.298.20.2361-a

[5] R. David *et al.*, "Be sustainable: EOSC life recommendations for implementation of FAIR principles in life science data handling," *The EMBO Journal*, vol. 42, no. 23, p. e115008, 2023. [Online]. Available: https://www.embopress.org/doi/abs/10.15252/embj.2023115008

[6] H. Moulaison-Sandy, "The Nelson Memo and US Federal Funder Requirements for Public Access: Implications for Technical Services Librarians," *Technical Services Quarterly*, vol. 40, no. 4, pp. 290–297, 2023. [Online]. Available: https://doi.org/10.1080/07317131.2023.2271278

[7] S. Plaga *et al.*, "Securing future decentralised industrial iot infrastructures: Challenges and free open source solutions," *Future Generation Computer Systems*, vol. 93, pp. 596–608, 2019.

[8] P. N. Mahalle, G. Shinde, and P. M. Shafi, *Rethinking Decentralised Identifiers and Verifiable Credentials for the Internet of Things*. Cham: Springer International Publishing, 2020, pp. 361–374.

[9] H. Cousijn *et al.*, "Connected research: The potential of the pid graph," *Patterns*, vol. 2, no. 1, 2021.

[10] A. Vukolov, "Openly reproducible Persistent Identifiers (PIDs) as a factor of FAIRness in data sharing practices," Jun. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4980522

[11] P. de Castro, U. Herb, L. Rothfritz, and J. Schöpfel, "Some reflections on the current PID landscape – with an emphasis on risks and trust issues," *Procedia Computer Science*, vol. 211, pp. 28–35, 2022, 15th International Conference on Current Research Information Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922016386

[12] J. Brown, "PID federation scoping study: final report," Sep. 2020.

[13] N. Soler *et al.*, "Final recommendations for FAIR Photon and Neutron Data Management," Jul. 2022.

[14] M. Klein and L. Balakireva, "On the persistence of persistent identifiers of the scholarly web," in *Digital Libraries for Open Knowledge*, M. Hall, T. Merčun, T. Risse, and F. Duchateau, Eds. Cham: Springer International Publishing, 2020, pp. 102–115.

[15] J. Klump *et al.*, "Towards globally unique identification of physical samples: Governance and technical implementation of the IGSN global sample number," *Data Science Journal*, vol. 20, no. 1, pp. 1–16, 2021.

[16] J. Kunze, "Towards electronic persistence using ARK identifiers," UC Office of the President, 2003. [Online]. Available: https://escholarship.org/uc/item/3bg2w3vs

[17] UNIRESOLVER, "Code Repository," Github, access date: 23.01.2024. [Online]. Available: https://github.com/decentralized-identity/universal-resolver

[18] M.-A. Sicilia, E. García-Barriocanal, S. Sánchez-Alonso, and J.-J. Cuadrado, "Decentralized persistent identifiers: a basic model for immutable handlers," *Procedia Computer Science*, vol. 146, pp. 123–130, 2019, 14th International Conference on Current Research Information Systems, CRIS2018, FAIRness of Research Information.

[19] DIDs, "Decentralized Identifiers v1.0. Core architecture, data model, and representations. W3C Recommendation 19 July 2022," Website, access date: 22.01.2024. [Online]. Available: https://w3c.github.io/did-core/

[20] N. Bach, "Dezentrale identifikatoren (dids): Die nächste pid-evolution: selbstsouverän, datenschutzfreundlich, dezentral," *o-bib. Das offene Bibliotheksjournal/Herausgeber VDB*, vol. 8, no. 4, pp. 1–20, 2021. [Online]. Available: https://doi.org/10.5282/o-bib/5755

[21] IPFS, "Official Documentation," Website, access date: 21.01.2024. [Online]. Available: https://docs.ipfs.tech/

[22] CID, "Specification," Website, access date: 21.01.2024. [Online]. Available: https://github.com/multiformats/cid

[23] IPLD, "Official Documentation," Website, access date: 23.01.2024. [Online]. Available: https://ipld.io/docs/

[24] A. Niehues *et al.*, "A multi-omics data analysis workflow packaged as a FAIR Digital Object," *GigaScience*, vol. 13, p. giad115, 01 2024. [Online]. Available: https://doi.org/10.1093/gigascience/giad115

 ISBN: 978-1-68558-144-2 

[25] dPID, “Codex,” Website, access date: 23.01.2024. [Online]. Available: https://codex.desci.com/desci-codex/design-goals
[26] Sidetree, “Protocol documentation,” Website, access date: 22.01.2024. [Online]. Available: https://identity.foundation/sidetree/spec/
[27] JSON-LD, “Documentation,” Website, access date: 21.01.2024. [Online]. Available: https://json-ld.org/learn.html
[28] dPID, “Nodes,” Code repository, access date: 23.01.2024. [Online]. Available: https://github.com/desci-labs/nodes
[29] “MIT License,” Website, access date: 23.01.2024. [Online]. Available: https://opensource.org/license/mit/
[30] RO-CRATE, “Research Object Crate,” Official Documentation, access date: 21.01.2024. [Online]. Available: https://w3id.org/ro/crate
[31] dPID, “Documentation,” Website, access date: 22.01.2024. [Online]. Available: https://docs.desci.com/
[32] Ceramic, “Protocol Documentation,” Website, access date: 23.01.2024. [Online]. Available: https://developers.ceramic.network/docs/protocol/js-ceramic/overview
[33] “dPID Nodes Web Interface,” Website, access date: 23.01.2024. [Online]. Available: https://nodes.desci.com

 ISBN: 978-1-68558-144-2 

# Mining Erasable Itemsets with Multiple Thresholds under the Loose Constraint

Tzung-Pei Hong[1,2], Yi-Chen Chang[2], Chun-Ho Wang[2], and Wei-Ming Huang[3]
[1]Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan
[2]Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan
[3]Department of Electrical and Control, China Steel, Inc., Kaohsiung, 806, Taiwan
Email: tphong@nuk.edu.tw, 4a0g0902@stust.edu.tw, dodo166577@gmail.com, granthill168@gmail.com

***Abstract*—In data mining, erasable-itemset mining is a popular research field widely applied in factory production management. Traditional erasable-mining algorithms typically use a single threshold as the criterion for mining erasable itemsets. It implicitly assumes that all items are of equal importance, neglecting the fact that each item has a distinct value in an application. In this paper, we use the concept of multiple thresholds and employ the loose constraint to calculate the threshold of an itemset. Since the downward-closure property is not applicable to the loose constraint, we thus utilize the sorted closure to narrow the search space and improve search efficiency. Through experiments, we compare the performance of the erasable-itemset mining using single and multiple thresholds.**

***Keywords-data mining; erasable-itemset mining; multiple thresholds, loose constraint.***

## I. INTRODUCTION

In the digital generation, we are confronted with vast amounts of data containing useful information and potential value. These massive datasets hide important patterns, trends, and knowledge. It is a challenging task to extract these treasures efficiently from the data. Data mining, an interdisciplinary technology, has thus been proposed to reveal latent information within data and help us catch the implicit meaning embedded in the data.

Data mining is a technology that combines statistics, mathematics, databases, and artificial intelligence. Its goal is to discover patterns, trends, and knowledge within big data. By applying various data analysis techniques, data mining can uncover previously unknown patterns, providing robust support for decision-making, prediction, and optimization. The applications of data mining are extensive, spanning areas such as business, healthcare, finance, social sciences, environmental sciences, and so on. In the business domain, data mining is widely used in market analysis [2][4][5], customer relationship management, risk management, and other aspects to enhance the competitiveness of a company.

Erasable itemset mining [8] is one of the essential mining problems with extensive applications. It deals with the problem that raw materials for making products cannot be entirely purchased in a factory due to some unforeseen problems encountered suddenly, such as insufficient funds, limited logistics transportation capacity, insufficient storage space, etc. Therefore, it is necessary to make a decision for choosing which raw materials should not be purchased so that the company's loss can be controlled within an acceptable range.

In traditional erasable-itemset mining, an item (raw material) or an itemset (a set of raw materials) is called erasable if the profit-loss ratio of not generating the products that need to use the materials is larger than or equal to a single threshold. In this case, each item (raw material) is treated as equally important. However, it is sometimes unfair and impractical because other properties of the materials are not considered, such as cost, volume, weight, etc. Therefore, in this paper, we adopt multiple thresholds to mine erasable itemsets. Different thresholds are given to individual items to meet the requirements of practical applications. Besides, we adopt the loose constraint to decide the threshold value of each itemset with more than one item, such that more erasable itemsets can be derived as candidates to decision makers.

The rest of this paper is organized as follows. Section II describes the related work. Section III gives the problem definition. Section IV explains the sorted closure property. Section V goes into the finer steps of the proposed algorithm. Section VI describes the experiments, and Section VII gives the conclusion.

## II. RELATED WORK

Erasable-itemset mining aims to identify combinations of materials that are not procured, allowing a factory to control loss within an acceptable range. This problem was introduced by Deng et al. in 2009, who also proposed a method called the META (Mining Erasable iTemsets with the Anti-monotone property) algorithm [8]. META employs an searching approach similar to the Apriori algorithm [1][2], progressively discovering all erasable itemsets layer by layer. It uses the downward-closure property effectively to reduce the search space of candidate itemsets. It, however, requires multiple database scans, leading to a lot of execution time. Subsequent to META, several improved algorithms have been introduced to address this issue, including VME (Vertical-format-based algorithm for Mining Erasable itemsets) [7], MERIT (fast Mining ERasable ITemsets) [9], MERIT+ (MERIT enhanced) [23], dMERIT+ (using Difference of NC_Set to enhance MERIT) [23], MEI (Mining Erasable Itemsets ) [22], and BREM (Bitmap-Representation Erasable Mining) [17].

 ISBN: 978-1-68558-144-2 

VME was proposed by Deng et al. in 2010 [7]. It employed a list structure, storing additional information within PID_list to reduce the database scans to only twice, resulting in faster execution time than META. However, each itemset had its own dedicated PID_list, leading to substantial memory usage. Hong et al. proposed an enhancement to VME called BERM [17]. BERM utilized bit vectors to simplify the recording of the PID_list.

In 2012, Deng et al. presented a tree-based algorithm called MERIT [9]. It initially constructed a WPPC (Weighted Pre-Post Code) tree in an FP-growth [10] manner and then calculated NC-sets for each itemset using the WPPC tree. The NC-sets were leveraged to reduce memory usage and enhance execution speed.

MERIT+ was subsequently introduced by Le et al. [23], building upon the original MERIT and incorporating a weighted index to solve the issue of not mining all erasable itemsets. However, the duplicate NC-sets increased memory usage. Then, dMERIT+ [23] adopted a new structure called dNC_sets and used a hash table to eliminate redundant information, optimizing both memory and execution time. Le et al. then proposed the MEI approach, which employed a depth-first search strategy and the dPID_set structure.

In recent years, numerous derivative problems and applications related to erasable mining have been continuously proposed [24][28]. A factory engages in the production of a diverse range of products, considering various additional factors. For example, each product may require a different quantity of materials [15], and some products may experience peak sales only within specific time frames [11][12][13][20]. The ordering sequence from customers is another consideration for certain products [16]. Over time, the variety of products produced by a factory may increase [6][21][25][27], or a factory may face product discontinuation [18][19]. Incremental erasable-itemset mining avoids the need to re-run erasable mining every time a new product is added or removed. It becomes crucial to perform mining selectively on only the itemsets that have an impact rather than the entire database. Different mining problems incessantly appear for actual applications.

## III. PROBLEM DESCRIPTION

Erasable-itemset mining is utilized in the management of manufacturing factories, where various products are produced, as illustrated in Table 1.

TABLE 1: AN EXAMPLE OF A PRODUCT DATABASE.

| *Product Database* | | |
|---|---|---|
| *PID* | *Items* | *Profit* |
| *Product 1* | *ABE* | 200 |
| *Product 2* | *DEF* | 200 |
| *Product 3* | *BCE* | 100 |
| *Product 4* | *ADF* | 100 |
| *Product 5* | *BF* | 300 |
| *Product 6* | *ACDF* | 100 |

In the product database, each product consists of three fields: PID, Items, and Profit. The PID (product identification) serves as a code to distinguish different products. The items represent the materials required to produce a product, and the profit is the earnings obtained after selling the product.

Let us envision a scenario where a factory encounters challenging situations, such as a decline in financial resources, limited logistics transportation capacity, or insufficient storage space for materials. This leads to the inability to procure production from all the raw materials, necessitating a decision on which materials to erase. Consequently, products reliant on these erased materials cannot be manufactured, causing the factory to be unable to sell them and resulting in a decline in profits. The challenge is to determine which materials not to purchase, thereby controlling losses within acceptable proportions for the factory. This problem is known as erasable mining, and the different combinations of materials identified in this process are termed erasable itemsets. Subsequently, we will provide detailed definitions for the relevant terms associated with erasable mining.

**Definition 1:** (Multiple maximum thresholds) The user or factory presets a value between 0 and 1 according to the characteristics of each item, which is used to represent the percentage of the maximum loss in the total revenue that the user or factory can accept if the item is not restocked. The maximum threshold value of each item is expressed as λ. An example is given in Table 2, in which $\lambda(A) = 0.6$ and $\lambda(B) = 0.5$.

TABLE 2: THE MAXIMUM THRESHOLDS OF THE ITEMS IN THE ABOVE EXAMPLE.

| Item | *A* | *B* | *C* | *D* | *E* | *F* |
|---|---|---|---|---|---|---|
| $\lambda$ | 0.6 | 0.5 | 0.4 | 0.7 | 0.3 | 0.8 |

**Definition 2:** (Maximum gain threshold) The maximum gain threshold represents the maximum practical loss acceptable to a user or a factory. The maximum gain threshold of an item *X* is expressed as *MGT(X)*, defined as follows:

$$MGT(X) = Profit_{Total} \times \lambda(X).$$

Take the items *A* and *B* as examples. The total profit of the whole set of products is 200 + 200 + 100 + 100 + 300 + 100 = 1000. According to the maximum threshold of each item in Table 2, *MGT(A)* = 1000 * 0.6 = 600, and *MGT(B)* = 1000 * 0.5 = 500.

**Definition 3:** (Gain) When a particular material cannot be purchased or stocked, the products that need to be produced with this material will be unable to be manufactured. The losses caused by these products that cannot be manufactured are called gains. The gain of the itemset *X* is expressed as *gain(X)* defined as follows:

$$gain(X) = \sum_{\{P | X \cap P.Items \neq \emptyset, P \in PD\}} P.Profit$$

where *P* is a product in the product database *PD*.

In multiple-threshold mining, the maximum thresholds of items are not the same. Different itemsets have their own distinct maximum thresholds. When an itemset contains only one item, we use its given threshold. However, when an itemset consists of two or more items, the calculation of the maximum threshold depends on different constraints. In this paper, we adopt the loose constraint for determining these thresholds. The formula for the loose constraint is defined as follows:

$$\lambda(X) = max(\lambda(I) | \ I \in X).$$

For example, the 2-itemset {*D, E*} contains both items *D* and *E*. Its maximum threshold is then set as *max(λ(D), λ(E)*), which is *max*(0.5, 0.4) = 0.5. Simultaneously, this constraint can also be applied to determine the maximum gain threshold.

Downward closure is a useful property in data mining and has been successfully applied in various mining algorithms, including the tight constraint in multiple threshold mining. However, this property does not apply to the loose constraint. A simple example illustrates this. In Table 1, *Gain(DE)* = 700 ≤ *MGT(DE)* = 700. {*D, E*} is thus an erasable itemset. But the gain of its subset *Gain(E)* = 400 > *MGT(E)* = 300. Thus, {*E*} is not an erasable itemset. Hence, the loose constraint does not possess the downward-closure property.

## IV. Sorted-Closure Property

As mentioned above, the multiple-threshold mining with the loose constraint does not have the downward-closure property. Liu et al. proposed a novel technique called the sorted-closure property to replace the downward-closure property and successfully applied it in frequent itemset mining with multiple minimum supports [26]. In the traditional erasable-itemset mining algorithm, each item in an itemset is usually arranged according to the linguistic order or numerical order, such as {*A, B, C*} or {$A_1$, $A_2$, $A_3$}. However, for using the sorted-closure property, items will be sorted in the descending order of their thresholds. Through such changes, some useful properties will be produced, and unpromising candidate itemsets can be effectively pruned off. Thus, the time spent on scanning databases for calculating gains and verification can be reduced.

For example, the items in Table 2 are sorted in descending order according to their maximum thresholds. The sorted result is shown in Table 3. According to this order, the itemset {*D, E, F*} is rearranged as {*F, D, E*}.

TABLE 3: THE SORTED ITEMS ACCORDING TO THEIR MAXIMUM THRESHOLDS.

| Item | *F* | *D* | *A* | *B* | *C* | *E* |
|---|---|---|---|---|---|---|
| *λ* | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 |

The following four theorems can then be deduced for the sorted items. By employing these theorems, we can effectively reduce the search space of candidate itemsets.

**Theorem 1** Assume item *X* is an erasable 1-itemset and an item *Y*, after sorting, is located before item *X*. The gain of the 1-itemset *X* must be less than or equal to the maximum gain threshold of the 1-itemset *Y*. That is, *gain(X)* ≤ *MGT(Y)*.

**Theorem 2** Assume itemset *X* is an erasable 2-itemset under the loose constraint and represented as {$item_1$, $item_2$}, where $item_1$ is located before $item_2$ after sorting. Then the following two conditions must be satisfied:

$gain(item_1) \le MGT(item_1)$, and
$gain(item_2) \le MGT(item_1)$.

**Theorem 3** Assume itemset *X* is an erasable *k*-itemset (*k* ≥ 3) under the loose constraint with the first and the second items in *X* having different maximum gain thresholds. Then, the (*k*-1)-subitemsets of *X* containing the first item in *X* must also be erasable under the loose constraint.

**Theorem 4** Assume itemset *X* is an erasable *k*-itemset (*k* ≥ 3) under the loose constraint with the first and the second items in *X* having the same maximum gain thresholds. Then, all the (*k*-1)-subitemsets of *X* must also be erasable under the loose constraint.

## V. The Proposed Algorithm under the Loose Constraint

In this section, we will introduce the proposed algorithm for finding erasable itemsets under the loose constraint. It uses the four theorems mentioned above to reduce candidate itemsets. The algorithm is described below.

STEP 1: Sort the items according to the descending order of their maximum thresholds.

STEP 2: Scan the whole database to calculate the total gain of the database and the gain of each item.

STEP 3: Calculate the maximum gain threshold of each item using the total gain of the database.

STEP 4: Check the sorted items one by one from the front until the first one *X* with its gain smaller than or equal to its maximum gain threshold; Put *X* in the set of candidate 1-itemsets $CI_1$.

STEP 5: Check the sorted items after the above *X* one by one; If the gain of an item is smaller than or equal to the maximum gain threshold of *X*, add the item to $CI_1$.

STEP 6: Check whether the gain of each candidate 1-itemset in $CI_1$ is less than or equal to its own maximum gain threshold. If the 1-itemset conforms to the above condition, put it into the set of erasable 1-itemsets $EI_1$.

STEP 7: If $EI_1$ is empty, then no erasable itemsets are found, and the algorithm stops.

STEP 8: Form candidate 2-itemsets $CI_2$ by joining the candidate 1-itemsets in $CI_1$.

STEP 9: For each candidate 2-itemset in $CI_2$ with its first item being an erasable 1-itemset in $EI_1$, check whether its gain is less than or equal to its maximum gain threshold. If the itemset conforms to the above condition, put it into the set of erasable 2-itemsets $EI_2$.

 ISBN: 978-1-68558-144-2 

STEP 10: If $EI_2$ is empty, then only erasable 1-itemsets are found; we output $EI_1$ and stop the algorithm.

STEP 11: Set $k$ =3, where $k$ is used to represent the number of items in an itemset to be processed.

STEP 12: Find any two itemsets in $EI_{k-1}$ with the same ($k$-2) items to join and generate a $k$-itemset $X$. If the first two items (according to the sorted list) in $X$ have different maximum gain thresholds, then add $X$ to the candidate $k$-itemsets $CI_k$ only if all the ($k$-1)-subitemsets of $X$ containing the first item of $X$ are in $EI_{k-1}$; If the first two items in $X$ have the same maximum gain thresholds, then add $X$ to $CI_k$ only if all the ($k$-1)-subitemsets of $X$ are in $EI_{k-1}$.

STEP 13: Check whether the gain of each candidate $k$-itemset in $CI_k$ is less than or equal to its maximum gain threshold. If the itemset conforms to the above condition, put it into the set of erasable $k$-itemsets $EI_k$.

STEP 14: If $EI_k$ is empty, then erasable 1-itemsets to ($k$-1)-itemsets are the desired; we output them and stop the algorithm; Otherwise set $k$ as $k$+1 and go to STEP 12.

Note that in STEP 12, $CI_k$ is formed effectively according to Theorems 3 and 4.

## VI. Experimental Results

To enhance clarity in assessing the algorithm's performance, we conducted a comparison between multiple thresholds under the loose and the tight constraints [14] and under single thresholds [8] in our experiments. The program was executed on a system with a CPU i7-9750H@2.60GHz, 8GB RAM, and Windows 10. The programming language used was Java 19.0.1. The synthetic dataset P100KI0.05KD10 was generated by the IBM data generator [3] and named based on its parameters, where $P$ represents the number of products, $I$ represents the number of items, and $D$ represents the average number of materials in each product. However, the data generated by the IBM data generator lacked profit information. To better simulate real-world scenarios, we thus introduced profit values using a normal distribution, denoted as $N(100, 20)$, where the two parameters represent the mean and the standard deviation, to ensure most product profits fall within a moderate range. Excessively high profits could result in unmarketable high prices, while excessively low profits might affect factory operations.

Concerning the thresholds set in the experiments, we initially defined an interval, generating the thresholds of the items uniformly within the range, denoted as $U(H, L)$. Here, $H$ is the highest threshold, and $L$ is the lowest within the interval. $H$ and $L$ are then used as the two thresholds, which were set, respectively, in the single-threshold algorithm. In the experiments, we first maintained other parameters constant while incrementally increasing each threshold by 0.01. The variation allowed us to observe its influence on the algorithm.

Figures 1 and 2 represent the quantities of erasable itemsets and candidate itemsets, respectively. Erasable itemsets are derived from the candidate itemsets. Hence, the number of erasable itemsets is always less than that of candidate itemsets. As the threshold values increase, both the quantities exhibit an upward trend. The two figures show that the quantities of the erasable itemsets and the candidate erasable itemsets under the tight and the loose constraints in multi-threshold mining fall between those of the single-threshold algorithm under two single thresholds. Specifically, the loose constraint tends to be closer to the highest single threshold, while the tight constraint tends to be closer to the lowest single threshold.

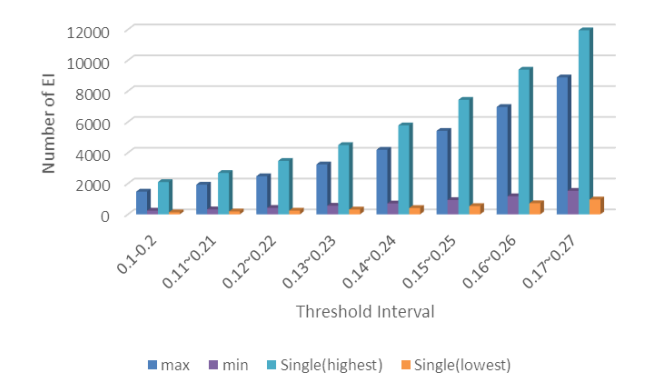


Figure 1. The numbers of erasable itemsets for different threshold intervals.

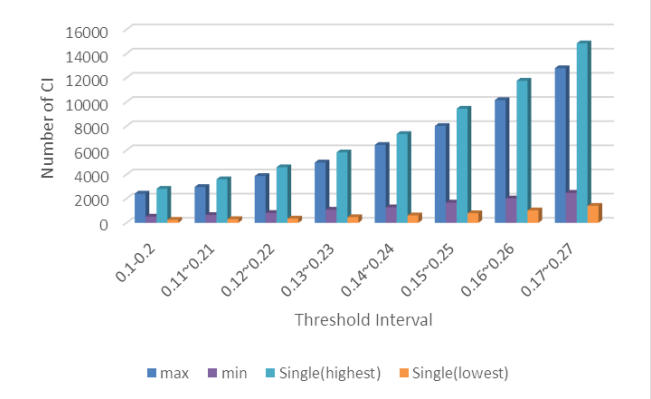


Figure 2. The numbers of candidate itemsets for different threshold intervals.

Figure 3 illustrates the memory usage for each program. The extent of memory usage is determined by the number of products, erasable itemsets, and candidate itemsets. As the number of products remains constant, the primary influencing factors are the quantities depicted in Figure 1 (erasable itemsets) and Figure 2 (candidate itemsets). With an increase in the quantities of both the erasable and candidate itemsets, the memory usage also increases accordingly.

 ISBN: 978-1-68558-144-2

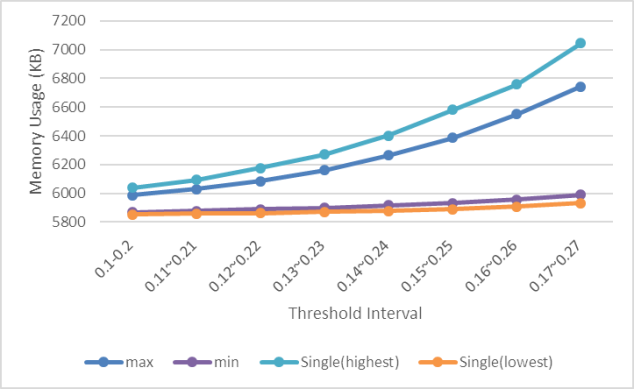


Figure 3. The memory usage for different threshold intervals.

Lastly, Figure 4 represents the execution time for each program. The most time-consuming aspect during the mining process is the generation and validation of candidate itemsets, which are closely related to the number of candidate itemsets, as shown in Figure 2. As the number of candidate itemsets increases, the execution time also rises accordingly.

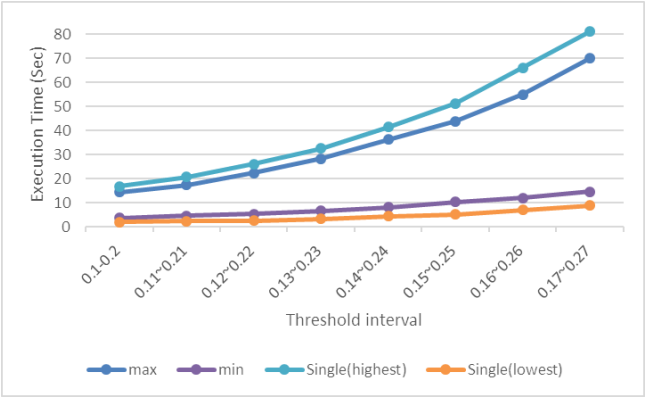


Figure 4. The execution time for different threshold intervals.

## VII. Conclusion and Future Work

In this paper, we have proposed an algorithm for the loose constraint in multiple-threshold mining. We have also demonstrated the applicability of relevant theorems derived from the sorted closure, successfully solving the issue of lacking downward closure in the loose constraint. In our experiments, we have not only compared the algorithms with different constraints but also contrasted them with the single-threshold META algorithm. We have also comprehensively analyzed execution time, memory usage, erasable-itemset quantity, and candidate-itemset quantity. We will continuously explore designing algorithms tailored to various constraints for future research. Optimizing the execution time of multi-threshold mining could also be a focus of further investigation.

## Acknowledgment

This work was supported by the National Science and Technology Council, Taiwan, under the grant NSTC 112-2221-E-390-014-MY3.

## References

[1] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” *The 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207-216, 1993.

[2] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” *The 20th International Conference on Very Large Data Bases*, pp. 487-499, 1994.

[3] R. Agrawal and R. Srikant, *Quest Synthetic Data Generation Code*. 1994.

[4] R. Agrawal and R. Srikant, “Mining sequential patterns,” *The Eleventh International Conference on Data Engineering*, pp. 3-14, 1995.

[5] R. Chan, Q. Yang, and Y.-D. Shen, “Mining high utility itemsets,” *The Third IEEE International Conference on Data Mining*, pp. 19-19, 2003.

[6] R. Davashi, “IME: efficient list-based method for incremental mining of maximal erasable patterns,” *Pattern Recognition*, vol. 148, pp. 110166, 2024.

[7] Z. Deng and X. Xu, “An efficient algorithm for mining erasable itemsets,” *The International Conference on Advanced Data Mining and Applications*, pp. 214-225, 2010.

[8] Z. H. Deng, G. D. Fang, Z. H. Wang, and X. R. Xu, “Mining erasable itemsets,” *The 2009 International Conference on Machine Learning and Cybernetics*, pp. 67-73, 2009.

[9] Z. H. Deng and X. R. Xu, “Fast mining erasable itemsets using NC_sets,” *Expert Systems with Applications*, vol. 39(4), pp. 4453-4463, 2012.

[10] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” *ACM Sigmod Record*, vol. 29(2), pp. 1-12, 2000.

[11] T.-P. Hong, J.-X. Li, Y.-C. Tsai, and W.-M. Huang, “Tree-based unified temporal erasable-itemset mining,” *The Asian Conference on Intelligent Information and Database Systems*, pp. 224-233, 2023.

[12] T. P. Hong, H. Chang, S. M. Li, and Y. C. Tsai, “A unified temporal erasable itemset mining approach,” *The 2021 International Conference on Technologies and Applications of Artificial Intelligence*, pp. 194-198, 2021.

[13] T. P. Hong, H. Chang, S. M. Li, and Y. C. Tsai, “A dedicated temporal erasable-itemset mining algorithm,” *The International Conference on Intelligent Systems Design and Applications*, pp. 977-985, 2022.

[14] T. P. Hong, Y. C. Chang, W. M. Huang, and W. Y. Lin, “Multiple-threshold erasable mining under the tightest constraint,” *The International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 369-377, 2022.

[15] T. P. Hong, H. W. Chen, W. M. Huang, and C. H. Chen, “Erasable pattern mining with quantitative information,” *The 2019 International Conference on Technologies and Applications of Artificial Intelligence*, pp. 1-4, 2019.

[16] T. P. Hong, Y. L. Chen, W. M. Huang, and Y. C. Tsai, “Erasable-itemset mining for sequential product databases,” *International Conference on Hybrid Intelligent Systems*, pp. 566-574, 2022.

[17] T. P. Hong, W. M. Huang, G. C. Lan, M. C. Chiang, and J. C. W. Lin, “A bitmap approach for mining erasable itemsets,” *IEEE Access*, vol. 9, pp. 106029-106038, 2021.

[18] T. P. Hong, C. C. Li, S. L. Wang, and C. W. Lin, “Maintenance of erasable itemsets for product deletion,” *The Fifth Multidisciplinary International Social Networks Conference*, pp. 1-4, 2018.

[19] T. P. Hong, C. C. Li, S. L. Wang, and J. C. W. Lin, “Reducing database scan in maintaining erasable itemsets from product

 ISBN: 978-1-68558-144-2 

deletion," *The 2018 IEEE International Conference on Big Data*, pp. 2627-2632, 2018.

[20] T. P. Hong, J. X. Li, Y. C. Tsai, and W. M. Huang, "Unified temporal erasable itemset mining with a lower-bound strategy," *The 2022 IEEE International Conference on Big Data*, pp. 6207-6211, 2022.

[21] T. P. Hong, K. Y. Lin, C. W. Lin, and B. Vo, "An incremental mining algorithm for erasable itemsets," *The 2017 IEEE International Conference on Innovations in Intelligent Systems and Applications*, pp. 286-289, 2017.

[22] T. Le and B. Vo, "MEI: an efficient algorithm for mining erasable itemsets," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 155-166, 2014.

[23] T. Le, B. Vo, and F. Coenen, "An efficient algorithm for mining erasable itemsets using the difference of NC-Sets," *The 2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2270-2274, 2013.

[24] T. Le, B. Vo, and G. Nguyen, "A survey of erasable itemset mining algorithms," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4(5), pp. 356-379, 2014.

[25] G. Lee, U. Yun, H. Ryang, and D. Kim, "Erasable itemset mining over incremental databases with weight conditions," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 213-234, 2016.

[26] B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports," *The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 337-341, 1999.

[27] H. Nam, U. Yun, E. Yoon, and J. C. W. Lin, "Efficient approach for incremental weighted erasable pattern mining with list structure," *Expert Systems with Applications*, Vol. 143, pp. 113087, 2020.

[28] D. M. D. Raj and M. Ranganathan, "A comprehensive survey on erasable itemset mining," *International Journal of Computer Science and Information Security*, vol. 15(7), pp. 184-201, 2017.

 ISBN: 978-1-68558-144-2 