AiLECS lab
Ai for law enforcement and community safety

Report TN22/06 (Honours thesis submission). Date of publication: March 28, 2022.

# Law Enforcement Data Interoperability

Mathew Lim

Monash University, Melbourne, Australia

**Abstract**

In law enforcement (LE), interoperability, i.e., the ability to exchange information between databases and systems, enhances the ability of agencies to detect and investigate crime. A fundamental way of improving interoperability is data integration, but integrating LE databases is often difficult due to heterogeneity of database types and the semantics of the data. In this study, an ontology-based and Linked Data approach for integrating heterogeneous LE databases is proposed. The approach is evaluated for use in an operational setting by LE data domain experts. The evaluation feedback indicates that the approach has the potential to address some of the common challenges faced when integrating heterogeneous LE databases, and could provide benefit if used in an LE agency's operational systems.

## 1. Introduction

In organisations, it is common for databases to be developed independently, each storing data with different granularity, schema and details. This leads to data silos, which refers to information sources that are isolated from each other and whose information cannot easily be exchanged. Organisations often want to increase interoperability for business gain [1], where interoperability is the ability for two or more systems or components to exchange information, and to use the information that has been exchanged. At an enterprise level, interoperability can refer to the ability for enterprises to interact, for example, through the sharing of data [2].

In law enforcement (LE), interoperability is highly desired, because the ability to exchange information between law enforcement agencies (LEAs) and their systems are "vital for effective and successful law enforcement and prosecution" [3]. It is natural for data silos to form as systems and databases are built to fulfil specific LEA functions. Functions can include digital forensics, incident response, counter-terrorism, criminal justice, forensic intelligence and situational awareness; for law enforcement investigations to be effective, information needs to be "shared in a form that is usable in any of these contexts" [4]. When LEAs require data to be combined from multiple databases to carry out LE effectively, poor interoperability becomes a major hindrance [47].

In this paper, an approach for integrating heterogeneous LE databases is proposed. To evaluate the utility of the approach for use in an operational setting, a prototype solution will be built, and then evaluated by data domain experts from an LEA.

The rest of this paper is organised as follows. The "Background" section provides an overview of data integration challenges and approaches. The "Methods" section describes the design-science methodology and system development research process used to build and evaluate a prototype solution. The "Results" section describes the design outputs and findings. The "Discussion" section discusses how well the proposed approach met its requirements, and discusses the results of the evaluation. The "Conclusions" section concludes the paper and suggests directions for future research.

## 2. Background

A fundamental way of increasing interoperability is data integration, which is the process of "combining data residing at different sources, and providing the user with a unified view of these data" [5]. For example, multiple siloed LE databases could be integrated by combining their data. However, the greater the heterogeneity of the databases, the more difficult it can be to combine their data.

There are different forms of heterogeneity, such as structural (differences in schemas), syntactic (differences in data representations), and semantic (differences in meanings) [7]. Semantic heterogeneity, in particular, "results from the fact that in many cases the same or overlapping data is replicated in two or more databases. Different conceptualizations and different database schemas are typically used to represent this replicated data" [6]. In data integration, semantic heterogeneity has proved to be a more difficult problem to overcome than structural or syntactic heterogeneity [7].

Ontologies have long been used in data integration approaches for overcoming semantic heterogeneity. Ontologies are useful for explicitly describing the semantics of the data sources, and then the association of semantic correspondences between concepts defined in the ontologies. There are three main architectures

for ontology-based data integration approaches. In the single ontology architecture, a single ontology specifies the semantics of all the databases. In the multiple ontology architecture, each database has a local ontology that specifies the semantics of the database, while an inter-ontology mapping specifies semantic correspondences between terms in the local ontologies. In the hybrid architecture, each database has a local ontology that specifies the semantics of the database, where local ontologies use a shared vocabulary. The shared vocabulary could be an ontology [8].

Having covered data integration in general, the discussion will now turn to the topic of integrating databases, including approaches for integrating databases that address the problem of heterogeneity.

### 2.1 Integrating Databases

When integrating relational databases, a global schema can be created to provide a unified view of the databases. Mappings between the global schema and the databases enable queries posed over the global schema to be reformulated in terms of a set of queries over the databases [5]. Relational database systems have been the primary type of database used in past years, and are relatively interchangeable due to standardisation; for example, relational database systems commonly use SQL, the standard query language [9]. However, NoSQL databases are becoming increasingly popular for applications for which relational databases are not well suited. Relational databases generally do not perform as well as NoSQL databases when extreme scaling is required, such as in massively accessed Web [10] and Big Data [9] applications. As LEAs are facing the spectre of substantial increases in digital evidence data from sources like mobile phones, cloud and the Internet of things [11], LEAs will likely turn to NoSQL databases to solve Big Data challenges. In LE, storing and querying entities of interest (e.g., persons, organisations, and locations) and their relationships is useful for criminal investigations. Graph databases, a type of NoSQL database, are being chosen for this purpose over relational databases. Graph databases can easily model the entities and relationships as the vertices and edges of a graph, and provide better support for querying this data than relational databases [36].

There is no standard query language across the various types of NoSQL databases, like graph, key-value, and document databases [9]. When integrating relational databases, the queries posed over the global schema and databases can use SQL. When integrating a mix of different types of NoSQL or relational databases, data integration becomes more difficult because queries must be reformulated into different query languages supported by the source databases.

To address the difficulty of integrating a mix of different types of databases, various approaches have been proposed. For example, one approach was to develop a virtualisation architecture for querying and joining NoSQL and relational databases in a single SQL query [9]. This approach, however, did not address semantic heterogeneity.

Ontology-based approaches have been used for integrating NoSQL databases while addressing semantic heterogeneity. For example, one ontology-based approach involved generating a local ontology from each database using non-standard description logic (DL) reasoning services. Semantic correspondences (i.e., an alignment) between concept definitions present in the local ontologies were then discovered using a novel alignment method, and a global ontology was generated from the set of semantic correspondences. Queries were expressed over the global ontology [12].

In another ontology-based approach that integrated NoSQL databases, each database was converted into a corresponding MongoDB [48] document-oriented database; then, local ontologies were generated from each MongoDB database by extracting concepts, relations, roles, domains and ranges. A global ontology was generated based on similarities discovery between concepts in the local ontologies [13].

### 2.2 Linked Data

Linked Data refers to “a set of best practices for publishing and interlinking structured data on the web” [14]. In Linked Data, data is represented using the Resource Description Framework (RDF) [14]. The source databases used in Linked Data are often relational, with the data mapped from the relational model to RDF [15].

In RDF, entities of interest are called resources. Identity linking refers to linking different resources that in fact represent the same real-world entity. It is a common occurrence for sets of data to be created independently and use different resources to represent the same real-world entity. Identity links “enable clients to retrieve further descriptions about an entity from other data sources” [14], and thus supports the integration of these data sources. Vocabulary linking supports the integration of data by linking between the schemata that are used by different data sources [14]. These identity and vocabulary links act as semantic correspondences in Linked Data and help to overcome semantic heterogeneity [16]. Ontologies are used extensively in Linked Data for the formal specification and integration of RDF data. A basic ontology might specify the classes, properties and relations of the data. Ontologies can be enriched with more complex axioms, such as identity and vocabulary links, which allows for richer inferences [17] by software called reasoners [18].

Ontologies have become popular for representing and exchanging LE data. Ontologies have been designed for LE domains such as security incidents

[19], digital forensics [20], [21], [22], [23], and organised crime [24]. LE ontologies have been intended to facilitate data exchange between tools [21], LEAs [24], and LE domains [4].

### 2.3 Summary

Ontology-based approaches have been used to overcome the challenges of semantic heterogeneity and NoSQL databases [12], [13]. Linked Data seems to be well suited for implementing ontology-based data integration solutions, even without an intention to publish the data on the web. However, the authors in [12] and [13] did not address data integration in an LE context, or Linked Data. There appears to be a gap regarding whether ontology-based approaches and Linked Data can be applied to the integration of heterogeneous LE databases. This leads to the research question:

(RQ) Can ontology-based approaches and Linked Data be used for the data integration of heterogeneous LE databases?

## 3. Methods

To address the research question, a prototype solution was built to integrate two databases that were representative of heterogeneous LE databases. The first database was a relational database produced by a digital forensics application, which detected person names in a collection of text files and stored the results in the database. The second database was a NoSQL graph database containing person, object, location and event (POLE) entities and the relations between them. To demonstrate integration, it was decided to link data in the two databases based on matching person names. For example, if "John Smith" was a person name in the digital forensics database, then this data should be linked with person entities named "John Smith" in the POLE database. Linking based on person names is a common use case in LE, as discussed below.

The prototype was created using a design science approach, where design science is an information systems research methodology. Design science focuses on the production of artefacts that are beneficial to people and organisations. Design science is inherently iterative; a build-and-evaluate cycle continues until the artefacts satisfy requirements. The artefacts are rigorously evaluated using well-executed evaluation methods [25].

A system development research process was used to build and evaluate the prototype solution [26]. The process consisted of these phases:

- Construct a conceptual framework.
- Develop a system architecture.
- Analyse and design the system.
- Build the prototype system.
- Observe and evaluate the system.

The evaluation of the prototype solution was done using a focus group. In design science, focus groups are often used as a method for evaluating and refining artefacts. Focus group participants are asked questions to solicit their feedback about an artefact's utility. The feedback is utilised to make design improvements to the artefact [33]. For the evaluation of the prototype solution, a focus group session was conducted with four LE data domain experts from the Australian Federal Police (AFP) using teleconferencing. In the first part of the session, an outline of the research was presented to the experts. Next, the experts received a demonstration of the prototype solution. The experts were then asked interview questions to elicit their feedback regarding the utility of the proposed approach. In accordance with the semi-structured interview structure [49], a prepared interview protocol was used, but follow-up questions were also asked to elicit further details from the experts.

## 4. Results

### 4.1 Conceptual Framework and System Requirements

Dissecting the research question led to the surfacing of these broad requirements:

1. *An ontology-based data integration approach should be considered for addressing heterogeneity:* Ontology-based approaches have long been used to address the key challenge of semantic heterogeneity [8], and have also been used for the integration of NoSQL databases [12], [13].
2. *The approach should minimise manual effort required to create ontologies:* The manual construction of ontologies could be time consuming and error-prone. Automatic generation of ontologies should be considered as a way to minimise manual effort [13].
3. *The approach should be amenable to changes in the databases:* In ontology-based data integration approaches, the multiple ontology and hybrid architectures are more amenable to changing the databases than the single ontology architecture [8]. In an operational setting, it is likely that the databases being integrated will be modified over time, or new databases will be added, so an architecture that makes it relatively easy to change the databases should be chosen.
4. *Building the solution using Linked Data practices should be considered:* Linked Data practices could be taken advantage of in order to build the ontology-based data integration solution.

### 4.2 System Architecture

In the architecture, each database has a local ontology, i.e., an ontology that describes only its database's data. An advantage of using local ontologies is that each ontology can be developed independently, so changes to a

database affects only its local ontology and no others [8]. Another advantage is that there are known ways to generate local ontologies, which reduces the manual effort required to create the ontologies [12], [13].

A single global ontology is used to described the unified view of the data. The global ontology imports the local ontologies, so the local ontologies also form part of the global ontology. Semantic correspondences between the imported local ontologies can then be defined in the global ontology. Using a global ontology to define semantic correspondences between local ontologies was previously done in [12].

Queries are expressed in terms of the global ontology. The querying system reformulates the query into appropriate sub-queries for each database. After the results are retrieved from each database, the querying system combines and returns the results, thus integrating the data from the databases.

The system architecture is illustrated in Figure 1.

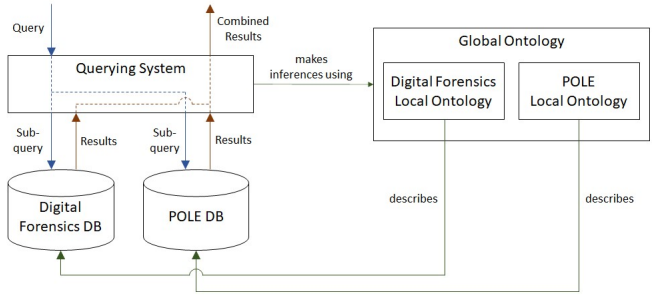


**Figure 1.** System Architecture

C. System Design The design was based around using Linked Data practices and technologies to achieve the data integration. This decision was made because Linked Data supports ontology-based data integration. In addition, Linked Data supports the integration of different types of databases. The digital forensics relational database and the POLE graph database can be integrated by firstly transforming their data to RDF, which serves as a common data representation for querying and integration.

In the design, the source database of the POLE data is a Neo4j graph database [37]. Neo4j is a market-leading graph database product [27]. POLE data focuses on various entities and their relations. As graph databases are well suited for modelling and querying entities and their relations [28], a graph database is a natural choice for storing POLE data. To transform the data to RDF, the data is exported from the Neo4j database as RDF to a file, using a Neo4j feature.

The Ontop ontology-based data access (OBDA) system is used to expose the digital forensics relational data as RDF [38]. The RDF can be queried using SPARQL, a standard query language [29]. An advantage of using the Ontop OBDA system is that the transformation from relational data to RDF is done “on the fly”, without needing to re-export to RDF whenever the relational data changes. Ontop offers support for only relational databases [39], and therefore could not be used to expose the POLE data in the Neo4j graph database as RDF.

One local ontology describes the digital forensics RDF data. Another local ontology describes the POLE RDF data. Both local ontologies are imported into the global ontology, and therefore also form part of the global ontology.

Stardog [40] is an RDF database product and serves several functions in this design. The POLE RDF data and the ontologies are imported into a Stardog database. Stardog provides the SPARQL querying system, which will execute queries on the RDF data while using the ontologies to make inferences. Stardog is able to execute queries that combine results from the POLE RDF data that is stored in the Stardog database, and the digital forensics RDF data which Stardog accesses via the Ontop OBDA system. This design thus supports the integration of the digital forensics and POLE data.

The system design is illustrated in Figure 2.

## 4.3 The Prototype Solution

### 4.3.1 Creating the digital forensics database:

The digital forensics database was created using Autopsy, a digital forensics application [41]. Autopsy allows a user to run forensics tasks, called ingest modules [42], on a variety of data source types. Ingest modules can be used to detect artefacts of interest in the data sources.

Autopsy was used to detect person names in the Enron email dataset; the Enron email dataset is a large collection of emails that was made public during a legal investigation into the Enron corporation [30]. This dataset was chosen because it is representative of a collection of files that would typically be analysed during a digital forensics investigation.

Autopsy did not provide an ingest module to extract person names, but allowed a user to develop and run custom ingest modules. Autopsy supported the installation of custom ingest modules as NetBeans modules [43]. Accordingly, a custom ingest module for extracting person names was implemented as a NetBeans module and installed into Autopsy. The custom ingest module initially used Apache OpenNLP, a framework for natural language processing (NLP), to extract person names from text [50]. However, the quality of the extraction was noticeably poor, as many of the extracted strings were clearly not person names, or contained extra characters on either side of the person names. The decision was made to use Stanford CoreNLP, another NLP framework [31], in place of Apache OpenNLP. NLP uses trained named entity recognition (NER) models for extracting named entities from text [51]. As training a model would have been a non-trivial task, a pre-trained model for extracting person

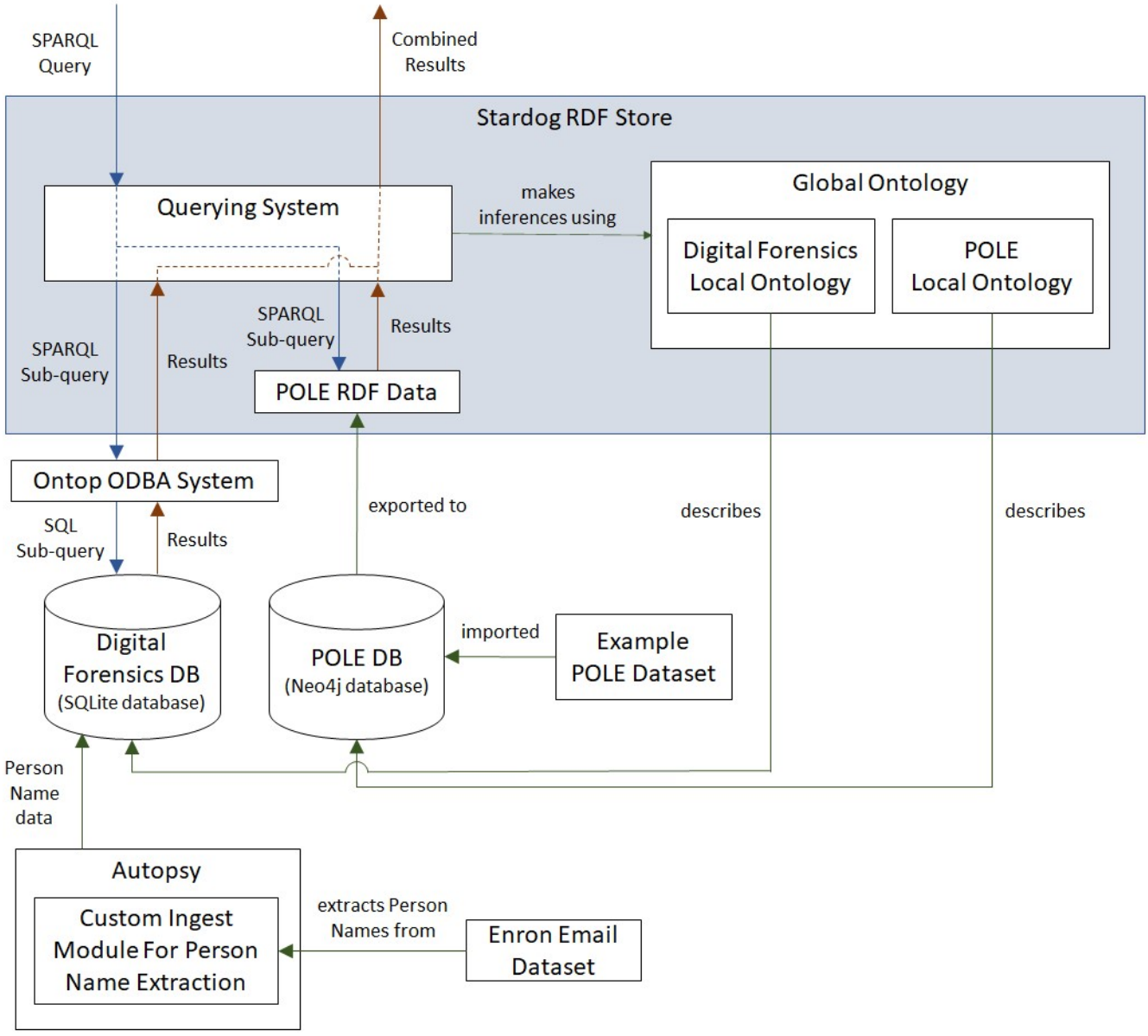


**Figure 2.** System Design

names, provided by Stanford CoreNLP, was used. The custom ingest module was able to extract hundreds of thousands of person names when run on a small portion of the Enron email dataset. The extracted person names were a variety of first, last and full names (e.g, “Pam Butler”, “Karen K Heathman”, “Michael”, “Alvarez”). While Stanford OpenNLP produced noticeably better results than Apache CoreNLP, a small proportion of the extracted strings still contained extra characters, (e.g, “Ted Murphy@ENRON”, “Andrew S Fastow@ECT”, “Rick http://www.laconiamcweek.com/”), or, more rarely, were clearly not person names (e.g., “Risktrac”). It is possible that the quality of the extraction could be improved by using a custom NER model trained on data that is similar to the Enron email dataset; however, investigating this was outside the scope of this study.

Autopsy creates an SQLite [44] relational database for storing the analysis results of a digital forensics case; this includes information about artefacts detected by ingest modules, which is stored in a set of related tables. To more easily view the person names found by the custom ingest module, I created a view in the SQLite database, which will be referred to as the Person Name view. The Person Name view consists of a column containing the person names found by the custom ingest module, and a column containing the artefact ID, which is a unique identifier given by Autopsy to artefacts of interest detected by ingest modules.

#### 4.3.2 Creating the POLE database:

The Neo4j vendor provided a publicly available dataset to demonstrate how POLE data can be used with Neo4j graph databases [45]. The dataset was downloaded and loaded into a Neo4j database. 3) Creating the digital forensics local ontology: The local ontology for the digital forensics RDF data was created using Ontop. Ontop supports the automatic generation (i.e., bootstrapping) of ontologies from relational databases [29]. The ontology describes the RDF data that will be mapped from the relational data. When supplied with the connection parameters to the digital forensics database, and an SQLite JDBC driver [46] file, Ontop generated an ontology file from the schema of the database. The ontology was written in the Web Ontology Language (OWL) [18] and contained a class definition for each table or view, an object property definition for each foreign key relationship, and a data type property definition for each column.

Manual modifications to the ontology file were made to support the Person Name view. The view had a column containing artefact IDs that referred to the primary key of a table containing artefact data, but as views do not have foreign keys, Ontop did not generate an object property definition to represent the relationship between the view and this table. Thus, the object property definition had to be added manually.

#### 4.3.3 Creating the POLE local ontology:

The local ontology for the POLE RDF data was partially generated using a Neo4j feature, which generated an OWL ontology containing a class definition for each kind of entity, and an object property for each relation type, in the Neo4j database. A data type definition for each property in each kind of entity was manually added to the ontology.

#### 4.3.4 Transforming the digital forensics data to RDF:

Ontop was used to deploy a HTTP SPARQL endpoint [29]. The endpoint allowed clients to execute SPARQL queries to retrieve data from the digital forensics database, with the data transformed to RDF before being returned to the client. Three files were supplied to Ontop; one contained connection parameters to connect to the digital forensics database, one was the local ontology and one was a mapping file. The mapping file defined how the relational data was mapped to RDF. Ontop was used to automatically generate the mapping file from the digital forensics database in accordance with the Direct Mapping standard [32].

Some manual modifications were made to the mapping file to support the Person Name view. While Ontop did generate mappings for this view, they mapped each row to a blank node, which in RDF represents an entity without using an URI to identify it [15]. This was expected behaviour when generating mappings for views, as views have no primary keys [32]. In RDF, URIs are used to identify things [14]. Resources with URIs can be linked with identity links, and identity linking supports data integration. Hence, the mappings file was manually modified such that rows in the Person Name view were mapped to RDF resources with URIs in the format:

```
http://www.example.org/global/person name
```

where `http://www.example.org/global/` is the namespace of the global ontology, and `{person name}` is the URL-encoded value of the view's person name column. For example, the URI for the person name "John Smith" would be

```
http://www.example.org/global/John%20Smith.
```

Assigning the resources a URI allowed them to be later linked to other resources using identity linking, thus supporting the data integration.

Another manual modification to the mapping file was made to support the Person Name view. The view had a column containing artefact IDs that referred to the primary key of a table containing artefact data; Ontop generated mappings for each foreign key relationship in the source database, but as views do not have foreign keys, Ontop did not generate a mapping for the relationship between the view and this table. This mapping was manually added to the mapping file so that this relationship would be represented in the RDF data, which in turn meant that SPARQL queries could use this relationship to retrieve artefacts related to the person names.

#### 4.3.5 Transforming the POLE data to RDF:

The POLE data was exported as RDF to a file using a Neo4j feature, and then imported into a Stardog database.

#### 4.3.6 Creating the global ontology:

The global ontology was created as an OWL ontology file. It imported the local ontologies, meaning that the definitions of the local ontologies formed part of the global ontology. The aim was to link the person resources in the POLE RDF data (which will be referred to as the POLE-person resources) with the person-name resources in the digital forensics RDF data (which will be referred to as the DF-person-name resources) who had matching person names. This was achieved by firstly defining, in the global ontology, a Person class.

```
@prefix : <http://www.example.org/global/> .
:Person rdf:type owl:Class .
```

Next, Person resources (which will be referred to as global-ontology-person resources) for each unique per-

son name that appeared in the POLE-person resources were defined in the global ontology, e.g.,

```
<http://www.example.org/global/John%20Smith>
  rdf:type owl:NamedIndividual, :Person .
```

If the global-ontology-person resource was created for the unique person name n, then an identity link was defined in the global ontology between the global-ontology-person resource and POLE-person resources with the person name n, e.g.,

```
<http://www.example.org/global/John%20Smith>
  owl:sameAs <neo4j://graph.individuals#448> .
```

where neo4j://graph.individuals#448 is the URI of a POLE-person resource with the person name *n*.

The URIs assigned to the global-ontology-person resources used the same namespace and format as the URIs assigned to the DF-person-name resources; thus, if both resources represented persons with the same name, their URIs would be identical, and they could be considered linked on the basis of having the same identifier.

Thus, DF-person-name resources and POLE-person resources with matching person names are in effect linked, with a global-ontology-person resource acting as an intermediary between them. For example, say there is a DF-person-name resource with the person name "John Smith", which has the URI

`http://www.example.org/global/John%20Smith`.

Also, there is a POLE-person resource with the person name "John Smith". Then, the global ontology will have a global-ontology-person resource with the URI

`http://www.example.org/global/John%20Smith`,

that links to the DF-person-name resource on the basis of having the same identifier, and links to the POLE-person resource using an identity link.

Another purpose of the global ontology was to define a schema that was more intuitive for expressing queries than the schemas defined in the local ontologies. An Artefact class was defined.

```
:Artefact rdf:type owl:Class .
```

A mentionedIn object property was defined, representing a relation where a Person is mentioned in an Artefact.

```
:mentionedIn rdf:type owl:ObjectProperty ;
             rdfs:domain :Person ;
             rdfs:range :Artefact .
```

A filePath data type property was defined in the global ontology, for which the domain is the Artefact class.

```
:filePath rdf:type owl:DatatypeProperty ;
          rdfs:domain :Artefact ;
          rdfs:range xsd:string .
```

Vocabulary links, defined in the global ontology, were used to link the filePath data type property to a corresponding property in the digital forensics local ontology. The vocabulary links allowed the Person class, Artefact class, mentionedIn object property and filePath data type property to be used in queries, in place of the corresponding definitions in the local ontologies.

#### 4.3.7 Querying the RDF Data:

Stardog was used to execute SPARQL queries on the RDF data. For the Stardog reasoner to make inferences during querying, the global and local ontology files were loaded into the same Stardog database as the POLE RDF data. Stardog supported the execution of a query that combined results from both the POLE RDF data, which was stored in the Stardog database, and the digital forensics RDF data, via the Ontop SPARQL endpoint. For example, a query could be used to retrieve the persons who had commited a crime (from the POLE data) and whose names had been detected in a Enron dataset file (from the digital forensics data), and related data from either data source, such as the type of crime committed, and the file name of the Enron dataset file. This query is shown in Appendix A.

### 4.4 Evaluate the System

In the focus group session, the LE data domain experts were presented with an outline of the research and a demonstration of the prototype solution. They were then asked interview questions to elicit their feedback on the utility of the proposed approach, which is discussed in the "Discussion" section below. The interview protocol is shown in Appendix B. The experts described the key data integration challenges they faced in the context of LE, which were:

1. Data cleaning: The cleanliness of data was seen by the experts as a major challenge. The surveillance data being collected is increasingly complex and disparate, making it difficult to use with LE systems. The experts emphasised that data should be cleaned as much as possible before being stored in systems.
2. Entity extraction: Entity extraction refers to the detection of entities in data text. In an LE context, person names, locations, dates and times, and GPS coordinates are entities of interest to the experts. Entity extraction is difficult because the text often has an unsuitable structure, contains spelling errors or is too short.
3. Data linking: The experts described data linking as challenging due to the prevalence of heterogeneous systems that produce data in different formats. Additionally, data from different sources that refer to the

same entity cannot easily be linked when the data does not use common identifiers. The experts wanted to be able to link data that referred to the same entity, and unlink data when links were found to be incorrect.
4. Legislation and policy: There is legislation and policy that controls how LE data is used, and sometimes data sources cannot be combined because legislation or policy prohibits it.

Challenges related to data cleaning, and legislation and policy, were outside the scope of this study. While entity extraction of person names was performed during this research, entity extraction was not a focus of this project, and was not investigated in detail. Approaches for addressing these challenges could nonetheless be explored in future research.

## 5. Discussion

The prototype solution used an ontology-based approach to address the key data integration challenges of NoSQL databases and semantic heterogeneity. The ontology-based approach was able to support the integration of a NoSQL graph database with a relational database. The use of semantic correspondences to link data based on matching person names was a demonstration of the ontology-based approach's ability to overcome semantic heterogeneity. Therefore, the first requirement of the prototype solution, that an ontology-based approach should be considered for addressing heterogeneity, was clearly met.

The second requirement was to minimise manual effort when creating the ontologies. Ontop was used to automatically generate the local ontology from the digital forensics database. As the digital forensics database contained forty-two tables, crafting the ontology by hand would have been tedious. While some manual modification was performed to support the Person Name view, generation significantly reduced the manual effort required to create the ontology. With regards to the POLE database, the local ontology was partly generated using a feature in Neo4j. The class and object property definitions were generated, but the Neo4j feature did not generate the data type property definitions. The data type property definitions had to be added manually; while this was a straightforward process, the solution would ideally allow a user to fully generate the ontology from Neo4j databases. The generation of the global ontology, which was done in [12] and [13], was not investigated in this study.

The third requirement, that the approach should be amenable to changes in the databases, was addressed by implementing the hybrid architecture, an architecture that makes it relatively easy to add or change databases [8]. The use of local ontologies for each database, and a global ontology, in the prototype solution was consistent with the hybrid architecture.

The fourth requirement was to consider using Linked Data practices to build the solution. It was demonstrated that Linked Data practices could be used to implement the ontology-based solution with the desired architecture. Specifically, the global and local ontologies were constructed in OWL, and the semantic correspondences defined using identity and vocabulary links. An RDF database product, Stardog, provided the querying system, which could execute SPARQL queries that combined results from both data sources, while using the OWL ontologies to make inferences.

During the focus group session with the LE data domain experts, the experts described the main challenges they faced with respect to data integration. This study mainly addressed the challenge of linking data that referred to the same entity. It was demonstrated that Linked Data could be used to link data from different sources that referred to the same entity; specifically, the prototype solution was able to link person entities by assigning them URIs and defining identity links. Another challenge was the unlinking of data found to not refer to the same entity. While not investigated in this study, methods have been developed to address the problem of incorrect identity links [34]. Linked Data is thus well suited for addressing the data linking challenges described by the experts.

The experts provided feedback regarding the utility of the proposed approach. With respect to the use of global ontologies and Linked Data, the feedback was positive. Expert A thought that the use of a global ontology to link the data sources was "very useful" for addressing how to combine the data of multiple systems, and that:

> *Principally the idea… presented has obviously got great utility because it's taking out that leg-work of moving between two different systems to generate very similar insights.* (Expert A)

Expert A felt that "certainly… this Linked Data concept is helpful" for linking data based on matching entities like people and locations, which Expert B explained was a common use case in LE.

As for the suitability of the proposed approach for use in an operational setting, Expert B believed that:

> *If you did it as a custom implementation for a specific job, it's relatively simple for us to do something like this… but to make it a sustainable thing, it comes down to also having the systems up to that level… then I think it's got legs.* (Expert B)

In other words, solutions similar to the prototype could be deployed in the AFP for operational use, but a more sustainable approach would be to incorporate the proposed ideas into the organisation's operational systems. However, Expert A noted that, while there was certainly

utility in being able to link data, the real test of a system's utility would be its ability to generate useful insights for end users. The proposed approach would therefore need to support the generation of useful insights for it to be considered suitable for operational use. This requirement was not considered in this study, but could be addressed in any future research that might occur.

The experts suggested several improvements that were outside the scope of this study, but would be interesting topics for future research.

1. *Addition of fuzzy matching:* Fuzzy matching refers to the linking of data that does not exactly match but refer to the same entity. Fuzzy matching can be done in Linked Data by using statistical methods to identify resources that refer to the same entity, and linking the resources with identity links [35].
2. *Support for different entity extraction models:* The ability to swap in suitable entity extraction models for different types of data or different data quality levels was seen as desirable by the experts.
3. *Support for insight generation:* Insight generation involves presenting the integrated data in a way that helps end users to gain useful insights, for example, by using data visualisation. In a layered architecture, insight generation would be a layer somewhere above the data linking layer.

The results of this study have to be seen in light of some limitations. The integration of only two types of databases, relational and graph, was demonstrated. A logical next step would be to demonstrate that the proposed approach can support the integration of other types of databases. In theory, the approach supports any type of database provided its data can be transformed to RDF and a suitable local ontology can be created.

In the prototype solution, only person entities were linked. However, there are many other entities of interest to LEAs, such as locations, dates and times, and GPS coordinates. The linking of various types of entities, in support of common LE use cases, should be explored. Linking entities based on person names that exactly match is useful in an LE context, but is not without its limitations. When a particular person's name appears in different places in text, the way the name is written often varies, due to the inconsistent use of nicknames, diminutives, middle names, initials, and so on. It is in this situation that fuzzy matching could prove useful. Also, person names are not unique, and it can be assumed that many of the identity links created by the prototype solution incorrectly linked resources that did not refer to the same person. The ability to identify and unlink these incorrect identity links would be desirable.

A limitation of the design was the inability to transform the data from the POLE graph database to RDF "on the fly". The data had to be exported from the POLE database as RDF, and stored in a Stardog database. This process would have to be repeated whenever the data in the POLE database was updated, which would be especially disadvantageous in an operational setting if the data was of a dynamic nature, and changed often. As the system design currently relies on Ontop to transform data to RDF "on the fly", but Ontop does not support NoSQL databases, the same limitation applies to all NoSQL databases. A possible solution could be to acquire or develop an OBDA system that supported the NoSQL database. Alternatively, a solution could be implemented to transform data from the NoSQL database into relational data "on the fly", and then Ontop could transform the relational data into RDF.

Another limitation was the need to manually add identity links to the global ontology. If more person entities were added to the POLE data, more identity links would need to be added to the global ontology, and this could become a tedious process over time. A solution could be to develop a system for automatically adding identity links as the POLE data changes.

## 6. Conclusions

The integration of siloed data – particularly in domains such as LE – is often difficult due to heterogeneity of database types and the semantics of the data. In response, an ontology-based approach, implemented using Linked Data practices, is proposed for the data integration of heterogeneous LE databases. To evaluate the utility of the proposed approach, a prototype solution was built, and was able to integrate relational and graph databases that were representative of heterogeneous LE databases. The prototype solution was then evaluated by LE data domain experts from the AFP. The use of an ontology-based approach and Linked Data received positive feedback.

Specifically, the experts believed that the proposed approach could provide benefit if incorporated into their agency's operational systems. However, the experts desired additional capabilities, such as fuzzy matching, entity extraction, and insight generation. It would be important to consider support for these capabilities if the opportunity arose to progress the proposed approach. The prototype solution should be developed further to demonstrate an ability to integrate non-graph NoSQL databases, and to link types of entities other than persons. It would be worthwhile investigating solutions for transforming data from NoSQL databases to RDF "on the fly", and automatically adding identity links to the global ontology, in future research.

## Acknowledgement

This report was originally submitted as a thesis paper for the degree of Bachelor of of Data Science (Honours) at

Monash University. The supervisors were Associate Professor Campbell Wilson (Director AiLECS Lab), and Dr Gregory Rolan (Research Fellow, AiLECS Lab).

## 7. References


[1] M. Stonebraker and I. F. Ilyas, 'Data Integration: The Current Status and the Way Forward.', IEEE Data Eng. Bull., vol. 41, no. 2, pp. 3–9, 2018.

[2] D. Chen and N. Daclin, 'Framework for enterprise interoperability', in Interoperability for Enterprise Software and Applications: Proceedings of the Workshops and the Doctorial Symposium of the Second IFAC/IFIP I-ESA International Conference: EI2N, WSI, IS-TSPQ 2006, 2006, pp. 77–88.

[3] R. Kozik et al., 'The Identification and creation of ontologies for the use in law enforcement AI solutions–MAGNETO platform use case', in International Conference on Computational Collective Intelligence, Sep. 2019, pp. 335–345.

[4] E. Casey, S. Barnum, R. Griffith, J. Snyder, H. van Beek, and A. Nelson, 'Advancing coordinated cyber-investigations and tool interoperability using a community developed specification language', Digital investigation, vol. 22, pp. 14–45, 2017.

[5] M. Lenzerini, 'Data integration: A theoretical perspective', in Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Jun. 2002, pp. 233–246.

[6] R. Hull, 'Managing semantic heterogeneity in databases: a theoretical prospective', in Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, May 1997, pp. 51–61.

[7] A. M. Ouksel and A. Sheth, 'Semantic interoperability in global information systems', ACM Sigmod Record, vol. 28, no. 1, pp. 5–12, 1999.

[8] H. Wache et al., "Ontology-Based Integration of Information – A Survey of Existing Approaches," Ois@ ijcai, Jan. 2001.

[9] R. Lawrence, 'Integration and virtualization of relational SQL and NoSQL systems including MySQL and MongoDB', in 2014 International Conference on Computational Science and Computational Intelligence, Mar. 2014, vol. 1, pp. 285–290.

[10] A. Corbellini, C. Mateos, A. Zunino, D. Godoy, and S. Schiaffino, 'Persisting big-data: The NoSQL landscape', Information Systems, vol. 63, pp. 1–23, 2017.

[11] A. Guarino, 'Digital forensics as a big data challenge', in ISSE 2013 securing electronic business processes, Springer, 2013, pp. 197–203.

[12] O. Curé, M. Lamolle, and C. L. Duc, 'Ontology based data integration over document and column family oriented NOSQL', arXiv preprint arXiv:1307.2603, 2013.

[13] H. Abbes and F. Gargouri, 'Big data integration: A MongoDB database and modular ontologies based approach', Procedia Computer Science, vol. 96, pp. 446–455, 2016.

[14] T. Heath and C. Bizer, 'Linked data: Evolving the web into a global data space', Synthesis lectures on the semantic web: theory and technology, vol. 1, no. 1, pp. 1–136, 2011.

[15] M. Hert, G. Reif, and H. C. Gall, 'A comparison of RDB-to-RDF mapping languages', in Proceedings of the 7th international conference on semantic systems, 2011, pp. 25–32.

[16] G. Rocher, J.-Y. Tigli, S. Lavirotte, and R. Daikhi, 'Leveraging ambient applications interactions with their environment to improve services selection relevancy', Univeristé Nice Sophia Antipolis, 2015.

[17] O. Corcho, M. Poveda-Villalón, and A. Gómez-Pérez, 'Ontology engineering in the era of linked data', Bulletin of the Association for Information Science and Technology, vol. 41, no. 4, pp. 13–17, 2015.

[18] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, 'OWL 2 web ontology language primer', W3C recommendation, vol. 27, no. 1, p. 123, 2009.

[19] H. Fani and E. Bagheri, 'An Ontology for Describing Security Events.', in SEKE, 2015, pp. 455–460.

[20] B. Schatz, G. Mohay, and A. Clark, 'Generalising event forensics across multiple domains', in 2nd Australian Computer Networks Information and Forensics Conference, 2004, pp. 136–144.

[21] B. Schatz and A. Clark, 'An open architecture for digital evidence integration', in Proceedings of the AusCERT Asia Pacific Information Technology Security Conference 2006, 2006, pp. 15–29.

[22] H. Park, S. Cho, and H.-C. Kwon, 'Cyber forensics ontology for cyber criminal investigation', in International Conference on Forensics in Telecommunications, Information, and Multimedia, 2009, pp. 160–165.

[23] A. M. Talib and F. O. Alomary, 'Towards a comprehensive ontology based-investigation for digital forensics cybercrime', Int J Commun Antenna Propag, vol. 5, no. 5, pp. 263–268, 2015.

[24] J. González-Conejero, R. V. Figueroa, J. Munoz-Gomez, and E. Teodoro, 'Organized crime structure modelling for european law enforcement agencies interoperability through ontologies', in International Workshop on AI Approaches to the Complexity of Legal Systems, 2013, pp. 217–231.

[25] A. R. Hevner, S. T. March, J. Park, and S. Ram, 'Design science in information systems research', MIS quarterly, pp. 75–105, 2004.

[26] J. F. Nunamaker Jr, M. Chen, and T. D. Purdin, 'Systems development in information systems research', Journal of management information systems, vol. 7, no. 3, pp. 89–106, 1990.

[27] J. Guia, V. G. Soares, and J. Bernardino, 'Graph Databases: Neo4j Analysis.', in ICEIS (1), 2017, pp. 351–356.

[28] R. Angles and C. Gutierrez, 'Survey of graph database models', ACM Computing Surveys (CSUR), vol. 40, no. 1, pp. 1–39, 2008.

[29] D. Calvanese et al., 'Ontop: Answering SPARQL queries over relational databases', Semantic Web, vol. 8, no. 3, pp. 471–487, 2017.

[30] B. Klimt and Y. Yang, 'Introducing the Enron corpus.', in CEAS, 2004.

[31] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, 'The Stanford CoreNLP natural language processing toolkit', in Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, Jun. 2014, pp. 55–60.

[32] M. Arenas, A. Bertails, E. Prud'hommeaux, and J. Sequeda, 'A direct mapping of relational data to RDF', W3C recommendation, vol. 27, pp. 1–11, 2012.

[33] M. C. Tremblay, A. R. Hevner, and D. J. Berndt, 'Focus groups for artifact refinement and evaluation in design research', Communications of the association for information systems, vol. 26, no. 1, p. 27, 2010.

[34] L. Papaleo, N. Pernelle, F. Saïs, and C. Dumont, 'Logical detection of invalid sameas statements in RDF data', in International conference on knowledge engineering and knowledge management, 2014, pp. 373–384.

[35] A. Hogan, A. Polleres, J. Umbrich, and A. Zimmermann, 'Some entities are more equal than others: statistical methods to consolidate linked data', in 4th International Workshop on New Forms of Reasoning for the Semantic Web: Scalable and Dynamic (NeFoRS2010), May 2010.

[36] G. C. van Erven, M. Holanda, and R. N. Carvalho, 'Detecting evidence of fraud in the brazilian government using graph databases', in World conference on information systems and technologies, Apr. 2017, pp. 464–473.

[37] 'Graph Database Platform | Graph Database Management System | Neo4j'. https://neo4j.com/ (accessed Oct. 14, 2021).

[38] ‘Ontop’. https://ontop-vkg.org/ (accessed Oct. 14, 2021).

[39] ‘Introduction | Ontop’. https://ontop-vkg.org/guide/ (accessed Oct. 14, 2021).

[40] ‘The Enterprise Knowledge Graph Platform | Stardog’. https://www.stardog.com/ (accessed Oct. 14, 2021).

[41] ‘The Sleuth Kit (TSK) & Autopsy: Open Source Digital Forensics Tools’. https://sleuthkit.org/ (accessed Oct. 14, 2021).

[42] ‘Autopsy User Documentation: Ingest Modules’. https://sleuthkit.org/autopsy/docs/user-docs/4.8.0/ingest_page.html (accessed Oct. 14, 2021).

[43] ‘Autopsy: Java Development Setup’. https://www.sleuthkit.org/autopsy/docs/api-docs/4.8.0/mod_dev_page.html (accessed Oct. 14, 2021).

[44] ‘SQLite Home Page’. https://www.sqlite.org/index.html (accessed Oct. 14, 2021). [45] ‘neo4j-graph-examples/pole’, Sep. 30, 2021. https://github.com/ neo4j-graph-examples/pole (accessed Oct. 14, 2021).

[46] T. L. Saito, SQLite JDBC Driver. 2021. Accessed: Oct. 14, 2021. [Online]. Available: https://github.com/xerial/sqlite-jdbc

[47] W. Mayer, M. Stumptner, P. Casanovas, and L. de Koker, ‘Towards a linked information architecture for integrated law enforcement’, 2017.

[48] ‘The most popular database for modern apps’, MongoDB. https://www.mongodb.com (accessed Oct. 18, 2021).

[49] B. L. Leech, ‘Asking questions: Techniques for semistructured interviews’, PS: Political Science & Politics, vol. 35, no. 4, pp. 665–668, 2002.

[50] ‘Apache OpenNLP’. http://opennlp.apache.org/ (accessed Oct. 21, 2021).

[51] H. Shelar, G. Kaur, N. Heda, and P. Agrawal, ‘Named entity recognition approaches and their comparison for custom ner model’, Science & Technology Libraries, vol. 39, no. 3, pp. 324–337, 2020.

 