



**NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS**

**SCHOOL OF SCIENCE**

**DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS**

**BACHELOR THESIS**

**A Web-Based Linked Data Application for  
Visualizing Historical United States of America Boundary  
Changes**

**Fotios G Vorloou**

**Supervisor:** Manolis Koubarakis, Professor  
**Co-Supervisor:** Georgios Stamoulis, Ph.D. Candidate

**ATHENS  
SEPTEMBER 2024**



**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Μια διαδικτυακή εφαρμογή διασυνδεδεμένων δεδομένων για  
Οπτικοποίηση ιστορικών αλλαγών των συνόρων των  
Ηνωμένων Πολιτειών της Αμερικής**

**Φώτιος Γ Βορλόου**

**Επιβλέπων: Μανόλης Κουμπάρακης, Καθηγητής  
Συνεπιβλέπων: Γεώργιος Σταμούλης, Υποψήφιος Διδάκτωρ**

**ΑΘΗΝΑ**

**ΣΕΠΤΕΜΒΡΙΟΣ 2024**

## **BACHELOR THESIS**

A Web-Based Linked Data Application for  
Visualizing Historical Boundary Changes

**Fotios G Vorloou**  
**R.N.: 1115201900026**

**Supervisor:**        **Manolis Koubarakis, Professor**  
**Co-Supervisor:**   **Georgios Stamoulis, Ph.D. Candidate**

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Μια διαδικτυακή εφαρμογή διασυνδεδεμένων δεδομένων για  
Οπτικοποίηση ιστορικών αλλαγών των συνόρων των Ηνωμένων Πολιτειών της  
Αμερικής

**Φώτιος Γ Βορλόου**

**A.M.: 1115201900026**

**Επιβλέπων:** Μανόλης Κουμπάρακης, Καθηγητής  
**Συνεπιβλέπων:** Γεώργιος Σταμούλης, Υποψήφιος Διδάκτωρ

## ΠΕΡΙΛΗΨΗ

Στην ψηφιακή εποχή, η διαχείριση και η ανάλυση ιστορικών γεωχωρικών δεδομένων αποκτούν ολοένα και μεγαλύτερη σημασία για την κατανόηση της εξέλιξης των γεωπολιτικών συνόρων με την πάροδο του χρόνου. Τα παραδοσιακά συστήματα γεωγραφικών πληροφοριών συχνά δυσκολεύονται με τη δυναμική φύση των ιστορικών δεδομένων, η οποία περιλαμβάνει αλλαγές στα εδάφη, τα διοικητικά όρια και τα τοπωνύμια. Οι τεχνολογίες συνδεδεμένων δεδομένων, που βασίζονται στις αρχές του Σημασιολογικού Ιστού, προσφέρουν μια ισχυρή λύση σε αυτές τις προκλήσεις, επιτρέποντας την ενσωμάτωση, την αναζήτηση και την οπτικοποίηση διαφορετικών συνόλων δεδομένων. Οι τεχνολογίες αυτές διευκολύνουν την απρόσκοπτη σύνδεση γεωγραφικών και χρονικών πληροφοριών, επιτρέποντας πιο σφαιρικές αναλύσεις και γνώσεις.

Η παρούσα πτυχιακή εργασία παρουσιάζει μια διαδικτυακή εφαρμογή για την ανάλυση και οπτικοποίηση ιστορικών αλλαγών στα όρια των κομητειών και των πολιτειών στις Ηνωμένες Πολιτείες Αμερικής με τη χρήση τεχνολογιών συνδεδεμένων δεδομένων. Αξιοποιώντας το σύνολο δεδομένων του GeoChangesQA, το οποίο είναι προσβάσιμο μέσω ενός endpoint GraphDB, η εφαρμογή επιτρέπει στους χρήστες να διερευνήσουν τις γεωχωρικές και χρονικές μεταβολές των γεωπολιτικών συνόρων. Οι βασικές λειτουργίες περιλαμβάνουν την αναζήτηση και την οπτικοποίηση των αλλαγών των συνόρων με την πάροδο του χρόνου, με τη δυνατότητα σύγκρισης προηγούμενων εκδόσεων συνόρων. Η εφαρμογή χρησιμοποιεί RDF για την αναπαράσταση των δεδομένων και SPARQL για την υποβολή ερωτημάτων στη βάση, το backend έχει υλοποιηθεί με Spring Boot, το frontend με Angular και για τους χάρτες χρησιμοποιήθηκε Leaflet, παρέχοντας μια διαδραστική εμπειρία χρήστη. Στόχος της εφαρμογής είναι να παρέχει ένα ισχυρό και φιλικό προς το χρήστη εργαλείο που δίνει τη δυνατότητα στους χρήστες να εξερευνήσουν και να κατανοήσουν την ιστορική εξέλιξη των συνόρων των κομητειών και των πολιτειών των Ηνωμένων Πολιτειών Αμερικής (ΗΠΑ) με ευκολία.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ:** Συστήματα γεωχωρικών πληροφοριών, ανάλυση ιστορικών δεδομένων, συνδεδεμένα δεδομένα.

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** Γεωχωρική οπτικοποίηση, ιστορικές αλλαγές ορίων, συνδεδεμένα δεδομένα, RDF, SPARQL, GraphDB.

## **ABSTRACT**

In the digital era, the management and analysis of historical geospatial data have become increasingly important for understanding the evolution of geopolitical boundaries over time. Traditional geographic information systems often struggle with the dynamic nature of historical data, which involves changes in territorial divisions, administrative boundaries, and place names. Linked data technologies, grounded in the principles of the Semantic Web, offer a powerful solution to these challenges by enabling the integration, querying, and visualization of diverse datasets. These technologies facilitate the seamless connection of spatial and temporal information, allowing for more comprehensive analyses and insights.

This thesis presents a web-based application for analyzing and visualizing historical boundary changes of counties and states in the United States of America using linked data technologies. Leveraging the GeoChangesQA dataset accessible via a GraphDB endpoint, the application enables users to explore geospatial and temporal shifts in geopolitical boundaries. The core functionalities include querying and visualizing boundary changes over time, with the capability to compare previous boundary versions. The application uses RDF for data representation and SPARQL for querying, integrated with Spring Boot for the backend, Angular for the frontend and Leaflet for map rendering, providing an interactive user experience. The goal of the application is to provide a robust and user-friendly tool that empowers users to explore and understand the historical evolution of United States of America county and state boundaries with ease.

**SUBJECT AREA:** Geospatial Information Systems (GIS), Historical Data Analysis, Linked Data.

**KEYWORDS:** Geospatial Visualization, Historical Boundary Changes, Linked Data, RDF, SPARQL, GraphDB.

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να ευχαριστήσω τους επιβλέποντες μου σε αυτή την εργασία, τον κύριο καθηγητή Μανόλη Κουμπάρακη για την ευκαιρία που μου έδωσε και τον κύριο Γεώργιο Σταμούλη για την πολύτιμη καθοδήγησή του. Επίσης, ευχαριστώ την οικογένεια μου για την στήριξη που μου παρείχε σε όλη τη διάρκεια εκπόνησης της πτυχιακής μου εργασίας καθώς και τους φίλους μου για την βοήθεια και την ενθάρρυνσή τους.

# Contents

<b>CONTENTS.....</b>	<b>8</b>
<b>1. INTRODUCTION.....</b>	<b>10</b>
<b>2. RELATED WORK.....</b>	<b>11</b>
2.1 Linked Data .....	11
2.2 The GeoChangesQA Dataset .....	12
2.3 Historical Geospatial Data Integration .....	13
2.4 Technologies Background .....	14
2.5 Visualizing Historical Geospatial Data .....	15
<b>3. APPLICATION.....</b>	<b>16</b>
3.1 Role.....	16
3.2 Architecture .....	17
3.3 Technologies .....	19
3.4 User Interface.....	23
3.4.1 The search bar .....	23
3.4.2 The tree menu .....	24
3.4.3 The slider and the calendar.....	24
3.4.4 The maps .....	25
3.4.5 Metadata table .....	25
<b>4. FUTURE WORK .....</b>	<b>27</b>
<b>5. CONCLUSION .....</b>	<b>28</b>
<b>ABBREVIATIONS - DIACRITICS - ACRONYMS .....</b>	<b>29</b>
<b>REFERENCES.....</b>	<b>30</b>



## List of Figures

FIGURE 1: THE APPLICATION'S ARCHITECTURE	18
FIGURE 2 - THE CLASS RELATIONSHIPS	19
FIGURE 3: A VISUAL EXAMPLE OF A PART OF THE RDF GRAPH	20
FIGURE 4: AN OVERVIEW OF THE USER INTERFACE	23
FIGURE 5 - THE SEARCH BAR WITH SUGGESTIONS	23
FIGURE 6 - THE TREE MENU	24
FIGURE 7 - THE SLIDER AND THE CALENDAR IN THEIR RESPECTIVE POSITIONS	25
FIGURE 8 - THE MAPS WITH THE METADATA TABLE SHOWING THE RESULTS OF A QUERY	26

## 1. INTRODUCTION

In the realm of data management and analysis, the integration and interoperability of diverse datasets have long posed significant challenges. Traditional data systems often operate in silos, making it difficult to combine and query data from different sources. This fragmentation leads to inefficiencies and limits the potential for comprehensive analysis and decision-making. Linked data technologies, which leverage the principles of the Semantic Web, have emerged as a powerful solution to these challenges.

In recent years, the advent of linked data technologies has revolutionized the way we manage and analyze historical geospatial data. Linked data principles allow for the integration and querying of diverse datasets using standard web technologies, thereby enabling more comprehensive and flexible analyses. Historical geospatial data, which captures changes in geographic boundaries, presents unique challenges and opportunities for researchers and practitioners.

The need to effectively manage and query historical geospatial data has led to the development of specialized ontologies and knowledge graphs. The Historical County Boundaries Ontology (HCB-O) and the Historical County Boundaries Knowledge Graph (HCB-KG) [1] represent significant advancements in this field.

Linked data principles provide a framework for connecting and querying diverse datasets using standard web technologies. In the context of historical geospatial data, linked data enables the integration of spatial and temporal information from various sources, facilitating more comprehensive and flexible analyses.

Ontologies such as the Territorial Statistical Nomenclature (TSN) and TSN-Change [1], along with the Simple Event Model (SEM) [2] and the Time Ontology [3], play a crucial role in structuring and querying historical geospatial data. The TSN and TSN-Change ontologies model territorial partitions and their changes over time, while SEM and the Time Ontology provide a framework for representing events and temporal intervals .

Building on these foundational ontologies and knowledge graphs, this thesis presents a web-based application that offers an intuitive and powerful platform for visualizing the historical changes in U.S. county and state boundaries from 1629 to 2000. The application leverages the linked data principles and the rich structure provided by the Historical County Boundaries Ontology (HCB-O) to integrate and query complex geospatial and temporal data seamlessly.

Through interactive maps and detailed metadata displays, users can explore the evolution of geographic boundaries across different time periods, compare historical changes side by side, and gain insights into the historical context of these changes. The application is designed to be user-friendly, making it accessible to the general public interested in the historical geography of the United States. By providing a dynamic and interactive way to engage with historical geospatial data, this application aims to be a significant advancement in the exploration and analysis of U.S. boundary changes over time.

## 2. RELATED WORK

### 2.1 Linked Data

Linked data represents a significant shift in how information is interconnected and accessed on the web, revolutionizing data integration and query processes across different domains. This approach is deeply rooted in the principles proposed by Tim Berners-Lee [16], who envisioned a web of data that is easily linked and accessed using standard web technologies. The core technologies that enable linked data include the Resource Description Framework (RDF) [4], SPARQL [5], ontologies, and Uniform Resource Identifiers (URIs). Each of these components plays a crucial role in the creation, management, and utilization of linked data, contributing to a more interconnected and semantically rich web of information.

The Resource Description Framework (RDF) is the foundational technology of linked data. RDF provides a standard model for data interchange on the web, allowing data to be represented as triples—consisting of a subject, predicate, and object. These triples form a directed graph where the subject is connected to the object by the predicate, effectively describing a relationship between two resources. This graph-based approach is highly flexible and can represent complex data relationships across diverse domains.

RDF's flexibility stems from its ability to represent data in a way that is both machine-readable and easily linked to other data sources. This is achieved through the use of URIs for naming each element in the triple, ensuring that data elements are uniquely identified and globally accessible. RDF can be serialized in multiple formats, including RDF/XML, Turtle, and JSON-LD, making it adaptable to various use cases and environments. The widespread adoption of RDF has made it a cornerstone of the Semantic Web, enabling the creation of rich, interconnected datasets that can be queried and analyzed in a unified manner.

SPARQL (SPARQL Protocol and RDF Query Language) is a powerful query language designed for retrieving and manipulating RDF data, much like SQL for relational databases. It supports various query types, including SELECT, CONSTRUCT, and ASK, enabling complex data retrieval from interconnected datasets. SPARQL's flexibility in handling patterns within RDF data allows for precise and targeted queries, making it particularly effective in linked data environments where data from multiple sources must be integrated and analyzed. Its capabilities extend to filtering, aggregation, and federated queries, which are crucial for deriving meaningful insights from large, diverse datasets.

In geospatial contexts, GeoSPARQL [6] extends SPARQL by adding support for querying geospatial data, making it a critical tool for applications involving spatial data. Developed by the Open Geospatial Consortium (OGC), GeoSPARQL introduces spatial predicates and functions that enable the querying of spatial relationships, geospatial data types, and spatial functions. This extension is particularly relevant for the thesis at hand, as it is used to query historical geospatial data to visualize and analyze changes in U.S. county and state boundaries over time. The combination of SPARQL's flexibility and GeoSPARQL's spatial querying capabilities provides a robust framework for managing and exploring historical geospatial data, enabling complex analyses that integrate both spatial and temporal dimensions.

Ontologies play a critical role in the linked data ecosystem by providing a formal specification of the relationships between different data entities within a specific domain. They define the structure and semantics of data, creating a shared vocabulary that ensures data consistency and interoperability across diverse datasets. Ontologies are

essential for enabling meaningful data integration, as they allow different datasets to be understood and used together in a coherent manner.

In the context of linked data, ontologies are used to describe the types of entities in a dataset and the relationships between them. For example, an ontology for a bibliographic dataset might define classes such as "Book," "Author," and "Publisher," along with properties that describe how these entities are related (e.g., "writtenBy," "publishedBy"). By providing a clear and unambiguous definition of these concepts, ontologies enable data from different sources to be linked and queried in a unified way. OWL (Web Ontology Language) [3] is a commonly used language for defining ontologies, offering a rich set of constructs for modeling complex relationships and constraints.

Uniform Resource Identifiers (URIs) are the building blocks of linked data, providing a standardized way to identify and access resources on the web. In the linked data context, URIs are used to uniquely identify data elements, such as entities, properties, and relationships. This global identification system is crucial for ensuring that data can be linked across different sources, enabling the creation of a truly interconnected web of data.

URIs serve as the foundation for linking data, as they allow different datasets to refer to the same entities using a common identifier. For example, a URI might be used to uniquely identify a specific person, location, or concept, making it possible to link information about that entity from different sources. The use of URIs ensures that data remains unambiguous and interoperable, facilitating the integration and analysis of data from multiple domains. Additionally, URIs can be dereferenced, meaning they can be used to retrieve additional information about the resource they identify, further enhancing the linked data experience.

In conclusion, linked data technologies, including RDF, SPARQL, ontologies, and URIs, provide the necessary infrastructure for creating a web of interconnected data. These technologies enable the seamless integration of data from diverse sources, allowing for more sophisticated and flexible data analysis. By leveraging these technologies, linked data has the potential to transform how we manage, share, and analyze data across the web, paving the way for new applications and insights.

## 2.2 The GeoChangesQA Dataset

The GeoChangesQA dataset [1] is a novel and extensive collection of historical geospatial knowledge, specifically focusing on the spatiotemporal dynamics of U.S. county boundaries from 1629 to 2000. The dataset was meticulously developed as part of the efforts to support spatiotemporal question answering systems, which require the integration of both geographical and temporal dimensions in a coherent manner.

The dataset itself is built upon the Historical County Boundaries Knowledge Graph (HCB-KG), which was constructed using data from the Atlas of Historical County Boundaries (AHCB). This dataset includes not only the geometrical representations of U.S. counties but also the historical changes that these counties underwent, such as boundary shifts, mergers, and name changes.

To create the HCB-KG, the developers utilized the Historical County Boundaries Ontology (HCB-O), an extension of the Territorial Statistical Nomenclature Ontology (TSN) and the TSN-Change Ontology. These resources model the changes in U.S. county and state boundaries from 1629 to 2000, providing a rich dataset for exploring historical geospatial dynamics and are essential for modeling territorial partitions and

their evolution over time, providing a robust framework for understanding how historical changes impact current geographical and political landscapes.

The GeoChangesQA dataset is particularly important in the context of linked data and the Semantic Web because it addresses a key challenge in geospatial knowledge representation: the simultaneous handling of spatial and temporal dimensions. Traditional knowledge graphs often treat spatial and temporal attributes in isolation, which limits their ability to represent historical dynamics accurately. In contrast, GeoChangesQA integrates these dimensions, allowing for more sophisticated querying and analysis.

In this thesis, the GeoChangesQA dataset plays a central role in the development and evaluation of a web-based application that visualizes the historical changes in U.S. county boundaries. By leveraging the rich, spatiotemporal data in the GeoChangesQA dataset, the application is able to provide users with an interactive and informative tool for exploring the history of U.S. states and counties, demonstrating the practical utility of linked data technologies in historical contexts.

### **2.3 Historical Geospatial Data Integration**

Historical geospatial data presents distinct challenges for integration, analysis, and visualization, given its inherently dynamic nature. Unlike contemporary datasets, historical geospatial data captures the evolution of geographic entities over time, necessitating sophisticated methods to simultaneously manage both temporal and spatial dimensions. The key challenges associated with historical geospatial data include the following:

One of the most complex aspects of historical geospatial data is its temporal dimension. Over time, geographic entities such as countries, states, or counties undergo changes in their boundaries, names, and administrative hierarchies. For instance, the boundaries of U.S. counties have changed multiple times since their inception, influenced by political, social, and legal developments. Accurately representing these changes over time requires data models that can capture and query these temporal dynamics effectively. Without such models, it is challenging to provide accurate historical analyses or visualize the geographic evolution of regions. Temporal data models must accommodate the fact that the same geographic location may have different attributes or boundaries at different points in time.

Historical geospatial data is often sourced from a wide variety of archives, each utilizing different formats, standards, and conventions. These sources may include historical maps, governmental records, and scholarly databases, all of which may represent geographic information in different ways. The challenge of data interoperability involves ensuring that these diverse sources can be harmonized and integrated into a single, coherent dataset that can be easily queried and analyzed. Achieving interoperability is essential for comprehensive analysis, as it allows researchers to combine and compare data from different times and places, providing a more complete picture of historical geographic changes.

The volume and complexity of historical geospatial data present significant challenges in terms of scalability. Large datasets that span centuries and cover extensive geographic areas require scalable methods for storage, querying, and visualization. Linked data technologies, with their emphasis on interconnected datasets and flexible querying capabilities, provide a robust framework for managing the scalability challenges inherent in historical geospatial data. These technologies enable efficient

handling of large datasets, ensuring that even complex queries can be executed quickly and that the data can be visualized in an interactive, user-friendly manner.

To address these challenges, linked data technologies offer powerful tools for integrating, managing, and analyzing historical geospatial data. By employing Resource Description Framework (RDF), SPARQL, and ontologies, linked data allows for the seamless combination of data from disparate sources, the representation of temporal dynamics, and the scalability needed to handle large datasets. These technologies are crucial for enabling the exploration and visualization of historical geospatial data, providing new insights into how geographic regions have evolved over time.

## 2.4 Technologies Background

The application developed in this thesis leverages a range of advanced technologies to effectively manage, query, and visualize linked data, particularly historical geospatial data. Each technology was chosen for its specific strengths and compatibility with the overall system architecture. Below is an overview of the key technologies used:

GraphDB [7] is a high-performance RDF database and query engine designed for handling large volumes of linked data. It is particularly well-suited for applications that require complex SPARQL queries and efficient data retrieval. GraphDB supports RDF-based storage, enabling the application to manage intricate relationships between historical geospatial data points. Its built-in reasoning capabilities allow for the inference of new data from existing datasets, enhancing the depth of analysis possible within the application. Additionally, GraphDB provides comprehensive tools for managing and visualizing RDF graphs, making it an essential component for querying and integrating the historical data used in this project.

Leaflet [10] is an open-source JavaScript library that offers a powerful and flexible framework for rendering interactive maps. It is lightweight, making it ideal for web applications that require responsive and dynamic geospatial visualizations. In this application, Leaflet is used to display historical boundaries of U.S. counties and states, allowing users to interact with the maps and explore changes over time. The library supports various mapping features such as zooming, panning, and layer toggling, which are crucial for providing an intuitive user experience. Moreover, Leaflet's extensive plugin ecosystem enables the addition of custom functionalities, such as geospatial data overlays and advanced map controls, further enhancing the application's capabilities.

Angular [9] is a widely-used web application framework developed by Google. It is designed for building dynamic and responsive single-page applications (SPAs). Angular's component-based architecture facilitates the development of modular, maintainable, and scalable user interfaces. In this project, Angular is employed to construct the front-end of the application, enabling users to interact seamlessly with the geospatial data. The framework's two-way data binding ensures that the user interface reflects real-time updates, while its dependency injection and services provide a clean separation between the application's logic and its presentation layer. Angular's extensive library of tools and components accelerates development and enhances the overall user experience by providing smooth transitions, responsive design, and robust client-side routing.

Spring Boot [8] is an open-source framework that simplifies the development of Java-based enterprise applications. It provides a convention-over-configuration approach, allowing developers to create stand-alone, production-ready applications with minimal setup. In this thesis, Spring Boot is used to develop the backend services that handle

data retrieval, processing, and communication with the GraphDB database. The framework's built-in support for RESTful web services enables efficient API creation, which is crucial for facilitating interactions between the front-end Angular application and the backend database. Spring Boot's robust ecosystem also includes tools for security, data access, and monitoring, ensuring that the backend services are secure, scalable, and maintainable.

Each of these technologies plays a vital role in the overall architecture of the application, contributing to its ability to efficiently manage and visualize complex historical geospatial data. The integration of these tools results in a cohesive system that offers both robust backend processing capabilities and a user-friendly interface for exploring the dynamic history of U.S. county and state boundaries.

## **2.5 Visualizing Historical Geospatial Data**

The visualization of historical geospatial data is crucial for understanding the evolution of geographic boundaries and territories over time. Several projects have made significant contributions in this area by providing tools and platforms that allow users to explore historical maps and boundary changes interactively. Among these, Pelagios [14] and Pleiades [15] stand out for their innovative approaches to linking, visualizing, and analyzing historical geospatial data.

Pelagios is a collaborative project that aims to connect online historical resources through geographic annotations. It focuses on linking various datasets that refer to historical places, enabling users to explore the ancient world through a network of interconnected resources. The strength of Pelagios lies in its ability to provide context to historical places by linking them with other relevant datasets, offering a rich and interconnected view of the ancient world. This project has been particularly valuable for researchers in archaeology and ancient history, providing a comprehensive framework for exploring the spatial relationships between historical entities. However, Pelagios is primarily focused on ancient geography and may not be as versatile for users interested in more recent historical periods.

Pleiades is an online gazetteer and a community-built resource for ancient geography. It provides detailed information about ancient places, including their locations, names, and relationships to other places. Unlike Pelagios, which focuses on linking datasets, Pleiades offers a more static but highly detailed catalog of ancient places. It allows users to explore ancient geography through a comprehensive database that is widely used in digital humanities and classics. Pleiades has been instrumental in standardizing place names and locations in ancient geography, making it a crucial resource for researchers. However, its focus on static data rather than dynamic visualization means it lacks the interactive features found in some other tools.

The application developed in this thesis draws upon the strengths of established tools like Pelagios and Pleiades while introducing the distinct advantage of using RDF graphs for data integration and querying. Unlike Pelagios, which connects historical datasets across a wide range of sources, our application focuses specifically on U.S. county and state boundaries, offering a tailored experience that benefits from the structured, standardized representation of data provided by RDF. Similar to how Pelagios links data to provide context, our application allows users to visualize the evolution of geographical boundaries over time, but with the added depth of RDF-based queries, enabling more complex data interactions. While Pleiades is unmatched in its detailed cataloging of ancient places, our application differentiates itself by using RDF to ensure high interoperability and consistency across datasets, making it a powerful, though

more focused, tool for exploring historical geospatial data. By combining the best features of these tools with the unique capabilities of RDF graphs, this application provides a complementary resource that enhances the exploration and analysis of historical geography, particularly within the context of U.S. history.

### 3. APPLICATION

#### 3.1 Role

The objective of this thesis is to design and develop a user-friendly application that effectively leverages RDF data from the comprehensive GeoChanges dataset, showcasing the historical evolution of U.S. county and state boundaries over time. This application demonstrates how linked data can be utilized to manage, query, and visualize complex geospatial and temporal datasets. By doing so, it provides a practical and interactive tool for users to explore the changes in geographic boundaries across different time periods. The application is intended to bridge the gap between complex RDF databases and accessible, everyday use cases, offering a simple yet powerful solution for visualizing linked historical geospatial data. The broader goal is to illustrate how such data can be transformed into actionable insights through an intuitive interface, thereby enhancing the understanding of historical geographic changes.

The application aims to provide an accessible tool for exploring and understanding the dynamic nature of U.S. county and state borders over time. By making use of RDF data from the extensive GeoChanges dataset, the application enables users to visualize how geographic boundaries have shifted across different periods. This involves not only displaying the current boundaries for a given date but also allowing users to compare them with previous versions, highlighting the changes that have occurred. Through this, the application serves as a practical demonstration of how linked data can be employed in a real-world context, making complex historical geospatial data more accessible and engaging.

The key role of this application is to offer an interactive, web-based platform where users can select specific states or counties, choose a date range, and view the corresponding boundary changes on an interactive map. The application handles the complexities of querying an RDF GeoSPARQL database, processing the returned data, and presenting it in a user-friendly manner. This not only enhances the user's understanding of historical geography but also showcases the practical applications of linked data in historical research and education. By transforming RDF data into an intuitive visual format, the application makes it easier for users to engage with and explore historical geospatial information, providing valuable insights into the changes that have shaped the United States' geographic landscape.

The application has been designed with accessibility and transparency in mind, allowing anyone to explore and understand historical changes in U.S. county and state boundaries. To further contribute to the open data and open-source community, the full source code of the application is available for public access. Developers, researchers, and enthusiasts can view, modify, or contribute to the project by accessing it on GitHub. This initiative aims to encourage collaborative development, further improvements, and adoption of linked data technologies in geospatial visualization. The codebase can be found at <https://github.com/fotisvor/Linked-Data-Visualization>.



### 3.2 Architecture

The architecture of the application is meticulously designed to handle the complexities of historical geospatial data while ensuring responsiveness and scalability. At its core, the system consists of three main components: the RDF endpoint, the backend powered by Spring Boot, and the Angular-based frontend.

The RDF endpoint is the central repository where the linked data is stored. This repository is built using the GeoChanges dataset, which is integrated into an RDF graph and organized into a structured graph using RDF standards. Each piece of data in the graph is represented by a subject, a predicate, and an object, following the principles of the Semantic Web.

This structure allows for the rich interlinking of data points, facilitating complex queries that can extract meaningful information about the historical changes in county and state boundaries. The RDF endpoint serves as the foundational layer of the application, providing a robust and scalable platform for managing and querying linked data.

This data repository is directly connected to the backend of the application, which is developed using Spring Boot, a powerful Java framework known for its ability to create standalone, production-ready applications. The Spring Boot backend acts as an intermediary between the RDF endpoint and the frontend, orchestrating the flow of data through the system. It interacts directly with the RDF endpoint through the Eclipse GraphDB API, establishing a secure and efficient connection for querying the database. When a user makes a request via the frontend, the backend processes the request by generating a SPARQL query tailored to the specific parameters provided by the user, such as the name of the location, the type of location (state or county), and the desired date range. This query is then executed against the RDF database, and the resulting data is processed by the backend.

For geospatial data, the backend performs a conversion from Well-Known Text (WKT) to GeoJSON, a format that is more easily handled by the frontend's mapping library. This conversion step is crucial as it ensures that the geospatial data is in a format that can be efficiently rendered on a map in the user's browser. By offloading this processing to the backend, the application minimizes the computational load on the client side, resulting in faster response times and a smoother user experience.

The frontend of the application is built using Angular, a popular framework for developing dynamic web applications. Angular is responsible for rendering the user interface, managing user interactions, and displaying the results of the backend queries. The interface is designed to be intuitive and user-friendly, with features such as a search bar, a tree menu, and a slider to help users easily select locations and dates. When a user submits a query, the Angular app sends a request to the backend, receives the processed data, and updates the map and metadata display accordingly. The Angular frontend and Spring Boot backend communicate through a RESTful API, where the frontend sends HTTP requests to the backend, which in turn processes these requests, retrieves the necessary data from the RDF endpoint, processes it, and sends the data back to the frontend.

Upon receiving the processed data, the Angular frontend takes the GeoJSON [11] data and uses it to update the map display. The map is rendered using Leaflet, a lightweight

JavaScript library for interactive maps. Leaflet is well-suited for handling GeoJSON data and allows the application to overlay historical boundary data on top of a modern world map. When a user selects a location and date, the map is updated with the corresponding boundary data. If a previous version of the boundary exists, a second map is displayed for comparison. This feature is particularly valuable for visualizing the evolution of boundaries over time, providing users with a clear and interactive way to explore historical changes.

The connection between these components is seamless, ensuring that the system operates efficiently from the moment a user initiates a query to the final rendering of the geospatial data on the map. The backend's ability to dynamically generate and execute SPARQL queries based on user input, coupled with its data processing capabilities, allows the application to handle a wide range of queries and datasets. The frontend, with its interactive features, provides a responsive and engaging user experience, making it easy for users to explore complex historical data.

The architecture is designed to be modular and extensible, allowing for future enhancements such as the inclusion of additional datasets, the introduction of more complex queries, or the integration of advanced visualization techniques. By separating the concerns of data management, backend processing, and frontend presentation, the application is able to handle complex linked data queries while maintaining a responsive and user-friendly interface. This modular design also facilitates the maintenance and upgrading of individual components without affecting the overall system, ensuring long-term scalability and adaptability.

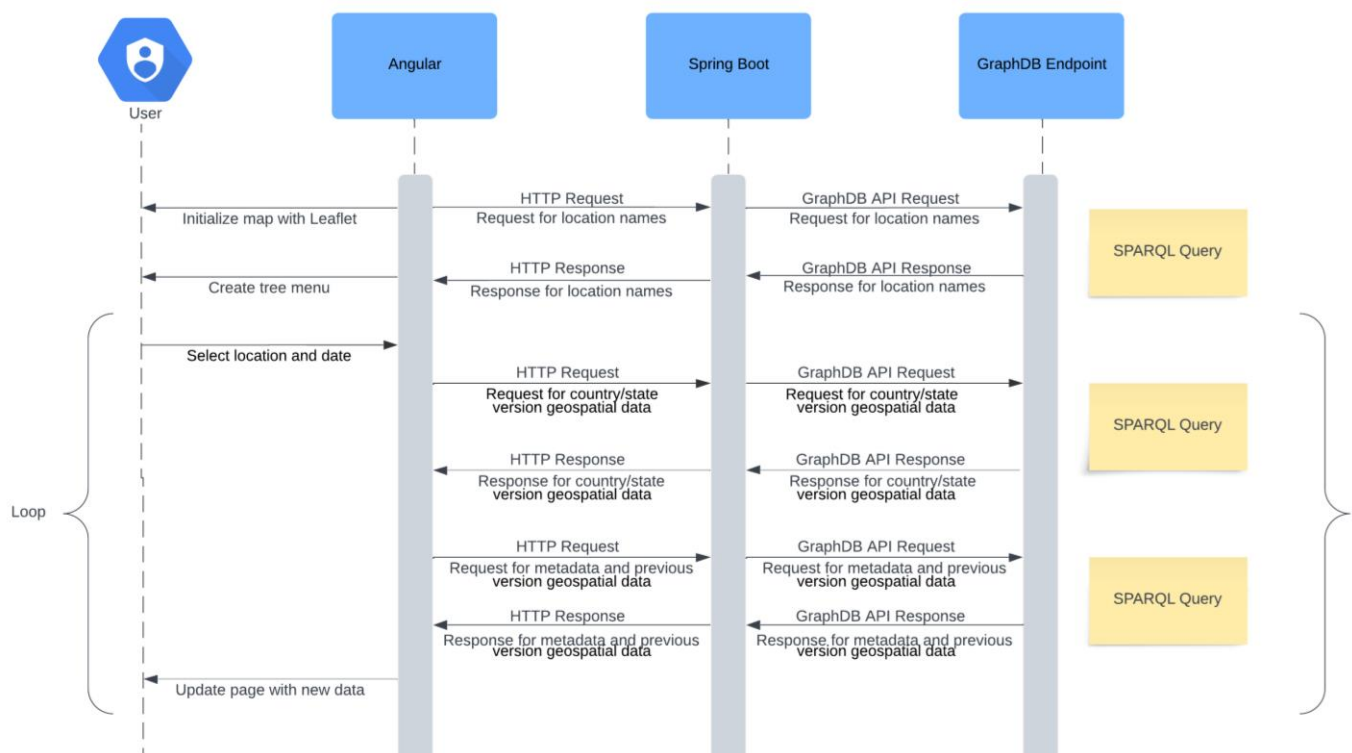


Figure 1: The application's architecture

### 3.3 Technologies

The application operates on a well-coordinated interaction between its various components, each of which plays a vital role in managing, processing, and displaying the historical geospatial data of U.S. counties and states. The foundation of the application lies in the GeoChanges dataset, which is imported into the GraphDB repository. The data is structured in RDF format, where each entry is expressed as a triple (subject, predicate, object). This structure forms the basis of the RDF graph, which allows for the seamless interlinking of different data points. The RDF graph is hierarchical, with entities like states containing multiple counties, and each county having multiple versions over time.

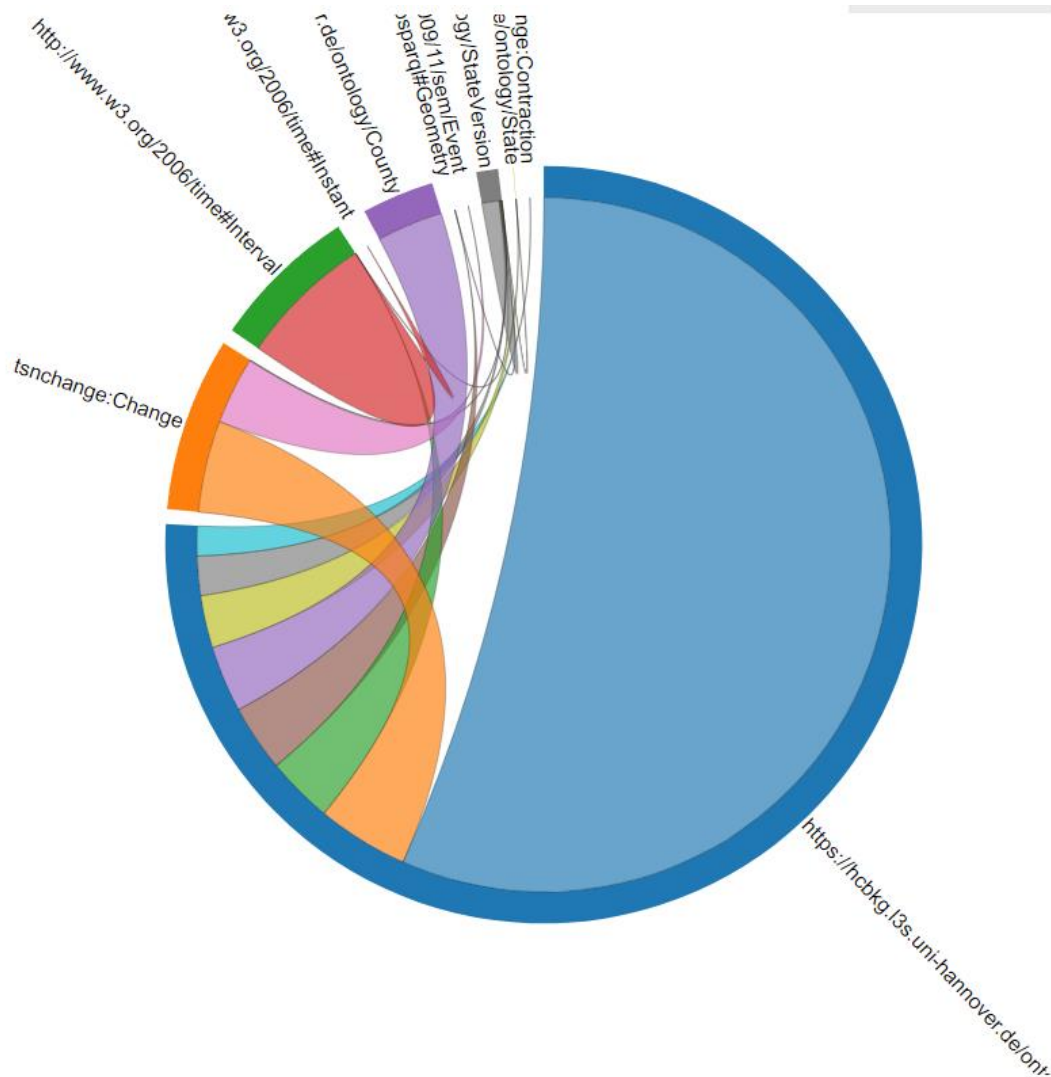
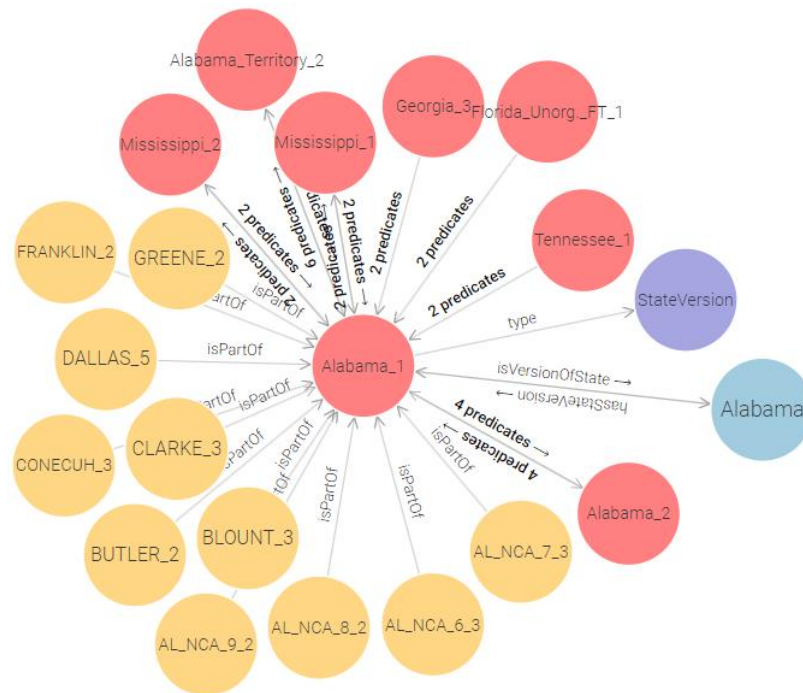


Figure 2 - The class relationships

When the dataset is ingested into GraphDB, the system automatically organizes these triples into a coherent graph structure, enabling complex relationships between entities such as counties, states, and their respective boundary changes. The interlinking between different entities allows for sophisticated queries that can traverse these relationships to extract meaningful data. For example, one could query for all versions of a county within a specific time range and get both the boundary data and metadata for each version. Geospatial data is represented using the “hasGeometry” predicate, where the object is typically a URI pointing to a geometric representation of the boundary in Well-Known Text (WKT) format. The graph, also, uses specific predicates

to define the start and end dates of each boundary version, as these temporal relationships allow the application to retrieve the correct version of a boundary based on a user-selected date.



**Figure 3: A visual example of a part of the RDF graph**

This RDF graph is hosted on a local instance of the GraphDB server with a dedicated repository, which listens for incoming SPARQL queries at the endpoint "localhost:7200". The data is accessible through this endpoint, enabling the backend to query the graph efficiently.

When a user interacts with the application, they initiate a process that begins with the formulation of a SPARQL query based on their input. The input typically includes the name of the location (state or county), the type of location, and a specific date. These parameters are captured via the Angular frontend and passed to the backend through an HTTP request.

Upon receiving the request, the Spring Boot backend establishes a connection to the GraphDB repository using the Eclipse RDF4J API [12]. This connection is validated to ensure that the server is accessible and ready to process queries. Once the connection is established, Spring Boot dynamically constructs a SPARQL query that targets the specific data requested by the user.

For example, if a user requests the boundaries of "Washington County, Arkansas" on a given date, the query is structured to filter results based on the county name, the state name, and the date range. The query is formulated to retrieve the relevant RDF triples that match these criteria. The system also accounts for cases where a location has multiple versions over time, by ensuring that the query retrieves the version that corresponds to the selected date.

The Spring Boot backend exposes several RESTful endpoints to handle different types of requests from the frontend. These endpoints are mapped to specific URLs, each serving a distinct purpose within the application:

- `/search`: This endpoint handles requests for geospatial data. The frontend sends a request with parameters such as `selectedLocation`, `locationType`, and `date`. The backend responds with GeoJSON data representing the boundaries of the requested location for the given date.
- `/metadata`: This endpoint is used to retrieve metadata for a specific location. When a user queries for metadata, the backend processes the request and returns relevant information such as the start and end dates of a county version, associated state, and any other available details.
- `/previousVersion`: This endpoint retrieves the previous version of a boundary, if available. This allows the application to display historical changes in the boundaries on a secondary map for comparison.

The use of distinct endpoints allows the frontend to interact with the backend in a modular way, ensuring that each request is handled efficiently and that the correct type of data is returned. After the SPARQL query is executed against the RDF graph, the resulting data is retrieved by the backend. The nature of the retrieved data depends on the query type—whether it's metadata or geospatial data.

**For Metadata:** If the query is for metadata, such as the start and end dates of a particular county version or its associated state, the backend processes the raw data into a structured format (e.g., JSON) that can be easily consumed by the frontend. This structured metadata is then transmitted back to the Angular frontend.

**For Geospatial Data:** If the query retrieves geospatial data, the backend encounters data in the Well-Known Text (WKT) format, which represents geometric shapes like polygons that define boundaries. However, WKT is not directly compatible with modern web mapping libraries like Leaflet. To address this, the backend performs a conversion of the WKT data into GeoJSON format, which is a widely used format for encoding geographic data. This conversion is handled server-side to reduce the processing load on the client and to ensure the frontend can render the data efficiently.

This GeoJSON data is then sent to the Angular frontend, where it is used to update the map displayed to the user. Upon receiving the processed data from the backend, the Angular frontend dynamically updates the user interface. This interface includes a search bar, a hierarchical tree menu of states and counties, a slider for selecting the year, and the main map display.

**Initial Map Setup:** The primary map is rendered using the Leaflet library, a lightweight and flexible JavaScript library specifically designed for interactive maps. When the application is first loaded, Leaflet initializes the map with a default view centered on the United States. The map is populated with a base layer, typically sourced from OpenStreetMap, which provides the underlying geographical context. This base layer serves as the foundation upon which the application's geospatial data will be overlaid. Leaflet's initialization includes configuring the map's zoom levels, center coordinates, and interaction controls to ensure a user-friendly experience.

**Rendering the Map with Data:** When the user selects a location and date, the Angular app triggers a request to the backend. Once the GeoJSON data is received, Leaflet updates the map by adding a new layer that contains the boundary data for the specified location and date. This layer is styled using predefined options, such as line color, fill color, and opacity, to visually distinguish the historical boundaries from the base map. The map layer is interactive, allowing users to click on the boundaries to view additional metadata or details.

**Handling Map Layers:** The application carefully manages map layers to ensure that the map remains accurate and responsive. Each time a new query is made, the existing boundary layer is removed from the map to prevent clutter and confusion. Leaflet's `removeLayer` function is employed to clear the previous data before the new GeoJSON layer is added. This approach ensures that only the most relevant data is displayed on the map, enhancing both performance and usability.

**Displaying Previous Versions:** In cases where a previous version of a boundary exists, the application creates a second map instance within the same interface. This map is initialized similarly to the primary map but is positioned adjacent to the current boundary map. The second map displays the historical boundary data, providing users with a side-by-side comparison of the current and previous boundaries. The layers on this map are managed in the same way as the primary map, with old layers being cleared before new data is added.

**Metadata Table Display:** Alongside the map, a metadata table is dynamically generated to present detailed information about the selected boundary version. This table includes fields such as the version's start and end dates, the state to which the county belongs, and any other relevant details. The metadata table is initially hidden and only appears after a successful query has been processed. The data for the table is sourced directly from the backend's JSON response and is updated each time the user submits a new query.

The table is designed to be responsive and easily readable, ensuring that users can quickly access the information they need. If a previous version of the boundary exists, the metadata for both the current and previous versions are displayed, providing a comprehensive overview of the boundary's historical changes.

**User Interface Dynamics:** The Angular framework ensures that the user interface remains responsive and interactive. Angular's two-way data binding is used extensively to synchronize user input with the backend queries. For instance, when a user adjusts the year slider, the corresponding date is automatically updated in the calendar, and a new query is triggered to update the map and metadata.

Moreover, Angular's reactive forms capture user input in real-time, providing instant feedback through live search suggestions and dynamic updates to the tree menu. The application also includes error handling to notify users when no data is available for a particular query, ensuring a smooth user experience.

The application's architecture is designed with scalability in mind. The modular nature of the backend and frontend allows for easy expansion. For instance, new datasets can be integrated into the GraphDB repository with minimal changes to the existing infrastructure. Similarly, the Angular frontend can be extended to include additional features or more complex user interactions, such as advanced filtering options or more detailed metadata displays.

Furthermore, the use of industry-standard technologies like SPARQL, GeoSPARQL, and GeoJSON ensures that the application remains compatible with other linked data resources and geospatial tools. This compatibility opens up possibilities for integrating external data sources or exporting the application's data for use in other systems.

In conclusion, the technical implementation of this application showcases the effective use of RDF data structures, SPARQL querying, data conversion processes, and modern web development practices. By combining these elements into a cohesive architecture, the application provides a robust platform for exploring the historical



changes in U.S. county and state boundaries, offering users both detailed metadata and rich geospatial visualizations.

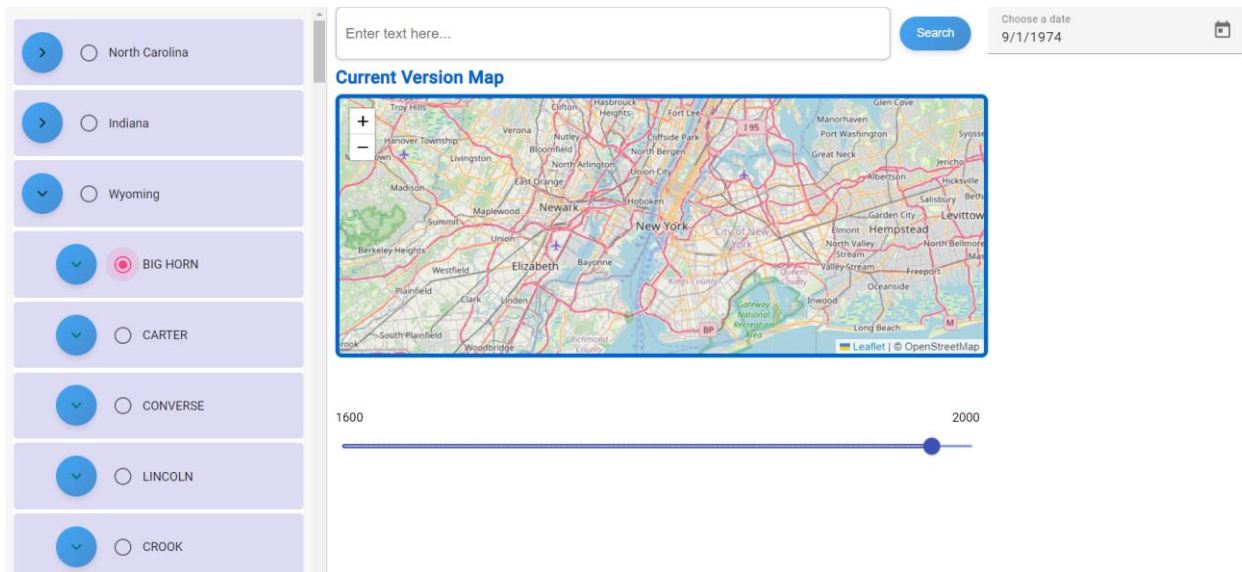


Figure 4: An overview of the User Interface

### 3.4 User Interface

The user interface of the application is designed to be simple yet powerful, providing users with a range of tools for exploring historical geospatial data. Upon opening the application, users are presented with a search bar, a tree menu, a world map, a slider, and a calendar.

#### 3.4.1 The search bar

The search bar is a key feature that allows users to quickly find specific states or counties. As users begin typing into the search bar, live suggestions are immediately displayed beneath it. These suggestions are dynamically generated based on a query to the RDF endpoint, which retrieves the names of all states and counties stored within the linked data repository. This query ensures that the search suggestions are comprehensive and up-to-date, covering every possible location that the user might be interested in. The live suggestion mechanism is particularly useful for users who may not remember the exact spelling of a location's name or who may be uncertain about the full name. By offering relevant suggestions in real-time, the application helps users to easily locate the exact area they are interested in, significantly improving the overall user experience.

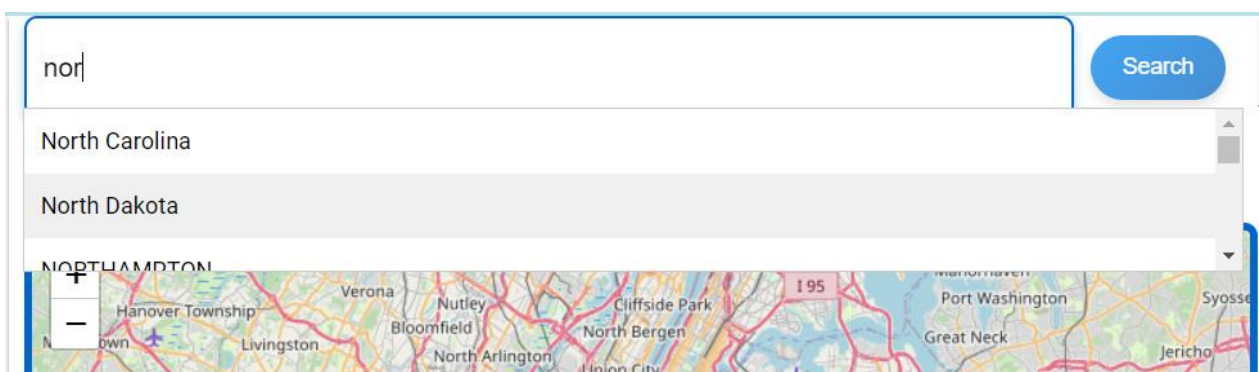


Figure 5 - The search bar with suggestions

### 3.4.2 The tree menu

The tree menu provides an alternative way to explore the data, allowing users to browse through the states and counties in a hierarchical format. The tree menu is populated by sending a specific SPARQL query to the RDF endpoint that requests the names of all states and their associated counties. The returned data is then structured hierarchically in the tree menu, where each state can be expanded to reveal its counties. Users can navigate through this menu to find and select the location they are interested in. This hierarchical structure makes it easier for users to visualize the relationships between different geographic entities, such as understanding which counties belong to which states.

If a user selects a location from the search bar, the corresponding entry in the tree menu is automatically highlighted, providing a seamless transition between the two methods of selection. This synchronization between the search bar and the tree menu ensures that the user interface remains intuitive and user-friendly, allowing users to switch between different methods of selection without losing their place in the data.

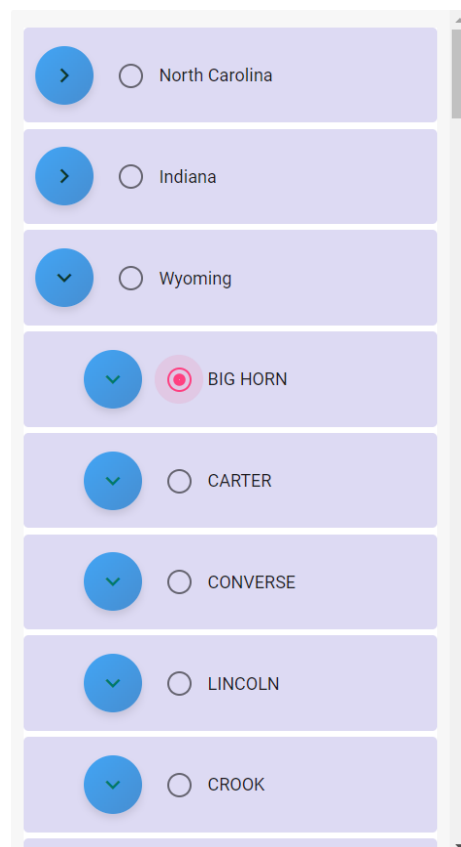


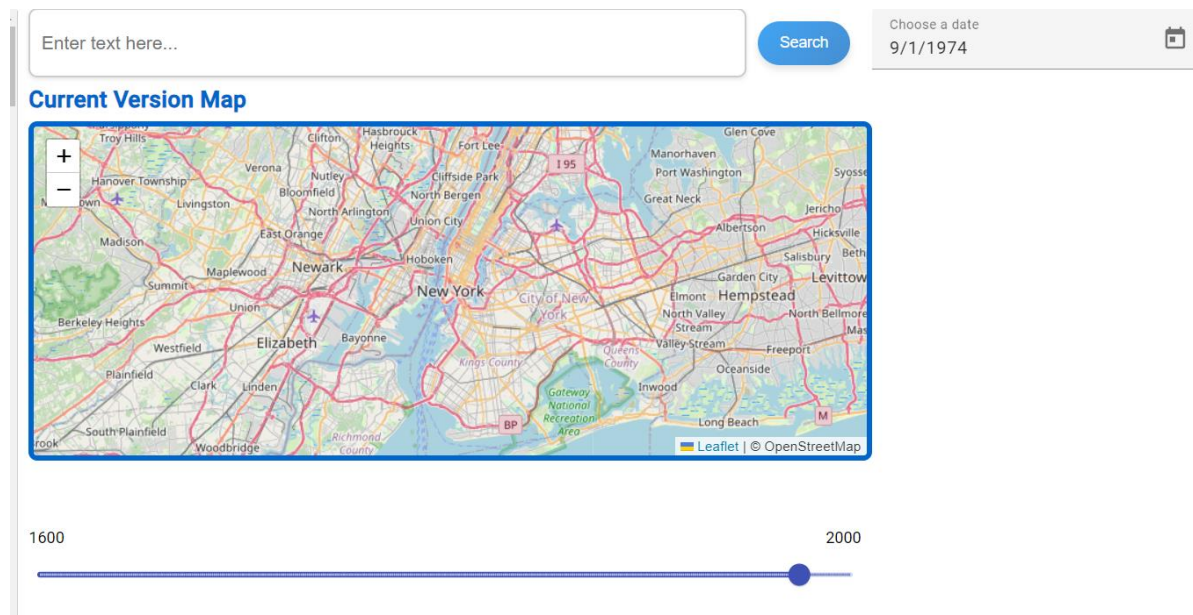
Figure 6 - The tree menu

### 3.4.3 The slider and the calendar

The slider and calendar work together to allow users to select a specific date. The slider is used to choose the year, while the calendar is used to select the exact day and month. The slider is particularly useful for quickly navigating through time, helping users to visualize the temporal distribution of boundary changes. When a user selects a year using the slider, the calendar is automatically updated to reflect that year, ensuring that the selected date is always consistent.



The synchronization between the slider and the calendar is designed to make the selection of dates both intuitive and efficient. Users can quickly scan through years using the slider, and then fine-tune their selection with the calendar, making it easy to pinpoint the exact historical moment they wish to explore.



**Figure 7 - The slider and the calendar in their respective positions**

### 3.4.4 The maps

The maps are the centerpiece of the user interface, providing a visual representation of the historical geospatial data that users query. These maps are rendered using the Leaflet library and are not limited to the borders of the USA. When a user submits a query, the application retrieves the relevant boundary data and updates the map in real-time. The map shows the exact boundaries of the selected state or county for the chosen date, giving users a clear visual context for the historical data.

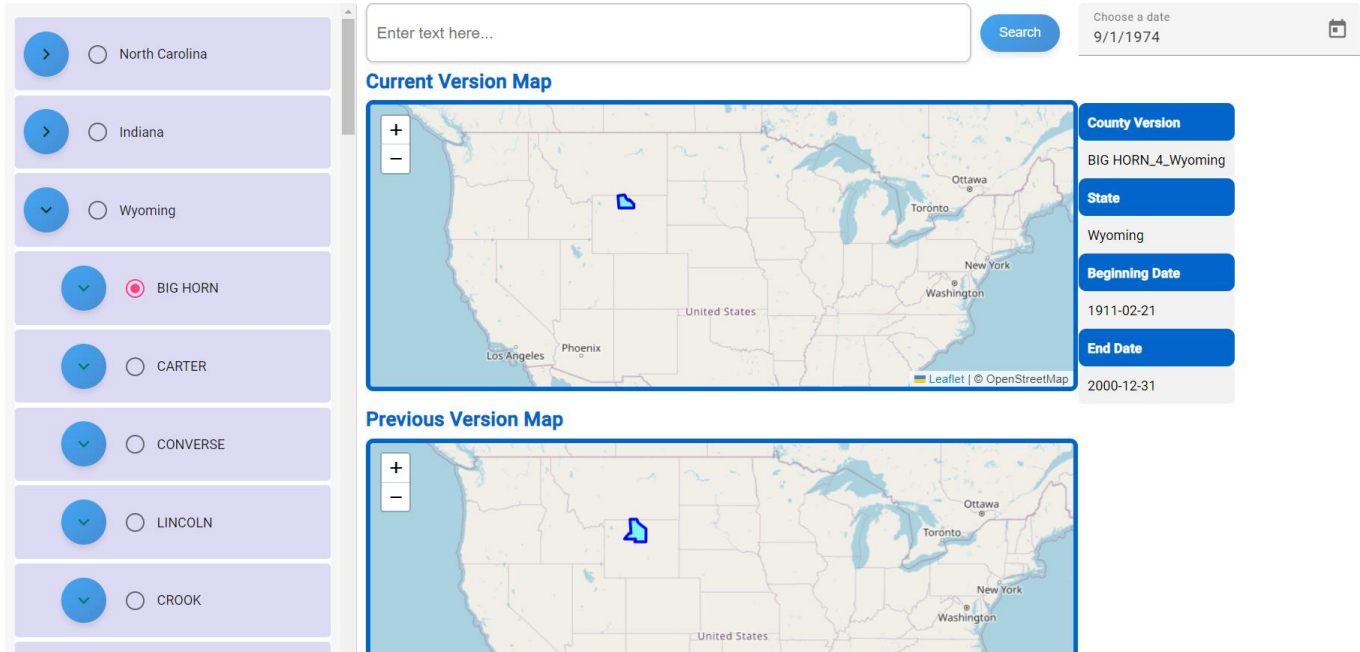
If a previous version of the boundary exists, a second map is displayed alongside the first. This feature allows users to directly compare the current boundary with its historical predecessor, highlighting any changes that have occurred over time. Users can zoom in and out, pan across different regions, and click on specific areas of the map to view more detailed information. The maps are fully integrated with the rest of the user interface. When a user selects a different date or location, the maps are automatically updated to reflect these changes, ensuring that the visual data is always synchronized with the user's selections.

### 3.4.5 Metadata table

In addition to the maps, a metadata table is displayed alongside the results, providing detailed information about the selected boundary version. This table includes the state name, the version number, and the date range for the boundary. This feature allows

users to gain a deeper understanding of the historical context of the boundary changes and to explore the data in more detail.

The metadata is retrieved along with the boundary data during the query process and is displayed in a structured format that makes it easy for users to digest and analyze. This detailed information complements the visual representation provided by the maps, offering users a comprehensive view of the historical geospatial data.



**Figure 8 - The maps with the metadata table showing the results of a query**

The user-friendly design of the interface ensures that all of these features are easily accessible, making the application a powerful tool for exploring historical geospatial data. The combination of live suggestions, a hierarchical tree menu, and synchronized date selection tools provides users with multiple pathways to find and explore the data they are interested in. The seamless integration of the maps with the metadata table further enhances the user's ability to analyze historical boundary changes, making this application an invaluable resource for anyone interested in the historical geography of the United States.

## 4. FUTURE WORK

This thesis has successfully demonstrated a simple yet effective approach to visualizing geospatial data in conjunction with temporal data. The application developed here can serve as a foundational stepping stone for future projects with broader features and enhanced capabilities. There are several areas where this application can be refined and expanded to better serve users and meet the demands of more complex use cases.

One significant avenue for improvement lies in enhancing the interactivity of the map. In the current implementation, users can view historical boundaries and associated metadata. However, future iterations could introduce the ability for users to interact more dynamically with the map, such as by clicking on specific locations to retrieve detailed metadata for that particular date. To enable this functionality, additional layers could be added to the map, making each location individually clickable and capable of displaying rich metadata on demand.

Another enhancement would involve real-time updates to the map as users interact with it. For example, as a user selects a different location or adjusts the date using the slider, the map could update immediately to reflect these changes. This would make the application feel more responsive and provide a more immersive experience. Implementing such real-time updates requires optimizing both the front-end and back-end processes to ensure smooth transitions and data rendering.

Moreover, scalability is another crucial area for future work. As the application currently handles geospatial data for specific locations, there is potential to expand its capabilities to manage and display data for all U.S. states and counties simultaneously. This would involve optimizing the underlying database to efficiently handle large-scale queries. The database would need to be capable of processing over 3,000 queries per second to manage such a high volume of data requests. This scalability could make the application a valuable tool for historians, researchers, and other professionals who require access to comprehensive and detailed geospatial data.

To support this increased functionality, significant optimizations would be required across the entire application stack. From database performance improvements to enhanced front-end processing and rendering techniques, every aspect of the application would need to be fine-tuned to maintain performance and usability.

Additionally, improving the user interface (UI) is paramount to making the application more intuitive and user-friendly. While the current UI provides the basic tools needed for interaction, there is room for making the interface more visually appealing and easier to navigate. For instance, implementing more sophisticated visualization techniques, such as heat maps or thematic layers, could provide users with deeper insights into the data. Furthermore, integrating more advanced search and filter options would allow users to tailor their data exploration to their specific needs.

In summary, while this application serves as a robust prototype for visualizing geospatial and temporal data, there is ample opportunity to expand its capabilities and improve its functionality. By making the map more interactive, enhancing real-time data updates, optimizing for large-scale data handling, and refining the user interface, this application could evolve into a powerful tool for a wide range of users. Future development could focus on these areas, ensuring that the application not only remains relevant but also becomes a go-to resource for anyone needing detailed, historical geospatial data.

## 5. CONCLUSION

The development of this application has underscored the power of linked data technologies in managing and visualizing complex historical geospatial data. By leveraging the RDF framework and integrating it with tools like GraphDB, Spring Boot, Angular, and Leaflet, the application successfully modeled, queried, and visualized the evolving boundaries of U.S. counties and states.

Key findings include the effectiveness of RDF in handling temporal and spatial dynamics, the seamless integration of various technologies to create a responsive and scalable solution, and the challenges inherent in visualizing historical geospatial data. The user-friendly interface, featuring live search, tree menus, and synchronized date selection, significantly enhanced accessibility, making the application a valuable tool for both casual users and researchers.

All in all, this thesis demonstrates the practical utility of linked data in historical research, particularly for geospatial data. The application's modular design ensures its scalability and adaptability for future enhancements, offering a solid foundation for further developments in the field. This work highlights the broader potential of linked data technologies to transform how we interact with historical information, paving the way for innovative research and educational tools.

## ABBREVIATIONS - DIACRITICS - ACRONYMS

RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
TSN	Territorial Statistical Nomenclature
HCB	Historical County Boundaries
SEM	Simple Event Model
W3C	World Wide Web Consortium
URI	Uniform Resource Identifier
ΕΚΠΑ	Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
JSON	JavaScript Object Notation
OWL	Web Ontology Language
SPA	Single-Page Application
USA	United States of America
ΗΠΑ	Ηνωμένες Πολιτείες Αμερικής

## REFERENCES

- [1] Mitsios, M., Punjani, D., Abdollahi, S., Gottschalk, S., Tsalapati, E., Demidova, E., & Koubarakis, M. (2024). *GeoChangesQA: A Question Answering Benchmark for Historical Geospatial Knowledge*. In Proceedings of the European Semantic Web Conference (ESWC) 2024.
- [2] van Hage, W. R., Malaisé, V., Segers, R., Hollink, L., & Schreiber, G. (2011). Design and use of the Simple Event Model (SEM). *Journal of Web Semantics*, 9(2), 128-136.; <https://doi.org/10.1016/j.websem.2011.03.003>
- [3] Hobbs, J. R., & Pan, F. (2006). Time Ontology in OWL. *W3C Working Draft* ; <https://www.w3.org/TR/owl-time/>
- [4] Manola, F., & Miller, E. (2004). RDF Primer. W3C Recommendation; <https://www.w3.org/TR/rdf-primer/>.
- [5] Harris, S., & Seaborne, A. (2013). SPARQL 1.1 Query Language. W3C Recommendation; <https://www.w3.org/TR/sparql11-query/>.
- [6] Open Geospatial Consortium. "OGC GeoSPARQL - A Geographic Query Language for RDF Data." OGC, 2012.
- [7] Vassiliadis, V., & Palpanas, T. (2018). GraphDB: Efficient RDF storage and management for Big Data. *Journal of Big Data*, 5(1), 1-20; <https://graphdb.ontotext.com/documentation/10.7/index.html>.
- [8] Pivotal Software, Inc. (2019). Spring Boot Reference Guide; <https://docs.spring.io/spring-boot/>.
- [9] Google Inc. (2022). Angular Documentation; <https://angular.io/docs>.
- [10] Agafonkin, V. (2010). Leaflet: An open-source JavaScript library for mobile-friendly interactive maps; <https://leafletjs.com/>.
- [11] Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., & Schmidt, C. (2008). The GeoJSON Format Specification; <https://geojson.org/>.
- [12] Broekstra, J., Kampman, A., & van Harmelen, F. (2002). Sesame: A generic architecture for storing and querying RDF and RDF Schema. In *International Semantic Web Conference* (pp. 54-68). Springer, Berlin, Heidelberg; <https://rdf4j.org/documentation/>.
- [13] Bernard, C., Viry, M., Villanova, M., & Gensel, J. (2023). *GeoChangeViz: Visualizing Knowledge Graphs about changes in geographical divisions*. In *ISWC 2023 Posters and Demos: 22nd International Semantic Web Conference* (pp. 1-8). Athens, Greece. CEUR Workshop Proceedings.
- [14] Pelagios - Pelagios Project Overview; <https://pelagios.org/>.
- [15] *Pleiades* - Online Gazetteer of Ancient Places; <https://pleiades.stoa.org/>
- [16] Linked Data: Berners-Lee, T. "Linked Data - Design Issues." W3C, 2006; <https://www.w3.org/DesignIssues/LinkedData.html>.