

# Travel & Tourism

## Ontology and Knowledge Graph

Leahu-Morie Robert-Andrei  
Dita Alin-Gabriel

# Project Scope

Domain: Travel & Tourism

## Objectives:

- Build an ontology and knowledge graph for tourist destinations
- Enable intelligent recommendations based on user preferences
- Connect destinations with attractions, activities, climate, and costs



# Main Functionalities

- Smart Destination Search: Find places based on activities, budget, or season
- Comparison Tool: Compare cities based on cost, weather, attractions
- Personalized Suggestions: Get recommendations based on interests
- Semantic Navigation: Discover connections between destinations and activities

# Why Use a Knowledge Graph?

- Semantic Flexibility: Easy schema evolution and expansion
- Rich Relationships: Handles complex links between locations, attractions, and activities
- Inference & Reasoning: Enables advanced recommendations beyond simple filtering



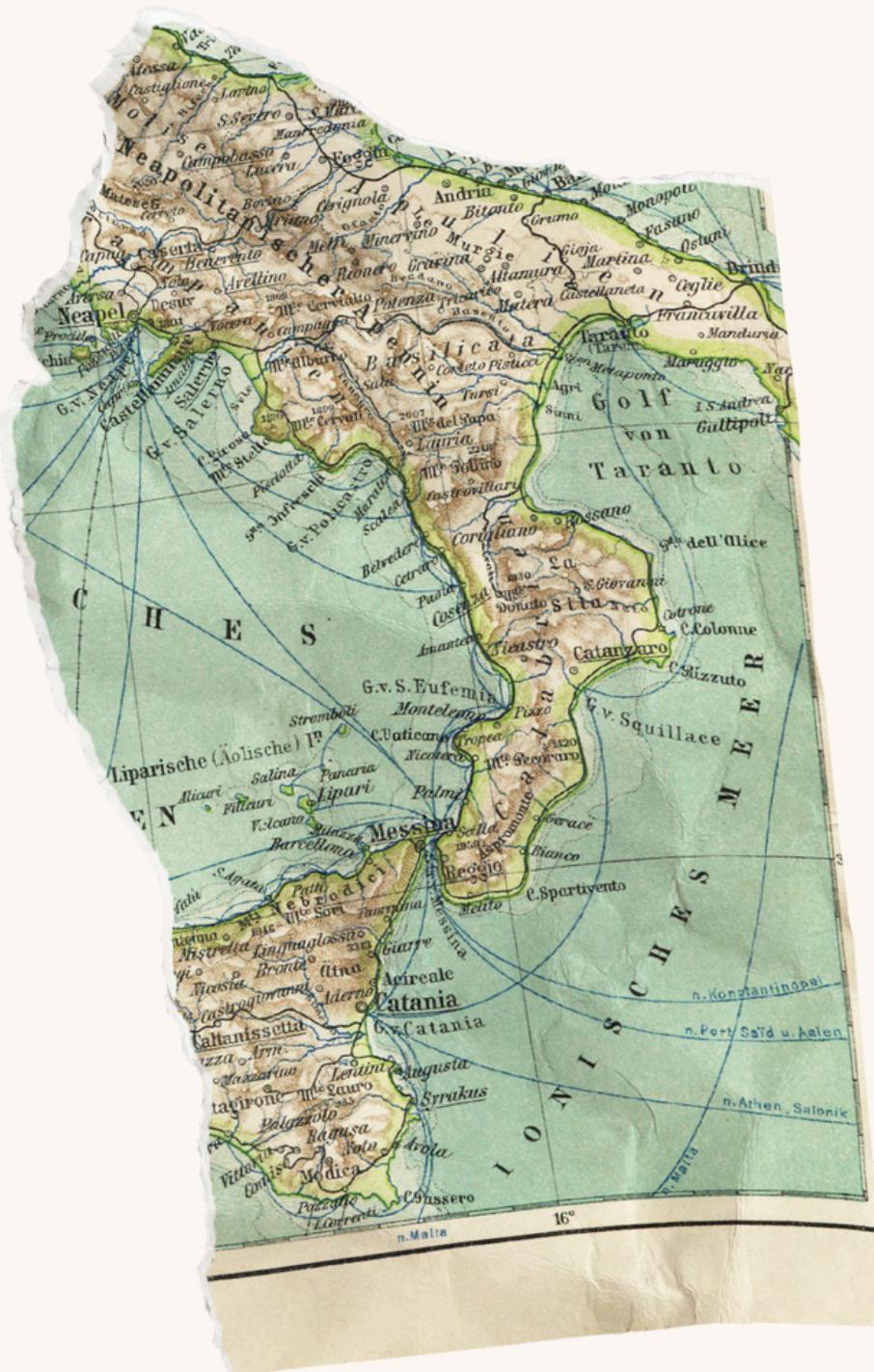
# Data Sources

## Primary Data Sources:

- Wikidata & DBpedia (geographic & cultural data)
  - OpenStreetMap (location & infrastructure data)
  - OpenWeather API (climate & seasonal info)
  - Travel websites (TripAdvisor, Booking, Google Travel)

## Extraction Methods:

- Web Crawling (Scrapy, BeautifulSoup)
  - API Integration (GraphQL, REST APIs)
  - CSV/Excel datasets (open government tourism data)



# Ontology Model - Key Classes & Properties

## Main Classes:

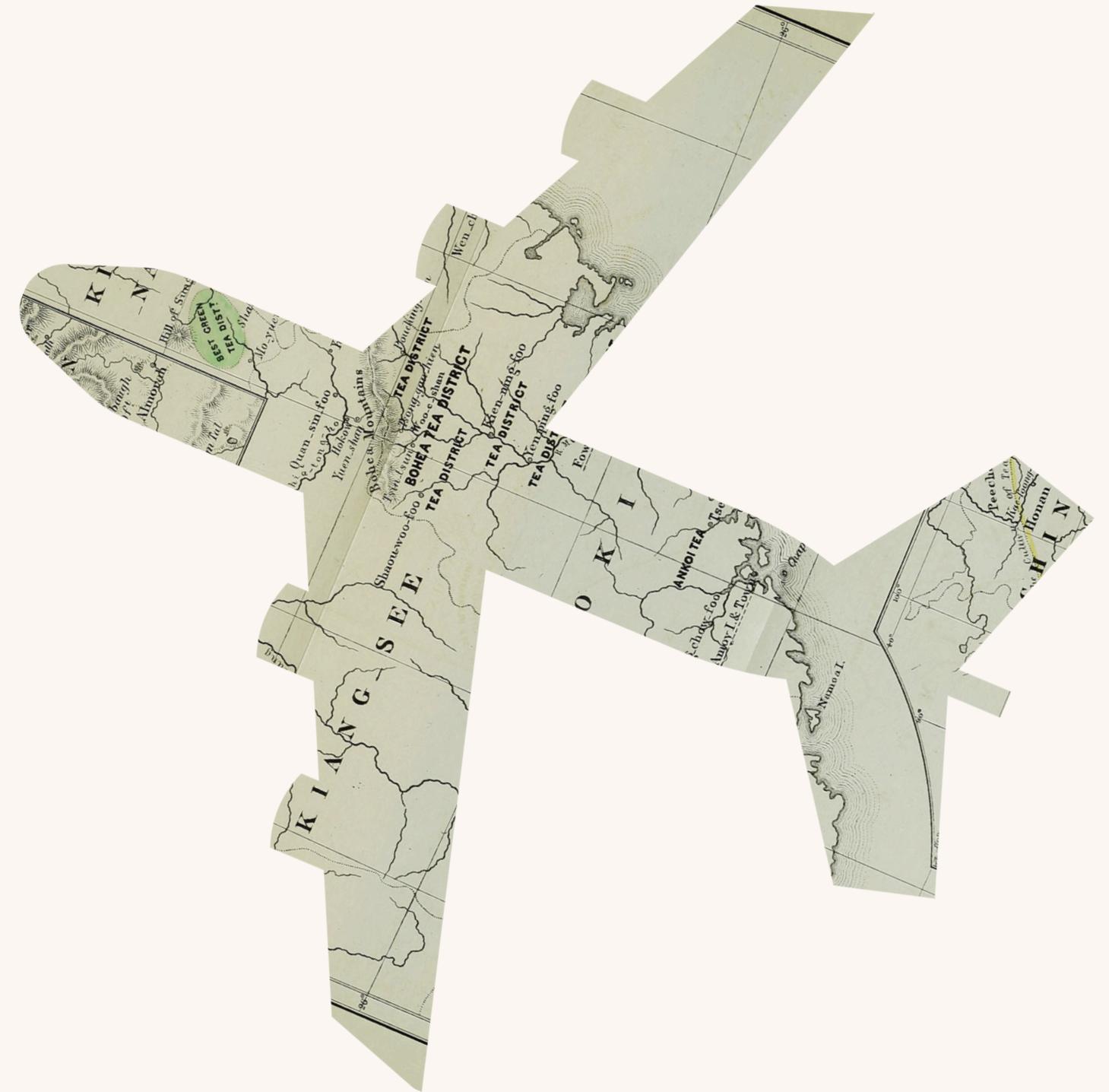
- `Destination` (city, region, country)
- `Attraction` (monuments, nature parks, museums)
- `Activity` (hiking, water sports, cultural tours)
- `Accommodation` (hotels, hostels, resorts)
- `Factor` (climate, safety, affordability)

## Key Properties:

- `located\_in` (Paris → France)
- `offers\_activity` (Rome → Cultural Tours)
- `has\_cost\_level` (London → Expensive)
- `has\_best\_season` (Thailand → November-March)

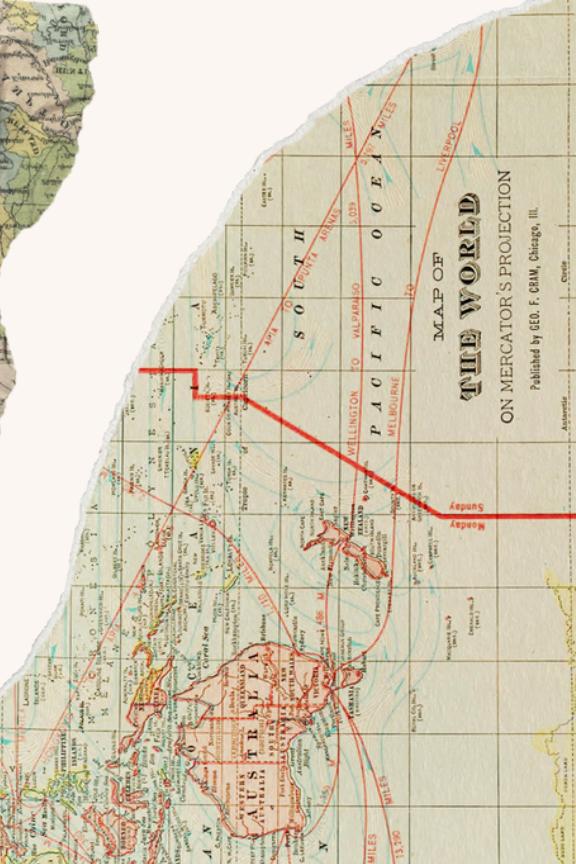
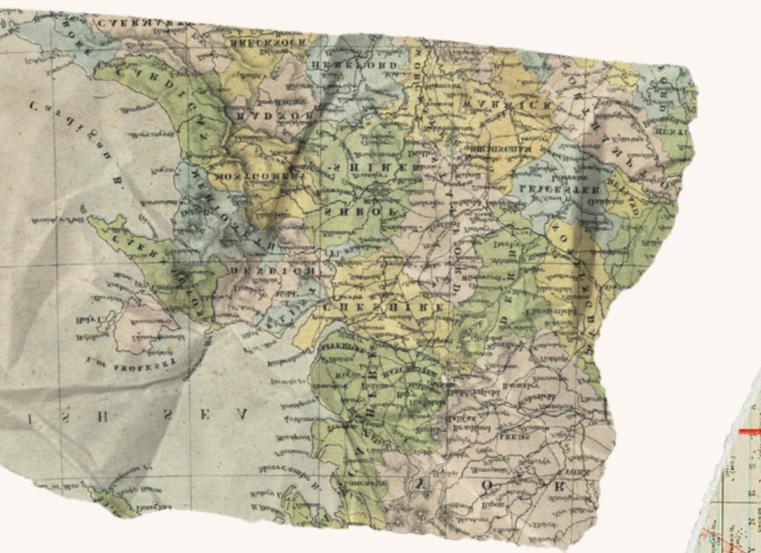
# Technologies to be Used

- Ontology Development: OWL, Protégé
- Backend & Data Processing:
  - > Python (Scrapy, BeautifulSoup, RDFLib, Pandas)
- Knowledge Graph Storage:
  - > GraphDB / Virtuoso / RDFLib
- Querying & Semantic Search:
  - > SPARQL
- Web Interface:
  - > React / Next.js (optional frontend)
  - > Angular/ Flask etc.



# Next Steps

- Model final ontology structure
- Collect & clean raw data
- Implement initial knowledge graph population
- Set up SPARQL endpoint



# Thank You

## Questions?