# 15-Minute City Accessibility Ontology

## Documentation Guide

**Version:** 1.0.0
**Date:** 2025-11-29
**Namespace:** `http://example.org/ontology/15min-city#`
**Prefix:** `fmc`

## Table of Contents

## 1. Executive Summary

The **15-Minute City Accessibility Ontology** is a formal OWL (Web Ontology Language) knowledge representation designed to model urban accessibility data, demographic information, and equity analysis within the context of the 15-minute city urban planning concept.

## Purpose

This ontology provides:

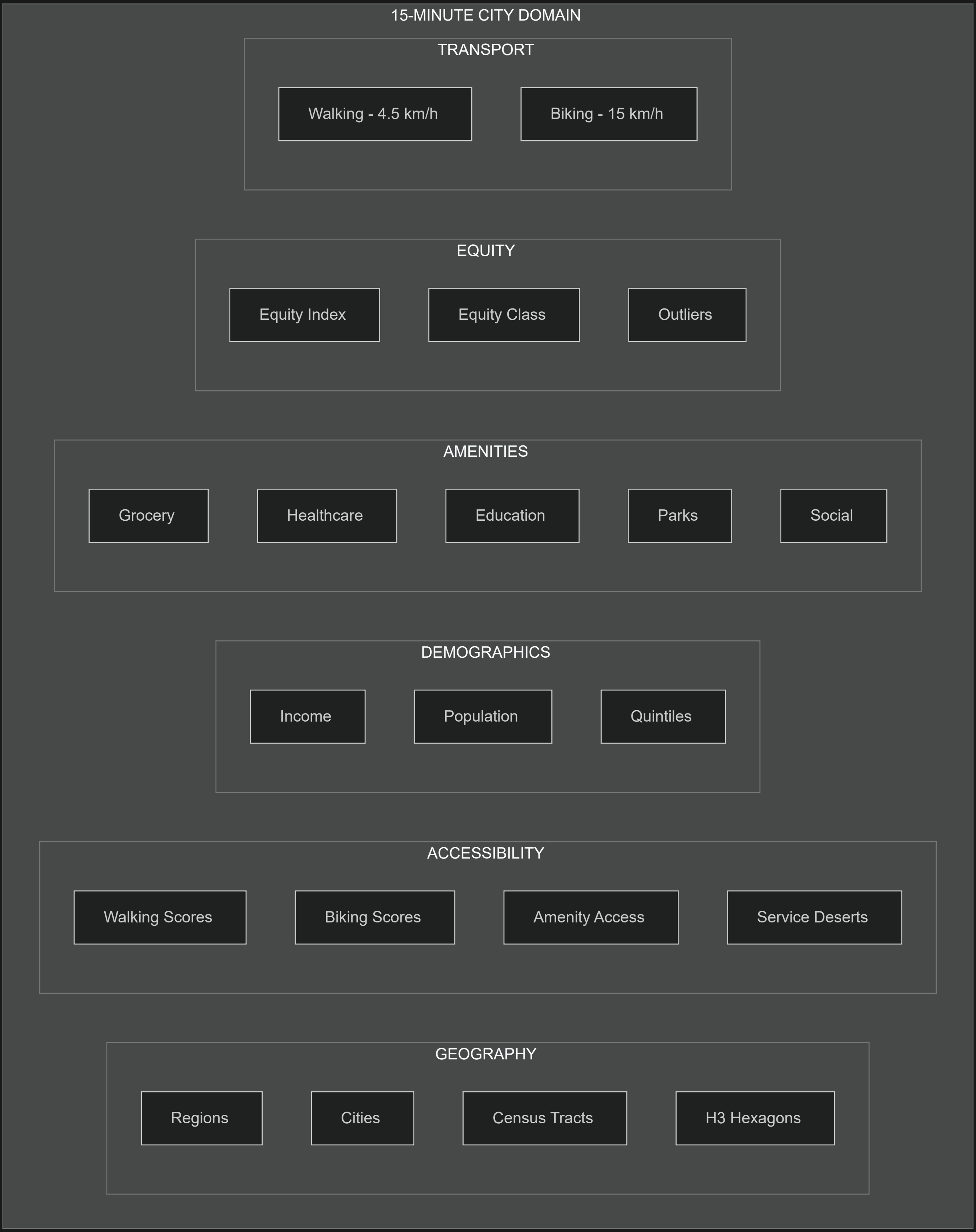
- **Formal semantics** for accessibility metrics and their relationships
- **Reasoning capabilities** to automatically classify census tracts (e.g., food deserts, high-equity areas)
- **Interoperability** with other linked data sources through standard RDF/OWL
- **Query support** via SPARQL for complex analytical questions
- **Documentation** of the domain knowledge captured in the dashboard

## Key Features

| Feature | Description |
|---|---|
| **Classes** | 25+ classes organized in hierarchical structure |
| **Object Properties** | 12 properties defining relationships |
| **Datatype Properties** | 30+ properties for quantitative data |
| **Named Individuals** | 35+ pre-defined instances (cities, regions, quintiles) |
| **Defined Classes** | 5 classes with logical axioms for reasoning |

# 2. Ontology Overview

## Domain Scope



15-MINUTE CITY DOMAIN

TRANSPORT
- Walking - 4.5 km/h
- Biking - 15 km/h

EQUITY
- Equity Index
- Equity Class
- Outliers

AMENITIES
- Grocery
- Healthcare
- Education
- Parks
- Social

DEMOGRAPHICS
- Income
- Population
- Quintiles

ACCESSIBILITY
- Walking Scores
- Biking Scores
- Amenity Access
- Service Deserts

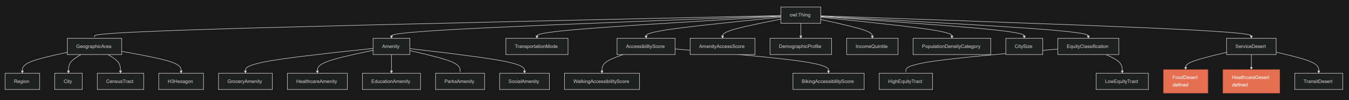
GEOGRAPHY
- Regions
- Cities
- Census Tracts
- H3 Hexagons

## Design Principles

1. **Modularity**: Classes organized into logical groupings
2. **Extensibility**: Easy to add new cities, amenities, or metrics
3. **Reasoning-ready**: Defined classes enable automatic classification
4. **Data-aligned**: Properties map directly to CSV/Parquet columns

# 3. Class Hierarchy

## Complete Class Tree



## Class Descriptions

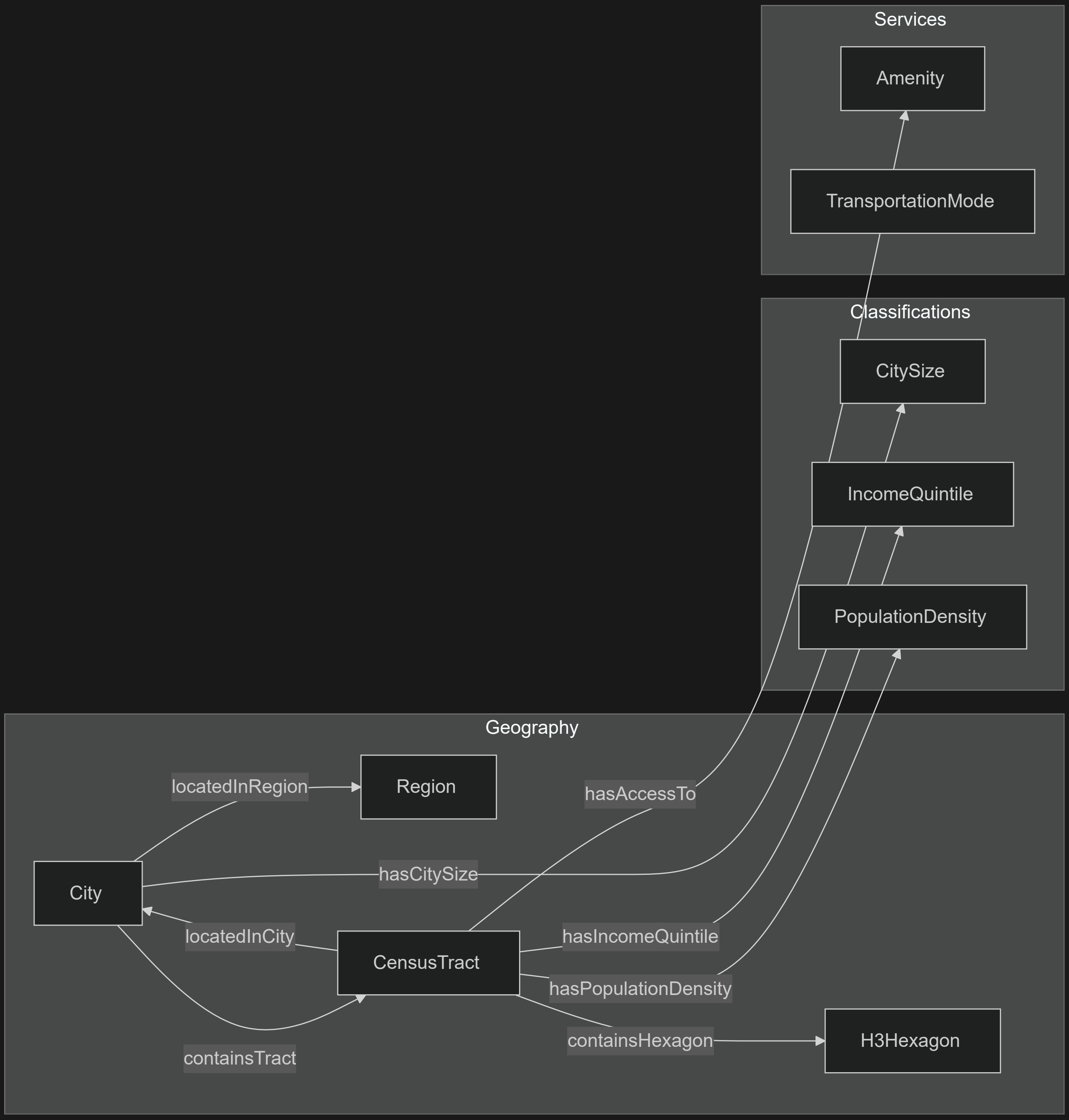
| Class | Description | Cardinality |
|---|---|---|
| `GeographicArea` | Abstract base class for all spatial entities | - |
| `Region` | U.S. geographic region (enumerated: Northeast, South, Midwest, West) | 4 |
| `City` | Metropolitan area containing census tracts | 16 |
| `CensusTract` | Statistical subdivision (~2,500-8,000 people) | 5,955 |
| `H3Hexagon` | Uber H3 hexagonal spatial unit | Variable |
| `Amenity` | Essential service category | 5 subtypes |
| `TransportationMode` | Travel method (enumerated: Walking, Biking) | 2 |
| `IncomeQuintile` | Income classification (enumerated: Q1-Q5) | 5 |
| `ServiceDesert` | Area lacking essential services | 3 subtypes |

# 4. Object Properties

## Property Summary

| Property | Domain | Range | Functional | Inverse |
|---|---|---|---|---|
| `locatedInRegion` | City | Region | ✓ | - |
| `locatedInCity` | CensusTract | City | ✓ | `containsTract` |
| `containsTract` | City | CensusTract | - | `locatedInCity` |
| `containsHexagon` | CensusTract | H3Hexagon | - | - |
| `hasCitySize` | City | CitySize | ✓ | - |
| `hasIncomeQuintileCity` | CensusTract | IncomeQuintile | ✓ | - |
| `hasIncomeQuintileNational` | CensusTract | IncomeQuintile | ✓ | - |
| `hasAccessQuintileCity` | CensusTract | IncomeQuintile | ✓ | - |
| `hasAccessQuintileNational` | CensusTract | IncomeQuintile | ✓ | - |
| `hasPopulationDensity` | CensusTract | PopulationDensityCategory | ✓ | - |
| `hasAccessTo` | GeographicArea | Amenity | - | - |
| `measuredBy` | AccessibilityScore | TransportationMode | - | - |

## Property Diagram



# 5. Datatype Properties

## Identifier Properties

| Property | Domain | Range | Description |
|---|---|---|---|
| `hasGEOID` | CensusTract | xsd:string | 11-digit FIPS code |
| `hasCityKey` | City | xsd:string | Normalized city identifier |
| `hasStateFIPS` | CensusTract | xsd:string | 2-digit state code |
| `hasCountyFIPS` | CensusTract | xsd:string | 3-digit county code |
| `hasTractNumber` | CensusTract | xsd:string | 6-digit tract number |

## Coordinate Properties

| Property | Domain | Range | Description |
|---|---|---|---|
| `hasLatitude` | GeographicArea | xsd:decimal | Centroid latitude |
| `hasLongitude` | GeographicArea | xsd:decimal | Centroid longitude |

## Accessibility Score Properties

| Property | Domain | Range | Maps to Column |
|---|---|---|---|
| `hasWalkingScore` | CensusTract | xsd:decimal | `score_walk` |
| `hasBikingScore` | CensusTract | xsd:decimal | `score_bike` |
| `hasGroceryAccessWalk` | CensusTract | xsd:decimal | `grocery_walk` |
| `hasGroceryAccessBike` | CensusTract | xsd:decimal | `grocery_bike` |
| `hasHealthcareAccessWalk` | CensusTract | xsd:decimal | `healthcare_walk` |
| `hasHealthcareAccessBike` | CensusTract | xsd:decimal | `healthcare_bike` |
| `hasEducationAccessWalk` | CensusTract | xsd:decimal | `education_walk` |
| `hasEducationAccessBike` | CensusTract | xsd:decimal | `education_bike` |
| `hasParksAccessWalk` | CensusTract | xsd:decimal | `parks_walk` |
| `hasParksAccessBike` | CensusTract | xsd:decimal | `parks_bike` |
| `hasSocialAccessWalk` | CensusTract | xsd:decimal | `social_walk` |
| `hasSocialAccessBike` | CensusTract | xsd:decimal | `social_bike` |

## Demographic Properties

| Property | Domain | Range | Maps to Column |
|---|---|---|---|
| `hasMedianIncome` | CensusTract | xsd:decimal | `median_income` |
| `hasTotalPopulation` | CensusTract | xsd:integer | `total_pop` |
| `hasNumHexagons` | CensusTract | xsd:integer | `num_hexagons` |

## Derived Metric Properties

| Property | Domain | Range | Formula |
|---|---|---|---|
| `hasEquityIndex` | CensusTract | xsd:decimal | `score_walk / (median_income / 10000)` |
| `hasBikeAdvantage` | CensusTract | xsd:decimal | `(score_bike - score_walk) / score_walk * 100` |
| `hasAccessibilityGapWalk` | CensusTract | xsd:decimal | `5.0 - score_walk` |
| `hasAccessibilityGapBike` | CensusTract | xsd:decimal | `5.0 - score_bike` |

### City Aggregate Properties

| Property | Domain | Range | Description |
|---|---|---|---|
| hasAvgWalkingScore | City | xsd:decimal | Mean walking score |
| hasAvgBikingScore | City | xsd:decimal | Mean biking score |
| hasAvgMedianIncome | City | xsd:decimal | Mean income |
| hasTotalCityPopulation | City | xsd:integer | Sum of tract populations |
| hasNumTracts | City | xsd:integer | Count of census tracts |

## 6. Named Individuals

### Regions (4)

| Individual | Label |
|---|---|
| fmc:Northeast | Northeast |
| fmc:South | South |
| fmc:Midwest | Midwest |
| fmc:West | West |

## Cities (16)

| Individual | Label | Region | Size |
|---|---|---|---|
| fmc:WashingtonDC | Washington, D.C. | Northeast | Small |
| fmc:Boston | Boston, MA | Northeast | Medium |
| fmc:Atlanta | Atlanta, GA | South | Large |
| fmc:Dallas | Dallas, TX | South | Large |
| fmc:SanAntonio | San Antonio, TX | South | Large |
| fmc:NewOrleans | New Orleans, LA | South | Small |
| fmc:Tulsa | Tulsa, OK | South | Small |
| fmc:Chicago | Chicago, IL | Midwest | Large |
| fmc:Detroit | Detroit, MI | Midwest | Small |
| fmc:Indianapolis | Indianapolis, IN | Midwest | Medium |
| fmc:Minneapolis | Minneapolis, MN | Midwest | Small |
| fmc:LosAngeles | Los Angeles, CA | West | Large |
| fmc:SanFrancisco | San Francisco, CA | West | Medium |
| fmc:Seattle | Seattle, WA | West | Medium |
| fmc:Albuquerque | Albuquerque, NM | West | Small |
| fmc:ColoradoSprings | Colorado Springs, CO | West | Small |

## Transportation Modes (2)

| Individual | Label | Speed |
|---|---|---|
| fmc:Walking | Walking | 4.5 km/h |
| fmc:Biking | Biking | 15.0 km/h |

### Income Quintiles (5)

| Individual | Label | Percentile Range |
|---|---|---|
| `fmc:Q1` | Q1 (Lowest) | 0-20% |
| `fmc:Q2` | Q2 | 20-40% |
| `fmc:Q3` | Q3 | 40-60% |
| `fmc:Q4` | Q4 | 60-80% |
| `fmc:Q5` | Q5 (Highest) | 80-100% |

### City Sizes (3)

| Individual | Label | Population Range |
|---|---|---|
| `fmc:SmallCity` | Small City | < 500,000 |
| `fmc:MediumCity` | Medium City | 500,000 - 1,000,000 |
| `fmc:LargeCity` | Large City | > 1,000,000 |

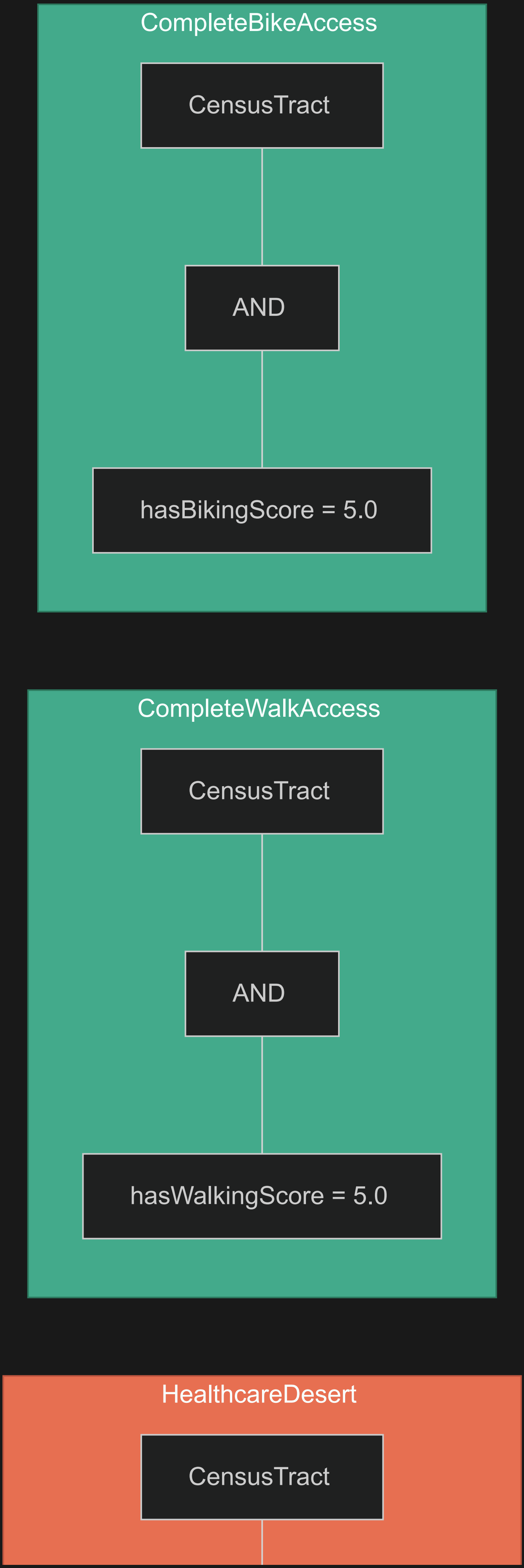
### Population Density Categories (3)

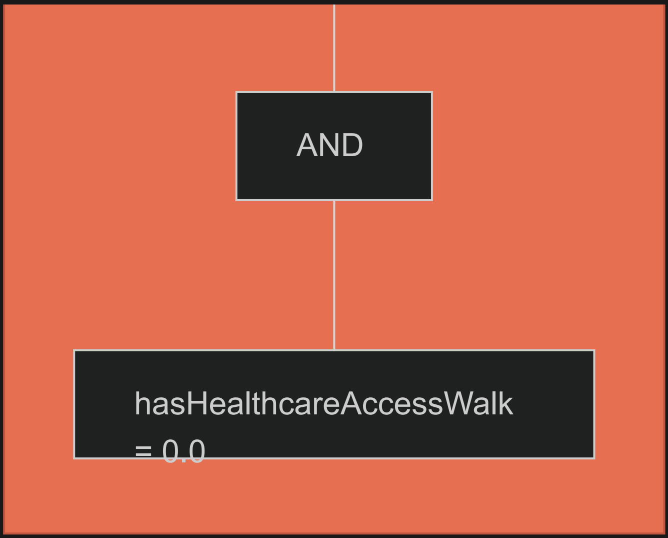
| Individual | Label |
|---|---|
| `fmc:LowDensity` | Low Density |
| `fmc:MediumDensity` | Medium Density |
| `fmc:HighDensity` | High Density |

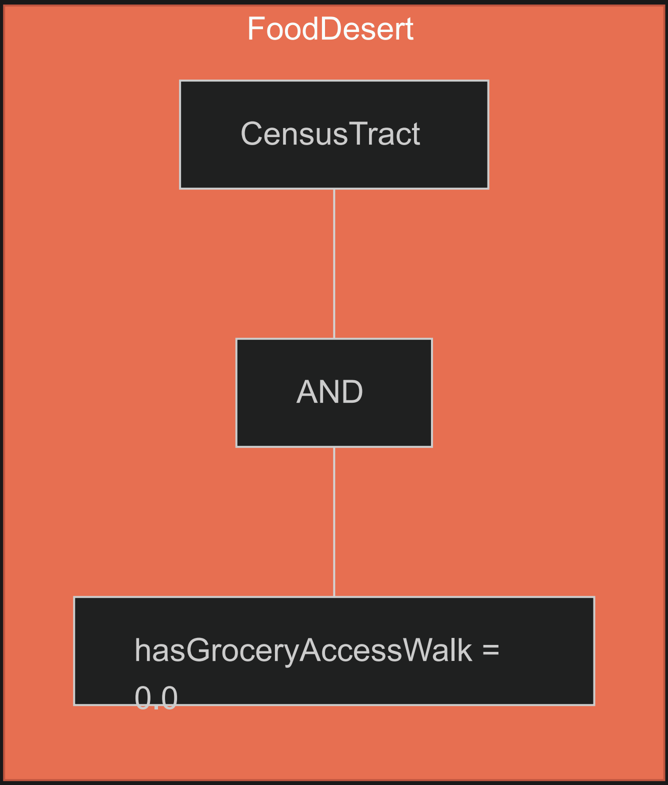
## 7. Defined Classes (Logical Axioms)

The ontology includes **defined classes** with OWL equivalence axioms that enable automatic classification through reasoning.

## Defined Class Axioms

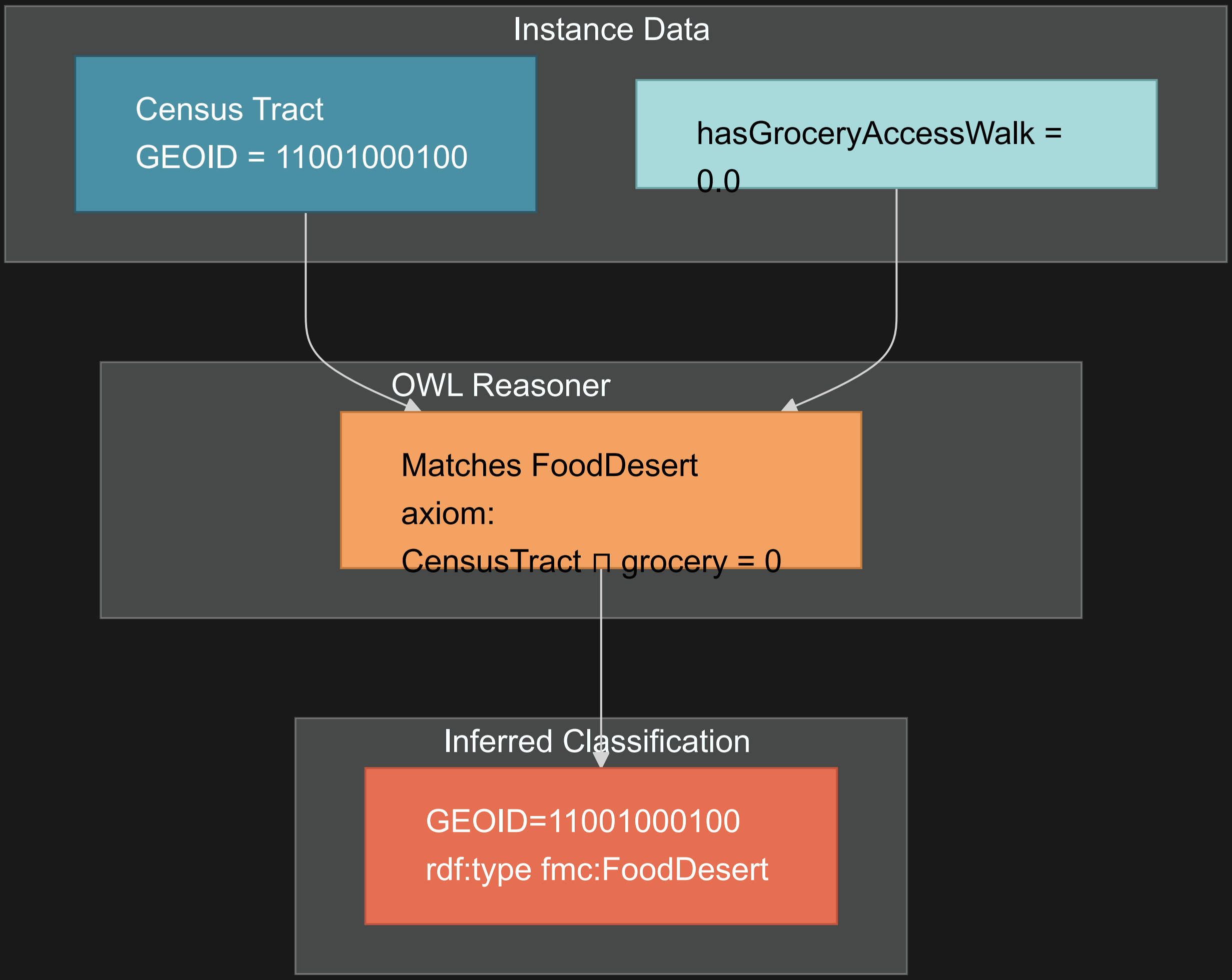

CompleteBikeAccess
CensusTract
AND
hasBikingScore = 5.0
CompleteWalkAccess
CensusTract
AND
hasWalkingScore = 5.0
HealthcareDesert
CensusTract





| Defined Class | Logical Axiom | Interpretation |
|---|---|---|
| **FoodDesert** | `CensusTract ⊓ ∃hasGroceryAccessWalk.{0.0}` | Auto-classified when grocery_walk = 0 |
| **HealthcareDesert** | `CensusTract ⊓ ∃hasHealthcareAccessWalk.{0.0}` | Auto-classified when healthcare_walk = 0 |
| **CompleteWalkAccess** | `CensusTract ⊓ ∃hasWalkingScore.{5.0}` | All 5 amenities within 15-min walk |
| **CompleteBikeAccess** | `CensusTract ⊓ ∃hasBikingScore.{5.0}` | All 5 amenities within 15-min bike |

### Reasoning Example



The classification happens automatically without explicit assertion in the data.

## 8. Competency Questions

The ontology is designed to answer these analytical questions via SPARQL:

### Geographic Questions

1. **Q1**: Which census tracts are located in a specific city?
2. **Q2**: What cities are in the Midwest region?
3. **Q3**: How many H3 hexagons does a census tract contain?

## Accessibility Questions

4. **Q4**: What is the average walking score for a city?
5. **Q5**: Which tracts have complete walking access (score = 5)?
6. **Q6**: What is the bike advantage for a specific tract?
7. **Q7**: Which amenity has the lowest access rate?

## Equity Questions

8. **Q8**: Which tracts are food deserts?
9. **Q9**: What is the relationship between income quintile and access quintile?
10. **Q10**: Which low-income tracts have high accessibility (equity outliers)?

## Comparative Questions

11. **Q11**: How do cities compare on average accessibility?
12. **Q12**: What is the accessibility gap between walking and biking?
13. **Q13**: Which region has the highest average equity index?

# 9. Usage Examples

## SPARQL Query: Find Food Deserts

```
PREFIX fmc: <http://example.org/ontology/15min-city#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?tract ?city ?income
WHERE {
  ?tract rdf:type fmc:FoodDesert .
  ?tract fmc:locatedInCity ?city .
  ?tract fmc:hasMedianIncome ?income .
}
ORDER BY ?income
```

## SPARQL Query: City Accessibility Comparison

```
PREFIX fmc: <http://example.org/ontology/15min-city#>

SELECT ?city ?avgWalk ?avgBike
WHERE {
  ?city rdf:type fmc:City .
  ?city fmc:hasAvgWalkingScore ?avgWalk .
  ?city fmc:hasAvgBikingScore ?avgBike .
}
ORDER BY DESC(?avgWalk)
```

## SPARQL Query: Equity Analysis

```
PREFIX fmc: <http://example.org/ontology/15min-city#>

SELECT ?tract ?incomeQ ?accessQ
WHERE {
  ?tract rdf:type fmc:CensusTract .
  ?tract fmc:hasIncomeQuintileCity ?incomeQ .
  ?tract fmc:hasAccessQuintileCity ?accessQ .
  FILTER(?incomeQ = fmc:Q1 && ?accessQ = fmc:Q5)
}
```

## Python RDFLib Integration

```
from rdflib import Graph, Namespace, RDF

# Load ontology
g = Graph()
g.parse("15min-city.owl", format="xml")

# Define namespace
FMC = Namespace("http://example.org/ontology/15min-city#")

# Query for cities in the West region
query = """
  PREFIX fmc: <http://example.org/ontology/15min-city#>
  SELECT ?city ?label
  WHERE {
    ?city fmc:locatedInRegion fmc:West .
    ?city rdfs:label ?label .
  }
"""

for row in g.query(query):
  print(f"{row.label}: {row.city}")
```

# 10. Integration with Semantic Model

## Mapping to Data Columns

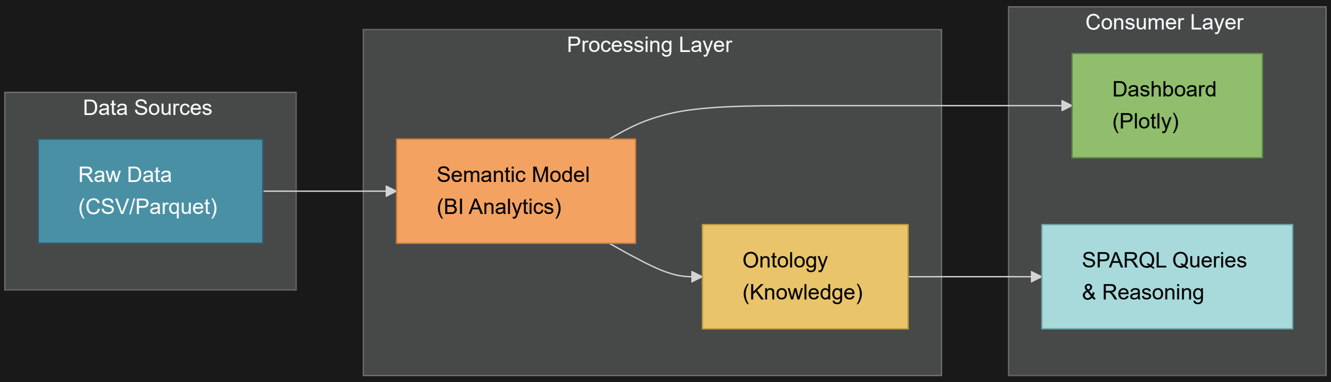
The ontology properties map directly to the semantic model's data columns:

| Semantic Model Column | Ontology Property | Notes |
|---|---|---|
| `GEOID` | `hasGEOID` | Primary identifier |
| `city` | `locatedInCity` | Object property |
| `region` | `locatedInRegion` (via city) | Transitive lookup |
| `score_walk` | `hasWalkingScore` | Core metric |
| `score_bike` | `hasBikingScore` | Core metric |
| `grocery_walk` | `hasGroceryAccessWalk` | Amenity access |
| `median_income` | `hasMedianIncome` | Demographic |
| `income_quintile_city` | `hasIncomeQuintileCity` | Classification |
| `equity_index` | `hasEquityIndex` | Derived metric |
| `food_desert` | (inferred via FoodDesert class) | Reasoner-derived |

## Complementary Roles

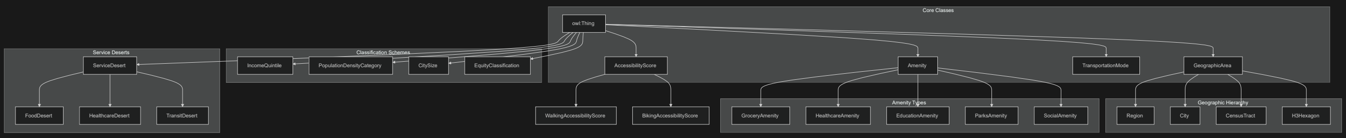
| Aspect | Semantic Model | Ontology |
|---|---|---|
| **Purpose** | BI & Visualization | Knowledge Representation |
| **Query Language** | SQL/DAX | SPARQL |
| **Relationships** | Foreign Keys | Object Properties |
| **Classifications** | Explicit Columns | Inferred via Reasoning |
| **Extensibility** | Add Columns | Add Classes/Properties |
| **Standards** | Proprietary | W3C OWL/RDF |

## Combined Workflow

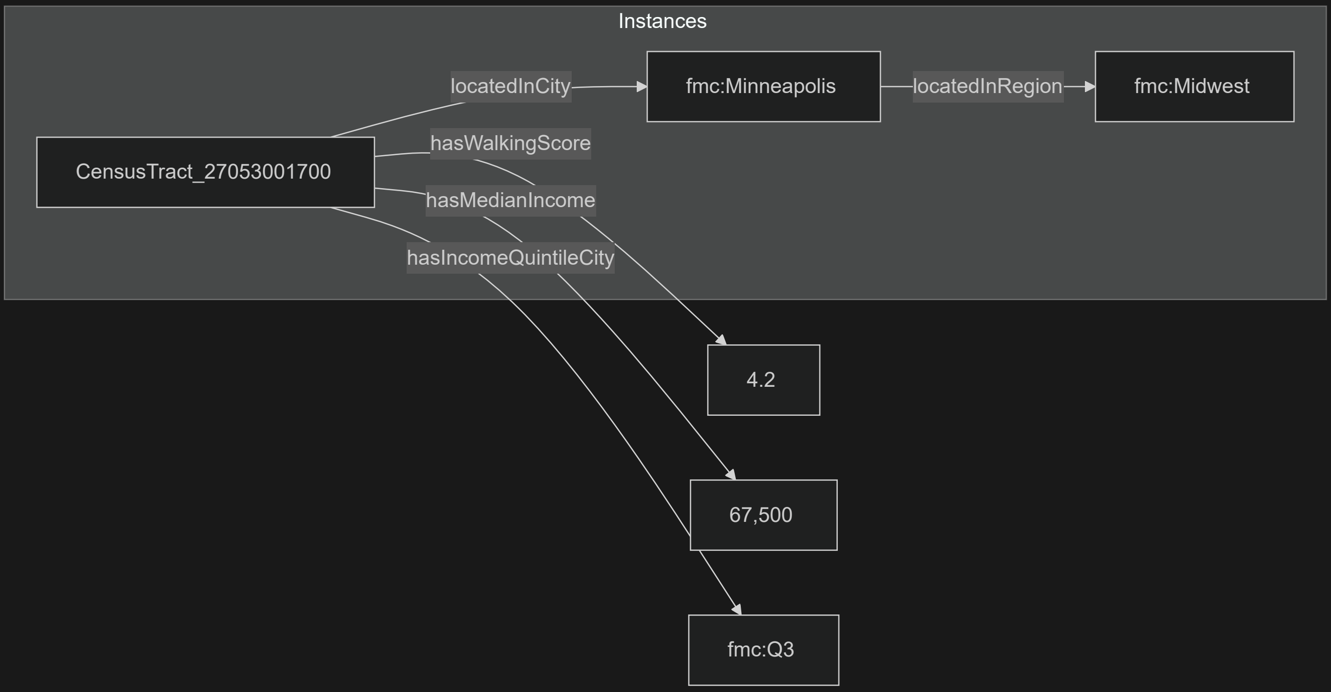


# 11. Visualization

## Class Hierarchy Diagram



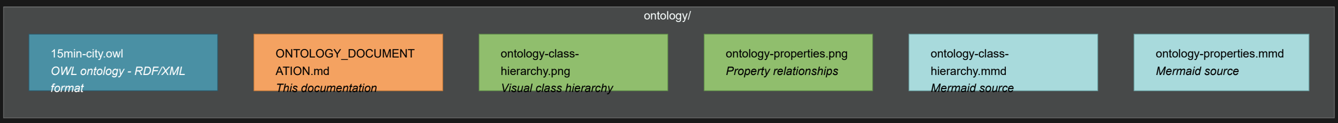
## Property Relationships



# Appendix A: Namespace Prefixes

| Prefix | URI |
|---|---|
| `fmc:` | `http://example.org/ontology/15min-city#` |
| `owl:` | `http://www.w3.org/2002/07/owl#` |
| `rdf:` | `http://www.w3.org/1999/02/22-rdf-syntax-ns#` |
| `rdfs:` | `http://www.w3.org/2000/01/rdf-schema#` |
| `xsd:` | `http://www.w3.org/2001/XMLSchema#` |
| `dc:` | `http://purl.org/dc/elements/1.1/` |
| `skos:` | `http://www.w3.org/2004/02/skos/core#` |

# Appendix B: File Structure



# Appendix C: Tools for Working with the Ontology

| Tool | Purpose | Link |
|---|---|---|
| Protégé | Ontology editor and visualization | https://protege.stanford.edu/ |
| RDFLib | Python RDF library | https://rdflib.readthedocs.io/ |
| Apache Jena | Java RDF framework + Fuseki SPARQL server | https://jena.apache.org/ |
| HermiT | OWL reasoner | http://www.hermit-reasoner.com/ |
| Pellet | OWL 2 reasoner | https://github.com/stardog-union/pellet |
| GraphDB | Triple store with reasoning | https://graphdb.ontotext.com/ |

---

*Generated: 2025-11-29*
*Accompanies: SEMANTIC_MODEL_REPORT.md*