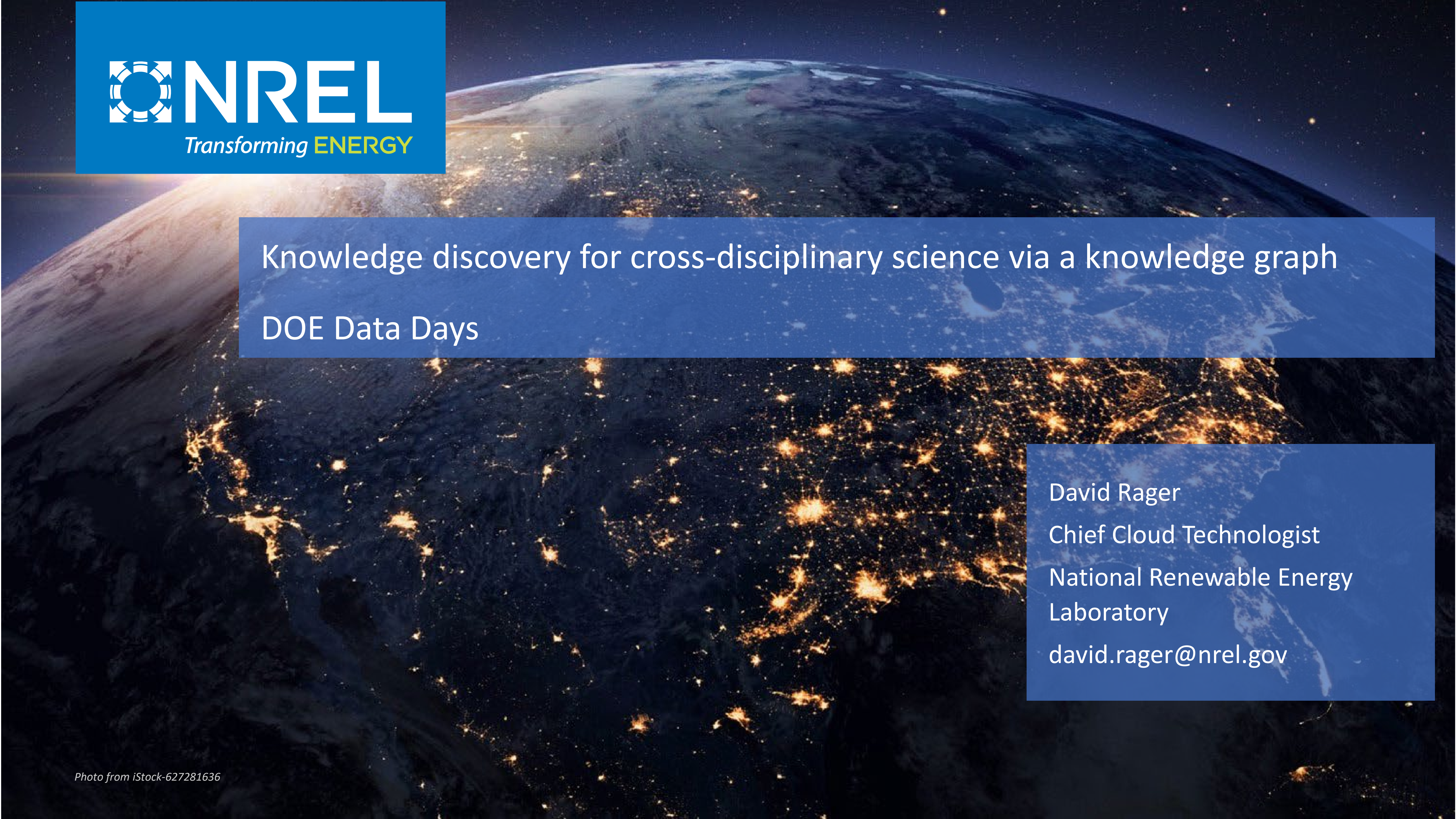


NREL Transforming ENERGY

Knowledge discovery for cross-disciplinary science via a knowledge graph

DOE Data Days

David Rager

Chief Cloud Technologist

National Renewable Energy Laboratory

david.rager@nrel.gov

Photo from iStock-627281636

NREL Knowledge Graph Team

Sagi Zisman

Data Scientist

NREL Knowledge Graph Principal Investigator

Kristi Potter

Data Analysis and Visualizations Lead

Robert White

Data Infrastructure Scientist

David Rager

Chief Cloud Technologist

Graham Johnson

Data Science and Visualization Researcher

Harrison Goldwyn

Data Scientist

Ambarish Nag

Data Scientist

Nalinrat Guba

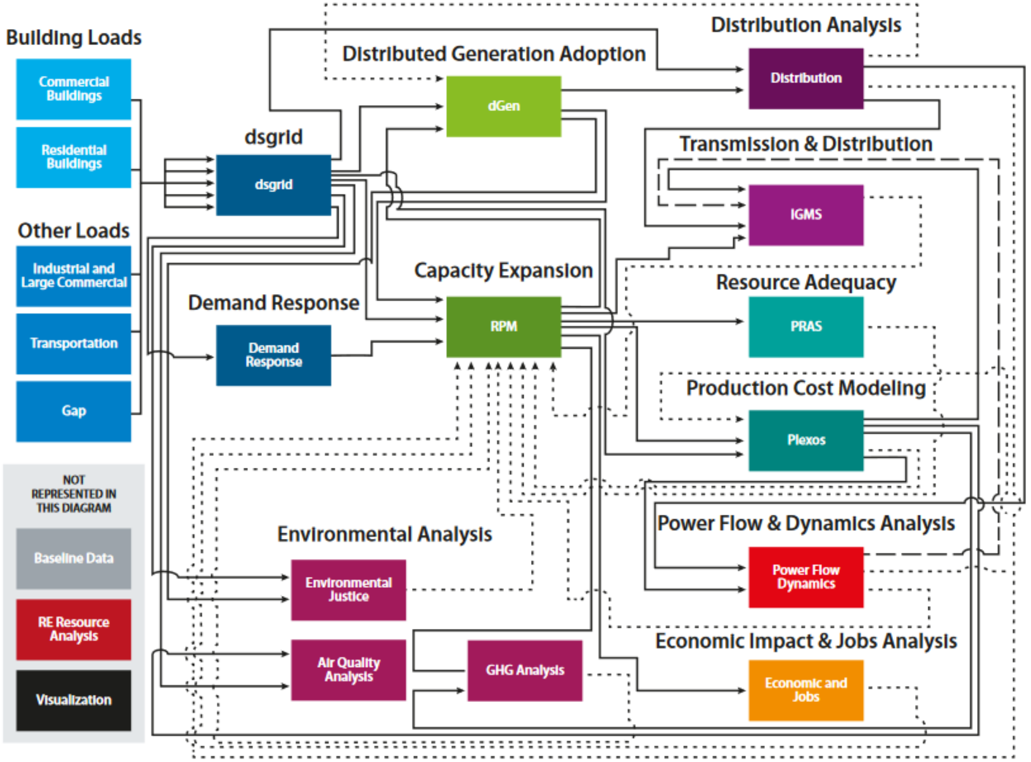
Software Engineer

Additional Contributors:
Dmitry Duplyakin, Data Scientist
Rachel Hurst, Technical Project Lead
Jordan Perr-Sauer, Data Scientist
Sam Molnar, Data Science and Visualization Researcher
Nick Wunder, Software Engineer
Struan Clark, Software Engineer
John Nangle, Project Manager
Kapil Dwadi, Research Engineer
Brennan Holcomb, Intern

Interdisciplinary Science at NREL Use Case: LA100

NREL provided rigorous, integrated techno-economic analysis to the Los Angeles Department of Water and Power (LADWP) through the Los Angeles 100% Renewable Energy Study (LA100).

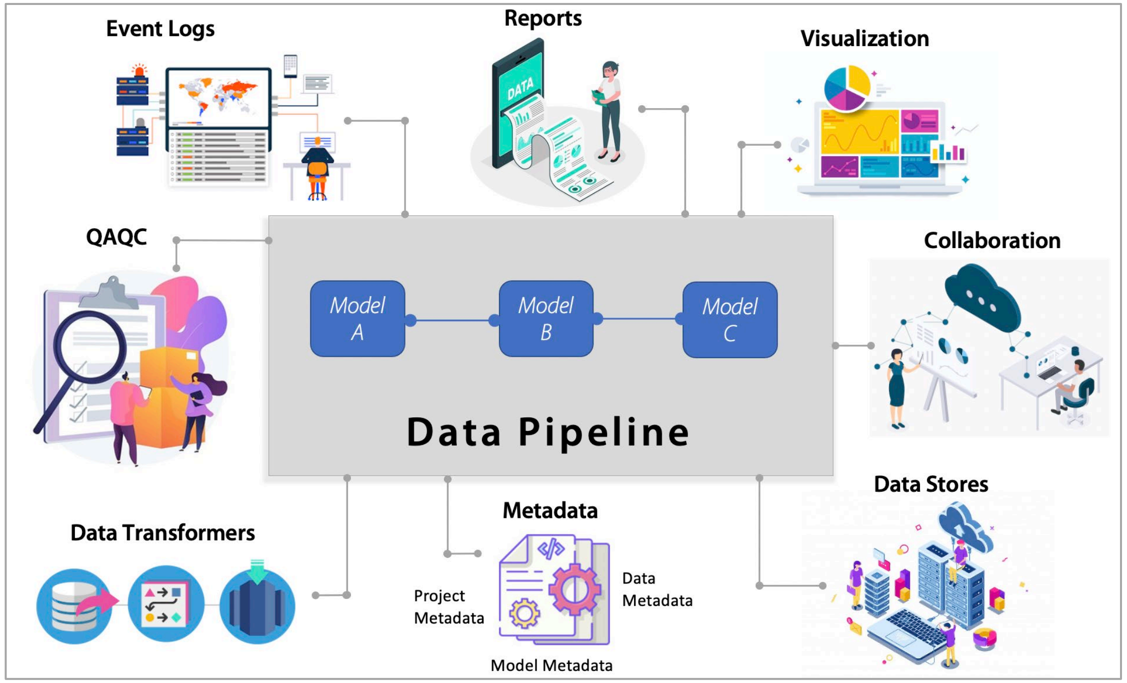
https://www.nrel.gov/analysis/los-angeles-100-percent-renewable-study.html



50 terabytes of load data

More than 100 million simulations

Pipeline for Integrated Projects in Energy Systems (PIPES)



https://github.com/nrel-pipes

The Challenge of Knowledge Organization

Data Diversity:

- Formats
- Sources
- Data Types
- Domains

Complexity of Interlinking:

- Identifying relationships between datasets and research projects, project teams and sponsors

Querying and Accessibility:

- Data locations
- Access methods (APIs, SQL, NoSQL, Python)

Scalability and Evolution:

- Big Data
- Evolving metadata – datasets

Integration of Unstructured Data:

- Augmenting data by addressing related unstructured data

Data Silos

- Data Domains
- Data Locations

Data Quality and Consistency

- Metadata standards
- Metadata sources
- Communicate quality controls
- Provenance and Lineage

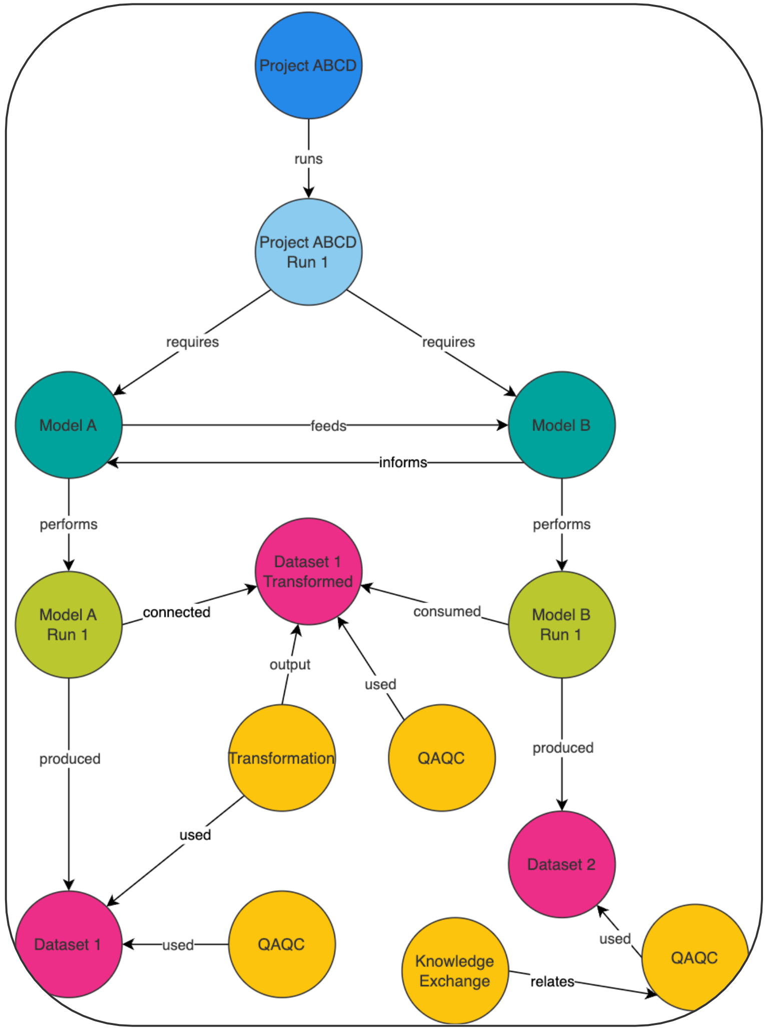
Security and Compliance

- Challenges with access management
- Sensitive data

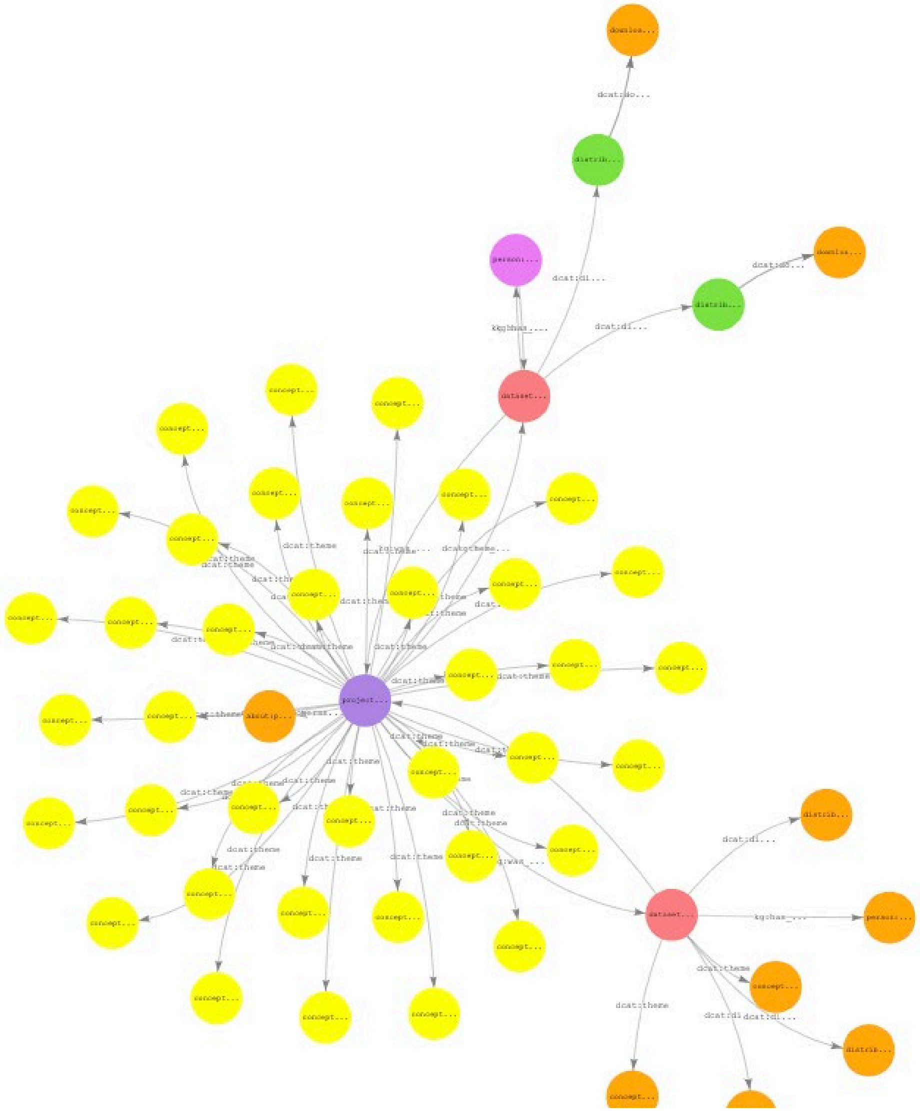
Adaptability to Technological Changes

- Data tooling evolves / specializes

PIPES Lineage and Workflow



Knowledge Graphs & RDF: A Unified Semantic Representation



Knowledge graphs provide a structured approach to organize data cohesively, enabling centralized exploration of knowledge within a domain or institution.

RDF (Resource Description Framework) offers a structured method to semantically represent and organize data in KGs. Information in RDF is represented as triples, which consist of: subject > predicate > object

For example:
David Rager > builds > knowledge graphs.
Knowledge graphs > semantically organize > data

Knowledge derived by combining triples, for example,
we can infer: David Rager constructs methods to organize data semantically.

This semantic representation not only intuitive for humans, but also enables machine learning models to comprehend the context and significance of the data beyond its mere structure.

Combining information from various sources and linking them semantically using triples is foundational to KGs.

Linking Data with KG in NREL

Magenta: Data maintainer (FOAF)
Source Data: Duramat Data Hub provides maintainer relationship.
NREL employee directory provides Title, Lab Contact Info

Red: Databases (DCAT)
Source Data: Duramat Data Hub provides metadata about the datasets, including description, with investigator information, DOIs

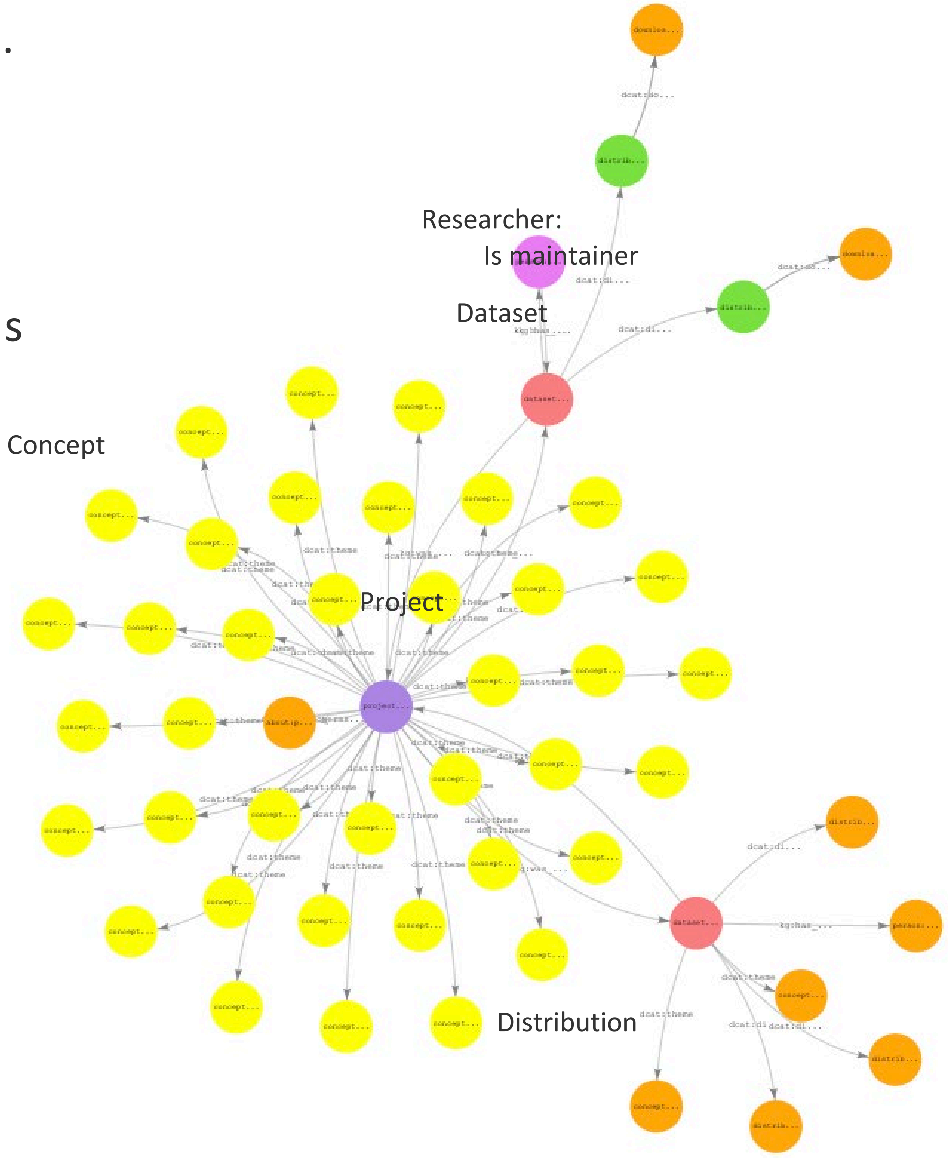
Purple: Project (DCAT)
Source Data: Duramat provides project description and related datasets generated or used by the project.

Orange: Distributions (DCAT)
Source Data: Duramat provides locations of data files for access.

Yellow: Concepts (SKOS)
Source Data: Topic extractions created by GPT from project descriptions.

Relationships – is maintainer, theme, has distribution, …

Two data sources from Duramat, NREL and GPT 4:



Ontology Development and Integration

- **Currently: 187 nodes (Classes), 360 edges (Relationships)**

WebVOWL
1.1.7

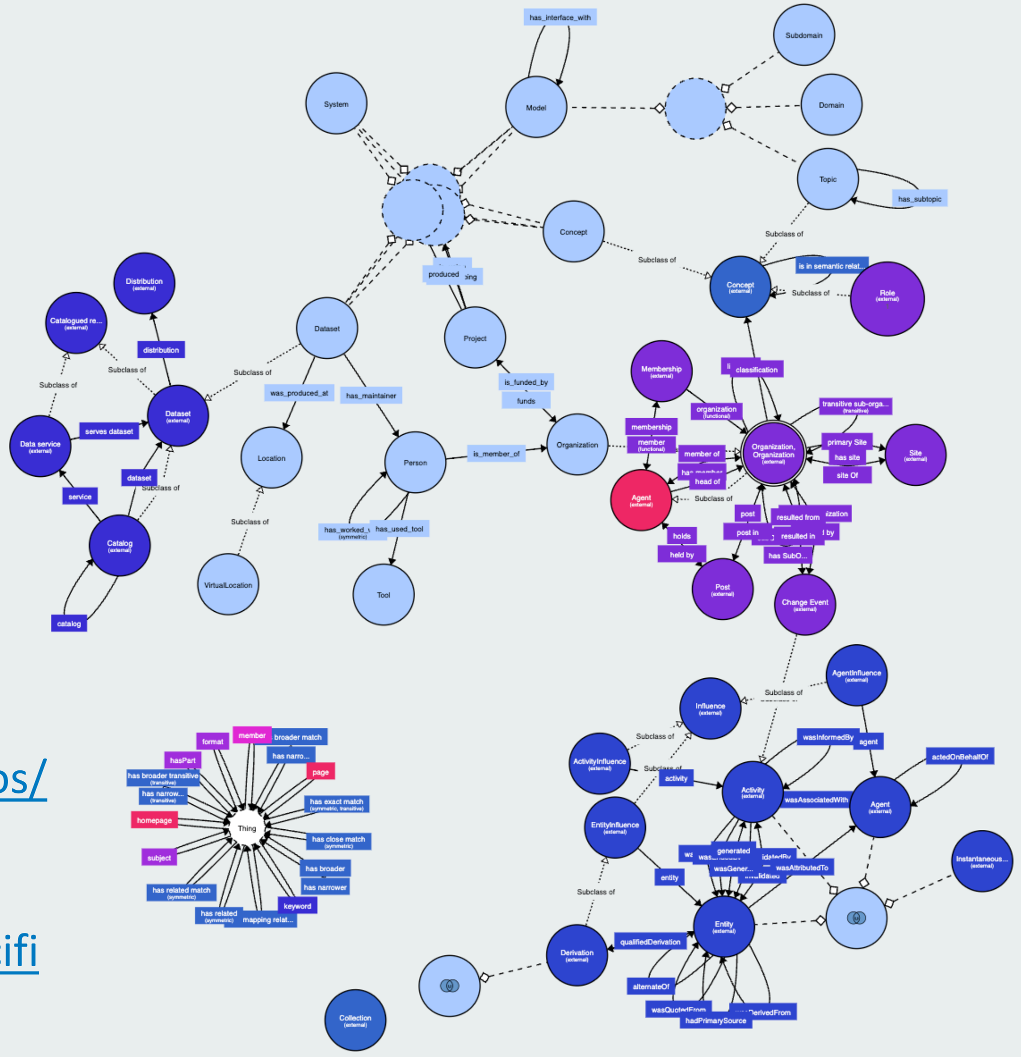
Integrated Ontologies:

DCAT – Data Catalog Vocabulary:
https://www.w3.org/TR/vocab-dcat-2/

FOAF – Friend of a Friend Vocabulary:
http://xmlns.com/foaf/0.1/

SKOS – Simple Knowledge Organization System:
https://www.w3.org/2004/02/skos/

DCMI Metadata Terms:
https://www.dublincore.org/specifications/dublin-core/dcmi-terms/



Ontology Classes and Relations (Sample):

- Subclassing from established ontologies
  - ***OWL:*** Web Ontology Language
  - ***DCAT***: Data Catalogue
  - ***FOAF***: Friend of a Friend
  - ***SKOS***: Simple Knowledge Org. System
- ***Projects***
  - *is_teamed_by* ***: People***
  - *Is_described_by* ***: Description***
- ***Datasets***
  - *Is_stored_in* ***: Location***
  - *Was_produced_by:* ***Person, Proj.***
- ***Models***
- ***Organizations***
  - *Funding Sources*
  - *Teams*
- ***Concepts***

Benefits of Graph Representation of Knowledge

Semantic Context & Explicit Relationships:

- Graphs, particularly when modeled using RDF, capture both entities and their relationships, representing data in a semantically meaningful way and aligning closely with human cognitive patterns.

Power of Ontologies:

- Ontologies define standardized vocabularies, concepts, and relationships within the graph, ensuring consistency, semantic clarity and interoperability across diverse data sources.

Machine Learning & AI Integration:

- Graphs can feed into machine learning algorithms, especially when complemented by the structured nature of RDF and ontological definitions with the structure and semantics of the graph aiding models in discerning complex relationships.

Unified View with RDF & SPARQL:

- RDF offers a standard model for data integration, while SPARQL ensures a unified query mechanism across diverse data types and sources.

Traceability & Provenance:

- Graphs, especially with RDF's structured representation, can track data origins and modifications to ensure data integrity, trustworthiness, and clear understanding of data lineage. See PROV-O ontology: https://www.w3.org/TR/prov-o/

Scalability, Evolution & Ontologies:

- Graphs are modular and adaptable. Ontologies can evolve by adding new concepts or relationships, ensuring the KG remains current.

Going Forward

Holistic Knowledge Graph for NREL staff to use as the basis for a **Data Mesh** approach to data management, interface networking between our experts, and helping define a metadata structure for tracking projects, related funding, related outputs (publications, visualizations, datasets, reports).

NREL may publish a **public version of the Knowledge Graph** with an eye towards participating in the NSF Prototype Open Knowledge Network that will be composed of knowledge graphs from NASA, NIH, NIJ, NOAA, USGS, USDA, and others to leverage knowledge integration and distribution methods.
https://www.nsf.gov/pubs/2023/nsf23571/nsf23571.htm

Develop Scientific Knowledge Graphs to provide metadata and described data within a graph structure to leverage the benefits of a Knowledge Graph approach including link prediction or node classification as means of knowledge discovery.

Develop Domain Specific Knowledge Graphs for research or information dissemination.

Exploring Hybrid / Multi-Modal Knowledge Systems including graph and geospatial data, virtual knowledge graphs, expanding GPT functions to other systems using graph-based validations.

We welcome anyone interested in any of these approaches to reach out to us. We'd love to discuss use cases and approaches.

Thank you! David.rager@nrel.gov

Addendum Enabling Smart Queries with GPT

NREL Proof of Concept using LangChain, GPT-4 and ChainLit to interact with the knowledge graph.

The process for this proof-of-concept (POC) is as follows (you can follow this process in debug mode):

1. The user asks a question about information in the Knowledge Graph.
2. The LLM identifies the goal and related subject of the user's question.
3. The LLM assesses whether it has sufficient information to answer the question. If not, it selects and executes an appropriate function to gather more information.
4. The LLM re-evaluates if it has enough information to answer the question. If not, it will select and execute another function for additional data.
5. This loop continues until the LLM has obtained the required information, or LLM determines it cannot acquire the needed data, or the request times out.
6. The returned information is either raw JSON from OpenSearch or triplets returned via parameterized SPARQL queries.
7. The gathered information is summarized for the user.
8. Both the conversation and data are stored in memory, allowing the user to refine their queries later.
9. The LLM understands both OpenSearch output and RDF formats. Our implementation of GPT-4 has also been trained on functional use.