aws *LDBC TUC June 2023*

# Why limit yourself to LPG when you can do RDF, too

**Dr. Ora Lassila**
Principal Technologist
Amazon Neptune

aws *LDBC TUC June 2023*

# Why limit yourself to RDF when you can do LPG, too

**Dr. Ora Lassila**
Principal Technologist
Amazon Neptune

# Who am I?

Principal Technologist in the Amazon Neptune graph database team

Co-chair of the W3C RDF-star Working Group (2022-)

Co-author of the original RDF specification (1997-1999)

Co-author of the seminal paper on the Semantic Web (2001)

Recipient of the 1st ISWC “10-year award” (2011)

W3C Fellow (1996-1997)

Elected member of the W3C Advisory Board (1998-2013)

Grand Prize Winner, Usenix Obfuscated C Code Contest (1989)

Education: Ph.D CS, Helsinki University of Technology



aws

# Game plan

1. Knowledge graphs, interoperability, and the Semantic Web
2. RDF vs. LPG
3. Project OneGraph



aws

# A brief history of graphs and ontologies

1730s: Graph theory (Euler)

1950s and onwards: Graphs as the essential underpinning of computer science

1960s: Social networks, “small-world experiment”, Erdős number (Milgram et al)

1960s-1970s: Network databases (CODASYL), semantic networks (Quillian et al)

3rd Century BCE: Categories & logic (Aristotle)

1730s: Taxonomical classification of plants and animals (Linnaeus)

1870s: Library classification (Dewey)

1900: Semantics, ontology and logic (Husserl)

1970s-1990s: Predicate logic as the foundation of Knowledge Representation (Hayes et al)

1997 and onwards: The Semantic Web, RDF, OWL, etc. (Lassila et al)

Today: Modern knowledge graphs and graph databases



aws

# RDF, SPARQL, the Semantic Web, and Knowledge Graphs

The W3C “Semantic Web stack” forms the basis of many modern knowledge graph systems

- **RDF:** First graph standard (1999)
- **SPARQL:** First graph query language standard (2004)

These standards were intended for data interchange

Knowledge graphs are often seen as
a “way out of the silos” for data

*however…*



aws

# Won’t get fooled again…

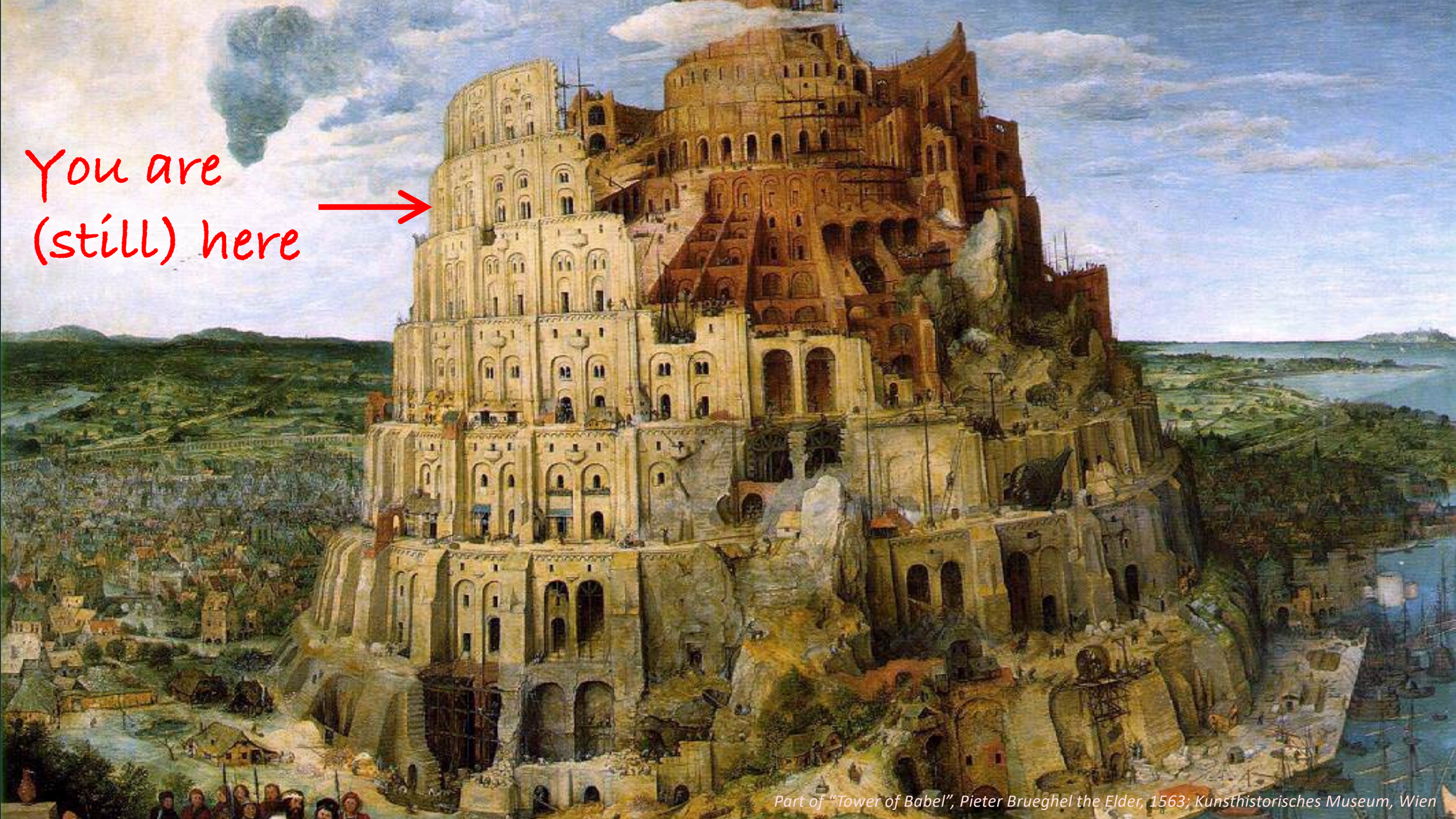
## “Meet the new silos… just like the old silos”

**Old silos:** Single-application -controlled data, at best behind a bespoke API

**New silos:** Single-purpose knowledge graphs built without interoperability and interlinking in mind



aws




*Part of "Tower of Babel", Pieter Brueghel the Elder, 1563; Kunsthistorisches Museum, Wien*

# Why do we need interoperability (and what does it mean)?

Common format enables information interchange

Common query language enables interworking and frees users from ”lock-in”

***BUT…***

We also need common, shared semantics

Special attention should be paid to how we identify things

Standards or technologies not designed for sharing and interchange of semantics should be rejected off-hand

- because they simply just reinforce the old “silo mindset”



aws

# A graph is a graph is a graph…?

For knowledge graphs, you typically need what the Semantic Web technologies offer

Other graph applications often treat the graph as a very large, potentially complex data structure

# A graph is a graph is a graph…?

For knowledge graphs, you typically need what the Semantic Web technologies offer


Other graph applications often treat the graph as a very large, potentially complex data structure

**Graph as a logical representation vs. graph as a data structure**

*RDF (and friends)*

*LPGs*

# RDF or LPG…?

This question (“the rift”) plagues the graph community

With Amazon Neptune, we chose to support both, to give customers a choice

- unfortunately, they have to choose (and this causes confusion)
- the choice limits what you can do (e.g., what query language you can use)

Both graph models have their pros and cons…



aws

# RDF

## Good features

- Graph merging
- Strong, global identifiers
- Schema/ontology language
- Self-describing data
- Standardized interchange formats
- Formal semantics that support reasoning
- Federated queries

## Missing features

- Programmer “friendliness”, good integration with programming languages
- Usable composite datatypes
- Path discovery
- Recursive queries
- Composable queries



aws

~~RDF~~ LPG

## ~~Good~~ Missing features

- Graph merging
- Strong, global identifiers
- Schema/ontology language
- Self-describing data
- Standardized interchange formats
- Formal semantics that support reasoning
- Federated queries

## ~~Missing~~ Good features

- Programmer “friendliness”, good integration with programming languages
- Usable composite datatypes
- Path discovery
- Recursive queries
- Composable queries



aws

# Project “OneGraph”

What if we could have the “best of both worlds”…?

- use all of the good features of RDF (and SPARQL) with LPGs, without having to reinvent them (and vice versa)
- no more complaints that RDF does not have “edge properties”
- mitigate SPARQL’s lack of path discovery
- Gremlin queries over RDF! (GQL over RDF, too)
- ontologies for LPG
- reasoning…
- etc.

Big goal: “graph interoperability” (i.e., no more confusion)



aws

# Project “OneGraph”

OneGraph (1G) is a metamodel that “unifies” RDF, RDF-star, and LPGs

Each of the existing metamodels is a “lower-dimensional view” of 1G data

Consequently, roundtrips:

- RDF → 1G → RDF: lossless, but
- 1G → RDF → 1G: not necessarily lossless
- etc.

There are several technical and definitional hurdles to accomplish this

- the main practical challenge is that RDF and LPGs are used differently
- more information: semantic-web-journal.net/system/files/swj3273.pdf



aws

~~RDF~~ ~~LPG~~ OneGraph

# END GOAL

## Good features

- Graph merging
- Strong, global identifiers
- Schema/ontology language
- Self-describing data
- Standardized interchange formats
- Formal semantics that support reasoning
- Federated queries

## More good ~~Missing~~ features

- Programmer “friendliness”, good integration with programming languages
- Usable composite datatypes
- Path discovery
- Recursive queries
- Composable queries



aws

# Thank you!

Contact:

- ora@amazon.com
- Twitter: @oralassila



aws