Enforcing Ontological Commitments:

From Process Mining to Dependent Types

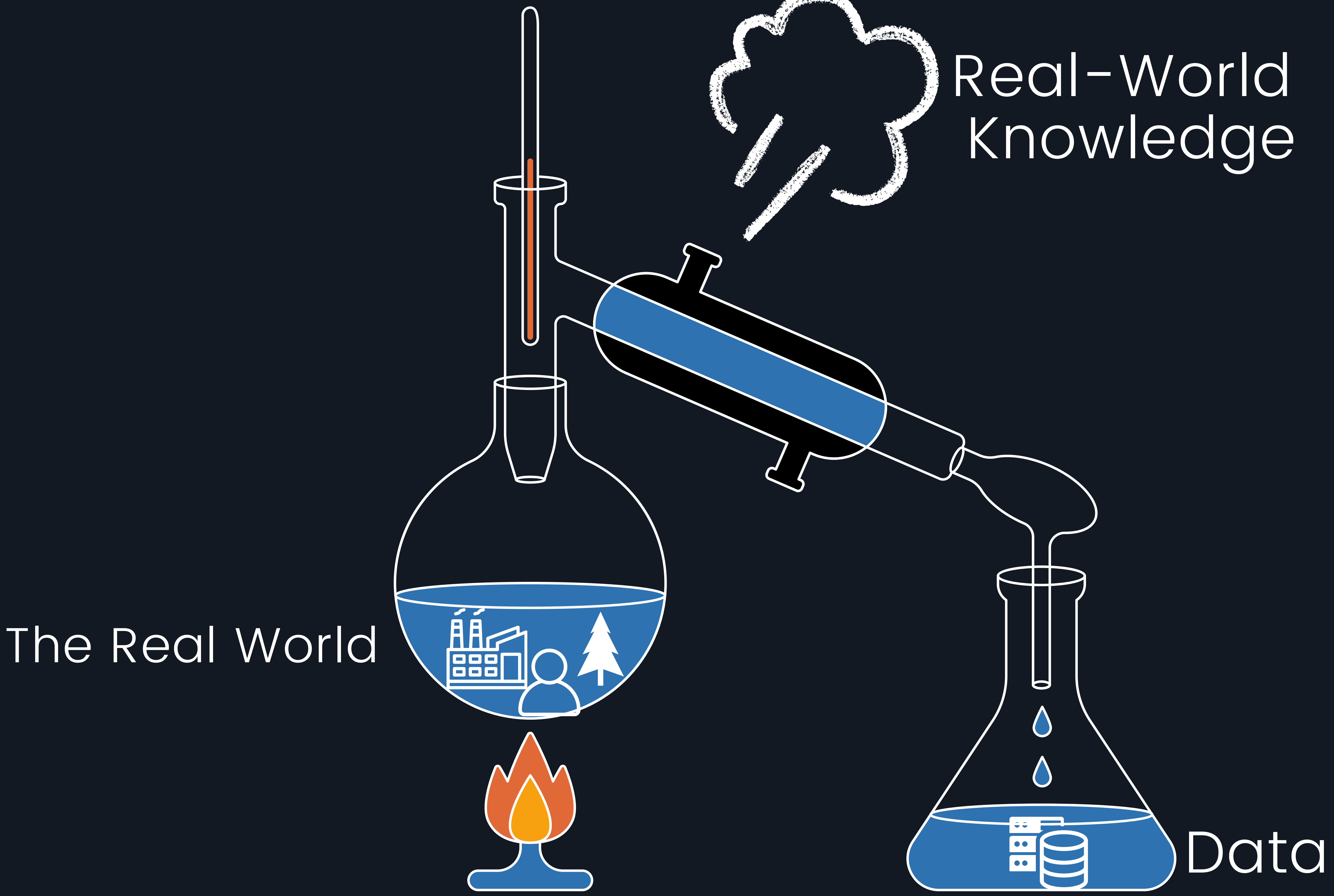
Riley Moher
31.10.2025
University at Buffalo

UNIVERSITY OF TORONTO

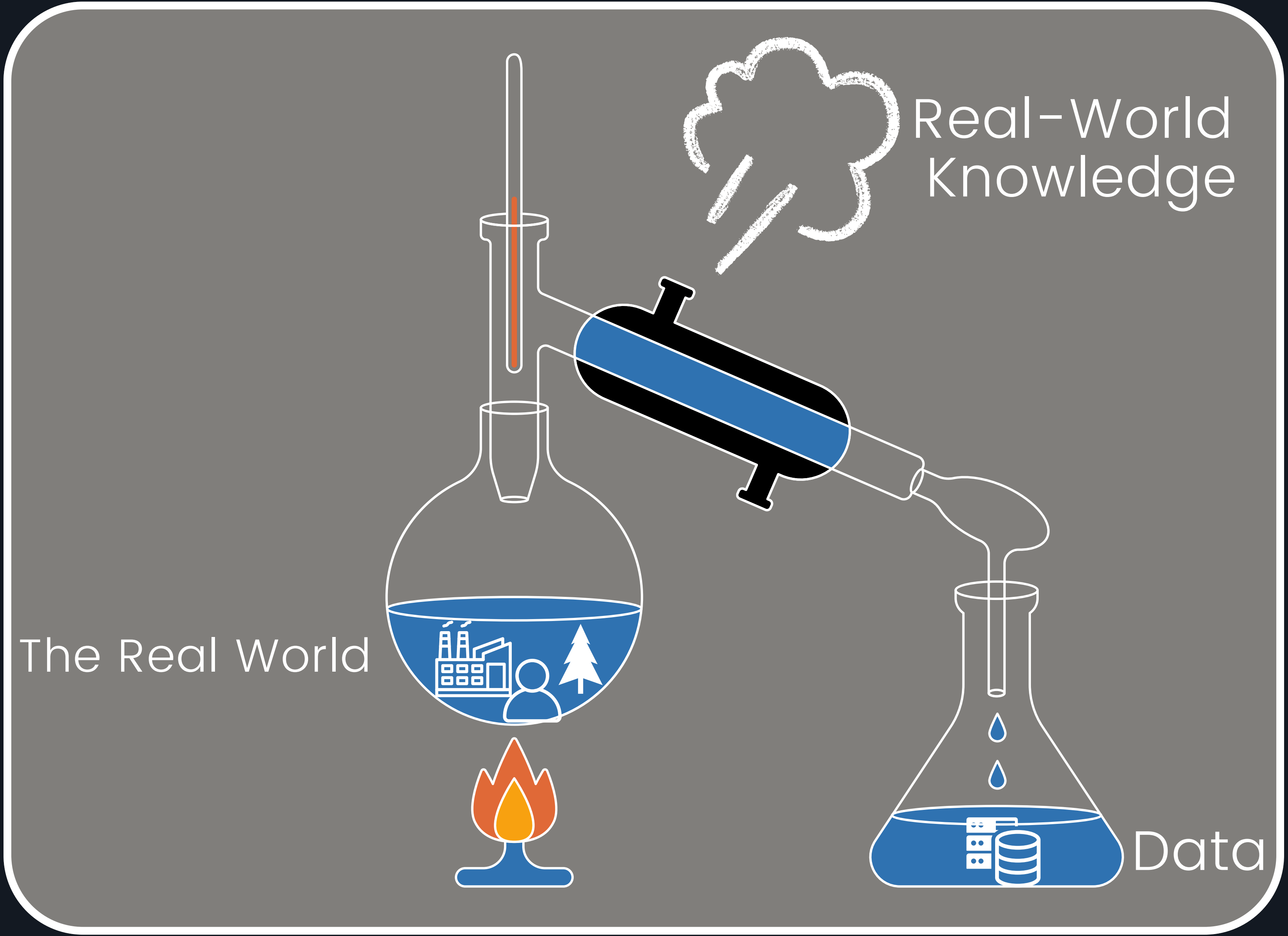
Semantic Technologies Lab

The Semantic Gap

The Semantic Gap

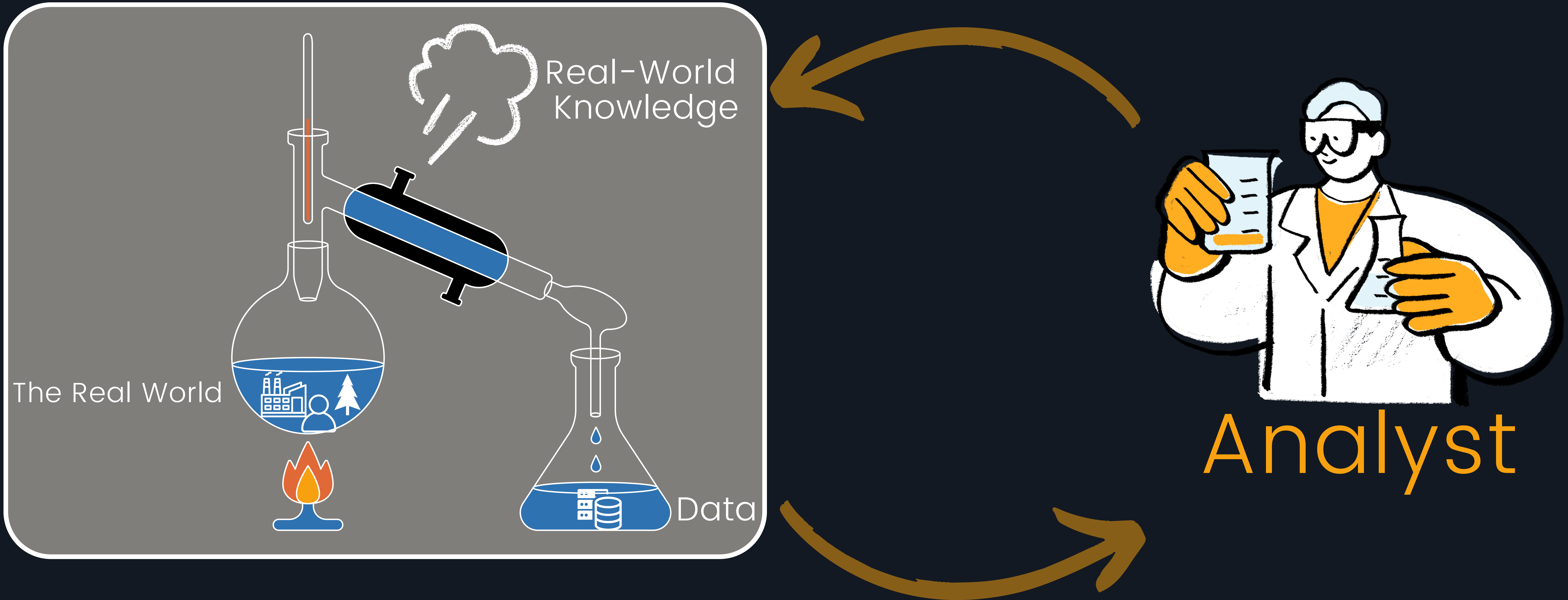


The Semantic Gap

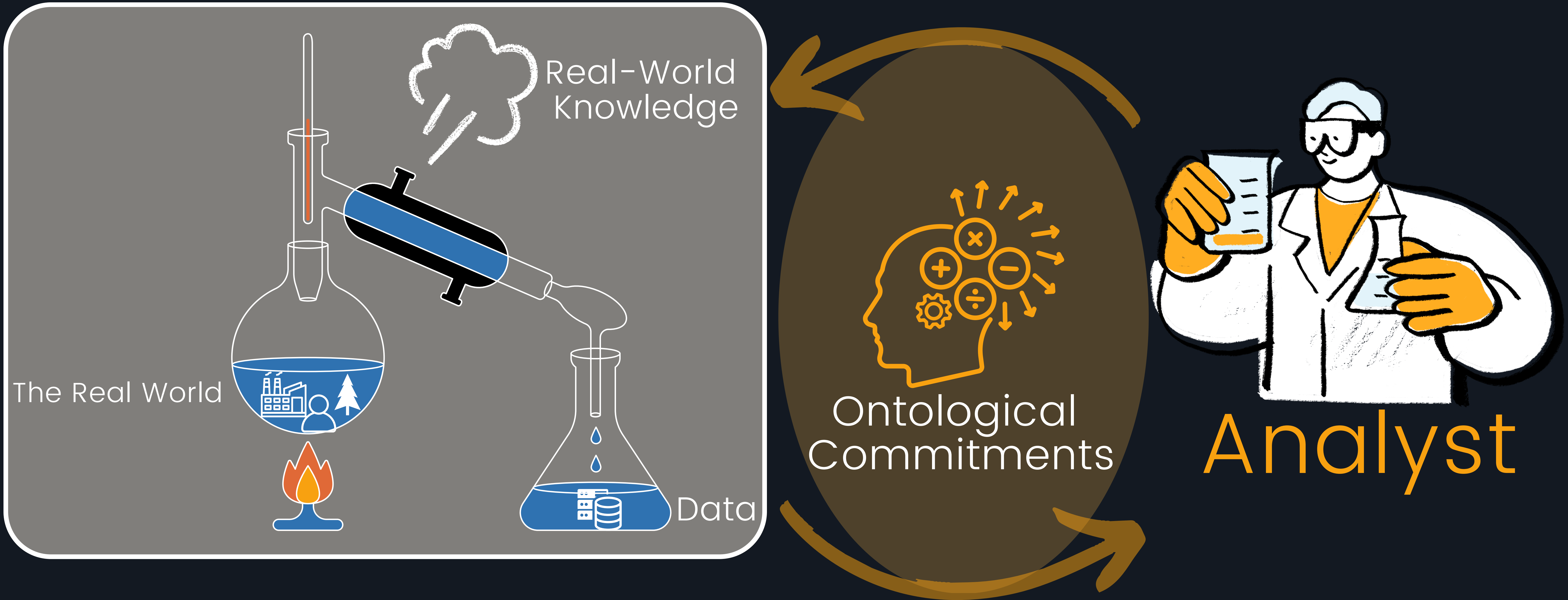


Analyst

The Semantic Gap



The Semantic Gap



Bridging The Semantic Gap

- Data analysis relies on external ontological commitments to make sense of underspecified & incomplete data

Bridging The Semantic Gap

- Data analysis relies on external ontological commitments to make sense of underspecified & incomplete data
- Goal: integrate these commitments into data-driven systems for reusability, transparency, verifiability

Bridging The Semantic Gap

- Data analysis relies on external ontological commitments to make sense of underspecified & incomplete data

- Goal: integrate these commitments into data-driven systems for reusability, transparency, verifiability

- Process mining
  - Ontological challenges of events
  - Formalizing event data interpretations

Bridging The Semantic Gap

- Data analysis relies on external ontological commitments to make sense of underspecified & incomplete data

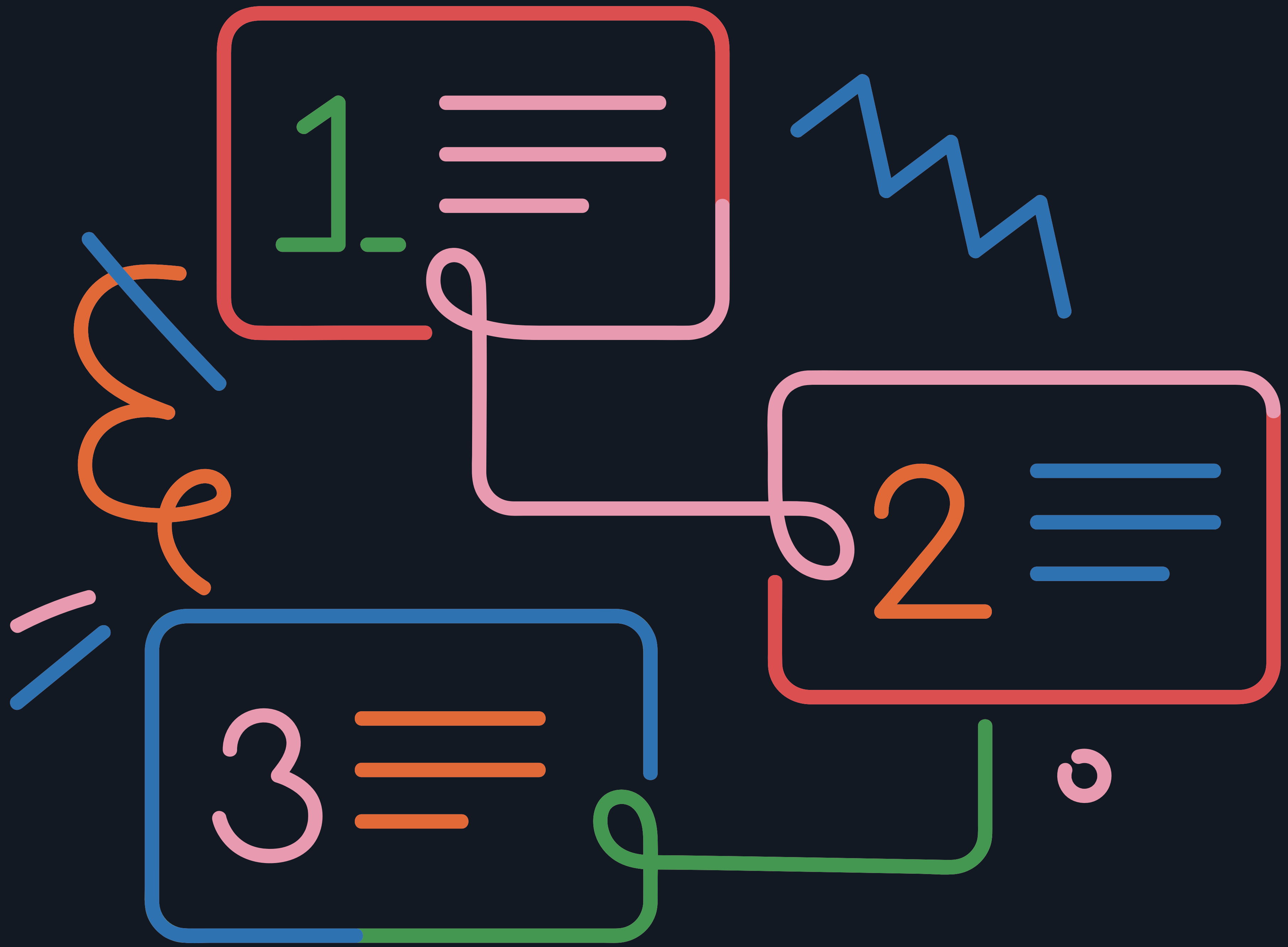
- Goal: integrate these commitments into data-driven systems for reusability, transparency, verifiability

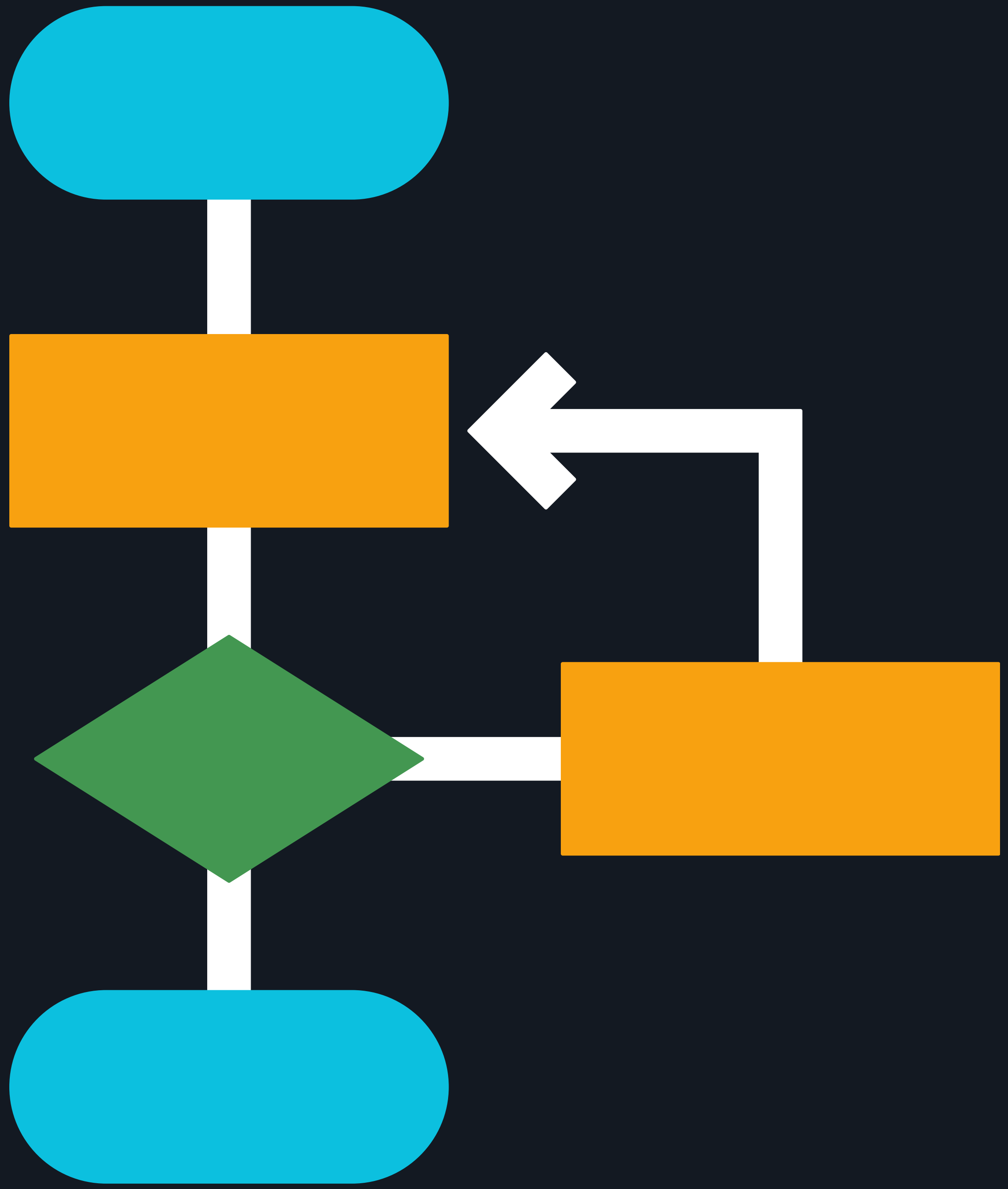
- Process mining
  - Ontological challenges of events
  - Formalizing event data interpretations

- Datatypes & data science
  - The limitations of simple datatypes
  - How to make types meaningful

Part 1:
Mining for Meaning

Business Processes





Processes and Data

Process Data (Event Logs)

Business Rules, Process Knowledge

Models, Compliance Checks, Insights

- Process mining as the analysis engine for event data

Process Mining

Timestamp	Event	Patient
12:02	Patient Intake	John Smith
12:05	Patient Intake	John Smith
12:06	Diagnostic	John Smith

- Issue: Patient Intake should only occur once for the same patient.

Process Mining

Timestamp	Event	Patient
12:02	Patient Intake	John Smith
12:05	Patient Intake	John Smith
12:06	Diagnostic	John Smith

- **Issue**: Patient Intake should only occur once for the same patient.
- **Solution**: The event label was wrong

Process Mining

Timestamp	Event	Patient
12:02	Patient Intake	John Smith
12:05	Additional Screening	John Smith
12:06	Diagnostic	John Smith

- **Issue**: Patient Intake should only occur once for the same patient.
- **Solution**: The event label was wrong

Process Mining

Timestamp	Event	Patient
12:02	Patient Intake	John Smith
12:05	Additional Screening	John Smith
12:06	Diagnostic	John Smith

- **Issue**: Patient Intake should only occur once for the same patient.
- **Solution**: The event label was wrong (external knowledge)

Process Mining: Events

- “Awaiting Assignment”
- “Document Under Review“
- “Review Document”
- “Wait User Status”

Process Mining: Events

- “Awaiting Assignment”
- “Document Under Review“
- “Review Document”
- “Wait User Status”

These are all simply “events” in an event log

Process Mining: Events

- “Awaiting Assignment”
- “Document Under Review“
- “Review Document”
- “Wait User Status”

These are all simply “events” in an event log

But they have very different process implications

Process Mining: Events

- “Awaiting Assignment”
- “Document Under Review“
- “Review Document”
- “Wait User Status”

These are all simply “events” in an event log

But they have very different process implications

Events are heavily overloaded and require interpretation for analysis

Process Mining

Process Data
(Event Logs)

Business Rules,
Process Knowledge

Models,
Compliance Checks,
Insights

Process Mining

Process Data (Event Logs)

Business Rules, Process Knowledge

Models, Compliance Checks, Insights

1. How does a process ontology fit?

Process Mining

Process Data
(Event Logs)

Business Rules,
Process Knowledge

Models,
Compliance Checks,
Insights

1. How does a process ontology fit?
2. What tools enable that fit?

Process Ontology Application: Challenges

- Current ontology applications are more representational, access-driven, and conceptual

Process Ontology Application: Challenges

- Current ontology applications are more representational, access-driven, and conceptual
- Application should instead focus on formalizing the ontological commitments of data analysis

Process Ontology Application: Challenges

- Current ontology applications are more representational, access-driven, and conceptual
- Application should instead focus on formalizing the ontological commitments of data analysis
- Operational realization capture this notion

Process Ontology Application: Challenges

- Current ontology applications are more representational, access-driven, and conceptual
- Application should instead focus on formalizing the ontological commitments of data analysis
- Operational realization capture this notion
- How do we apply operational realization to process mining?

Ontology-Driven Process Mining

Raw Event Data

Ad-Hoc Interpretations

Business Questions

Business Validations

Knowledge Base

Data Theory

Logical Queries

Logical Proofs

Ontology-Driven Process Mining

Raw Event Data

Ad-Hoc Interpretations

Business Questions

Business Validations

Map

Formalize

Analyze

Knowledge Base

Data Theory

Logical Queries

Logical Proofs

Mapping Event Log Data

- Tabular data from event logs becomes ground facts

Timestamp	Event	Lifecycle Transition
14:20	Process Application	Start
15:30	Credit Check	Complete
15:38	Credit Check	Start

Mapping Event Log Data

- Tabular data from event logs becomes ground facts

Timestamp	Event	Lifecycle Transition
14:20	Process Application	Start
15:30	Credit Check	Complete
15:38	Credit Check	Start

Event(e2)
hasActivity(e0, creditCheck)
hasTransition(e2, complete)

Mapping Event Log Data

- Tabular data from event logs becomes ground facts

Timestamp	Event	Lifecycle Transition
14:20	Process Application	Start
15:30	Credit Check	Complete
15:38	Credit Check	Start

Event(e2)
hasActivity(e0, creditCheck)
hasTransition(e2, complete)

Event(e3)
hasActivity(e3, creditCheck)
hasTransition(e3, start)

Ground Data

Mapping Event Log Data

Ground Data

Event(e2)
hasActivity(e0, creditCheck)
hasTransition(e2, complete)

Event(e3)
hasActivity(e3, creditCheck)
hasTransition(e3, start)

Mapping Event Log Data

Ground Data

Event(e2)
hasActivity(e0, creditCheck)
hasTransition(e2, complete)

Event(e3)
hasActivity(e3, creditCheck)
hasTransition(e3, start)

+

Ontology Translations

"Two events sharing an activity each with a start and end transition indicates an acitivty occurrence"

$$\forall e_s \forall e_e \forall a \Big(\texttt{Event}(e_s) \wedge \texttt{Event}(e_e) \wedge$$
$$\texttt{hasActivity}(e_s, a) \wedge \texttt{hasActivity}(e_e, a) \wedge$$
$$\texttt{hasTransition}(e_s, \mathsf{start}) \wedge \texttt{hasTransition}(e_e, \mathsf{complete}) \rightarrow$$
$$\exists o \, (\texttt{activity\_occurrence}(o) \wedge$$
$$\texttt{occurrence\_of}(o, a) \wedge$$
$$\texttt{beginOf}(o, e_s) \wedge$$
$$\texttt{endOf}(o, e_e))\Big).$$

Mapping Event Log Data

Ground Data

Event(e2)
hasActivity(e0, creditCheck)
hasTransition(e2, complete)

Event(e3)
hasActivity(e3, creditCheck)
hasTransition(e3, start)

Ontology Translations

"Two events sharing an activity each with a start and end transition indicates an acitivty occurrence"

$$\forall e_s\, \forall e_e\, \forall a\, \Big(\texttt{Event}(e_s) \wedge \texttt{Event}(e_e) \wedge \\ \texttt{hasActivity}(e_s, a) \wedge \texttt{hasActivity}(e_e, a) \wedge \\ \texttt{hasTransition}(e_s, \mathsf{start}) \wedge \texttt{hasTransition}(e_e, \mathsf{complete}) \;\rightarrow \\ \exists o\, (\texttt{activity\_occurrence}(o) \wedge \\ \texttt{occurrence\_of}(o, a) \wedge \\ \texttt{beginOf}(o, e_s) \wedge \\ \texttt{endOf}(o, e_e))\Big).$$

Mapping Event Log Data

Ground Data

Event(e2)
hasActivity(e0, creditCheck)
hasTransition(e2, complete)

Event(e3)
hasActivity(e3, creditCheck)
hasTransition(e3, start)

Ontology Translations

"Two events sharing an activity each with a start and end transition indicates an acitivty occurrence"

$$\forall e_s \, \forall e_e \, \forall a \, \Big(\texttt{Event}(e_s) \wedge \texttt{Event}(e_e) \wedge$$
$$\texttt{hasActivity}(e_s, a) \wedge \texttt{hasActivity}(e_e, a) \wedge$$
$$\texttt{hasTransition}(e_s, \text{start}) \wedge \texttt{hasTransition}(e_e, \text{complete}) \;\rightarrow$$
$$\exists o \, (\texttt{activity\_occurrence}(o) \wedge$$
$$\texttt{occurrence\_of}(o, a) \wedge$$
$$\texttt{beginOf}(o, e_s) \wedge$$
$$\texttt{endOf}(o, e_e)) \Big).$$

Mapping Event Log Data

Ground Data

Event(e2)
hasActivity(e0, creditCheck)
hasTransition(e2, complete)

Event(e3)
hasActivity(e3, creditCheck)
hasTransition(e3, start)

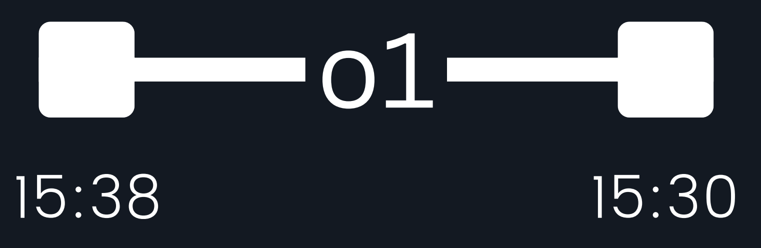
Ontology Translations

"Two events sharing an activity each with a start and end transition indicates an acitivty occurrence"

$$\forall e_s \, \forall e_e \, \forall a \, \Big(\texttt{Event}(e_s) \wedge \texttt{Event}(e_e) \wedge$$
$$\texttt{hasActivity}(e_s, a) \wedge \texttt{hasActivity}(e_e, a) \wedge$$
$$\texttt{hasTransition}(e_s, \text{start}) \wedge \texttt{hasTransition}(e_e, \text{complete}) \rightarrow$$
$$\exists o \, \big(\texttt{activity\_occurrence}(o) \wedge$$
$$\texttt{occurrence\_of}(o, a) \wedge$$
$$\texttt{beginOf}(o, e_s) \wedge$$
$$\texttt{endOf}(o, e_e) \big) \Big).$$

Translated Data

occurrence(o1)
beginOf(o1, 15:38)
endOf(o1, 15:30)

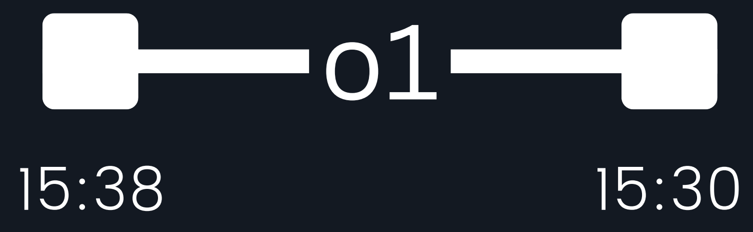


Event Log Data Quality

Map
Formalize
Analyze

Translated Data

occurrence(o1)
beginOf(o1, 15:38)
endOf(o1, 15:30)



Event Log Data Quality

Map
Formalize
Analyze

Translated Data

occurrence(o1)
beginOf(o1, 15:38)
endOf(o1, 15:30)

Event Log Data Quality

- Map
- Formalize
- Analyze

Translated Data

occurrence(o1)
beginOf(o1, 15:38)
endOf(o1, 15:30)

Ontology

"Activity occurrences start points are less than or equal to their end points"

$$(\forall o\, \text{activity\_occurrence}(o) \implies (\text{begin\_of}(o) \leq \text{end\_of}(o)))$$

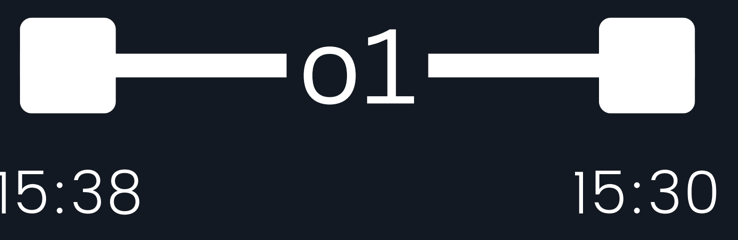
$$t_1 \leq t_2$$

Event Log Data Quality

- Map
- Formalize
- Analyze

Translated Data

occurrence(o1)
beginOf(o1, 15:38)
endOf(o1, 15:30)



+

Ontology

"Activity occurrences start points are less than or equal to their end points"

$$(\forall o\, \text{activity\_occurrence}(o) \implies (\text{begin\_of}(o) \leq \text{end\_of}(o)))$$

$$t_1 \leq t_2$$

Proof of Inconsistency

$$t_1 = 15:38$$

$$t_2 = 15:30$$

$$\boxed{\begin{array}{c} t_1 > t_2 \\ t_1 \leq t_2 \end{array}}$$

Getting Specific: Knowledge Patterns

- How do we extend our process ontology to model domain process knowledge?

Getting Specific: Knowledge Patterns

- How do we extend our process ontology to model domain process knowledge?
- Focusing on standardization and tractability of these extensions

Getting Specific: Knowledge Patterns

- How do we extend our process ontology to model domain process knowledge?
- Focusing on standardization and tractability of these extensions
- Use knowledge patterns to create a data theory

Knowledge Patterns

- “When a fragile object is dropped, it breaks”
- “Patient Intake should only occur once for the same patient.”

Knowledge Patterns

- “When a fragile object is dropped, it breaks”
- “Patient Intake should only occur once for the same patient.”

Knowledge Patterns

- “When a fragile object is dropped, it breaks”

State Based Effect

- “Patient Intake should only occur once for the same patient.”

Occurrence Constraint

Knowledge Patterns

- "When a fragile object is dropped, it breaks"

  State Based Effect

- "Patient Intake should only occur once for the same patient."

  Occurrence Constraint

Knowledge Patterns

- State-Based Effects (SBE)
- "When a fragile object is dropped, it breaks"
- While some initial condition holds, and an occurrence happens, some resulting condition holds afterwards

Knowledge Patterns

- State-Based Effects (SBE)
- “When a fragile object is dropped, it breaks”
- While some initial condition holds, and an occurrence happens, some resulting condition holds afterwards
- Patterns abstract common process knowledge
- SBE(c1, a, c2)

Knowledge Patterns

- State-Based Effects (SBE)

$$SBE(a, f_1, f_2) := (\forall o)\, occurrence\_of(o, a) \wedge prior(f_1, o) \implies holds(f_2, o)$$

$$(\forall o)\, occurrence\_of(o, drop) \wedge prior(fragile, o) \implies holds(broken, o)$$

Knowledge Patterns

- State-Based Effects (SBE)

$$SBE(a, f_1, f_2) := (\forall o)\ occurrence\_of(o, a) \land prior(f_1, o) \implies holds(f_2, o)$$

Pattern Definition - abstracts extensions of the process ontology

$$(\forall o)\ occurrence\_of(o, drop) \land prior(fragile, o) \implies holds(broken, o)$$

Knowledge Patterns

- State-Based Effects (SBE)

$$SBE(a, f_1, f_2) := (\forall o)\ occurrence\_of(o, a) \wedge prior(f_1, o) \implies holds(f_2, o)$$

Pattern Definition - abstracts extensions of the process ontology

$$(\forall o)\ occurrence\_of(o, drop) \wedge prior(fragile, o) \implies holds(broken, o)$$

Pattern Instatiation - concrete extension of the process ontology

Knowledge Patterns

- State-Based Effects (SBE)

$$SBE(a, f_1, f_2) := (\forall o)\ occurrence\_of(o, a) \wedge prior(f_1, o) \implies holds(f_2, o)$$

Pattern Definition - abstracts extensions of the process ontology

$$(\forall o)\ occurrence\_of(o, drop) \wedge prior(fragile, o) \implies holds(broken, o)$$

Pattern Instatiation - concrete extension of the process ontology

Knowledge Patterns

- State-Based Effects (SBE)

$$SBE(a, f_1, f_2) := (\forall o)\ occurrence\_of(o, a) \land prior(f_1, o) \implies holds(f_2, o)$$

Pattern Definition - abstracts extensions of the process ontology

$$(\forall o)\ occurrence\_of(o, drop) \land prior(fragile, o) \implies holds(broken, o)$$

Pattern Instatiation - concrete extension of the process ontology

T-Box?

Knowledge Patterns

- State-Based Effects (SBE)

$$SBE(a, f_1, f_2) := (\forall o)\ occurrence\_of(o, a) \wedge prior(f_1, o) \implies holds(f_2, o)$$

Pattern Definition - abstracts extensions of the process ontology

$$(\forall o)\ occurrence\_of(o, drop) \wedge prior(fragile, o) \implies holds(broken, o)$$

Pattern Instatiation - concrete extension of the process ontology

T-Box?
A-Box?

Knowledge Patterns

- State-Based Effects (SBE)

$$SBE(a, f_1, f_2) := (\forall o)\, occurrence\_of(o, a) \wedge prior(f_1, o) \implies holds(f_2, o)$$

Pattern Definition - abstracts extensions of the process ontology

$$(\forall o)\, occurrence\_of(o, drop) \wedge prior(fragile, o) \implies holds(broken, o)$$

Pattern Instatiation - concrete extension of the process ontology

T-Box?
A-Box?

Domain Ontology?

Knowledge Patterns

- State-Based Effects (SBE)

$$SBE(a, f_1, f_2) := (\forall o)\ occurrence\_of(o, a) \wedge prior(f_1, o) \implies holds(f_2, o)$$

Pattern Definition - abstracts extensions of the process ontology

$$(\forall o)\ occurrence\_of(o, drop) \wedge prior(fragile, o) \implies holds(broken, o)$$

Pattern Instatiation - concrete extension of the process ontology

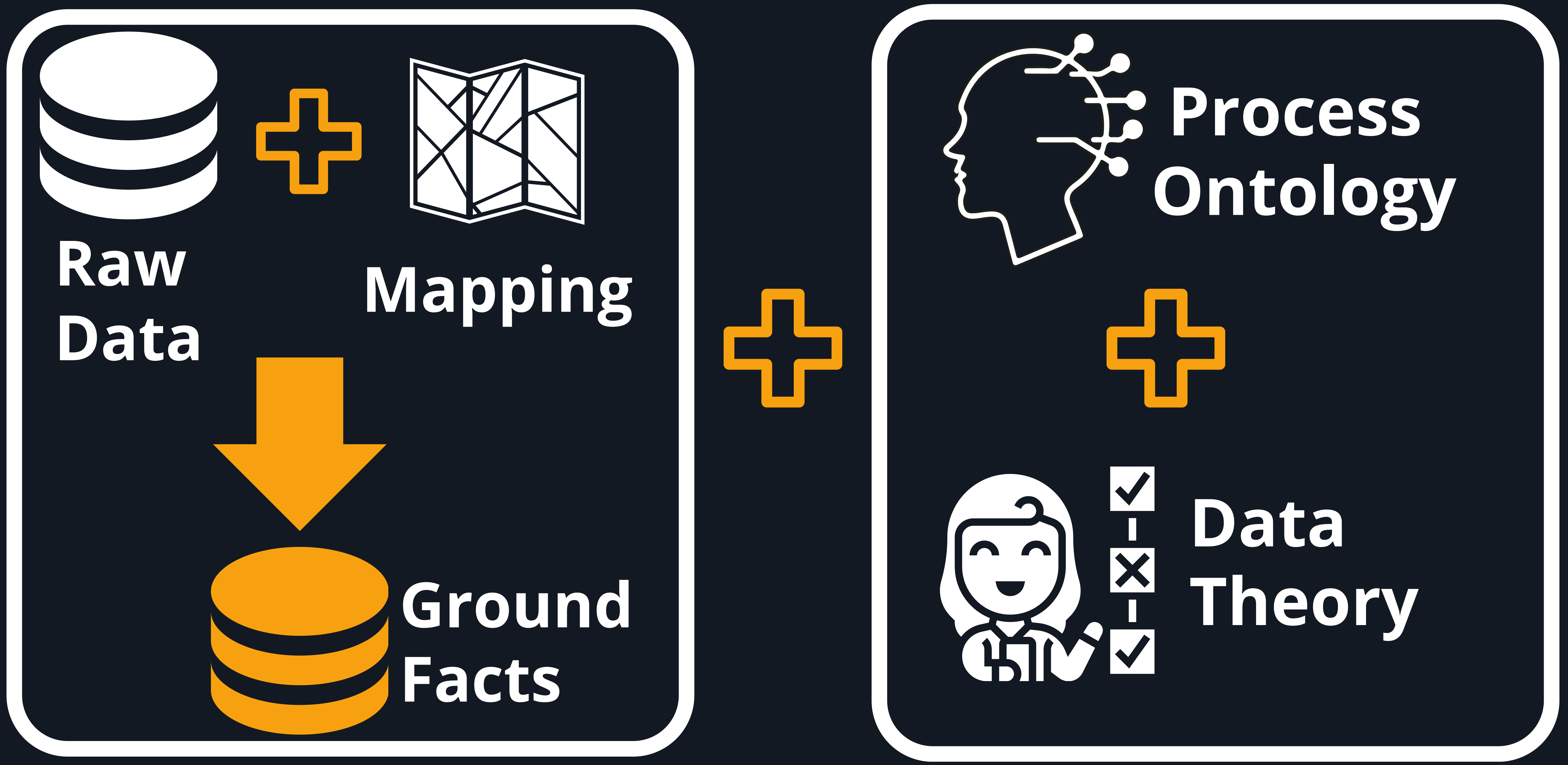
Data Theory

Process Ontology

A-Box

T-Box

Data-Driven Process Ontology



Part 2:
What’s in a (data) type?

What’s in a (data) type?

- Data is representative of real-world phenomena

What’s in a (data) type?

- Data is representative of real-world phenomena

- Data is represented using simple datatypes

What's in a (data) type?

- Data is representative of real-world phenomena
- Data is represented using simple datatypes
- Datatypes do not typify the real world

Datatype Problems

Datatype Problems

The Real World
- Implicit rules
- Complex concepts and relationships

Simple Datatypes
- Numbers are numbers*
- String OR Integer OR Float OR ...
- Same datatype represents many different concepts

Datatype Problems

Datatype Problems

Geospatial Region

Datatype Problems

Population

Datatype Problems

Distance

Datatype Problems: Mereology

Date	# Fully Vaccinated	% of Eligible Pop. Fully Vaccinated
6/22/2021	1 200 000	3.50%
...	...	...
7/29/2021	1 335 000	2.60%
...	...	...
5/28/2022	1 086 100	2.20%

Datatype Problems: Mereology

Date	# Fully Vaccinated	% of Eligible Pop. Fully Vaccinated
6/22/2021	1 200 000	3.50%
...	...	...
7/29/2021	1 335 000	2.60%
...	...	...
5/28/2022	1 086 100	2.20%

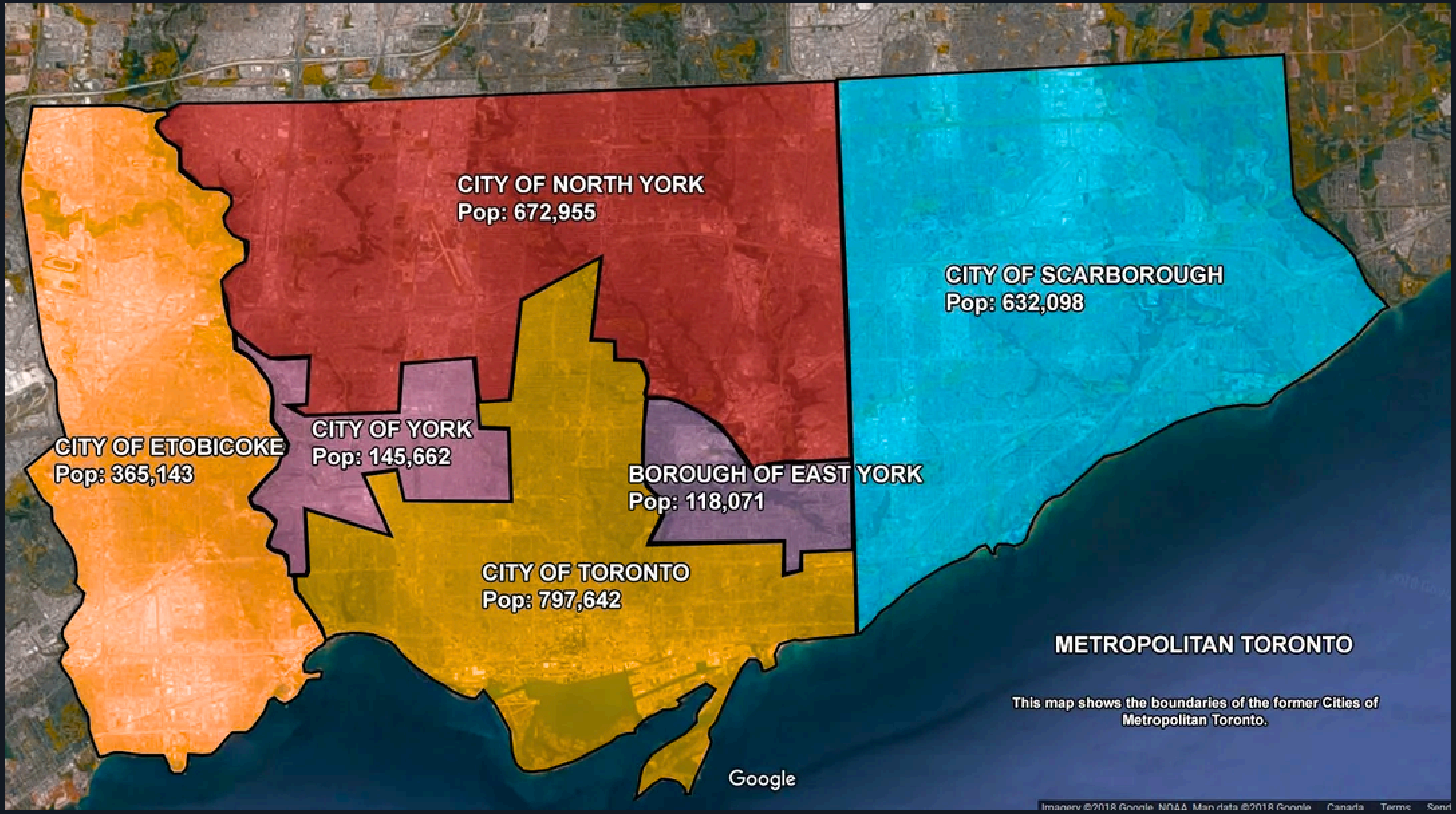
Children become eligible

Datatype Problems: Mereology

Date	# Fully Vaccinated	% of Eligible Pop. Fully Vaccinated
6/22/2021	1 200 000	3.50%
…	…	…
7/29/2021	1 335 000	2.60%
…	…	…
5/28/2022	1 086 100	2.20%

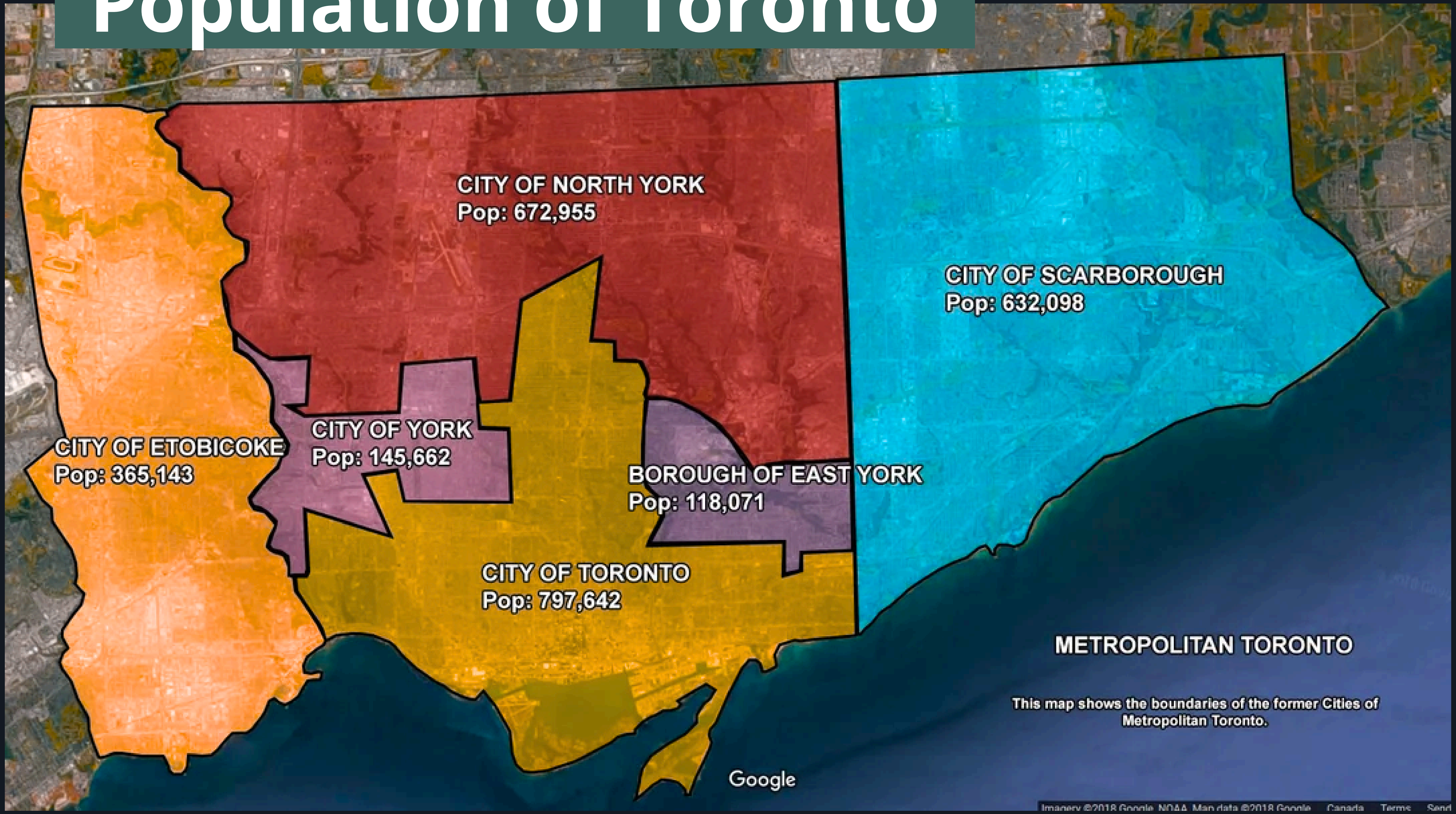
“Fully vaccinated” now defined as 3+ doses

Datatype Problems: Mereology



Datatype Problems: Mereology

"Population of Toronto"



Datatype Problems: Provenance

- How does analysis itself change the meaning of data?

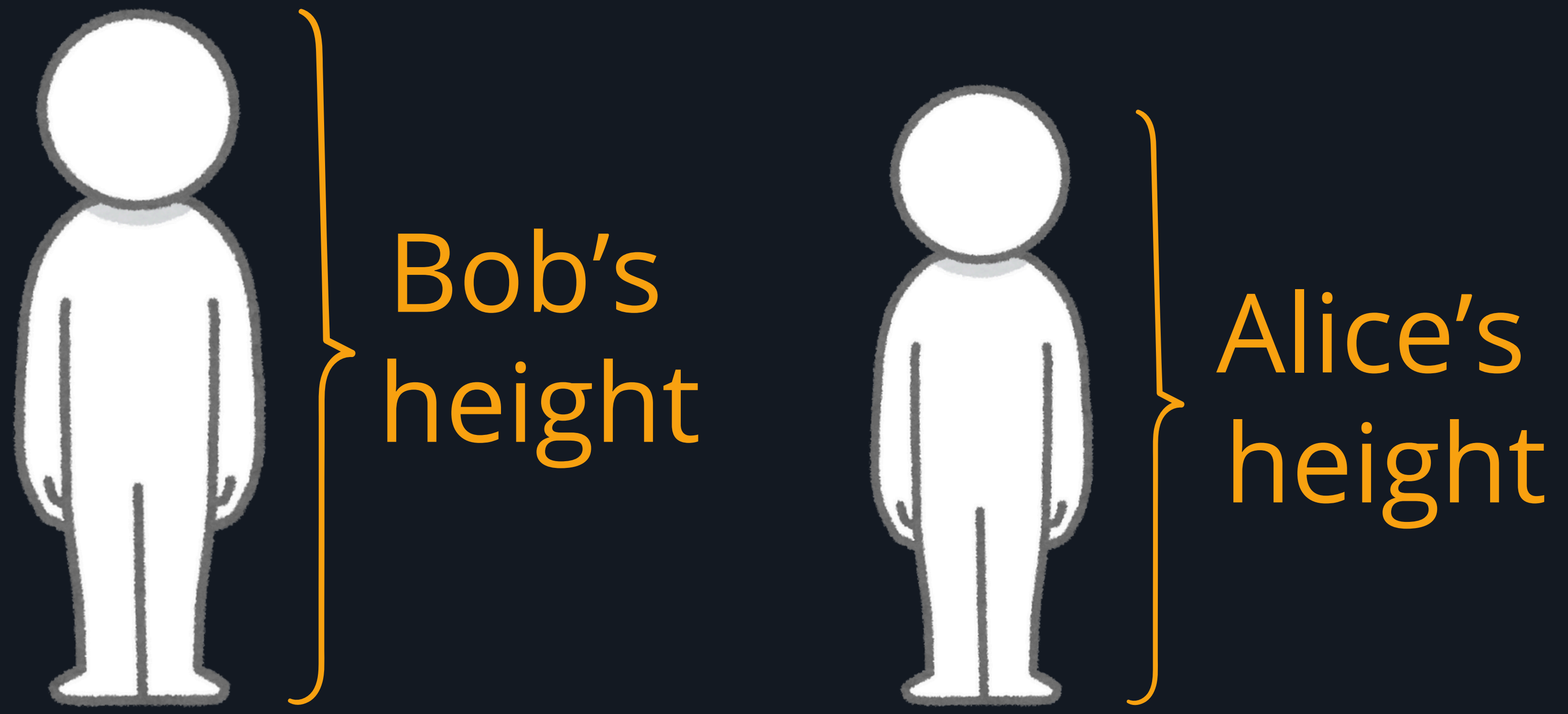
Datatype Problems: Provenance

- How does analysis itself change the meaning of data?
- Ex: Distance measurements

Datatype Problems: Provenance

- How does analysis itself change the meaning of data?

- Ex: Distance measurements

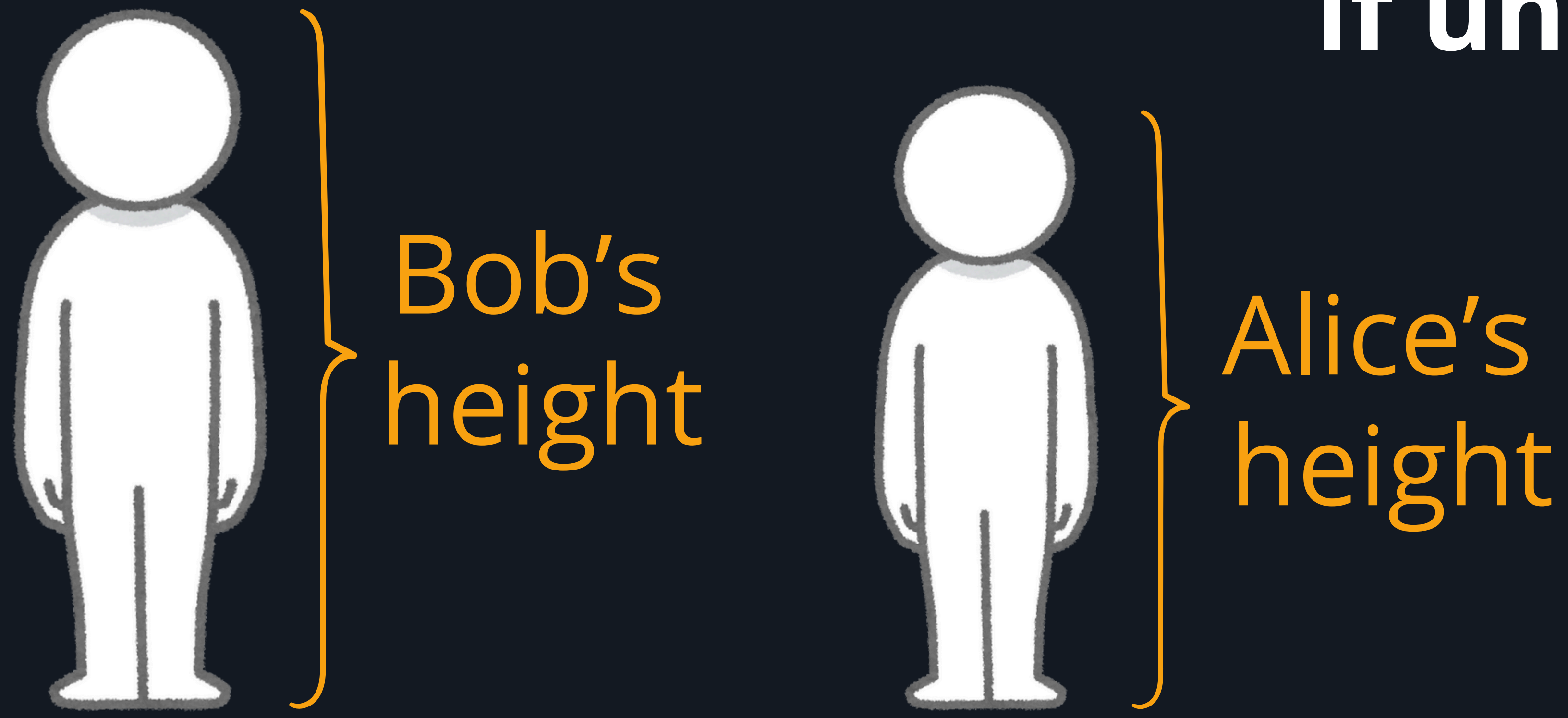


Datatype Problems: Provenance

- How does analysis itself change the meaning of data?

- Ex: Distance measurements

“If units match, all is well!”



Datatype Problems: Provenance

- How does analysis itself change the meaning of data?
- Ex: Distance measurements

Data Operation Semantics

- What can we meaningfully measure the “totals” of?

Data Operation Semantics

- What can we meaningfully measure the “totals” of?

- Population of a total region
- Height of a total person
- Temperature of a total region

Data Operation Semantics

- What can we meaningfully measure the “totals” of?

- Population of a total region
- Height of a total person
- Temperature of a total region

Property of (mereological) sum = (arithmetic) sum of properties

$$p(A + B) = p(A) + p(B)$$

Data Operation Semantics

- What can we meaningfully measure the "totals" of?

- Population of a total region
- Height of a total person
- Temperature of a total region

Property of (mereological) sum = (arithmetic) sum of properties

$$p(A + B) = p(A) + p(B)$$

Datatype Problems: Provenance

- How to formally represent real-world rules about quantitative data?

- How to integrate data provenance?

Datatype Problems: Provenance

- How to formally represent real-world rules about quantitative data?

- How to integrate data provenance?

Make types stronger!

Meaningful type safety

Meaningful Type Safety

- Leverage the power of dependent types
  - Types depend on values, Natural Numbers (Int >0)

Meaningful Type Safety

- Leverage the power of dependent types
  - Types depend on values, Natural Numbers (Int >0)

- Enforce preconditons for operations

```
RegionSum : (Disjoint [Region]) → Region
```

- Integrated provenance → types change after operations

```
PopAvg(regions) = Avg over regions
```

Meaningful Type Safety

- Encoding complex real-world constraints through type-checking

```
PopAvg( (Toronto2024 - Toronto2016),
     (Buffalo2024 - Albany2024))
```

```
PopAvg : ([Population]
(differsOverTime XOR differsOverRegion))
→ Population
```

Formal Ontologies and Type Safety

- Types are propositions, proofs are programs

Formal Ontologies and Type Safety

- Types are propositions, proofs are programs
- Ontologies can be constructed which represent* type safety

Formal Ontologies and Type Safety

- Types are propositions, proofs are programs
- Ontologies can be constructed which represent* type safety

* Proving this representation is tricky

Takeaways

- We must build the ontological commitments underlying our analysis directly into data pipelines for verifiability, reusability, and transparency.
- Process data theories are a pragmatic & modular way to model event data interpretation, separating data-dependent definitions from foundational logic.
- Meaningful type safety offers a powerful a-priori approach, enforcing foundational rules as type-checks to prevent invalid operations.

Contact / More Details

Looking Ahead

- Process ontology benchmarking - Providing new data-driven challenges for existing foundational ontologies to demonstrate their capabilities for operational realization

- Extensions for object-centric event logs and event knowledge graphs

- Using LLMs as the interface to instantiate knowledge patterns and create data theories

Contact / More Details