Link Indexes™ and Ontologies

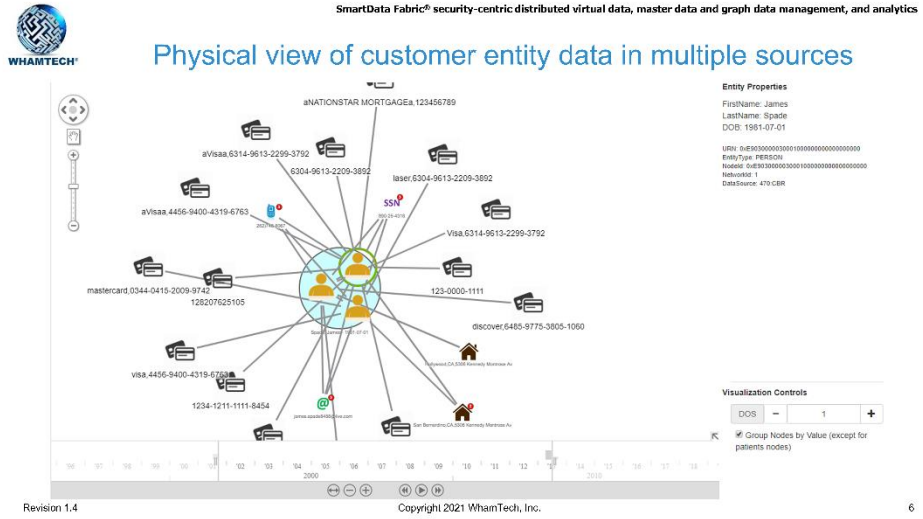
December 2023

Link Indexes™

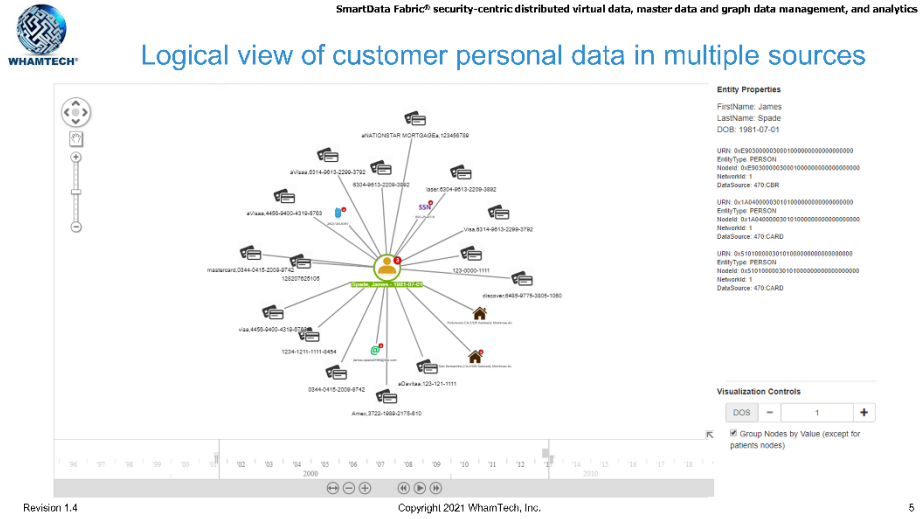
- Developed from hyperlink mapping for a Web search engine, enabling community network representations and network-oriented navigation
  - Combined content with structure (and ranking)
- See separate presentation “WhamTech Link Indexes Overcome Conventional Link Analysis Problems and Provide Other Unique Solutions”:
  - Link Indexes are effectively join indexes based on matching high cardinality values – entities, keys, URLs, categories, algorithms, views, etc.
  - Normal content indexes are used to find matches to build and maintain Link Indexes
  - Normal content indexes are combined with Link Indexes on query result-sets (previously used virtual and physical Boolean bitmaps) to provide graph-based solutions
  - Graph-based solutions include social network analysis (SNA)/link analysis, social media analysis, fraud detection, suspicious activity reports (SARs), money laundering, cause and effect, and effects-based operations (EBO)
  - Other solutions include CRM, single customer, employee and patient and other entity (product, organization, etc.) views

Link Indexes™ and Ontologies

- Link Indexes represent unique links either within records (self-joins) or between records (internal or external joins) containing matching values (entities, keys, etc.)
- The matching value entity, key, etc. type is retained as part of the link – the actual matching value is not retained, but could be if needed
- Additional links are implied, but not captured, between entities in the same record
- When an RDF-oriented query is performed, normal content indexes are used in conjunction with Link Indexes
  - Both resolve to universal record numbers (URNs) that allow for Boolean operations on interim result-sets
- MDM is used to group same entity records together as a single logical node, but the physical links are retained to show details

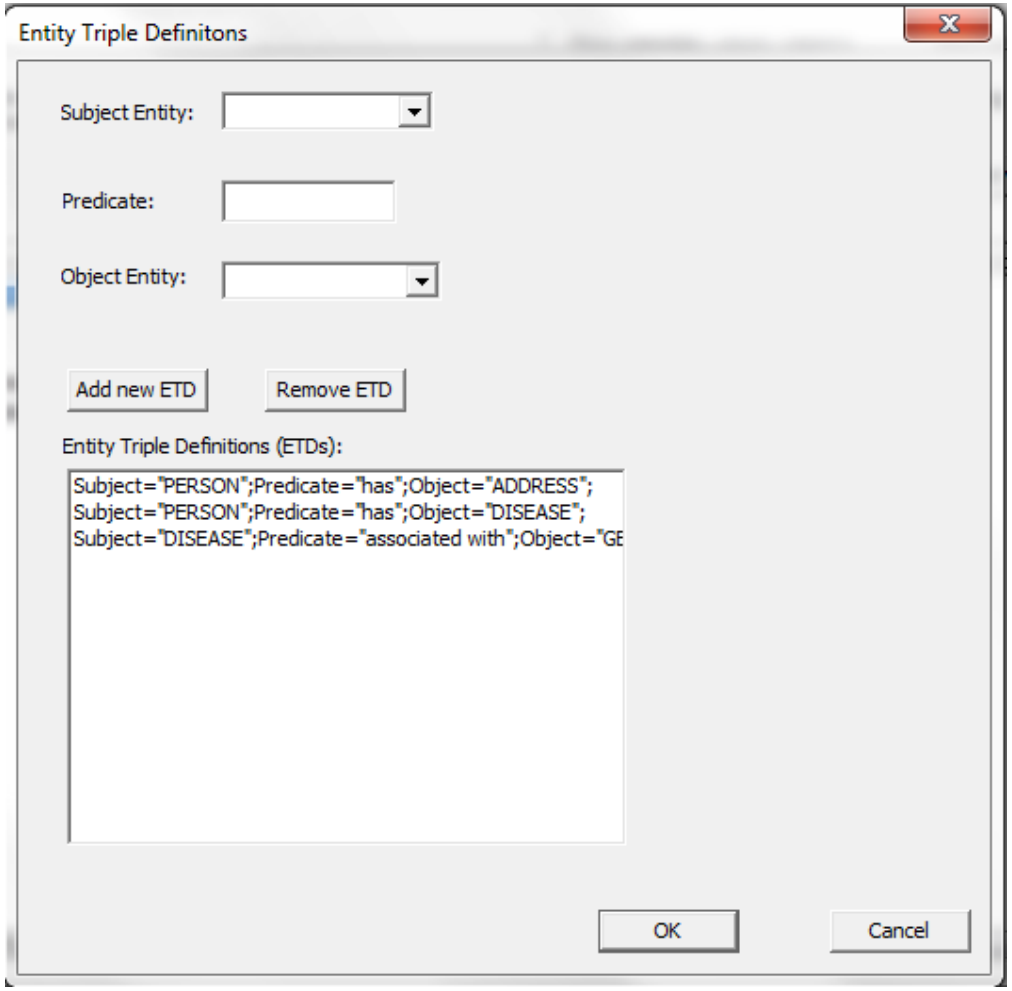


with MDM =>



- WhamTech uses SQL to execute graph query language (QL)-type queries on External Index and Query (EIQ) Adapter indexes
  - PostgreSQL-based EIQ adapters can accommodate graph QLs, but would have to test

Virtual Triple Store



Entities are defined in the Standard Data View to which data source indexes are mapped

Entity triple definitions adds predicates to:

- Execute SQL (or a graph query language conversion to SQL), directly on indexes, using a combination of content, link and MDM indexes
  - Allows queries to include other terms and constraints/filters
- Export triples to a triple store, maybe with other attributes, for graph database work

Both the above were planned projects and could be slated to be completed

Future options may be to import RDFs into the SmartData Fabric tool instead of manually defining them, e.g., HL7 FHIR RDF (https://www.hl7.org/fhir/rdf.html)

Link Query Execution (1 of 2)

- When an RDF-oriented query through SQL (or graph QL with conversion to SQL) is submitted, e.g., PERSON, “lives at”, ADDRESS where PERSON = some value(s)
  - SQL isolation of PERSON = some value(s) is performed on normal content indexes resulting in a list of matching PERSON* URNs
  - List of matching PERSON URNs are submitted to Link Indexes™ to find previously linked records resulting in a list of linked URNs
  - The list of linked URNs are checked against the metadata ontology model for data sources and tables that contain a “lives at” ADDRESS (not “owns” or any other predicate)
  - The match ADDRESS** and “lives at” URNs are then used to read the associated records and returned as a final result set

*The PERSON match makes use of a composite weighted multi-attribute probabilistic match, based on a number of attributes, e.g., full name, fuzzy match on names, DOB, SSN, ADDRESS, EMAIL, PHONE_C, PHONE_H, etc.

**The ADDRESS match would be exact after address clean-up for indexes

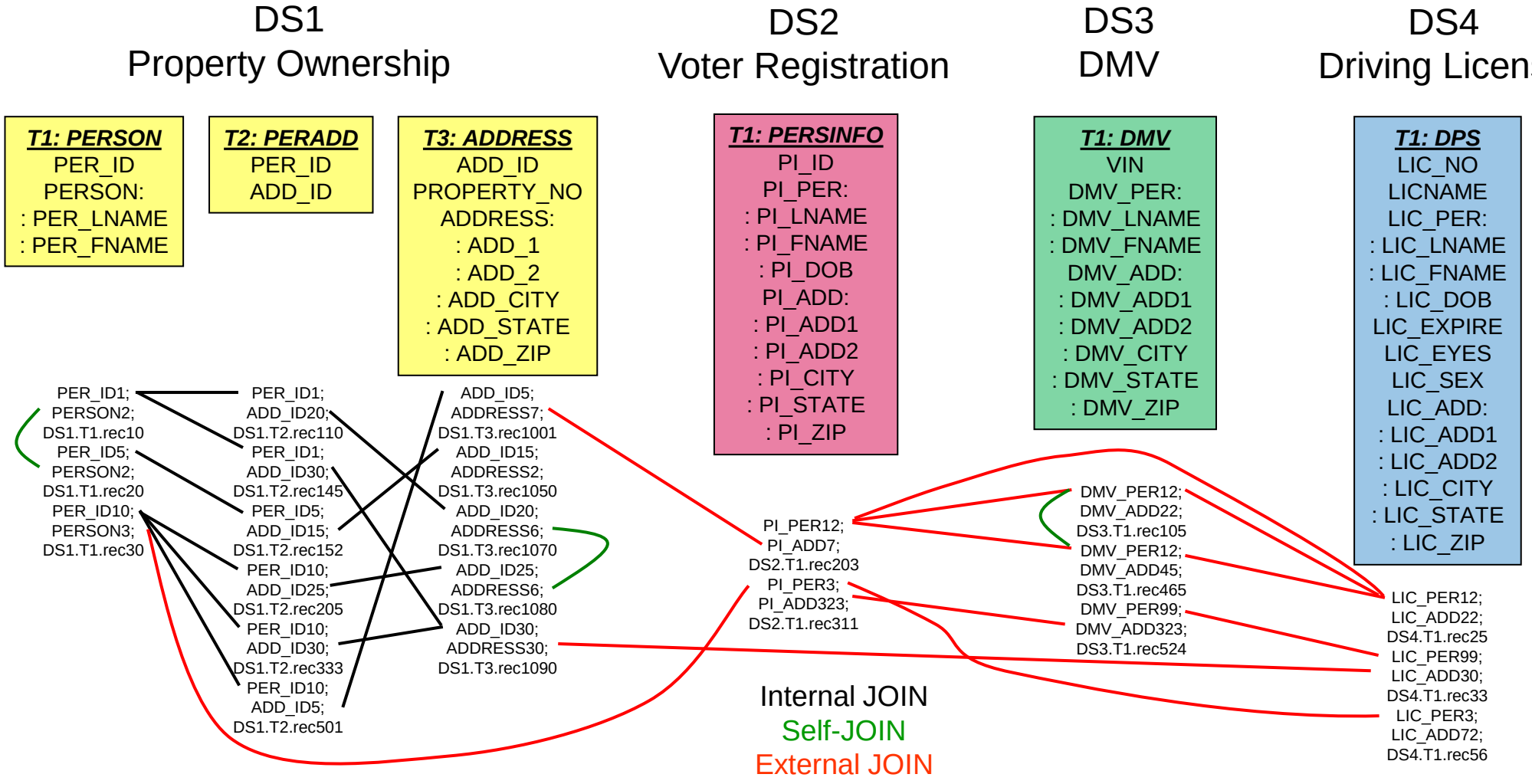
Link Query Execution (2 of 2)

- Multiple RDF triples could be resolved in one query
  - Example: PERSON, “lives at”, ADDRESS; PERSON, “works for”, EMPLOYER and PERSON, “owns”, VEHICLE
- RDF triples could be combined in degrees of separation-type queries or for link/network analysis
  - Example: PERSON, “lives at”, ADDRESS; other PERSONS, “live at”, same ADDRESS and another PERSON, “owns property at”, same ADDRESS

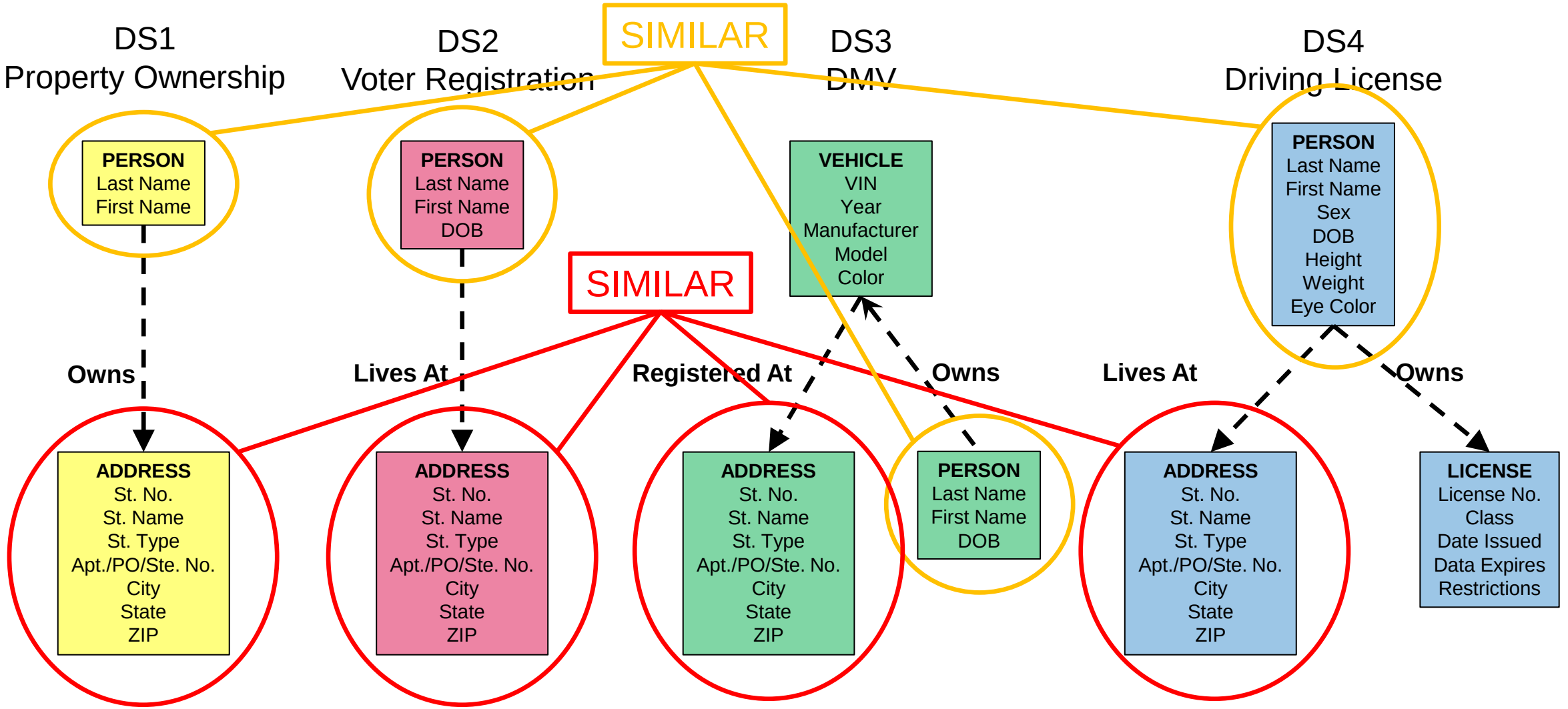
Simple Example from Link Indexes Presentation*

*“WhamTech Link Indexes Overcome Conventional Link Analysis Problems and Provide Other Unique Solutions”

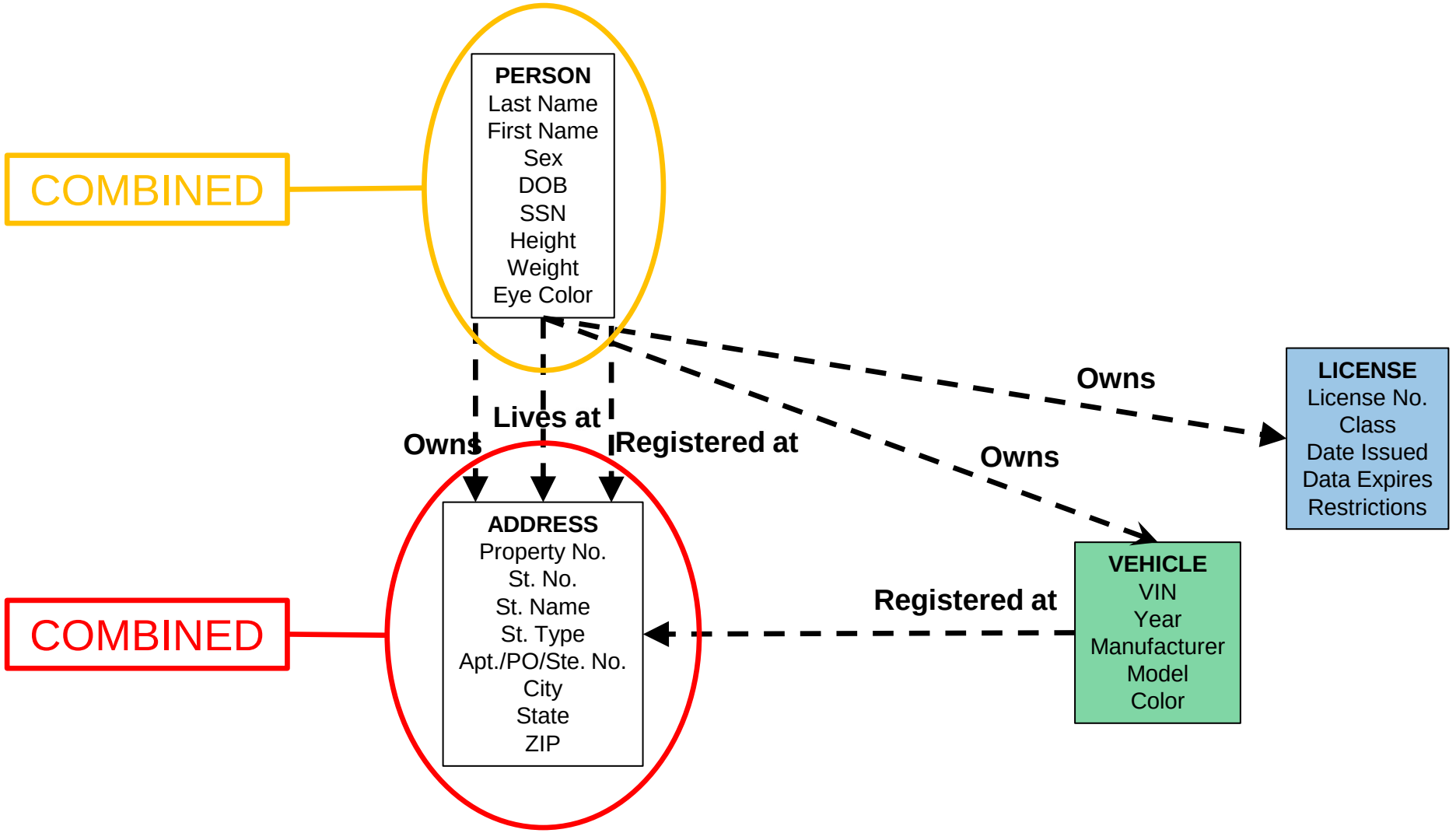
Link mapping within and across multiple data sources



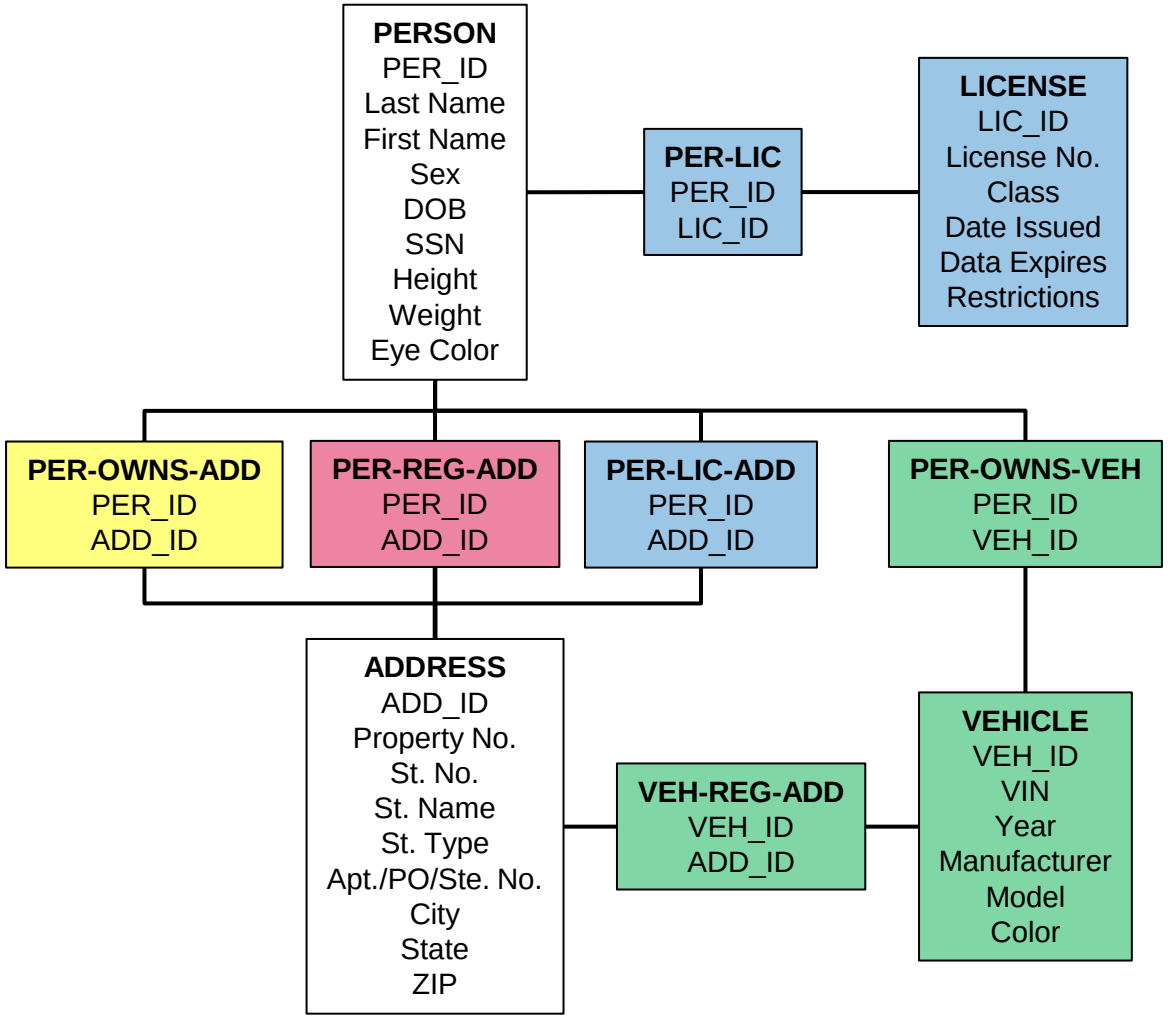
Individual data source ontologies representation



Combined enterprise ontological model



Combined enterprise relational model



Levels of data representations and processes (1 of 2)

DATA SOURCE SCHEMA

T1: PERSON
PER_ID
PERSON:
: PER_LNAME
: PER_FNAME

T2: PERADD
PER_ID
ADD_ID

T3: ADDRESS
ADD_ID
PROPERTY_NO
ADDRESS:
: ADD_1
: ADD_2
: ADD_CITY
: ADD_STATE
: ADD_ZIP

L1

Optional copy or conversion to another format, e.g., document

OPTIONAL DATA LAKE SCHEMA

T1: PERSON
PER_ID
PERSON:
: PER_LNAME
: PER_FNAME

T2: PERADD
PER_ID
ADD_ID

T3: ADDRESS
ADD_ID
PROPERTY_NO
ADDRESS:
: ADD_1
: ADD_2
: ADD_CITY
: ADD_STATE
: ADD_ZIP

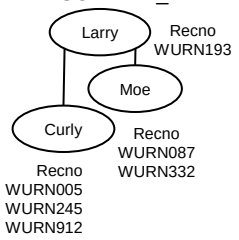
PROCESS

INDEXES

Typically, do not store data, but, optionally, can
Index schemas are usually the same as or very similar to data source schemas

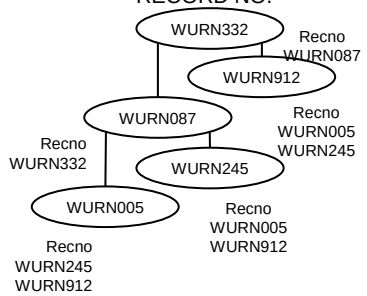
CONTENT INDEX

T1:PERSON.PER_FNAME



LINK INDEX

RECORD NO.



INDEXED VIEW

e.g., materialized aggregation
Can be virtual and hierarchical

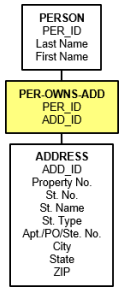
Recno	Last Name	First Name	No. Properties Owned
1	Smith	Curly	7
2	Jones	Larry	1
3	Parker	Moe	3
…	…	…	…

L2 L3 L4 L5

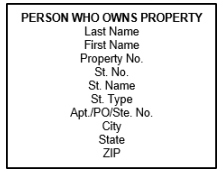
QUERY

ALL VIRTUAL LOGICAL DATA VIEWS

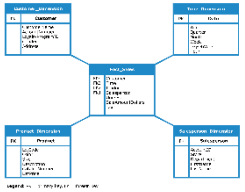
STANDARD DATA VIEW



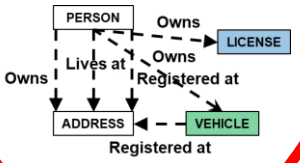
ADDITIONAL BUSINESS OBJECT(S)



DATA MART(S)



GRAPH DB

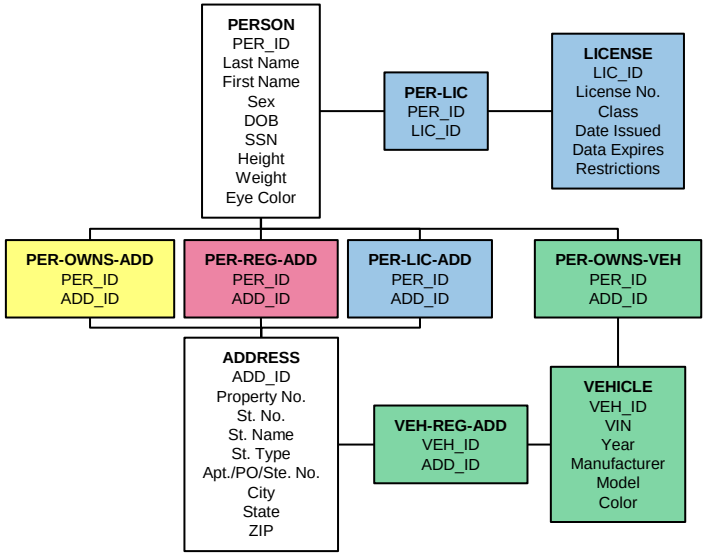


L6

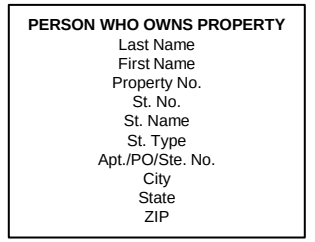
IMPORT

STANDARD DATA MODEL

Could be industry standard, e.g., ACORD, HL7 or NIEM



ADDITIONAL BUSINESS OBJECT(S)



What applications/ end-users see

L7

WHAMTECH®

Levels of data representations and processes (2 of 2)

Level	Description	Schemas	Comments
1	Data sources	Native	Hierarchical/mainframe, relational, flat, document/unstructured, HTML, API, streaming, etc.
2	Optional Data Lake	Native or modified	Same as data sources or converted into a document or other format
3a	Major schema processing	Hierarchical to relational	Typical of mainframe data files involving deduplication/normalization and can apply to XML and other formats
3b	Minor schema processing	E.g., full name to last, first, middle and nick names or data type/attribute changes	Adds, removes or modifies columns/fields in tables in schemas
3c	Content processing	Entity extraction and other analytics	Adds tables to schemas
4a	Indexes	Mainly, same as data source native schemas, modified content and master data	Data subject to cleansing, transformation, standardization and data security Usually, master data content in added table
4b	Link Indexes™	No impact	Captures links/relationships among data, including PK-FK, self-joins and external joins, usually on defined (high cardinality) entities after master data management
4b	Indexed views	Pre-aggregations, calculations, joins and hierarchical – virtual and materialized	Adds tables to schemas
5	Query Indexes	Mainly, same as data source native schemas, but could involve multiple tables and data sources	Query business rules stored in adpaters
6a	Standard Data View (SDV)	A virtual Logical Data View (LDV) that is usually a subset of the Standard Data Model (SDM)	Imported from
6b	Additional Business Object(s)	Usually, a virtual single table that has embedded business rules, e.g., “people that own property”.	As there can be multiple ways to execute a query using the SDV, and/or there are rules/query constraints, Business Objects specify how specific queries are executed
6c	Data Mart(s)	Usually, a central fact table with multiple dimensions	Hierarchical virtual indexed join views that are maintained in near real-time and populated on-query – pre-aggregations, calculations and joins can be materialized, i.e., physical
6d	Virtual Graph Database	A virtual ontological/semantic representation of entities and links/relationships among them	Simple SQL with a ontological/semantic model allows SmartData Fabric to be viewed as a virtual triple store or also can be used to perform advanced link analysis
7	Standard Data Model (SDM) and additional business objects	Can be simple or complex	Usually, industry-provided, e.g., HL7 in healthcare, ACORD in insurance and NIEM in government

Conclusions

Combination of WhamTech content indexes, Link Indexes™ and master data indexes could be used to automatically:

1. DISCOVER relationships between entities from data sources up using WhamTech Link Indexes and thereby define at least the subjects and objects of RDFs - predicates may or may not be present in the data source, however, it would be a start in capturing ontologies
2. MERGE separate ontologies using the discovered relationships
3. COMBINE variations of ontologies to arrive at a single representation
4. VALIDATE discovered ontologies by superimposing and obtaining a best fit of actual entity link maps on established ontologies
5. PRESENT highly normalized ontological views of data to applications using standard drivers, Web services and SQL, or SPARQL, regardless of where or how the data resides

NOTE: This would be in addition to capabilities already associated with Link Indexes, e.g., accelerate internal and external joins, degrees of separation queries, link analysis, probabilistic linking, MDM, CCPA, GDPR, etc.

The End