Link Indexes™ Overcome Conventional Graph Database Problems and Provide Other Unique Solutions

December 2023

WhamTech takes a bottom-up approach

- Conventional graph database applications take a top-down approach
  - Start with presentation/visualization
  - Data accessed in a centralized database, usually a triple-store, and/or federated data access with ETL to a centralized store
- WhamTech tackles data challenges, including graph database, from the bottom-up
  - Start with data fundamentals of discovery, cleansing, transformation, standardization and security, and MDM, using index-based federated adapters and federation servers
  - Link entity data in the index-based data access layer, aka SmartData Fabric®, and capture with Link Indexes™
  - Enable almost any application, including interactive graph visualization, to work with data access and Link Indexes™ through standard drivers and SQL

Problems with conventional graph database solutions

1. Data access
   - Federated access with conventional adapters used for incremental ETL- low quality results, low performance and high data source query load
   - Data warehouse - copy data using ETL - responsibility, accountability, security, privacy and legal issues
   - Either way, limited data available for query processing, constraints/filters and values

   => All data that can be represented has to be extracted and processed for links in a centralized database
2. Low data quality – impacts query success and subsequent links - GIGO
3. No integration with MDM
4. Difficult to scale or incrementally update - large amounts of data in a centralized database
5. Near real-time access to data or updates usually not possible
6. As data is extracted from sources, many inherent links and associations among data are lost
   - Example: All data in a record in one table indirectly associated with all data in a record in another table through one PK-FK relationship, i.e., one link in a Link Index™
   - Sequence of records and/or date-time stamps in data sources
7. Absence of probabilities of links and confidence levels of data
8. Absence of threat/favorability, sentiment or other scores

Link Indexes™ defined

- Link Indexes™ are representations of links between data pointers to records/files/documents within the same data source or across multiple data sources
  - Typically, based on data/entity matches, e.g., PK-FK, exact match and fuzzy match – these are direct or obvious relationships, but there are other options – see following slide
  - As a consequence, all data in the same record, file or document/paragraph is connected to all data in the matching record, file or document/paragraph – these are indirect or non-obvious relationships
- Link Indexes™ are all that are needed for complex network representations and combined through simple SQL
  - More than one network representation can be layered
  - Content is layered on top of network representations along with filters, probabilities, threat/favorability scores, and any other constraints and information
- Link Indexes™ can be used in conjunction with content and master data indexes to accelerate internal and external joins, for degrees of separation (DOS) queries, and for link mapping and link analysis type solutions, again, using simple SQL
  - Single person (customer, employee, patient, etc.) and other entity views, social network analysis (SNA)/link analysis, social media analysis, fraud detection, suspicious activity reports (SARs), anti-money laundering, effects-based operations (EBO), etc.
- Link Indexes™, and content and master data indexes can support almost any solution – not just virtual graph database

Examples of link types

1. Hyperlink-to-hyperlink (exact match – URI, URL and URN harvesting)
2. Primary key–to-foreign key (exact match)
3. Entity–to-entity (exact and fuzzy (%) match through queries on structured data and extracted entities)
   - Self-joins (same table)
   - Internal joins (multiple tables in the same data source)
   - External joins (multiple tables in multiple data sources)
4. Topic–to-topic (% match from categorization tools)
5. Keyword-to-keyword (exact and fuzzy (%) match through queries on unstructured data) using select words from key material and highly weighted words from the combined support vector from IG models
   - Can include entity–to-keyword and keyword–to-entity
6. Algorithms from predictive analytics

Basic premise behind Link Indexes™

All structures…

hierarchical, relational, network and even 3D objects

… can be represented as combinations of pairs of data

… for example: Primary Key (PK) – Foreign Key (FK) relationships in relational databases

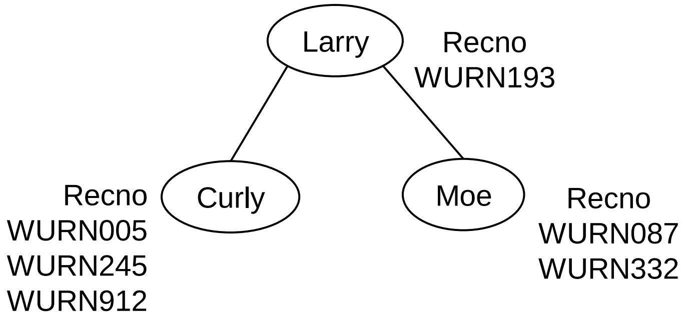
Link Indexes™ include basic join indexes, which can also be used to accelerate joins

Link Indexes™ are for high cardinality entities

- High cardinality entities reduce the number of links and therefore improve match probabilities, information value, indexing and query processing
- A NAME is likely low cardinality, whereas a PERSON is high cardinality as not just a name, but comprise more than one attribute, aka complex entity – combination is unique:
  - SSN
  - Birth country
  - DOB
  - Passport country
  - Passport no.
  - E-mail address
- Other entities are clearly unique or very high cardinality and are usually a single attribute, aka simple entity:
  - SSN
  - E-mail address
  - Phone no.
  - Address (although multiple attributes, can be checked as a simple entity)
  - Vehicle Identification No. (VIN)

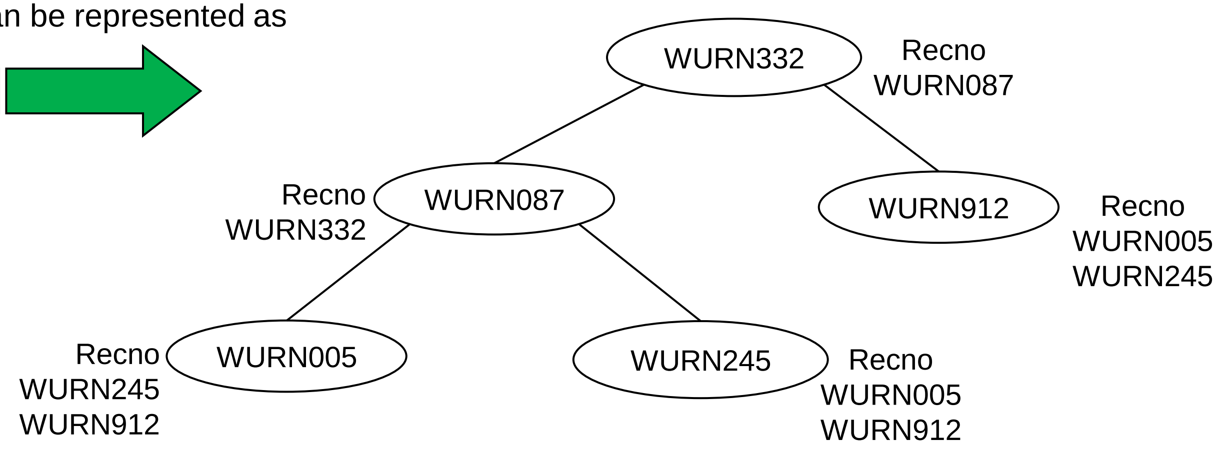
Simple example of a content index and a Link Index™

Example: Name



Links can be represented as

Recno



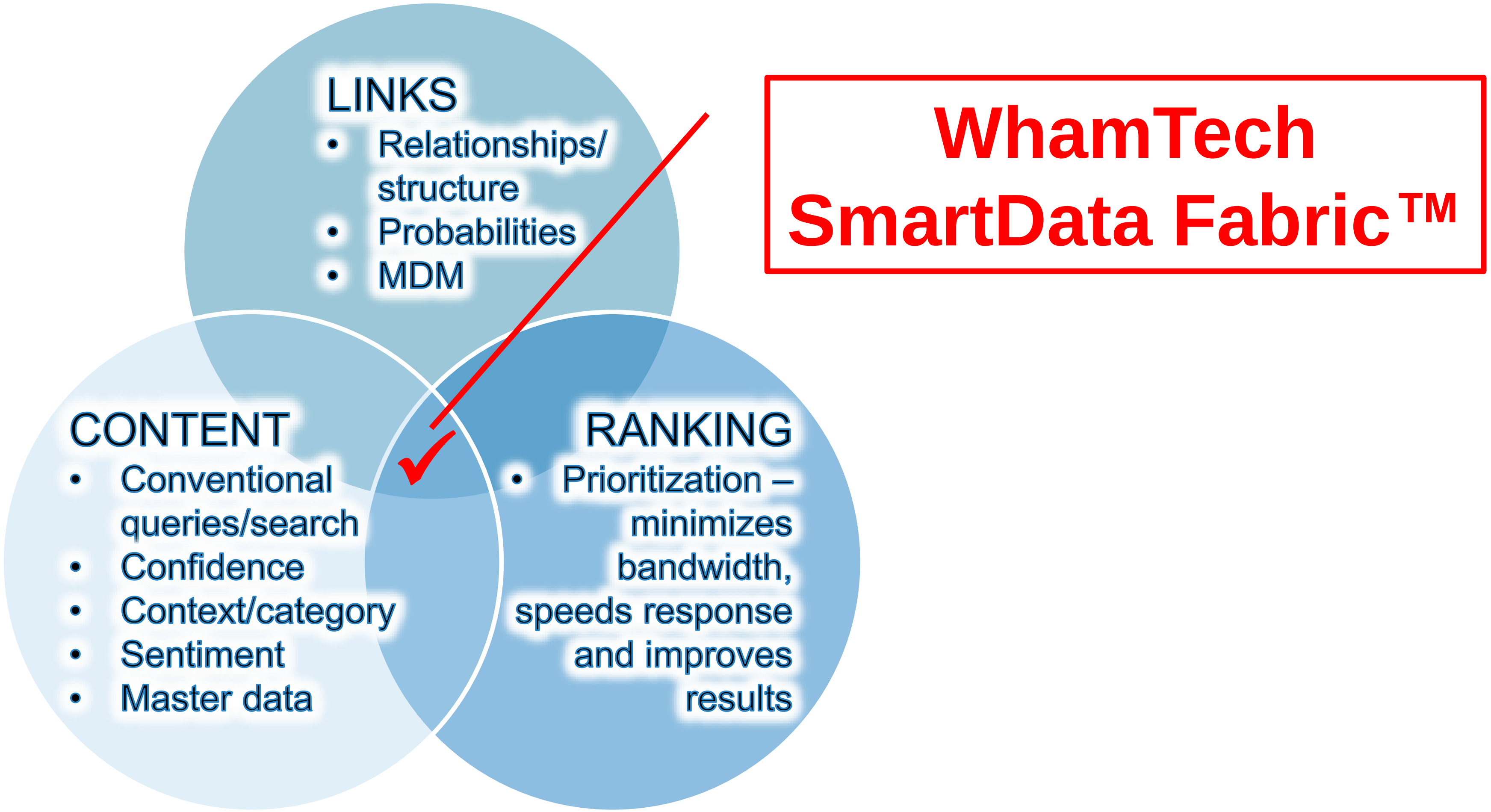
Normal Content Index

Link Index™

These indexes can be accessed and combined through SQL: CONTENT + LINKS and then RANKED

Note: WURN = WhamTech Universal Record Number

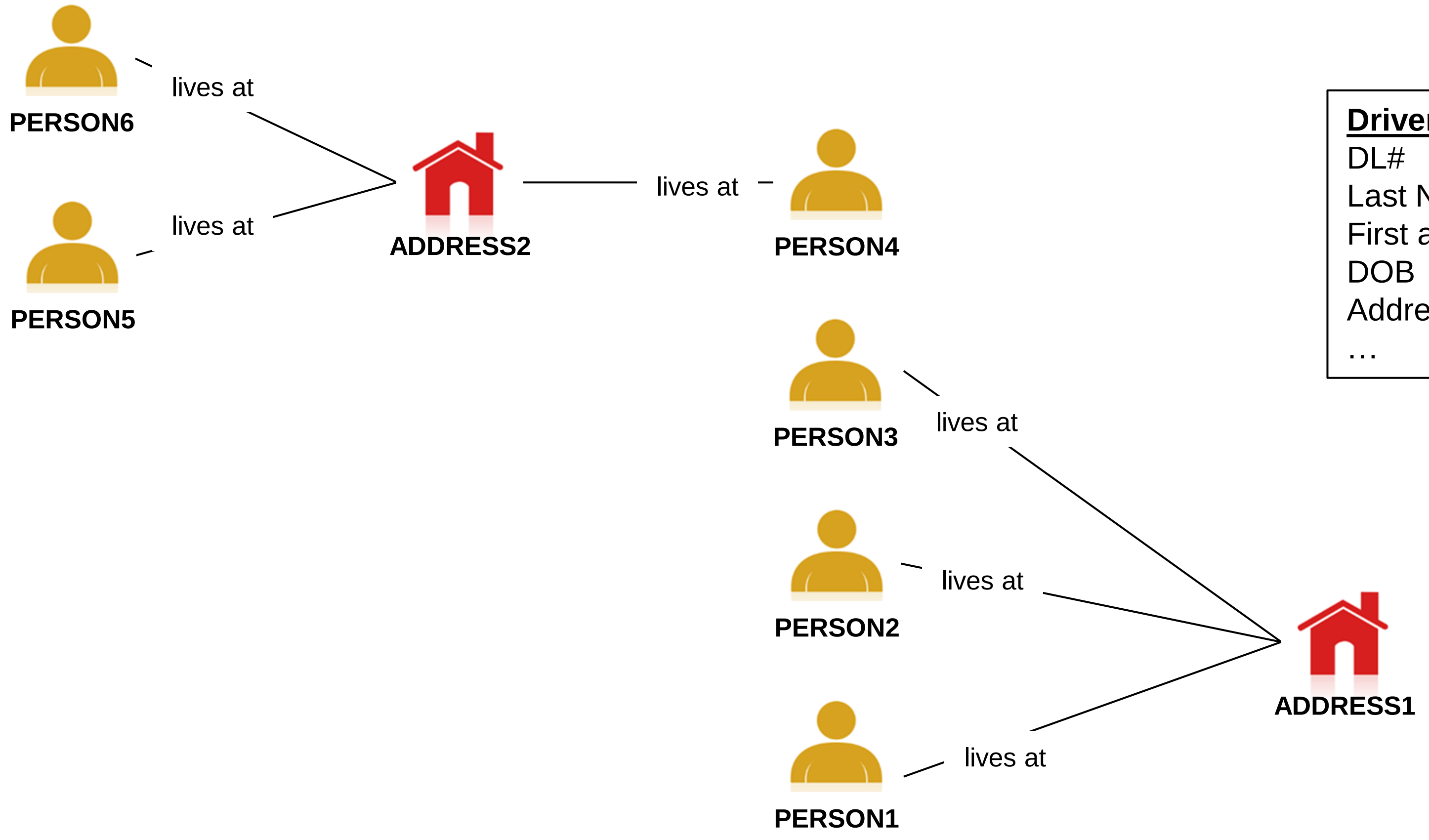
Content + Links + Ranking



Attributes common to content indexes and Link Indexes™

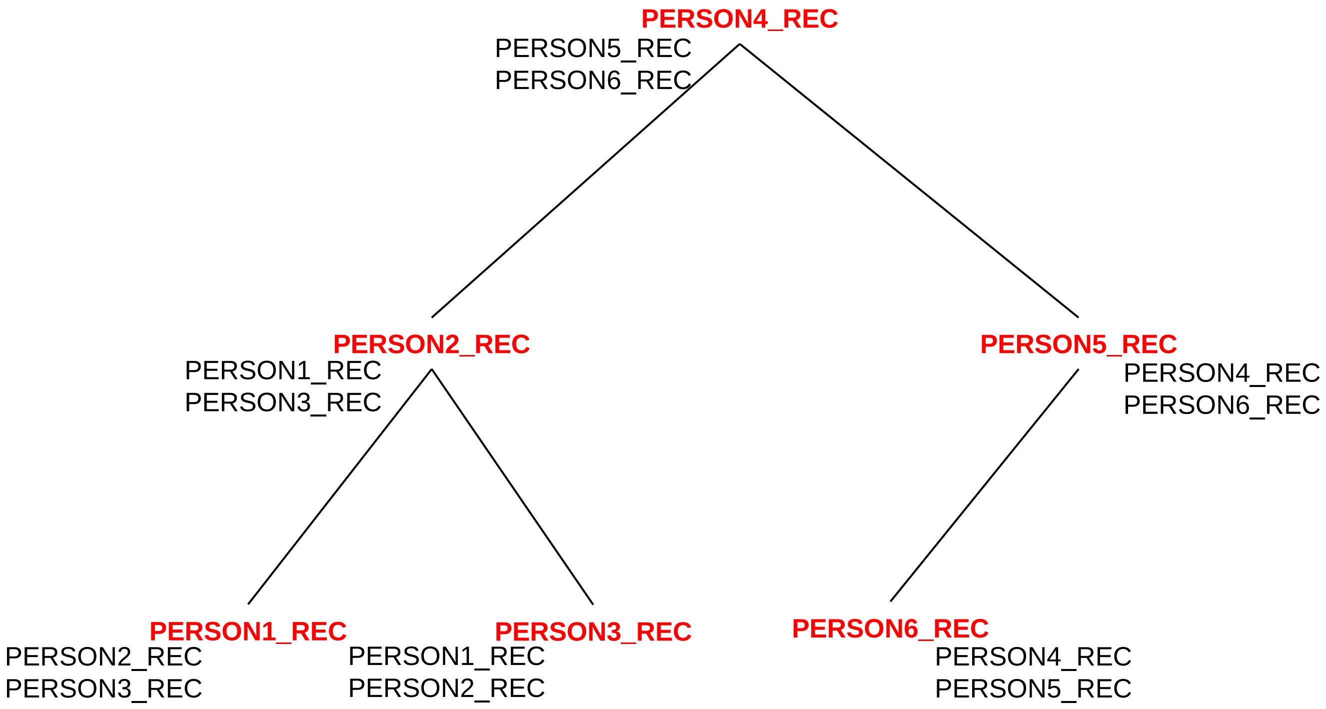
- Can be stored and maintained at the same level
  - Do not have to be
- Both resolve to the same data pointers
- Queries interact between them
  - Created an SQL extension called LINK and qualifier DOS (degrees of separation) – need to migrate to PostgreSQL-based EIQ Adapters
  - Used in conjunction for MDM
- Both accessible from standard applications, including graph visualization applications, such as Keylines®, reporting, BI and analytics

Example Link Indexes™: Data from DMV driver licenses (1 of 2)



Driver License Record
DL#
Last Name
First and Middle Names
DOB
Address
…

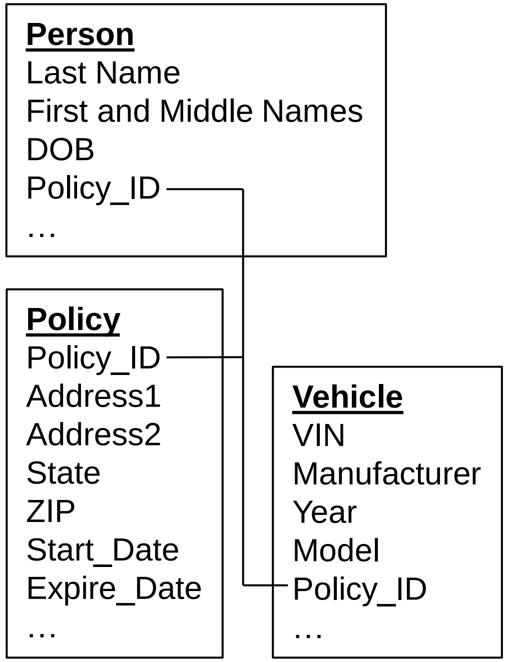
Example Link Indexes™: Data from DMV driver licenses (2 of 2)

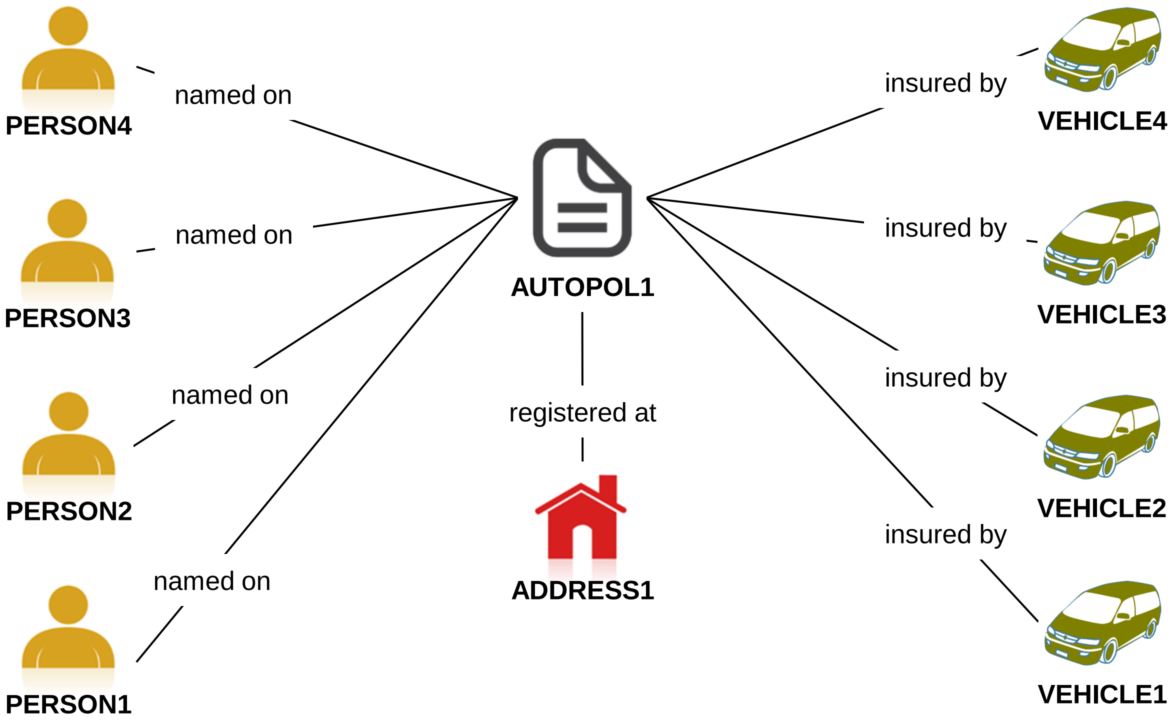


<u>Driver License Record</u>
DL#
Last Name
First and Middle Names
DOB
Address
…

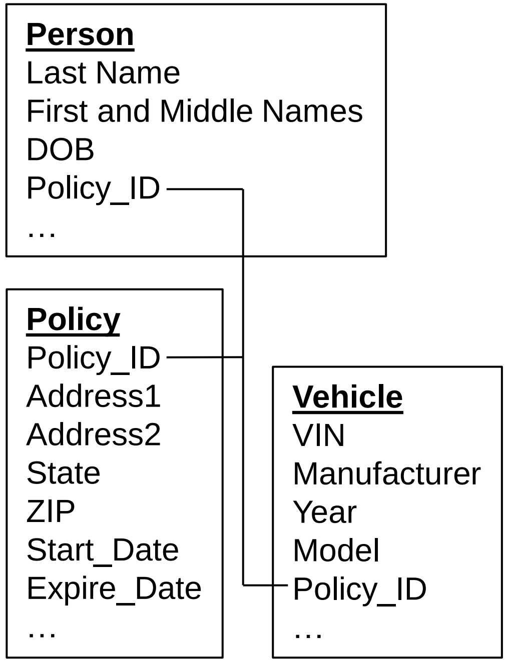
PERSON records linked to other PERSON records through common ADDRESSes in DMV Driver Licenses single table database, aka self-joins

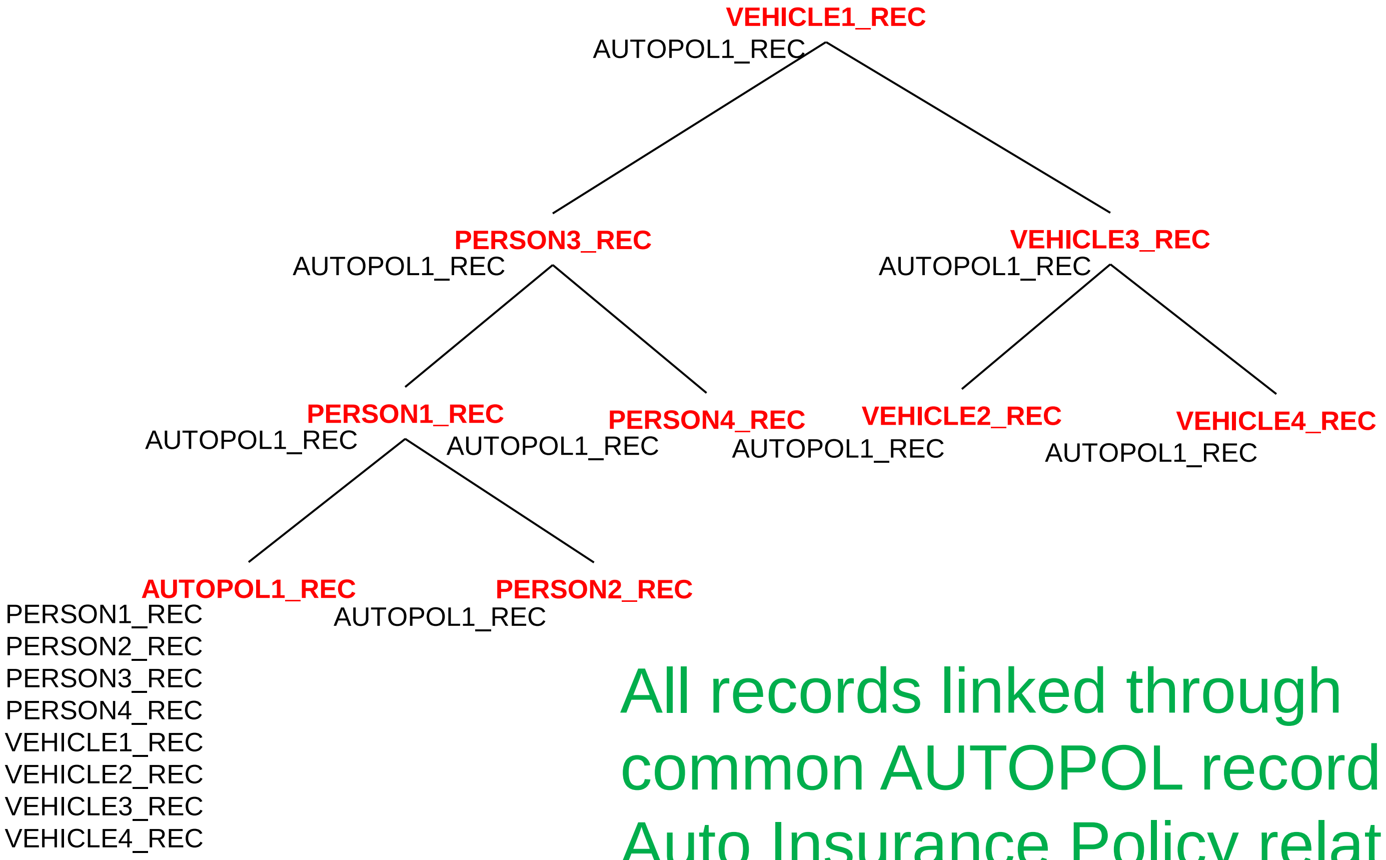
Example Link Indexes™: Data from auto insurance policy (1 of 2)





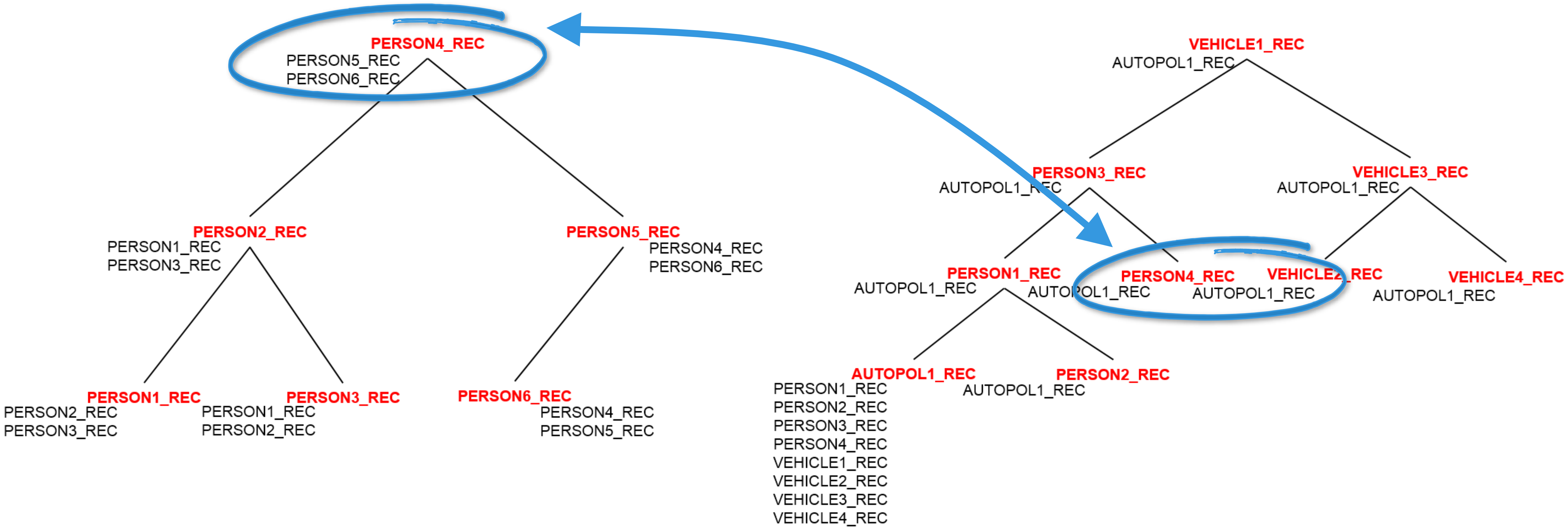
Example Link Indexes™: Data from auto insurance policy (2 of 2)





All records linked through common AUTOPOL record in Auto Insurance Policy relational database, aka PK-FK joins

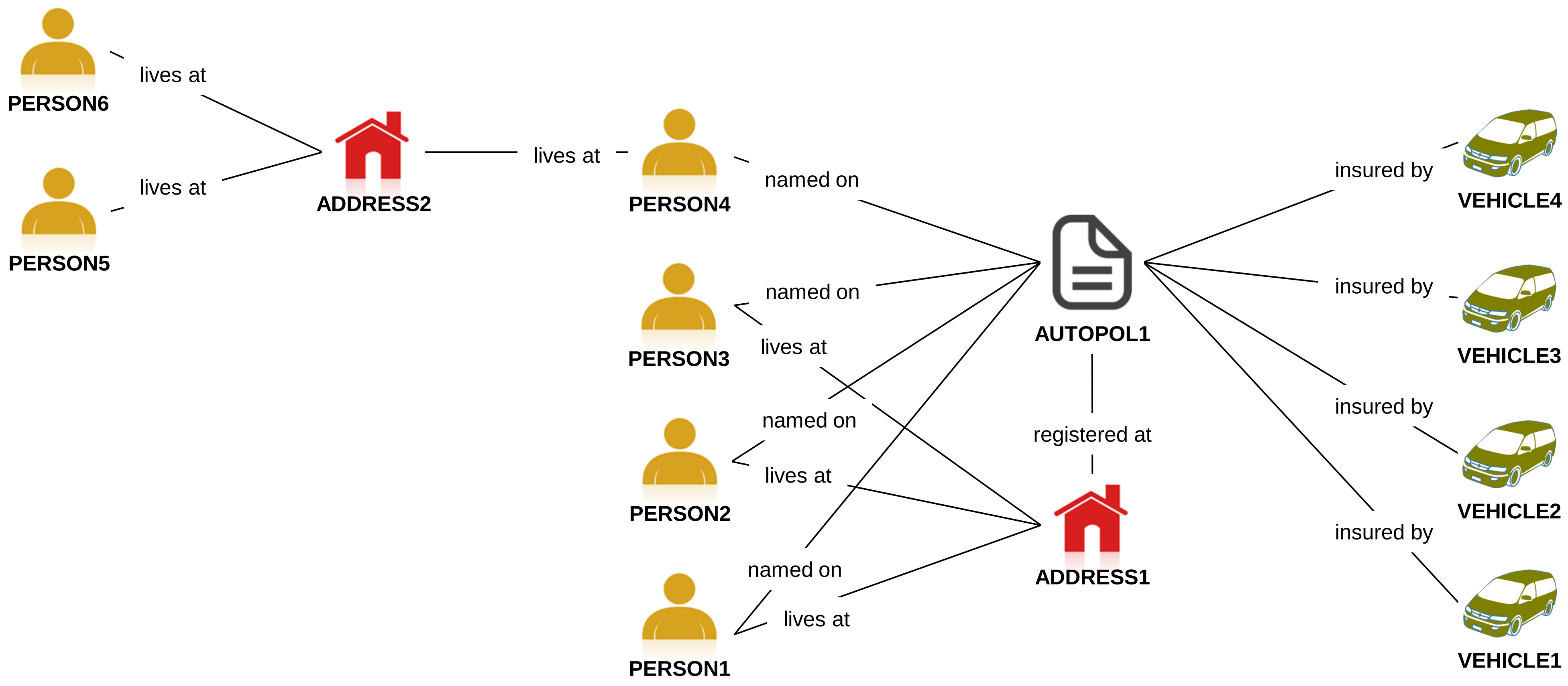
Example Link Indexes™: Combined view (1 of 2)



DMV Driver Licenses Link Index

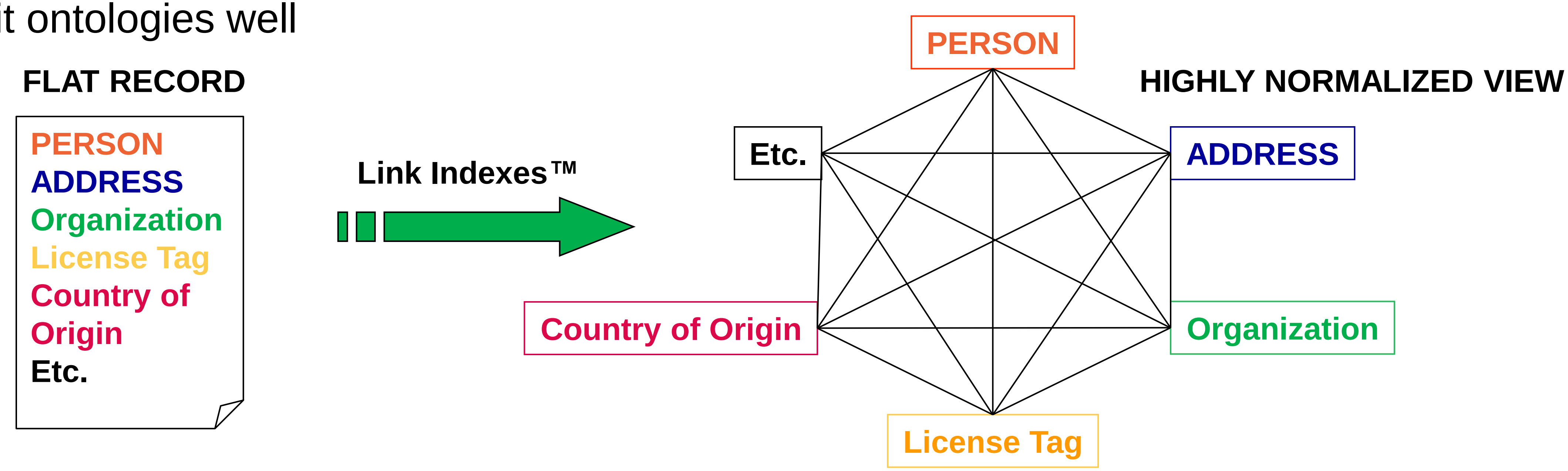
Auto Insurance Policy Link Index

Example content and Link Indexes™: Combined view (2 of 2)



Link Indexes™ normalize data

- Link Indexes™ offer highly normalized views of data; tending towards 5th Normal Form
  - Regardless of how or where data is stored
  - Change the way data is queried, retrieved and presented
  - Fit ontologies well



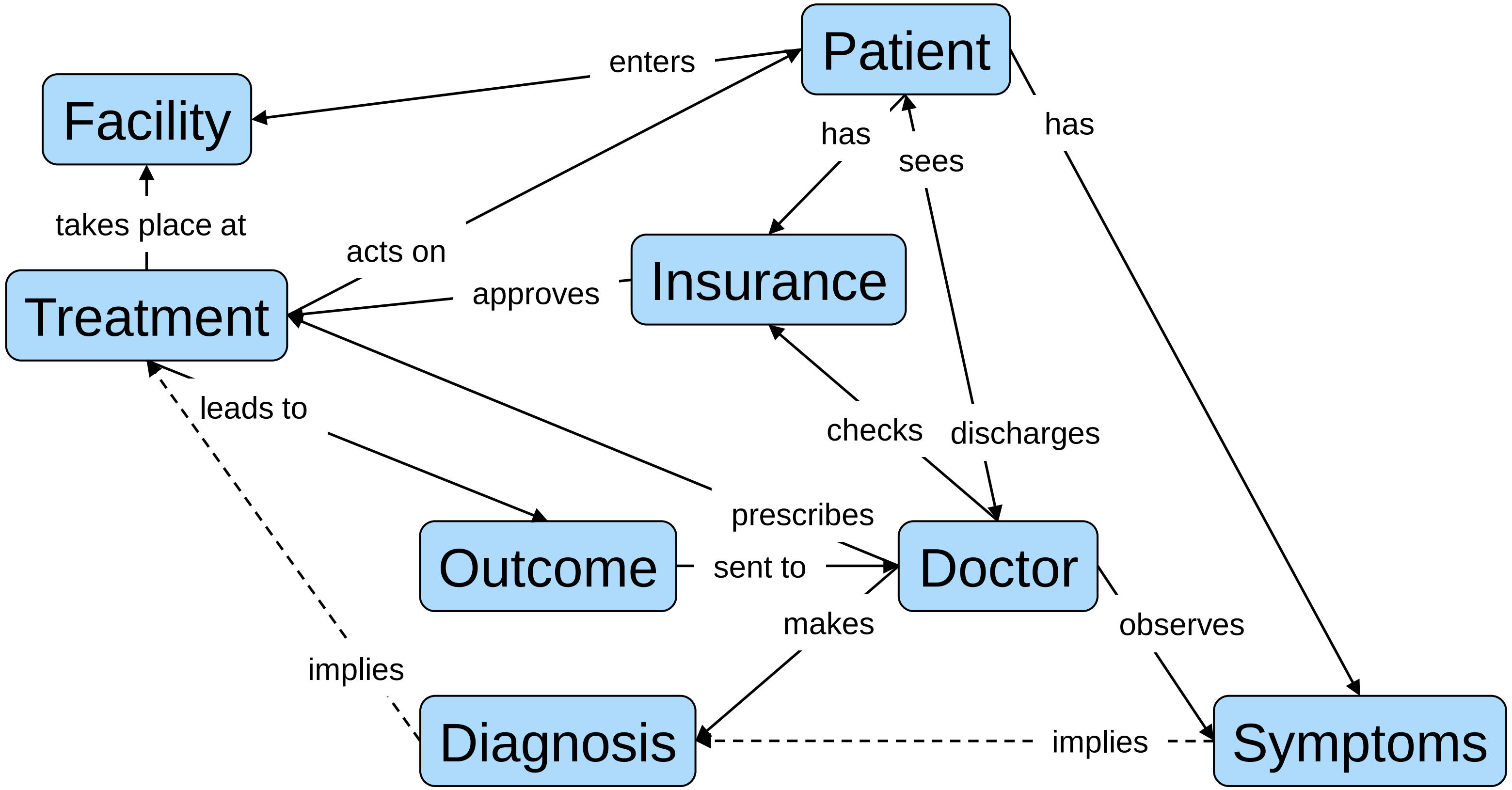
Other uses for Link Indexes™

- Create and maintain master data indexes
  - Predicated on good data quality in content indexes
  - Locate and map same or similar data within a data source and across multiple data sources
  - Similar data determined through fuzzy match indexes
  - Use composite weighted multi-attribute scoring to match complex entities such as PERSON, e.g., type CUSTOMER, EMPLYEE, PATIENT, DOCTOR, NURSE or ADMINISTRATOR
  - Create and maintain registry, repository and/or hybrid master data indexes
- Accelerate internal joins within a data source and external joins across multiple data sources
  - Support joins across segmented indexes by co-existing Link Indexes with content indexes
- In raw form with no entity definitions, data and schema discovery within a data sources and across multiple data sources
- In refined form with entity definitions, enable ontology/semantic modelling discovery within a data sources and across multiple data sources
- Eventually, could use to optimize query processing

EOS

Link Indexes™ help simplify complex solutions

For example, ontology representations, model mapping and query execution, in combination with BPM software. For more information, see separate presentation entitled “WhamTech Link Indexes and Ontologies.”



EIQ Adapters and Link Indexes overcome conventional graph database problems (1 of 2)

1. Data access
   - EIQ Adapters enable data warehouse success and capabilities, but using federated data access/data virtualization
   - Complete access to all data available in data sources using SQL for data selection, constraints/filters, and values for node and edge metadata
2. Low data quality
   - EIQ Adapters cleanse, transform and standardize data as it is being indexed and applies the same processing to raw results from data sources or uses results data from indexes
   - Cleansed, transformed and standardized indexes allow queries to be highly successful
3. Difficult to scale
   - EIQ Adapters operate through distributed federated data access, allowing scalability – only results consolidated at higher levels
4. Near real-time updates usually not possible
   - EIQ Adapter indexes are updated in near real-time and can be available immediately, or used to update any subscriptions or link models

EIQ Adapters and Link Indexes overcome conventional graph database problems (2 of 2)

5. No integration with MDM
   - MDM based on good data quality, automatically and logically groups entities and associated logical records, i.e., triples, while still retaining representations of physical records in data sources and associated metadata – essential for audit, drilldown and apps such as CCPA and GDPR
6. As data is extracted from sources, many inherent links and affiliations among data are lost
   - SmartData Fabric® can use almost any algorithm to capture and retain links among data – see earlier slide on Link Types
7. Absence of probabilities of links and confidence levels of data
   - When content is retrieved, the probabilities of matches can be computed on the fly and user-defined confidences in data sources included
8. Absence of threat/favorability, sentiment or other scores
   - SmartData Fabric® link analysis can use any available threat/favorability data to create associative threat/favorability, sentiment or other link models

The End

Appendix: Backup slides

Link mapping to build and maintain Link Indexes™

Past work on link analysis using Link Indexes™

Physical and associative to logical graphs

Graph visualization

Living Networks™

Link Indexes™ metadata

Link mapping to build and maintain Link Indexes™

Link mapping process (1 of 2)

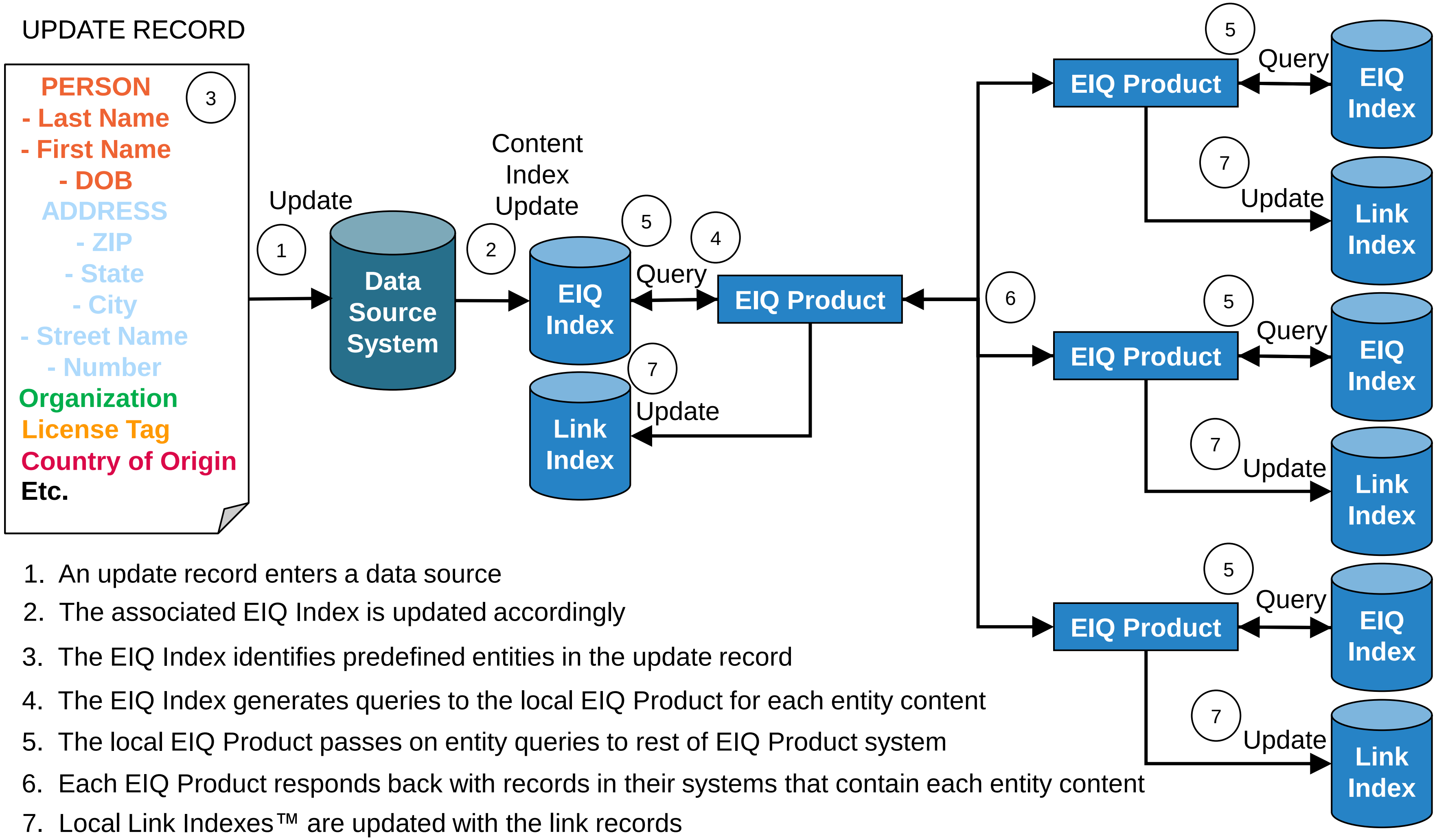
1. Start with a logical entity-relationship/ontological/semantic data model specific to a market or community of interest (COI), e.g., HL7 and NIEM, to identify relevant entities to track
2. Using the same logical data model from step 1 as an EIQ Adapter standard data model, designate entities (and associated attributes) to be tracked for potential links, e.g., PERSON [last name + first name + DOB + SSN + EMAIL + PHONE], ADDRESS [ZIP + state + city + street name + number + suite/PO box], VEHICLE, COUNTRY_OF_ORIGIN, EMAIL, PHONE, ORGANIZATION, DOB, SSN, etc.
3. When a new, updated or deleted record from the data source containing designated entity/entities updates indexes, query EIQ Adapter system with that entity/these entities, passing the originating data source record pointer with the queries
   a. For complex entities, e.g., PERSON, use composite weighted multi-attribute probabilistic match and master data merge techniques to determine (i) an above threshold match and therefore link, and (ii) a master data record containing best data (see separate presentation on Virtual MDM for more)
   b. For simple entities, e.g., DOB, PHONE and SSN, there is no need for master data for link analysis/graph database, as data quality allows for an exact match. Note: Same for ADDRESS even though a complex entity

Link mapping process (2 of 2)

4. If the entity/entities query is matched, obtain result-sets of record pointers from other data source adapters containing matched entity/entities
   a. Note: Result-set data does not need to be read from data sources to obtain result-set record pointers; only the indexes to data sources need to be read – higher performance
   b. Note: All data associated with the result-set records is now indirectly associated with all data associated with the originating data source record, i.e., if a record contains n entities, then there are n(n-1)/2 associations among these entities -- some of these associations make sense; others do not and the logical data model is eventually used to filter the nonsensical associations out
5. Update the queried data source associated local Link Index™ with result-set record pointers and originating data source record pointer
6. Pass queried data source pointers back as a result-set to the originating data source EIQ Adapter and update the associated local Link Index

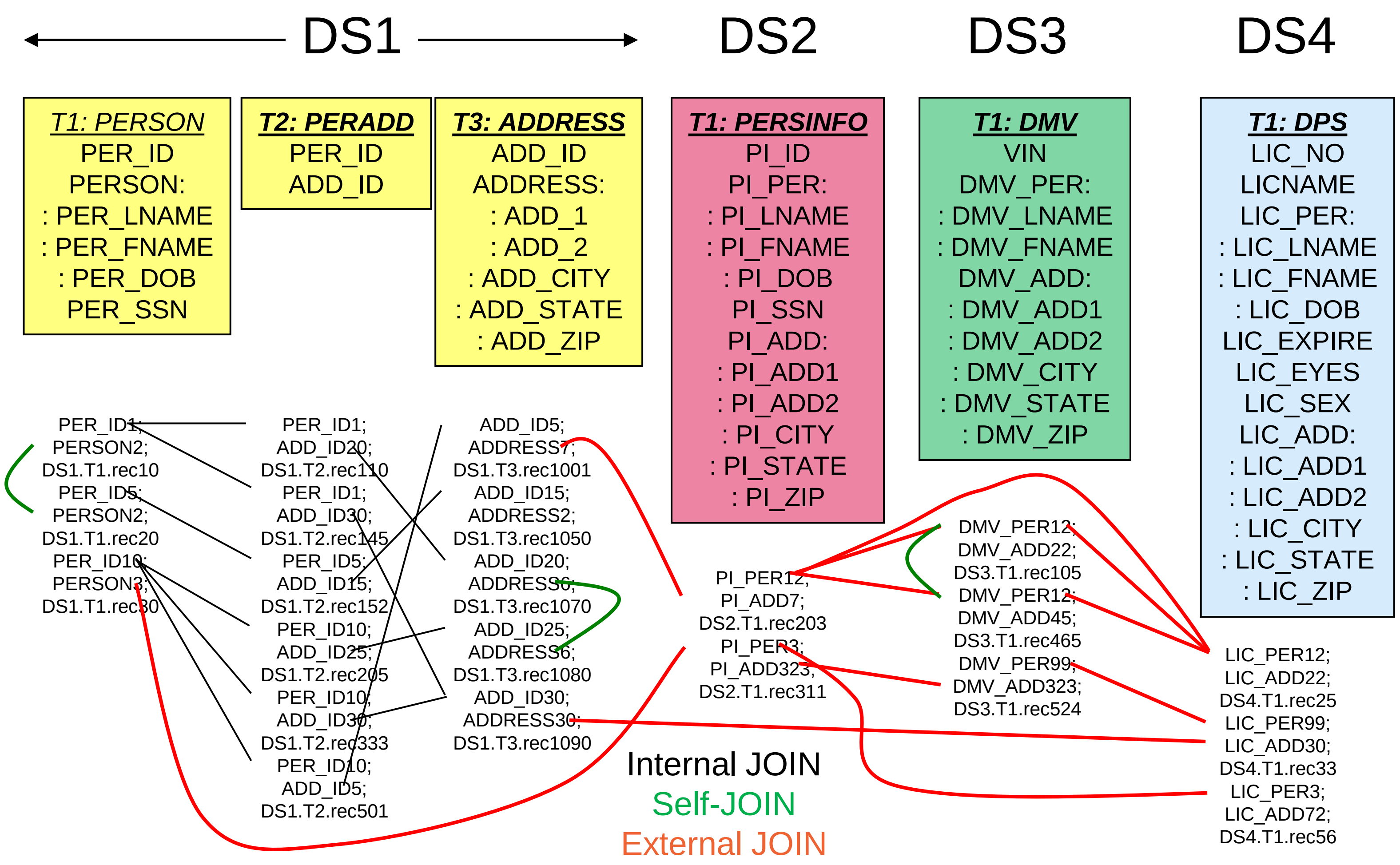
Note: Link Indexes can reside locally, regionally or centrally, but remain 100% contiguous

Link mapping process

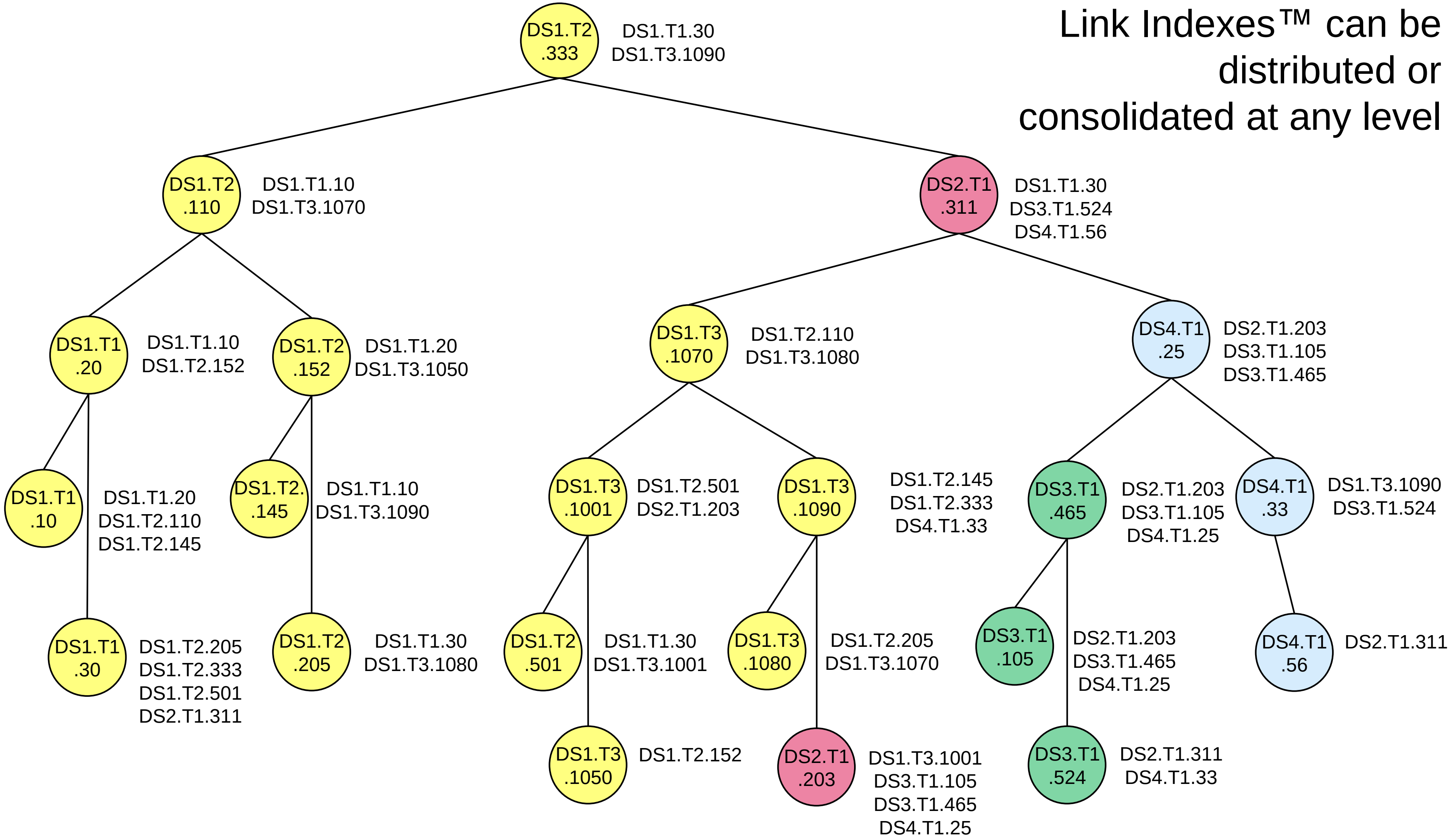


1. An update record enters a data source
2. The associated EIQ Index is updated accordingly
3. The EIQ Index identifies predefined entities in the update record
4. The EIQ Index generates queries to the local EIQ Product for each entity content
5. The local EIQ Product passes on entity queries to rest of EIQ Product system
6. Each EIQ Product responds back with records in their systems that contain each entity content
7. Local Link Indexes™ are updated with the link records

Link mapping process



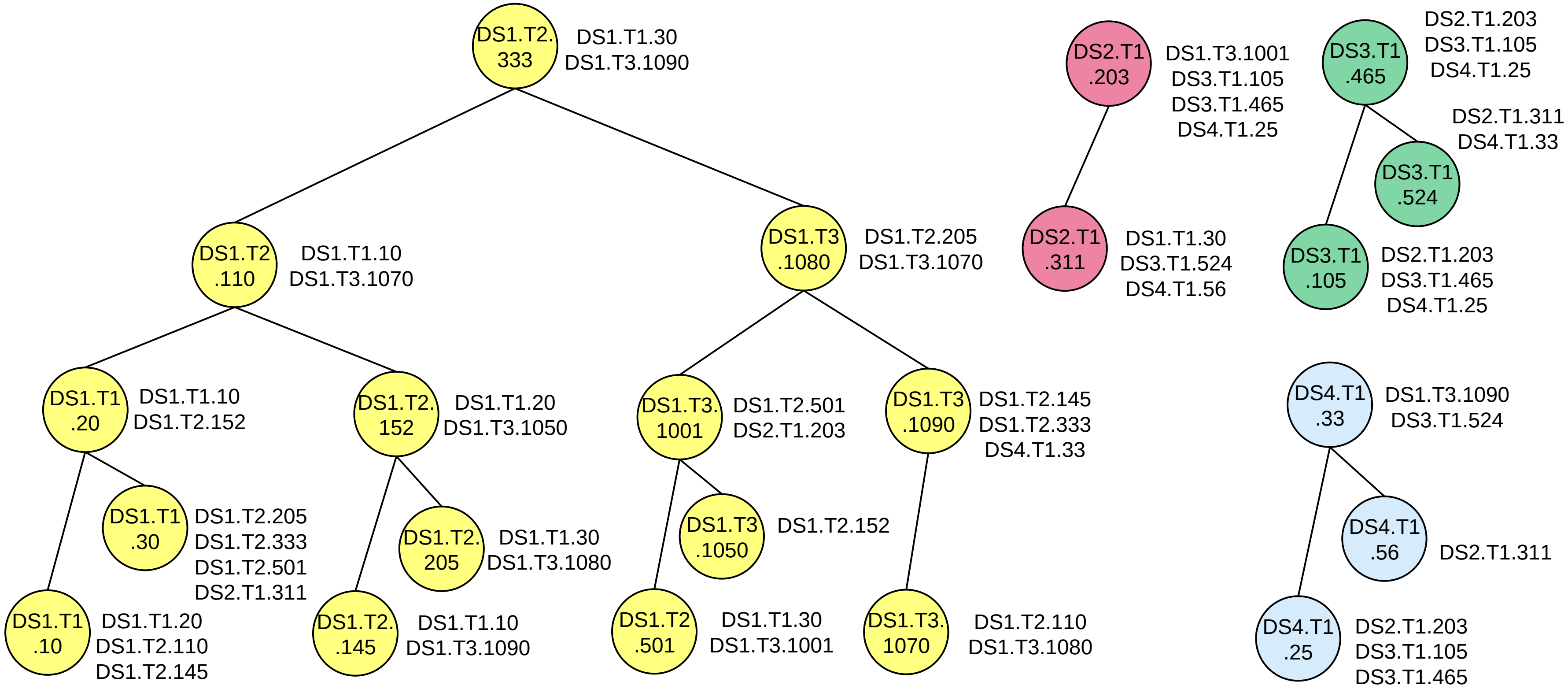
Example of a consolidated central Link Index™



Link Indexes™ can be distributed or consolidated at any level

Example of a distributed Link Index™

Link Indexes™ can be distributed or consolidated at any level



Past work on link analysis using Link Indexes™

Link networks building

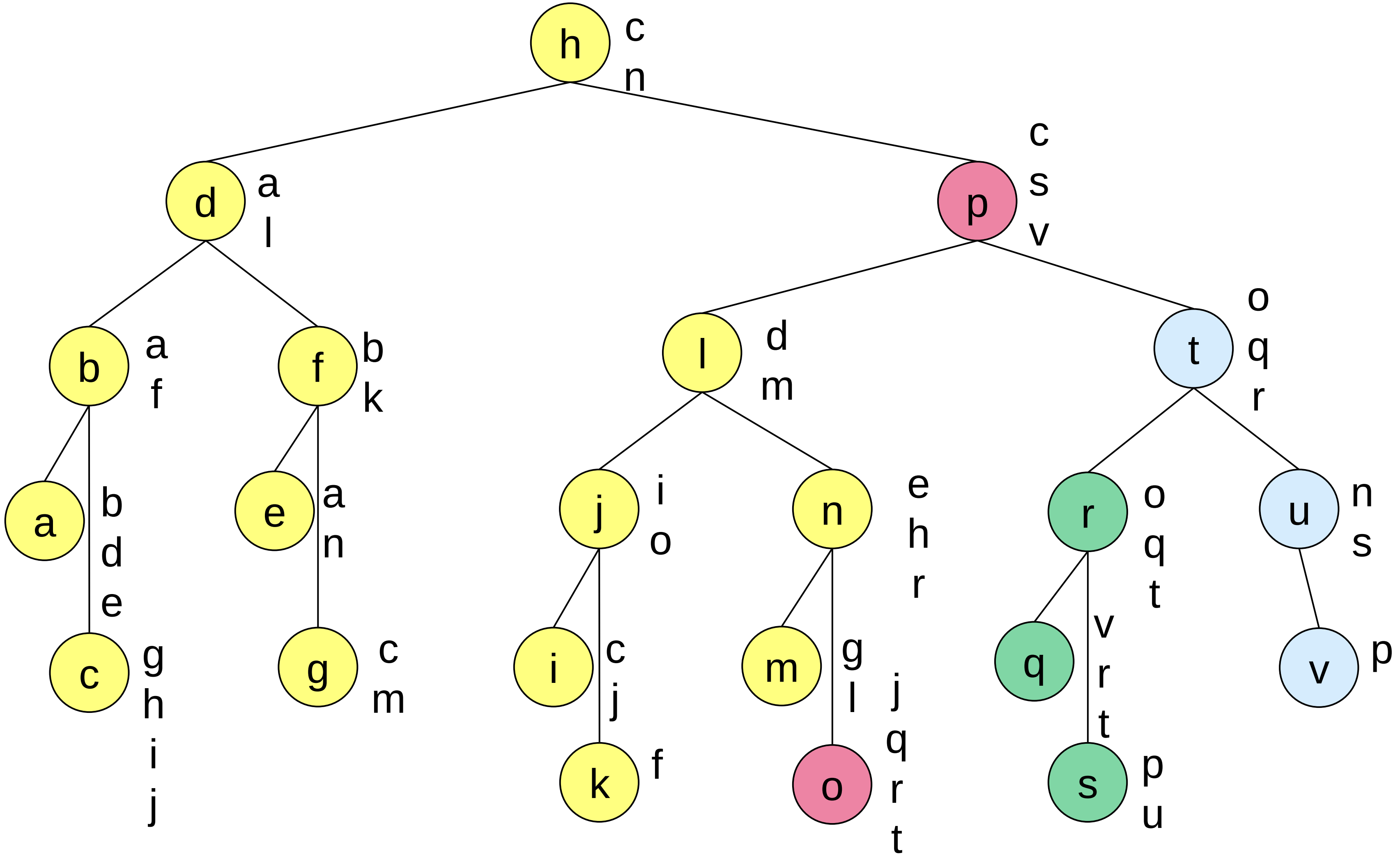
- Combination of walking through Link Indexes™ and Boolean operations on bitmaps to obtain initial network
  - Analysis same for central, regional and/or local indexes (content, Link Indexes™ and master data)
  - Virtual bitmaps in most cases
  - Physical bitmaps for very low cardinality data only
  - Decision made between virtual and physical bitmaps at each index tree node
  - Boolean link operations generally occur in memory or cache and are therefore very fast
  - Link operations on physical links from data sources as per the Link Indexes™
  - Logical Data model ensures that only logical links are filtered in from link operations
  - Multiple structures from multiple link types can be combined or filtered in/out

Link analyses

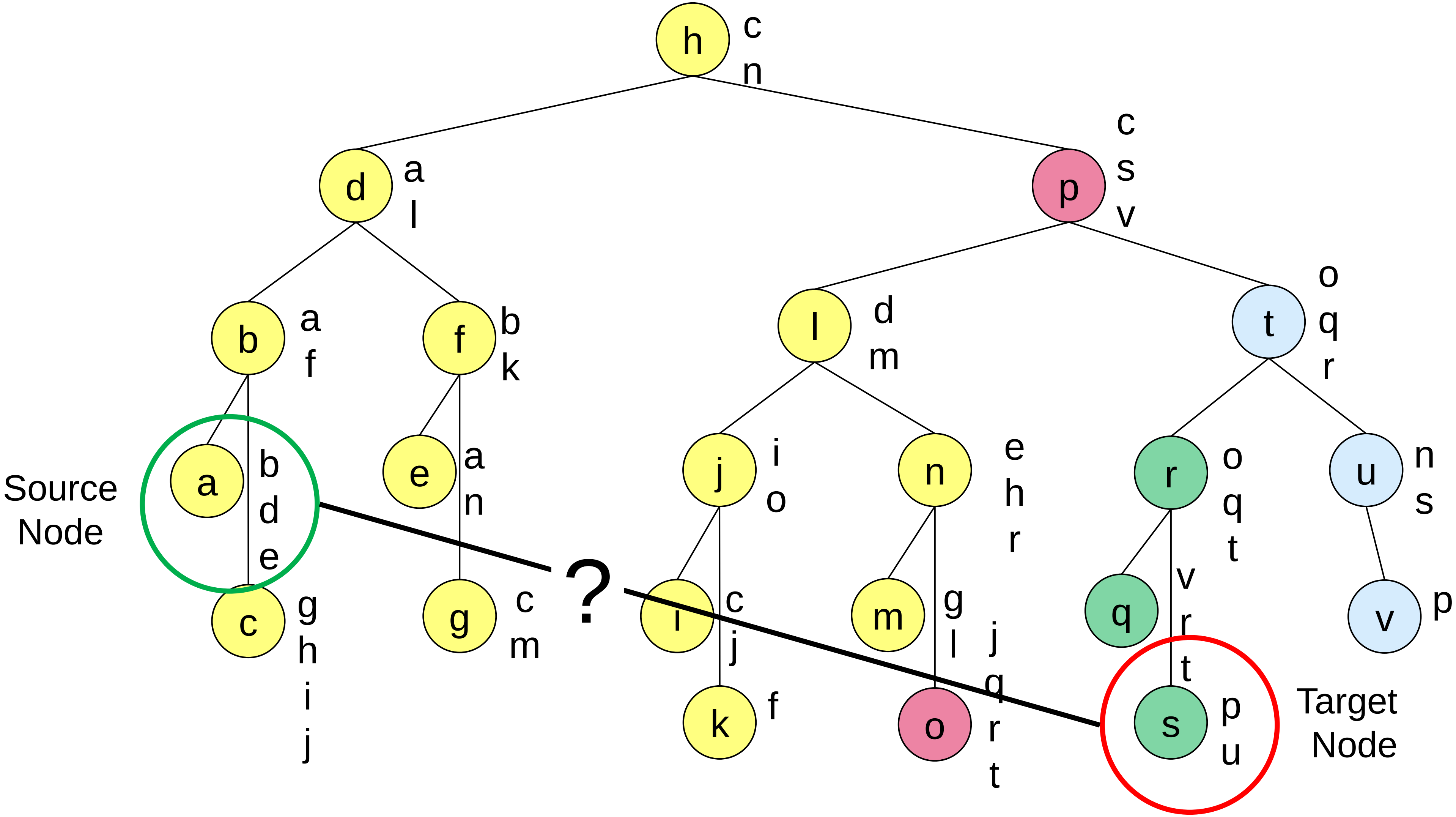
DEMO

1. One (node) to one (node)
2. One to many
3. Many to one
   (2–3: Similar)
4. Many to many
5. One to none
6. None to one
   (5–6: Similar)
7. None to none

Simplified version of previous consolidated central Link Index™



Example of a one (node) to one (node) link analysis



Link analysis step 1

Try to direct connect source Node a and target Node s

```
        abcdefghijklmnopqrstuv
Node a  0101100000000000000000
AND
Node s  0000000000000001000010
------------------------------
     =  0000000000000000000000
```

Link analysis step 2

Select the first node referenced by Node a: Node b

```
       abcdefghijklmnopqrstuv
Node a 0101100000000000000000
```

Link analysis step 3

Try to connect Node d and Node s

```
        abcdefghijklmnopqrstuv
Node b  1000010000000000000000
   AND
Node s  0000000000000000100010
------------------------------
     =  0000000000000000000000
```

Link analysis step 4

Select the second node referenced by Node a: Node d

```
         abcdefghijklmnopqrstuv
Node a   0101100000000000000000
```

Link analysis step 5

Try to connect Node d and Node s

	abcdefghijklmnopqrstuv
Node d	1000000000010000000000
AND	
Node s	0000000000000001000010
=	0000000000000000000000

Link analysis step 6

Select the third node referenced by Node a: Node e

```
       abcdefghijklmnopqrstuv
Node a 0101100000000000000000
```

Link analysis step 7

Try to connect Node e and Node s

	abcdefghijklmnopqrstuv
Node e	1000000000000100000000
AND	
Node s	0000000000000001000010
=	0000000000000000000000

Link analysis step 8

As no more referenced nodes at Node a,
select the first node referenced by Node s: Node p

```
        abcdefghijklmnopqrstuv
Node s  0000000000000001000010
```

Link analysis step 9

Try to connect the first node referenced by Node a: Node d, with the first node referenced by Node s: Node p

```
        abcdefghijklmnopqrstuv
Node b  1000010000000000000000
   AND
Node p  0010000000000000001001
--------------------------------
     =  0000000000000000000000
```

WHAMTECH®

Link analysis step 10

Try to connect the second node referenced by Node a: Node d, with the first node referenced by Node s: Node p

```
        abcdefghijklmnopqrstuv
Node d  1000000000010000000000
AND
Node p  0010000000000000001001
------------------------------
=       0000000000000000000000
```

Link analysis step 11

Try to connect the third node referenced by Node a: Node e, with the first node referenced by Node s: Node p

```
        abcdefghijklmnopqrstuv
Node e  1000000000000100000000
AND
Node p  0010000000000000001001
-------------------------------
     =  0000000000000000000000
```

Link analysis step 12

Select the second node referenced by Node s: Node u

```
         abcdefghijklmnopqrstuv
Node s | 0000000000000001000010 |
```

Link analysis step 13

Try to connect the first node referenced by Node a:
Node b, with the second node referenced by Node s: Node u

```
        abcdefghijklmnopqrstuv
Node b  1000010000000000000000
AND
Node u  0000000000000100001000
------------------------------
=       0000000000000000000000
```

WHAMTECH®

Link analysis step 14

Try to connect the second node referenced by Node a:
Node d, with the second node referenced by Node s: Node u

```
        abcdefghijklmnopqrstuv
Node d  1000000000010000000000
   AND
Node u  0000000000000100001000
------------------------------
     =  0000000000000000000000
```

WHAMTECH®

Link analysis step 15

Try to connect the third node referenced by Node a:
Node e, with the second node referenced by Node s: Node u

```
        abcdefghijklmnopqrstuv
Node e  1000000000000100000000
AND
Node u  0000000000000100001000
-------------------------------
=       0000000000000100000000
```

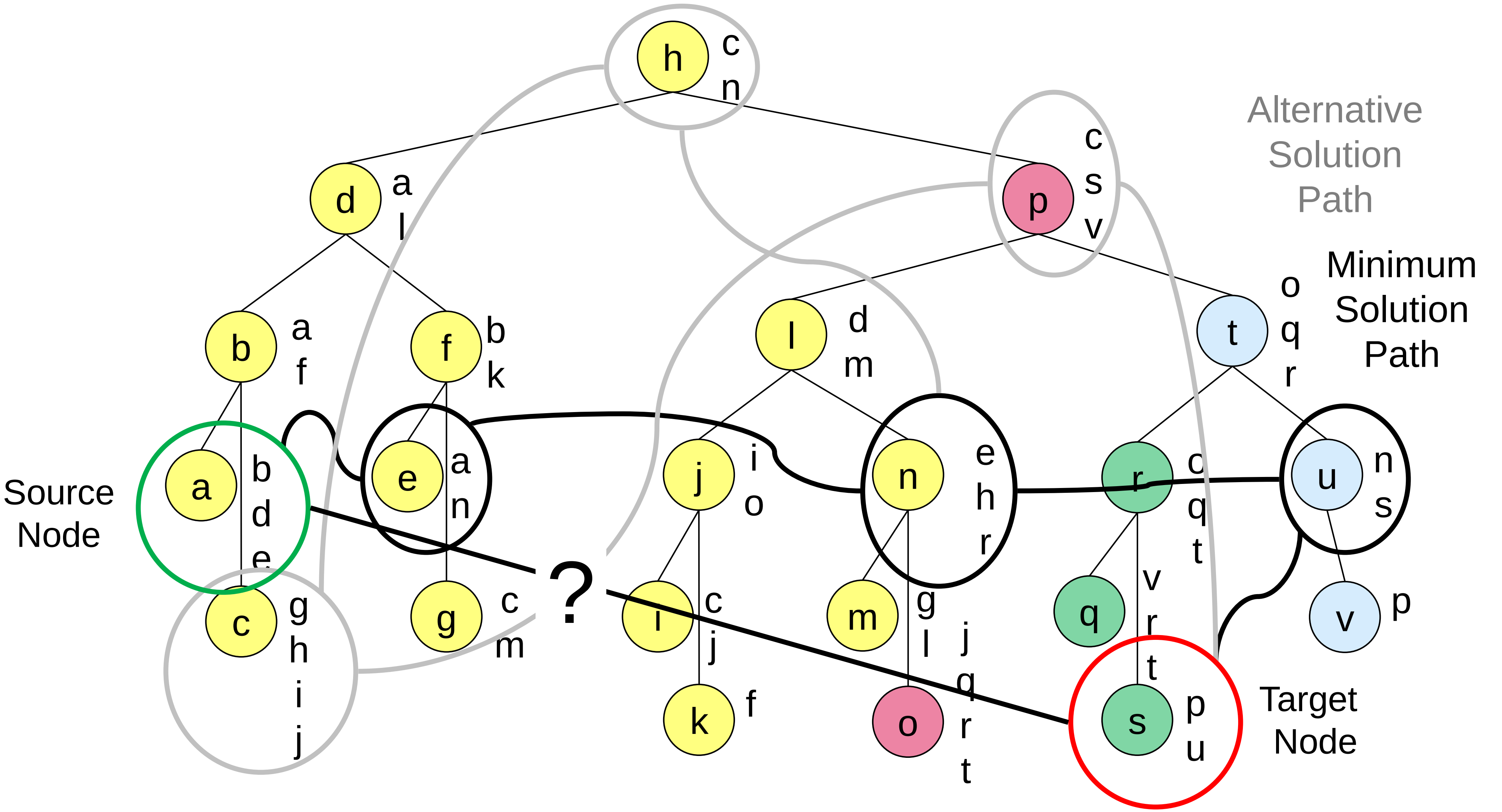
Link analysis step 16

Solution	abcdefghijklmnopqrstuv
List	1000100000000100001010

Solution Path a -> e -> n -> u -> s

Alternative Solution Path a -> e -> n -> h -> c -> p -> s

Solution to an one (node) to one (node) link analysis



What is the best solution?

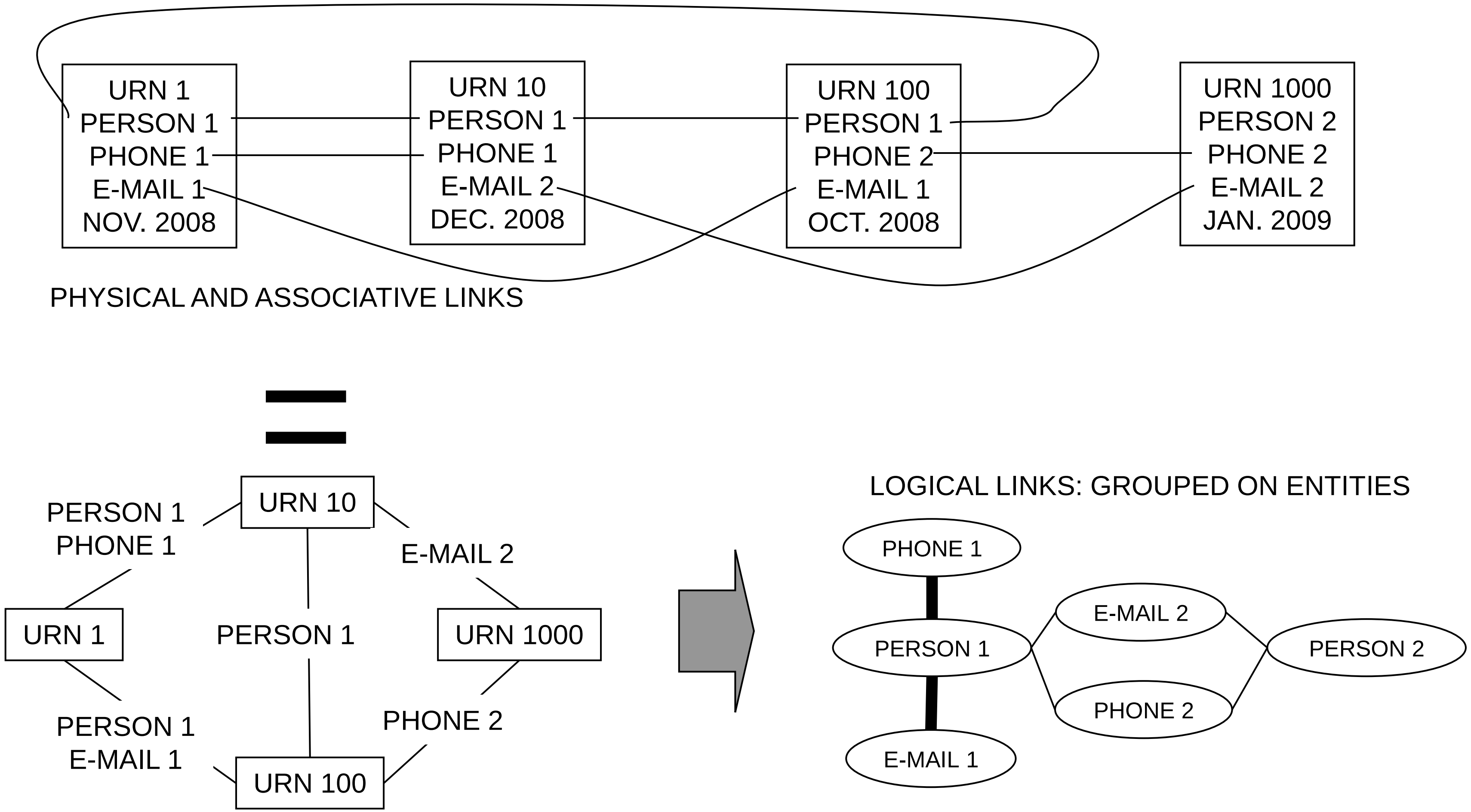
- Maybe the alternative solution is the better one as:
  - Links between Nodes n -> u -> s could have a lower probability and/or the nodes a lower confidence
  - Links between Nodes n -> h -> c ->p -> s could have a higher probability and/or nodes a higher confidence

Physical and associative to logical graphs

Physical and associative links to logical Links

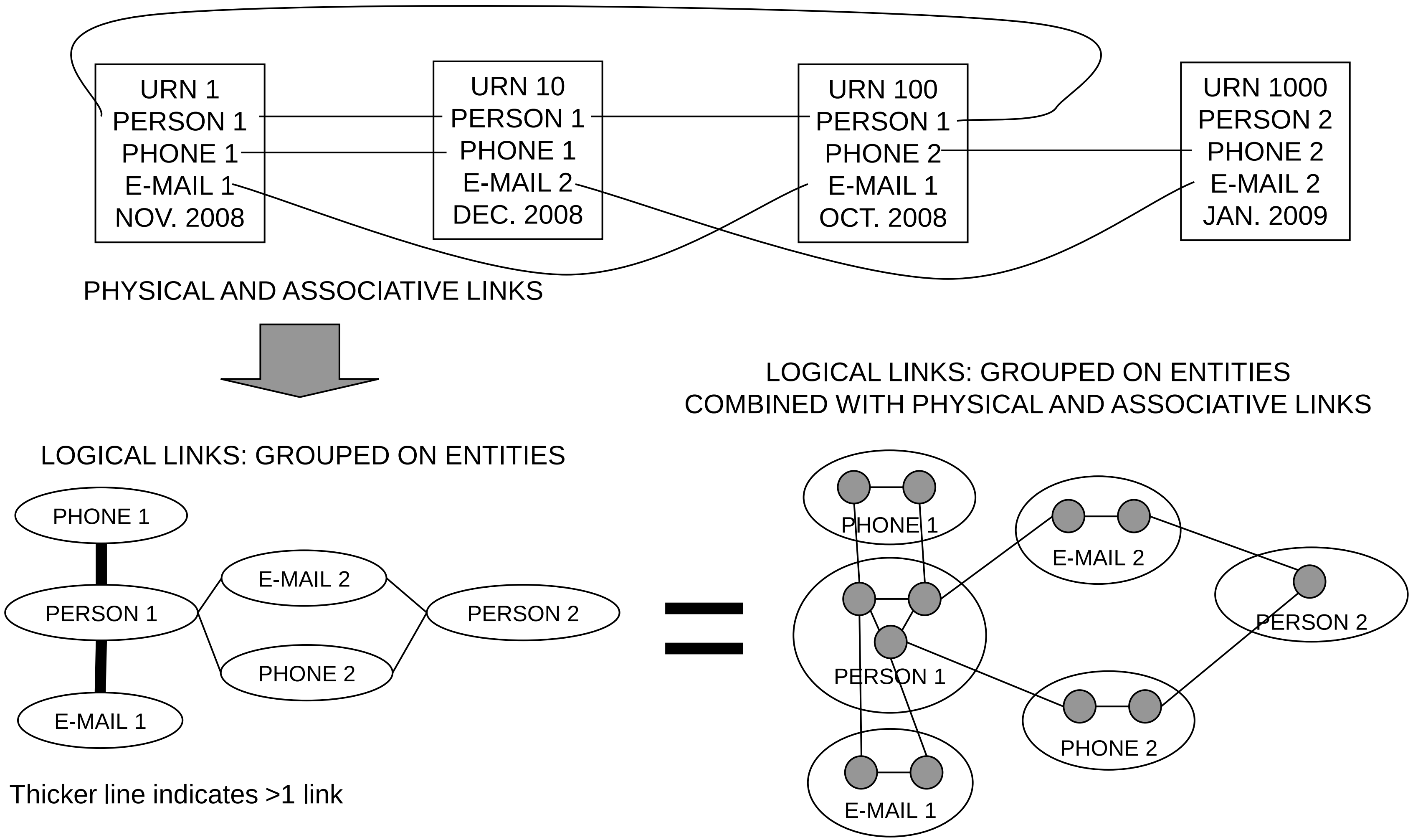
- Physical links are the connections captured in Link Indexes™ within and across data sources, e.g. PK-FK relationships are links between database records
- Associative links are the connections among entities, e.g., PERSON 1 and PHONE 1, that can be in the same record, so, there is no physical record link between these two entities, as it is the same record, but there is an associative link that may or not make sense, i.e., they may or may not fit a logical data model
  - Other associative links may not make sense, e.g., SSN 1 and PHONE 1
- In general, the physical and associative links will always be >= the logical links
- Boolean operations for join execution, degrees of separation queries and link analysis are always performed on physical links, and then pruned to fit the logical link query
- Physical and logical links coexist and can be presented on the same display
- Generally, link queries, including degrees of separation (DOS) qualifiers, are on Logical Data Models, i.e., logical entities (nodes) and links (edges) between logical entities

Physical and associative links to logical Links

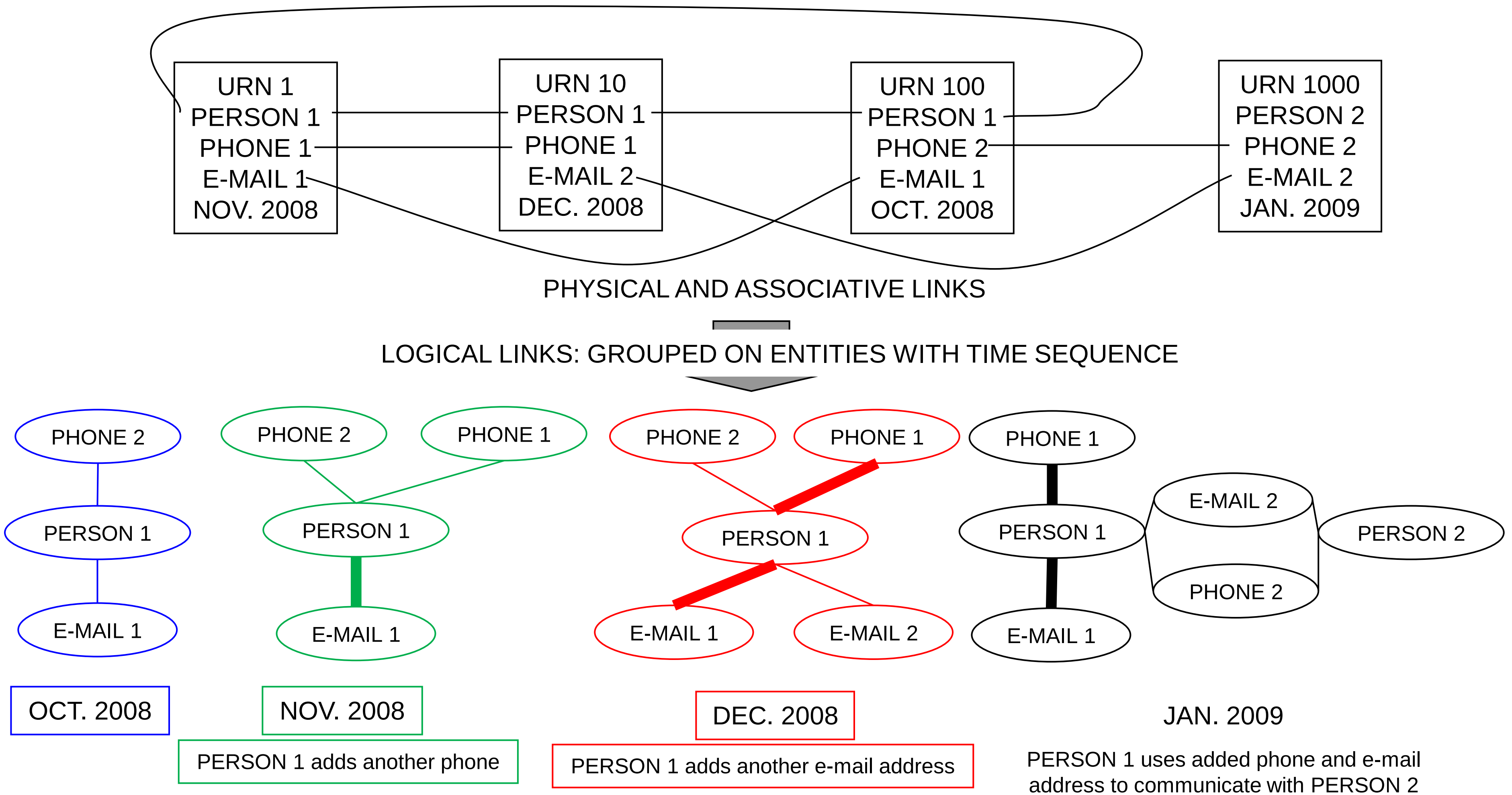


Thicker line indicates >1 link

Combined physical, associative and logical Links

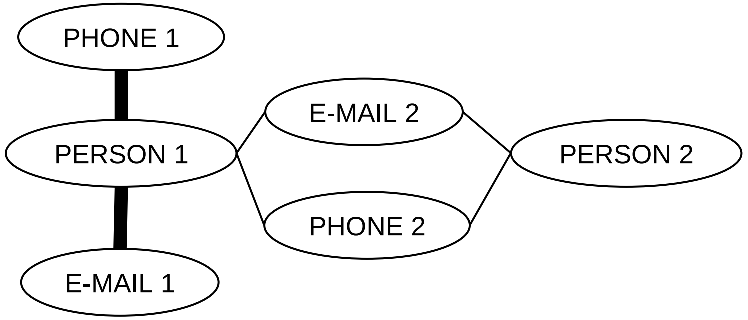


Combined content and links, e.g., using timeline filters

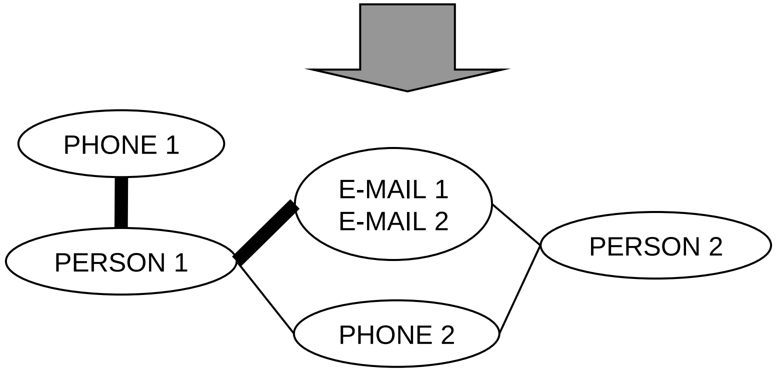


Manual interaction with logical links, e.g., entity grouping

LOGICAL LINKS: GROUPED ON ENTITIES



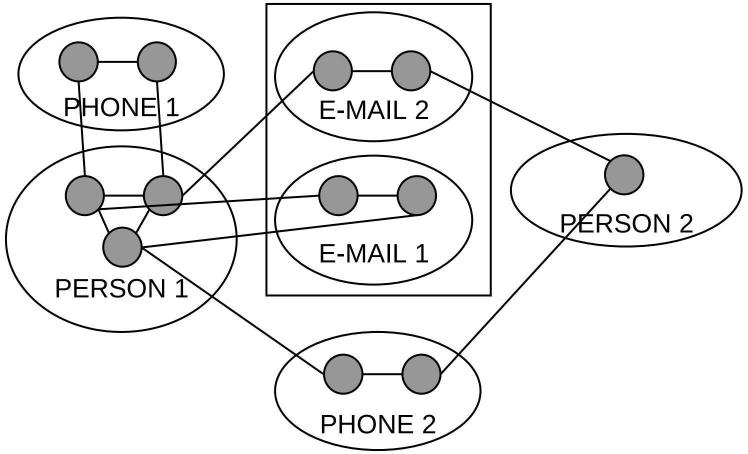
Manually group E-MAIL 1 and E-MAIL 2, as considered the same



Thicker line indicates >1 link

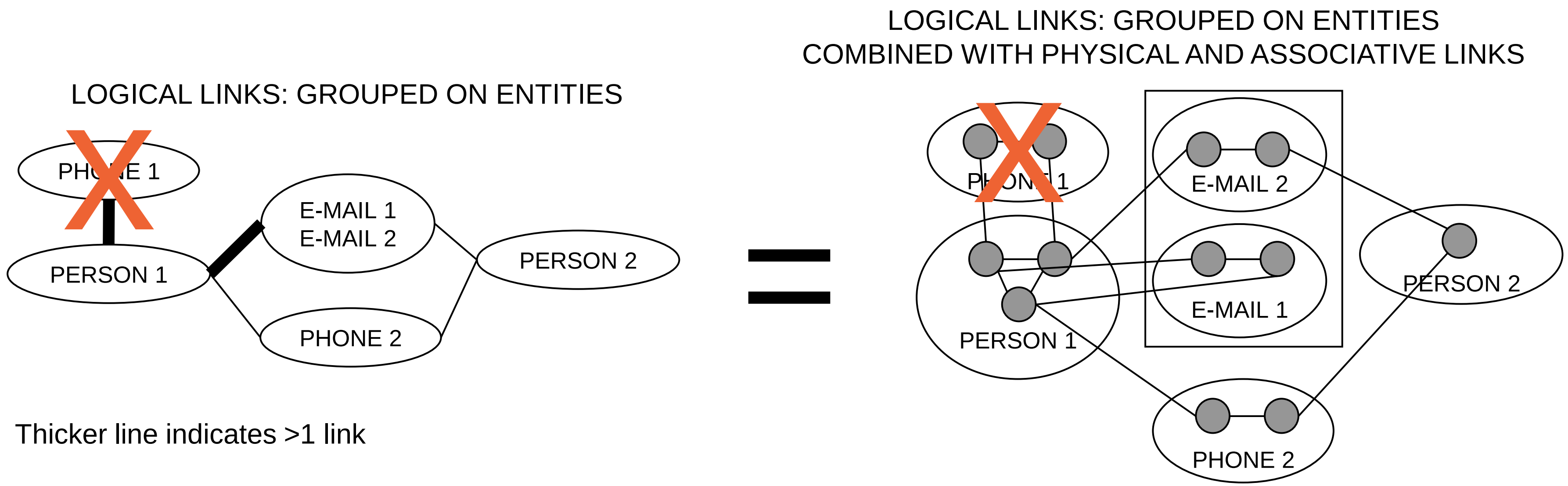
=

LOGICAL LINKS: GROUPED ON ENTITIES
COMBINED WITH PHYSICAL AND ASSOCIATIVE LINKS

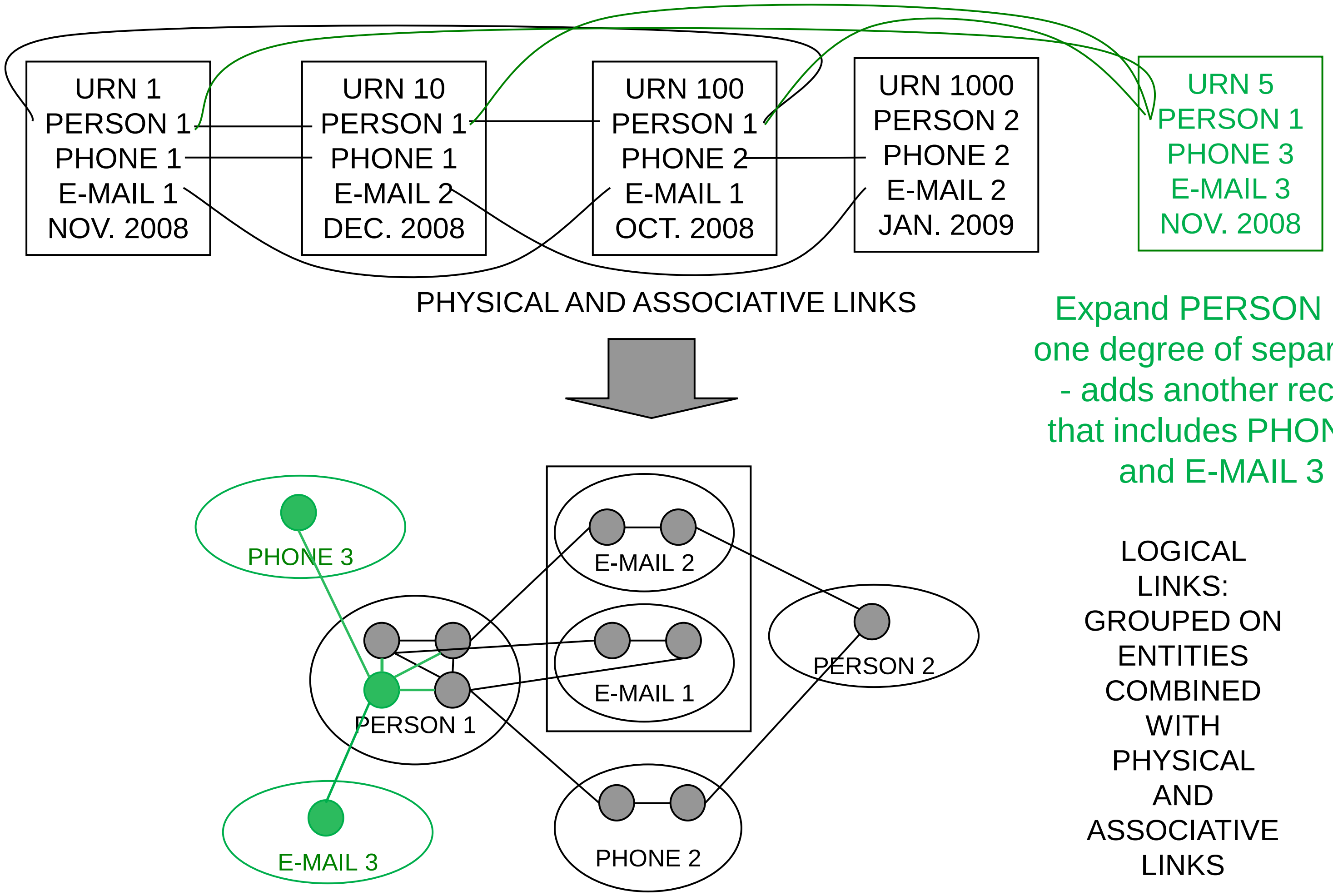


Manual interaction with logical links, e.g., entity deletion

An entity is (or entities are) deleted from the logical links. Physical links are retained.



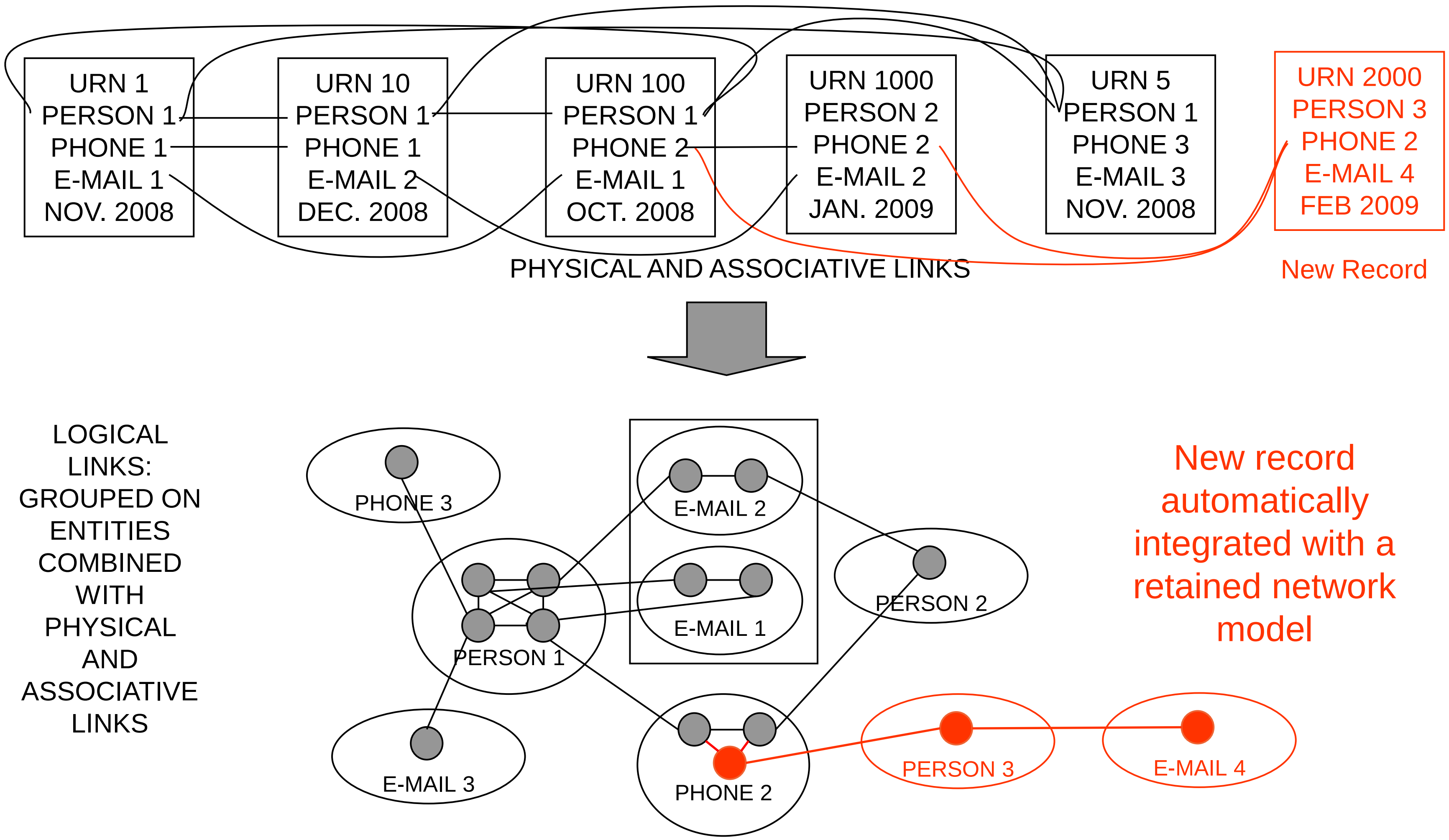
Manual interaction with logical links, e.g., entity expansion



Expand PERSON 1 to one degree of separation - adds another record that includes PHONE 3 and E-MAIL 3

LOGICAL LINKS: GROUPED ON ENTITIES COMBINED WITH PHYSICAL AND ASSOCIATIVE LINKS

Near real-time update of a graph

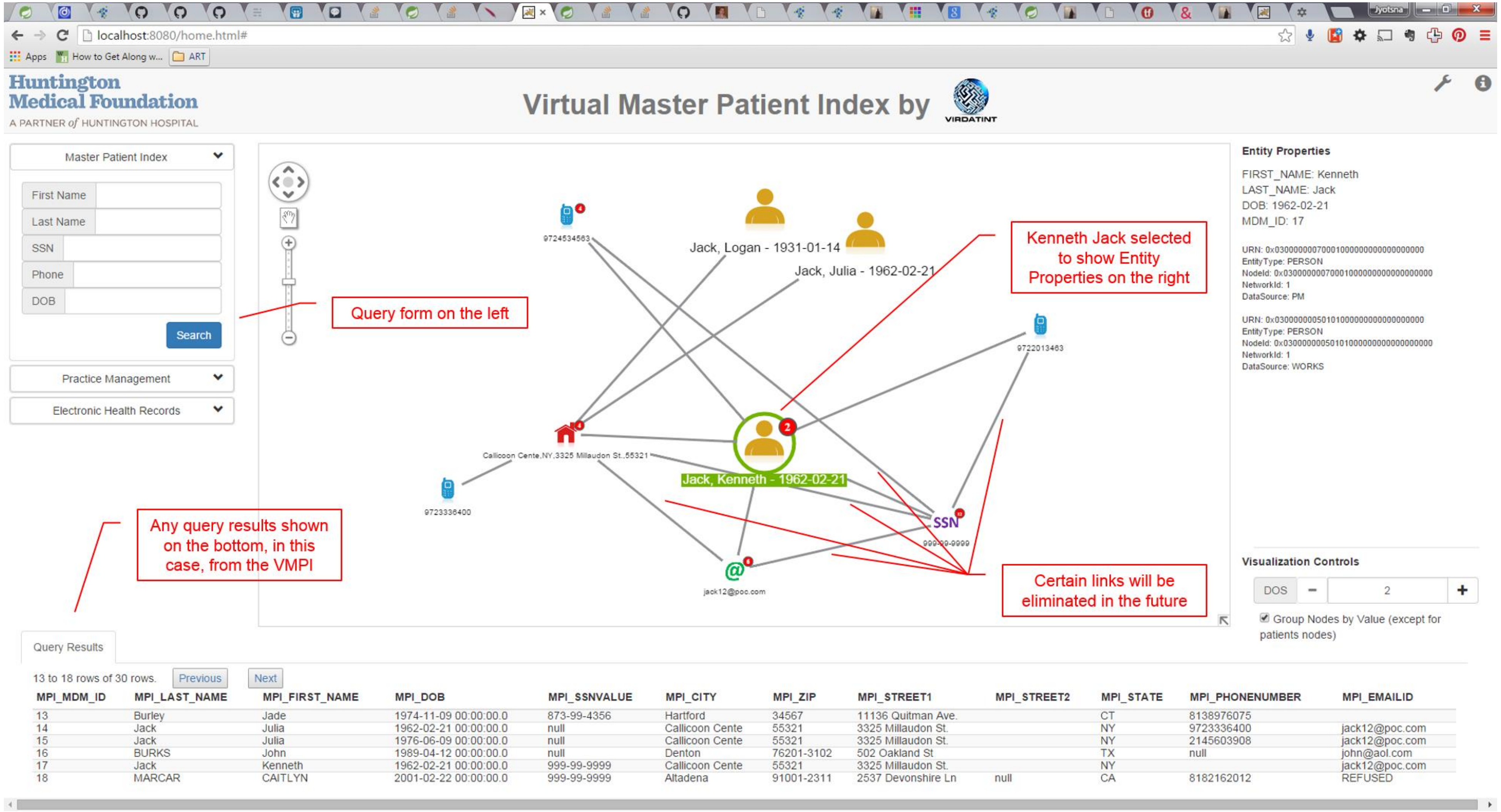


New record automatically integrated with a retained network model

Link Indexes™ can work with almost any link visualization software

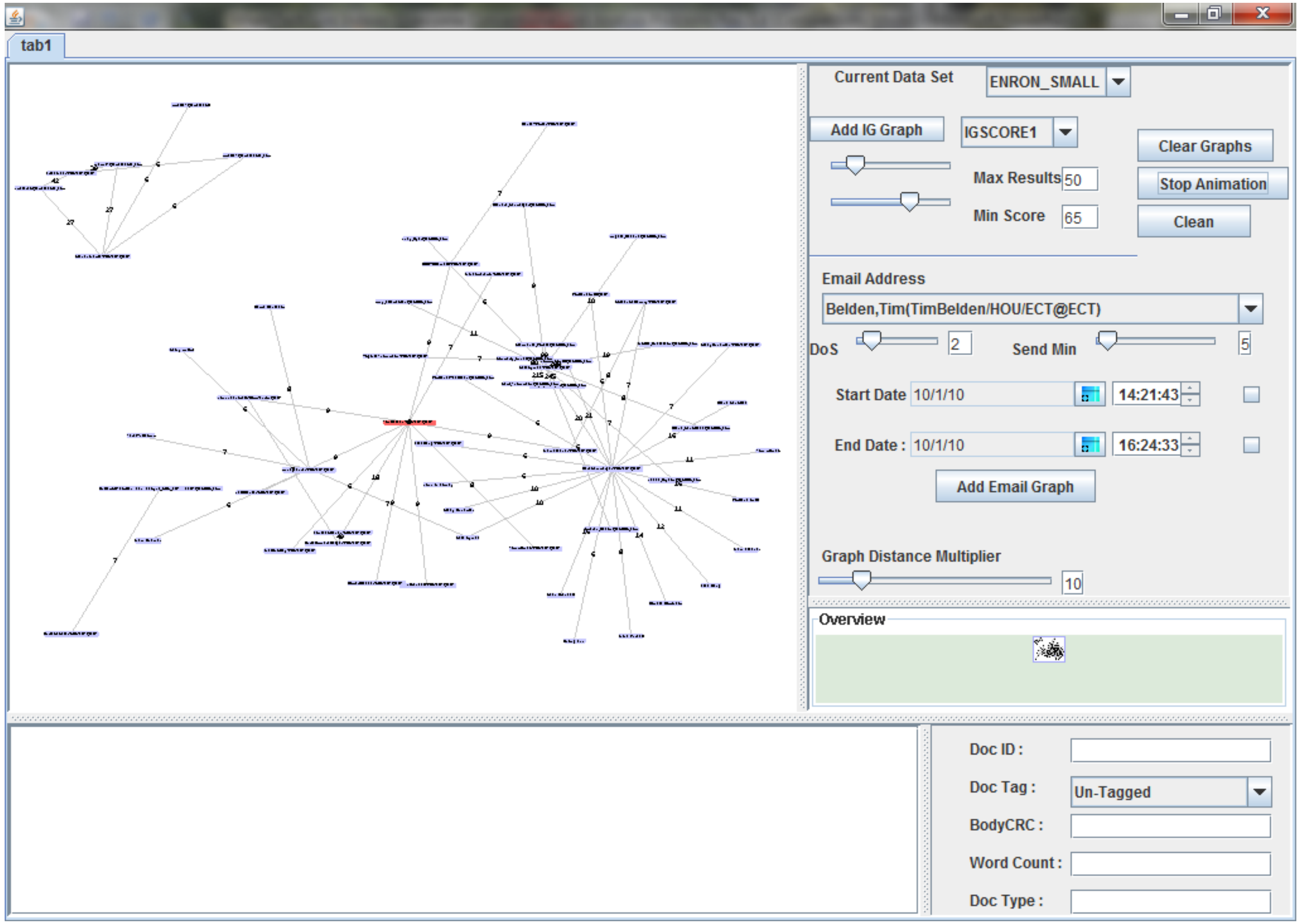
WhamTech has middleware that manages data access and link analysis

Example from WhamTech's general query interface, based on commercial KeyLines



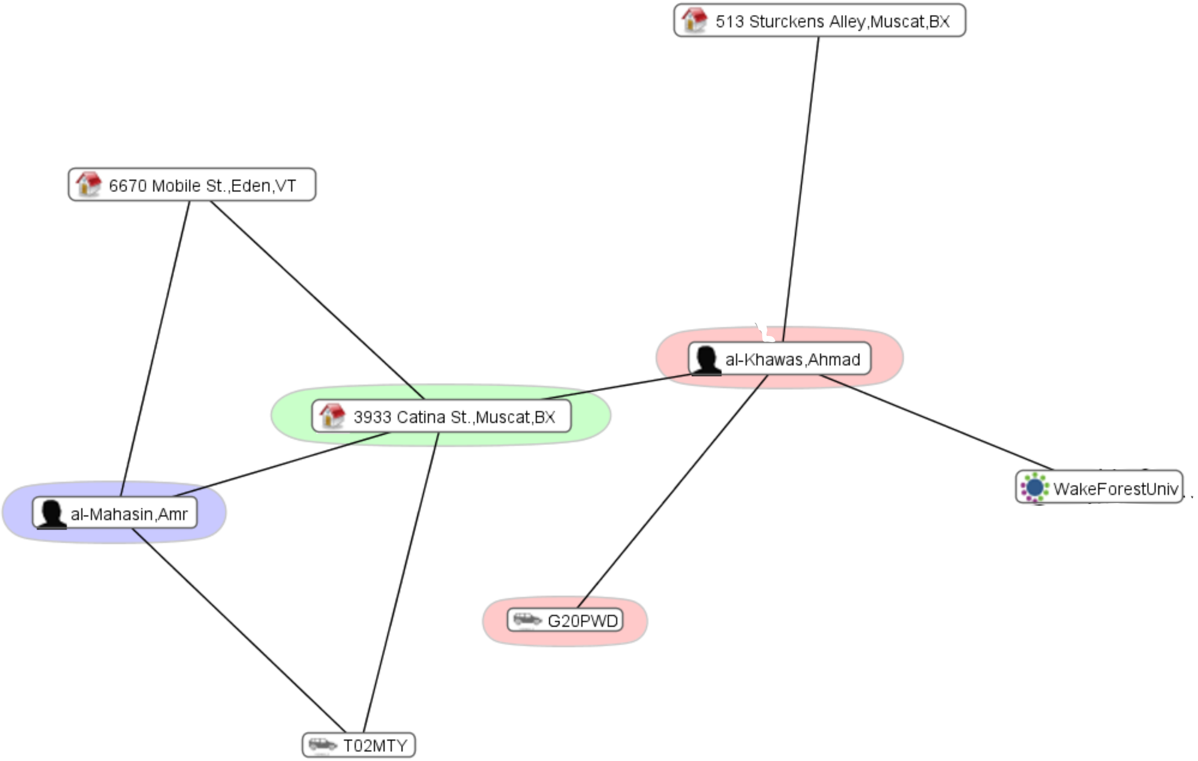
MPI_MDM_ID	MPI_LAST_NAME	MPI_FIRST_NAME	MPI_DOB	MPI_SSNVALUE	MPI_CITY	MPI_ZIP	MPI_STREET1	MPI_STREET2	MPI_STATE	MPI_PHONENUMBER	MPI_EMAILID
13	Burley	Jade	1974-11-09 00:00:00.0	873-99-4356	Hartford	34567	11136 Quitman Ave.		CT	8138976075	
14	Jack	Julia	1962-02-21 00:00:00.0	null	Callicoon Cente	55321	3325 Millaudon St.		NY	9723336400	jack12@poc.com
15	Jack	Julia	1976-06-09 00:00:00.0	null	Callicoon Cente	55321	3325 Millaudon St.		NY	2145603908	jack12@poc.com
16	BURKS	John	1989-04-12 00:00:00.0	null	Denton	76201-3102	502 Oakland St		TX	null	john@aol.com
17	Jack	Kenneth	1962-02-21 00:00:00.0	999-99-9999	Callicoon Cente	55321	3325 Millaudon St.		NY		jack12@poc.com
18	MARCAR	CAITLYN	2001-02-22 00:00:00.0	999-99-9999	Altadena	91001-2311	2537 Devonshire Ln	null	CA	8182162012	REFUSED

Example from WhamTech legacy eDiscovery tool, Teracase, based on open source Gephi

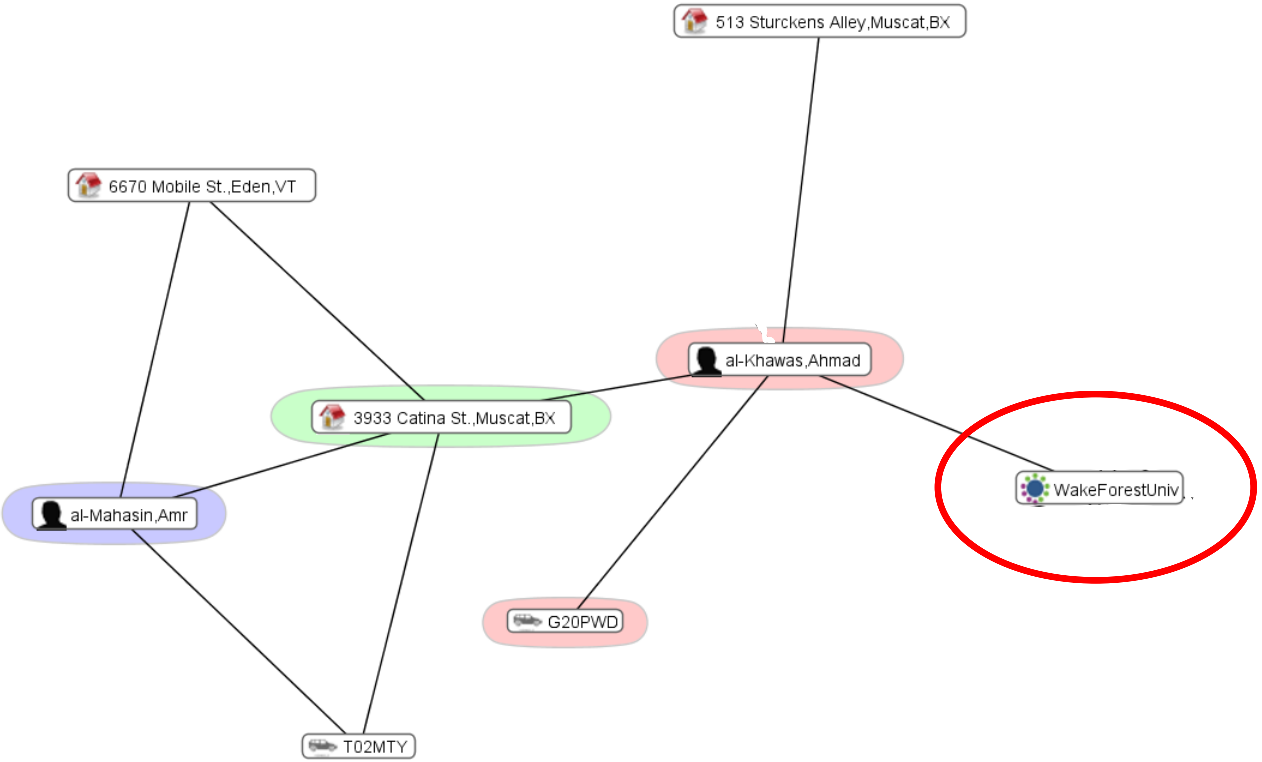


Simple graph visualization using open source network visualization tool, Pajek

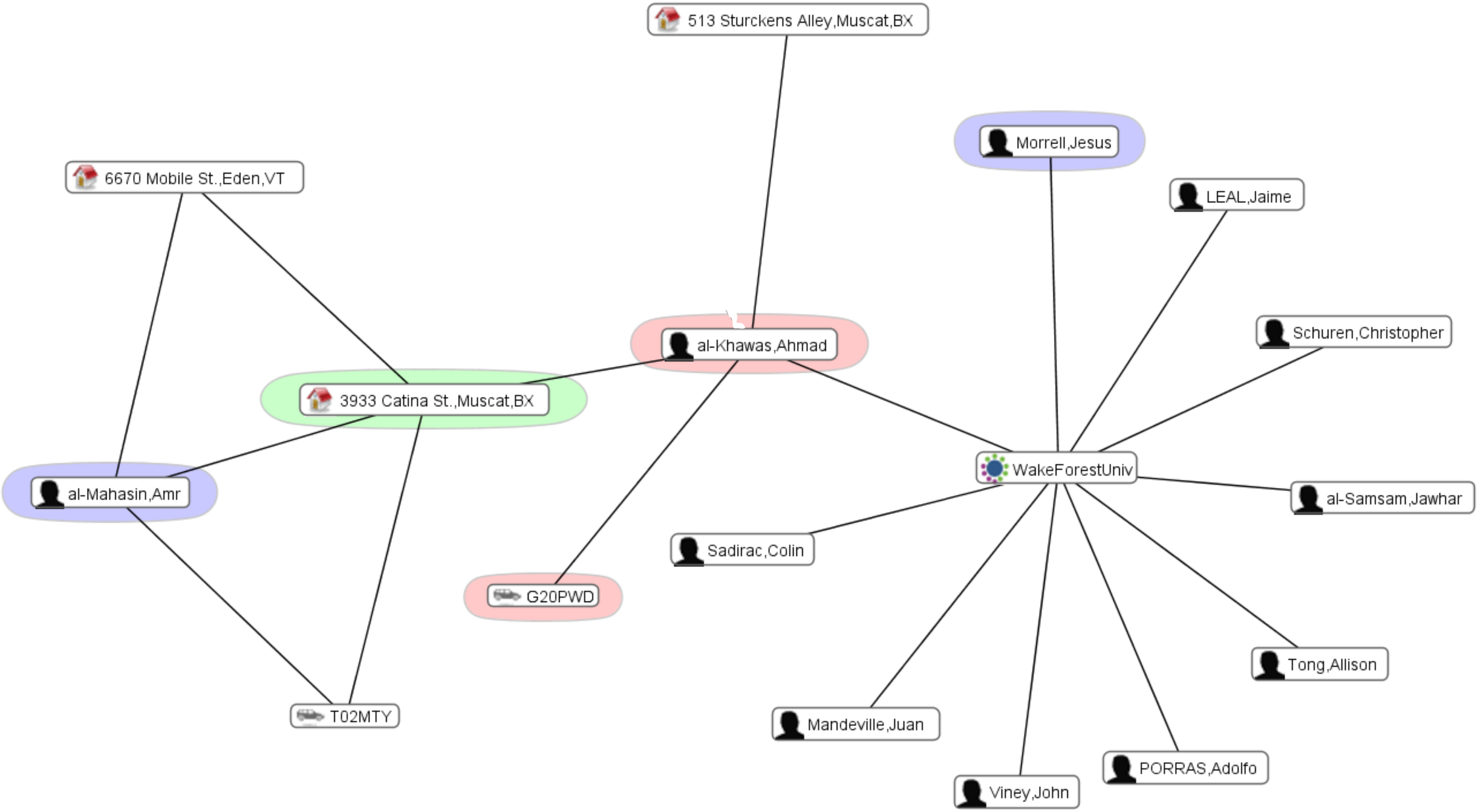
Initial graph view



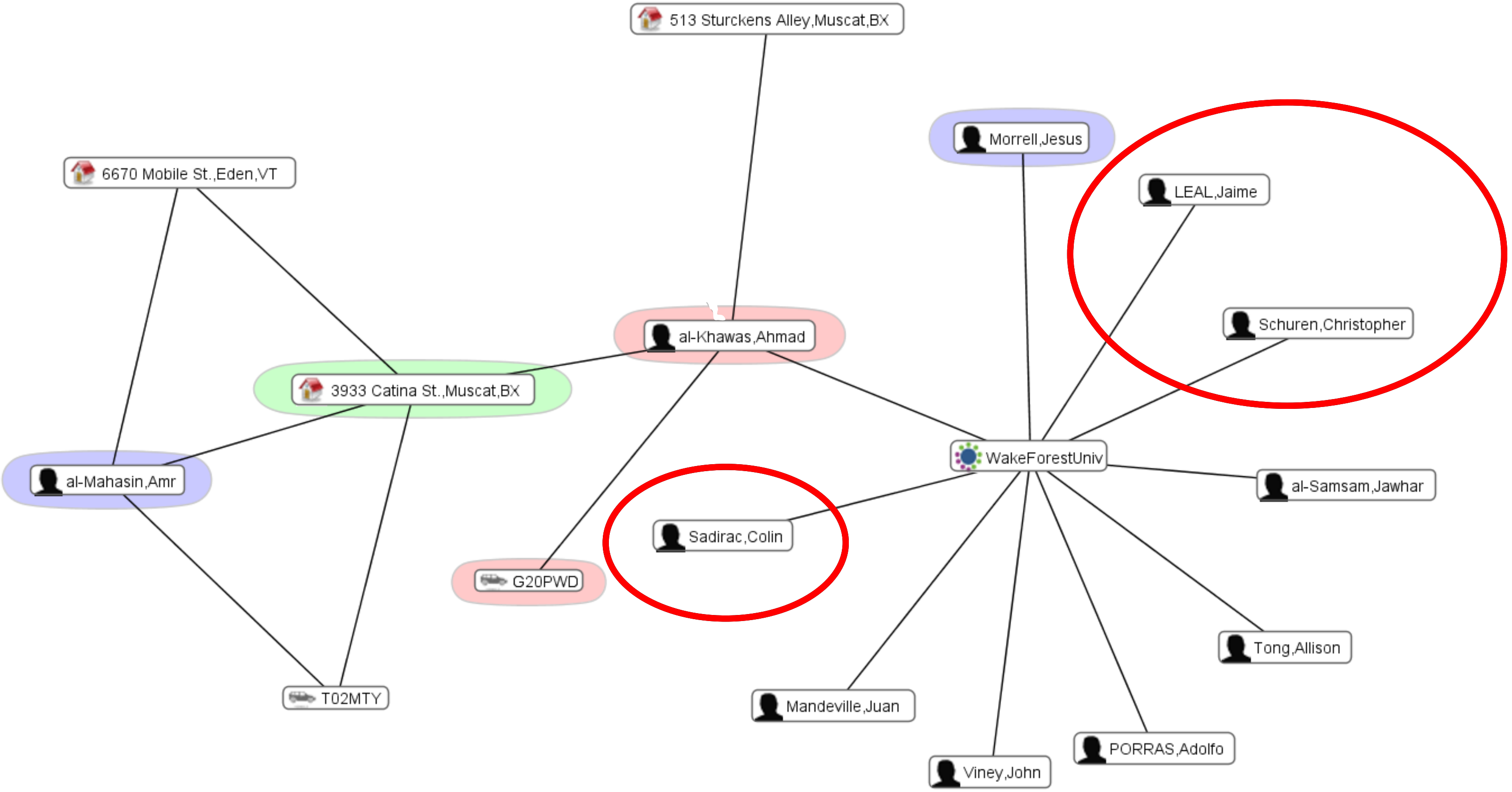
Select an entity



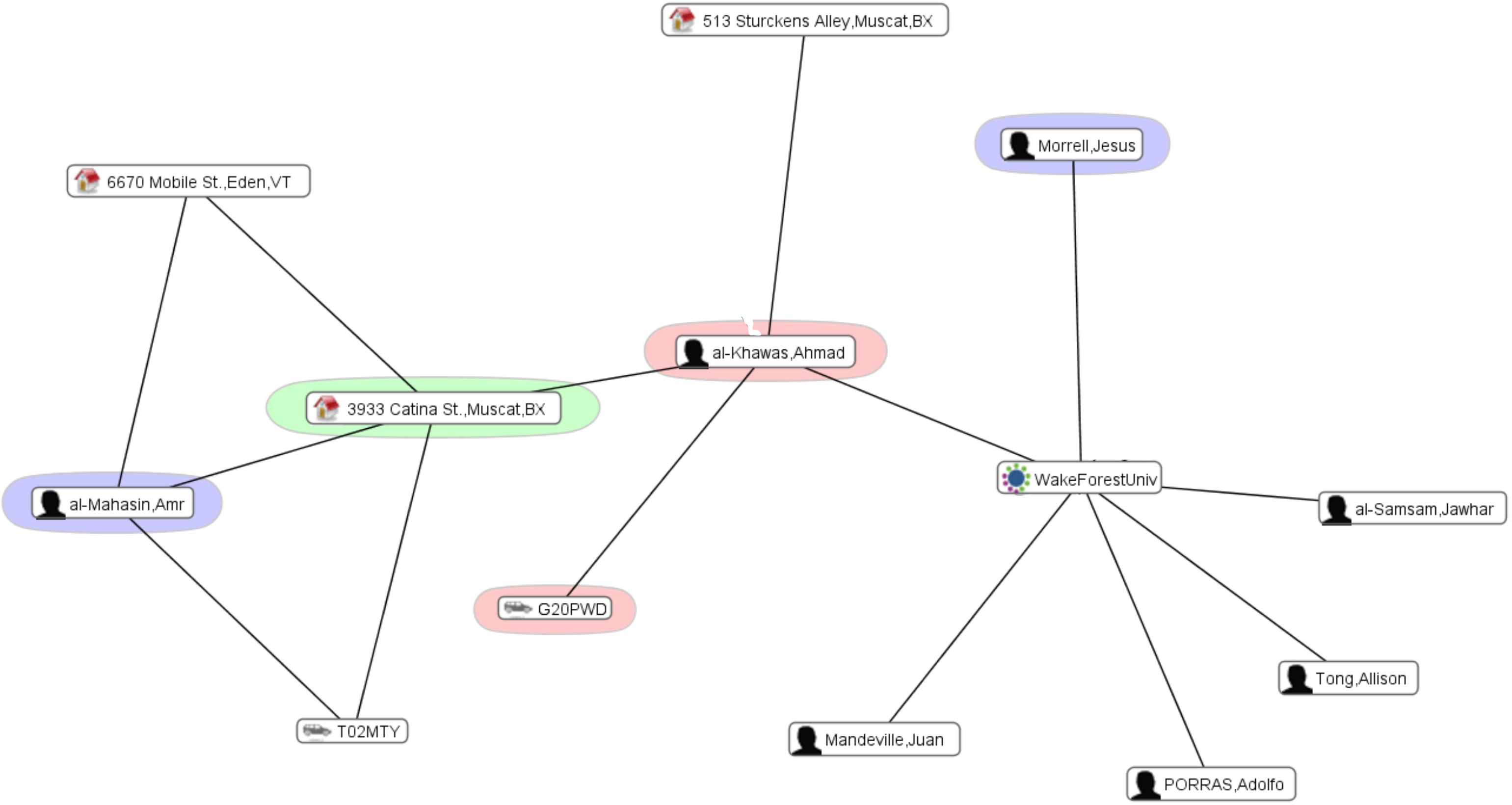
Expand one degree of separation



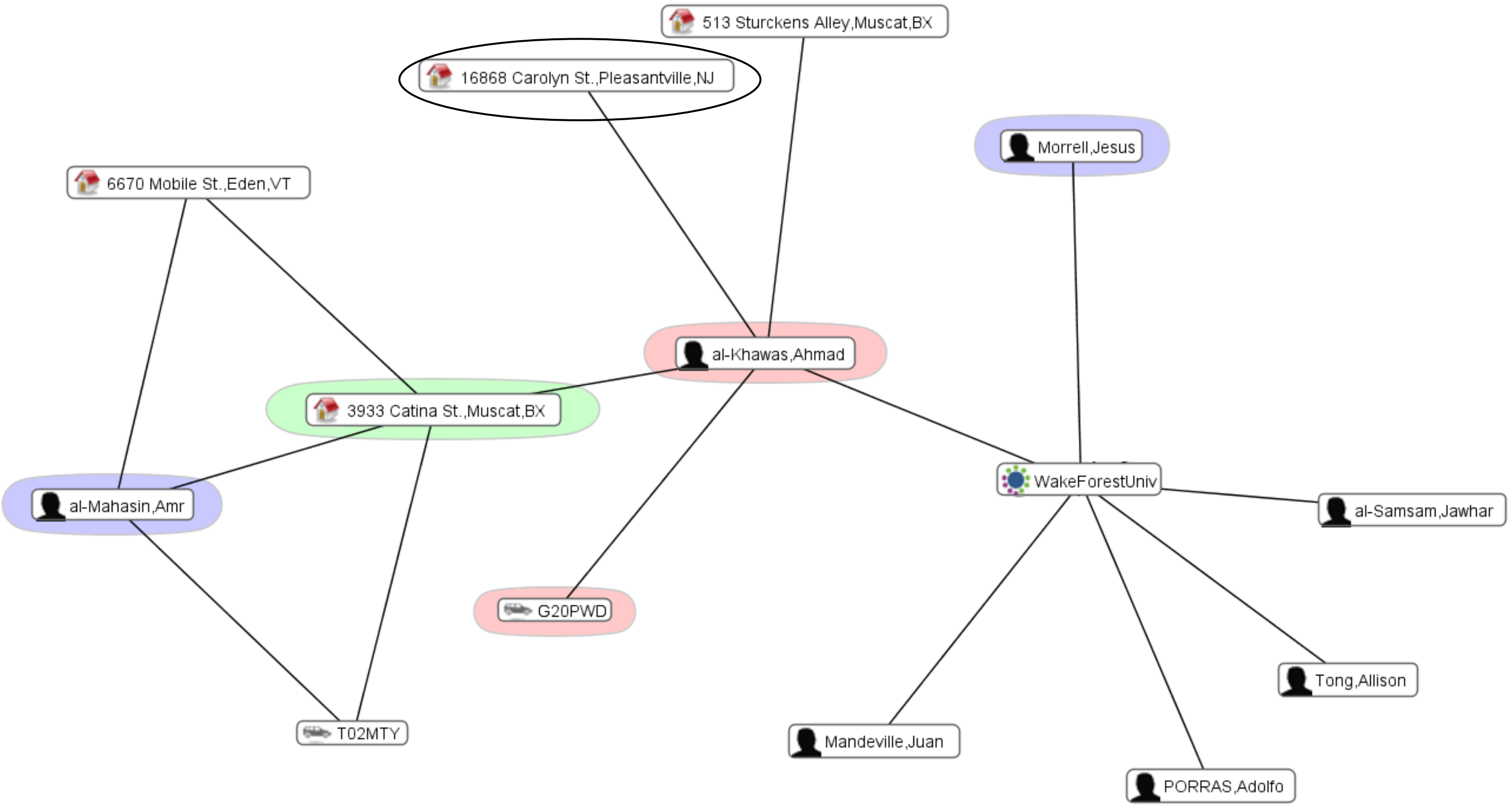
Manually remove entities



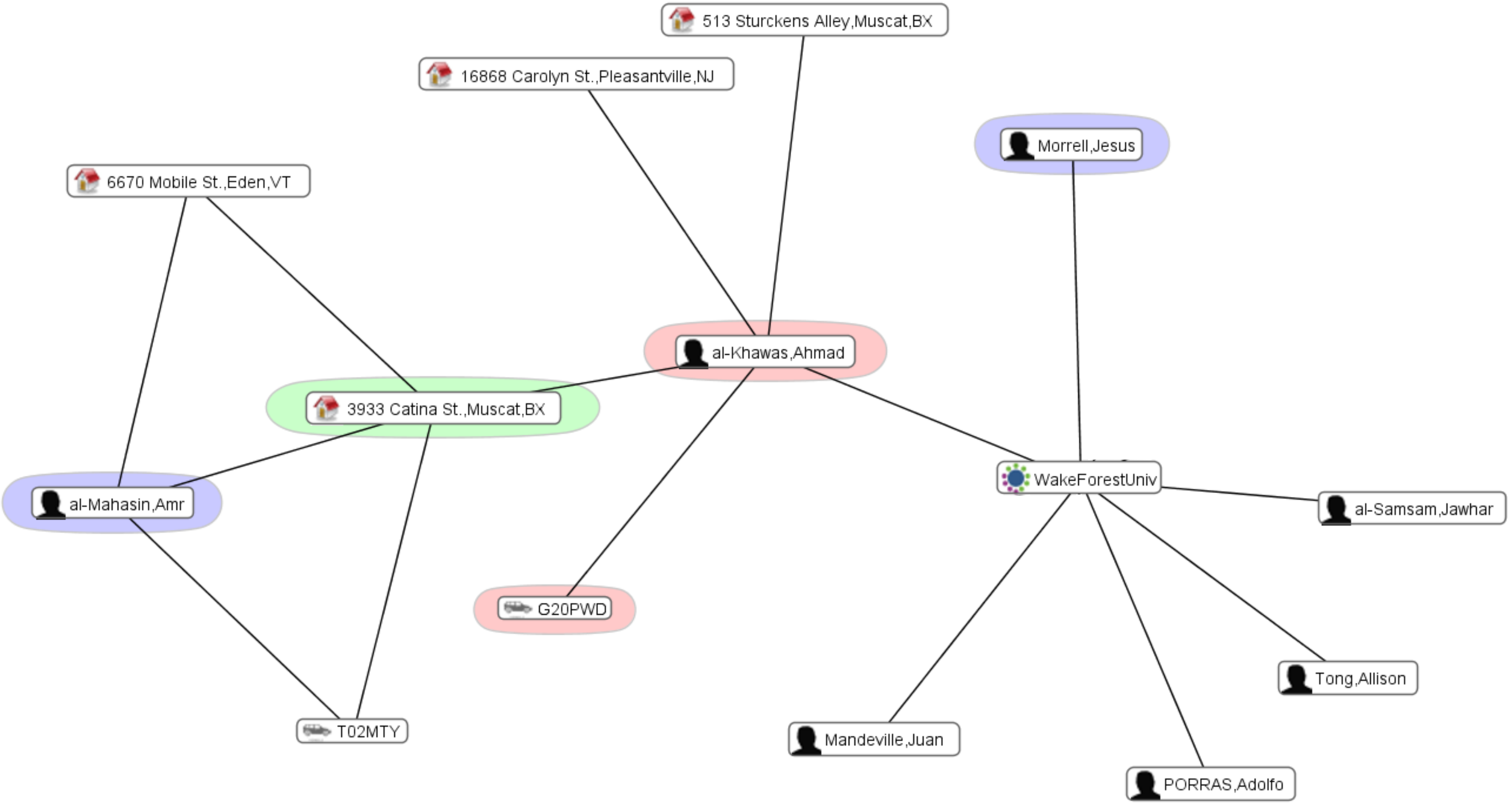
Retain graph and set subscriptions



Near real-time graph update



Graph update reviewed and retained



SmartData Fabric®-based Living Networks™

Living Networks™ is a combination of SmartData Fabric® Capabilities

- Federated data access
- Entity extraction
- Indexes, including Link Indexes™
- Fuzzy matching
- Link mapping
- Link analysis
- Real-time updates
- Alerts/notifications

Living Networks™ process (1 of 3)

- Start with a federated query to WhamTech system
- Return a result set of n rows conforming to a standard data model, ranked as needed
- Choose rows or a row
  - All, select/deselect individuals or just one
- If more than one row, then an automatic link analysis could be performed, seeking to find at least one connection between one or more rows
- Or, specify n degrees of separation expansion and any common entities will be automatically grouped together

Living Networks™ process (2 of 3)

- Display a link map/network
- Entities automatically extracted from rows, matched to neighboring entities, and match probabilities calculated
- Analyst interacts with display to expand further, remove entities and links, add links, change entities and other data, apply filters, change views, sequence displays, etc.
  - A representation of the steps taken to initially generate and modify the display, and changed data, can be stored for later retrieval and audit
  - Decisions can be made on changed data to (a) just retain the changes in a separate file, (b) update indexes, (c) update data sources with changed data or (d) combination of these

Living Networks™ process (3 of 3)

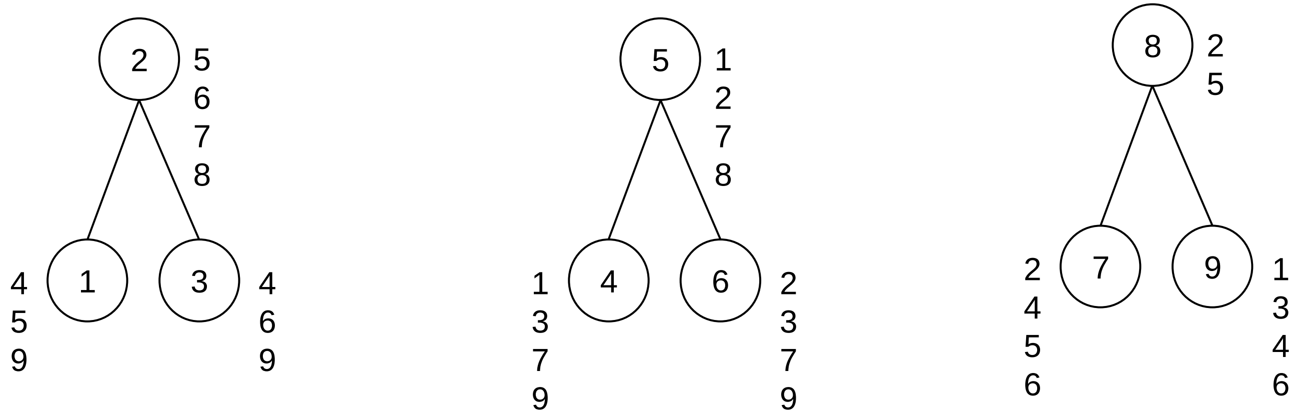
- Subscriptions set on real-time updates can specify thresholds on threat scores, probabilities/confidence levels, degrees of separation, etc.
- Highly visual and interactive front-end, similar to i2, Palantir and Visual Analytics – based on an OEM version of Keylines
  - Almost any third-party visualization application can be used

Link Indexes™ metadata

Determine number of links among data source tables

Assume internal PK-FK and self-join links in separate Link Indexes™ – only external links in separate Link Index

Table A | Table B | Table C



∑ links = 10 | ∑ links = 12 | ∑ links = 10

∑ Total links = 32
= 2n

AB + AC = 10 => AB = 10 – AC
BA + BC = 12 => AB = 12 - BC
CA + CB = 10 => AC = 10 - BC
AB = BA, CA = AC and CB = BC

10 – AC = 12 – BC => 10 – 10 + BC = 12 – BC
⇒BC = 6 => AB = 6 and AC = 4
⇒ n = AB + AC + AB = 16, which is correct