Scalable Automated Linking Technology

Edin Muharemagic, Ph.D., Sr. Architect, HPCC Systems

October 2015

LexisNexis®

This presentation was made using materials created by David Bayliss, HPCC Systems

Could you handle a Billion Dollar Query if I gave it to you?

- Cost of Supporting a Billion Dollar Query
  - Platform – little impact on a billion dollar revenue
    - Service Queries for our entire customer base on ~$500K worth of HW
    - HPCC Systems® available for free
  - Data Development – labor intensive, hence expensive
    - write a query, review new data sources for possible inclusion, write the data ingest routines, perform the data fusion, QA, capture, define and produce the business logic to perform analytics, regression test all of the above as the dataflow changes over the months and years
    - Got the right Big Data Resources – Data Scientists?
    - Big Queries Results Matter
    - Big Queries Results Hard to Get Right
- SALT to the rescue - Automate Data Integration Process to
  - Increase Productivity
  - Decrease Dependence on Data Scientists
  - Make high precision of 99.99+% realizable and defendable
- SALT and HPCC Systems power our $1.7B/year business

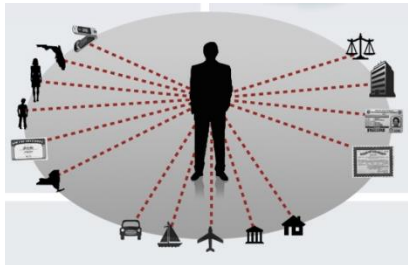
The Platform Has to Solve the Problem: Thor, Roxie, ECL, SALT, KEL...

Data

Public record and proprietary data on consumers and businesses

Linking (SALT)

Advanced technology which matches and links records across disparate data sources



Analytics: (KEL)

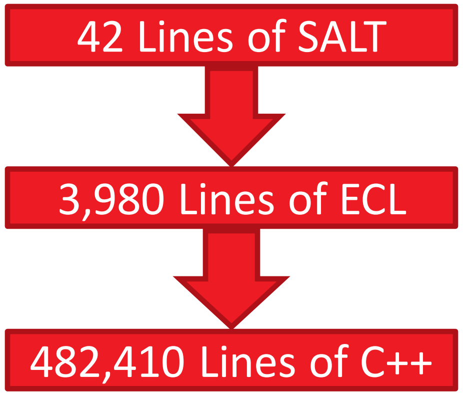
Knowledge Engineering Language for Graph Analytics

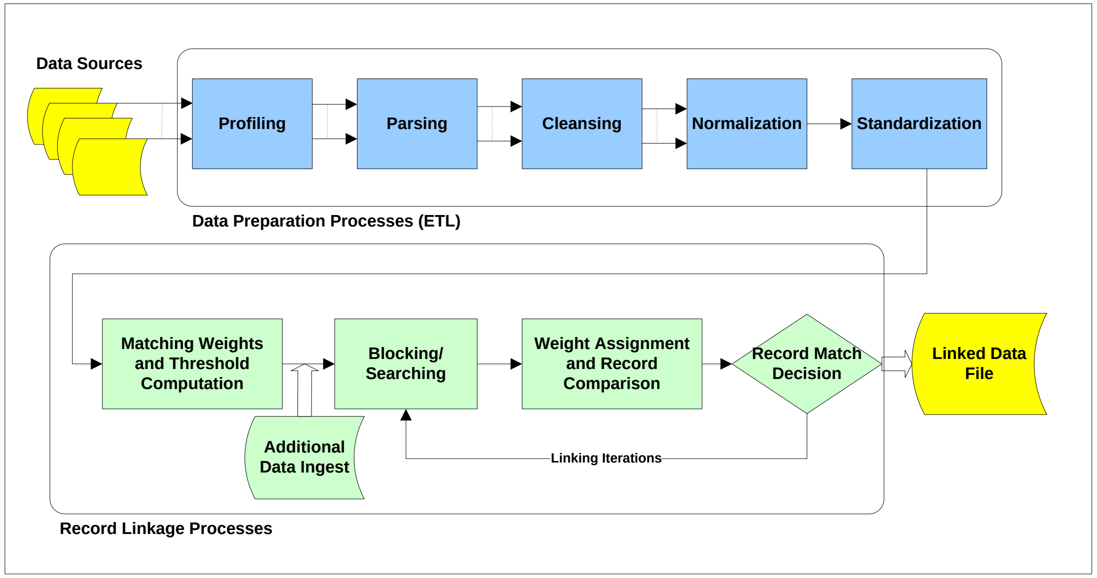
HPCC Systems (Open Source High-Performance Computing Cluster)

Data integration needs scalable processing power to allow for complex matching, scoring and clustering of Big Data

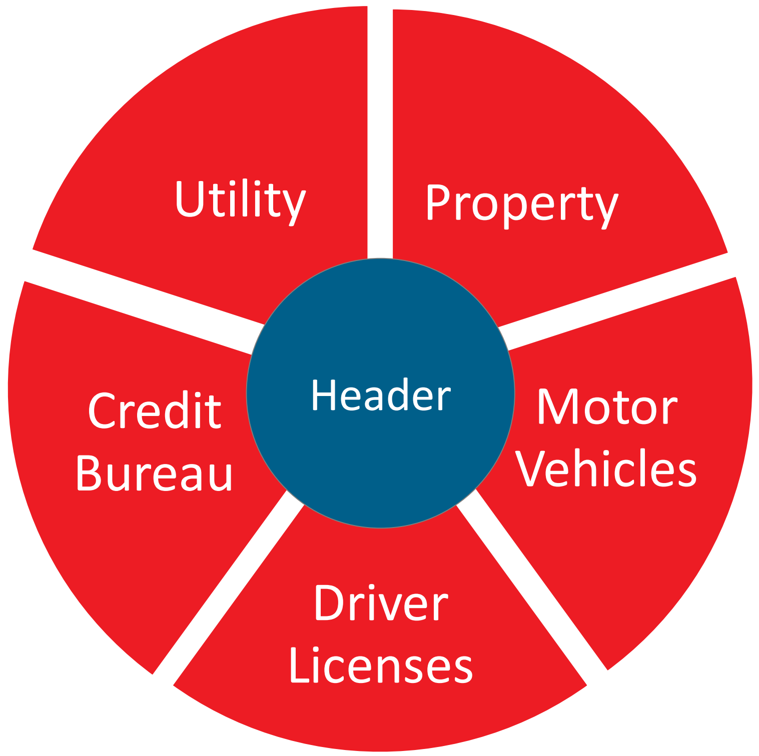
Beyond ECL: Scalable Automated Linking Technology (SALT)

- Templates based ECL code generator
- Provides for automated data profiling, parsing, cleansing, normalization and standardization
- Sophisticated specificity and relatives based linking and clustering
- Dramatically reduces development time, complexity and lines of code to implement data integration.
- Encapsulates a significant amount of ECL programming knowledge, experience, and best practices gained





The Hole in the Donut



Fill the hole to create the Header!

Person Header example:

- Contains 20B+ records
- Contains ~270M active entities
- Pulls data from 180+ sources
- Has data since 1985
- Ingests ~80M records per month

Filling The Hole – Internal Linking

Combining information from multiple heterogeneous data sources and matching records that refer to the same entity

Header

- Matching based on field specificity
- Fuzzy matching supported
- CONCEPT - when fields are related
- WORDBAG – when multiple words/tokens exist
- In cluster value propagation - filling the blanks on skinny records
- UNICODE – go international
- Plug-In fuzzy logic

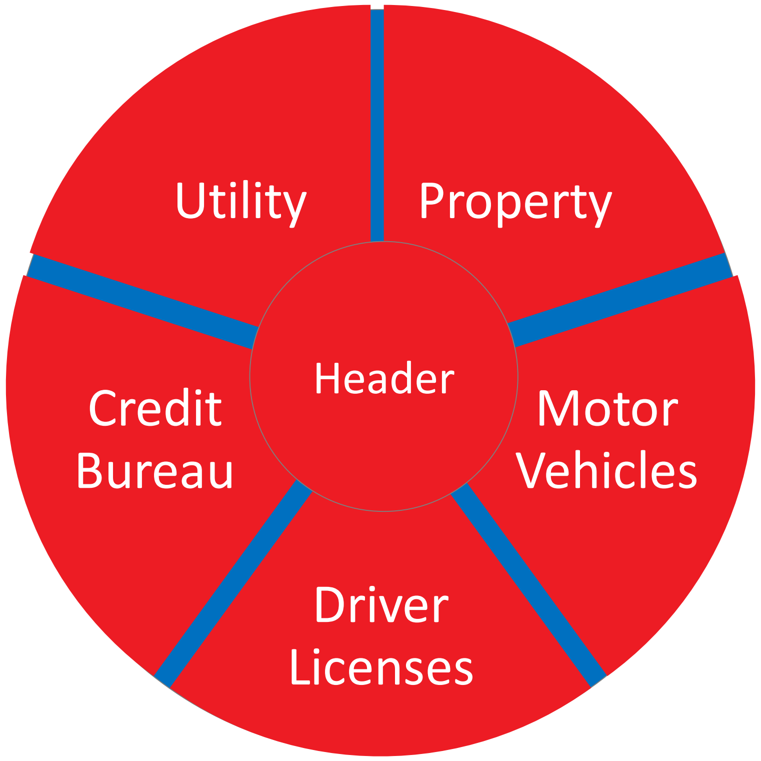
Assumptions

- Individual Fields are Independent
- All fields in the file represent the entity that the record represents

Linking Challenges

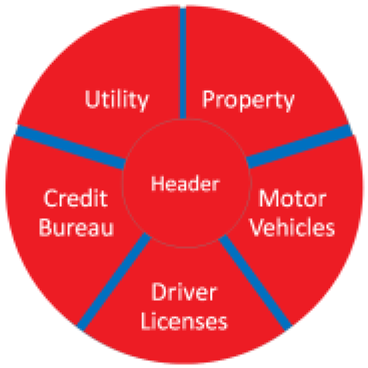
- Assumptions not always correct: Jose Wang
- Similar twin names: Jayla vs Kayla, Jada vs Jaden
- Sons given father's name

The Pieces Fall Apart



Binding The Pieces Together – External Linking (a.k.a. Entity Resolution)

Assign a unique entity identifier to Property, Utility, … records



Requirements

- Header linked with high precision

Back End – LINKPATH driven

- Multiple keys built for consistent and predictable results
- Low level Roxie optimizations abstracted
- UBERKEY – the key which always matches
- LINKPATH suggestions provided
- Optional – enforce what you wish to enforce
- Extra Credit – and weight what you wish to allow

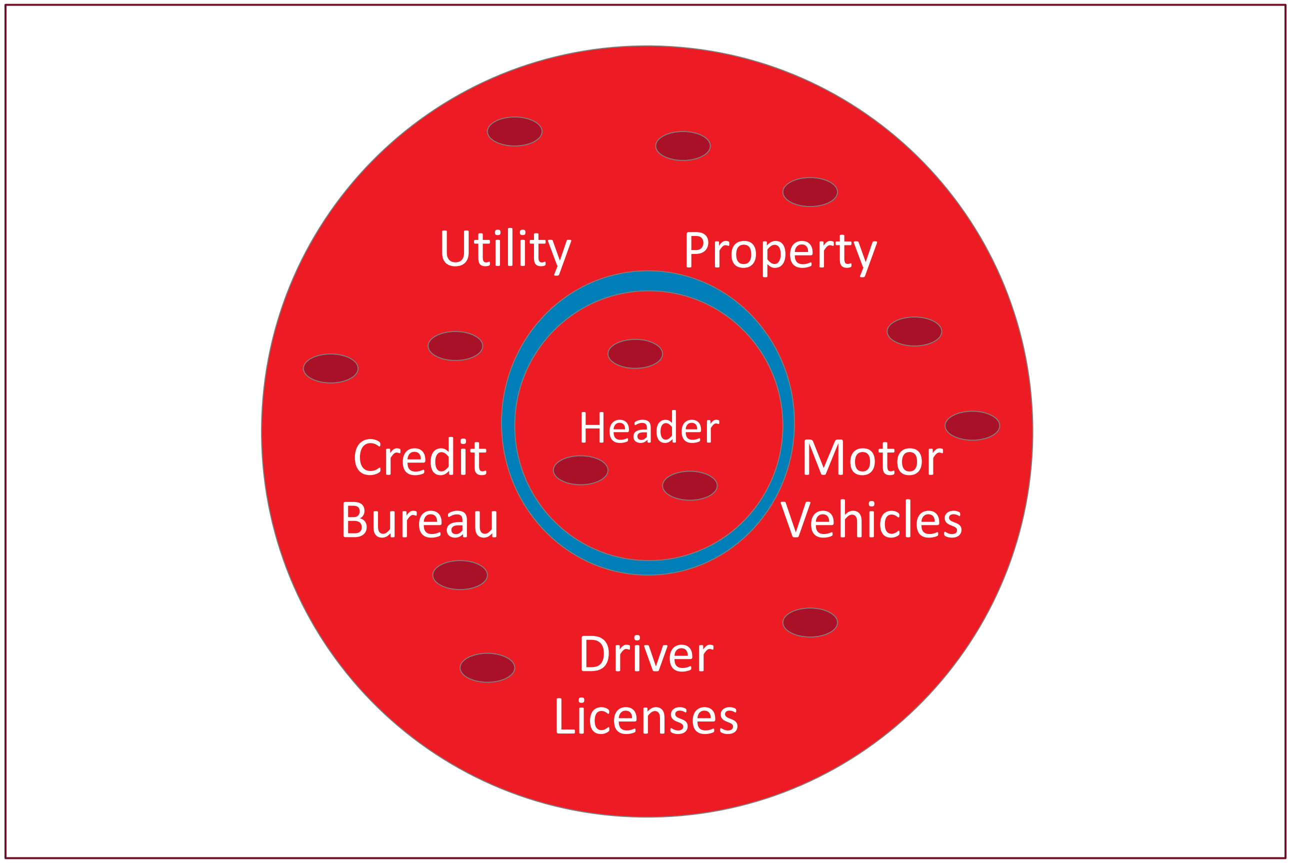
Generated Code

- Online, Thor to Online, Keyed Thor and Big Thor execution models
- Comprehensive statistics on keys tried/failed
- Statistics on composition of incoming batch file (help with LINKPATH)
- Results detailing how and why certain matches were allocated

Front End Capabilities

- Word Wheels
- MULTIPLE supported at field level
- LATLONG capability for smooth radius searching
- RANGE, WILDCARD,…

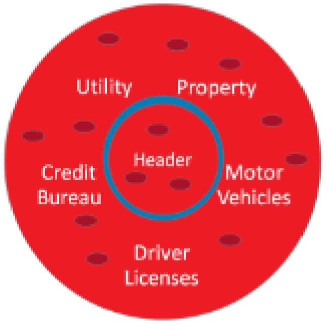
What if bad data comes in? The Flies in the Donut Batter



Data Integrity – The FieldType

Formalizing our Data Model

- Lexical definition of valid fields
- Definitions are declarative
- Can automatically ‘force-clean’, blank or reject
- FieldTypes have an inheritance system; like that but also …



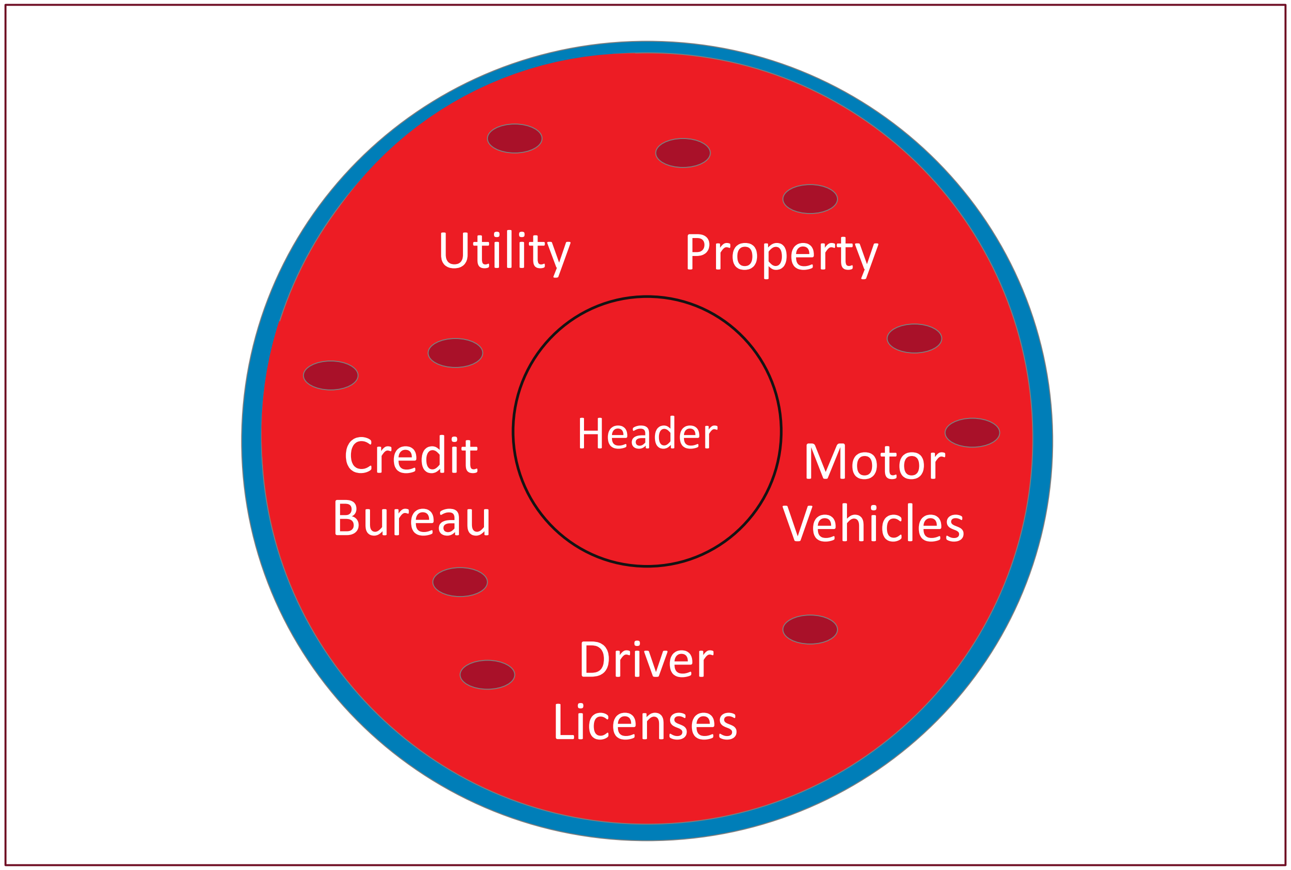
```
FIELDTYPE:DEFAULT:LEFTTRIM:NOQUOTES("'):
FIELDTYPE:WORDBAG:CAPS:ALLOW(ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'):SPACES( <>{}[]-
^=!+&,./):ONFAIL(CLEAN):

FIELD:company_name:BAGOFWORDS:LIKE(WORDBAG):TYPE(STRING120):INITIAL:ABBR:EDIT1:
```

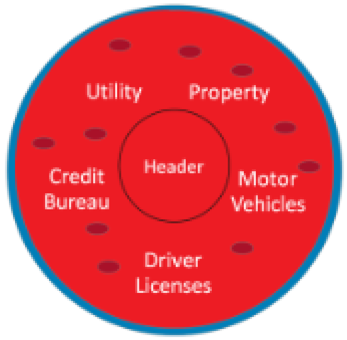
Capabilities

- profile data sample and generate FIELDTYPEs automatically; use those FIELDTYPEs to check that all the data coming in fits that sample; Flag the changes
- generate FIELDTYPEs to match the most common 99.9% of data; identify the 0.1% of data that is unlike all the rest; float bad data to the top
- Checking code can be applied to ANY process
- Can break down results for a header file by source
- Automated anomaly detection for multi-source files (primitive semantic analysis)

Finding the errors before they happen ….



Profiling, Scrubs & Hygiene System



Formalizing Best Practice

- Easily the single most popular SALT feature
- Field by Field breakdown of lexical structure of data
- Finds commonest and the least common things in file
- Can generate optimized ECL layout from data structure
- Produce FIELDTYPES

Profiling Capabilities

- Individual data field profile report
- Summary profile report
- Correlation report for each field in a record – useful for defining CONTEXT and CONCEPT
- Field combination analysis report – top combinations of non-blank fields sorted by frequency (LINKPATH definitions)

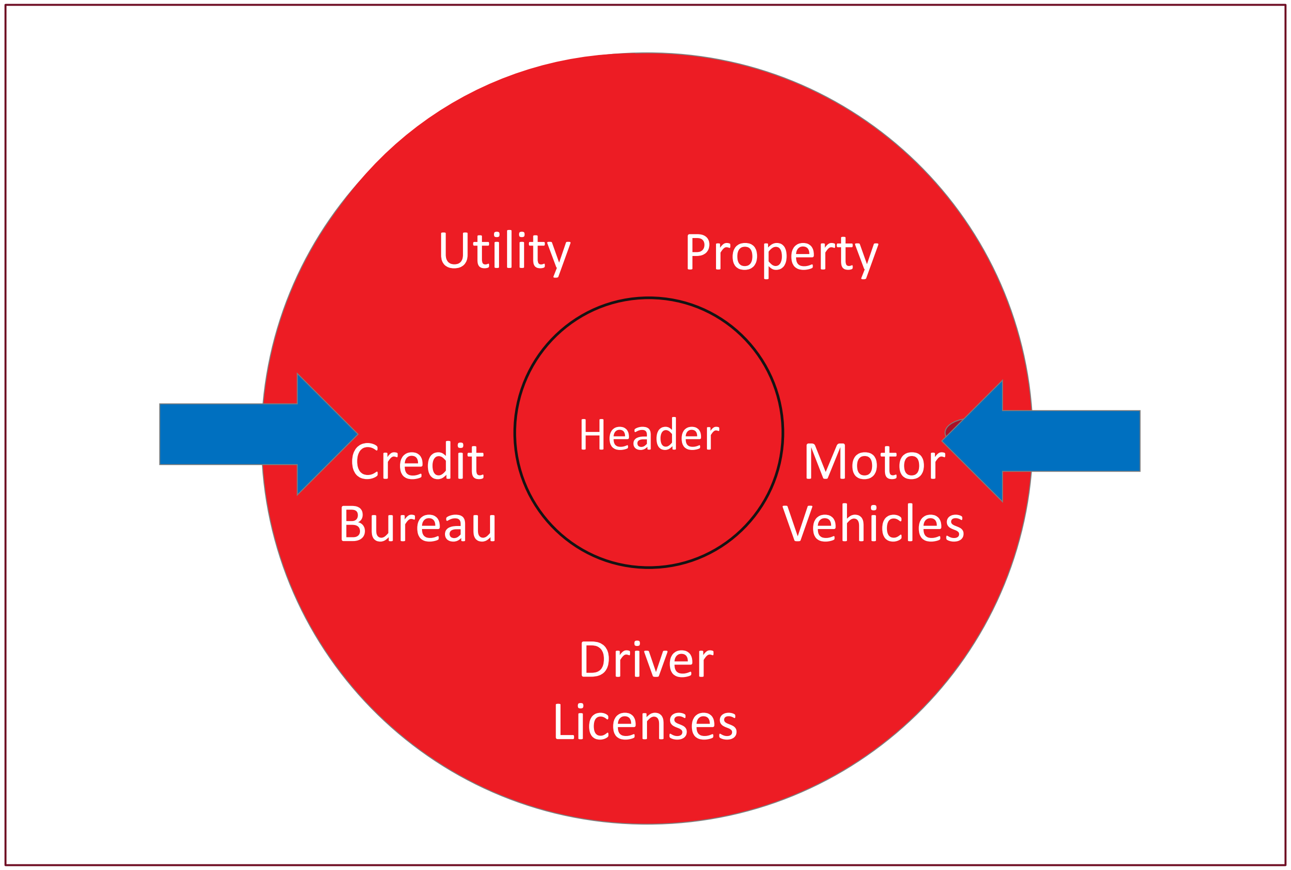
Scrubs Capabilities

- Helps track quality of source files
- Appends a bitmap to the file showing which fields and records are valid (or invalid)

Hygiene Capabilities

- Helps clean the field values based on FIELDTYPEs

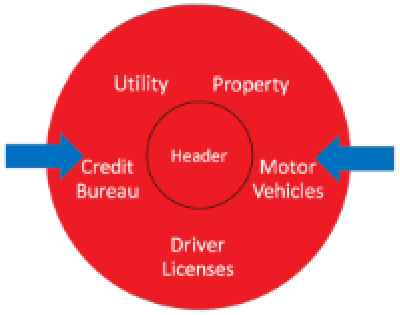
How do you keep your data current? Getting the Data In ….



Ingest System

Formalizing Best Practice

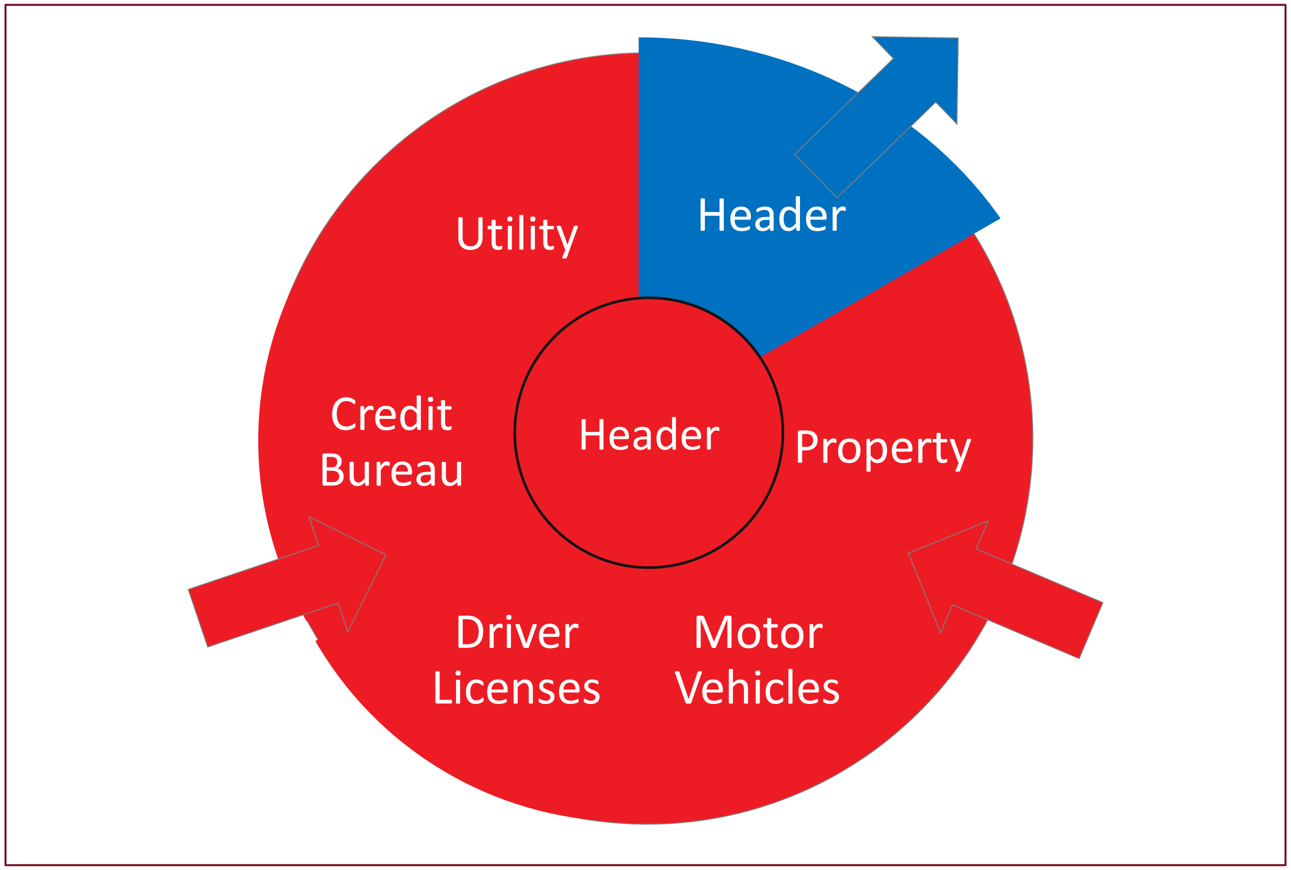
- Can define the basefiles going into the header
- Can define the source files going into a base file
- Automatically handles RID initialization
- Handles 'deduping' in the case of repeated data dumps



Capabilities

- Automatically handles date first/last seen and vendor variants
- Applies FIELDTYPE constraints (as defined) during ingest
- Distinguishes 'fields that need a new record' from 'fields that can update'

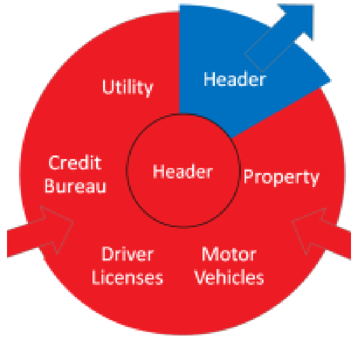
Getting the Header Data Out ….



Header Access System

I know this thing about the fields that are in your data – give me back the answers!

- Relevancy ranking of entities or records (or both)
- Field by field match level scoring
- Consistent with (and uses same keys) as External Linking
- Search and filter on any field (UBERKEY)



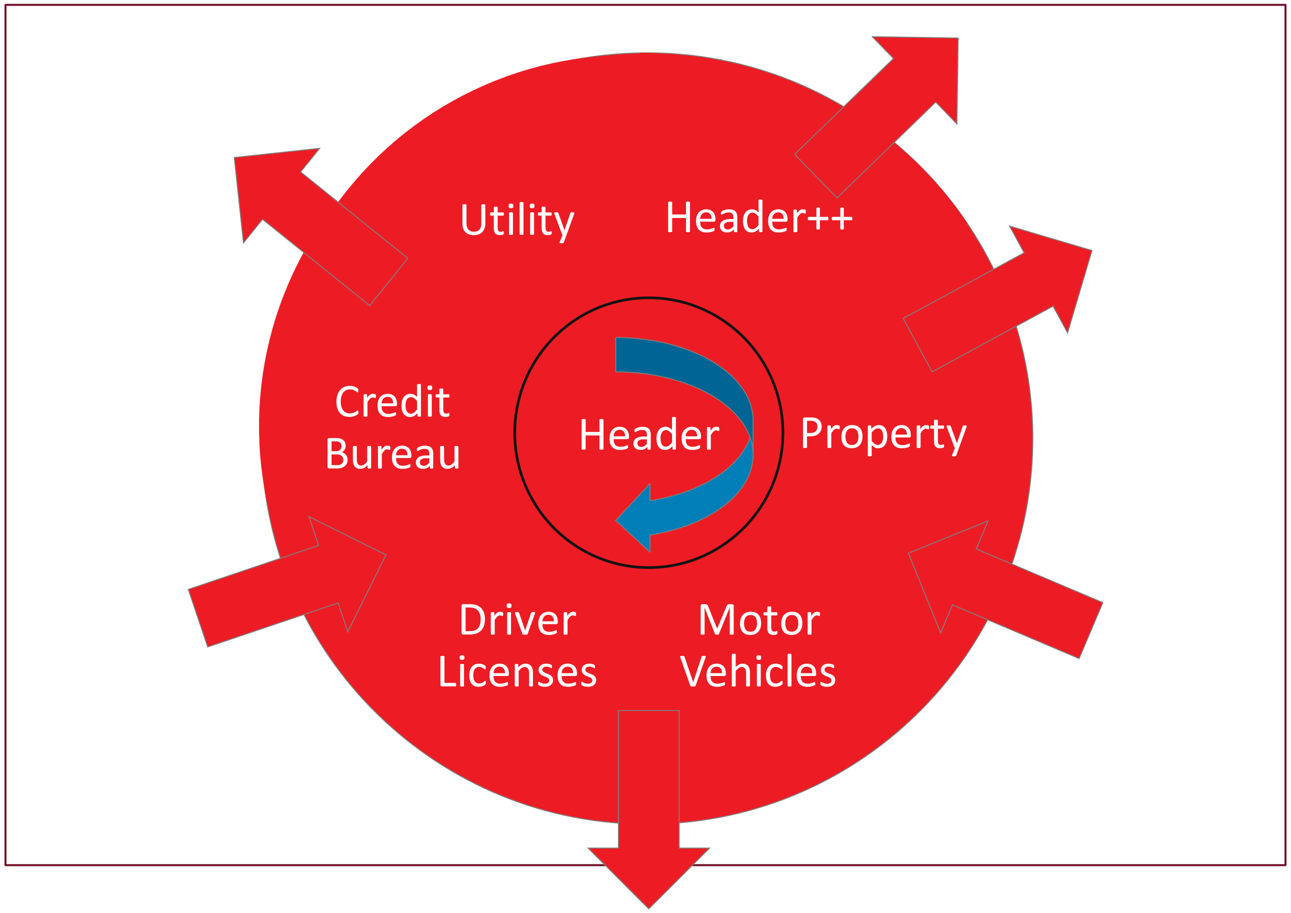
Best Capability

- Over 300 variations upon what ‘best’ might mean; configurable field by field
- Produces best for any/all data restrictions in a single pass of the data
- ‘Real’ field values can be flagged for ‘bestness’
- Reports available showing which sources are best for which fields

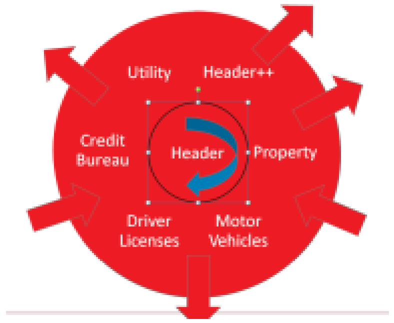
Scored Search – When I do not know what I am looking for…

- Execute a filtered and scored linear regression model on the fly
- Find the entities that best matches **this model**
  - Give me the cities with the oldest average age
  - Give me the cities with 'wealthiest, oldest people‘, and attach a certain amount of weight to each factor
- Uses Roxie in memory capability
- User extensible

Non-Obvious Relationships



Relatives And Associates



Does statistically significant link exist between entities?

- Mathematical model for 'how likely is this'
- Can be used to:
  - Identify Relatives and Associates
    - People share address and last name
    - People share more than one address
  - Find 'oddities' - complex graph analytics
    - Find people with the same name, and the same address who appear in different states
  - Feed the results back into Linking
    - Relationship based linking: A r B ➔ A = B ?
    - Co-Relationship linking: A r B, A r C ➔ B = C ?

```
RELATIONSHIP:relationshipname:BASIS(FieldList):DEDUP(FieldList)
[:SCORE(FieldList)][:TRACK(FieldList)][:MULTIPLE(n)][:SPLIT(n)]
[:THRESHOLD(n)][:BLOCKTHRESHOLD(n)]
```

In Summary

- Data integration is a complex process which requires scalable platform, and proper tools
- HPCC Systems provides scalable processing power to allow for complex matching, scoring and clustering of Big Data
- SALT automates data integration process, encapsulates a significant amount of ECL programming knowledge, experience, and best practices gained, and dramatically reduces development time

Example of Productive Data Integration using SALT: Develop new People Header Search

- 1.5B records from dozen of sources
- 10-15 internal linking iterations to link from scratch
- 12-24 hours / iteration
- Keys built and External Linking completed
- Compared to the old system, this SALT based version is:
  - Done quickly and efficiently – 18 person months
  - Twice as fast as the old one
  - Better quality (both in recall and precision)
  - More maintainable

For more information, visit http://hpccsystems.com/salt