Large-scale entity extraction and probabilistic record linkage in LexisNexis Risk Solutions

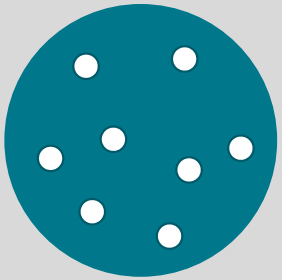
HPCC SYSTEMS®

LexisNexis® RISK SOLUTIONS

RELX Group

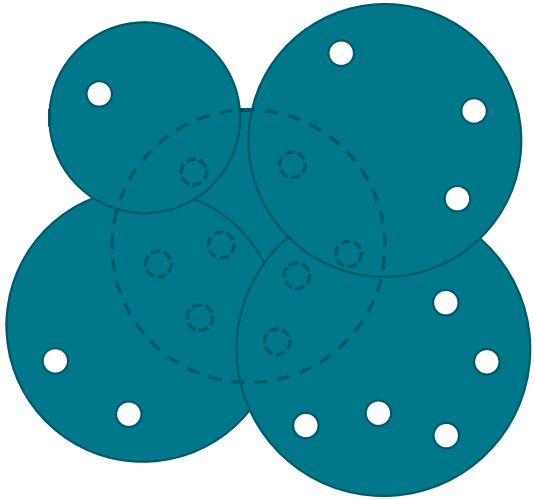
The Data Centric Approach

A single source of data is insufficient to overcome inaccuracies in the data



The holes represent inaccuracies found in the data.

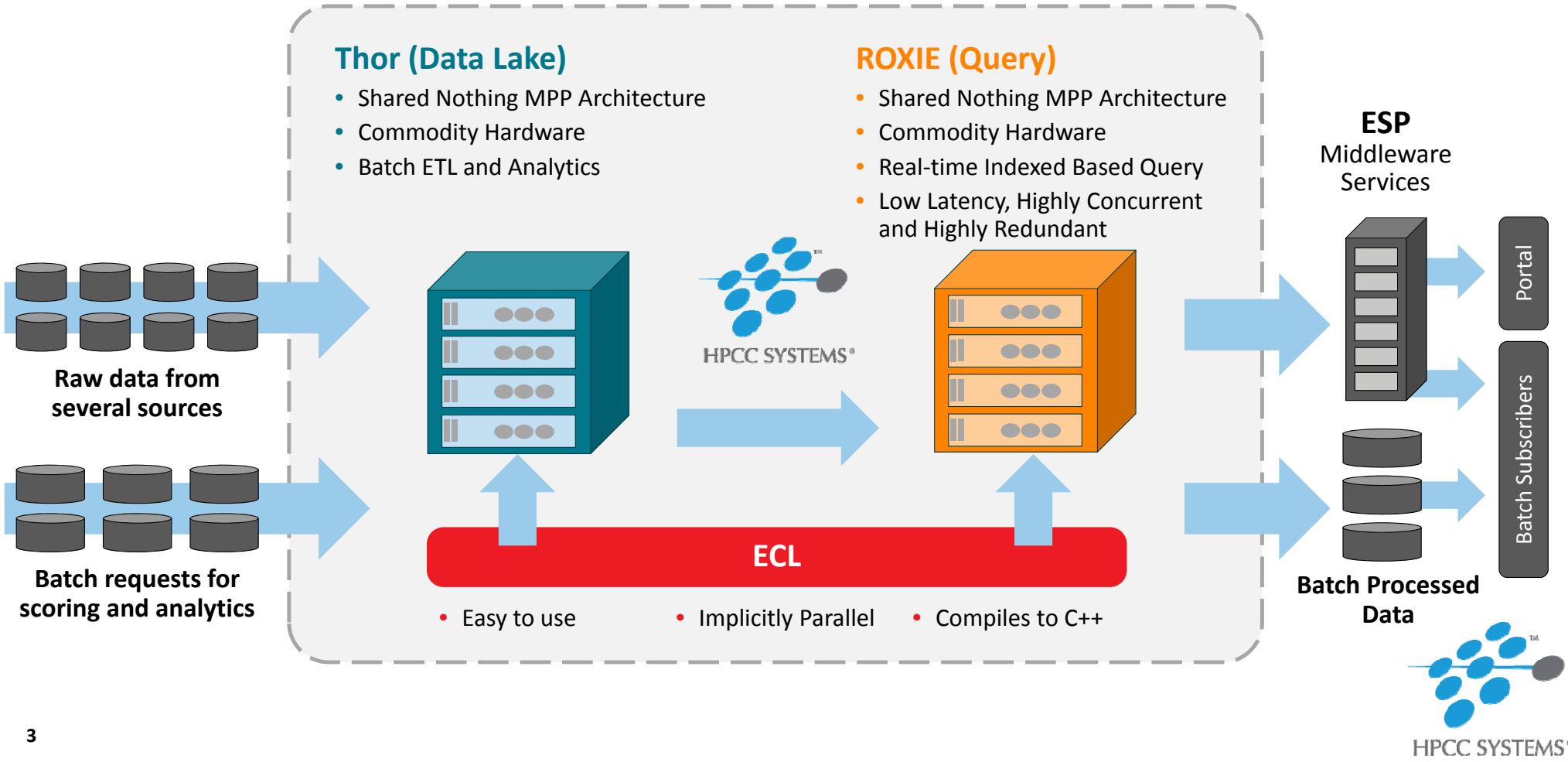
Our data services rely on our ability to integrate data from **multiple data sources** and to transform it into a **highly intelligent social network graph** that can be queried to identify non-obvious relationships.



The holes in the core data have been eliminated.

HPCC SYSTEMS®

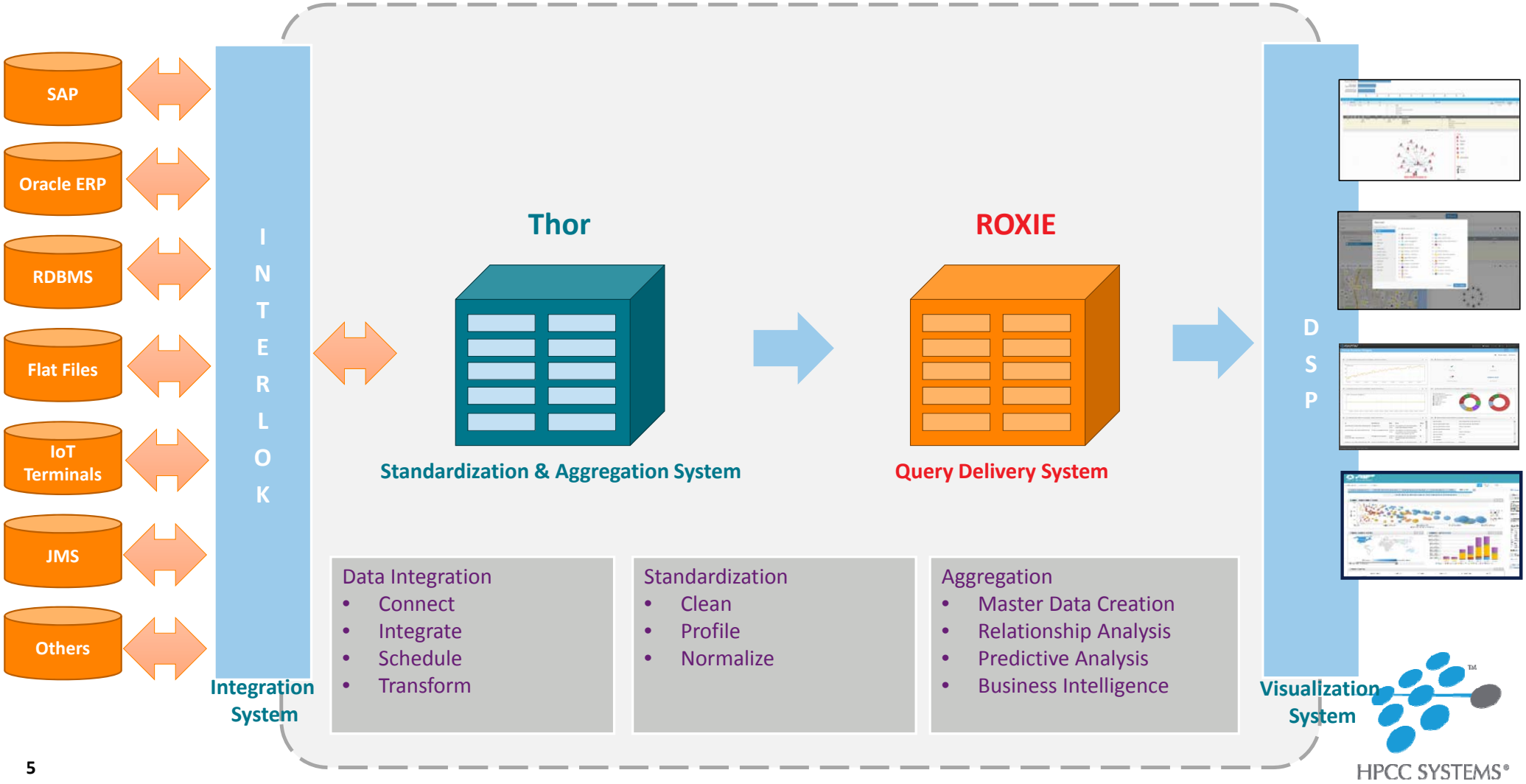
Data Flow Oriented Big Data Platform



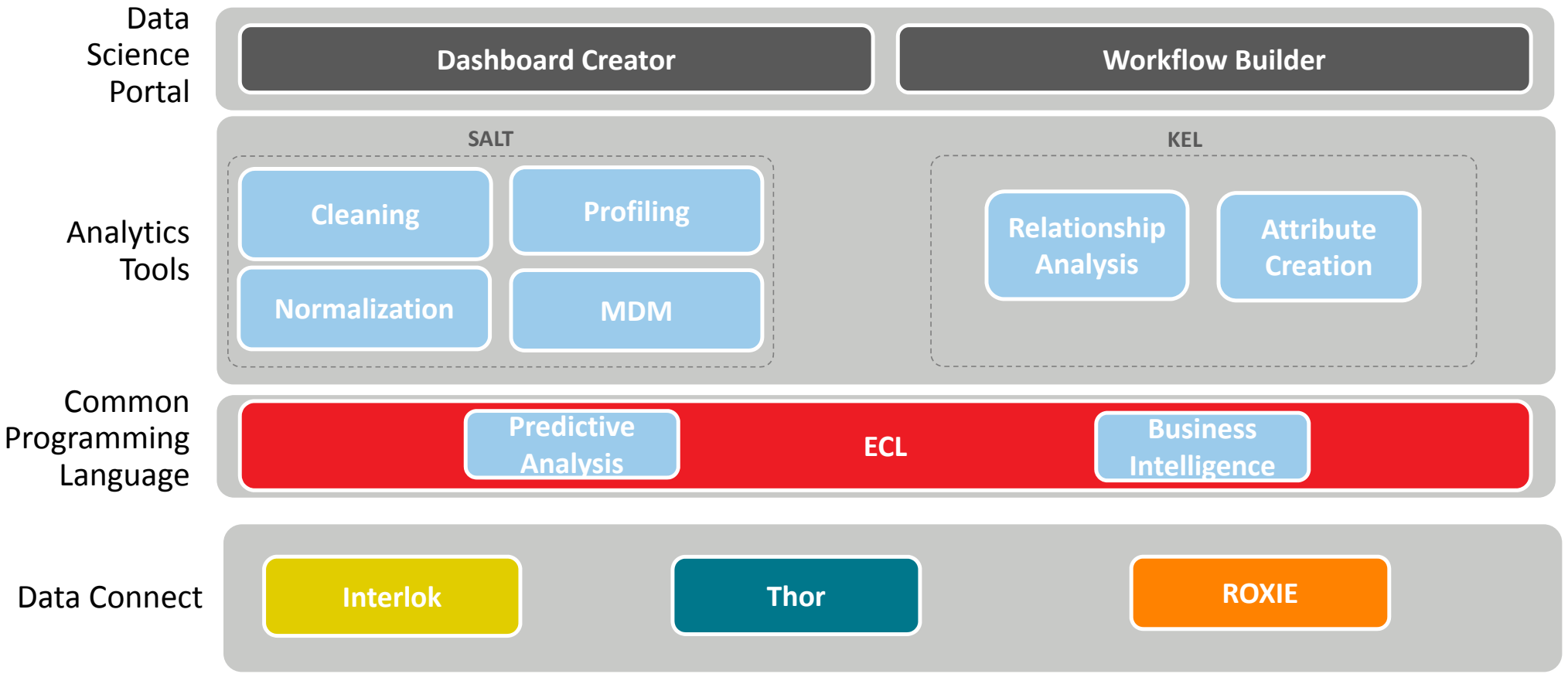
The STRIKE stack

SALT	• Large scale data integration • Probabilistic linking • Entity disambiguation and resolution
Thor	• Batch oriented Big Data processing and analytical engine • Machine learning supervised model training • Clustering
Roxie	• Horizontally scalable and fault tolerant real-time disk-based retrieval and analytics • Horizontally scalable in-memory analytics
Interlok	• Seamless data integration with hundreds of data stores • Real-time data ingest • Flexible stream processing
KEL	• Graph/Network data models • Complex queries based on n-degree relationships and attributes • Highly efficient
ECL	• Dataflow oriented declarative data programming language • Compiles to C++ for optimal performance • High expressivity and conciseness

STRIKE Technology **Component** View



STRIKE Technology **Layer** View



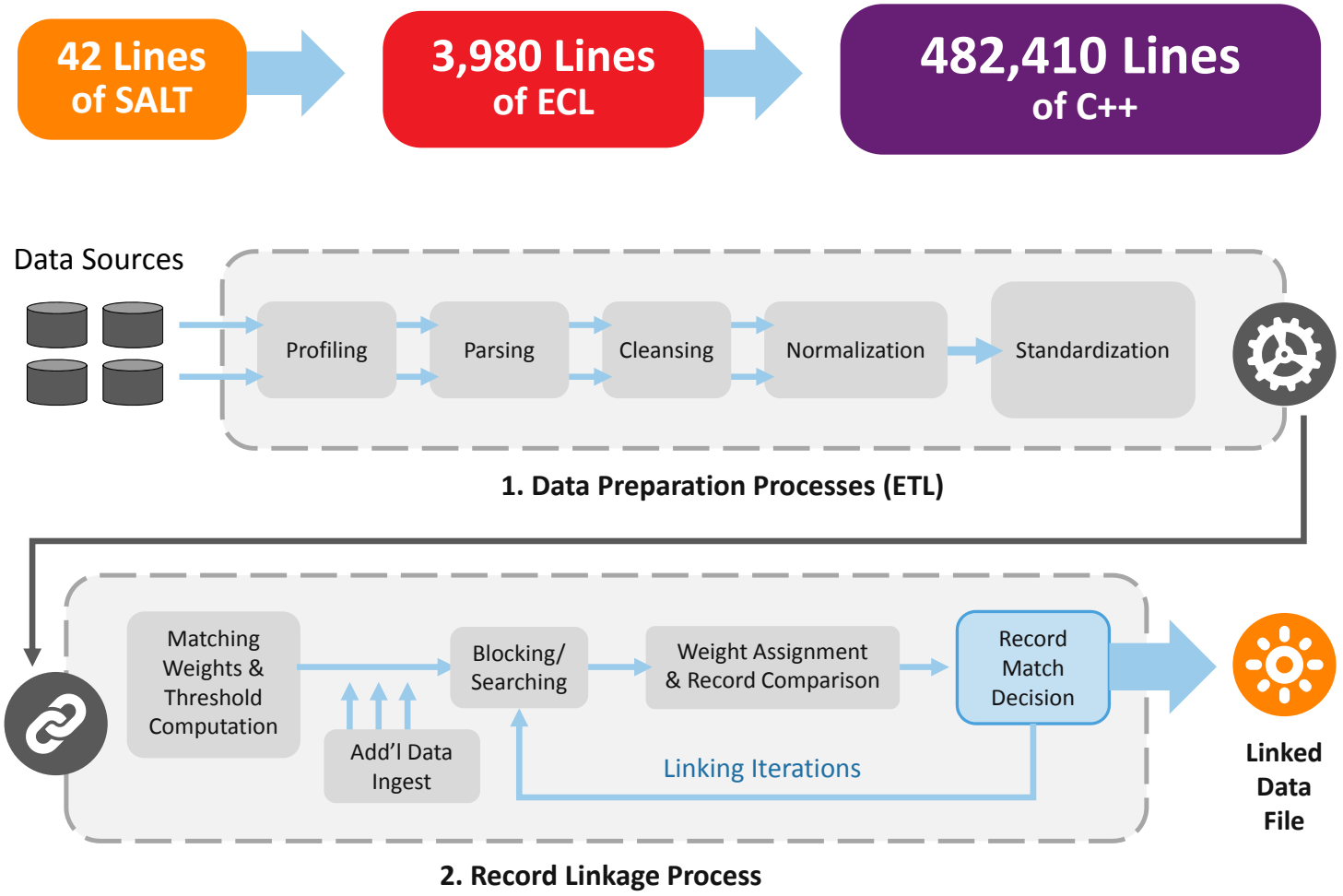
Master Data Management with SALT

From disparate data, to clustering, to showing relationships

HPCC SYSTEMS®

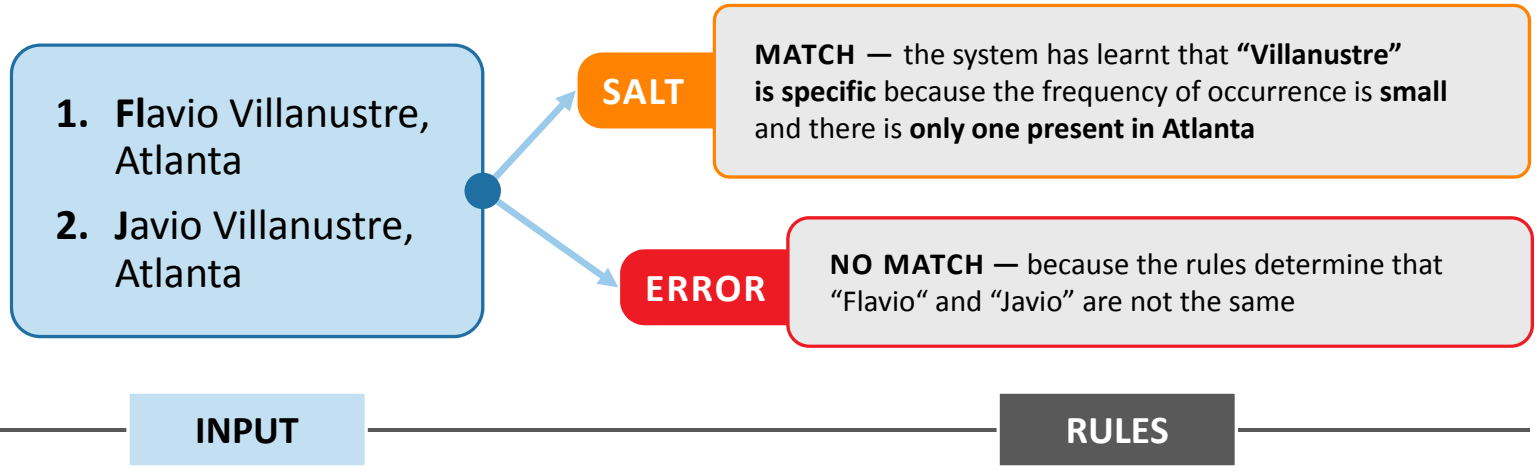
SALT Data Integration

- The acronym stands for "Scalable Automated Linking Technology"
- Entity disambiguation using Inference Techniques
- Templates based ECL code generator
- Provides for automated data profiling, parsing, cleansing, normalization and standardization
- Sophisticated specificity and relatives based linking and clustering

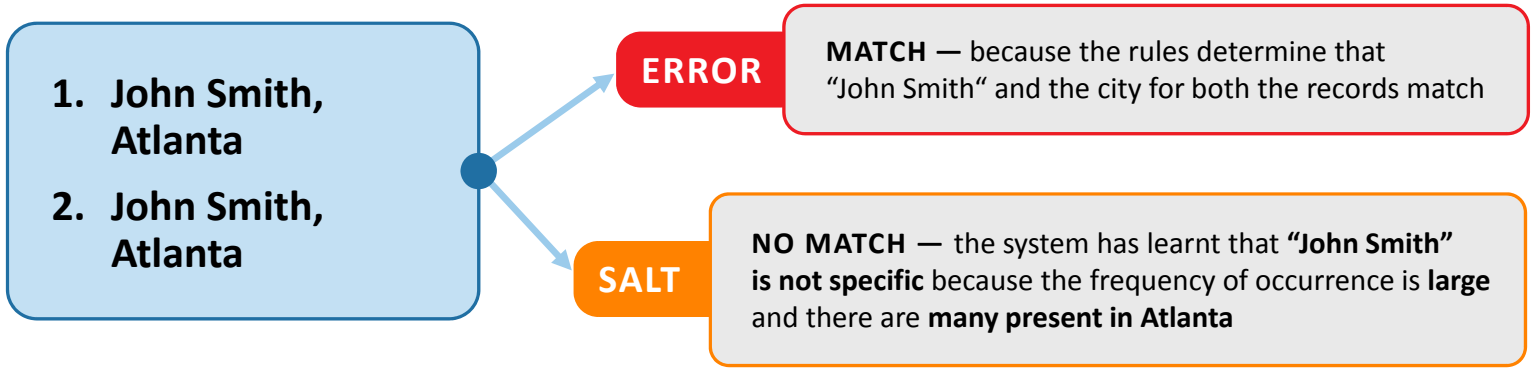


SALT's Superior Linking Technology

SALT eliminates **FALSE NEGATIVES** using probabilistic learning



SALT eliminates **FALSE POSITIVES** using probabilistic learning



Relationship Analysis With KEL

KEL — an abstraction for network/graph processing

- Declarative model: describe what things are, rather than how to execute
- High level: vertices and edges are first class citizens
- A single model to describe graphs and queries
- Leverages Thor for heavy lifting and ROXIE for real-time analytics
- Compiles into ECL (and ECL compiles into C++, which compiles into assembler)

Architectural/scalability/performance considerations

- I/O performance is typically the limiting factor
- Different I/O performance profiles between Thor and Roxie
  - Thor resorts to mostly sequential full data scans – heavy read and write
  - Roxie's data access is inherently index-based and random – read-only
- Significant trends in hardware evolution in the last decade
  - CPU single thread performance hasn't increased
  - More CPU parallelism (after all we don't want to disappoint Mr. Moore)
  - Performance and density in spinning drives hasn't improved that much
  - Advent of power efficient, dense and fast NVRAM (SSD and NVME)
  - Random access performance of NVRAM gives it an edge over spinning drives
  - Cost of NVRAM has dropped significantly
  - Durability of NVRAM is comparable to spinning drives for most loads

Evolution

- Moved from spinning drives to SSD's in Roxie
  - Reduced transaction times by 50%
  - Tripled transactional capacity
  - Maintained total storage and physical footprint
  - Reduced power consumption
  - Provided encryption at rest with no additional CPU utilization
- Moved from spinning drives to SSD's in Thor
  - Reduced the impact of multi-threading and random I/O on data stores
  - Increased overall real-world performance
  - Maintained total storage and reduced physical footprint
  - Reduced power consumption
  - Provided encryption at rest with low CPU utilization impact
- Increased network bandwidth
- Increased internal parallelism

HPCC SYSTEMS®

Opportunities

- Reduce/speed up spills (Thor)
  - spinning drives -> SSD -> NVME -> more RAM)
- Push computation closer to the data
  - pre-filtering on SSD
  - in-memory computation (Automata, Hybrid Memory Cube, EMU)
  - dedicated co-processors (Xeon Phi, GPU, FPGA)
- Faster memory interfaces
  - NVDIMM/NVME – Tiered memory interfaces
  - Open Power CAPI-attached NVME
- More efficient algorithms – Better code generators (ECL, SALT, KEL)

Resources

- Portal: http://hpccsystems.com
- ECL Language Reference: https://hpccsystems.com/ecl-language-reference
- SALT: https://hpccsystems.com/enterprise-services/purchase-required-modules/SALT
- KEL: https://hpccsystems.com/download/free-modules/kel-lite
- Machine Learning: http://hpccsystems.com/ml
- Online Training: http://learn.lexisnexis.com/hpcc
- HPCC Systems Blog: http://hpccsystems.com/blog
- HPCC Systems Wiki & Red Book: https://wiki.hpccsystems.com
- Our GitHub portal: https://github.com/hpcc-systems
- Community Forums: http://hpccsystems.com/bb
- Case Studies: https://hpccsystems.com/resources/case-studies

HPCC SYSTEMS®