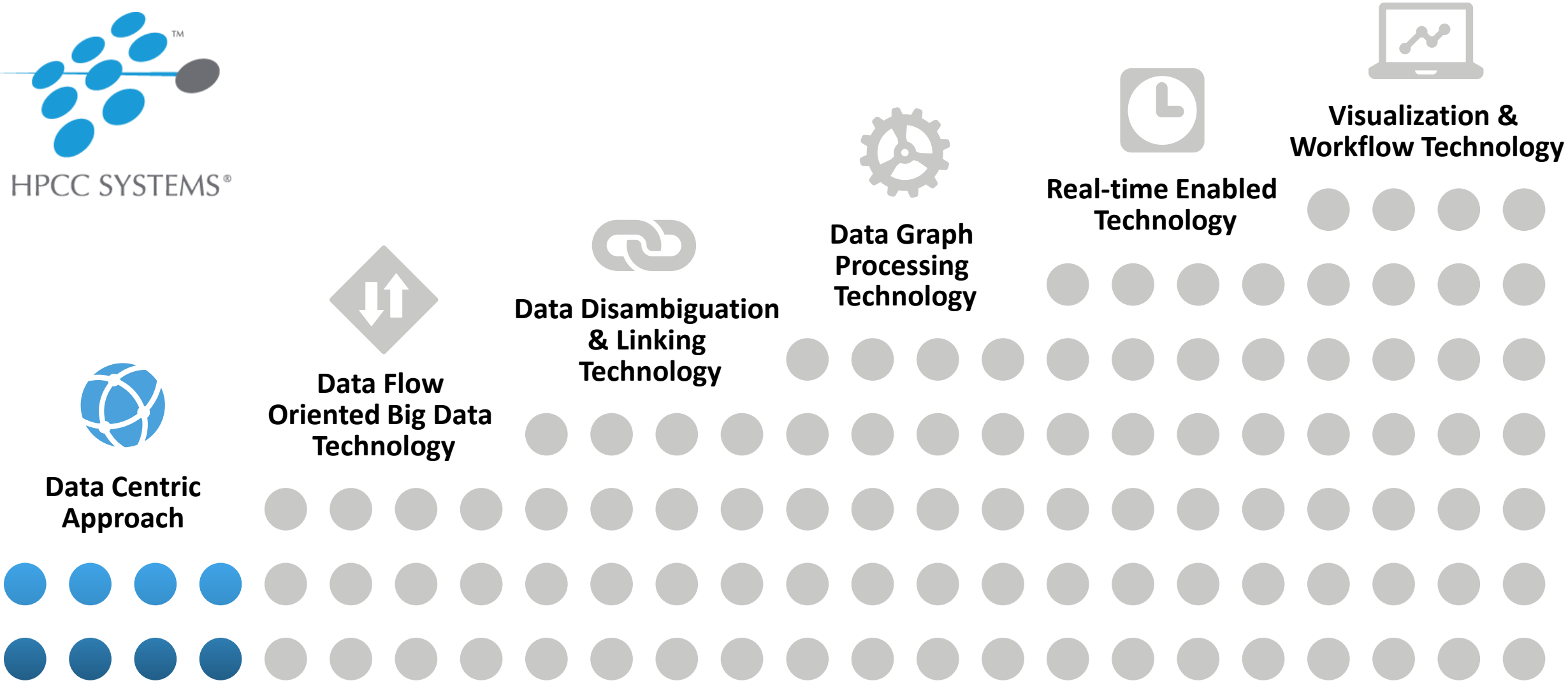
Introduction to HPCC Systems®

Powered by RELX Group

Flavio Villanustre
VP, Technology

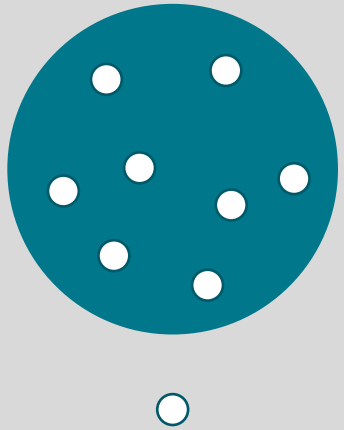
October 20, 2015

Section One: Data Centric Approach



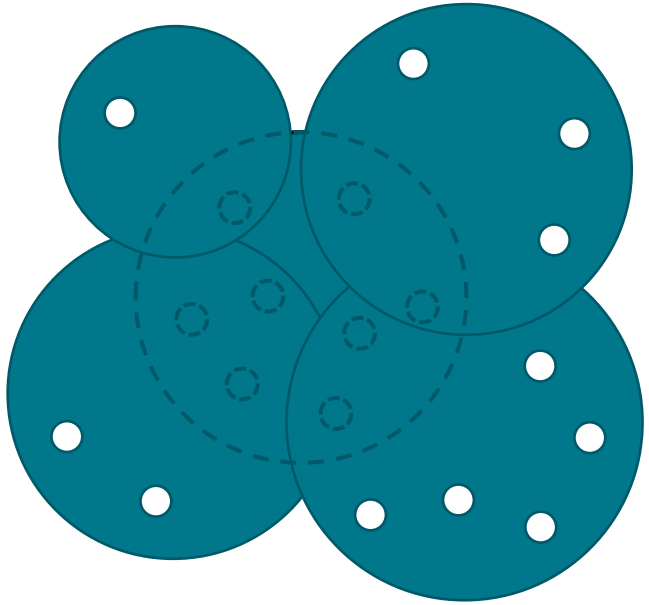
The Data Centric Approach

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



The holes are inaccuracies found in the data.

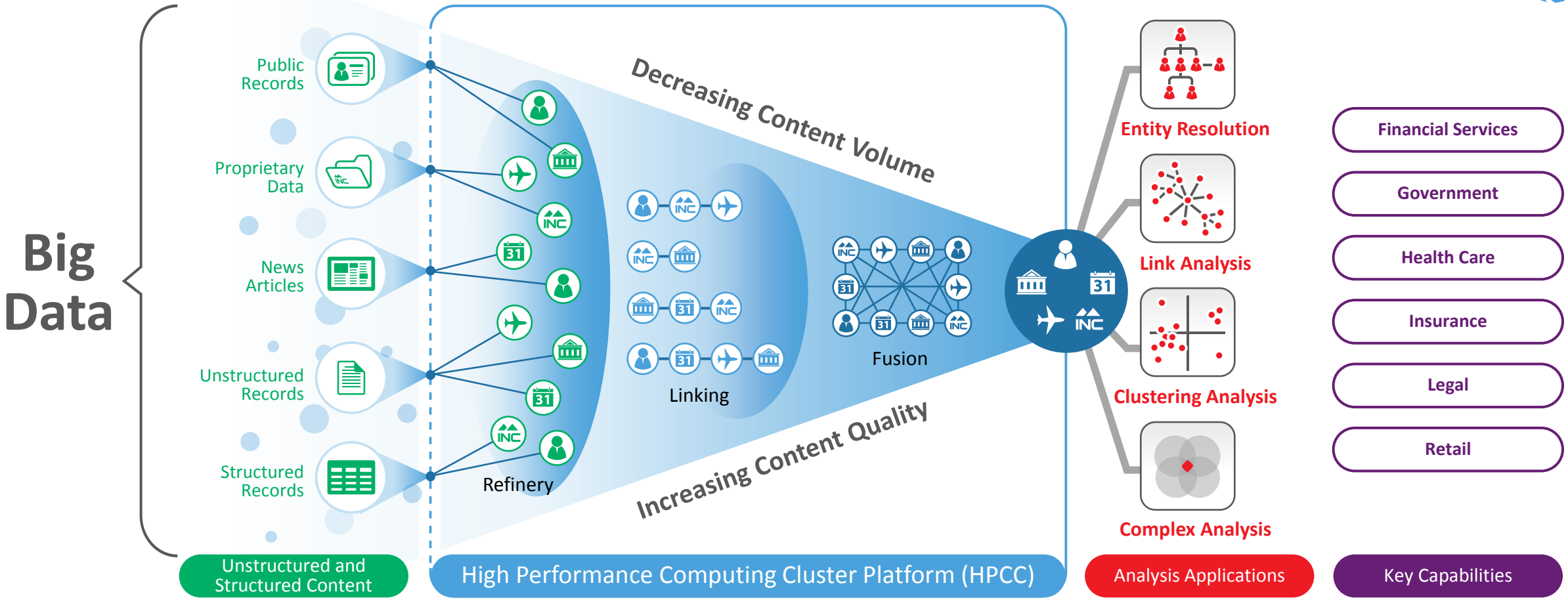
Our platform is built on the premise of absorbing data from **multiple data sources** and transforming them to a **highly intelligent social network graphs** that can be manipulated to extract the non-obvious value.



The holes in the core data have been eliminated.

HPCC SYSTEMS®

Our Solutions Are Powered by HPCC at Their Core



- Over 4 petabytes of content
- 50 billion records
- 10,000 sources
- 7.5 billion unique name and address combinations

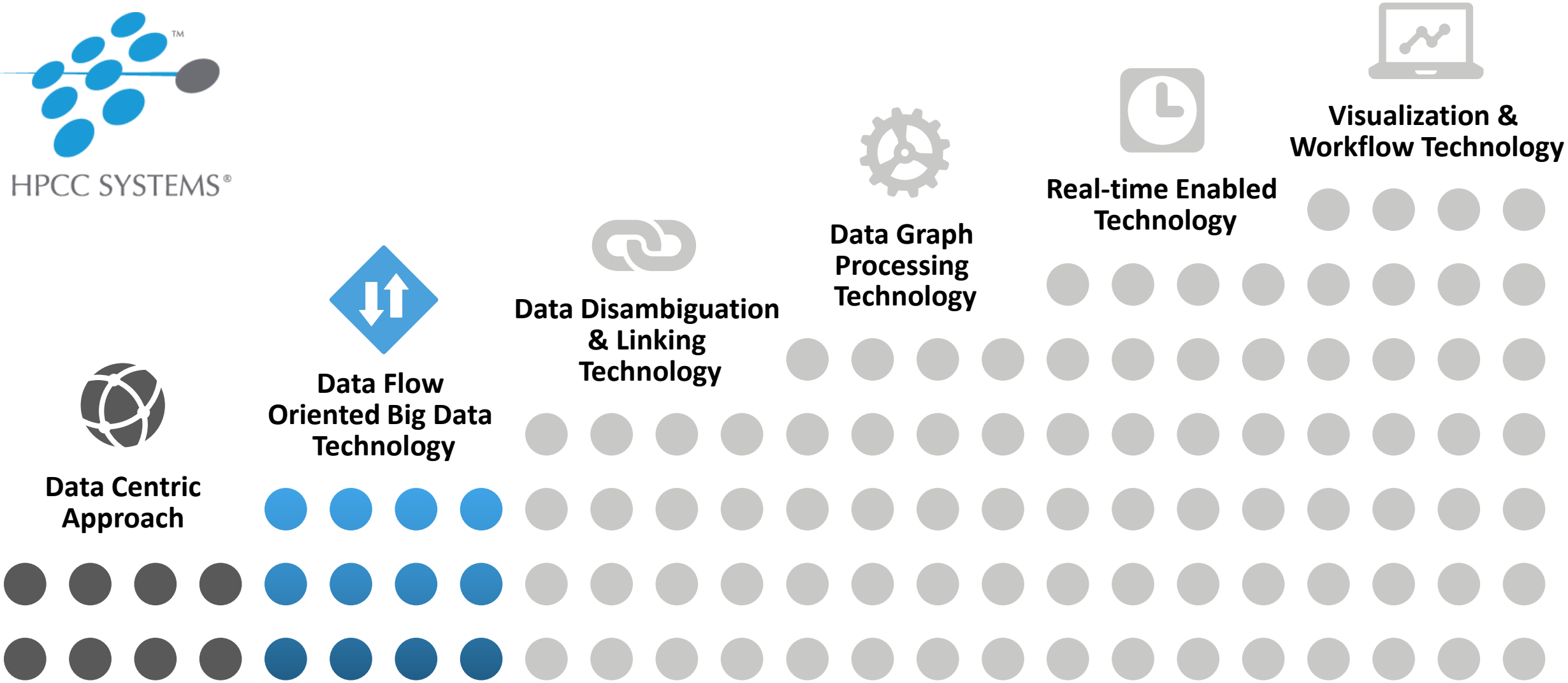
- Grid computing
- Data-centric language (ECL)
- Integrated delivery system that offers data plus analytics

Open Source Components

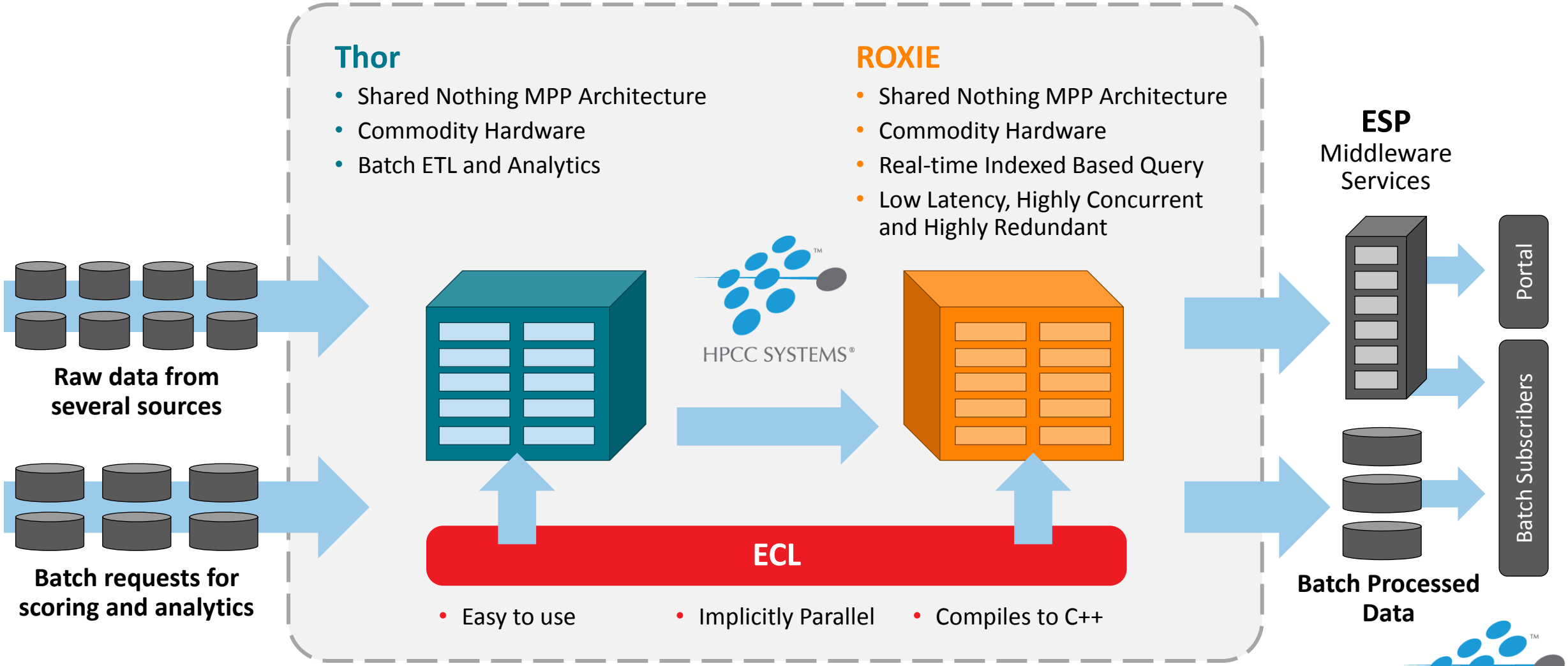
- Multi-bureau/multi-source models and bureau roll-over support
- Extensive experience leveraging atomic level data, combining and leveraging disparate data
- Approximately 400 models deployed (custom and flagship)

- Data and analytics
- Identity verification and authentication
- Fraud detection and prevention
- Investigation
- Screening
- Receivables management

Section Two: Data Flow Oriented Big Data Technology

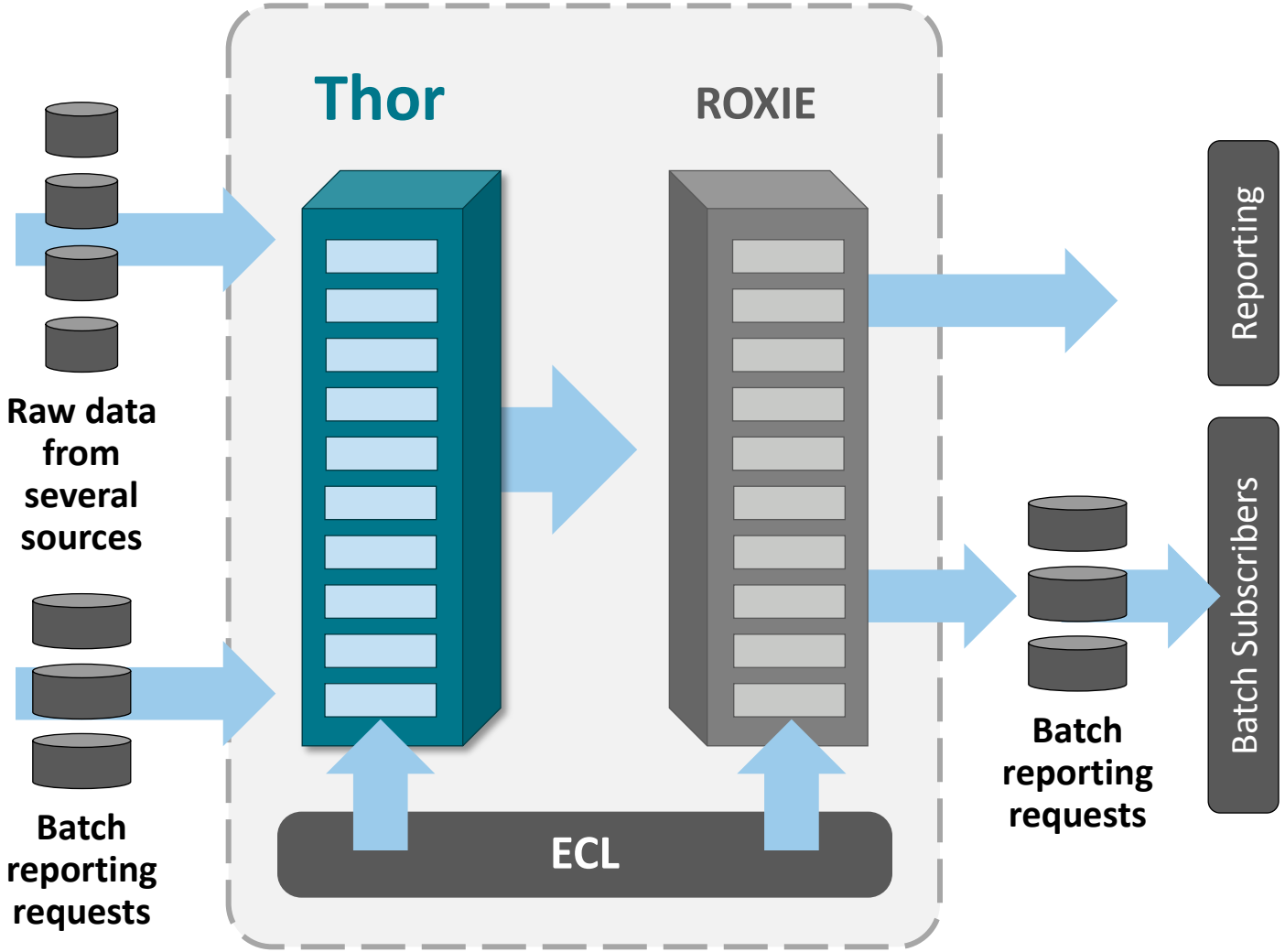


Data Flow Oriented Big Data Platform



HPCC SYSTEMS®

Thor – The Batch Processing Analytics Engine



Massively Parallel Extract Transform and Load (ETL) engine

- Built from the ground up as a parallel data environment.
- Leverages inexpensive locally attached storage.
- Doesn't require a SAN infrastructure.

Enables data integration on a scale not previously available

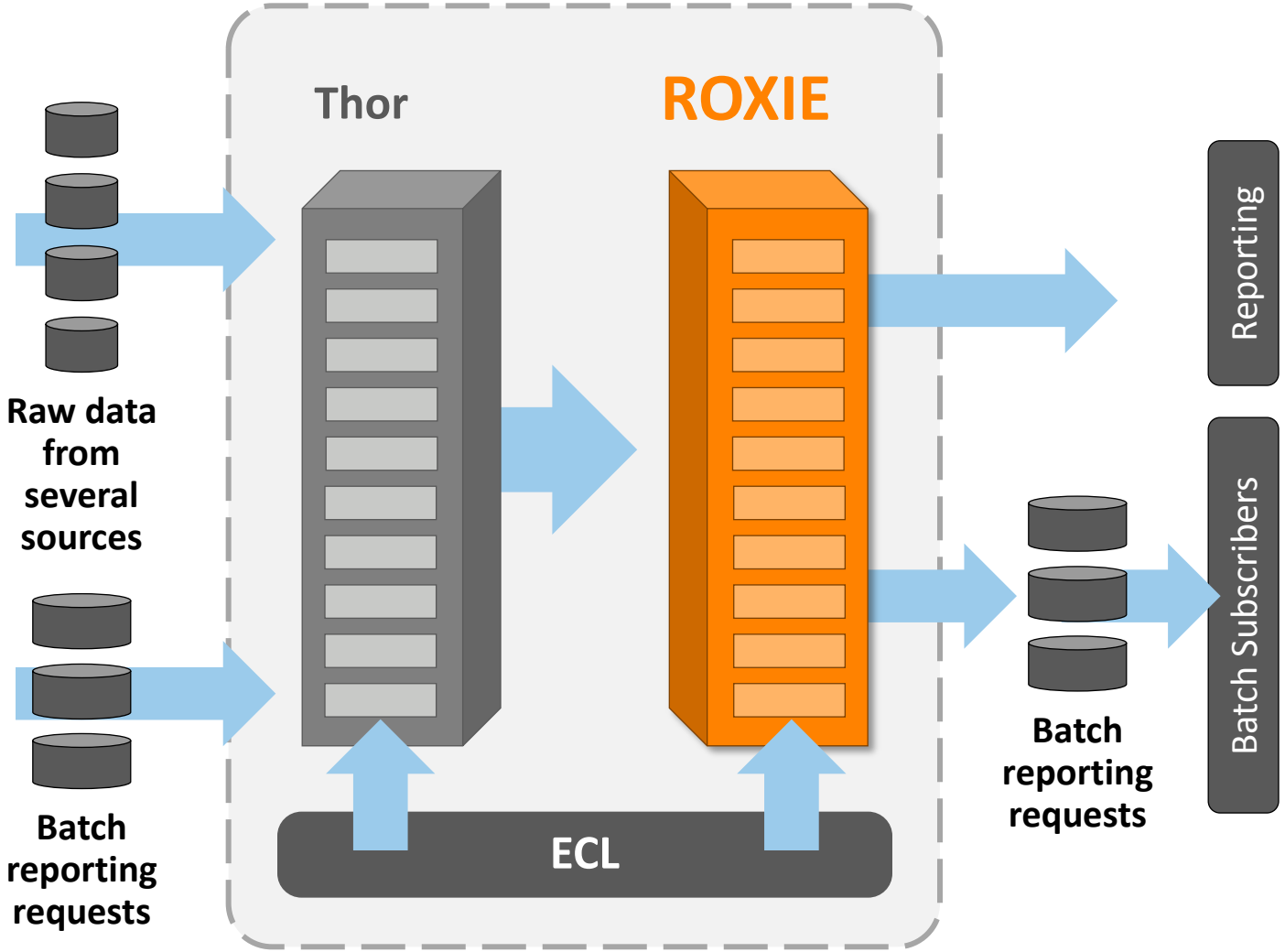
- Current LexisNexis person data build process generates 350 billion intermediate results at peak.

Suitable for:

- Massive joins/merges
- Massive sorts and transformations
- Any N2 problem

HPCC SYSTEMS®

ROXIE – The Real-Time Analytics Delivery Engine



A massively parallel, high throughput, structured query response engine.

Ultra fast due to its read-only nature.

Allows indices to be built onto data for efficient multi-user retrieval of data.

Suitable for:

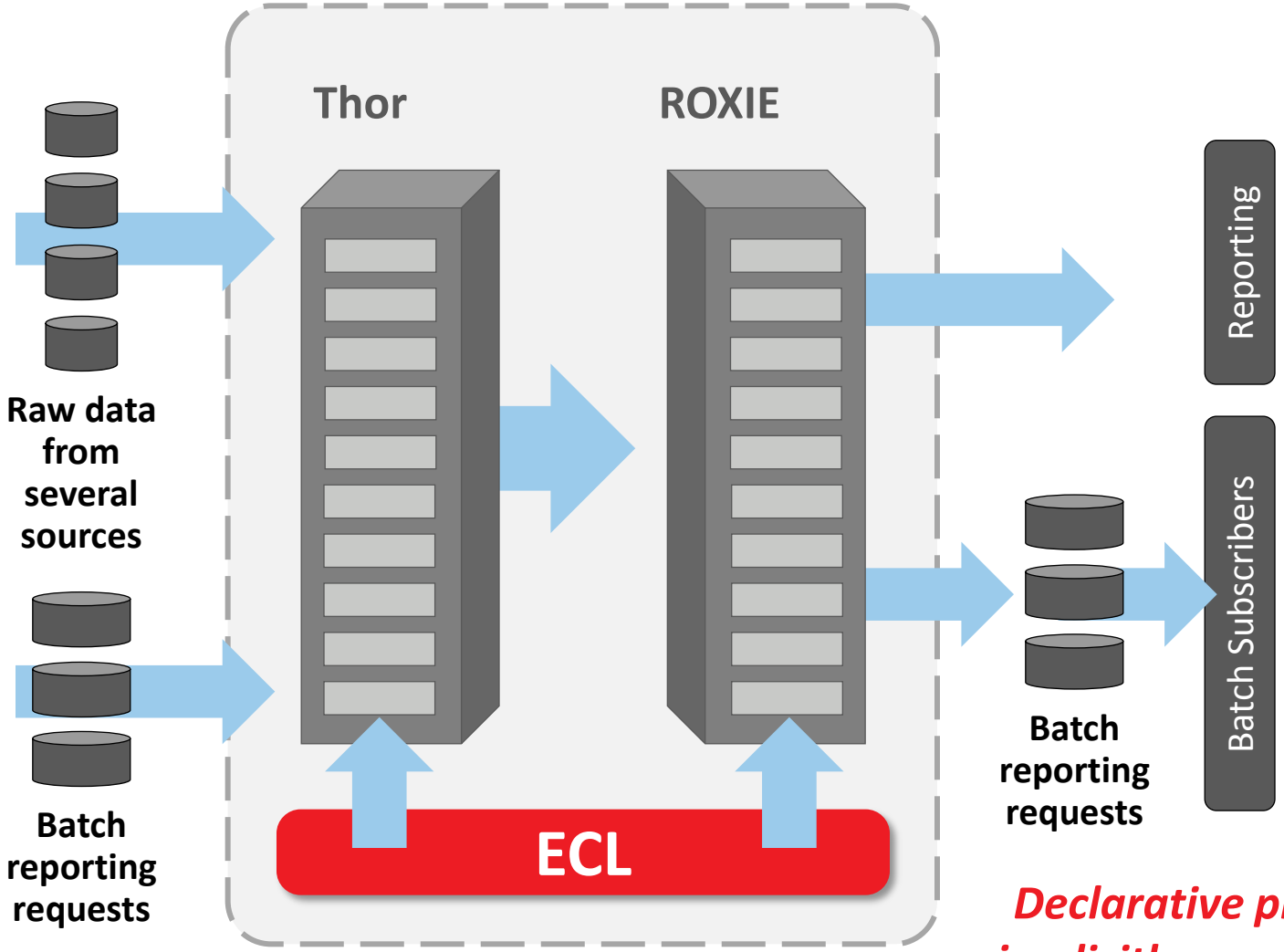
- Volumes of structured queries
- Full text ranked Boolean search

HPCC SYSTEMS®

Scalability Profiles – Thor vs. ROXIE

Thor		ROXIE
• Scales to available memory • Spills to disk as needed • Optimal architecture minimizes disk spills	**MEMORY**	• Scales to available memory • Can run in a memory only model if required
• CPU intensive • Maximize available CPU cores by optimal slave node configuration	**PROCESSING**	• Not typically CPU intensive, but complex processing can be performed as needed because of ECLs power
• Sequential reads and writes • Not heavily IO bound • SAS drives are typical • Disk spills occur as needed	**DISK**	• Random disk read • Heavily IO bound • SAS drives are typical. SSD provide the best performance
• Network bandwidth matters only when processing jobs that exchange a lot of data between nodes	**NETWORK**	• Network bandwidth is required for fulfilling a number of requests as well as exchange of data between nodes

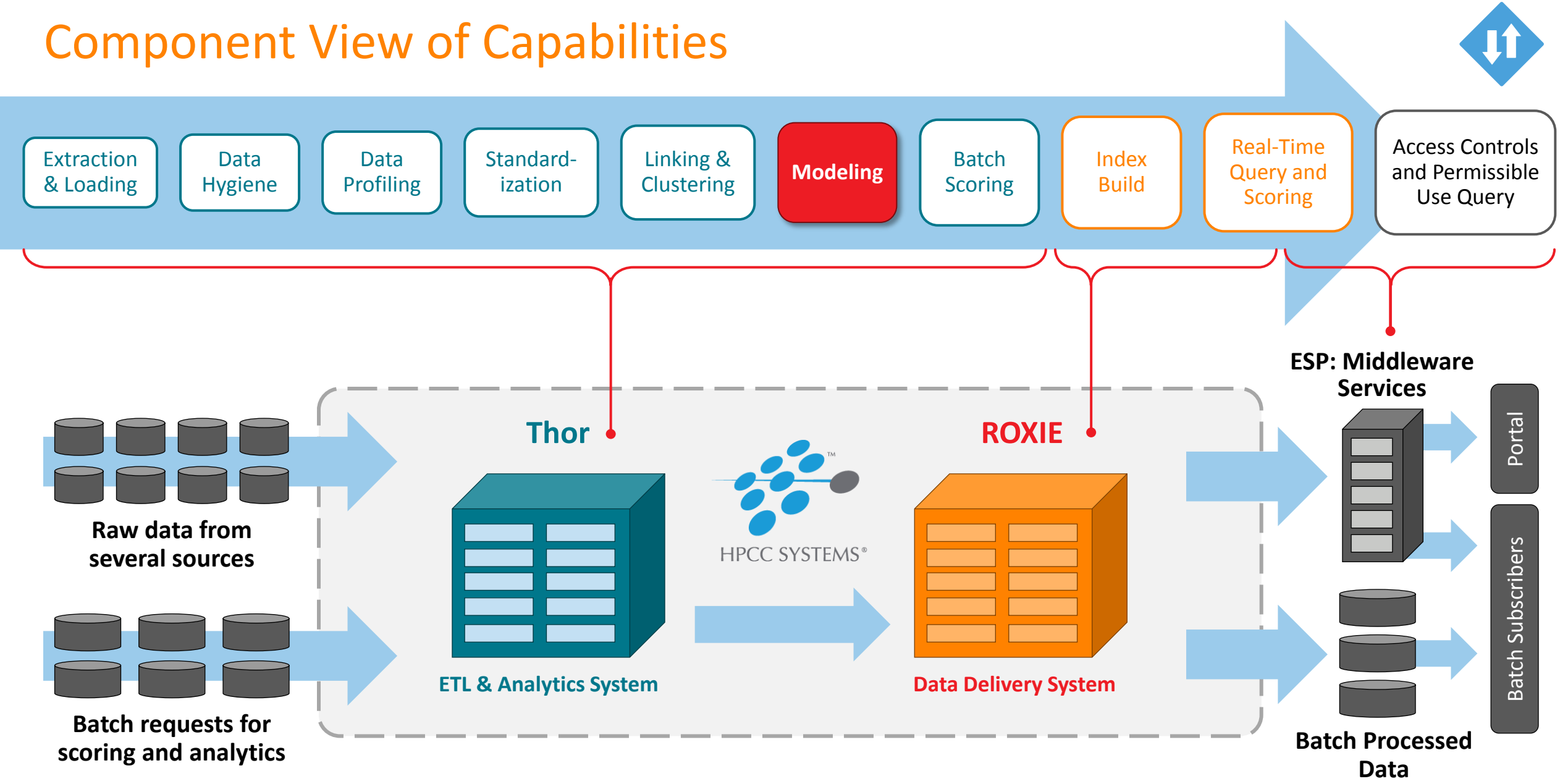
ECL – The Data Flow Oriented Programming Language



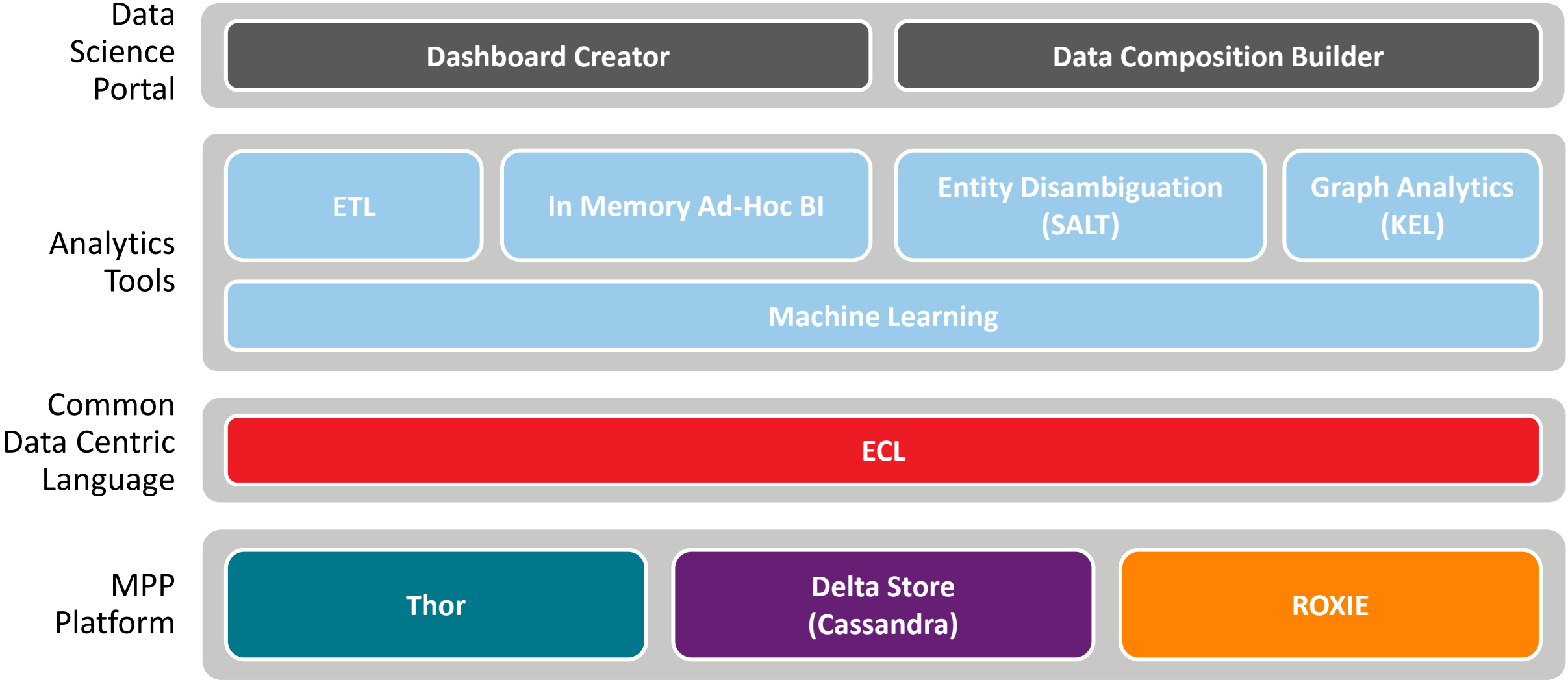
- An easy to use, data-centric programming language optimized for **large-scale data management and query processing**
- **Highly efficient** — automatically distributes workload across all nodes
- Industry analysts: "**80% more efficient** than C++, Java and SQL — 1/3 reduction in programmer time to maintain/enhance existing applications"
- Benchmark against SQL **(5 times more efficient)** for code generation
- Automatic parallelization and synchronization of sequential algorithms for **parallel and distributed processing**
- **Large library of built-in modules** to handle common data manipulation tasks

Declarative programming language … powerful, extensible, implicitly parallel, maintainable, complete and homogeneous

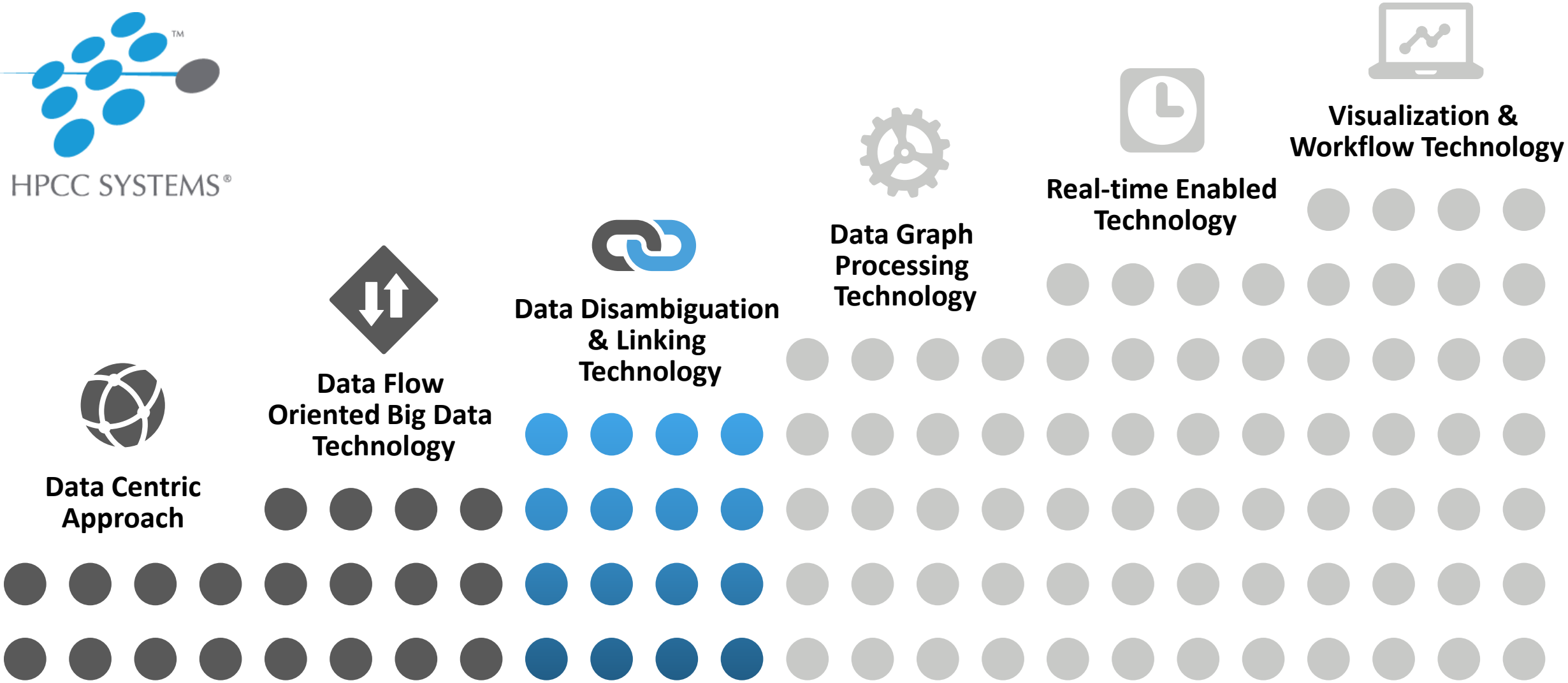
Component View of Capabilities



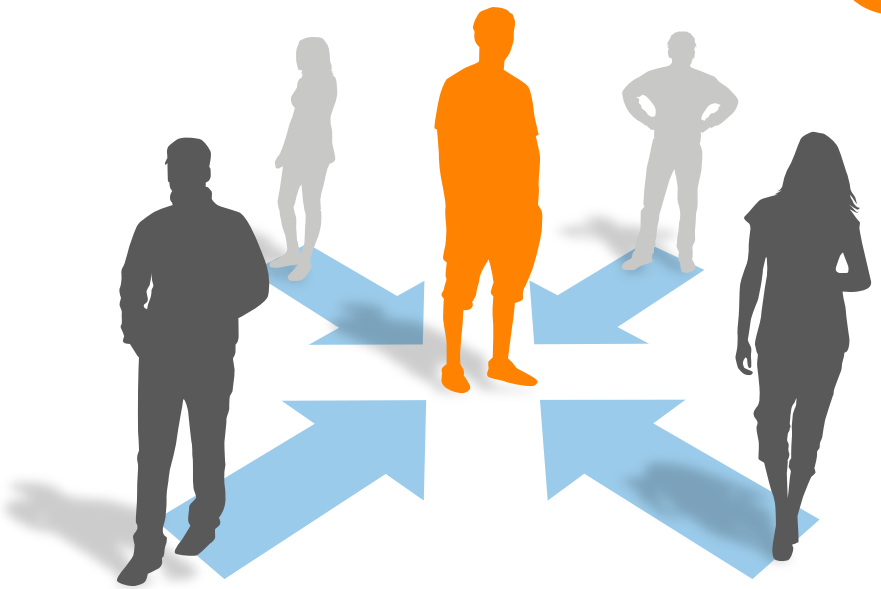
Layer View of Platform Capabilities



Section Three: Data Disambiguation & Linking Technology



An Introduction to SALT
(Scalable Automated Linking Technology)



From disparate data, to clustering, to showing relationships

HPCC SYSTEMS®

Data Disambiguation and Linking Technology

- A process that links together all records common to a single individual
- Assigns a unique identifier to those records
- Not rules based — Calculates a weight (the specificity) for each value of each field, relative to all of the other values for that field found within the data
- Includes fuzzy matching and other capabilities to help find the best matches

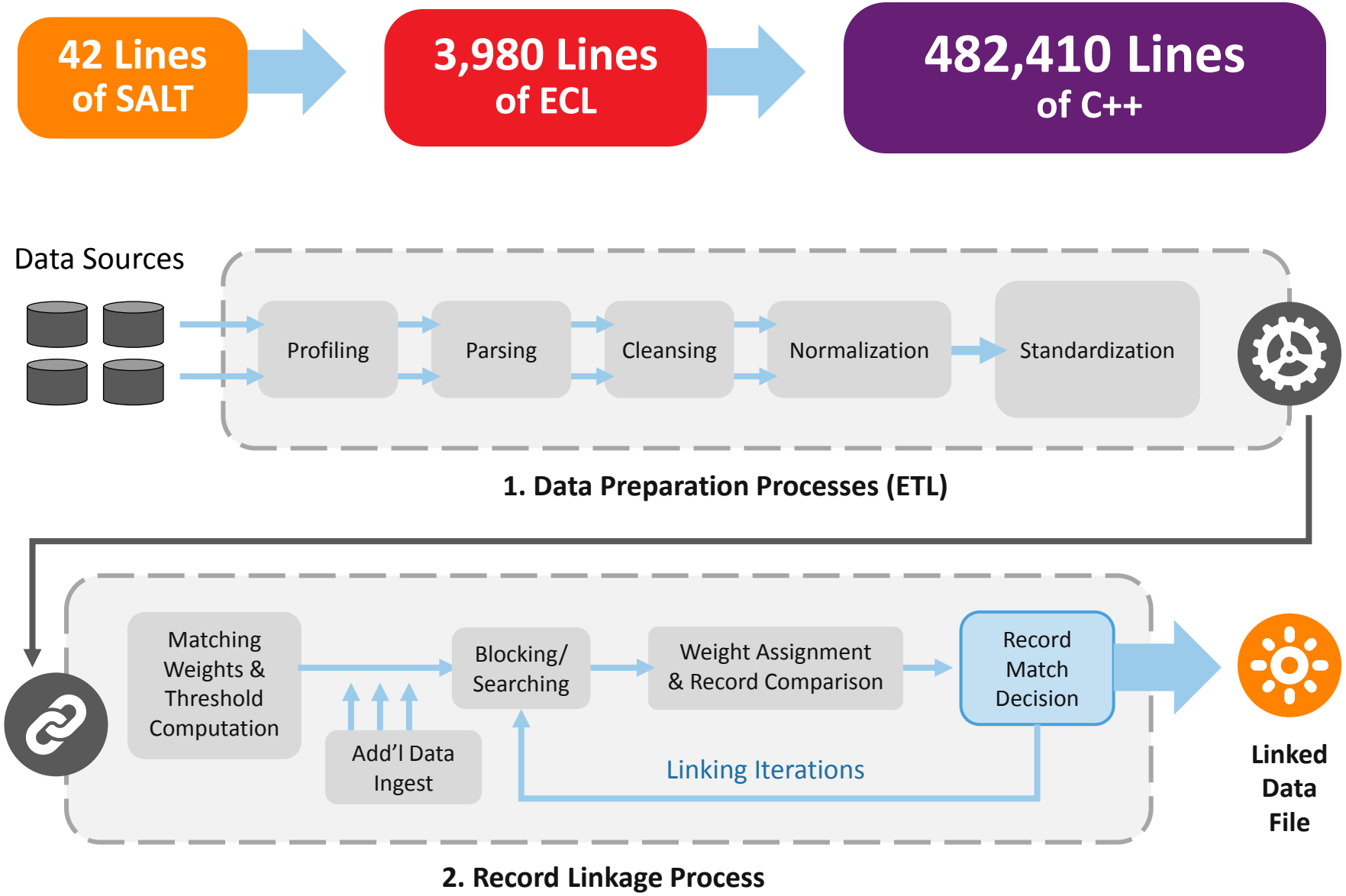
LexID: 123456
John Doe

This linking technology (SALT) powers many of our products and solutions through:

- **Linking**
- **Data Enhancement**
- **Clustering**
- **Search Processes**
- **Analytics**

SALT Increases Developer Productivity

- The acronym stands for "Scalable Automated Linking Technology"
- 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

1. **Fl**avio Villanustre, Atlanta
2. **J**avio Villanustre, Atlanta

SALT

MATCH — the system has learnt that **"Villanustre" is specific** because the frequency of occurrence is **small** and there is **only one present in Atlanta**

ERROR

NO MATCH — because the rules determine that "Flavio" and "Javio" are not the same

INPUT | **RULES**

SALT eliminates **FALSE POSITIVES** using probabilistic learning

1. **John Smith, Atlanta**
2. **John Smith, Atlanta**

ERROR

MATCH — because the rules determine that "John Smith" and the city for both the records match

SALT

NO MATCH — the system has learnt that **"John Smith" is not specific** because the frequency of occurrence is **large** and there are **many present in Atlanta**

An Entity Disambiguation Example

Record	First Name	Last Name	Address	City	State	Zip
1	MARCIA	MARSUPIAL	6035 JONES ST	ARVADA	CO	80004
2	KAREN	KANGAROO	5865 W OHIO AVE	LAKEWOOD	CO	80226

Do these two records belong to the same person?

HPCC SYSTEMS®

An Entity Disambiguation Example

Among the thousands of data sources, LexID found a series of related records

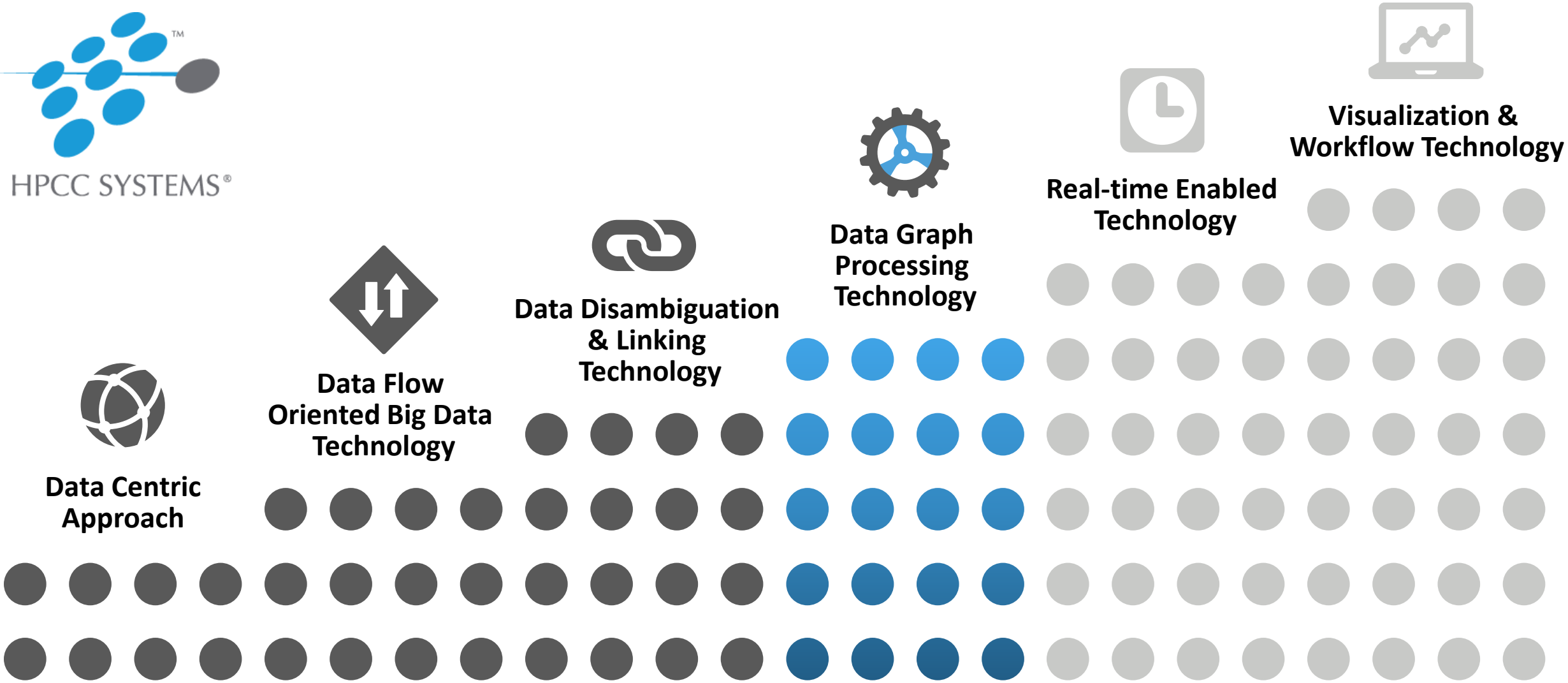
	First	M.I.	Last	St #	dir	St Name	suffix	city	st	zip
Matched record 1	**MARCIA**	**K**	**MARSUPIAL**	6035		JONES	ST	ARVADA	CO	80004
Moved	MARCIA	K	MARSUPIAL	**9170**	**W**	**14TH**	**AVE**	**LAKEWOOD**	**CO**	**80226**
Changed **Last Name**	MARCIA	K	**KANGAROO**	9170	W	14TH	AVE	LAKEWOOD	CO	80226
Used **Middle Initial** as First Name	**K**	MARCIA	KANGAROO	9170	W	14TH	AVE	LAKEWOOD	CO	80226
Moved & matched record 2	**KAREN**	**MARCIA**	**KANGAROO**	**5865**	**W**	**OHIO**	**AVE**	**LAKEWOOD**	**CO**	**80226**

Thus these two seemingly unrelated records were matched for the response analysis!

* Example is real scenario from actual campaign analysis with the names changed to protect the customers information

HPCC SYSTEMS®

Section Four: Data Graph Processing Technology



Graph Processing Technology

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)

ECL	KEL
Generates C++ (1->100)	Generates ECL (1->12)
Files and Records	Entities and associations
Detailed control of data format	Loose control of input format; none of processing
Can write graph and statistical algorithms	Major algorithms built in
Thor/ROXIE split by ***human design***	Thor/ROXIE split by ***system design***
Solid, reliable and mature	R&D

HPCC SYSTEMS®

Easily Traverse, Query and Build Attributes on Graph Relations

KEL is a graph representation language used to represent:

1 **Entities**

```
Person := ENTITY(FLAT(UID=id, STRING fname, STRING mname,
              STRING lname, STRING sex, STRING bdate, INTEGER age,
              INTEGER income, STRING address1, STRING address2, STRING city, STRING state));

Vehicle := ENTITY(FLAT(UID=id, INTEGER myear, STRING make=name.make, STRING model=name.model,
               STRING body=name.body, STRING vin, INTEGER msrp, STRING purchasedate,
               INTEGER purchaseprice, INTEGER value));

Claim := ENTITY(FLAT(UID=id, INTEGER amount, DATE claim_date, STRING totalled));
```

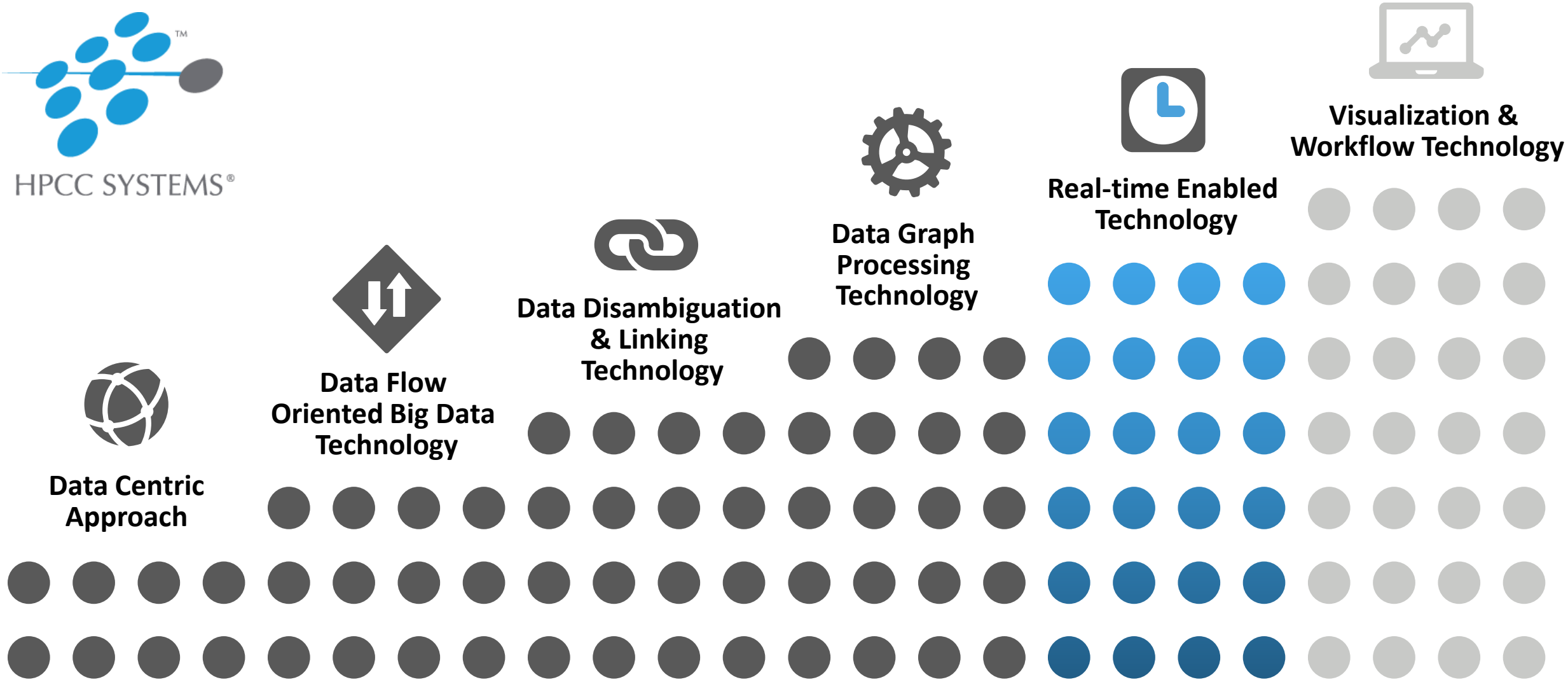
2 **Associations** (or Relations)

```
Owns := ASSOCIATION(FLAT(Person owner, Vehicle veh));
ClaimFor := ASSOCIATION(FLAT(Vehicle veh, Claim claim));
```

3 **Attributes**

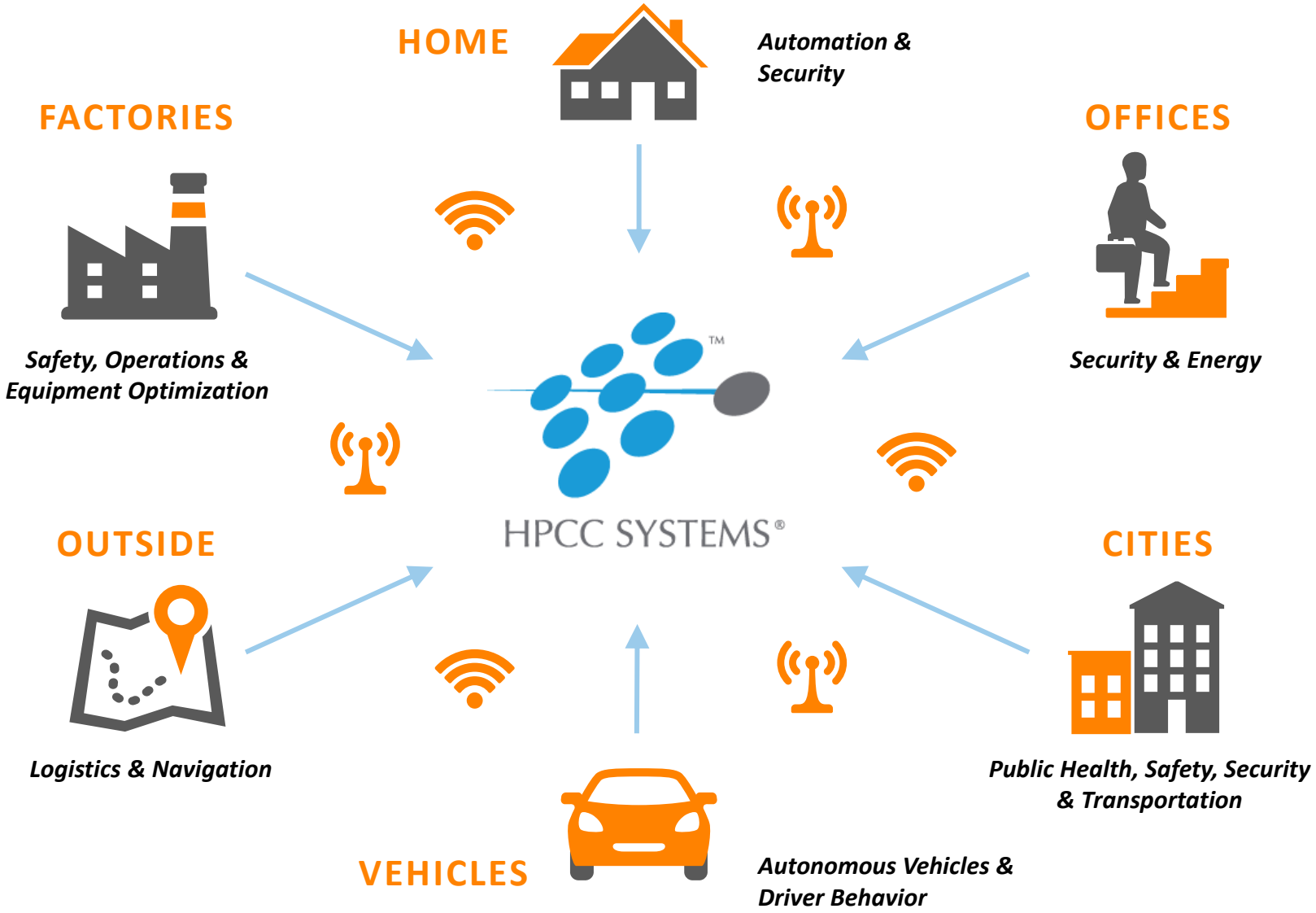
```
Claim => AgeMo := MONTHSBETWEEN (claim_date, TODAY());
Person: => ClaimAmountCountLess100 := ClaimFor.claim (amount<=100, AgeMo<=12):Count;
```

Section Five: Real-time Enabled Technology

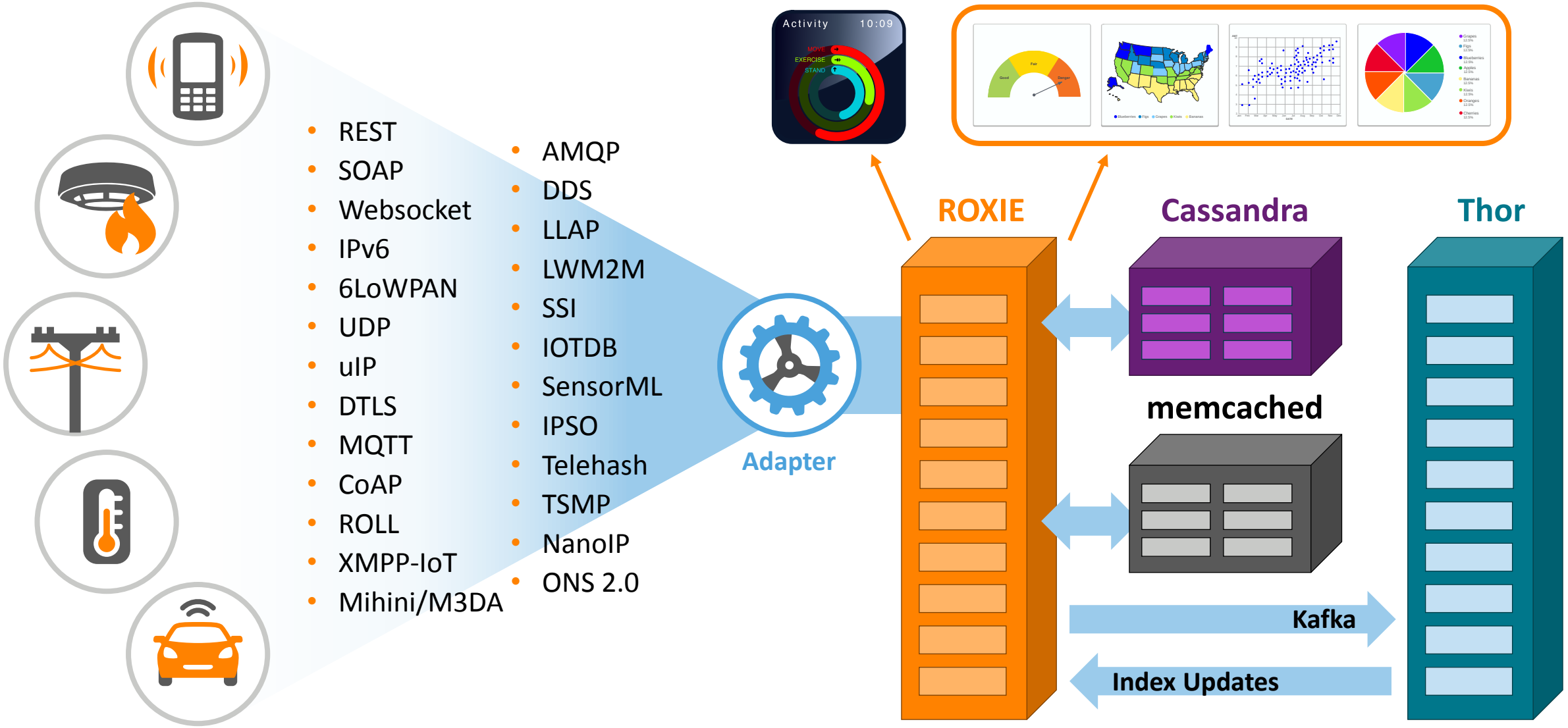


Enhanced Architecture to Support IoT

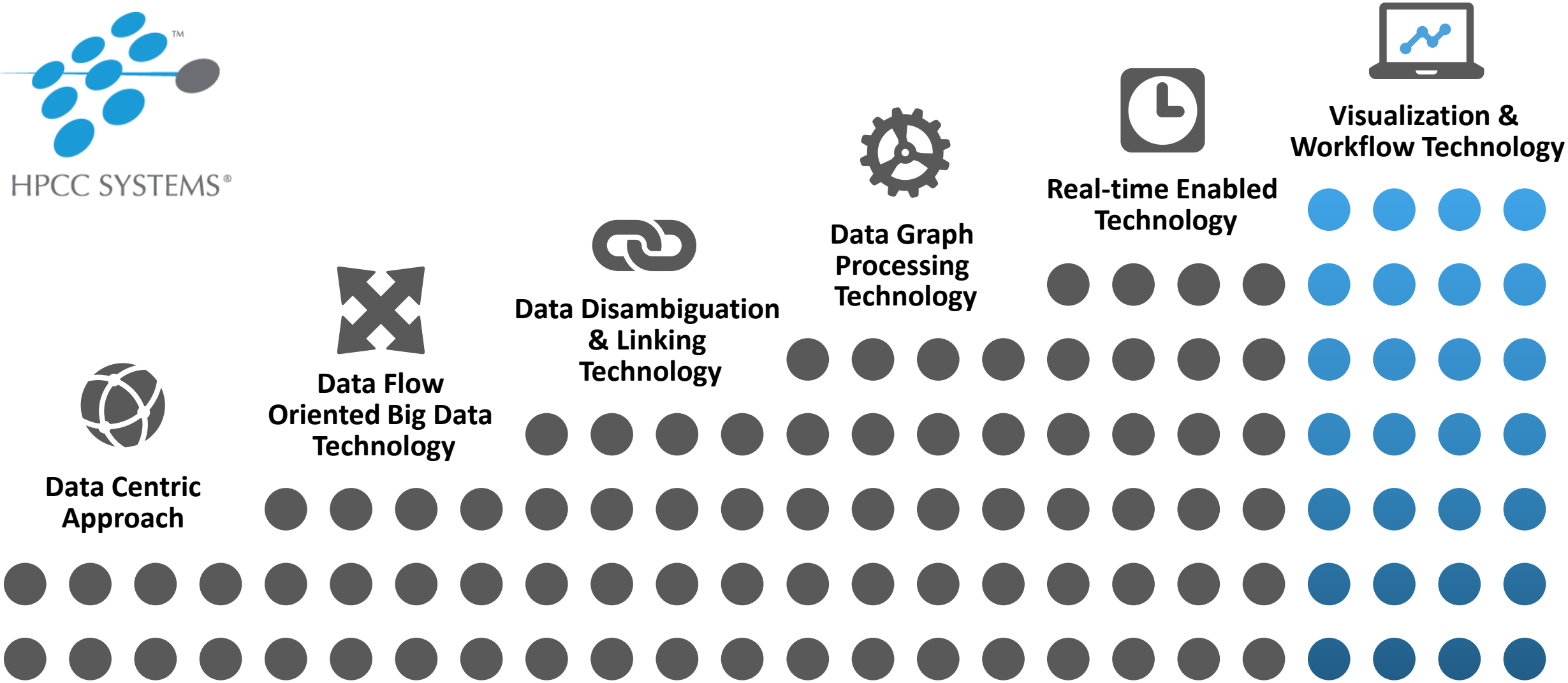
- Real-time and/or stream based data collection
- Real-time rules
- Real-time alerts
- Historic trends and analytics
- Massively scalable platform
- Real-time search and business intelligence



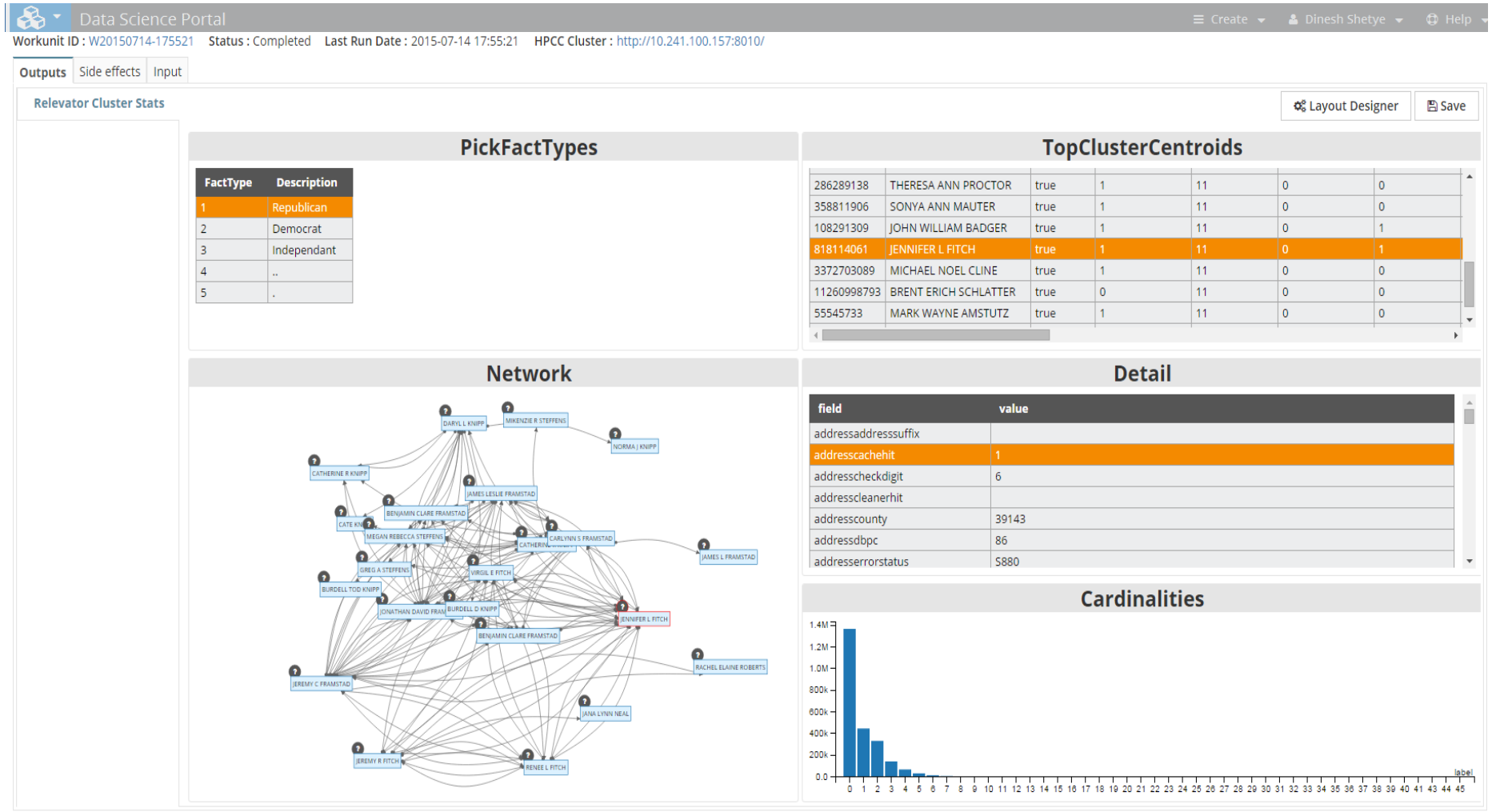
HPCC: Internet of Things Architecture



Section Six: Visualization & Workflow Technology



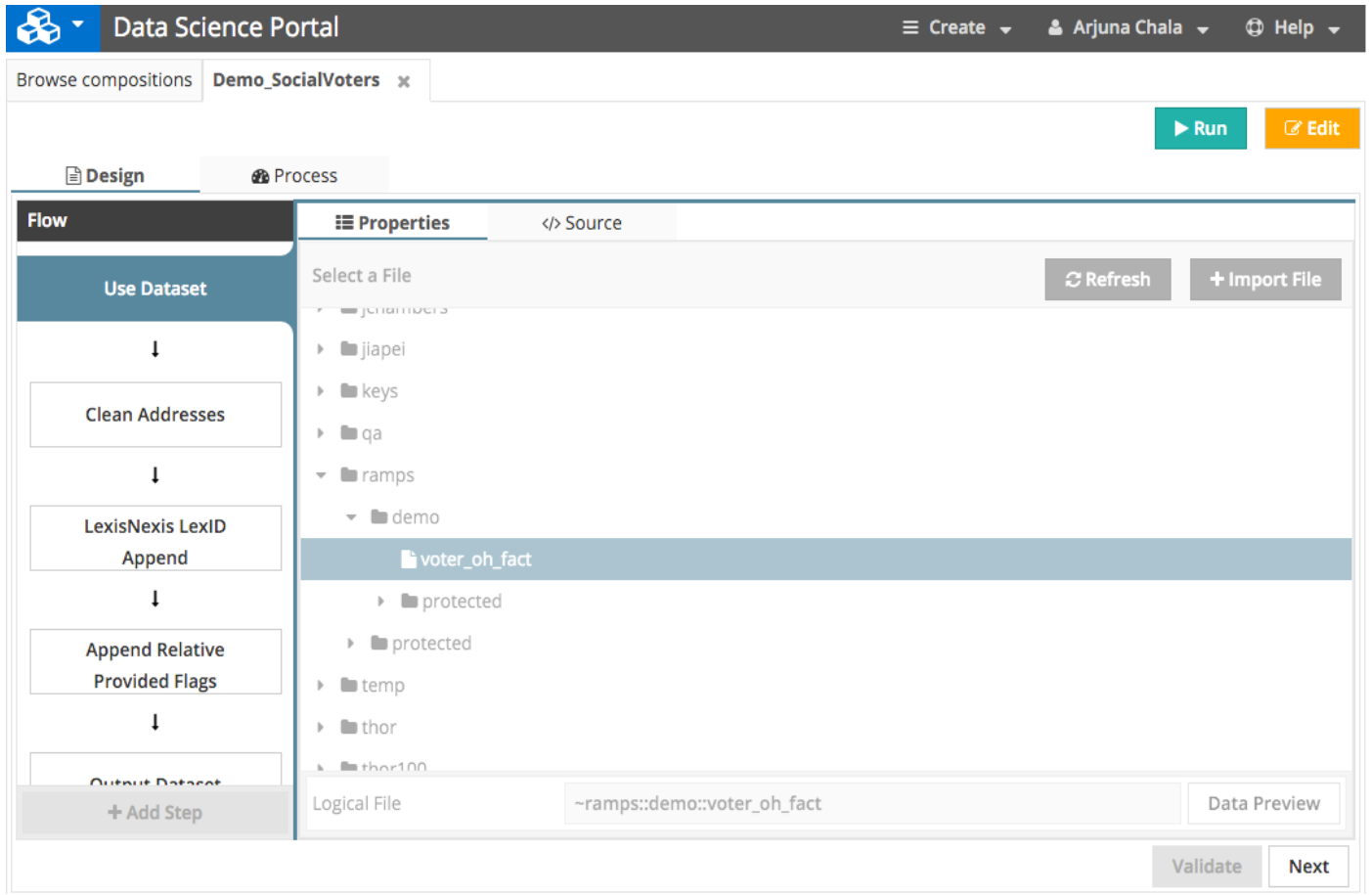
Provide a Powerful Visual Tool for Managing Data Workflows and Visualization

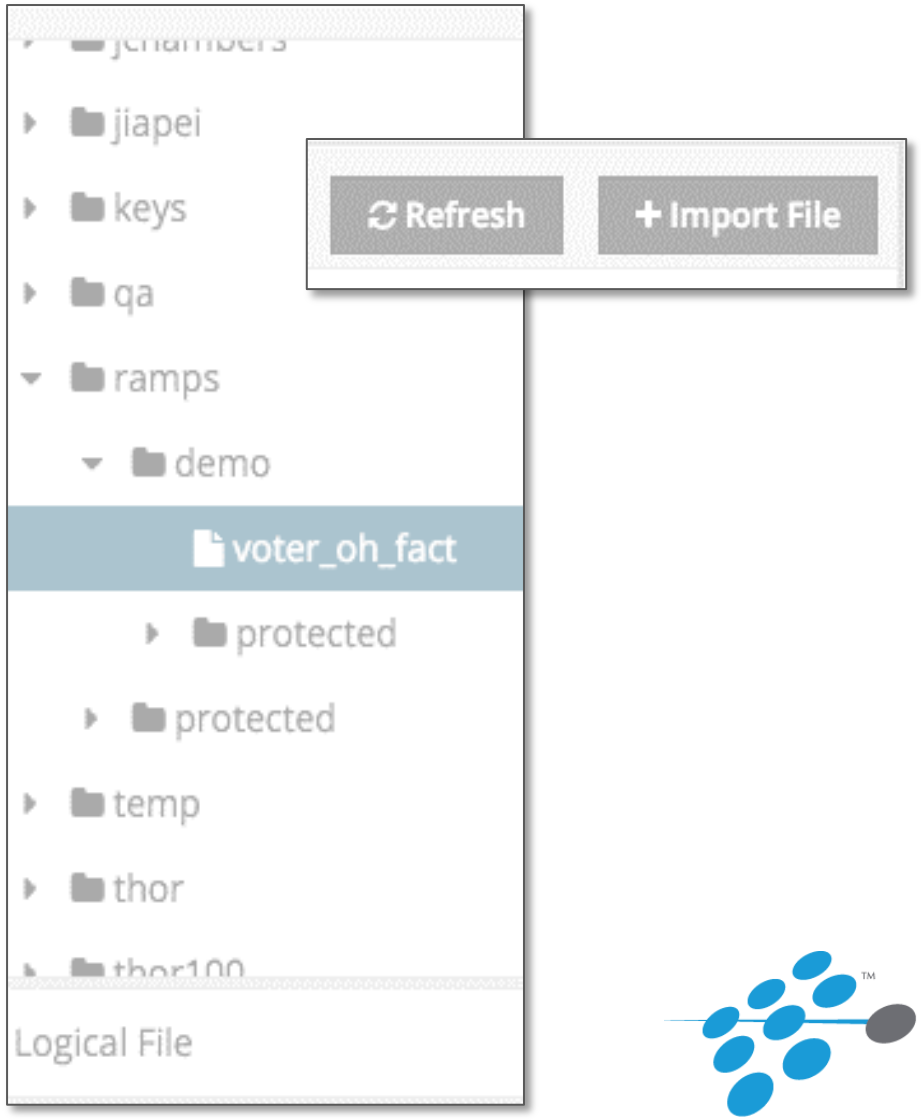


- Build complex relationship graph easily using a **drag and drop interface**
- Gain insight into hidden relationships using **powerful visualization techniques**

HPCC SYSTEMS®

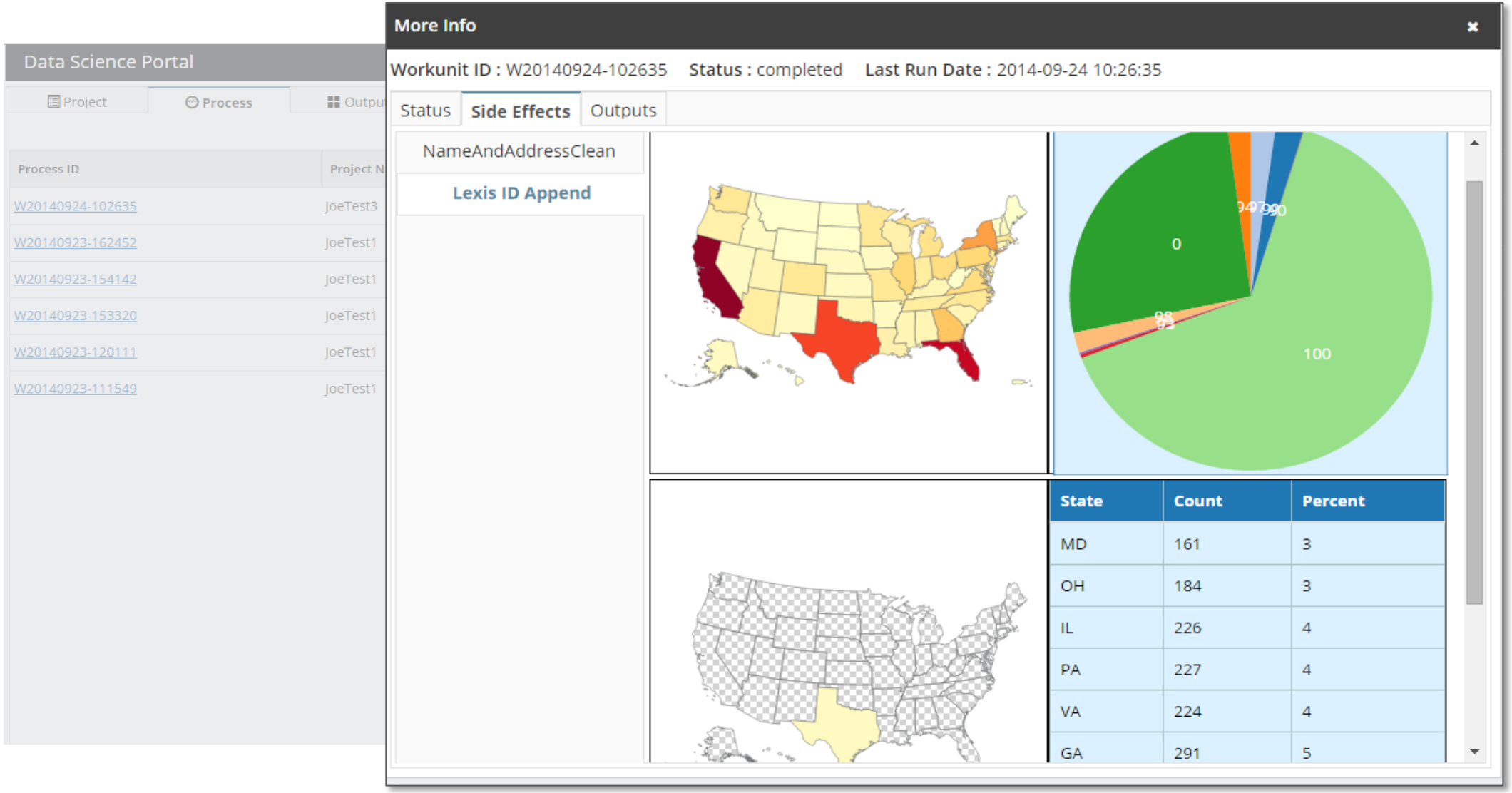
A Plugin Architecture to Developing Data Workflows





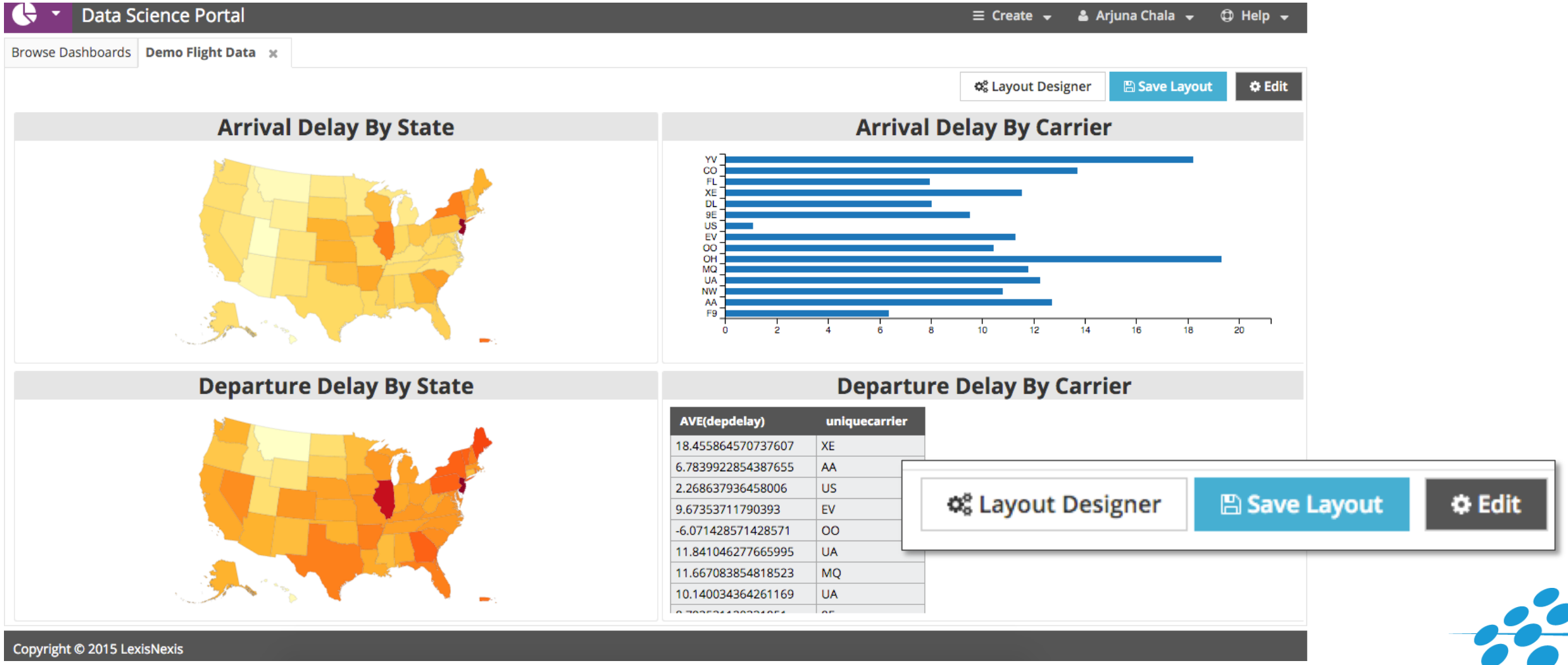
HPCC SYSTEMS®

Presenting the Side Effects Paradigm



HPCC SYSTEMS®

Powerful Dashboard Creation for End Users



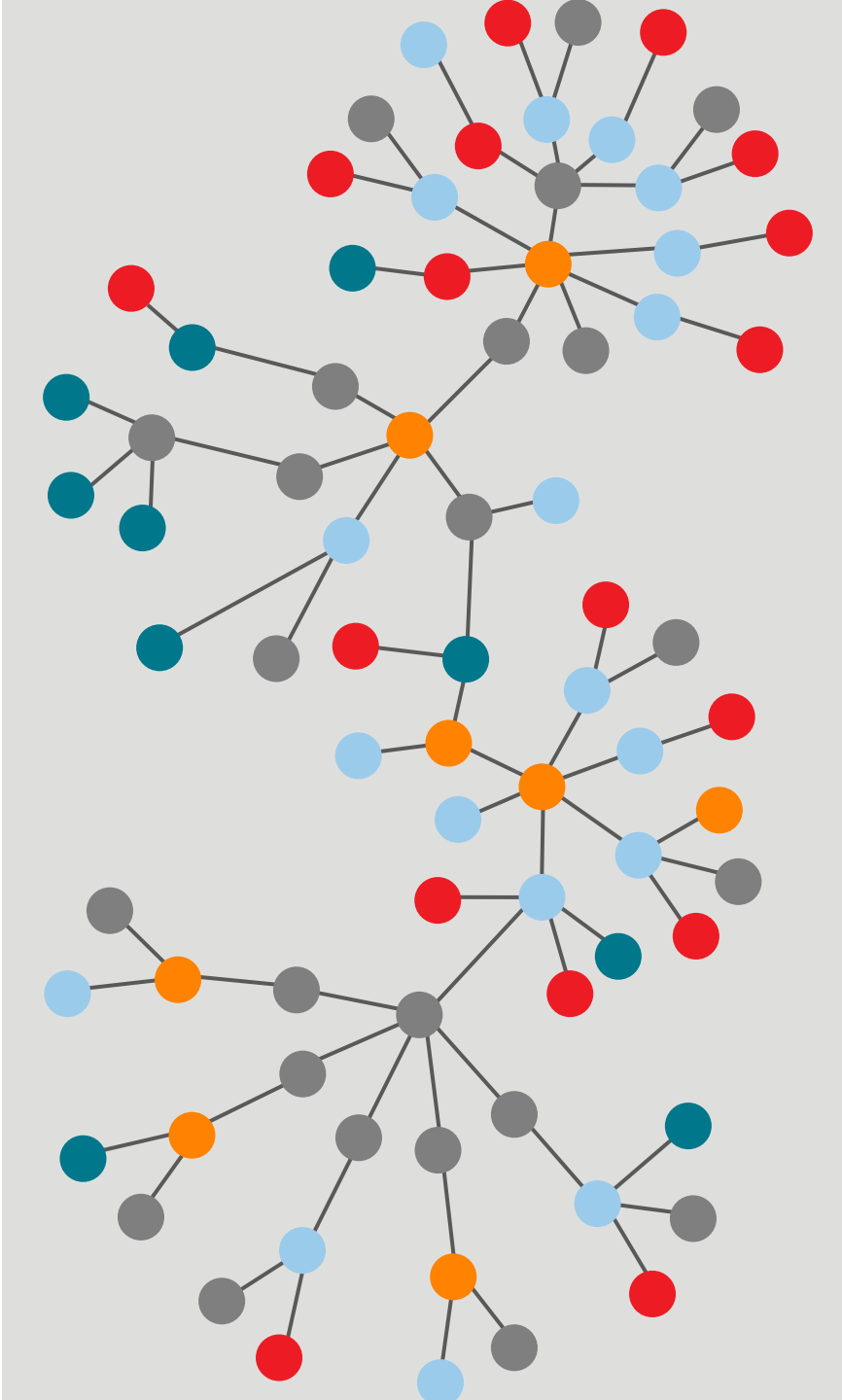
HPCC SYSTEMS®

Network Graph

A natural way to solve hard big data problems — Representing data as a set of relationships

1. Clearly identify entities of interest
2. Accurately represent relationship between the entities
3. Perform queries using degrees of separation
4. Build attributes using filters based on relationships and degrees of separation
5. Develop accurate learning models based on a social footprint

HPCC SYSTEMS®



Example #1: Understanding People Relations Helps Us Predict Risk

16.5 B	8.9 B	4 B	3.7 B	417 M	269 M	188.5 M	37 M
consumer records	unique name/ address combos	property records	motor vehicle registrations	criminal records	auto and home claim records	unique cell phones	unique businesses

- Collect largest, broadest, deepest, most accurate, up-to-date repository of public record and contributory data
- Clean and standardize the data
- Identify unique entities using sophisticated learning techniques
- Create the social relationships

HPCC SYSTEMS®

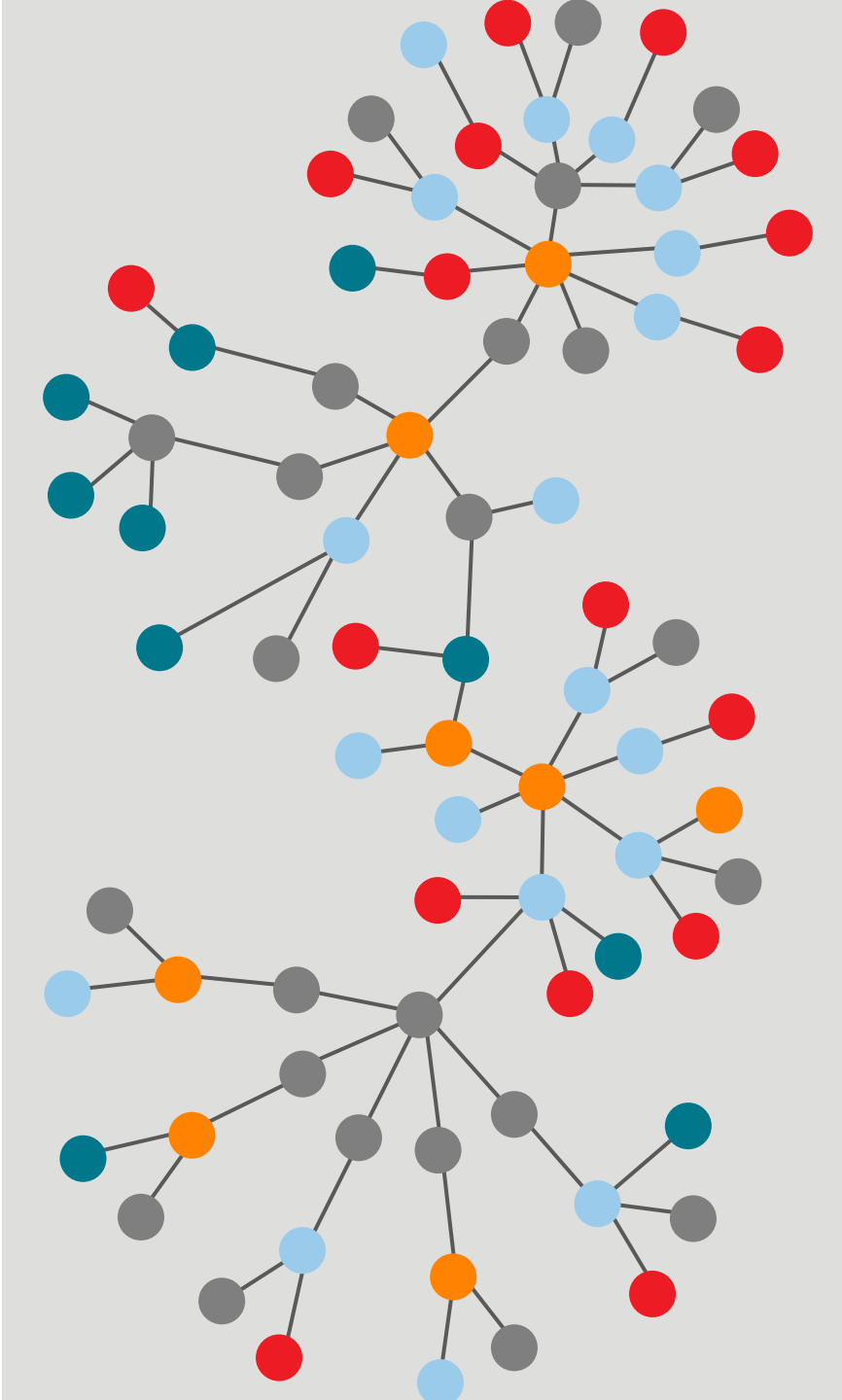
Example #2: Network Intrusion Detection by Understanding the Relationships Between IP Addresses, Country, Botnets, etc.

- Use millions of application transaction log records per day
- Combine information from network routers, DNS servers, etc.
- Use third party sources to identify blacklisted information
- Apply learning techniques to match the digital footprint to identify the virtual identifies
- Prevent intrusions by detecting anomalies

HPCC SYSTEMS®

Real World Examples on Using the Social Graph to Solve Problems

1. Ability to query the graph in real time is critical
2. Attribute calculation at various degrees of separation helps to identify non obvious clusters

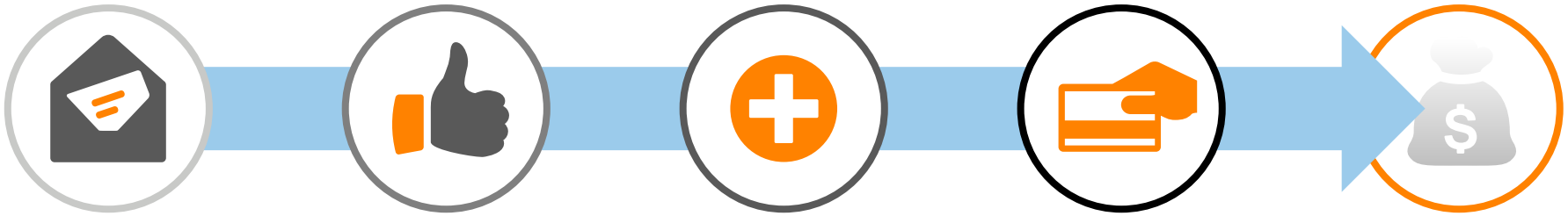


HPCC SYSTEMS®

Example #1: Graph Analysis
Bust-Out Fraud

SCENARIO

- Perpetrators typically apply for credit 4 to 24 months before busting out
- Hard to predict at an individual level



More than $2 billion in annual losses

HPCC SYSTEMS®

Example #1: Graph Analysis
The Solution Highlights

SCENARIO

- Financial company hypothesized that organized groups were targeting them and desired to tackle bust-out fraud at a social level
- Although some risk was not individually large, the company wanted to ascertain where that risk was growing without them realizing the connections

Example #1: Graph Analysis
The Solution Highlights

DATA & ANALYSIS

- 5 million accounts flagged with active, known fraud, charge offs and preemptively closed tags
- We investigated every address, asset and business connection attached to the accounts

OUR APPROACH

- Standardized input, applied LexID and joined to our Social Graph (5 million into **4 billion relationships**)
- Calculated grouped social aggregates, scored and rank ordered the result

Example #1: Graph Analysis
Results Overview

1 Known Fraud

# in 1 degree	# accounts
0	4,055,099
1	39,266
2	638
3	31
4	2

31 associated with 3 known fraud accounts

2 Charge Off

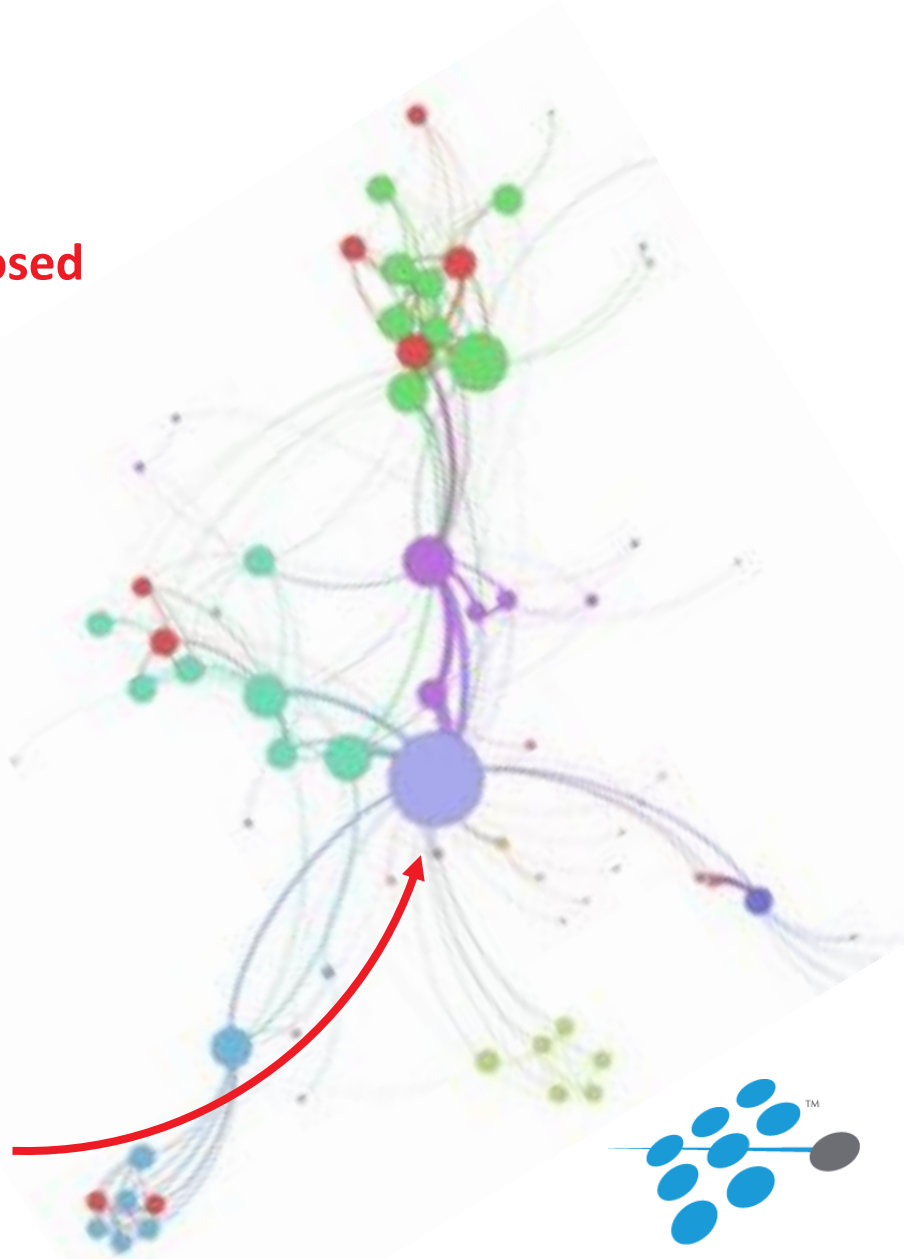
# in 2 degrees	# accounts
0	4,079,930
1	14,894
2	212

212 associated with 2 charge accounts

3 Preemptively Closed

# in 1 degree	# accounts
0	3,834,840
1	239,472
2	19,026
3	1,322
4	194
5	71
6	21
7	10
8	7
9	14
10	9
11	7
12	7
13	14
14	14
15	3
16	2
17	1
21	2

3 accounts associated with 15 preemptively closed accounts



HPCC SYSTEMS®

Example #1: Graph Analysis
Account Overview

Aggregated group behavior variables for a single account

- Flatten the graph
- Calculate aggregate group behavior measurements
- Drive predictive analytics at a granular level using graph variables

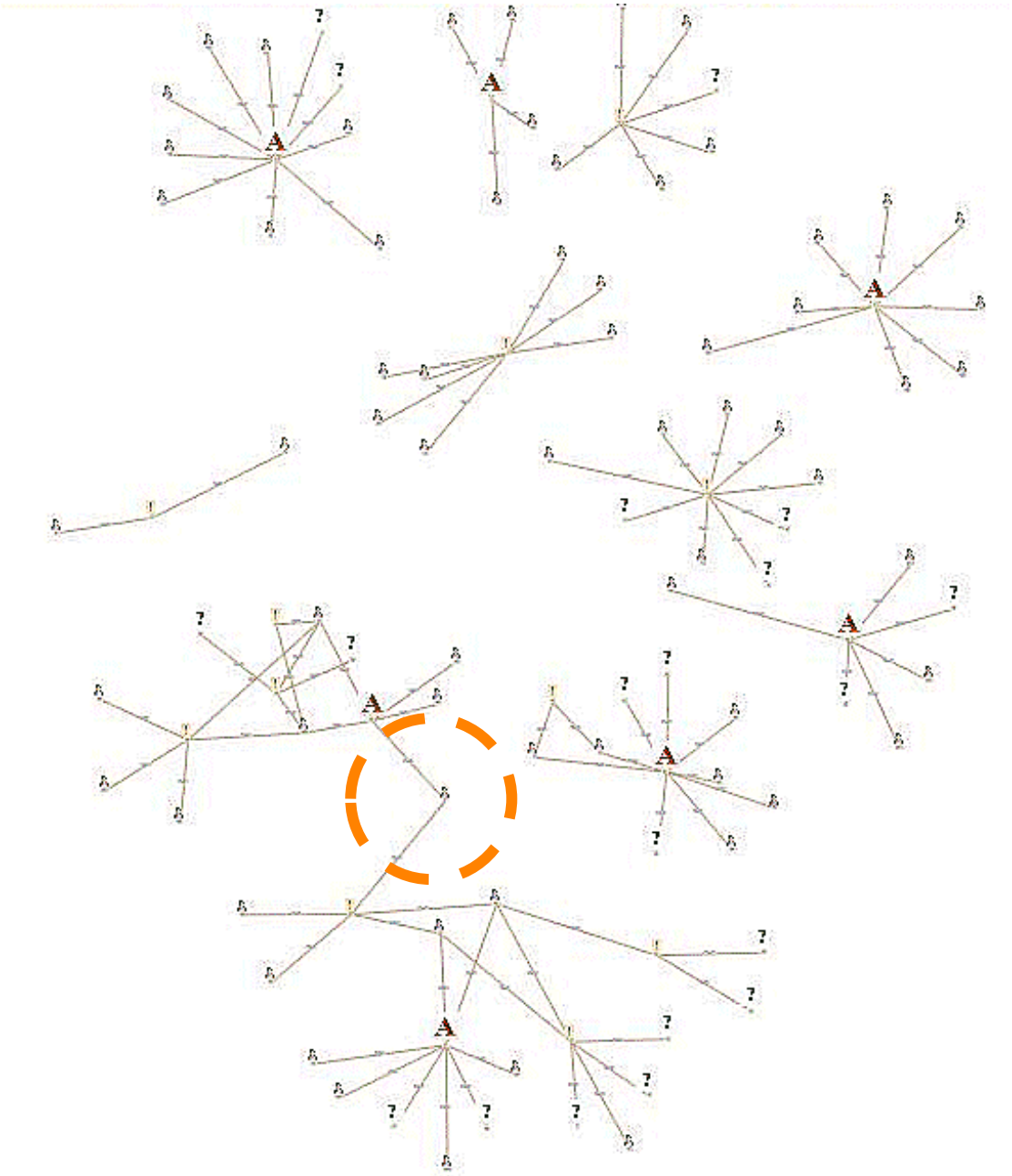
LexID: 312873
Jack Johnson
Cluster ID: 1214379

# Network Neighborhood		**81**	Small network
# 1st Degrees		8	
# 2nd Degrees		72	
Cohesivity		**1.67**	Fairly tightly connected
# Active Accounts	Total	3	**3 active accounts** within the network
	Degree 0	**1**	
	Degree 1	**1**	
	Degree 2	**2**	
# Fraud Accounts	<=2 Degrees	9	9 accounts within that network that are **known fraud**
	Degree 0	0	
	Degree 1	**3**	
	Degree 2	**6**	
# Charge Off Accounts	<=2 Degrees	0	
	Degree 0	0	
	Degree 1	0	
	Degree 2	0	
# Preemptive Close Accounts	<=2 Degrees	1	1 account within the network that's been **preemptively closed**
	Degree 0	0	
	Degree 1	0	
	Degree 2	**1**	
	SCORE	**6.6**	

Example #2: Graph Analysis
Insurance Collusion

SCENARIO

- This view of carrier data shows seven known fraud claims and an additional linked claim
- The Insurance company data **only finds a connection between two of the seven claims,** and only identified one other claim as being weakly connected



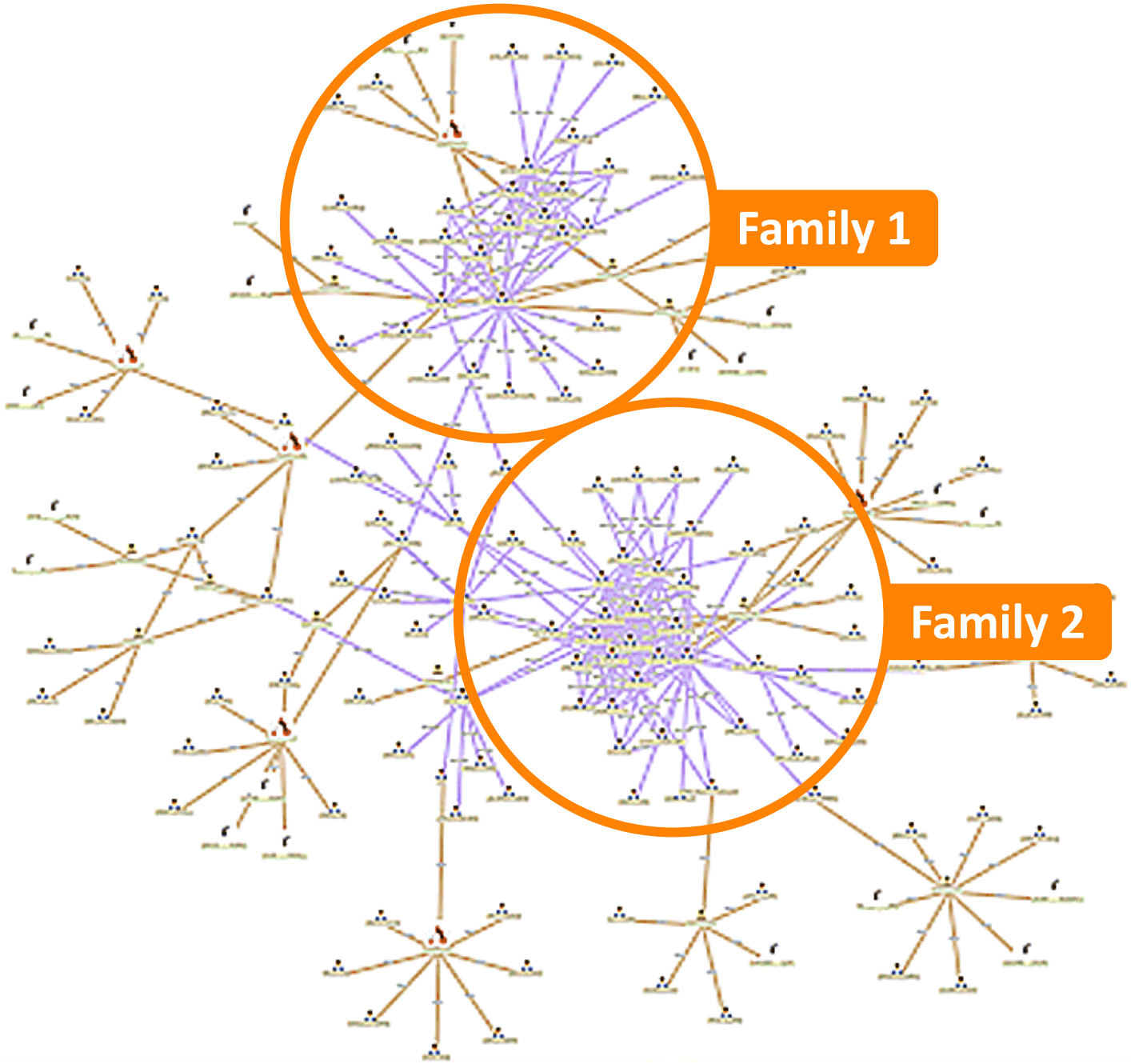
Example #2: Graph Analysis
Insurance Collusion

TASK

- After adding the Link ID to the carrier data, our HPCC technology then added 2 additional degrees of relatives

RESULT

- The results showed **two family groups interconnected on all of these seven claims**
- The links were much stronger than the carrier data previously supported



Questions?

- Open Source HPCC Systems Platform: http://hpccsystems.com
- Free Online Training: http://learn.lexisnexis.com/hpcc
- SALT: http://hpccsystems.com/salt
- KEL: https://hpccsystems.com/download/free-modules/kel-lite
- Machine Learning portal: http://hpccsystems.com/ml
- The HPCC Systems blog: http://hpccsystems.com/blog
- Community Forums: http://hpccsystems.com/bb
- Source Code: https://github.com/hpcc-systems
- WIKI: https://wiki.hpccsystems.com

HPCC SYSTEMS®