

•DEMO TALK•



**Bob Foreman**

Senior Software Engineer  
**LexisNexis Risk  
Solutions**



**Hugo Watanuki**

Senior Technical Support Engineer  
**LexisNexis Risk  
Solutions**

# HPCC Systems – The Kit and Kaboodle for Big Data and Data Science

**ODSC WEST**

DATA SCIENCE  
VIRTUAL TRAINING CONFERENCE

**October  
28–29**

# HPCC Systems: End to End Data Lake Management



Completely  
free

open source data  
lake solution



Out of the box capabilities  
for consistency and  
ease of use



Less coding  
and more using (even  
though we love to code)



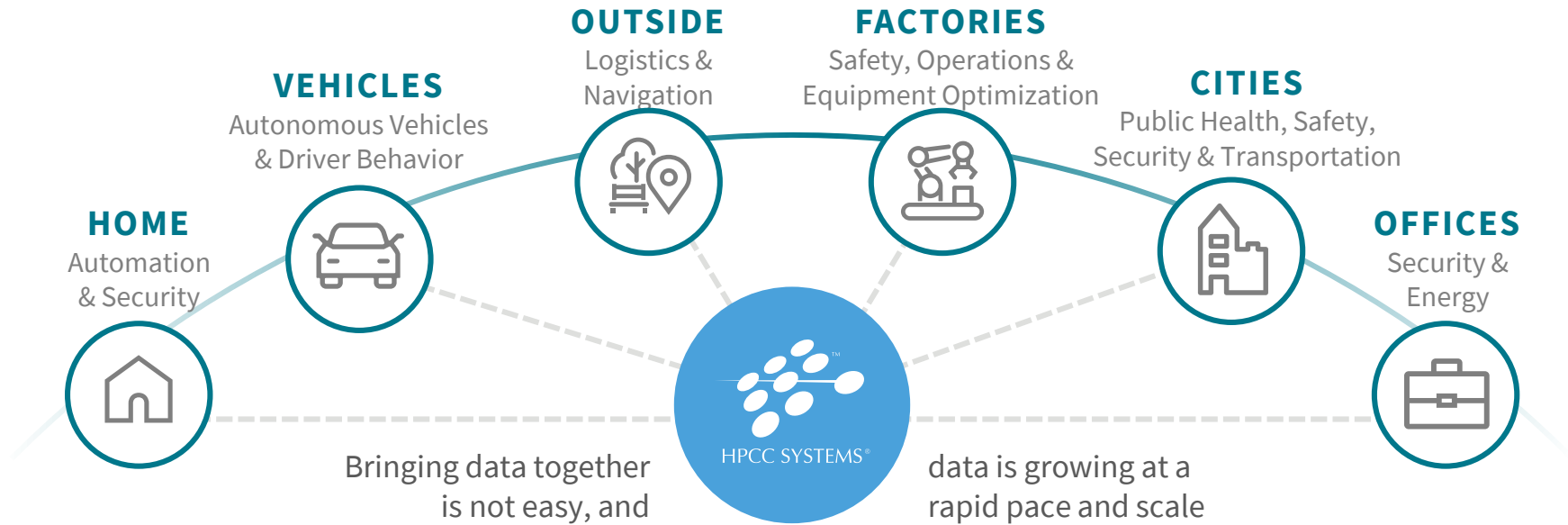
Aiming at being  
your one stop  
shop for all your  
data integration,  
querying and  
analytical needs

# A Brief History of HPCC!

## Why does HPCC Systems exist?

- ✓ It was NOT developed with the idea of selling the technology to anybody else!
- ✓ It was all created only to solve some of the data-handling problems that we encountered as we were developing our products.
- ✓ HPCC defined is a *distributed data parallel processing* platform.

# A platform purpose-built for high-speed data engineering



A processing platform is vital for bringing all your data together across all verticals

# HPCC Systems Evolution

2001



Original version  
of HPCC  
Systems  
released

2011



**Open source** Apache  
license and code  
release to GitHub

Exceeded market-  
leading performance  
benchmark achieved

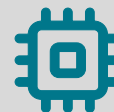
2012 – 16



Continuous  
**QUALITY-FOCUSED**  
improvements

Better support and  
training with improved  
integration — faster  
and easier to use

2017-2020



Improved processing  
architecture

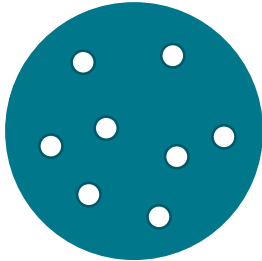
IoT enabled

More Bundles and ML  
Expansion!

# The Data Centric Approach

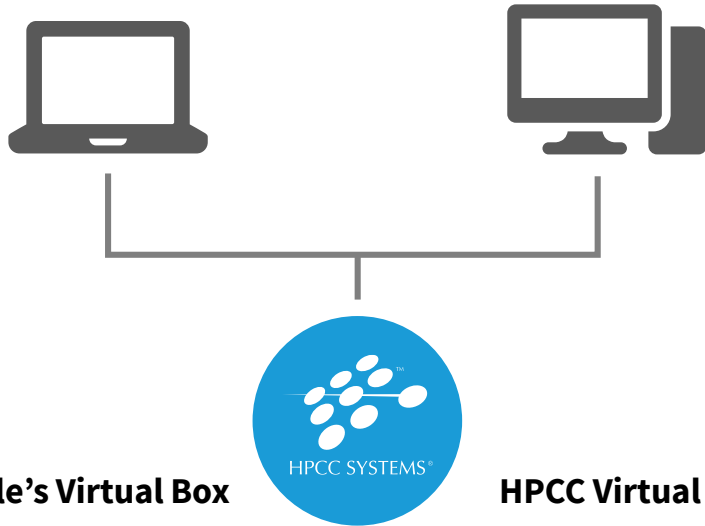
A single source of data is insufficient to overcome inaccuracies

Our platform is built on the premise of absorbing data from many data sources and transforming them to actionable smart data



# Scale from Small to Big

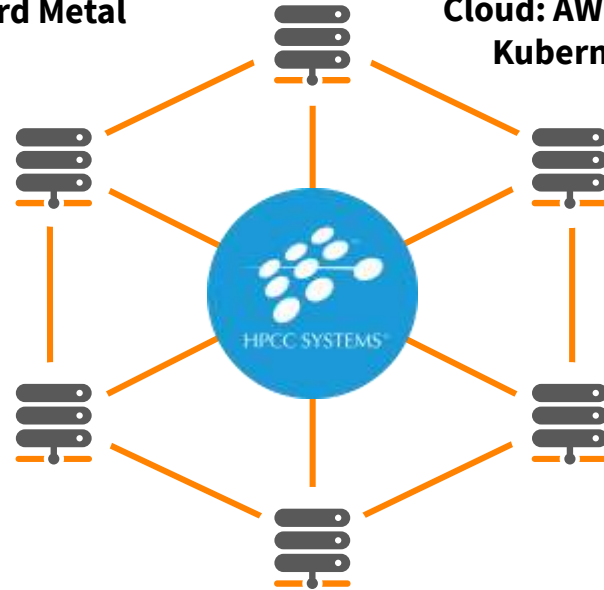
The stack can run on a single laptop or desktop.



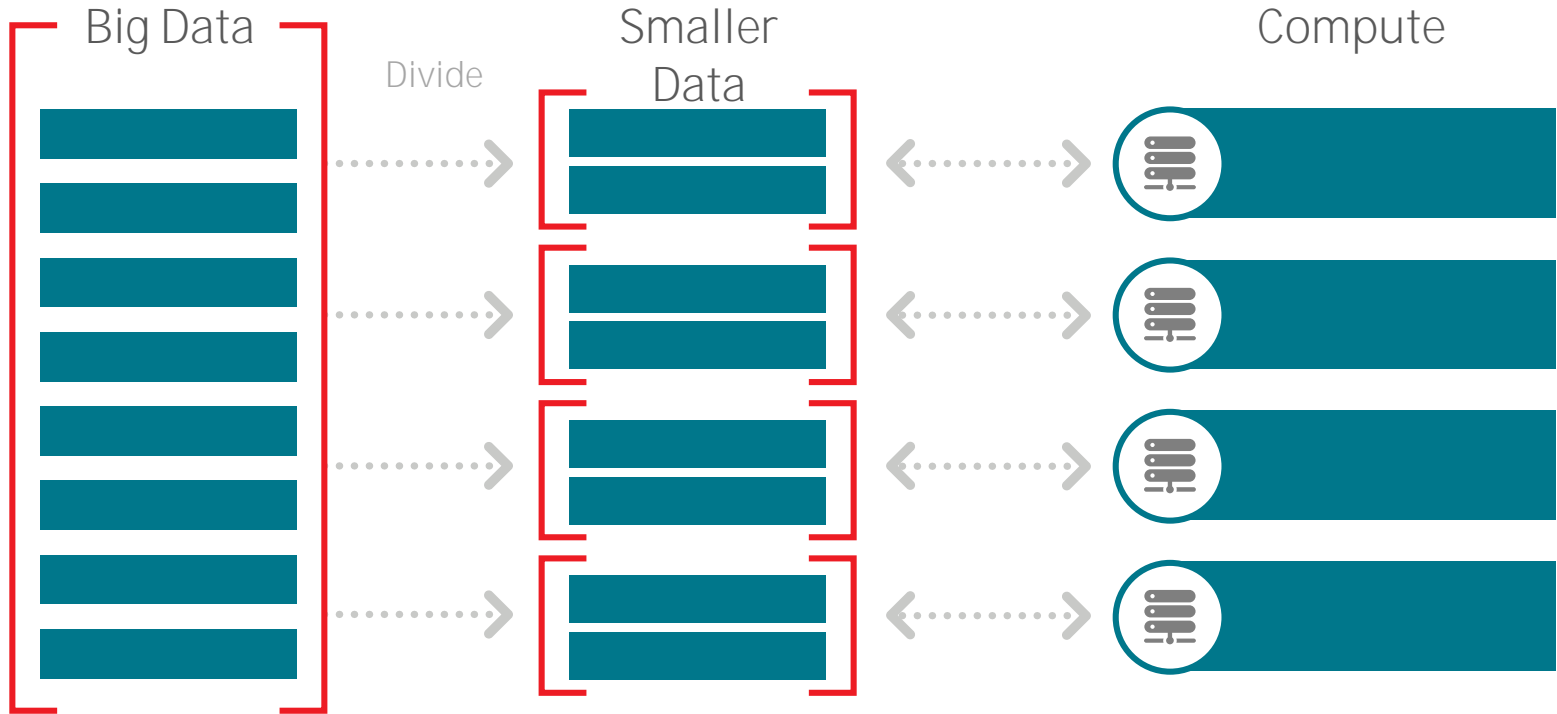
In more sophisticated cases, HPCC Systems run *clusters*, hundreds of servers working as a single processing entity, to transform and deliver big data.

**Hard Metal**

**Cloud: AWS/Azure  
Kubernetes**

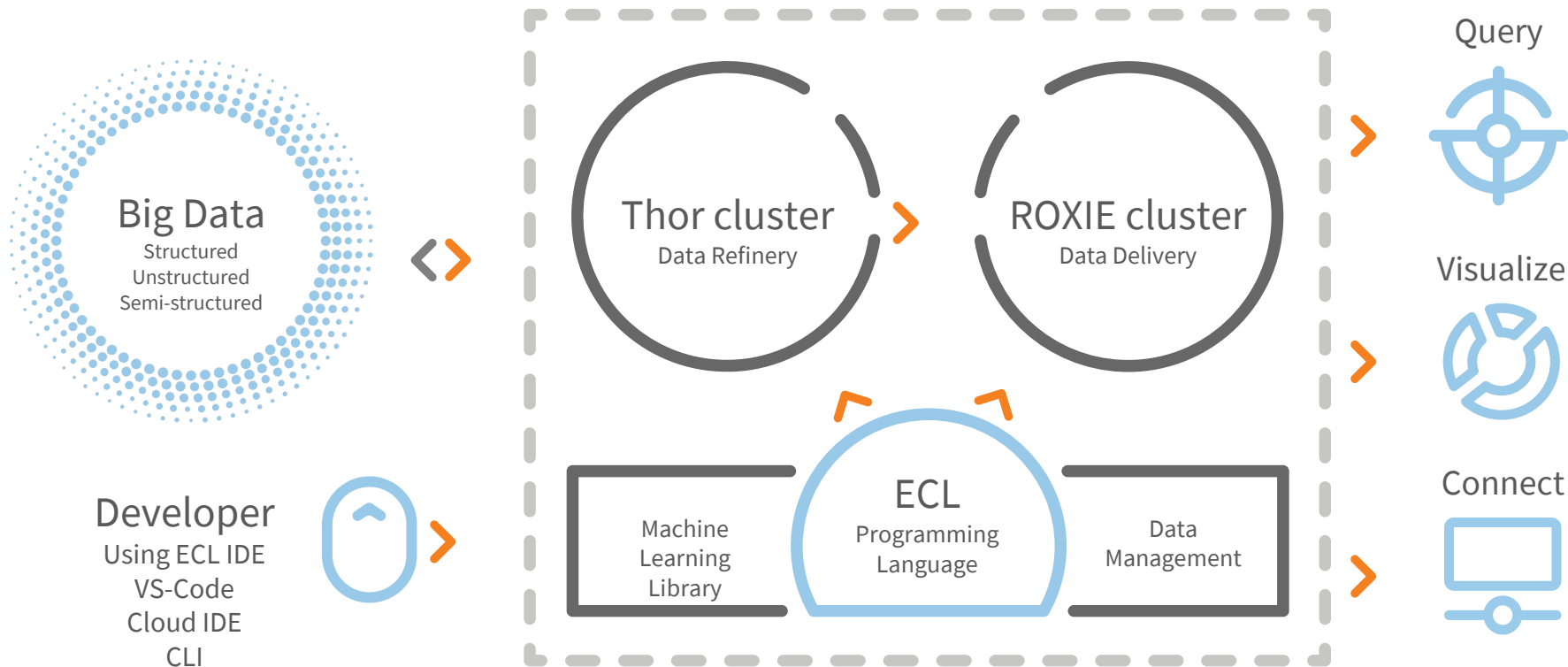


# Anatomy of a Big Data Processing System





# The HPCC Systems Components



# Technology — The Open Source Stack



## Thor: Data Refinery Cluster

Extraction, loading, cleansing, transforming, linking and indexing



## ROXIE: Data Delivery Engine

Rapid data delivery cluster with high-performance online query delivery for big data



## Data Management Tools

Data profiling, cleansing, snapshot data updates, consolidation, job scheduling and automation



## Machine Learning Library

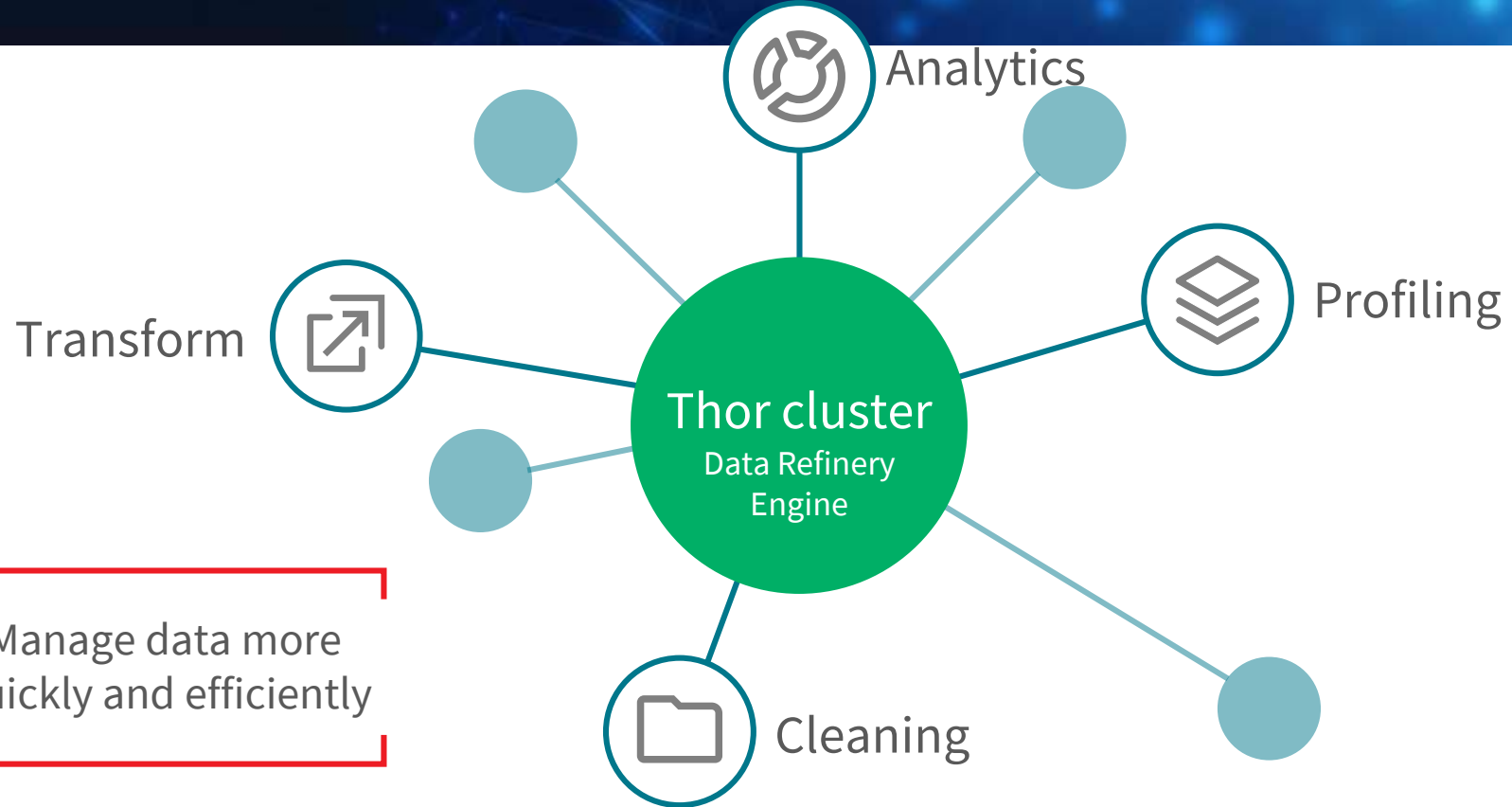
Linear regression, logistic regression, decision trees and random forests



## Connectivity & Third-Party Tools

New plugins to help integrate third party tools with the HPCC Systems platform

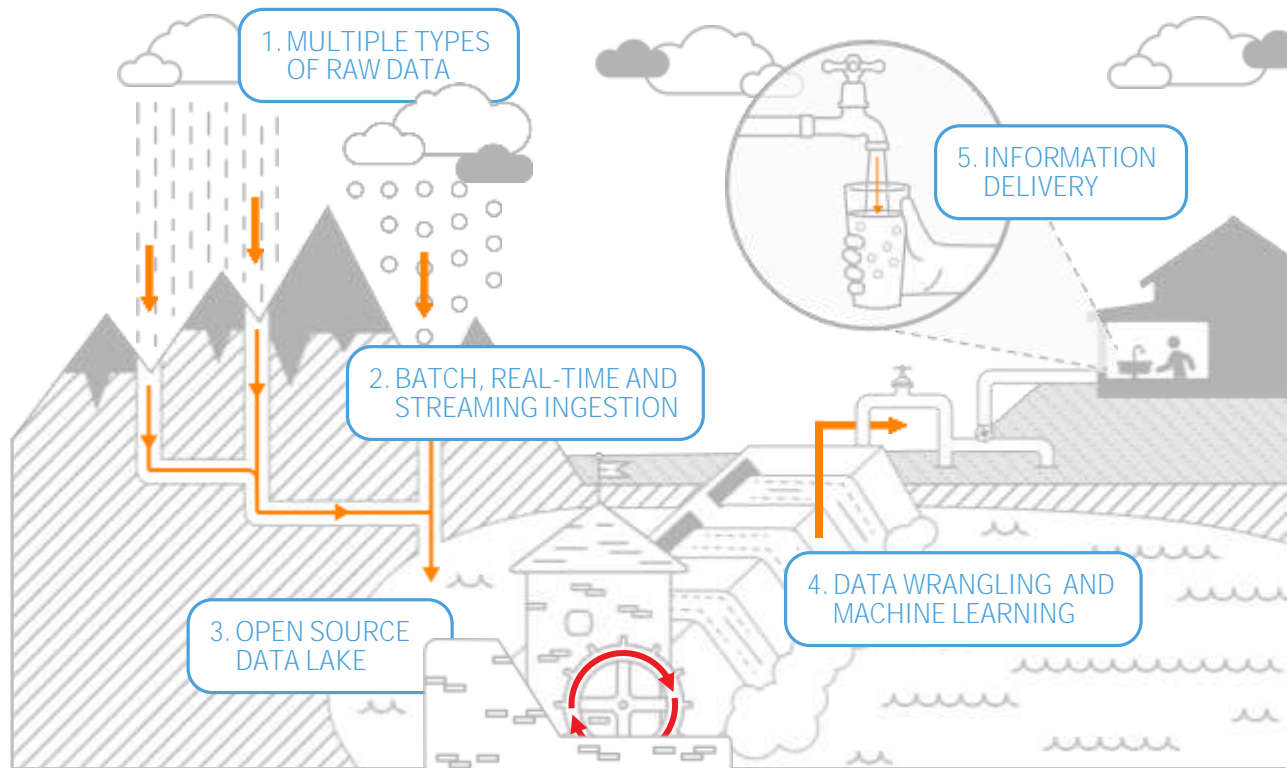
# THOR at a glance:



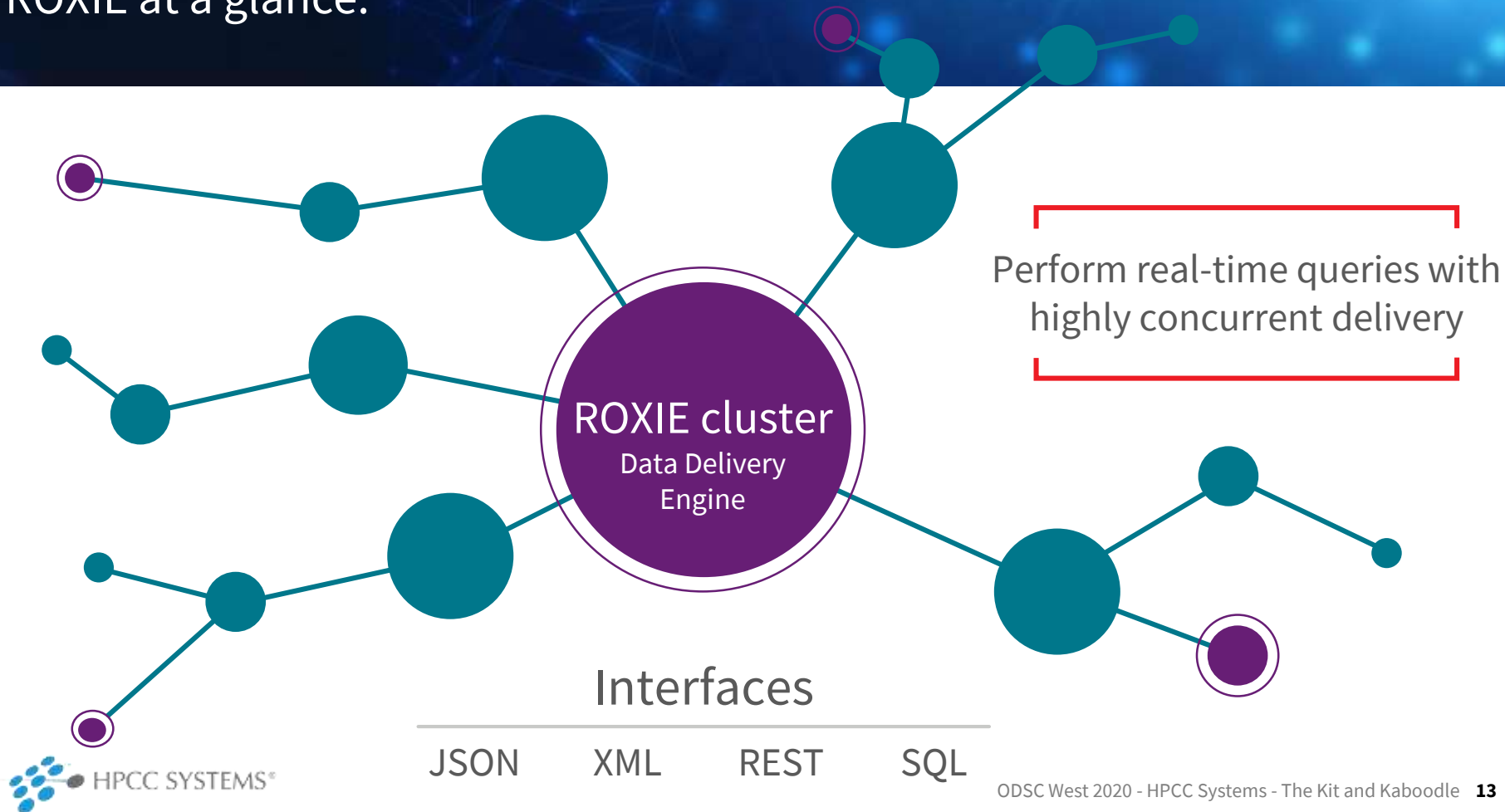
# Key aspects of our data lake solution

## The HPCC Systems advantage

- Open source data lake platform
- Batch, real-time and streaming data ingestion
- Built-in data enhancement and Machine Learning APIs
- Scalable to many petabytes of data
- Runs on commodity hardware and in the cloud
- Increased responsiveness to customers and stakeholders



# ROXIE at a glance:



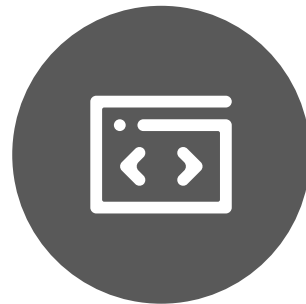
# An Introduction to ECL

ECL

Enterprise Control  
Language



```
IMPORT $, STD, ML;
EXPORT Func(UNSIGNED C, UNSIGNED2 Dist, UNSIGNED size, STRING Fld, REAL Parm1=0, REAL Parm2=0, REAL Parm3=0) := MODULE
  SHARED Node := STD.system.ThorLib.Node()+1;
  SHARED PersistPrefix := $.Params.PersistPrefix;
  SHARED TotalRecs := $.Params.RecCnt*CLUSTERSIZE;
  SHARED UIDval := IF(C=1, node, node + ((C-1)*CLUSTERSIZE));
  SHARED BOOLEAN IsRandFile := $.Params.Randomness = $.ut.RandomSrc.File;
  SHARED Normal := FUNCTION
    Thisdist := IF(Parm3=0,
      ML.Distribution.Normal(Parm1, Parm2),
      ML.Distribution.Normal(Parm1, Parm2, Parm3));
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1); PERSIST(PersistPrefix + 'NormalDistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
  SHARED Normal2 := FUNCTION
    Thisdist := IF(Parm3=0,
      ML.Distribution.Normal2(Parm1, Parm2),
      ML.Distribution.Normal2(Parm1, Parm2, Parm3));
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1); PERSIST(PersistPrefix + 'Normal2DistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
  SHARED Uniform := FUNCTION
    Thisdist := IF(Parm3=0,
      ML.Distribution.Uniform(Parm1, Parm2),
      ML.Distribution.Uniform(Parm1, Parm2, Parm3));
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1); PERSIST(PersistPrefix + 'UniformDistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
  SHARED StudentT := FUNCTION
    Thisdist := ML.Distribution.StudentT(Parm1, Parm2);
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1); PERSIST(PersistPrefix + 'StudentTDistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
END;
```



- Transparent and implicitly parallel programming language
- Both powerful and flexible

- Optimized for data-intensive operations, declarative, non-procedural and dataflow oriented
- Uses intuitive syntax which is modular, reusable, extensible and highly productive

How to do it



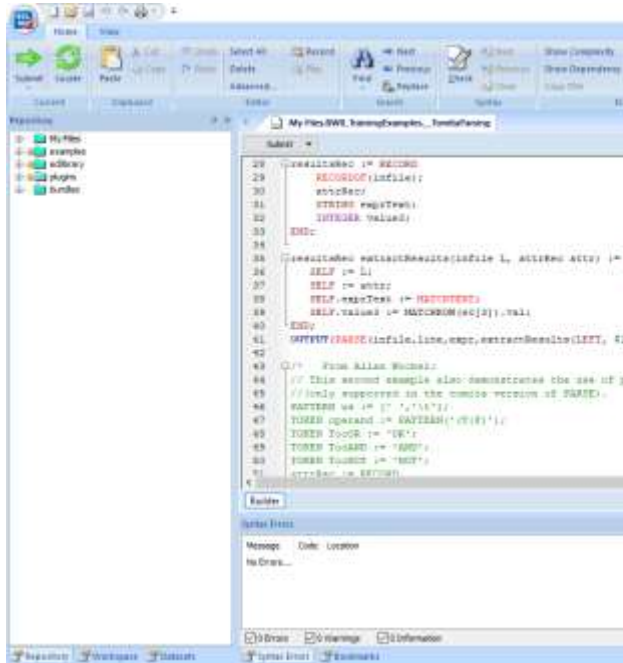
vs.



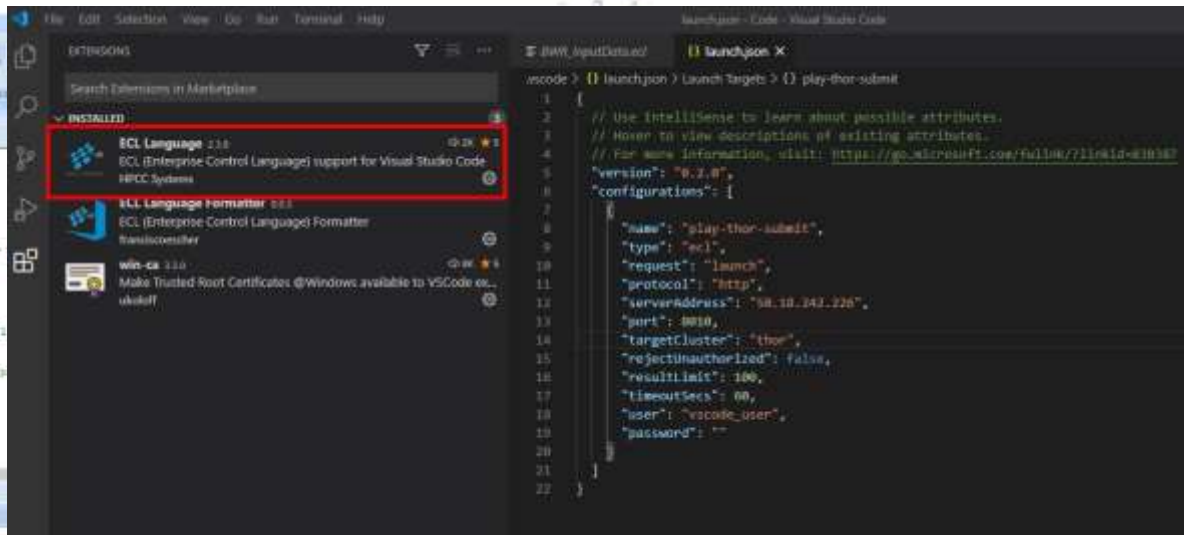
What to do

# Integrated Development Environments

✓ ECL IDE (Win)



✓ VSCode (Ux/MacOS)



✓ And CLI too! ECL.EXE

# ECL IDE Features:

A full-featured GUI for ECL development providing access to the ECL repository and many of the ECL Watch capabilities.

Uses various ESP services via SOAP.



Provides the easiest way to create:

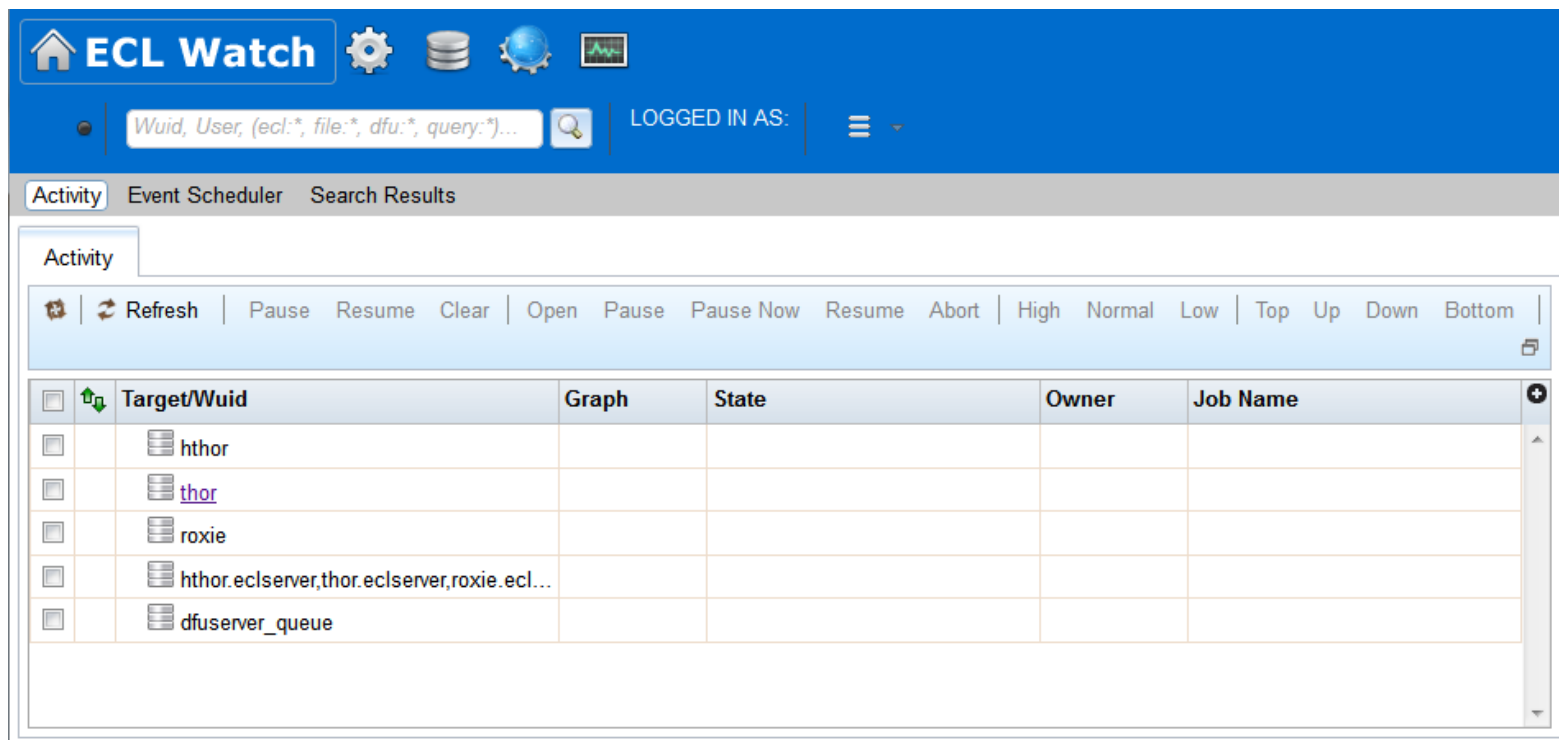
Queries into your data.

ECL Definitions to build your queries which:






- Are created by coding an expression that defines how some calculation or record set derivation is to be done.
- Once defined, can be used in succeeding ECL definitions.



# The ECL Watch



The screenshot displays the ECL Watch web application interface. At the top, there is a blue header bar with the "ECL Watch" logo, navigation icons (gear, database, globe, line graph), and a search bar containing the text "Wuid, User, (ecl:\*, file:\*, dfu:\*, query:\*)...". To the right of the search bar, it says "LOGGED IN AS:" followed by a menu icon. Below the header, a grey bar contains tabs for "Activity", "Event Scheduler", and "Search Results". The "Activity" tab is selected, showing a sub-header "Activity" and a toolbar with buttons: Refresh, Pause, Resume, Clear, Open, Pause, Pause Now, Resume, Abort, and a filter dropdown set to "High". Below the toolbar is a table with the following columns: Target/Wuid, Graph, State, Owner, and Job Name. The table contains five rows of data, each with a checkbox in the first column.

<input type="checkbox"/>	Target/Wuid	Graph	State	Owner	Job Name
<input type="checkbox"/>	 hthor				
<input type="checkbox"/>	 <a href="#">thor</a>				
<input type="checkbox"/>	 roxie				
<input type="checkbox"/>	 hthor.eclserver,thor.eclserver,roxie.ecl...				
<input type="checkbox"/>	 dfusever_queue				

# ECL Watch Features:

A web-based query execution, monitoring and file management interface. It can be accessed via ECL IDE or a web browser.

ECL Watch allows you to:

See information about active workunits.

Monitor cluster activity.

Browse through previously submitted WUs:

- See a visual representation of the data flow within the WU.
- Complete with statistics which are updated as the job progresses.

Search through files and see information including:

- Record counts and layouts.
- Sample records.
- The status of all system servers whether they are in clusters or not.

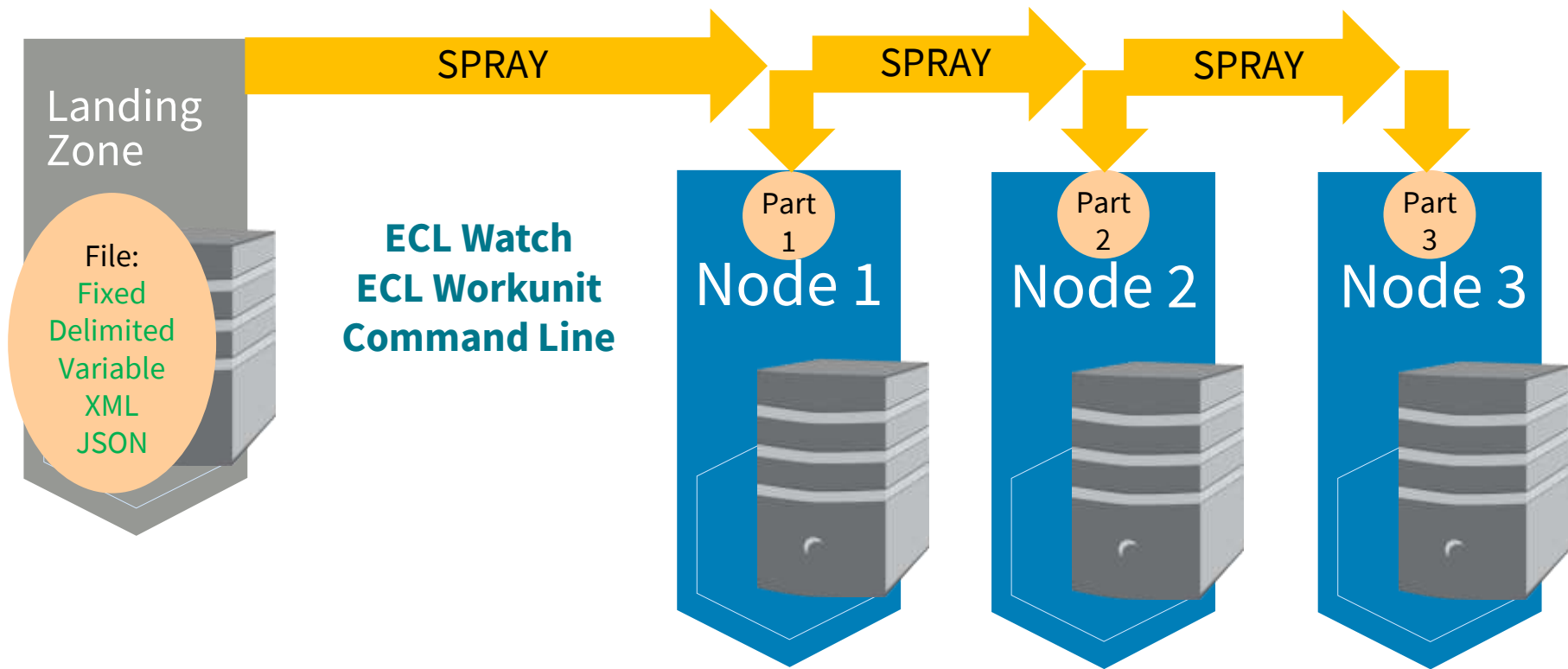
View log files.

Start and stop processes.



# SPRAY Operation

## HPCC Cluster



# HPCC Systems (Small to Big Data) ETL



Integrate >



Profile >



Clean >



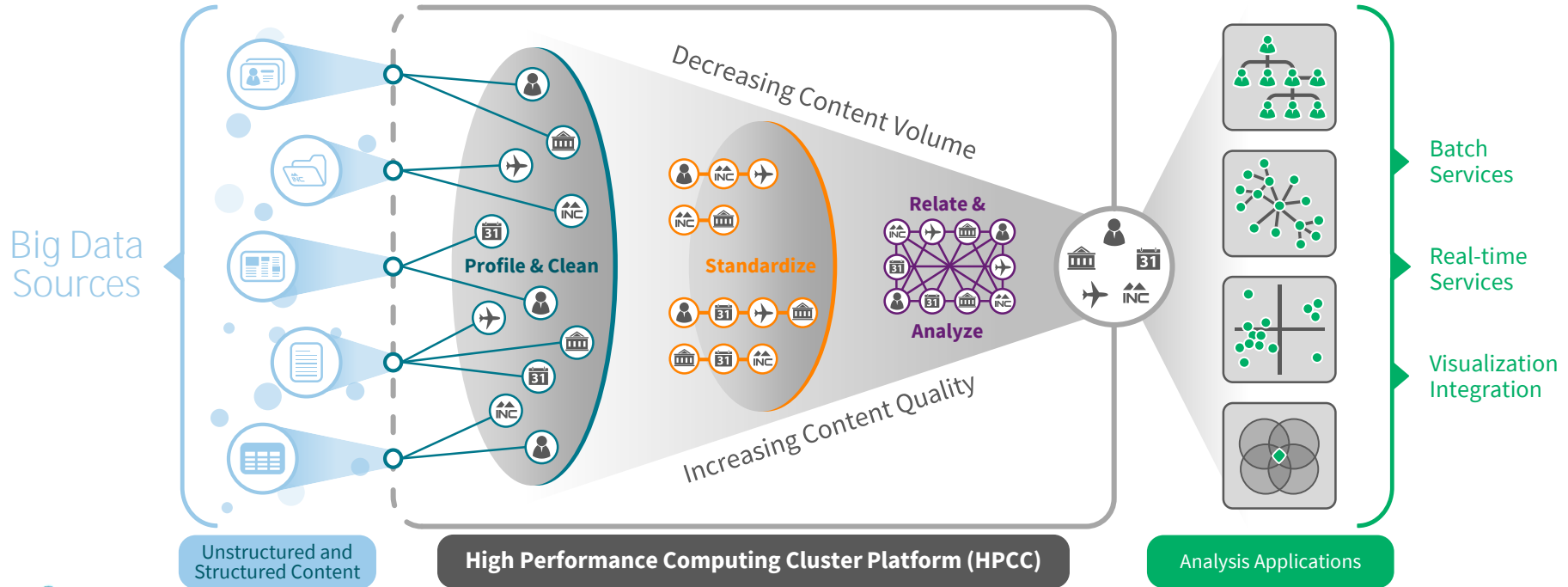
Standardize >



Analyze >



Deliver





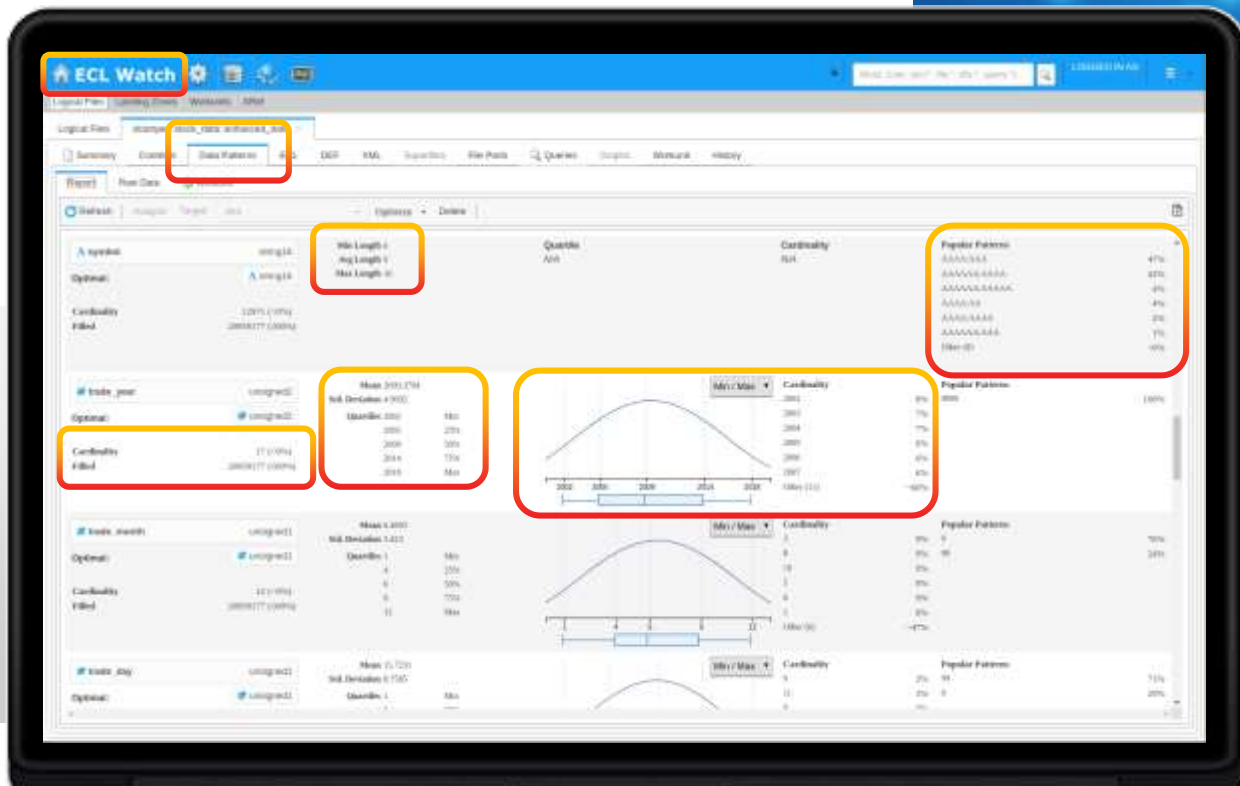
# Libraries and Plugins

# Integrated Data Profiling



Built-in data profiling exposes field-level details

- ▶ Fill rates and cardinality details
- ▶ Numeric range detail, including quartiles
- ▶ Textual patterns highlight common and rare formats



# It's a Machine Learning world

## Classical Machine Learning



### Unsupervised

Clustering  
DBSCAN  
K-Means

Pattern Search  
Text Vectors

Levenshtein Deletion  
Neighborhood

Dimension Reduction  
PCA



### Supervised

Classification  
SVM  
Decision Trees  
Logistic Regression  
Classification Forest  
Latent Dirichlet Allocation  
(Topic Modeling)

Regression  
Linear Regression  
Regression Forest



### Neural Nets & Deep Learning

Autoencoders

Convolutional  
Neural Networks

Recurrent Neural  
Networks

Perceptrons



### Ensemble Methods

Random Forest

Gradient Boosted  
Forest

Gradient Boosted  
Trees

# ECL: Modules, Bundles, & Plugins

**Machine Learning**  
Bundle

**Databases**

MySQL Import  
WsSQL

# ECL

**Visualization**

Bundle  
Cell Formatter

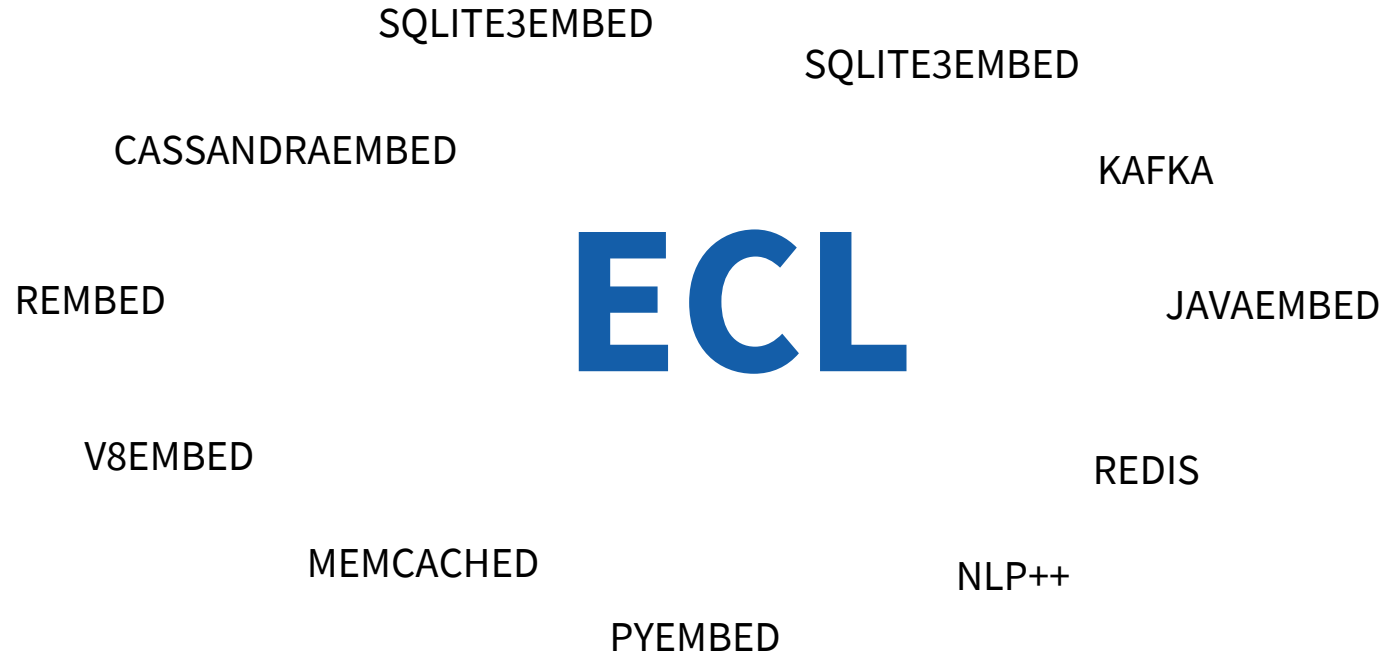
**Plugins** – see next slide

**Text Processing**

Trigram  
Web Log Analytic Module  
Prefix Tree



# ECL: Plugins



# Library and Datastore PlugIns

Plugin interface with ECL Watch

Many Built ins (Debug, File Services)

Audit and Logging

dMetaphone (double metaphone)

Apache Kafka

Security Manager

Redis

Memcached

Spark

Eclipse IDE

ECL Data Integration Plugin for Pentaho

R Integration

ECL Extension for VS Code

JDBC Driver

Java API

ODBC Driver

## References:

<https://github.com/hpcc-systems/HPCC-Platform/tree/master/plugins>

# Embedded Language Plugins

C++

R Integration

Couchbase

- Java

- JavaScript

- MySQL

- Python/Python 3

- SQLite3

- Cassandra

- AWS SQS (Simple Queue Service)



```
1  IMPORT java;
2
3  INTEGER add1(INTEGER val) := IMPORT(java, 'JavaCat.add1:(I)I');
4  STRING add2(STRING val) := IMPORT(java, 'JavaCat.add2:(Ljava/lang/String;)Ljava/lang/String;');
5  STRING add3(VARSTRING val) := IMPORT(java, 'JavaCat.add2:(Ljava/lang/String;)Ljava/lang/String;');
6  UTF8 add4(UTF8 val) := IMPORT(java, 'JavaCat.add2:(Ljava/lang/String;)Ljava/lang/String;');
7  UNICODE add5(UNICODE val) := IMPORT(java, 'JavaCat.add2:(Ljava/lang/String;)Ljava/lang/String;');
8  STRING addChar(STRING c) := IMPORT(java, 'JavaCat.addChar:(C)C');
9  STRING cat(STRING s1, STRING s2) := IMPORT(java, 'JavaCat.cat:(Ljava/lang/String;)Ljava/lang/String;');
10
11
12  add1(10);
13  add2('Hello');
14  add3('World');
15  add4('Leavesades');
16  add5('你好世界');
17  addChar('A');
```

<https://hpccsystems.com/resources/blog/lchapman/using-your-favorite-language-or-data-source-hpcc-systems>

<https://hpccsystems.com/resources/blog/richardkchapman/projecting-fields-embeds>

Use and abuse of the EMBED feature: <https://hpccsystems.com/bb/viewtopic.php?f=41&t=1509>

# Embedded Language Plugins

CODE: SELECT ALL

```
IMPORT python;
SET OF STRING split(STRING text) := EMBED(python)
    return text.split()
ENDEMBED;
split('Once upon a time');
```

CODE: SELECT ALL

```
IMPORT python;
r := RECORD
    STRING word;
    UTF8 tags;
END;
DATASET(R) tag(STRING text) := IMPORT(python, './ex2.tag');
tag('Once upon a time there was a boy called Richard');
```

CODE: SELECT ALL

```
IMPORT java;
STRING jcat(STRING a, STRING b) :=
    IMPORT(java,
        'JavaCat.cat:(Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;' :
    classpath('/opt/HPCCSystems/classes'));

jcat('Hello ', 'world!');
```

CODE: SELECT ALL

```
import nltk
tokenizer = None
tagger = None
def init_nltk():
    global tokenizer, tagger
    tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+|[\^\w\s]+' )
    tagger = nltk.UnigramTagger(nltk.corpus.brown.tagged_sents())
def tag(text):
    global tokenizer, tagger
    if not tokenizer:
        init_nltk()
    tokenized = tokenizer.tokenize(text)
    return tagger.tag(tokenized)
```

CODE: SELECT ALL

```
IMPORT MySQL;
stringrec := RECORD
    string name
END;
sqlrec := RECORD
    string ssn;
    string address;
END;
DATASET(sqlrec) MySQLJoin(dataset(stringrec) inrecs) := EMBED(mysql)
    SELECT * from tbl1 where name = ?;
ENDEMBED;
MySQLJoin(indata);
```

# Summary

Discover HPCC Systems, an end-to-end data lake management solution. HPCC Systems is a mature platform that has been heavily used in commercial applications for almost two decades, predating the development of Hadoop. Created by LexisNexis Risk Solutions, an innovative pioneer in big data processing, and open source for nearly a decade now, HPCC Systems features a vibrant development community that continues to push the boundaries of big data.

This powerful, versatile platform makes it easier for developers to see the data they're working with and manipulate it as needed. Flexible information delivery makes it easier for your clients to query and find the data they need — and it runs analysis and queries faster than other platforms such as SQL or Hadoop.

Indeed, it is the Kit and Kaboodle for your Big Data Solutions!

Thank you!

Want more? Remote HPCC/ECL Workshop today at 2 P.M. (PDT)/5 P.M. (EDT)

**Title:**

**Data Visualization, NLP, and ML made easy with HPCC Systems ECL  
A 3-hour workshop**

**Abstract:**

<https://odsc.com/speakers/remote-hpcc-systems-ecl-training/>

**Class Materials:**

<https://github.com/hpccsystems-solutions-lab/hpcc-systems-BR>