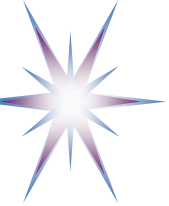


Cloud Based Big Data Infrastructure: Architectural Components and Automated Provisioning

The 2016 International Conference on High Performance Computing & Simulation
(HPCS2016)
18-22 July 2016, Innsbruck, Austria

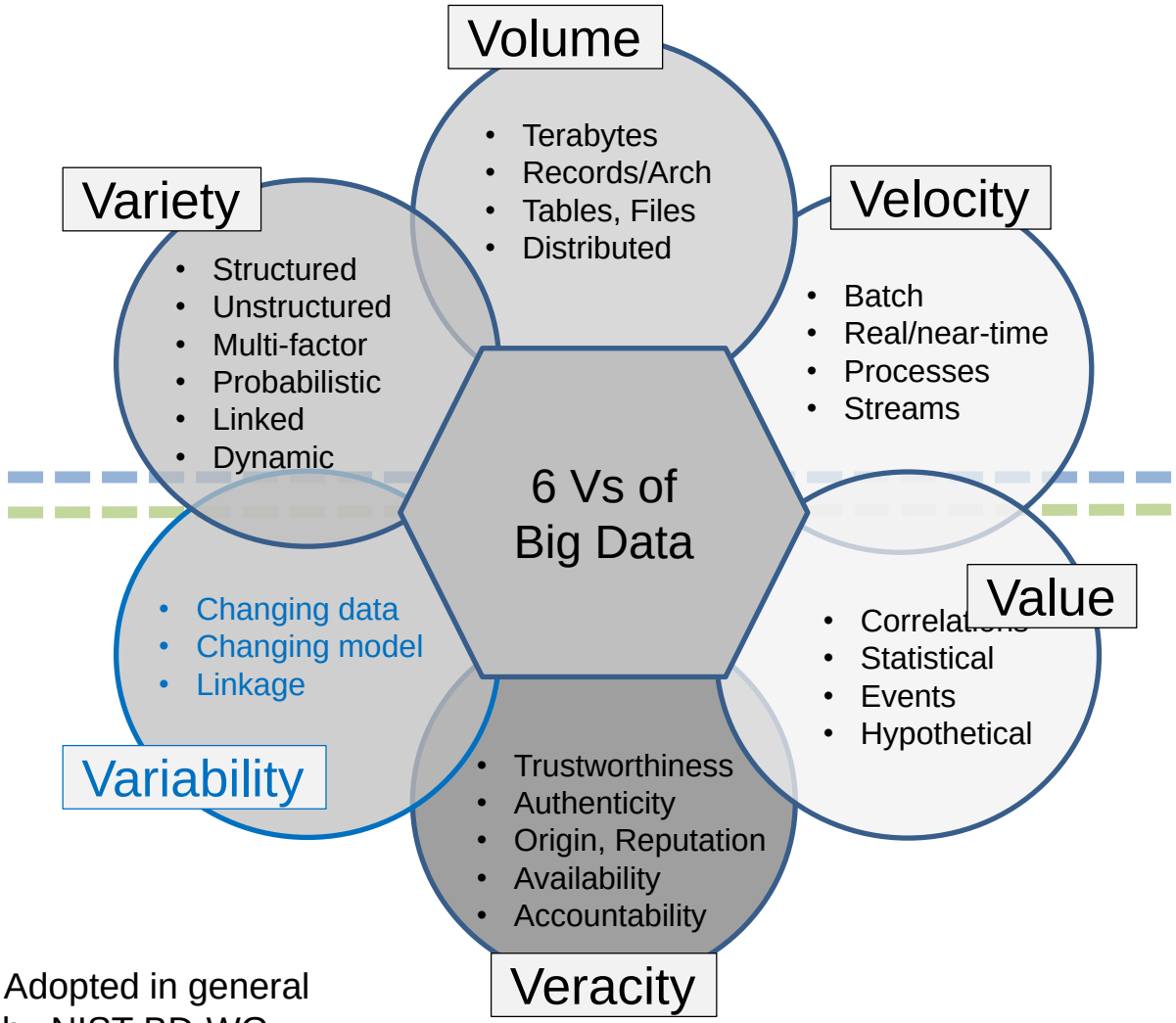
Yuri Demchenko, University of Amsterdam

Outline

- Big Data and new concepts
  - Big Data definition and Big Data Architecture Framework (BDAF)
  - Data driven vs data intensive vs data centric and Data Science
- Cloud Computing as a platform of choice for Big Data applications
  - Big Data Stack and Cloud platforms for Big Data
  - Big Data Infrastructure provisioning automation
- CYCLONE project and use cases for cloud based scientific applications automation
- Slipstream and cloud automation tools
  - Slipstream recipe example
- Discussion

The research leading to these results has received funding from the Horizon2020 project CYCLONE

Big Data definition revisited: 6 V's of Big Data



Adopted in general by NIST BD-WG

Generic Big Data Properties

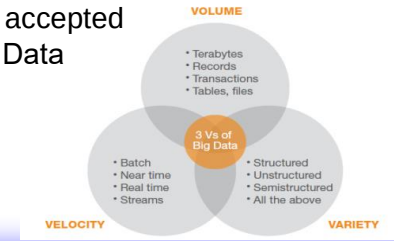
- Volume
- Variety
- Velocity

Acquired Properties (after entering system)

- Value
- Veracity
- Variability

Commonly accepted 3V's of Big Data

- Volume
- Velocity
- Variety



Big Data definition: From 6V to 5 Components

(1) Big Data Properties: 6V
- Volume, Variety, Velocity
- Value, Veracity, Variability

(2) New Data Models
- Data linking, provenance and referral integrity
- **Data Lifecycle and Variability/Evolution**

(3) New Analytics
- Real-time/streaming analytics, machine learning and iterative analytics

(4) New Infrastructure and Tools
- High performance Computing, Storage, Network
- Heterogeneous multi-provider services integration
- New Data Centric (multi-stakeholder) service models
- New Data Centric security models for trusted infrastructure and data processing and storage

(5) Source and Target
- High velocity/speed data capture from variety of sensors and data sources
- Data delivery to different visualisation and actionable systems and consumers
- Full digitised input and output, (ubiquitous) sensor networks, full digital control

Moving to Data-Centric Models and Technologies

- Current IT and communication technologies are host based or host centric
  - Any communication or processing are bound to host/computer that runs software
  - Especially in security: all security models are host/client based
- Big Data requires new data-centric models
  - Data location, search, access
  - Data integrity and identification
  - Data lifecycle and variability
  - Data centric (declarative) programming models
  - Data aware infrastructure to support new data formats and data centric programming models
- Data centric security and access control

Moving to Data-Centric Models and Technologies

RDA developments (2016):
Data become Infrastructure themselves

- PID, Metadata Registries, Data models and formats
- Data Factories

- Big Data requires new data-centric models
  - Data location, search, access
  - Data integrity and identification
  - Data lifecycle and variability
  - Data centric (declarative) programming models
  - Data aware infrastructure to support new data formats and data centric programming models
- Data centric security and access control

NIST Big Data Working Group (NBD-WG) and ISO/IEC JTC1 Study Group on Big Data (SGBD)

- NIST Big Data Working Group (NBD-WG) is leading the development of the Big Data Technology Roadmap - http://bigdatawg.nist.gov/home.php
  - Built on experience of developing the Cloud Computing standards fully accepted by industry
- Set of documents published in September 2015 as NIST Special Publication NIST SP 1500: NIST Big Data Interoperability Framework (NBDIF) http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-1.pdf

  Volume 1: NIST Big Data Definitions

  Volume 2: NIST Big Data Taxonomies

  Volume 3: NIST Big Data Use Case & Requirements

  Volume 4: NIST Big Data Security and Privacy Requirements

  Volume 5: NIST Big Data Architectures White Paper Survey

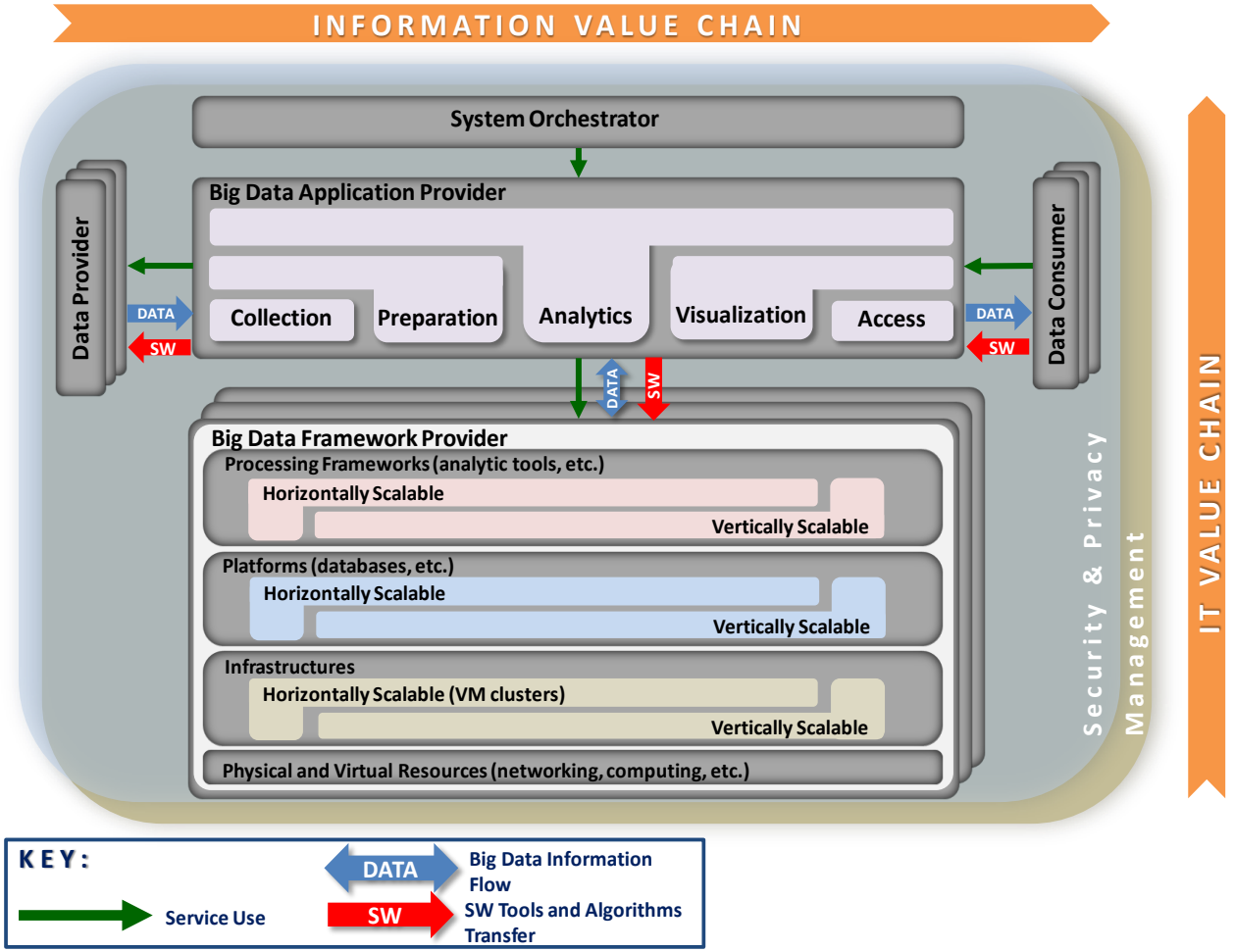
  Volume 6: NIST Big Data Reference Architecture

  Volume 7: NIST Big Data Technology Roadmap

- NBD-WG defined 3 main components of the new technology:
  - Big Data Paradigm
  - Big Data Science and Data Scientist as a new profession
  - Big Data Architecture

The **Big Data Paradigm** consists of the distribution of data systems across horizontally-coupled independent resources to achieve the scalability needed for the efficient processing of extensive datasets.

NIST Big Data Reference Architecture



Main components of the Big Data ecosystem

- Data Provider
- Big Data Applications Provider
- Big Data Framework Provider
- Data Consumer
- Service Orchestrator

Big Data Lifecycle and Applications Provider activities

- Collection
- Preparation
- Analysis and Analytics
- Visualization
- Access

Big Data Ecosystem includes all components that are involved into Big Data production, processing, delivery, and consuming

[ref] Volume 6: NIST Big Data Reference Architecture. http://bigdatawg.nist.gov/V1_output_docs.php

Big Data Architecture Framework (BDAF) by UvA

(1) Data Models, Structures, Types
- Data formats, non/relational, file systems, etc.

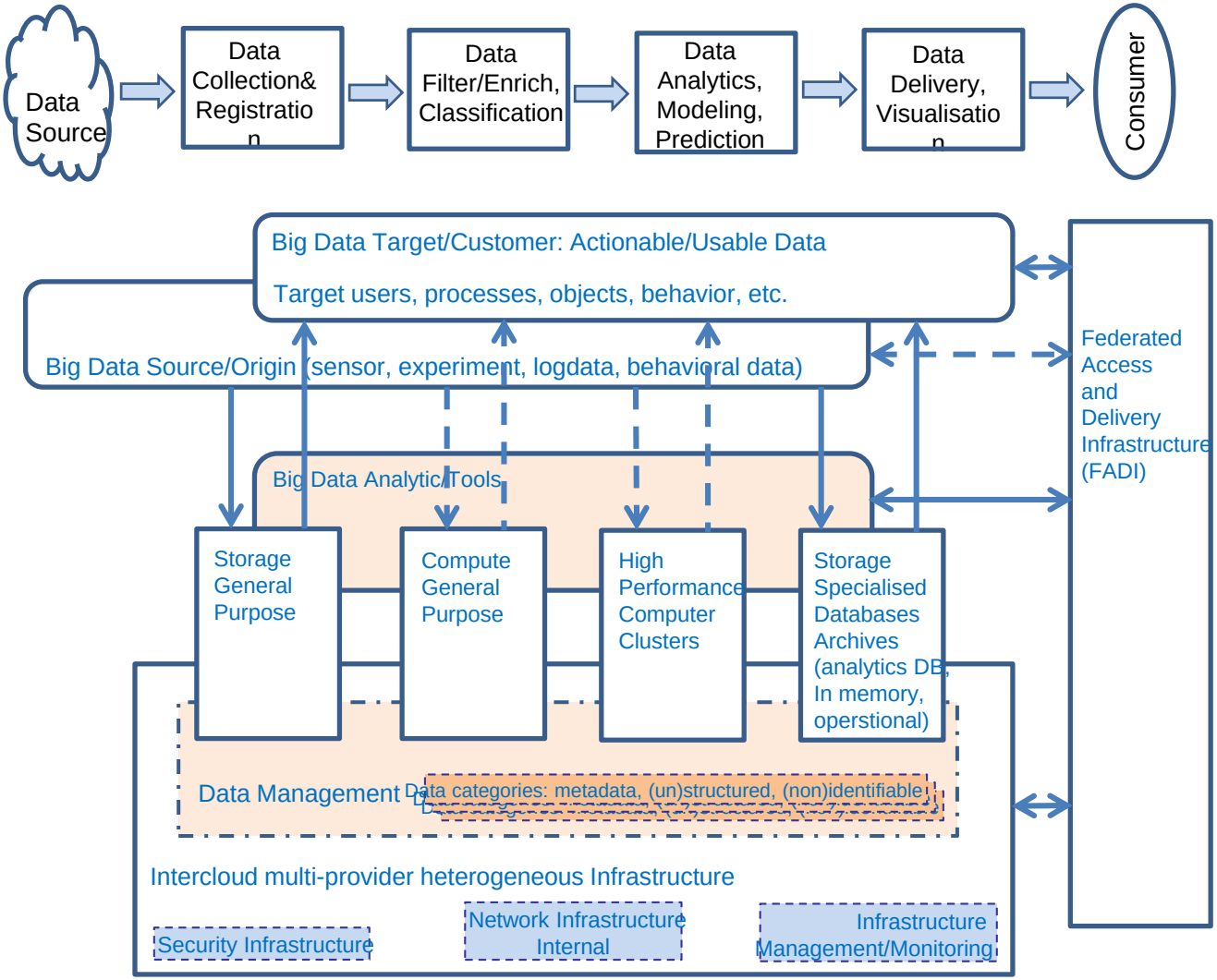
(2) Big Data Management
- Big Data Lifecycle (Management) Model
  - Big Data transformation/staging
- Provenance, Curation, Archiving

(3) Big Data Analytics and Tools
- Big Data Applications
  - Target use, presentation, visualisation

(4) Big Data Infrastructure (BDI)
- Storage, Compute, (High Performance Computing,) Network
- Sensor network, target/actionable devices
- Big Data Operational support

(5) Big Data Security
- Data security in-rest, in-move, trusted processing environments

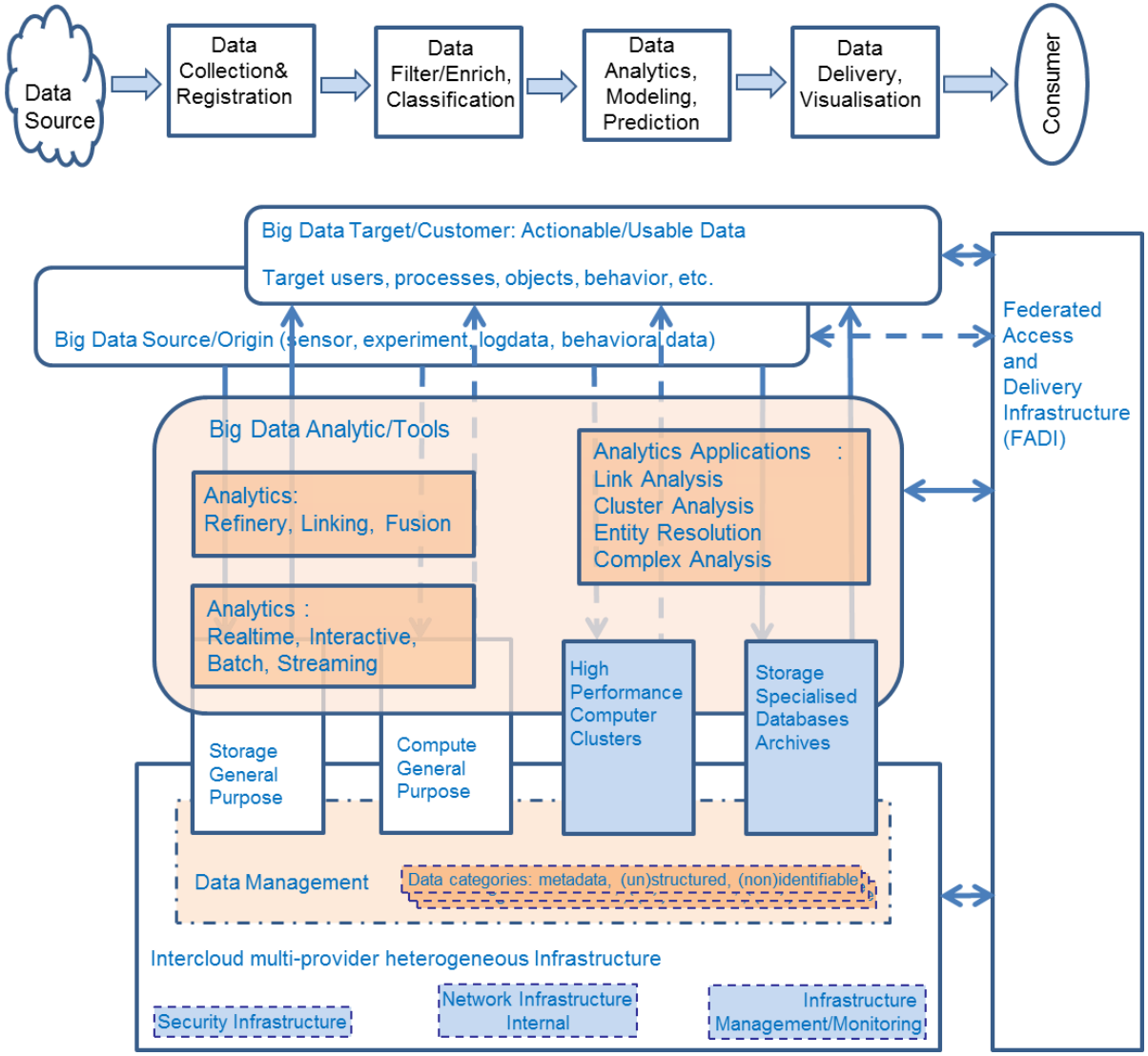
Big Data Ecosystem: General BD Infrastructure Data Transformation, Data Management



Big Data Infrastructure

- Heterogeneous multi-provider inter-cloud infrastructure
- Data management infrastructure
- Collaborative Environment (user/groups managements)
- Advanced high performance (programmable) network
- Security infrastructure

Big Data Infrastructure and Analytics Tools



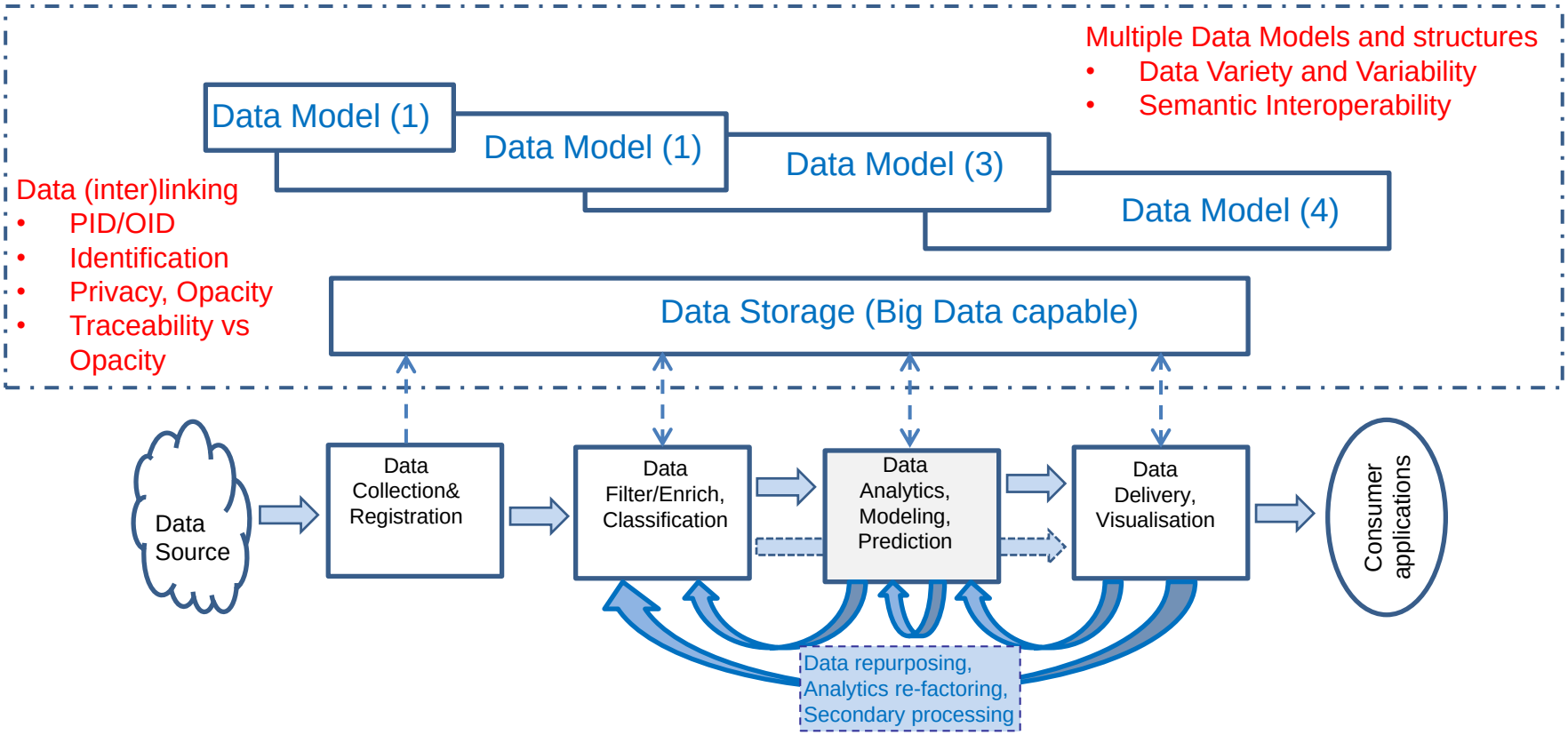
Big Data Infrastructure

- Heterogeneous multi-provider inter-cloud infrastructure
- Data management infrastructure
- Collaborative Environment
- Advanced high performance (programmable) network
- Security infrastructure
- Federated Access and Delivery Infrastructure (FADI)

Big Data Analytics Infrastructure/Tools

- High Performance Computer Clusters (HPCC)
- Big Data storage and databases SQL and NoSQL
- Analytics/processing: Real-time, Interactive, Batch, Streaming
- Big Data Analytics tools and applications

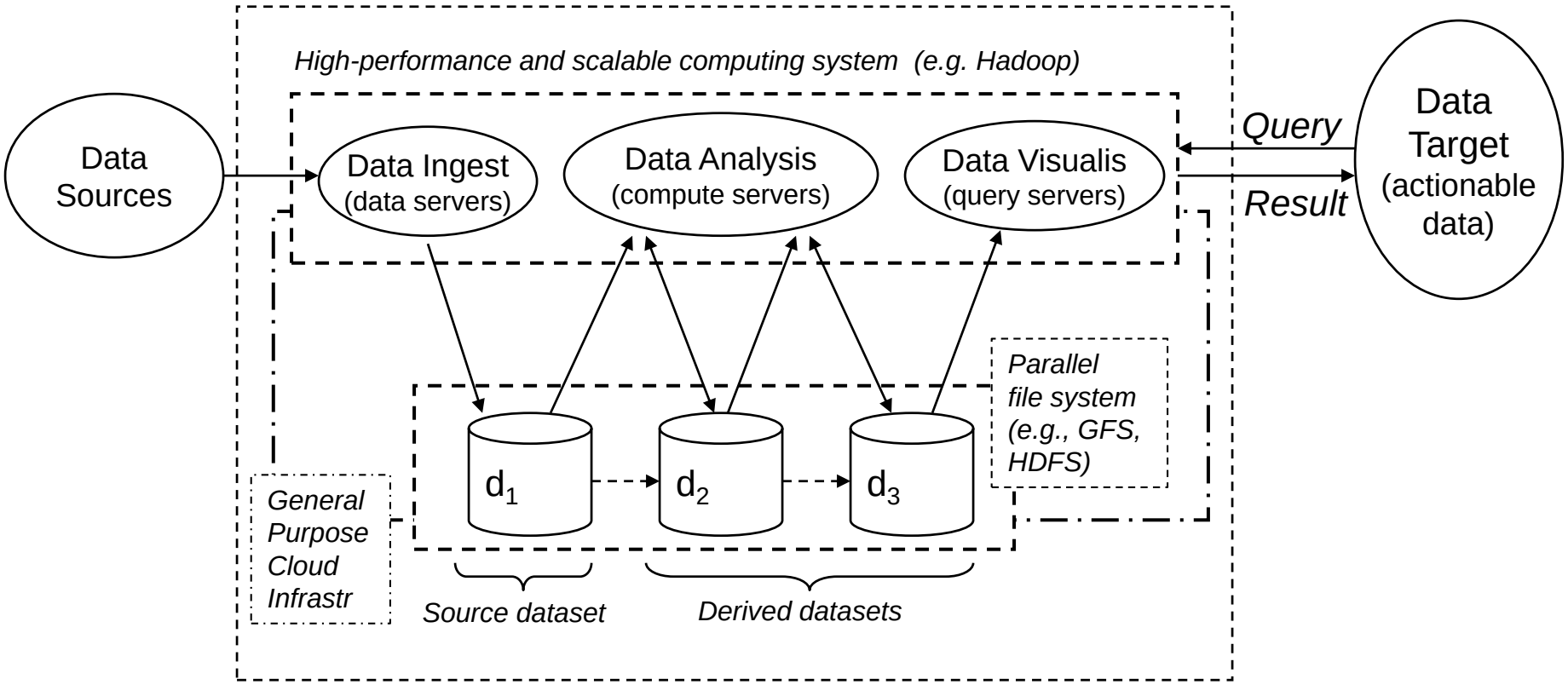
Data Lifecycle/Transformation Model



- Data Model changes along data lifecycle or evolution
- Data provenance is a discipline to track all data transformations along lifecycle

- Identifying and linking data
  - Persistent data/object identifiers (PID/OID)
  - Traceability vs Opacity
  - Referral integrity

Cloud Based Big Data Services



Characteristics:
Massive data and computation on cloud, small queries and results

Examples:
Search, scene completion service, log processing

Big Data Stack components and technologies

The major structural components of the Big Data stack are grouped around the main stages of data transformation

- **Data ingest:** Ingestion will transform, normalize, distribute and integrate to one or more of the Analytic or Decision Support engines; ingest can be done via ingest API or connecting existing queues that can be effectively used for handles partitioning, replication, prioritisation and ordering of data
- **Data processing:** Use one or more analytics or decision support engines to accomplish specific task related to data processing workflow; using batch data processing, streaming analytics, or real-time decision support
- **Data Export:** Export will transform, normalize, distribute and integrate output data to one or more Data Warehouse or Storage platforms;
- **Back-end data management, reporting, visualization:** will support data storage and historical analysis; OLAP platforms/engines will support data acquisition and further use for Business Intelligence and historical analysis.

Big Data Stack

Hook into an existing **queue** to get copy / subset of data. Queue handles partitioning, replication, and ordering of data, can manage backpressure from slower downstream components

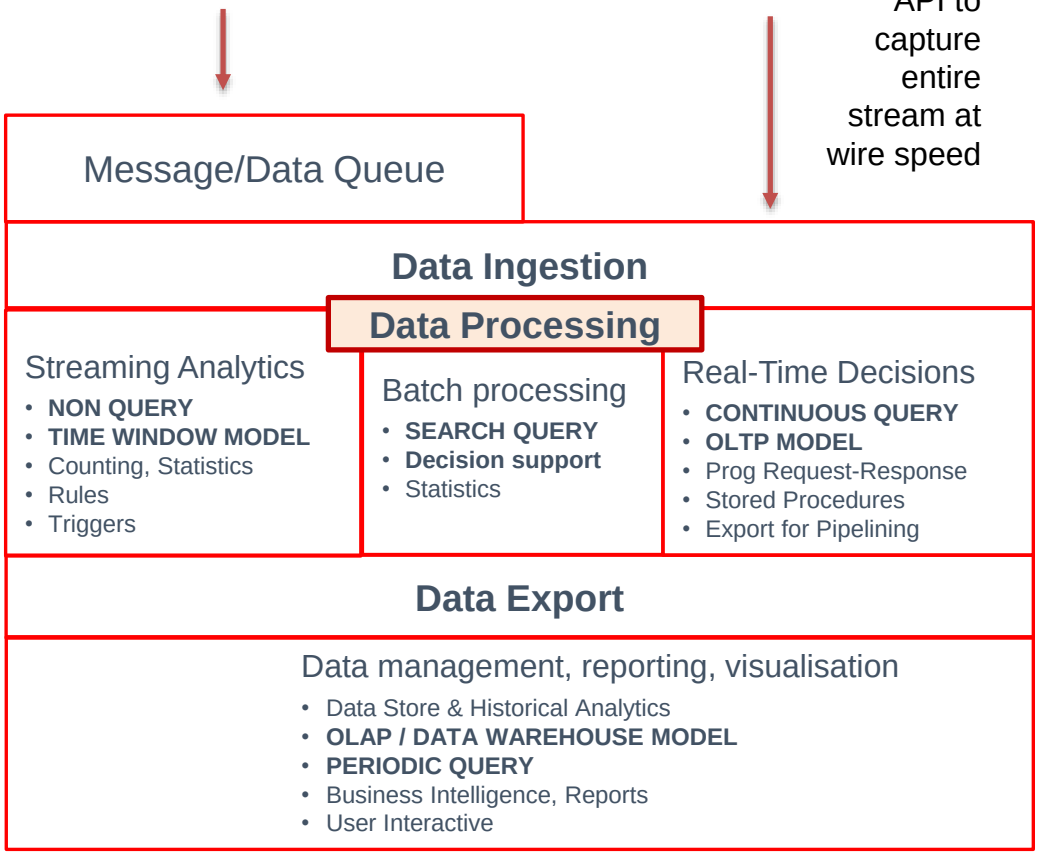
Use **direct Ingestion** API to capture entire stream at wire speed

Real-Time Decisions and/or Results can be fed back “up-stream” to influence the “next step”

Ingestion will transform, normalize, distribute and integrate to one or more of the Analytic / Decision Engines of choice

Use one or more Analytic / Decision engines to accomplish specific task on the Fast Data window

OLAP Engines to support data acquisition for later historical data analysis and Business Intelligence on entire Big Data set



Export will transform, normalize, distribute and integrate to one or more Data Warehouse or Storage Platforms

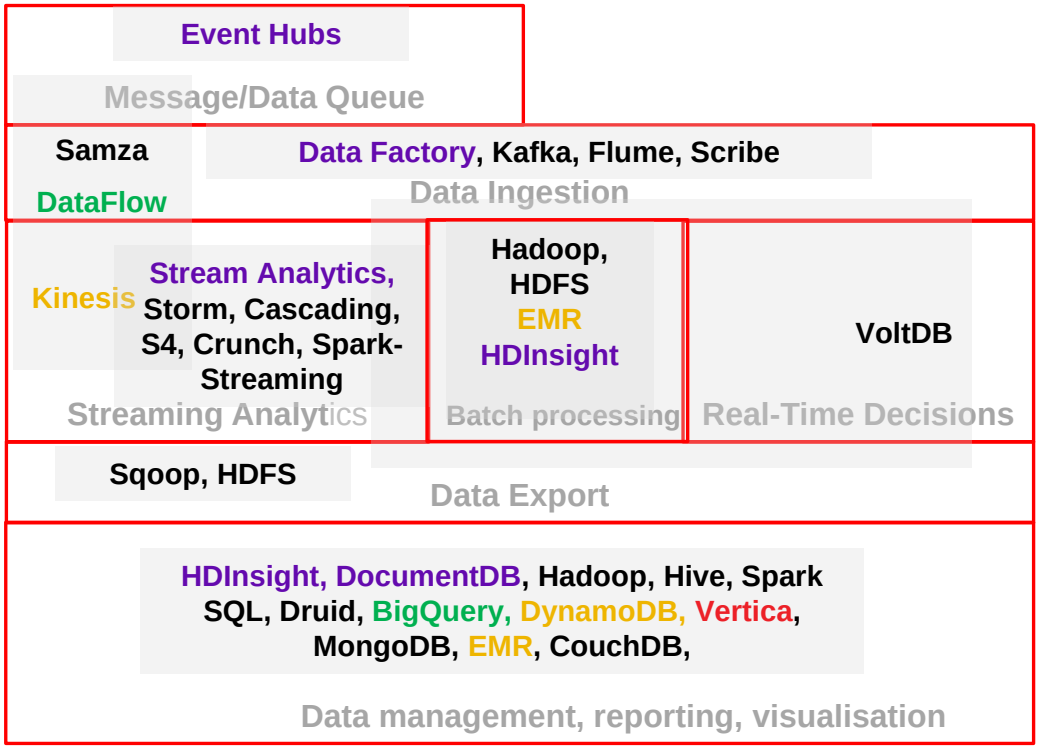
Important Big Data Technologies

Microsoft Azure:
Event Hubs
Data Factory
Stream Analytics
HDInsight
DocumentDB

Google GCE:
DataFlow
BigQuery

Amazon AWS:
Kinesis
EMR
DynamoDB

Proprietary:
Vertica
HortonWorks



Open Source
Samza
Kafka
Flume
Scribe
Storm
Cascading
S4
Crunch
Spark
Hadoop
Hive
Druid
MongoDB
CouchDB
VoltDB

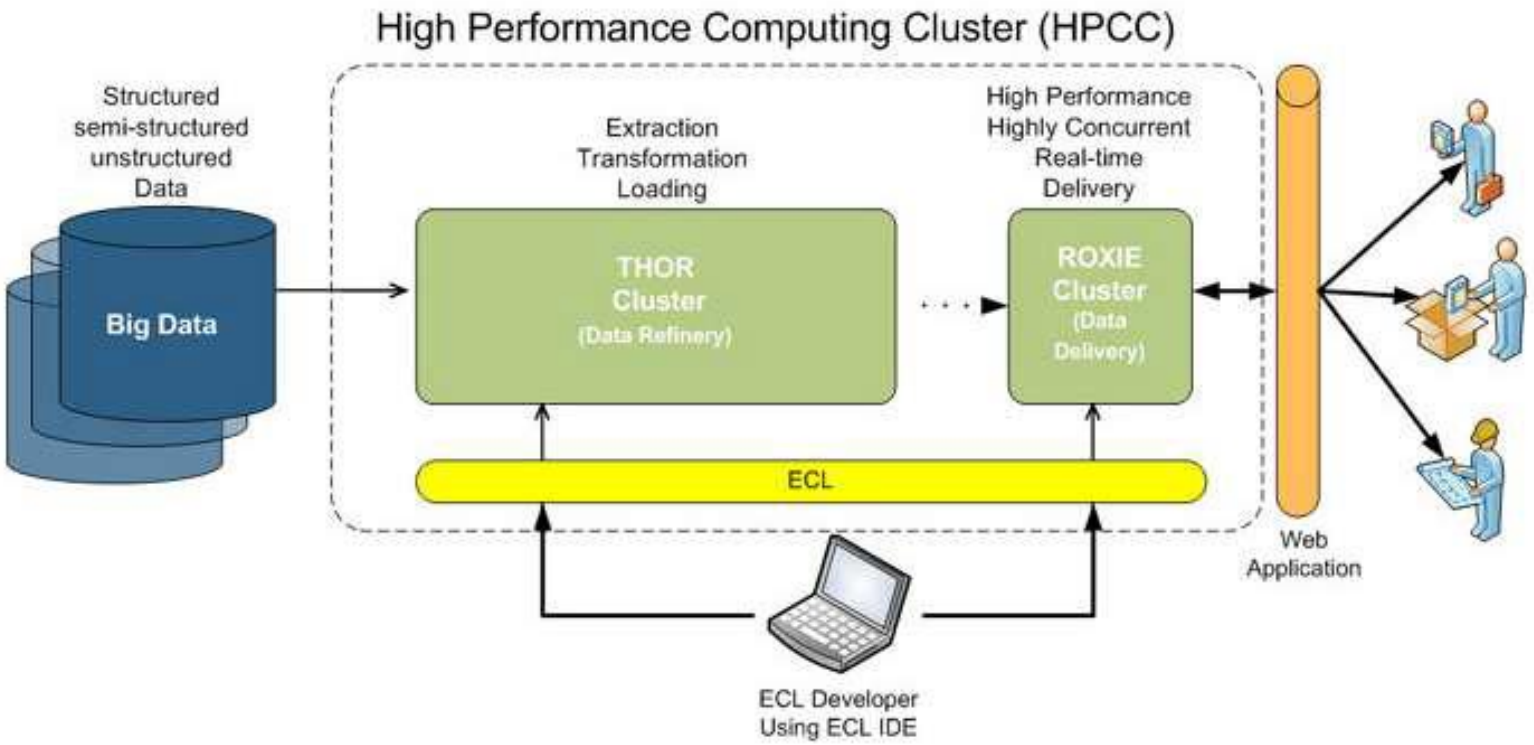
Cloud Platform Benefits for Big Data

- Cloud deployment on virtual machines or containers
  - Applications portability and platform independence, on-demand provisioning
  - Dynamic resource allocation, load balancing and elasticity for tasks and processes with variable load
- Availability of rich cloud based monitoring tools for collecting performance information and applications optimisation
- Network traffic segregated and isolation
  - Big Data applications benefit from lowest latencies possible for node to node synchronization, dynamic cluster resizing, load balancing, and other scale-out operations
  - Clouds construction provides separate networks for data traffic and management traffic
  - Traffic segmentation by creating Layer 2 and Layer 3 virtual networks inside user/application assigned Virtual Private Cloud (VPC)
- Cloud tools for large scale applications deployment and automation
  - Provide basis for agile services development and Zero-touch services provisioning
  - Applications deployment in cloud is supported by major Integrated Development Environment (IDE)

Cloud HPC and Big Data Platforms

- HPC on cloud platform
  - Special HPC and GPU VM instances as well as Hadoop/HPC clusters offered by all CSPs
- Amazon Big Data services
  - Amazon Elastic MapReduce, Kinesis, DynamoDB, Regshift, etc
- Microsoft Analytics Platform System (APS)
  - Microsoft HD Insight/Hadoop ecosystems
- IBM BlueMix applications development platform
  - Includes full cloud services and data analytics services
- LexisNexis HPC Cluster System
  - Combing both HPC cluster platform and optimized data processing languages
- Variety of Open Source tools
  - Streaming analytics/processing tools: Apache Kafka, Apache Storm, Apache Spark

LexisNexis HPCC Systems Architecture



- THOR is used for massive data processing in batch mode for ETL processing
- ROXIE is used for massive query processing and real-time analytics

LexisNexis HPCC Systems as an integrated Open Source platform for Big Data Analytics

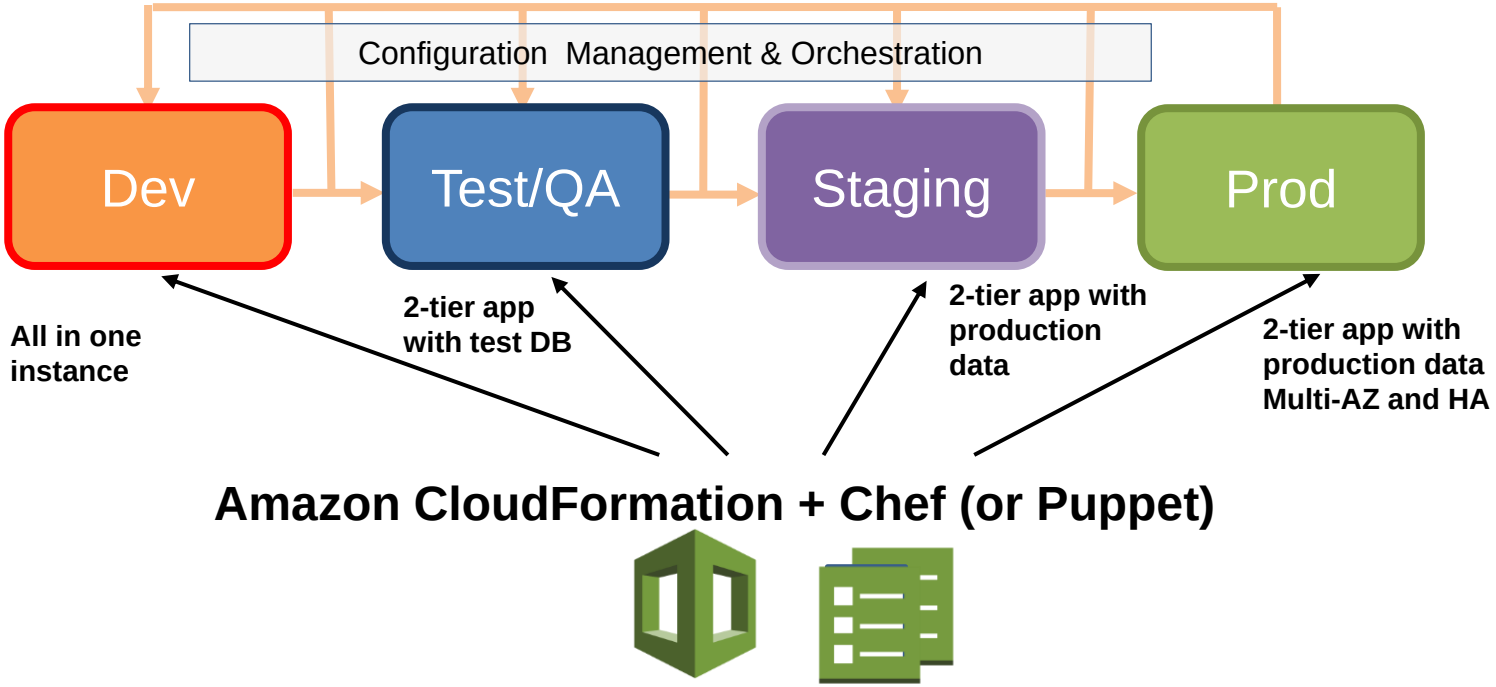
HPCC Systems data analytics environment components and HPCC Systems architecture model is based on a distributed, shared-nothing architecture and contains two cluster

- **THOR Data Refinery**: Massively parallel Extract, Transform, and Load (ETL) engine that can be used for variety of tasks such as massive: joins, merges, sorts, transformations, clustering, and scaling.
- **ROXIE Data Delivery**: Massively parallel, high throughput, structured query response engine with real time analytics capability

Other components of the HPCC environment: data analytics languages

- **Enterprise Control Language (ECL)**: An open source, data-centric declarative programming language
  - The declarative character of ECL language simplifies coding
  - ECL is explicitly parallel and relies on the platform parallelism.
- LexisNexis proprietary record linkage technology **SALT (Scalable Automated Linking Technology)**: automates data preparation process: profiling, parsing, cleansing, normalisation, standardisation of data.
  - Enables the power of the HPCC Systems and ECL
- **Knowledge Engineering Language (KEL)** is an ongoing development
  - KEL is a domain specific data processing language that allows using semantic relations between entities to automate generation of ECL code.

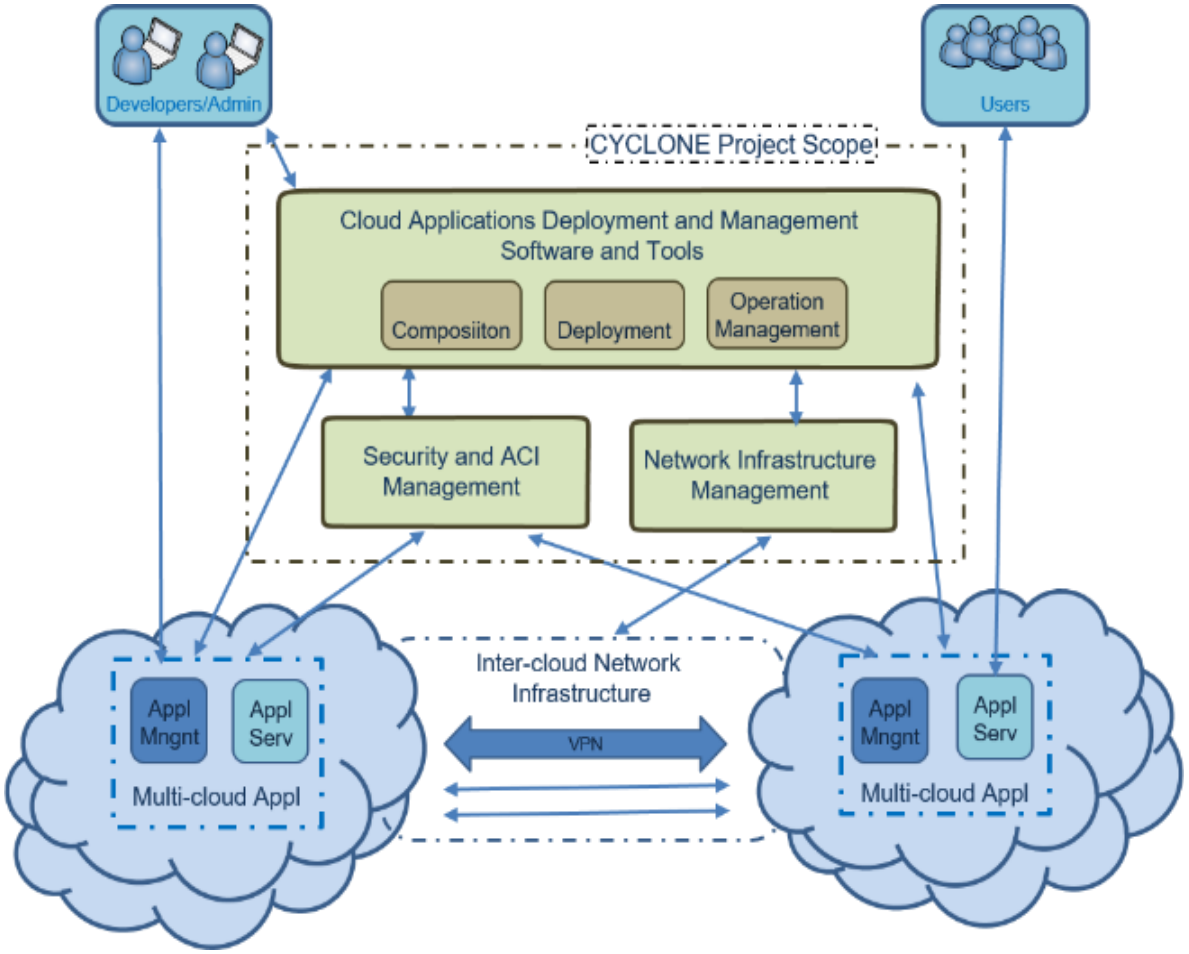
Cloud-powered Services Development Lifecycle: DevOps == Continuous service improvement



- Easily creates test environment close to real
- Powered by cloud deployment automation tools
  - To enable configuration Management and Orchestration, Deployment automation
- Continuous development – test – integration
  - CloudFormation Template, Configuration Template, Bootstrap Template
- Can be used with Puppet and Chef, two configuration and deployment management systems for clouds

[ref] Building Powerful Web Applications in the AWS Cloud" by Louis Columbus
http://softwarestrategiesblog.com/2011/03/10/building-powerful-web-applications-in-the-aws-cloud/

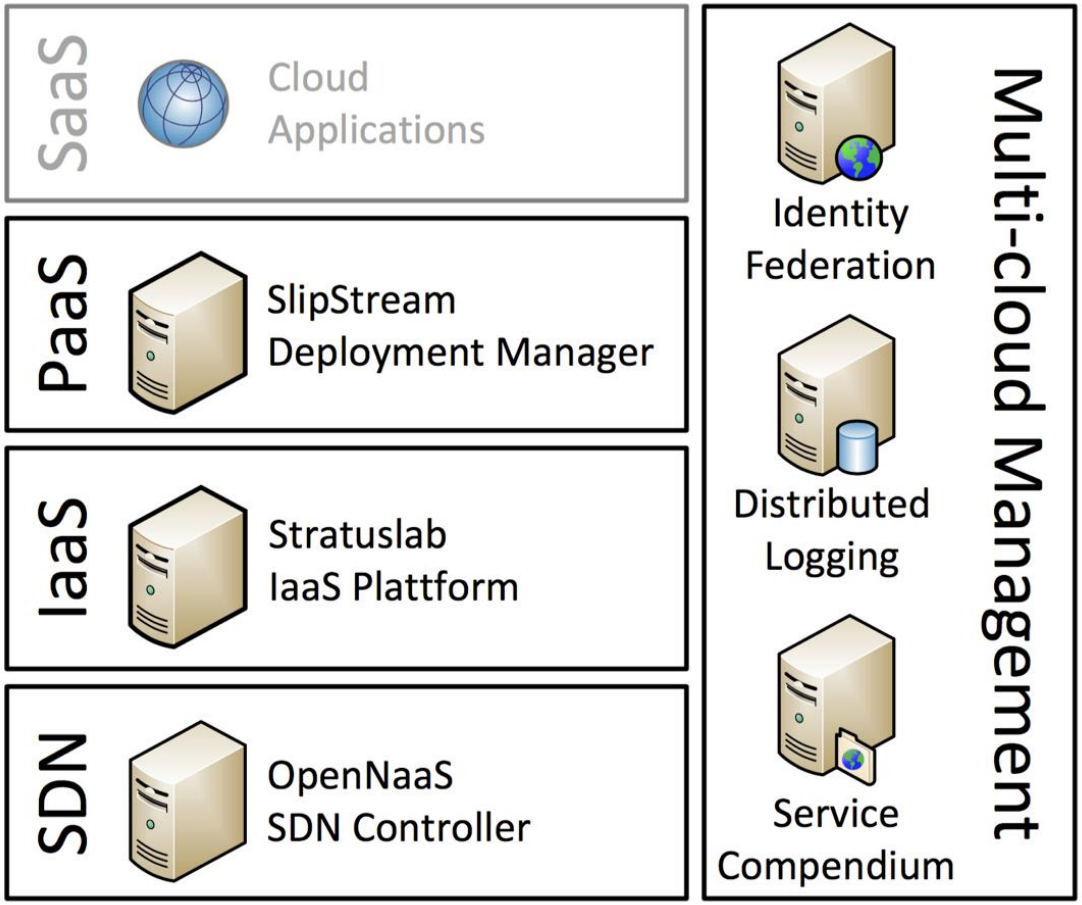
CYCLONE Project: Automation platform for cloud based applications



- Biomedical and Energy applications
- Multi-cloud multi-provider
- Distributed and data processing environment
- Network infrastructure provisioning
  - Dedicated and virtual overlay over Internet
- Automated applications provisioning

Cyclone

CYCLONE Components



- Biomedical and Energy applications
- Multi-cloud multi-provider
- Distributed and data processing environment
- Network infrastructure provisioning
  - Dedicated and virtual overlay over Internet
- Automated applications provisioning

Cyclone

SlipStream – Cloud Automation and Management Platform

Providing complete engineering PaaS supporting DevOps processes

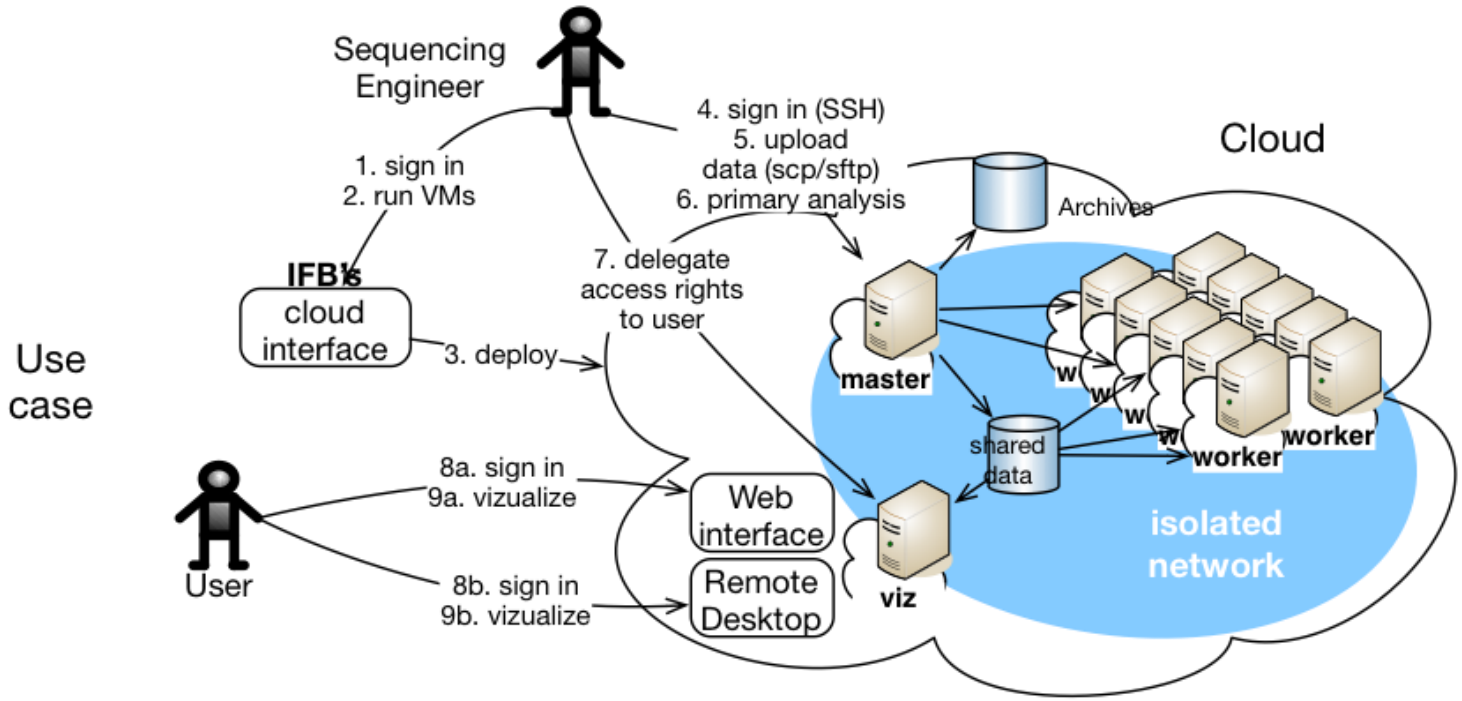
- Deployment engine
- “App Store” for sharing application definitions with other users
- “Service Catalog” for finding appropriate cloud service offers
- Proprietary Recipe format
  - Stored and shared via AppStore
- All features are available through web interface or RESTful API

- Similar to Chef, Puppet, Ansible
- Supports multiple cloud platforms: AWS, Microsoft Azure, StratusLab, etc.

SlipStream

http://sixsq.com/products/slipstream/index.html

Bioinformatics Use Cases



1. Securing human biomedical data
2. Cloud virtual pipeline for microbial genomes analysis
3. Live remote cloud processing of sequencing data

On-demand bandwidth, compute as well as complex orchestration.

Functionality used for use cases deployment

The definition of an application component consists of a series of **recipes** that are executed at various stages in the lifecycle of the application.

- **Pre-install**: Used principally to configure and initialize the operating system's package management.
- **Install packages**: A list of packages to be installed on the machine. SlipStream supports the package managers for the RedHat and Debian families of OS.
- **Post-install**: Can be used for any software installation that can not be handled through the package manager.
- **Deployment**: Used for service configuration and initialization. This script can take advantage of SlipStream's "parameter database" to pass information between components and to synchronize the configuration of the components.
- **Reporting**: Collects files (typically log files) that should be collected at the end of the deployment and made available through SlipStream.

The master node deployment

The master node deployment script performs the following actions:

- Initialize the yum package manager.
- Install bind utilities.
- Allow SSH access to the master from the slaves.
- Collect IP addresses for batch system.
- Configure batch system admin user.
- *Export NFS file systems to slaves.*
- Configure batch system.
- Indicate that cluster is ready for use.

Example script to export NSF directory

```
 ss-display "Exporting SGE_ROOT_DIR..."
echo -ne "$SGE_ROOT_DIR\t" > $EXPORTS_FILE
for ((i=1; i<=`ss-get
      Bacterial_Genomics_Slave:multiplicity`; i++ ));
do
  node_host=`ss-get
      Bacterial_Genomics_Slave.$i:hostname`
  echo -ne $node_host >> $EXPORTS_FILE
  echo -ne "(rw,sync,no_root_squash) ">> $EXPORTS_FILE
done
echo "\n" >> $EXPORTS_FILE # last for a newline
exportfs -av
```

- ss-get command retrieves a value from the parameter database.
- It determines the number of slaves and then loops over each one
  - Retrieves each IP address (hostname) and add it to the NFS exports file.

Questions and Discussion

- More information about CYCLONE project http://www.cyclone-project.eu/
- SlipStream http://sixsq.com/products/slipstream/index.html