Big Data Architecture

Guido Schmutz

Guido Schmutz

Working for Trivadis for more than 18 years

Oracle ACE Director for Fusion Middleware and SOA

Co-Author of different books

Consultant, Trainer Software Architect for Java, Oracle, SOA and Big Data / Fast Data

Member of Trivadis Architecture Board

Technology Manager @ Trivadis

More than 25 years of software development experience

Contact: guido.schmutz@trivadis.com

Blog: http://guidoschmutz.wordpress.com

Twitter: gschmutz

daniel LIEBHART
guido SCHMUTZ
marcel LATTMANN
markus HEINISCH
michael KÖNINGS
mischa KÖLLIKER
perry PAKULL
peter WELKENBACH

ARCHITECTURE BLUEPRINTS

EIN LEITFADEN ZUR KONSTRUKTION VON SOFTWARESYSTEMEN MIT JAVA SPRING, .NET, ADF, FORMS UND SOA

trivadis

HANSER

2., aktualisierte und erweiterte Auflage

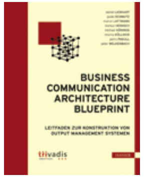
daniel LIEBHART
guido SCHMUTZ
marcel LATTMANN
markus HEINISCH
michael KÖNINGS
mischa KÖLLIKER
perry PAKULL
peter WELKENBACH

INTEGRATION ARCHITECTURE BLUEPRINT

LEITFADEN ZUR KONSTRUKTION VON INTEGRATIONSLÖSUNGEN

trivadis

HANSER



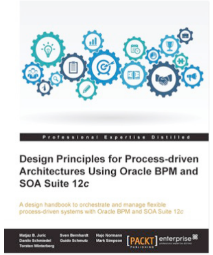
daniel LIEBHART
guido SCHMUTZ
marcel LATTMANN
markus HEINISCH
michael KÖNINGS
mischa KÖLLIKER
perry PAKULL
peter WELKENBACH

BUSINESS COMMUNICATION ARCHITECTURE BLUEPRINT

LEITFADEN ZUR KONSTRUKTION VON OUTPUT MANAGEMENT SYSTEMEN

trivadis

HANSER



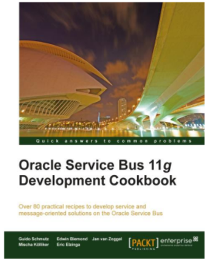
Professional Expertise Distilled

Design Principles for Process-driven Architectures Using Oracle BPM and SOA Suite 12*c*

A design handbook to orchestrate and manage flexible process-driven systems with Oracle BPM and SOA Suite 12*c*

Matjaz B. Juric
Danilo Schmiedel
Torsten Winterberg
Sven Bernhardt
Guido Schmutz
Hajo Normann
Mark Simpson

[PACKT] PUBLISHING enterprise



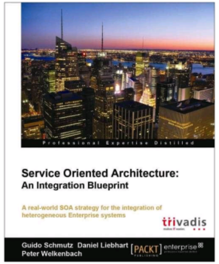
Quick answers to common problems

Oracle Service Bus 11*g* Development Cookbook

Over 80 practical recipes to develop service and message-oriented solutions on the Oracle Service Bus

Guido Schmutz
Mischa Kölliker
Edwin Biemond
Eric Elzinga
Jan van Zoggel

[PACKT] PUBLISHING enterprise



Professional Expertise Distilled

Service Oriented Architecture: An Integration Blueprint

A real-world SOA strategy for the integration of heterogeneous Enterprise systems

trivadis

Guido Schmutz Daniel Liebhart
Peter Welkenbach

[PACKT] PUBLISHING enterprise

trivadis
makes IT easier.

Agenda

1. Introduction
2. Traditional Architecture for Big Data
3. Streaming Analytics Architecture for Fast Data
4. Lambda/Kappa/Unifed Architecture for Big Data
5. Summary

Introduction

Big Data is still “work in progress”

Choosing the right architecture is key for any (big data) project

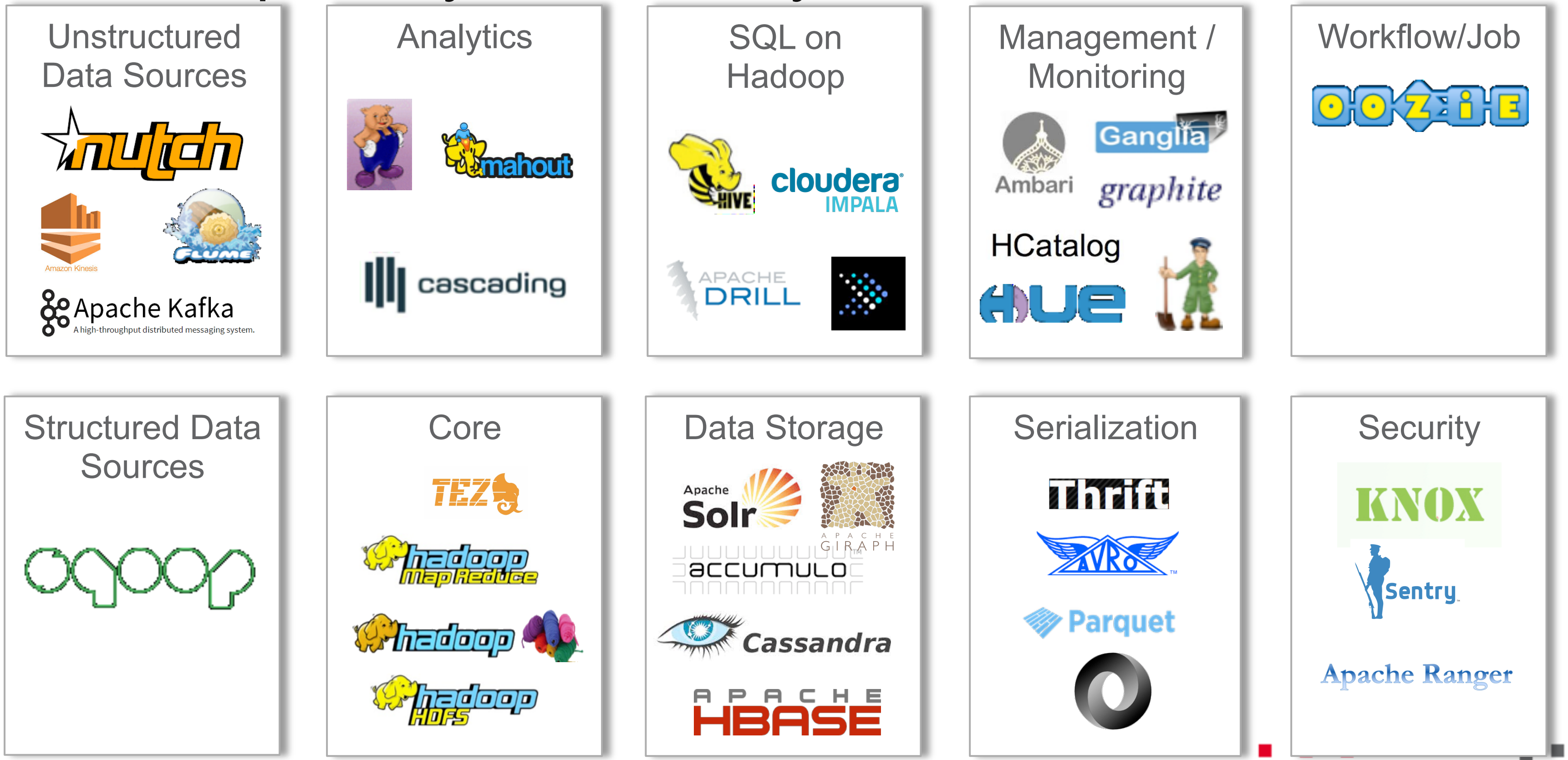
Big Data is still quite a young field and therefore there are no standard architectures available which have been used for years

In the past few years, a few architectures have evolved and have been discussed online

Know the use cases before choosing your architecture

To have one/a few reference architectures can help in choosing the right components

Hadoop Ecosystem – many choices ….



Unstructured Data Sources

Analytics

SQL on Hadoop

Management / Monitoring

Workflow/Job

Structured Data Sources

Core

Data Storage

Serialization

Security

Important Properties to choose a Big Data Architecture

Latency

Keep raw and un-interpreted data “forever” ?

Volume, Velocity, Variety, Veracity

Ad-Hoc Query Capabilities needed ?

Robustness & Fault Tolerance

Scalability

…

From Volume and Variety to Velocity

Big Data has evolved …

Past	Present
Big Data = Volume & Variety	Big Data = Volume & Variety & Velocity

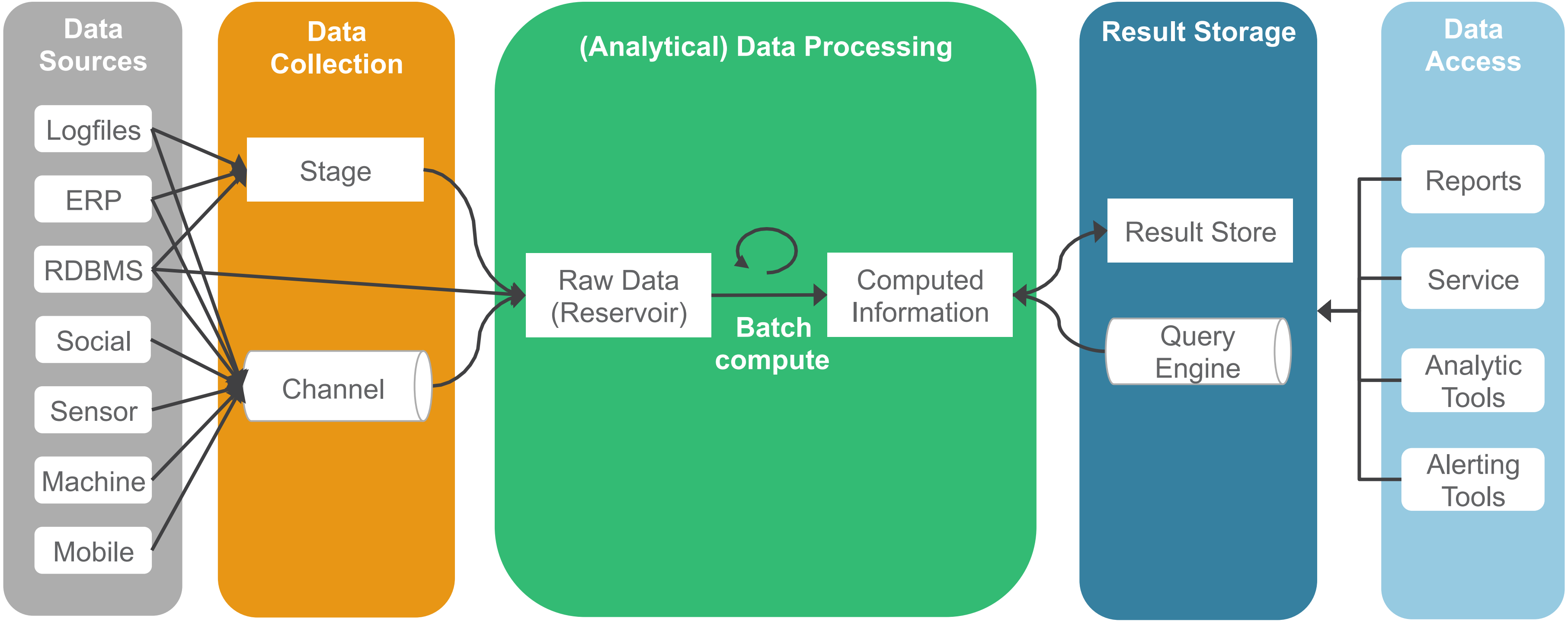
and the Hadoop Ecosystem as well ….

Past	Present
Batch Processing	Batch & Stream Processing
Time to insight of **Hours**	Time to insight in **Seconds**

Adapted from Cloudera Blog article

Traditional Architecture for Big Data

“Traditional Architecture” for Big Data

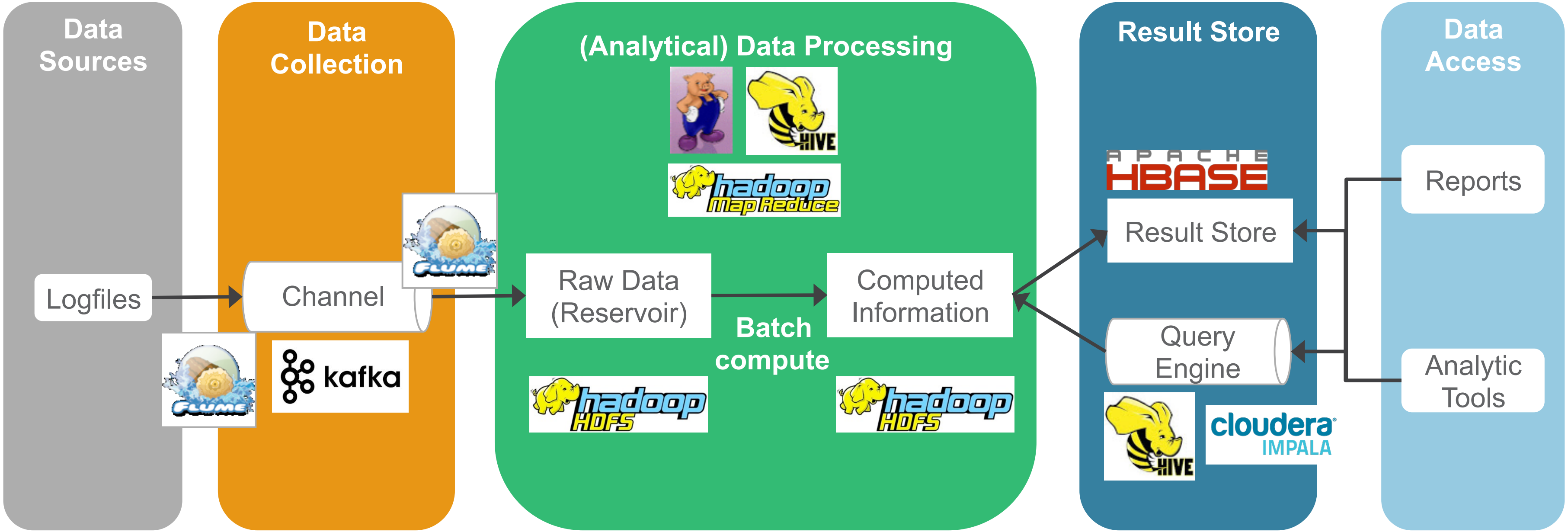


⬭ = Data in Motion

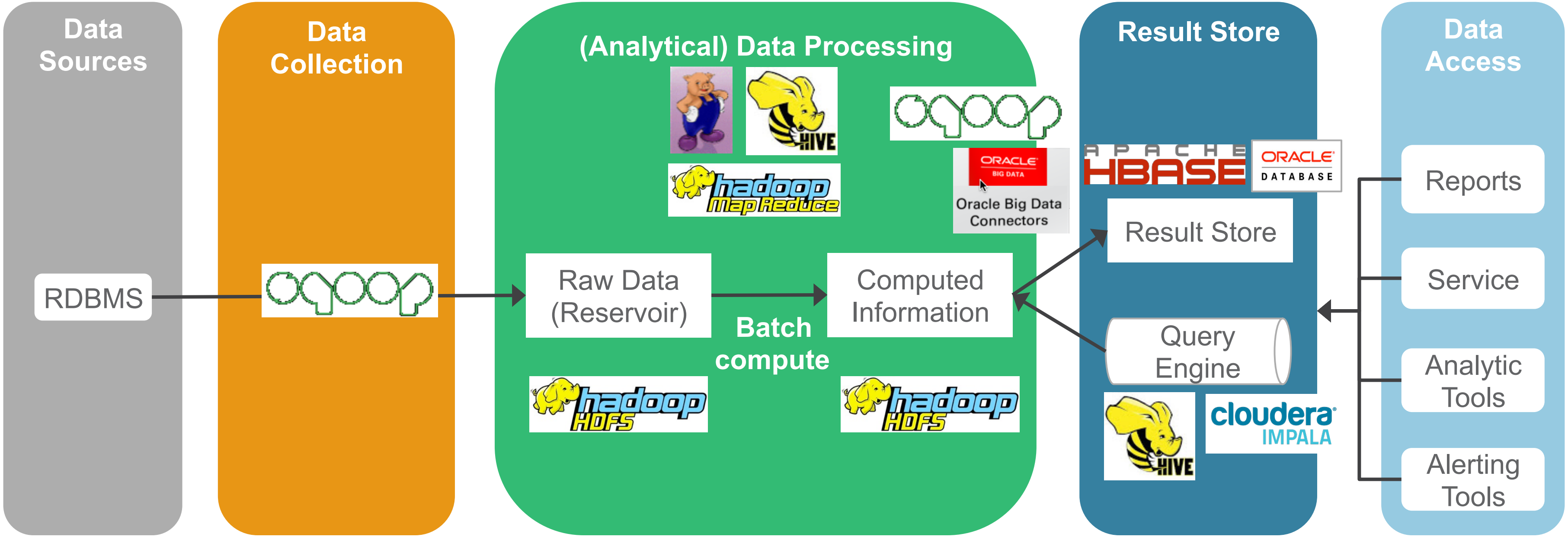
▭ = Data at Rest

trivadis
makes IT easier.

Use Case 1) – Click Stream analysis: 360 degree view of customer

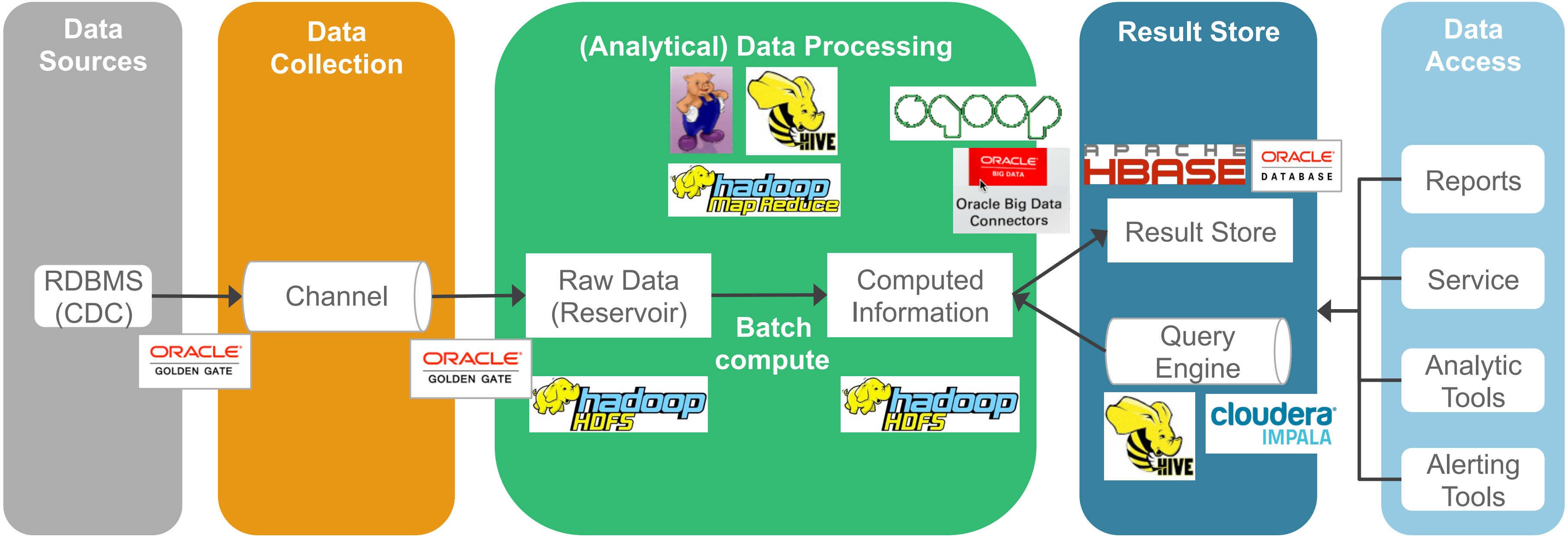


Use Case 2) – Ingest Relational Data into Hadoop and make it accessible



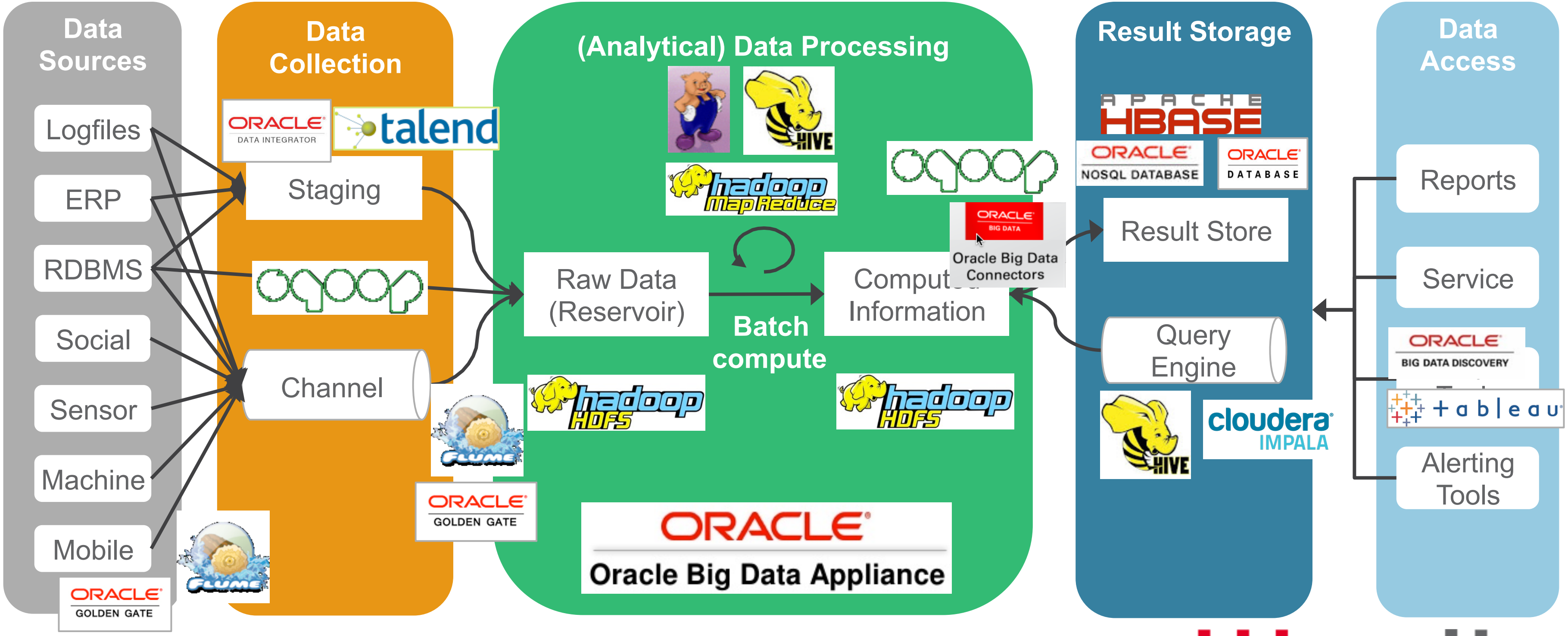
trivadis makes IT easier.

Use Case 2a) – Ingest Relational Data into Hadoop and make it accessible



trivadis makes IT easier.

“Hadoop Ecosystem” Technology Mapping



= Data in Motion

= Data at Rest

trivadis
makes IT easier.

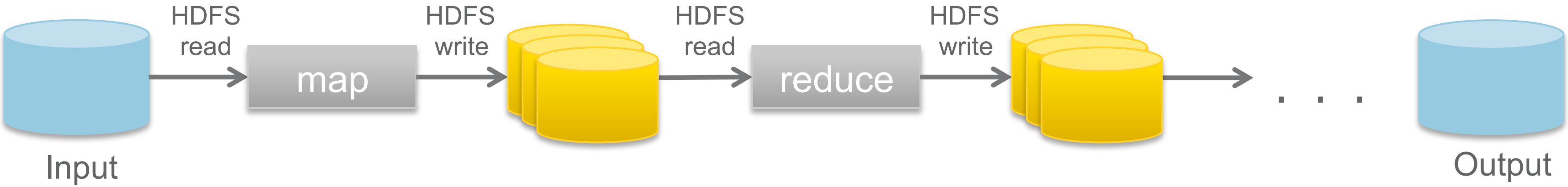
Apache Spark – the new kid on the block

Apache Spark is a fast and general engine for large-scale data processing

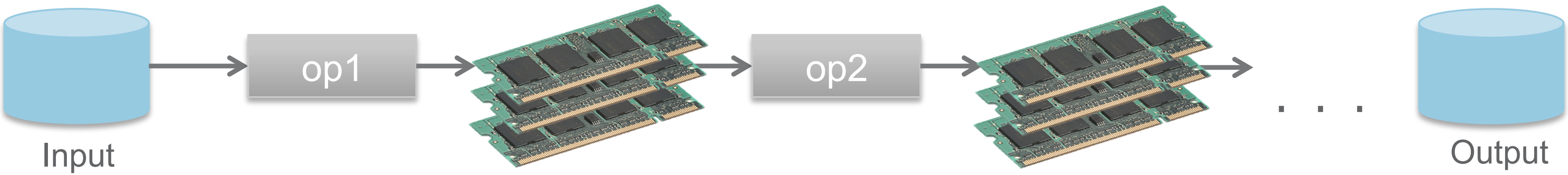
- The hot trend in Big Data!
- Originally developed 2009 in UC Berkley's AMPLab
- Can run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk
- One of the largest OSS communities in big data with over 200 contributors in 50+ organizations
- Open Sourced in 2010 – since 2014 part of Apache Software foundation
- Supported by many vendors

Motivation – Why Apache Spark?

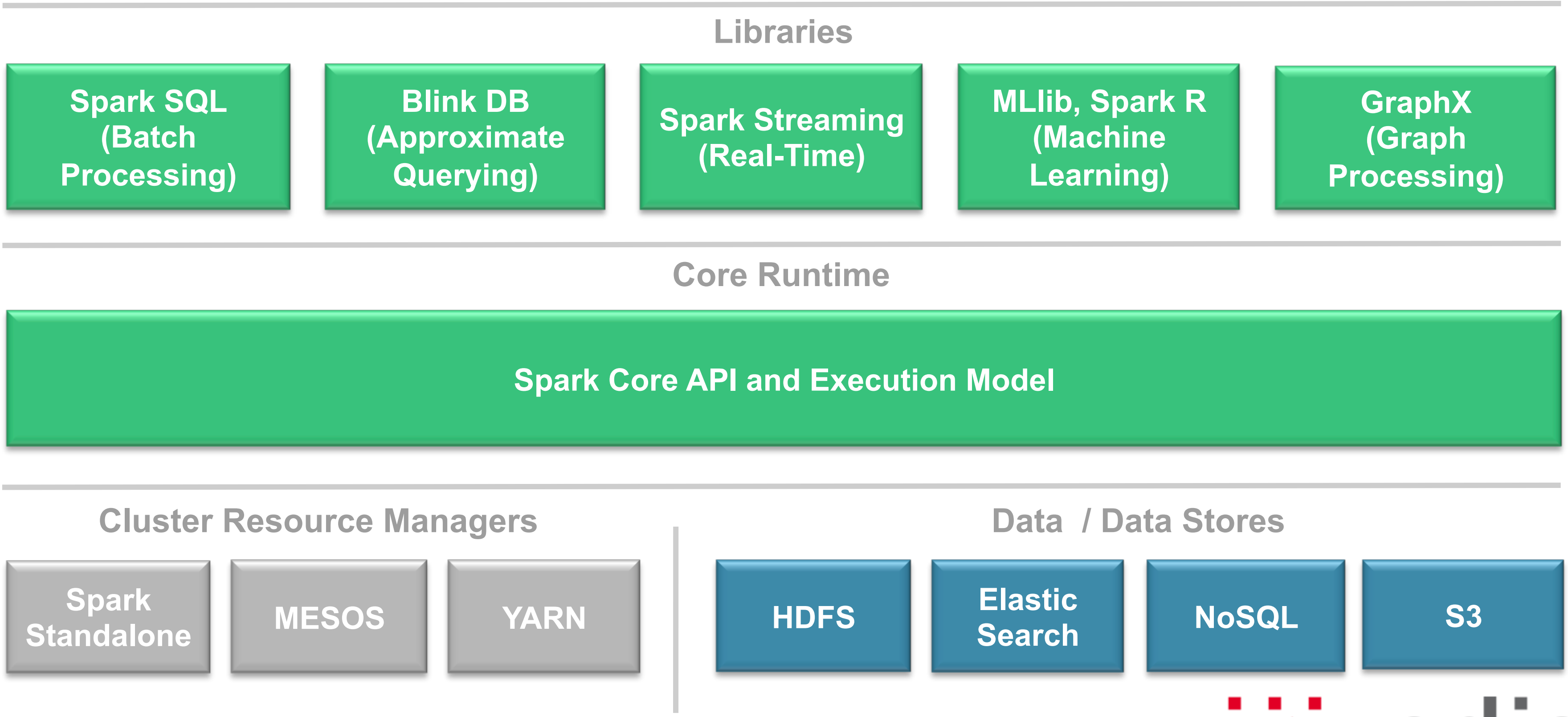
Apache Hadoop MapReduce: Data Sharing on Disk



Apache Spark: Speed up processing by using Memory instead of Disks



Apache Spark “Ecosystem”



Libraries

Spark SQL (Batch Processing)

Blink DB (Approximate Querying)

Spark Streaming (Real-Time)

MLlib, Spark R (Machine Learning)

GraphX (Graph Processing)

Core Runtime

Spark Core API and Execution Model

Cluster Resource Managers

Spark Standalone

MESOS

YARN

Data / Data Stores

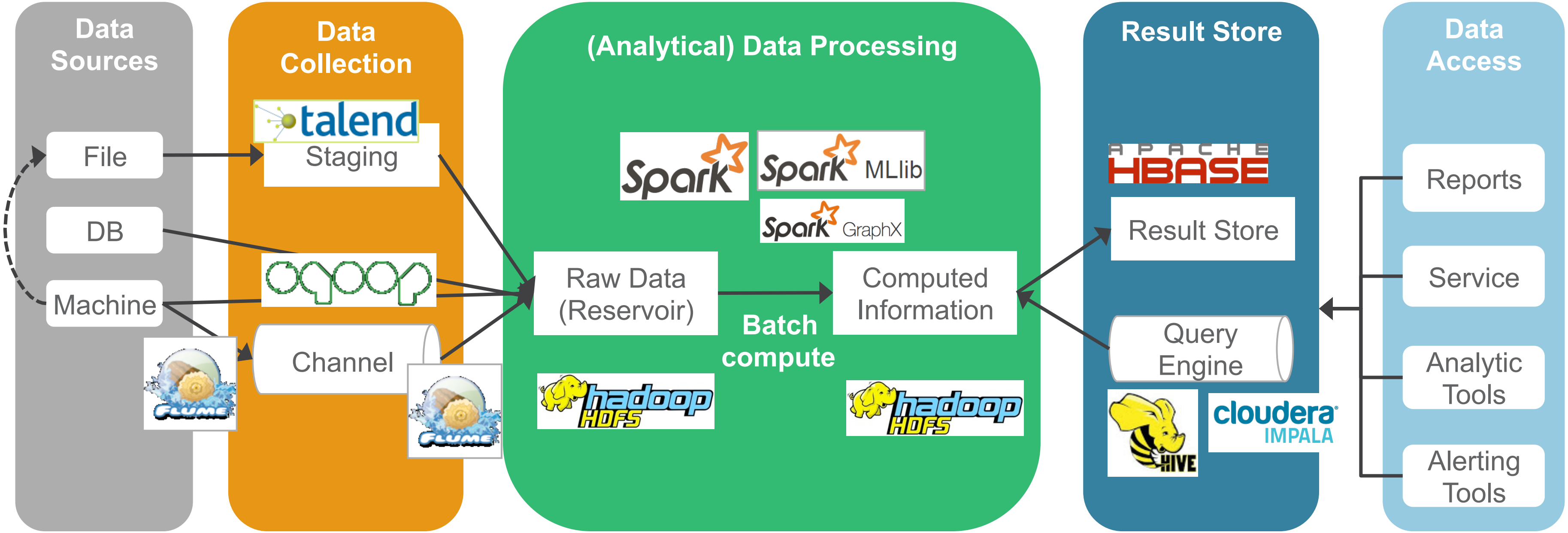
HDFS

Elastic Search

NoSQL

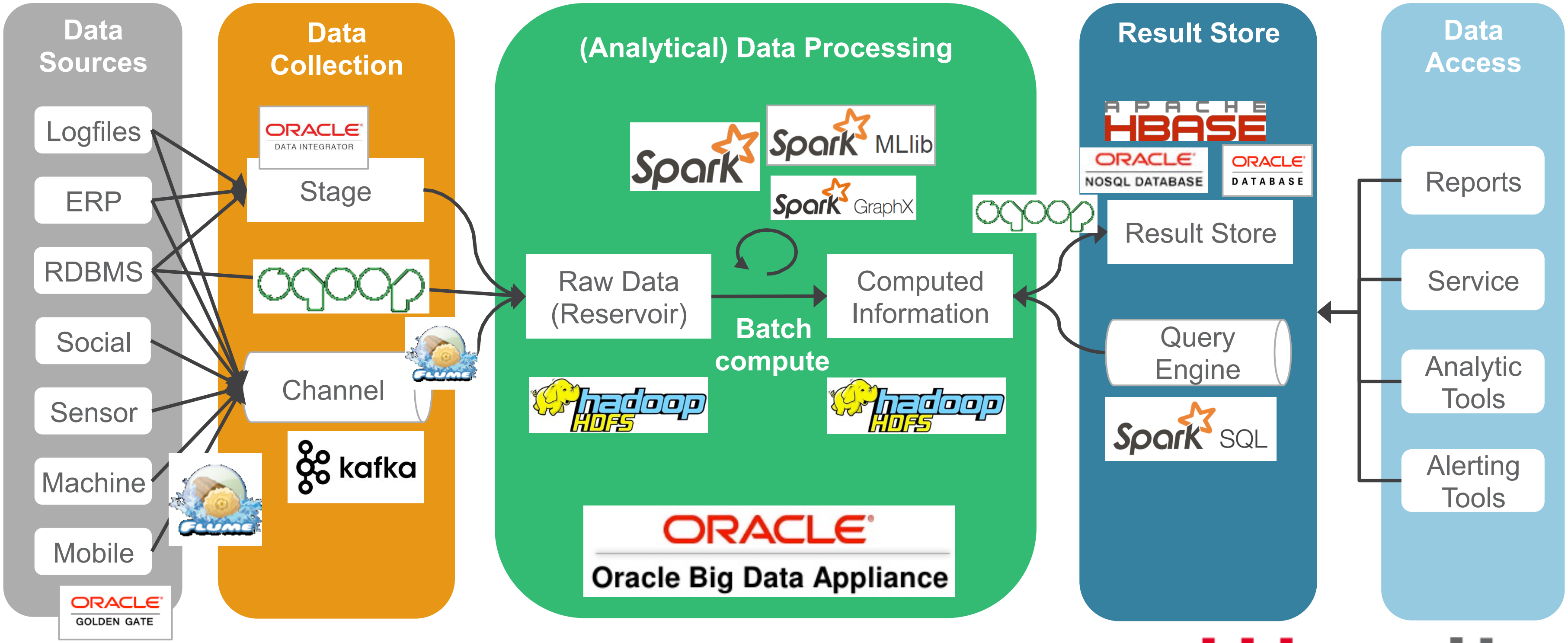
S3

Use Case 3) – Predictive Maintenance through Machine Learning on collected data



trivadis
makes IT easier.

“Spark Ecosystem” Technology Mapping



Data Sources

Logfiles

ERP

RDBMS

Social

Sensor

Machine

Mobile

ORACLE GOLDEN GATE

Data Collection

ORACLE DATA INTEGRATOR

Stage

sqoop

Channel

FLUME

kafka

(Analytical) Data Processing

Spark

Spark MLlib

Spark GraphX

Raw Data (Reservoir)

Computed Information

Batch compute

hadoop HDFS

hadoop HDFS

ORACLE

Oracle Big Data Appliance

Result Store

APACHE HBASE

ORACLE NOSQL DATABASE

ORACLE DATABASE

sqoop

Result Store

Query Engine

Spark SQL

Data Access

Reports

Service

Analytic Tools

Alerting Tools

▭ = Data in Motion

□ = Data at Rest

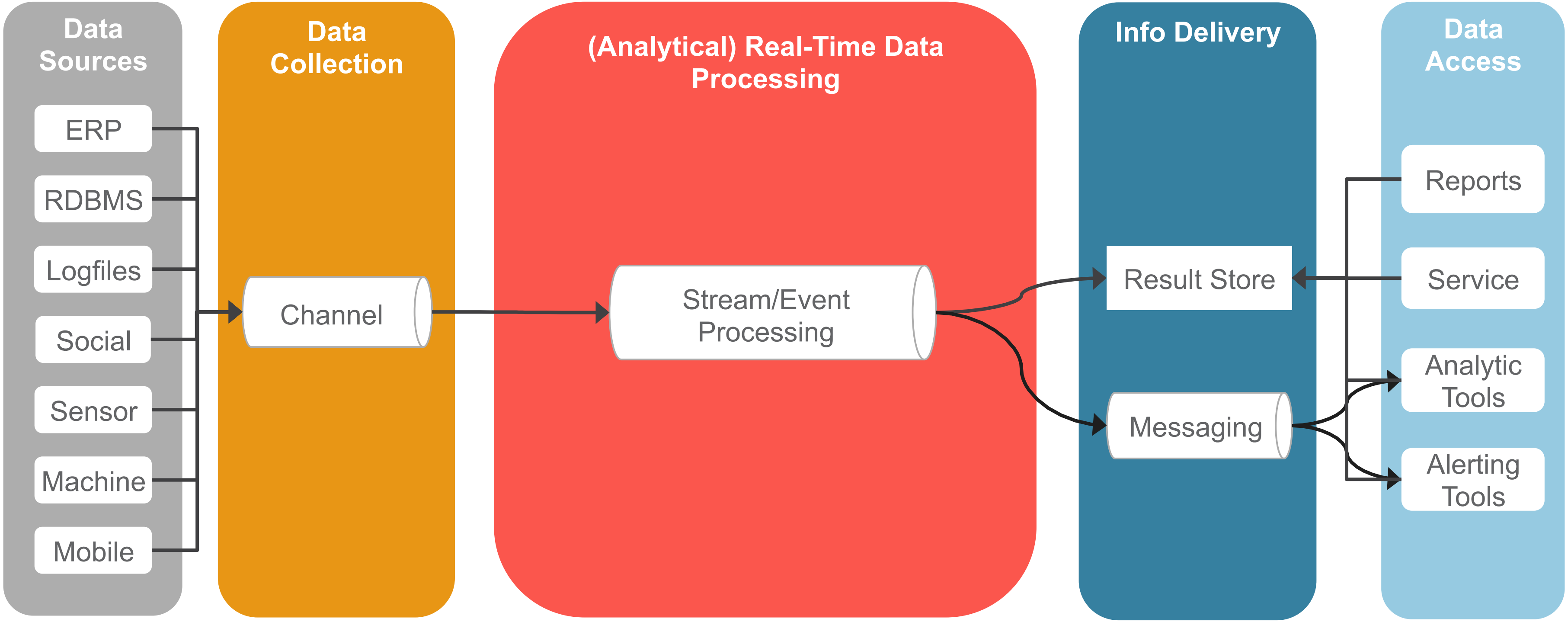
Traditional Architecture for Big Data

- Batch Processing
- Not for low latency use cases
- Spark can speed up, but if positioned as alternative to Hadoop Map/Reduce, it’s still Batch Processing
- Spark Ecosystems offers a lot of additional advanced analytic capabilities (machine learning, graph processing, …)

Streaming Analytics Architecture for Big Data

Streaming Analytics Architecture for Big Data

aka. (Complex) Event Processing)

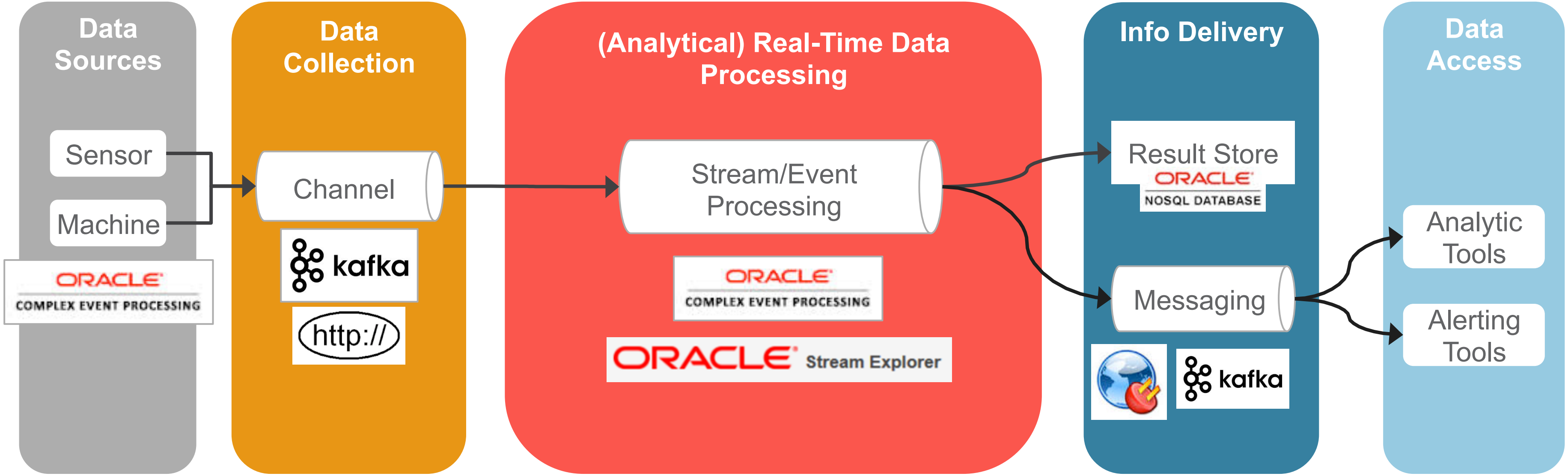


= Data in Motion

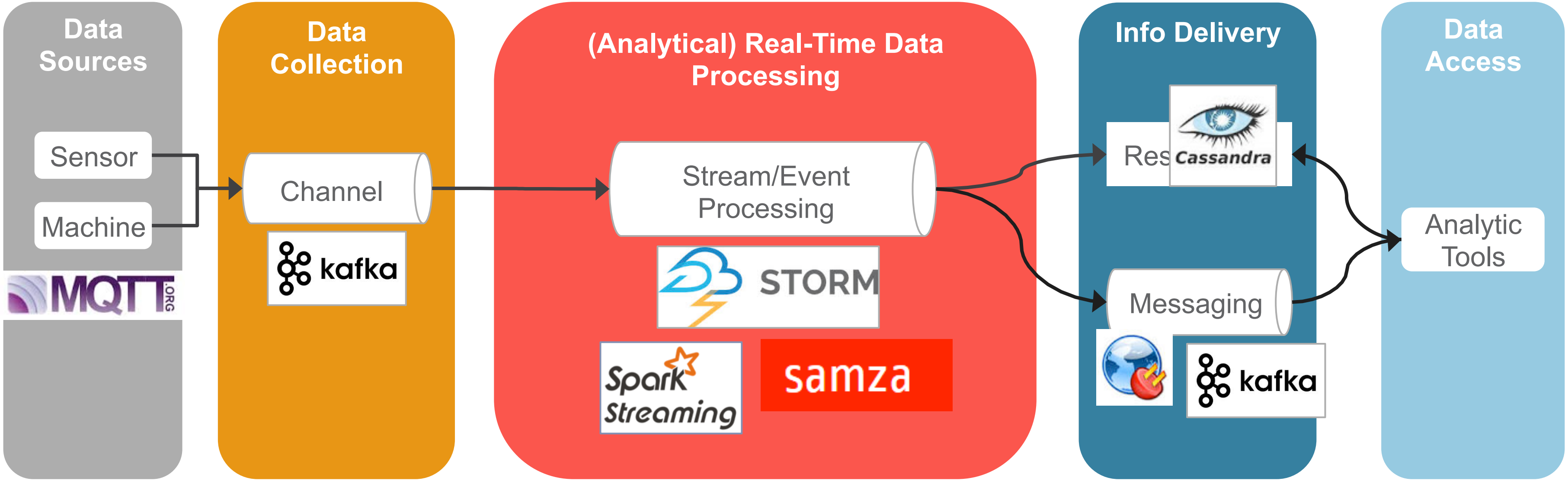
= Data at Rest

trivadis makes IT easier.

Use Case 4) Alerting in Internet of Things (IoT)



Use Case 5) Real-Time Analytics on Sensor Events

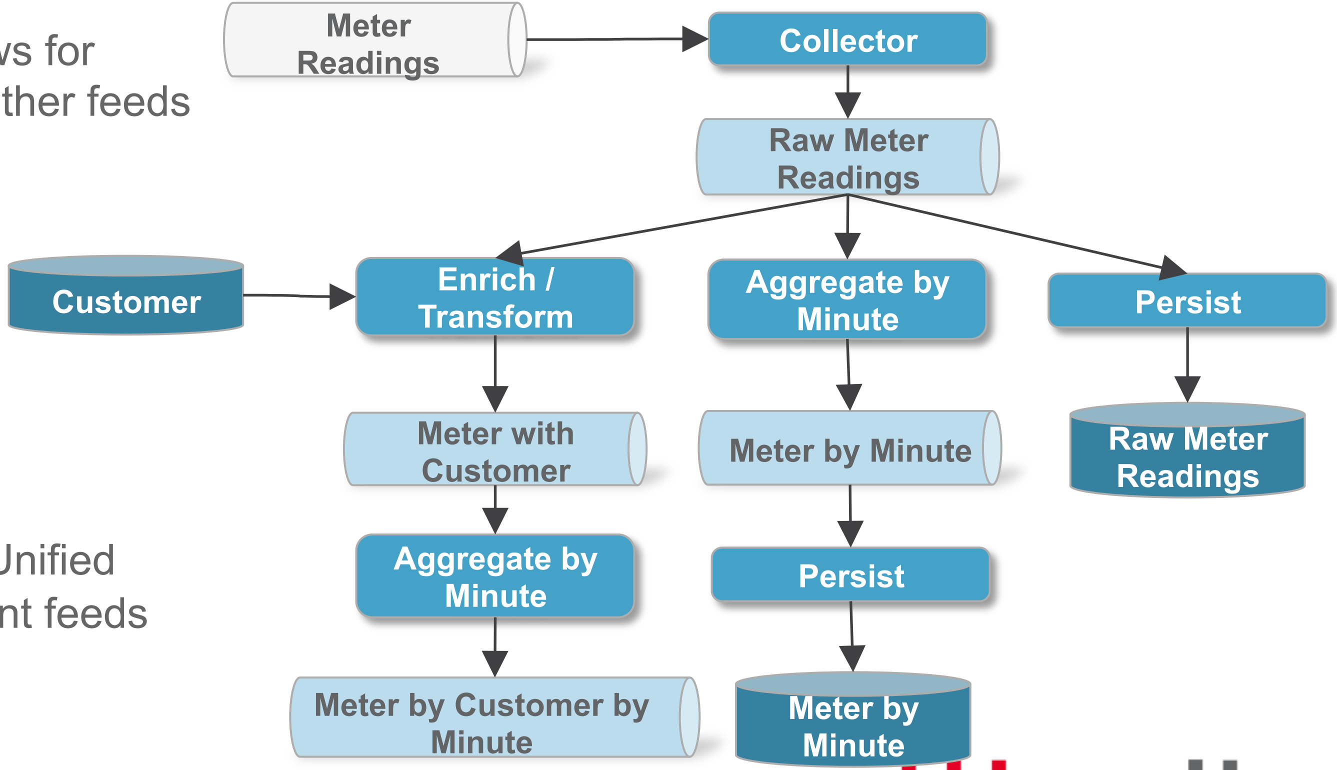


Unified Log (Event) Processing

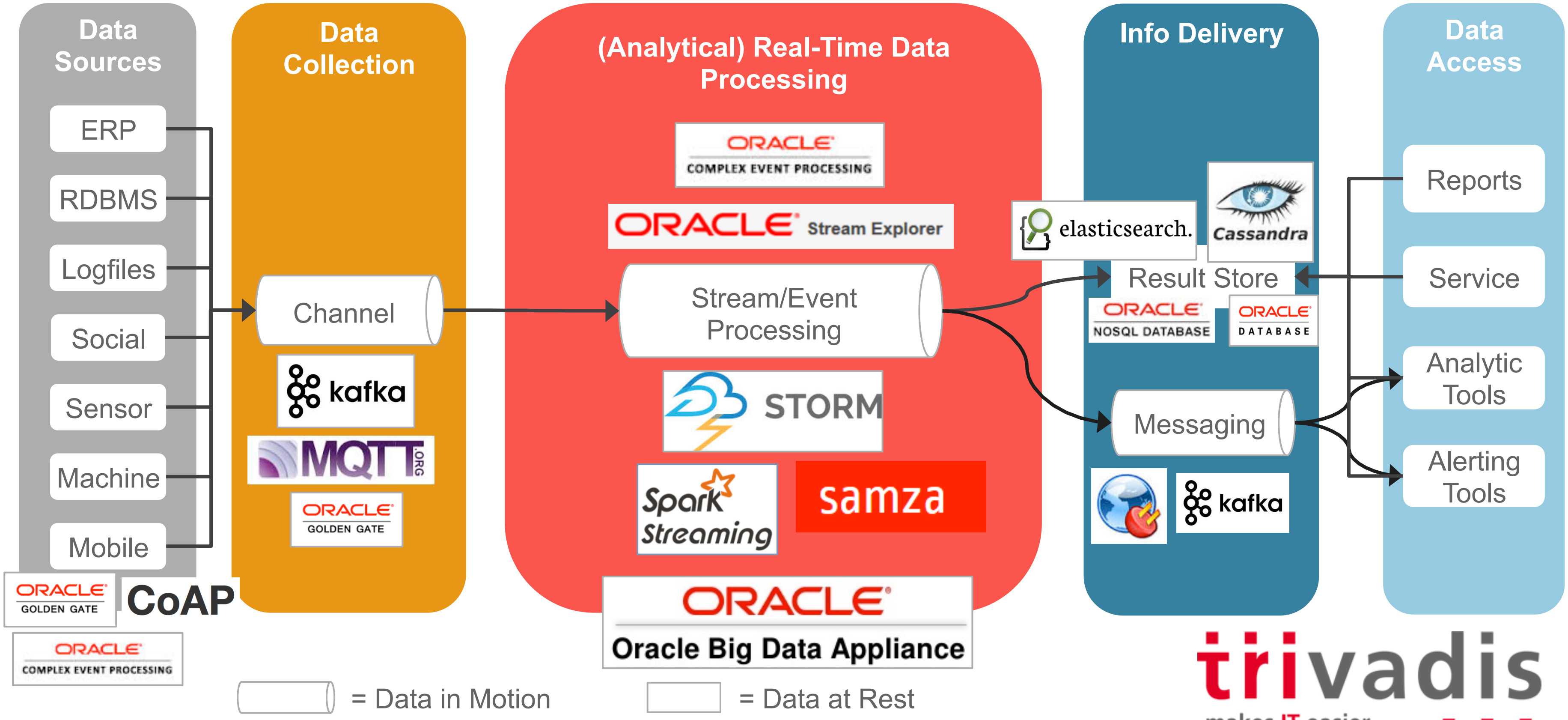
Stream processing allows for computing feeds off of other feeds

Derived feeds are no different than original feeds they are computed off

Single deployment of "Unified Log" but logically different feeds



Streaming Analytics Technology Mapping



trivadis
makes IT easier.

Streaming Analytics Architecture for Big Data

The solution for low latency use cases

Process each event separately => low latency

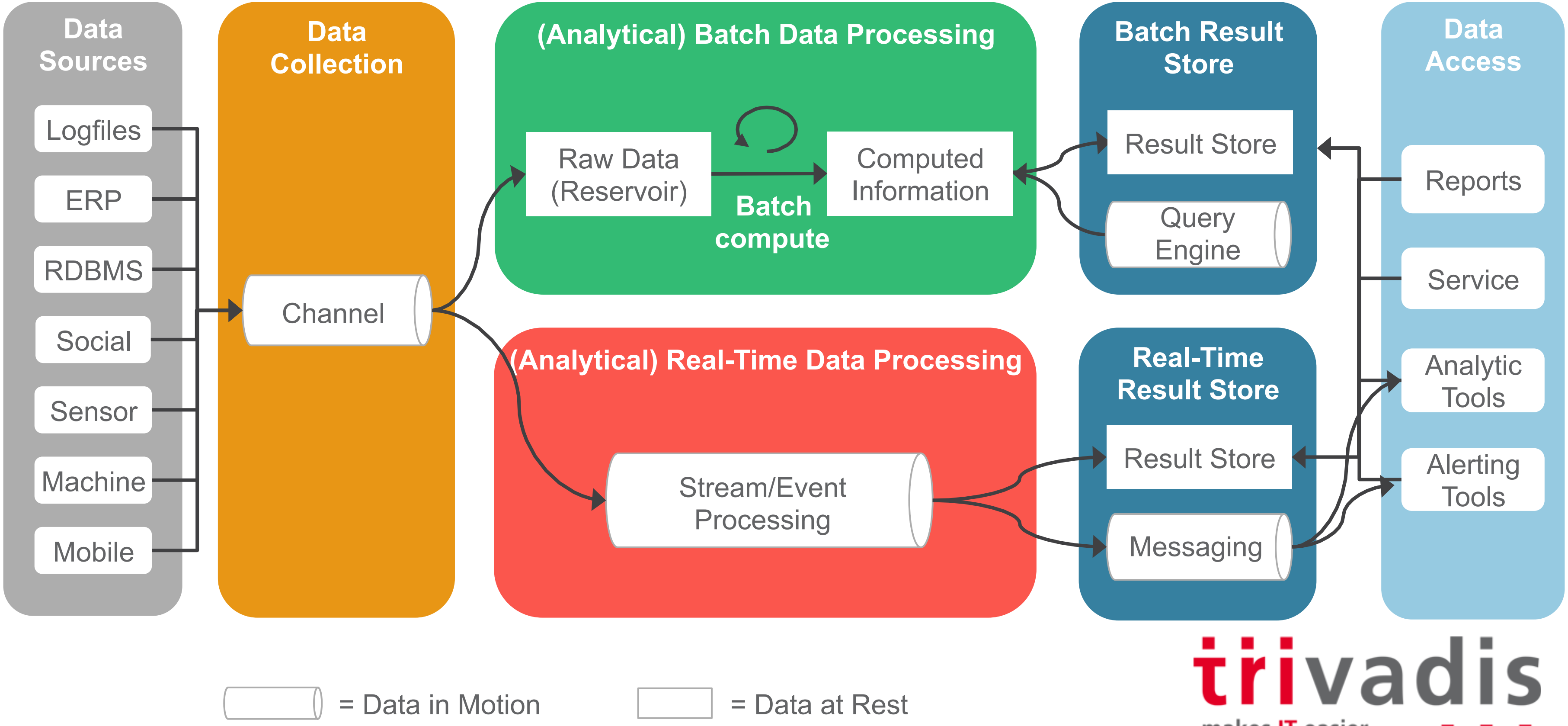
Process events in micro-batches => increases latency but offers better reliability

Previously known as “Complex Event Processing”

Keep the data moving / Data in Motion instead of Data at Rest => raw events are (often) not stored

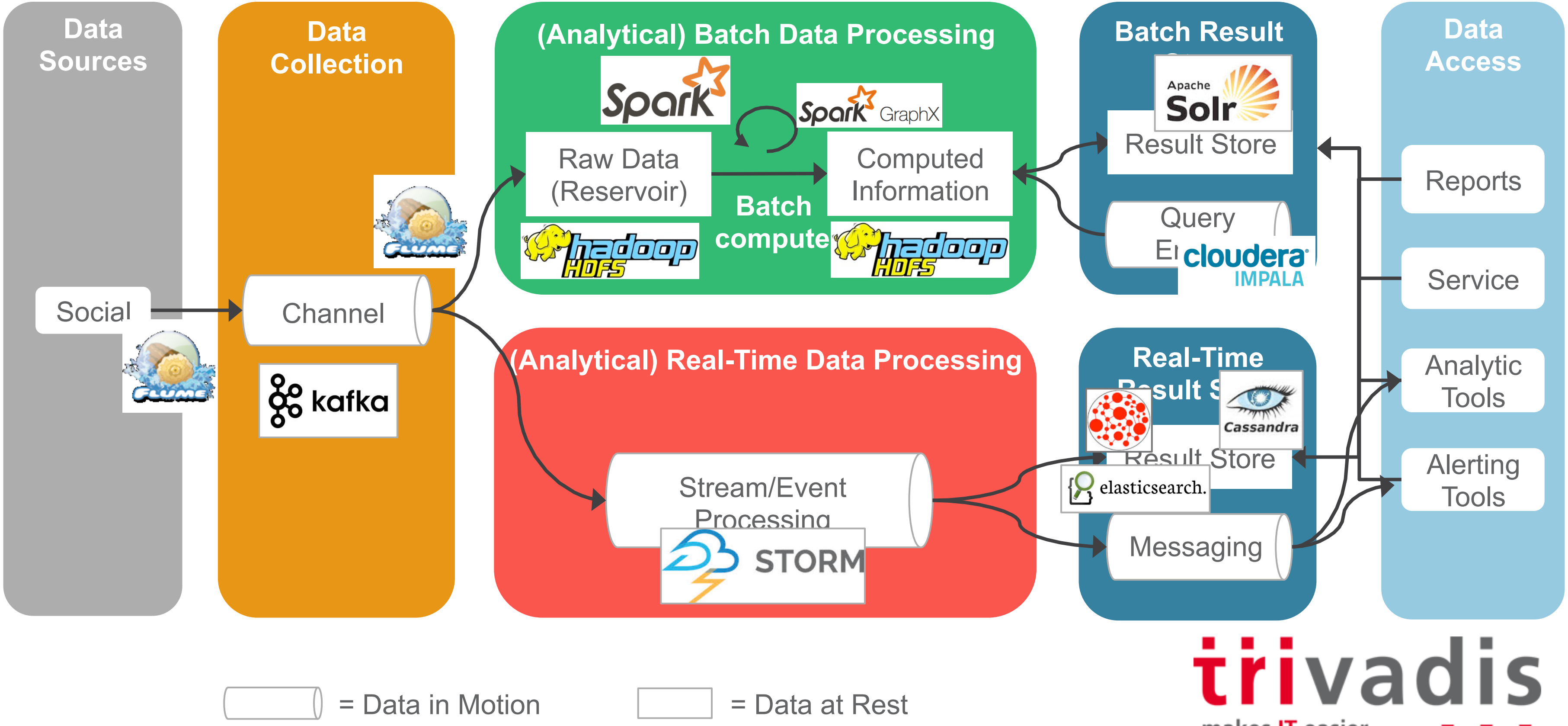
Lambda Architecture for Big Data

"Lambda Architecture" for Big Data



trivadis makes IT easier.

Use Case 6) Social Media and Social Network Analysis



trivadis makes IT easier.

Lambda Architecture for Big Data

Combines (Big) Data at Rest with (Fast) Data in Motion

Closes the gap from high-latency batch processing

Keeps the raw information forever

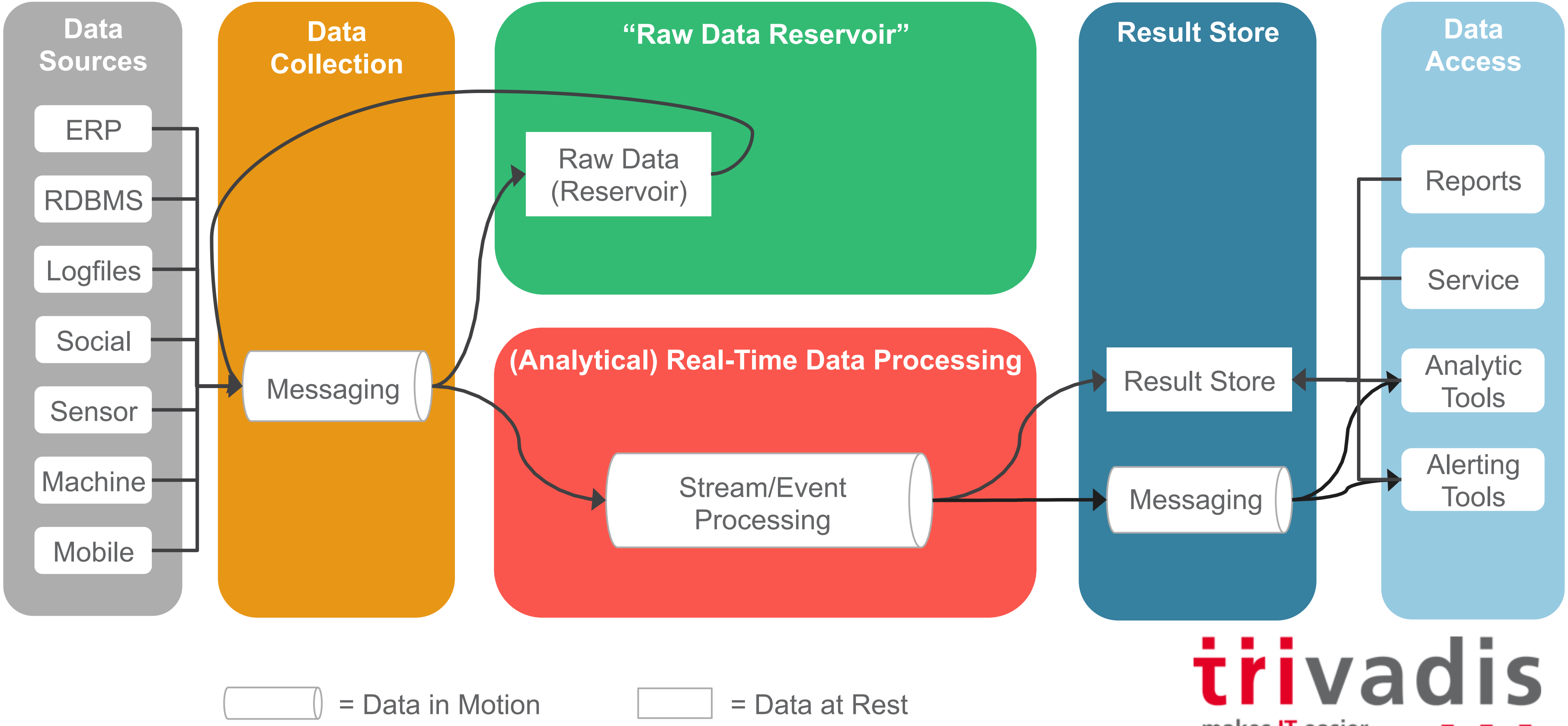
Makes it possible to rerun analytics operations on whole data set if necessary
=> because the old run had an error or
=> because we have found a better algorithm we want to apply

Have to implement functionality twice

- Once for batch
- Once for real-time streaming

„Kappa“ Architecture for Big Data

"Kappa Architecture" for Big Data



trivadis
makes IT easier.

Kappa Architecture for Big Data

The solution for low latency use cases

Process each event separately => low latency

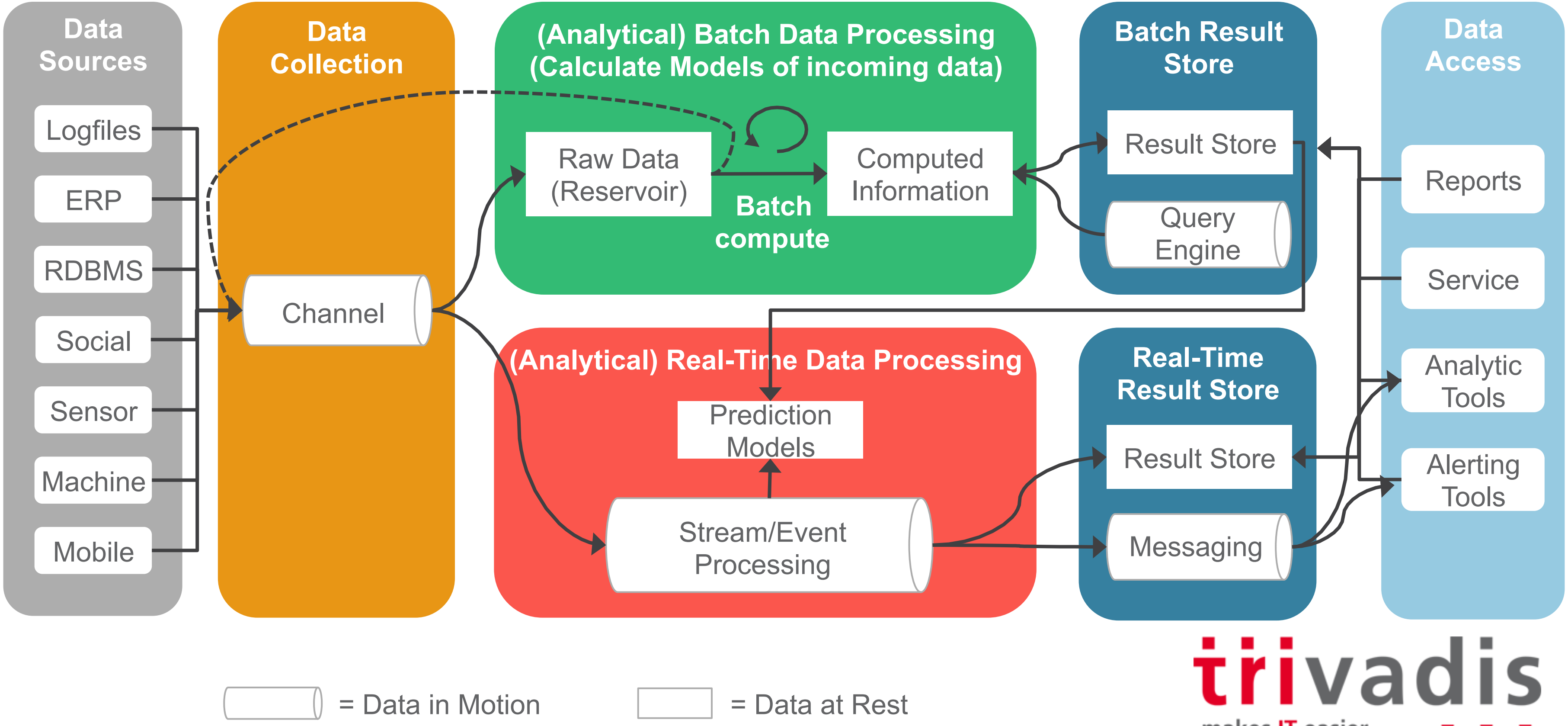
Process events in micro-batches => increases latency but offers better reliability

Previously known as “Complex Event Processing”

Keep the data moving / Data in Motion instead of Data at Rest

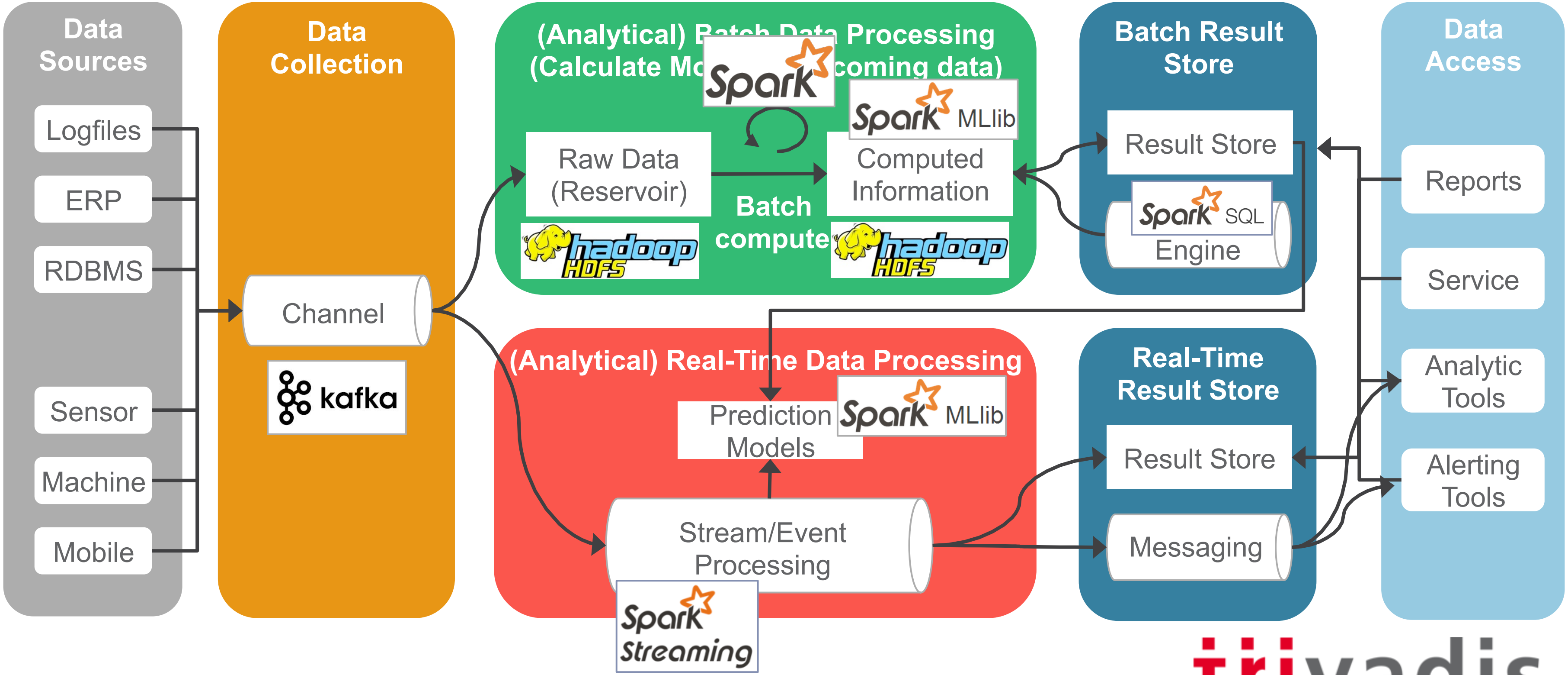
„Unified“ Architecture for Big Data

“Unified Architecture” for Big Data



trivadis makes IT easier.

Use Case 7) Fraud Detection



trivadis
makes IT easier.

Summary

Summary

Know your use cases and then choose your architecture and the relevant components/ products/frameworks

You don’t have to use all the components of the Hadoop Ecosystem to be successful

Big Data is still quite a young field and therefore there are no standard architectures available which have been used for years

Lambda, Kappa Architecture are best practices architectures which you have to adapt to your environment

trivadis

makes IT easier.

Name Referent
Titel Referent

Tel. +00 00 000 00 00
vorname.name@trivadis.com

trivadis
makes IT easier.