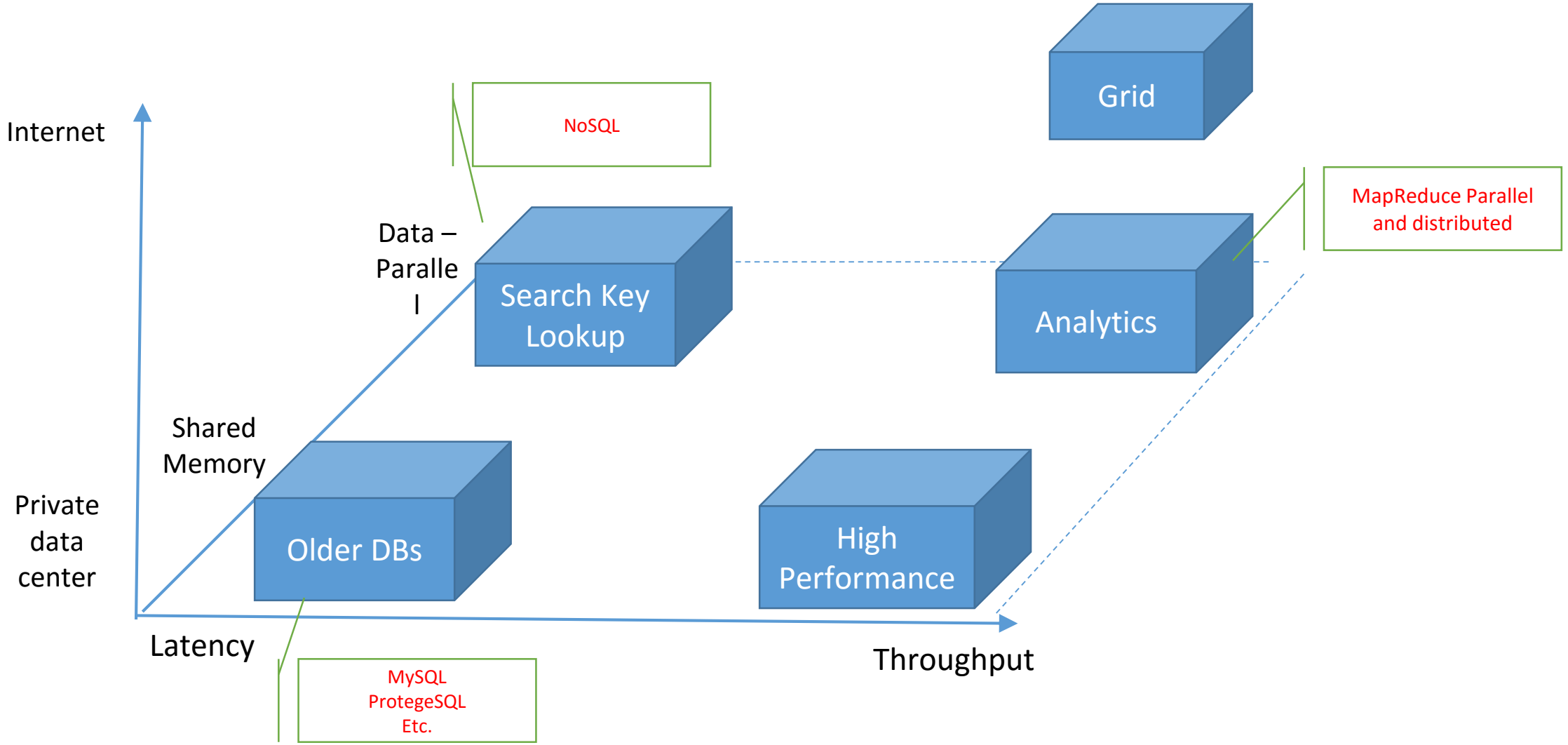
Databases

What's a Database?

At a high-level, here are some ways to think about databases (**there are more**) – the data must have some (even if flexible) structure assumption, must be stored in some location or series of locations with a method to submit an inquiry and process a result

Structuring of Data	Storing Data	Inquiry / Processing Data
• Tabular form with rows and columns • Key-value pairs • Document • Graphs • Hierarchical / Tree • Large Column	• On a single computer • Across several computers • Within a single intranet or data cluster • Through the internet across many disparate computers • Backup • Fault Tolerant? • Multiple access • Consistency • Etc.	• This will depend on the structure and storage of the data (and potential on the parallelism of the processing too) • Relational algebra • SQL • MapReduce • Other types of inquiry / processing methods

Database Design



Ref - by Michael Isard at Microsoft Research

Database Structures

Others

Relational

Examples: MySQL, SQL Server, Oracle

Description: a database structured to recognize relations among stored items of information usually with a defined schema

Key Characteristics: Relational Algebra, Physical / logical independence, *“forced to be clean”

When do you use:

Distributed + Parallel

Examples: Hadoop, Teradata

Description: Data is distributed across a set of nodes and processing is performed in parallel

Key Characteristics: Scalable, Fault Tolerance (during processing), “no loading”, bring operations to data.

When do you use: Massive data, capable or real time, long and complicated calculating and processing, unstructured

NoSQL

Examples: MongoDB, CouchDB, Vertica, Cassandra and many others

Description: Storage and retrieval of data in alternative form to traditional relational data (row and column), data can be stored in many ways aimed at solving particular problem: column, document, key-value, graph and others

Key Characteristics: scalable, some search and retrieval is faster, finer control of data consistency

When do you use: Real time web applications and large data sets, unstructured

More on Distributed Systems

Distributed Query

What happens during the query:

- "Rewrite" the query as a union of subqueries
- Workers communicate through standard interfaces, so compatible with federated, heterogeneous, or distributed databases

In other words – data is distributed, we do a series of steps to find what we want, assimilate it together and reduce it down to our result

Buzz word: Map Reduce

Example: Hadoop – an open source distributed database system from Apache with a million extensions.

Parallel Query

What happens during the query

- Each operator is implemented with a parallel algorithm
- Algorithm gets run on each "AMP" or server and results are returned

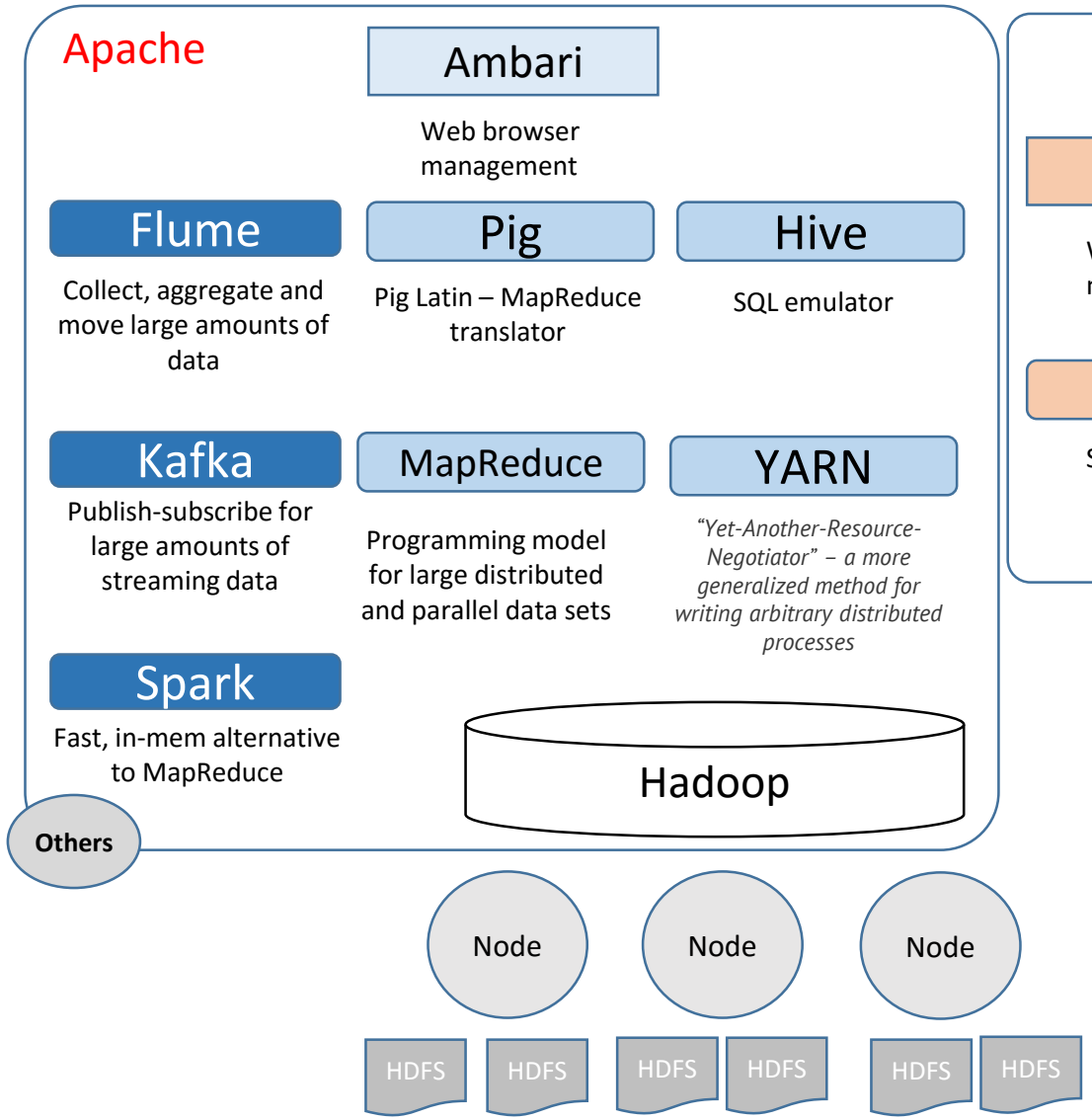
In other words – data is distributed, but more importantly, so is the "query". It is run separately on each computer and results are returned and then aggregated.

Buzzword: Hash buckets & IDs

Example: Teradata – a powerful proprietary distributed and parallel processing database.

In a nutshell, Hadoop is excellent for storing unstructured data and running parallel transformations to 'sanitize' incoming data, where Teradata / RDBMSs (Relational Database Management System) excel at executing complex queries quickly. Though there are extensions and methods to improve Hadoop's performance on large complex queries, it is best for ingestion, storing, filtering and aggregating.

Some Well Known Extensions of Hadoop



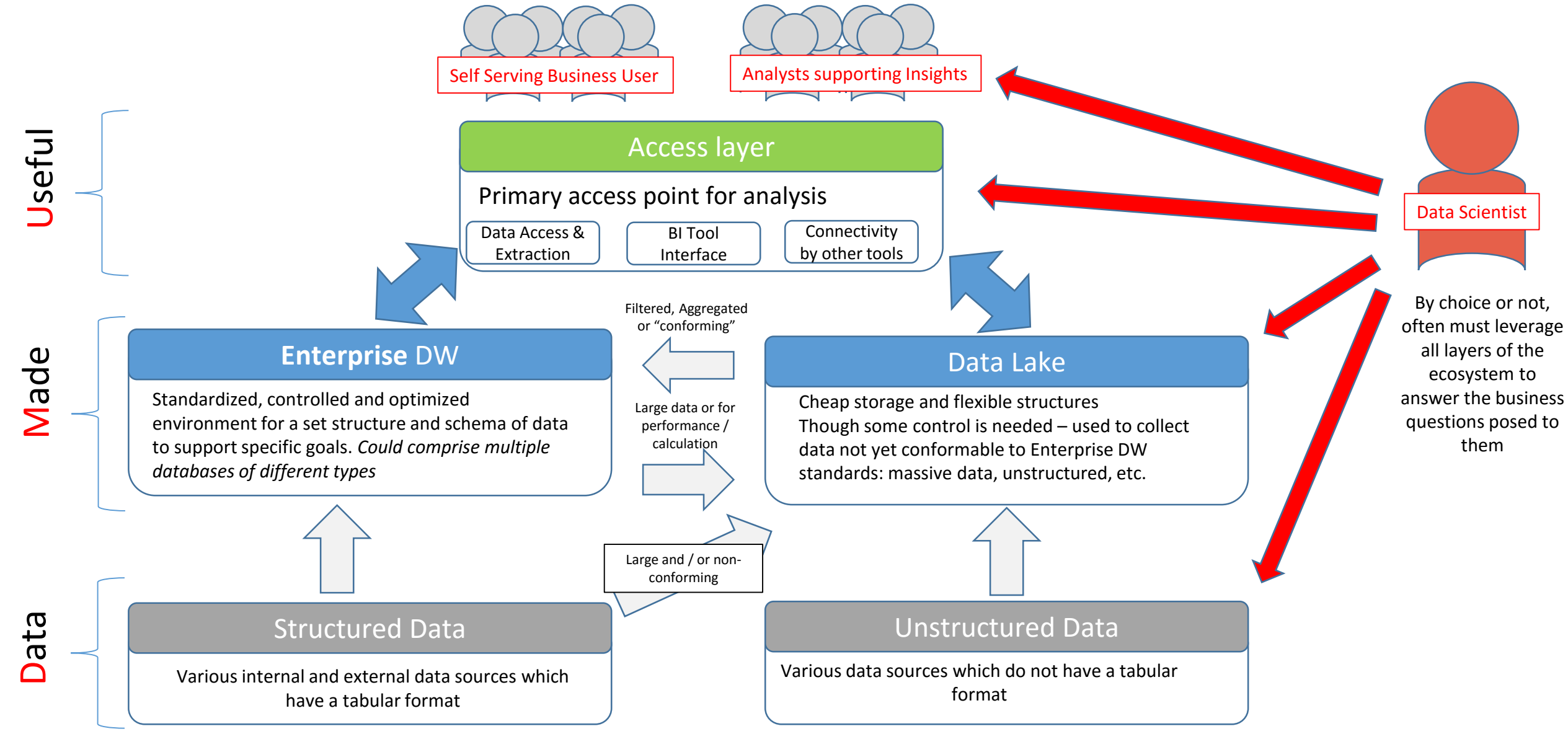
https://hadoopecosystemtable.github.io/

High level survey of a selection of databases

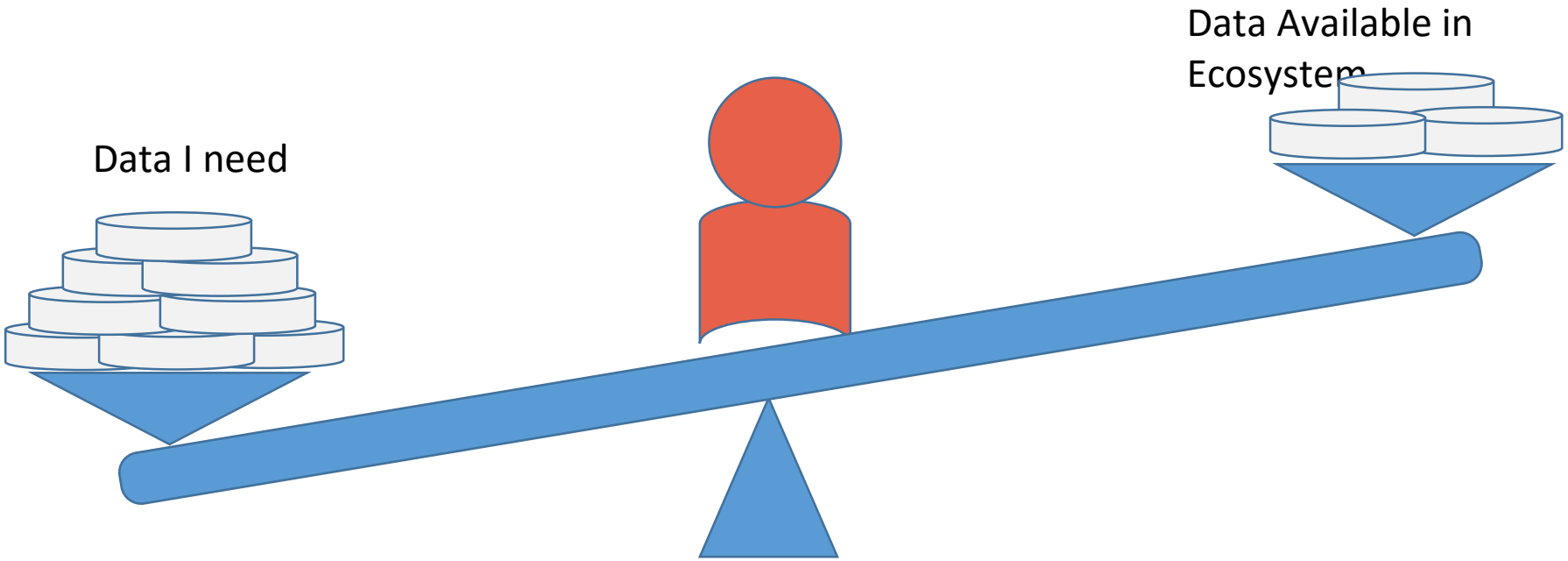
Year	System/ Paper	Scale to 1000s	Primary Index	Secondary Indexes	Transactions	Joins/ Analytics	Integrity Constraints	Views	Language/ Algebra	Data model	my label
1971	RDBMS	o	✔	✔	✔	✔	✔	✔	✔	tables	sql-like
2003	memcached	✔	✔	o	o	o	o	o	o	key-val	nosql
2004	MapReduce	✔	o	o	o	✔	o	o	o	key-val	batch
2005	CouchDB	✔	✔	✔	record	MR	o	✔	o	document	nosql
2006	BigTable/Hbase	✔	✔	✔	record	compat. w/MR	/	o	o	ext. record	nosql
2007	MongoDB	✔	✔	✔	EC, record	o	o	o	o	document	nosql
2007	Dynamo	✔	✔	o	o	o	o	o	o	ext. record	nosql
2008	Pig	✔	o	o	o	✔	/	o	✔	tables	sql-like
2008	HIVE	✔	o	o	o	✔	✔	o	✔	tables	sql-like
2008	Cassandra	✔	✔	✔	EC, record	o	✔	✔	o	key-val	nosql
2009	Voldemort	✔	✔	o	EC, record	o	o	o	o	key-val	nosql
2009	Riak	✔	✔	✔	EC, record	MR	o			key-val	nosql
2010	Dremel	✔	o	o	o	/	✔	o	✔	tables	sql-like
2011	Megastore	✔	✔	✔	entity groups	o	/	o	/	tables	nosql
2011	Tenzing	✔	o	o	o	o	✔	✔	✔	tables	sql-like
2011	Spark/Shark	✔	o	o	o	✔	✔	o	✔	tables	sql-like
2012	Spanner	✔	✔	✔	✔	?	✔	✔	✔	tables	sql-like
2013	Impala	✔	o	o	o	✔	✔	o	✔	tables	sql-like

Ref – Bill Howe, University of Washington

Enterprise Data Ecosystem – High level DMU



Curse of the Data Scientist



Trends in Database Systems

- There is still use for “basic” relational databases
- However, it is becoming easier to use flexible, cloud based (think accessible, distributed and scalable) storage systems
- As business owners latch on to buzz words in the data science industry and as companies “exhaust” the information they can get out of more traditional data, there is a push for using more “advanced” data storing, manipulating and analytical techniques.
- You will find many extensions of tools related to distributed data systems and cloud based systems
- handing ling of semi/unstructured data
- the need to process massive amounts of data
- reduce data management time and increase allowed analytical insight time - solving the challenge of navigating through an enterprise of disparate data sources and manipulating it to get what you need
- maintaining some sort of definitional integrity
- Faster access and delivery of results / insights
- Pushing the calculations and analytics to the data not bringing the data to the analytics.

Apache Hadoop	Revolution R	Teradata	Google Cloud Datastore	Amazon Web Services	Many Others
Apache Spark	Cloudera	Vertica	BigQuery and BigTable	Looker	
Apache Kafka	Hortonworks	Alteryx	Github	Tamr	

What is MySQL

In short, **MySQL** is an open source database and **MySQL workbench** is a “GUI” which enables database management

“Customer” Facing

Point of interaction + Production systems. Platforms and applications directly touching customers / end users.

Examples:

- Billing Systems
- Websites
  - With collection steps
  - With tagging
- Survey’s and Forms
- Support Systems (chat, phone, email, etc.)
- Etc.

Save the data in a meaningful way to enable historical reference, centralization and speed of access

Through access of the data, munge, massage, merge and augment the data to inform provide better structure or inform an inquiry

Examples:

- **MySQL + MySQL Workbench**
- SQL Server
- MS Access
- Oracle
- Teradata
- Hadoop
- Etc.

Analysis

When do you use MySQL / MySQL Workbench?

Different Companies operate in various ways when it comes to the spread of responsibility of data base management and information management vs. analysis and BI. You may never need to use MySQL or any other database other than know how to “query” data from it with a designated tool. Here are some instances where you may want to use MySQL

Leverage “larger” data sets	• Stores tables with millions of rows, easily • Enables full database development interim table development, indexing, etc.
Centralizing and organizing	• Allows SQL to be leveraged – very efficient when manipulating tabular data • Don’t keep data files everywhere, make databases with similar data tables
Supporting other analyses	• Access data with other tools of interest like R, SAS, other databases, etc. • If on Server, allow others to access your data for collaborative projects or to remove yourself from the manual step of delivery

A Quick Start – Optional

Downloading / install:

- http://dev.mysql.com/downloads/workbench/
- You will need to make a free account
- Install the default is fine, but if you run into issues just focus on MySQL server, MySQL workbench and what ever connectors are available.

Starter scripts and common codes:

- Code Bank in Google Docs Shared folder: https://drive.google.com/?tab=mo&authuser=0#folders/0B-Ps9CP_qLYpOU9uV2tweWdQaGM
  - Load file code gives an example of how to create a table skeleton and load data into it
  - Common MySQL codes are just base common statements you may use when loading and manipulating data
  - RCODE_MySQL has some basic R code that allows you to connect to the MySQL instance with R through an ODBC connection
    - you will need to ensure you have the correct MySQL ODBC drive installed and the data source established – I can help with this as needed

Go to MySQL Workbench – Walk through of interface and examples.