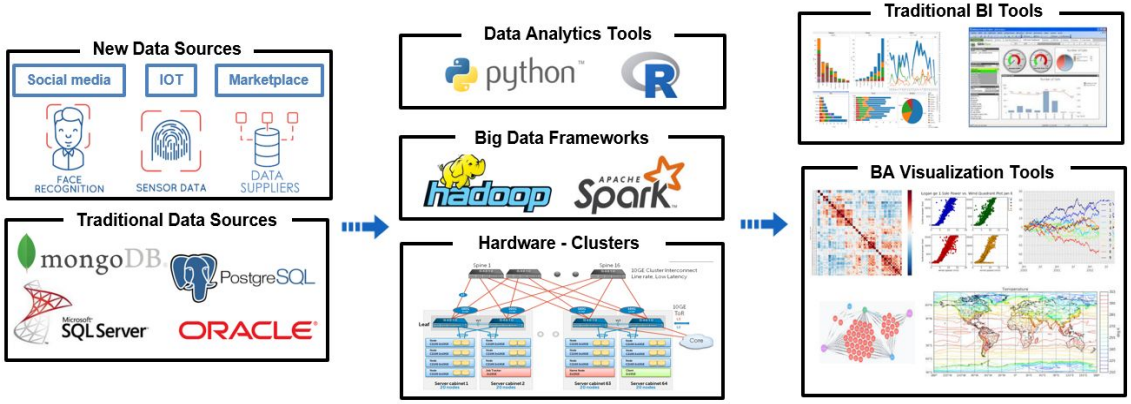


Data Analytics for Senior IT Managers (Day 2)

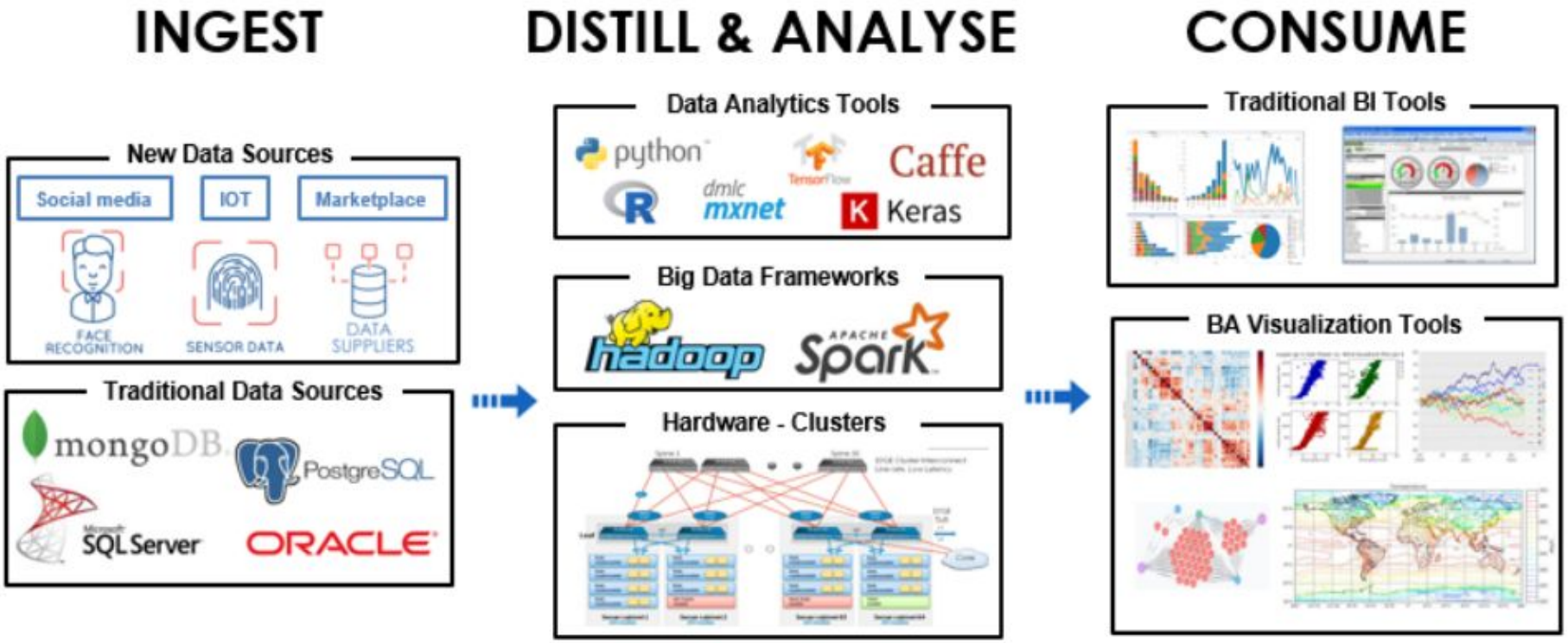
Laurence Liew
Dr Guo Lei
Jeanne Choo

NUS National University of Singapore | School of Continuing and Lifelong Education

Analytics Journey

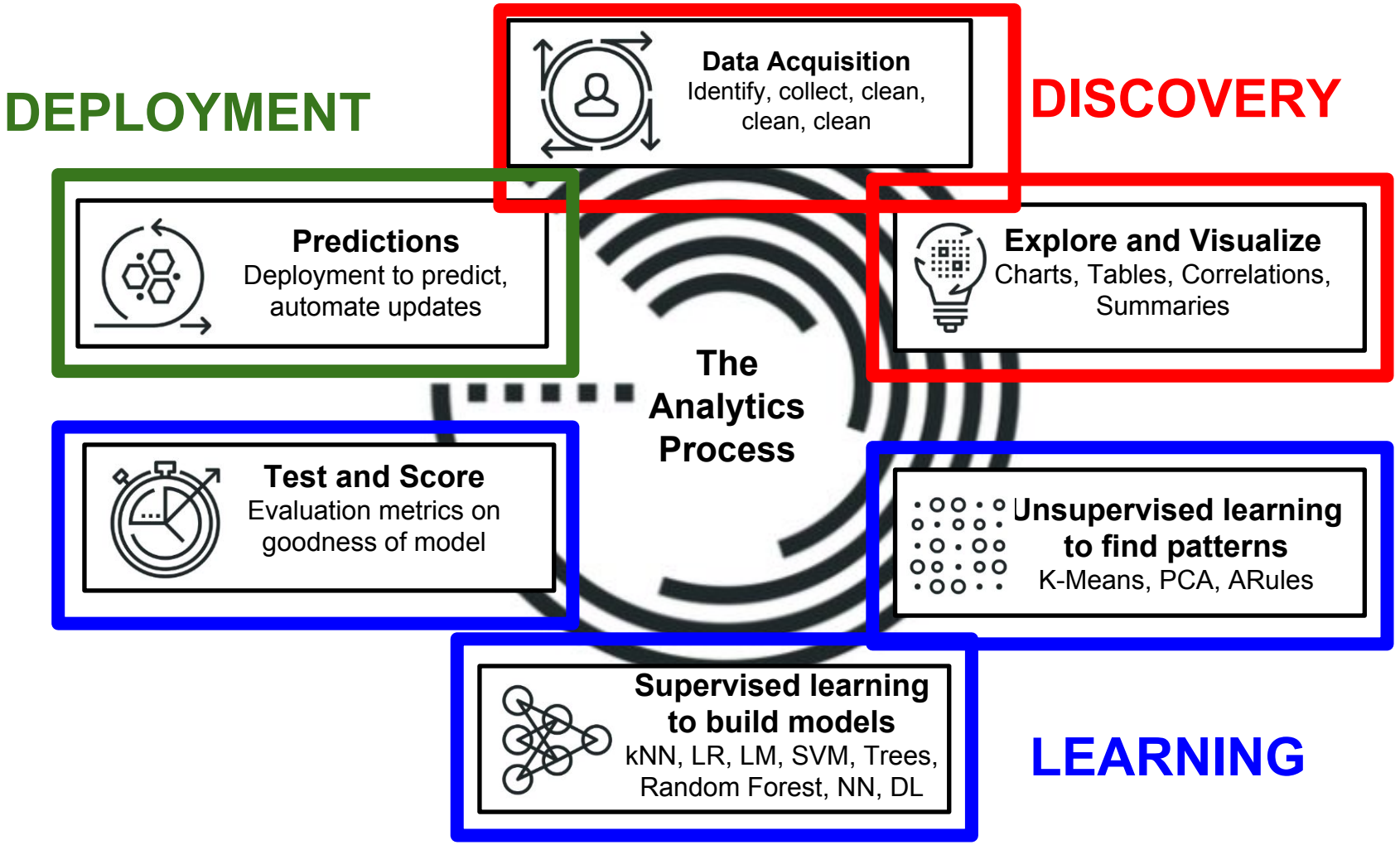
WHAT DID WE LEARN YESTERDAY?

BIG DATA ANALYTICS OVERVIEW

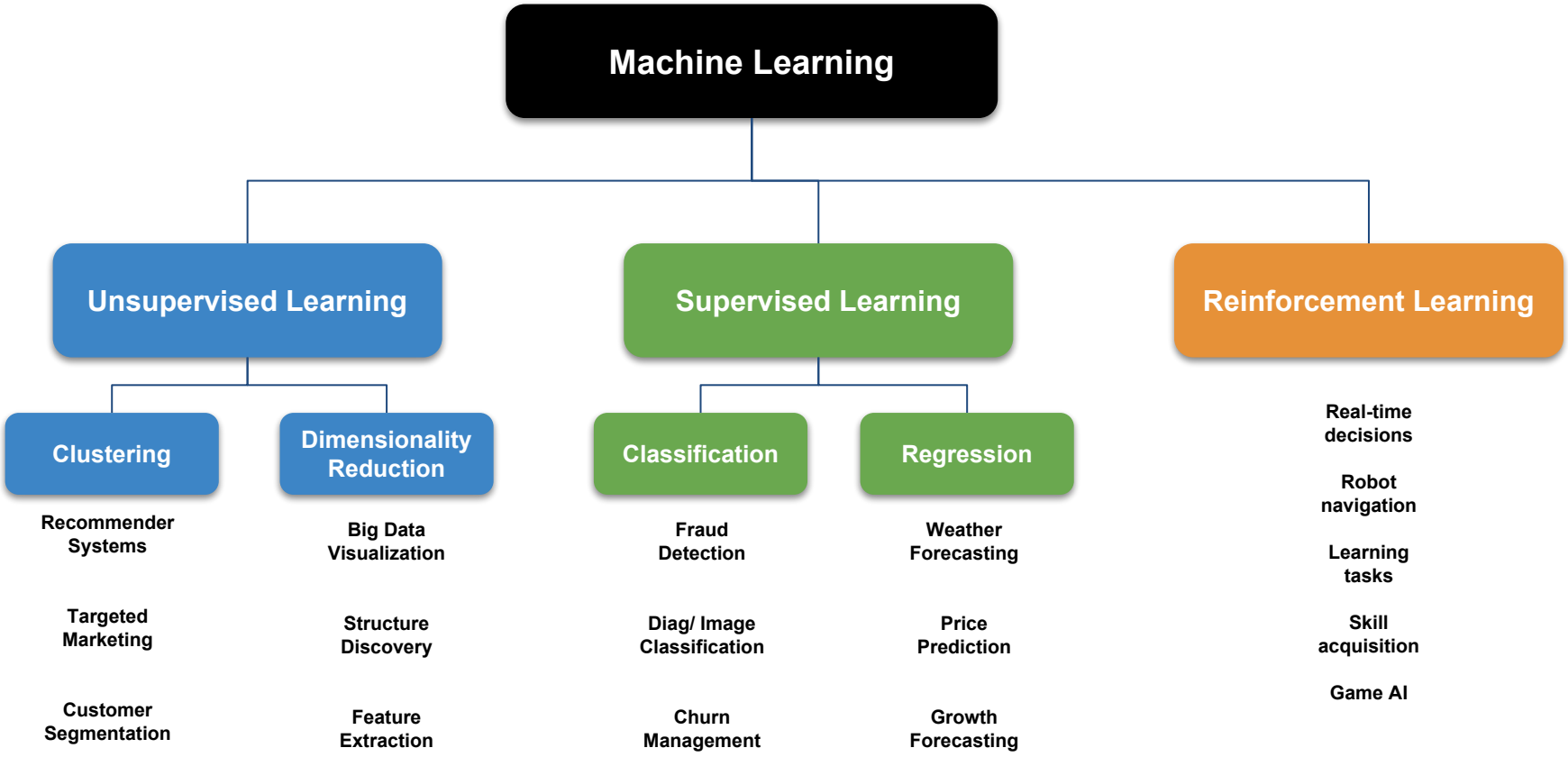


We are going to dig deeper into the infrastructure required to support the analytics process.

THE ANALYTICS PROCESS



MACHINE LEARNING OVERVIEW



FIVE QUESTIONS THAT DATA SCIENCE ANSWERS

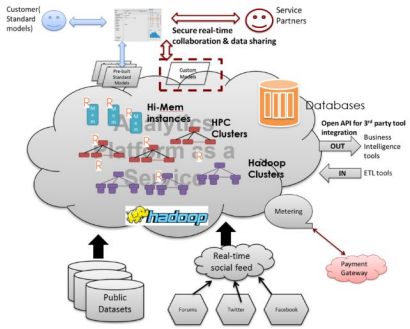
- Is this X or Y?
- Is this abnormal?
- How much does it cost in the future? Or how many will there be?
- How are they organized?
- What should I do next?

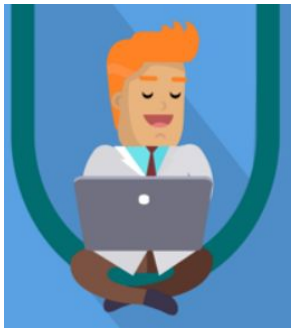
Each of these questions can be answered by an algorithm or recipe by using your data.

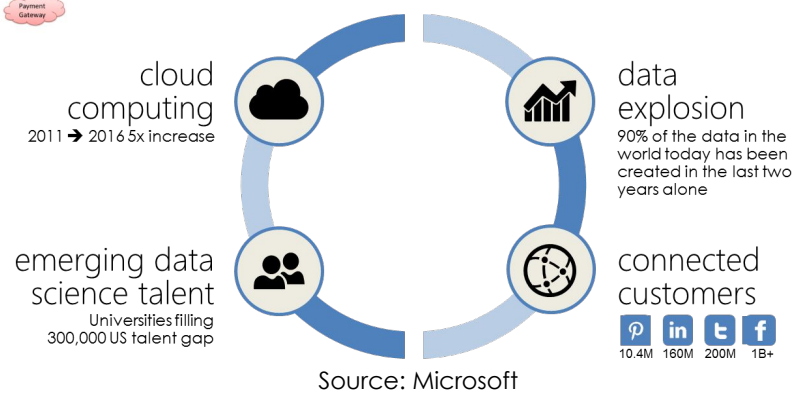
Innovations from the open source community

OPEN SOURCE ANALYTICS FOR INNOVATION

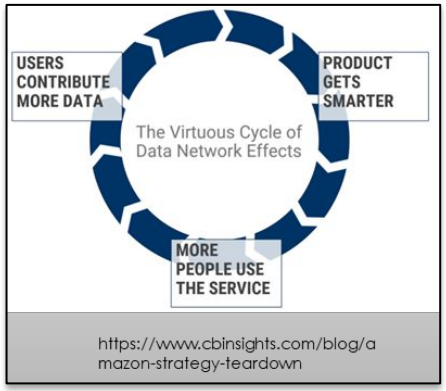
TRANSFORMATIONAL TRENDS



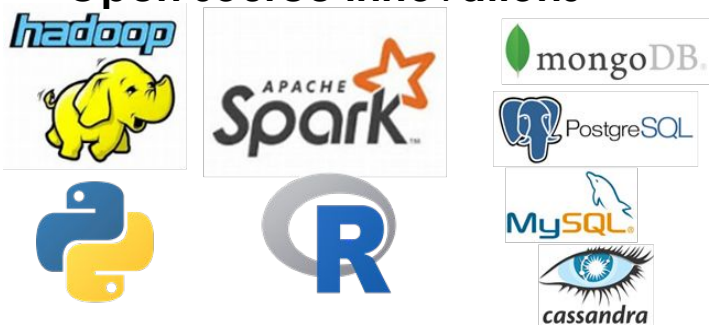




Source: Microsoft



Open source innovations



CLOSED VS OPEN INNOVATION…

SAS

SPSS

OPEN SOURCE ANALYTICS

NUS National University of Singapore

GROWTH OF OPEN SOURCE ANALYTICS TOOLS

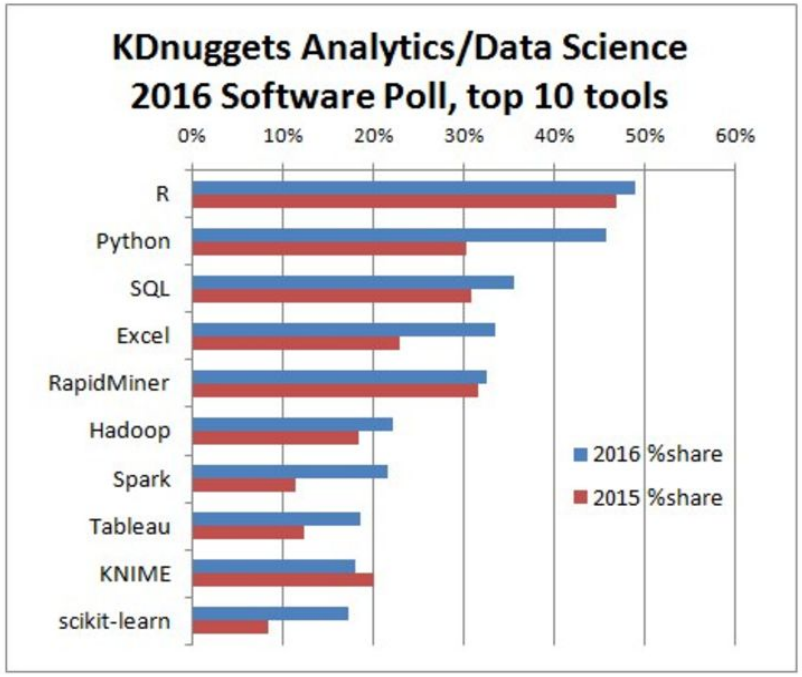


Fig 1: KDnuggets Analytics/Data Science 2016 Software Poll: top 10 most popular tools in 2016

Compared to 2015 KDnuggets Analytics/Data Science Poll results, the only newcomer in top 10 was scikit-learn, displacing SAS.

Source: KDnuggets™

New (in this poll) tools that received at least 1% share votes in 2016 were

- Anaconda, 16%
- Microsoft other ML/Data Science tools, 1.6%
- SAP HANA, 1.2%
- XLMiner, 1.2%

Among tools with at least 15 votes in 2015, the largest decline in 2016 was for the tools below, which includes probably a combination of decline of popularity for free tools like F# and lack of a voter drive for some of commercial tools this year.

- Ayasdi, down 85%, to 0.3% share from 2.0%
- Actian, down 83%, to 0.3% share from 2.0%
- Datameer, down 52%, to 0.4% share from 0.9%
- SAP Analytics, down 51%, to 1.5% share from 3.0%
- SAS Enterprise Miner, down 49%, to 5.6% from 10.9%
- Alteryx, down 46%, to 3.0% share from 5.6%
- F#, down 42%, to 0.4% share from 0.7%
- TIBCO Spotfire, down 36%, to 2.8% share from 4.3%
- JMP, down 36%, to 2.0% share from 3.1%

OPEN SOURCE ANALYTICS

Why are people adopting open source software for analytics?

- Fastest innovation
- All the last decade of big data, data science/analytics and now AI – is driven by the open source community and open source software
  - Hadoop, Spark, R, Python, Tensorflow, CNTK…
- If you want to be innovative, you HAVE to use open source software

OPEN SOURCE ANALYTICS

RISKS?

- Yes.. But proprietary software also have risks! What if the company closes?
- At least with open source software, you have the source code to continue to support yourself (or a new company/community will step up to continue the support if the adoption is there)
- Patent protection available from the bigger open source players like Red Hat
- Choose wisely.. Go with the crowd (the software with biggest adoption), but sometimes you push the edge and want to choose an outlier!
- Understand the type of open source license and respect the open source license and you will generally be fine

Reference only

POPULAR OPEN SOURCE LICENSES

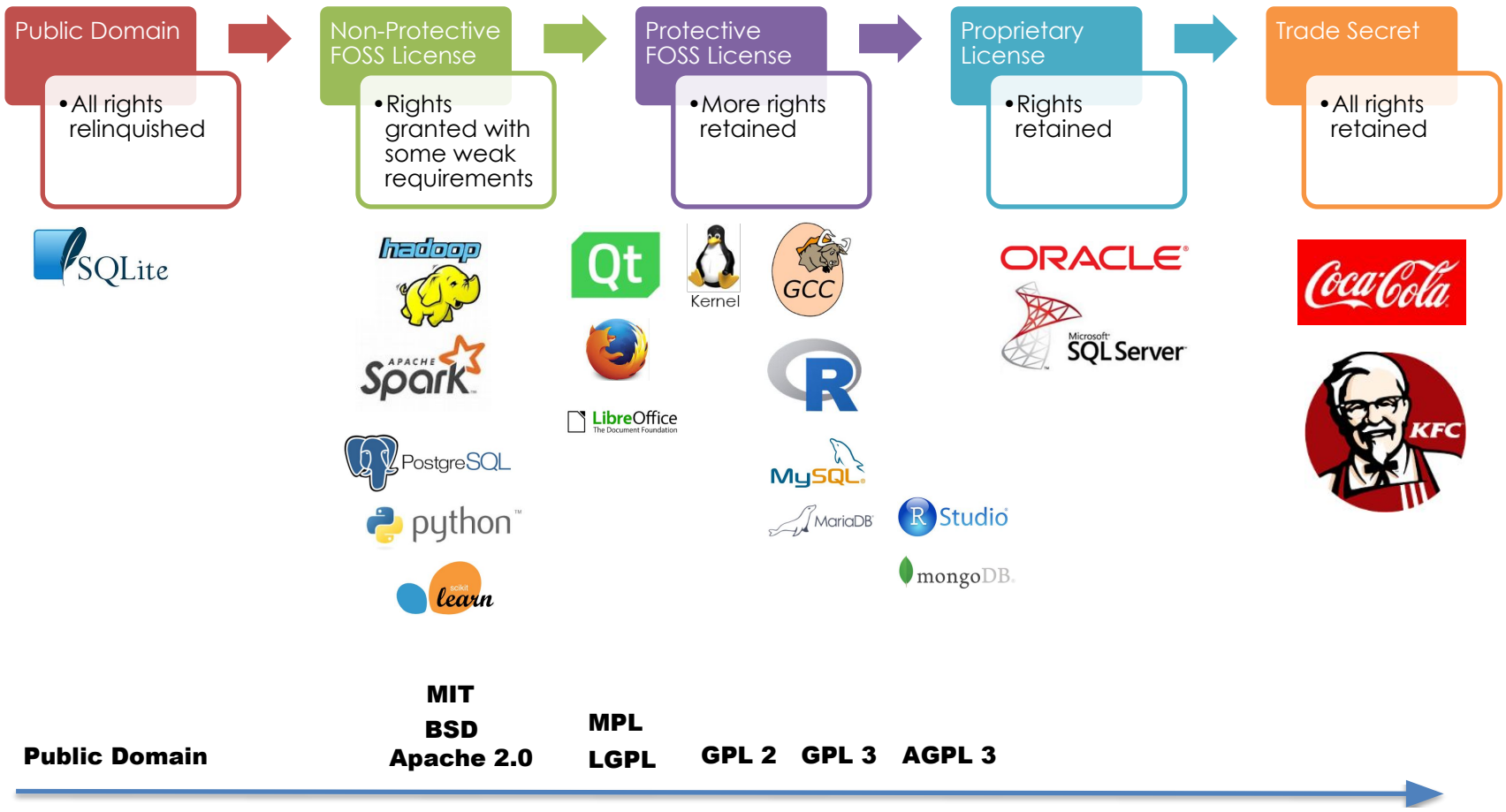
Popular Licenses

The following OSI-approved licenses are popular, widely used, or have strong communities:

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

Source: https://opensource.org/licenses

LEVEL OF PERMISSIVENESS



Reference only

HOW DOES A FOSS LICENSE LOOK LIKE?

APACHE, MIT, BSD, MPL

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 3.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 3.6.1 alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright © 2001-2017 Python Software Foundation; All Rights Reserved" are retained in Python 3.6.1 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 3.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 3.6.1.

4. PSF is making Python 3.6.1 available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 3.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 3.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 3.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python 3.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

You can use it anyhow as long as you maintain and acknowledge this license.

The Mozilla Foundation

Licensing & Trademarks

Source Code

All Mozilla software is open source and free software. This means that it is not only available for download free of charge, but you have access to the source code and may modify and redistribute our software subject to certain restrictions.

For more details, please read about our source code license and software licensing policy.

Trademarks

Although our code is free, it is very important that we strictly enforce our trademark rights, in order to be able to protect our users against people who use the marks to commit fraud. Our trademarks include, among others, the names Mozilla®, mozilla.org®, Firefox®, Thunderbird®, Bugzilla™, Camino®, Sunbird®, SeaMonkey®, and XUL™, as well as the Mozilla logo, Firefox logo, Thunderbird logo and the red lizard logo. (The full list is in the Mozilla Trademark Policy.) This means that, while you have considerable freedom to redistribute and modify our software, there are tight restrictions on your ability to use the Mozilla names and logos in ways which fall in the domain of trademark law, even when built into binaries that we provide.

You can use it anyhow as long as you maintain and acknowledge this license AND respect their TRADEMARKS: names and logo.

Reference only

HOW DOES A FOSS LICENSE LOOK LIKE?

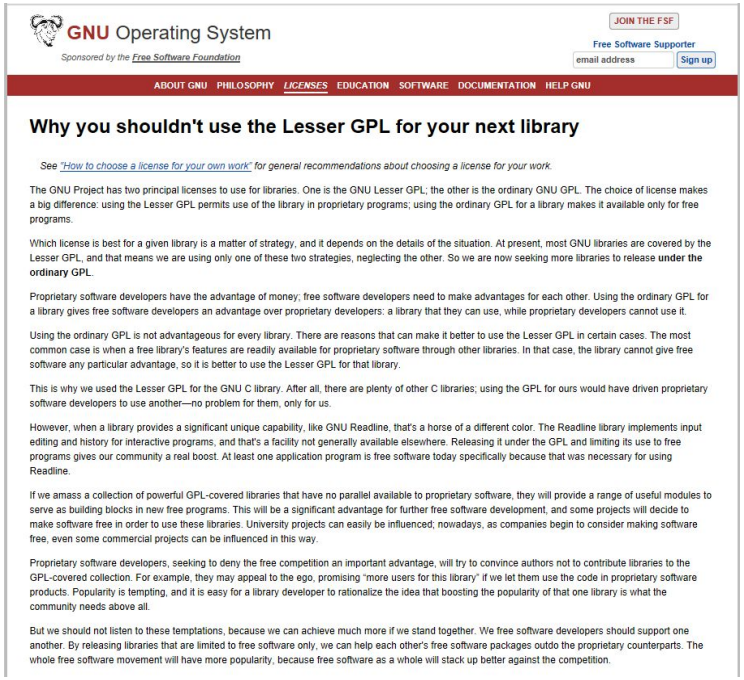
LGPL, GPL

The Foundations of the GPL

Nobody should be restricted by the software they use. There are four freedoms that every user should have:

- the freedom to use the software for any purpose,
- the freedom to change the software to suit your needs,
- the freedom to share the software with your friends and neighbors, and
- the freedom to share the changes you make.

Removes the need for you to open your own extensions when you build upon LGPL software. Eg. Some versions of Mysql/MariaDB client libraries are LGPL to allow you to distribute your applications which may use those client libraries.



GNU Operating System

Sponsored by the Free Software Foundation

JOIN THE FSF

Free Software Supporter

email address Sign up

ABOUT GNU PHILOSOPHY LICENSES EDUCATION SOFTWARE DOCUMENTATION HELP GNU

Why you shouldn't use the Lesser GPL for your next library

See "How to choose a license for your own work" for general recommendations about choosing a license for your work.

The GNU Project has two principal licenses to use for libraries. One is the GNU Lesser GPL; the other is the ordinary GNU GPL. The choice of license makes a big difference: using the Lesser GPL permits use of the library in proprietary programs; using the ordinary GPL for a library makes it available only for free programs.

Which license is best for a given library is a matter of strategy, and it depends on the details of the situation. At present, most GNU libraries are covered by the Lesser GPL, and that means we are using only one of these two strategies, neglecting the other. So we are now seeking more libraries to release **under the ordinary GPL**.

Proprietary software developers have the advantage of money; free software developers need to make advantages for each other. Using the ordinary GPL for a library gives free software developers an advantage over proprietary developers: a library that they can use, while proprietary developers cannot use it.

Using the ordinary GPL is not advantageous for every library. There are reasons that can make it better to use the Lesser GPL in certain cases. The most common case is when a free library's features are readily available for proprietary software through other libraries. In that case, the library cannot give free software any particular advantage, so it is better to use the Lesser GPL for that library.

This is why we used the Lesser GPL for the GNU C library. After all, there are plenty of other C libraries; using the GPL for ours would have driven proprietary software developers to use another—no problem for them, only for us.

However, when a library provides a significant unique capability, like GNU Readline, that's a horse of a different color. The Readline library implements input editing and history for interactive programs, and that's a facility not generally available elsewhere. Releasing it under the GPL and limiting its use to free programs gives our community a real boost. At least one application program is free software today specifically because that was necessary for using Readline.

If we amass a collection of powerful GPL-covered libraries that have no parallel available to proprietary software, they will provide a range of useful modules to serve as building blocks in new free programs. This will be a significant advantage for further free software development, and some projects will decide to make software free in order to use these libraries. University projects can easily be influenced; nowadays, as companies begin to consider making software free, even some commercial projects can be influenced in this way.

Proprietary software developers, seeking to deny the free competition an important advantage, will try to convince authors not to contribute libraries to the GPL-covered collection. For example, they may appeal to the ego, promising "more users for this library" if we let them use the code in proprietary software products. Popularity is tempting, and it is easy for a library developer to rationalize the idea that boosting the popularity of that one library is what the community needs above all.

But we should not listen to these temptations, because we can achieve much more if we stand together. We free software developers should support one another. By releasing libraries that are limited to free software only, we can help each other's free software packages outdo the proprietary counterparts. The whole free software movement will have more popularity, because free software as a whole will stack up better against the competition.

Reference only

GPL VS LGPL

- Released under GPL v2
- Internal usage is free
  - GPL license only affects code that you distribute to other parties
    - You may wish to check with your own legal if your develop and distribute code to other agencies, departments and/or subsidaries does it mean “internal use” or “other parties”
  - So for internal programs which you own all the copyrights, there is essentially no risk in using GPL software
- Distributing an application with a MariaDB connector/client (not the server)
  - If your software is free software – no issue
  - If you use a standard framework – ODBC, JDBC – no issue.
  - Recommend to use MariaDB’s LGPL client libraries anyways
- Distributing a proprietary application with MariaDB/MySQL Server
  - Use MariaDB/MySQL LGPL client libraries
  - Design your application works to work independently of MariaDB/MySQL, eg via ODBC/JDBC interface. The reason for this is that MariaDB would only be an optional, independent component in your software distribution and section 2 of the GPL explicitly allows this

> "In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License."

Reference only

COMMERCIAL FOSS LICENSE?

RED HAT
ENTERPRISE AGREEMENT
SINGAPORE

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE PURCHASING AND/OR USING SOFTWARE OR SERVICES FROM RED HAT. BY USING RED HAT SOFTWARE OR SERVICES, CLIENT SIGNIFIES ITS ASSENT TO AND ACCEPTANCE OF THIS AGREEMENT AND ACKNOWLEDGES IT HAS READ AND UNDERSTANDS THIS AGREEMENT. AN INDIVIDUAL ACTING ON BEHALF OF AN ENTITY REPRESENTS THAT HE OR SHE HAS THE AUTHORITY TO ENTER INTO THIS AGREEMENT ON BEHALF OF THAT ENTITY. IF CLIENT DOES NOT ACCEPT THE TERMS OF THIS AGREEMENT, THEN IT MUST NOT USE RED HAT SOFTWARE OR SERVICES. This Agreement incorporates those appendices at the end of this Agreement.

This Red Hat Enterprise Agreement, including all referenced appendices and documents located at URLs (the "Agreement"), is between Red Hat Asia Pacific Pte Ltd ("Red Hat") and the purchaser or user of Red Hat software and services who accepts the terms of this Agreement ("Client"). The effective date of this Agreement ("Effective Date") is the earlier of the date that Client signs or accepts this Agreement or the date that Client uses Red Hat's software or services.

8. Limitation of Liability and Disclaimer of Damages

8.1 Limitation of Liability. FOR ALL EVENTS AND CIRCUMSTANCES, RED HAT AND ITS AFFILIATES' AGGREGATE AND CUMULATIVE LIABILITY ARISING OUT OF OR RELATING TO THIS AGREEMENT AND ALL ORDER FORMS, INCLUDING WITHOUT LIMITATION ON ACCOUNT OF PERFORMANCE OR NON-PERFORMANCE OF OBLIGATIONS, REGARDLESS OF THE FORM OF THE CAUSE OF ACTION, WHETHER IN CONTRACT, TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STATUTE OR OTHERWISE WILL BE LIMITED TO DIRECT DAMAGES AND WILL NOT EXCEED THE AMOUNTS RECEIVED BY RED HAT DURING TWELVE (12) MONTHS IMMEDIATELY PRECEDING THE FIRST EVENT GIVING RISE TO LIABILITY, WITH RESPECT TO THE PARTICULAR ITEMS (WHETHER SOFTWARE, SERVICES OR OTHERWISE) GIVING RISE TO LIABILITY UNDER THE MOST APPLICABLE ORDERING DOCUMENT. [THE FORGOING LIMITATION SHALL NOT APPLY TO CLAIMS FOR BODILY INJURY (INCLUDING DEATH) AND DAMAGE TO TANGIBLE PERSONAL PROPERTY CAUSED BY THE NEGLIGENCE OF RED HAT OR ITS EMPLOYEES.]

8.2 Disclaimer of Damages. NOTWITHSTANDING ANYTHING TO THE CONTRARY CONTAINED IN THIS AGREEMENT OR AN ORDER FORM, IN NO EVENT WILL RED HAT OR ITS AFFILIATES BE LIABLE TO CLIENT OR ITS AFFILIATES FOR DAMAGES OTHER THAN DIRECT DAMAGES, INCLUDING, WITHOUT LIMITATION: ANY INCIDENTAL, CONSEQUENTIAL, SPECIAL, INDIRECT, EXEMPLARY OR PUNITIVE DAMAGES, WHETHER ARISING IN TORT (INCLUDING WITHOUT LIMITATION, NEGLIGENCE), CONTRACT, STATUTE OR OTHERWISE; OR ANY DAMAGES ARISING OUT OF OR IN CONNECTION WITH ANY MALFUNCTIONS, REGULATORY NON-COMPLIANCE, DELAYS, LOSS OF DATA, LOST PROFITS, LOST SAVINGS, INTERRUPTION OF SERVICE, LOSS OF BUSINESS, GOODWILL OR ANTICIPATORY PROFITS, EVEN IF RED HAT OR ITS AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. LIABILITY FOR THESE DAMAGES WILL BE LIMITED AND EXCLUDED EVEN IF ANY EXCLUSIVE REMEDY PROVIDED FOR IN THIS AGREEMENT FAILS OF ITS ESSENTIAL PURPOSE.

- More of a trademark on the logo.
- Software within have range of licenses.

Oracle Linux (OL, formerly known as **Oracle** Enterprise **Linux**) is a **Linux** distribution packaged and freely distributed by **Oracle**, available partially under the GNU General Public License since late 2006. It is compiled from **Red Hat** Enterprise **Linux** source code, replacing **Red Hat** branding by **Oracle's**.

WHY WOULD I LICENSE FOSS?

IF PUBLIC DOMAIN OR FOSS LICENSE ALLOWS YOU TO USE THE SOFTWARE FOR FREE, WHY WOULD YOU WANT TO BUY A LICENSE?

- Your company desires warranty of title and/or indemnity against claims of copyright infringement.
- You are using the software in a jurisdiction that does not recognize the public domain or FOSS license.
- You are using software in a jurisdiction that does not recognize the right of an author to dedicate their work to the public domain or FOSS.
- You want to hold a tangible legal document as evidence that you have the legal right to use and distribute the software
- Your legal department tells you that you have to purchase a license.

SUMMARY

- If you want to innovate with data science, accept and work with open source software
- Educate your legal on open source licenses
- The biggest business challenge I have faced??

Indemnity clause
Unlimited liability clause

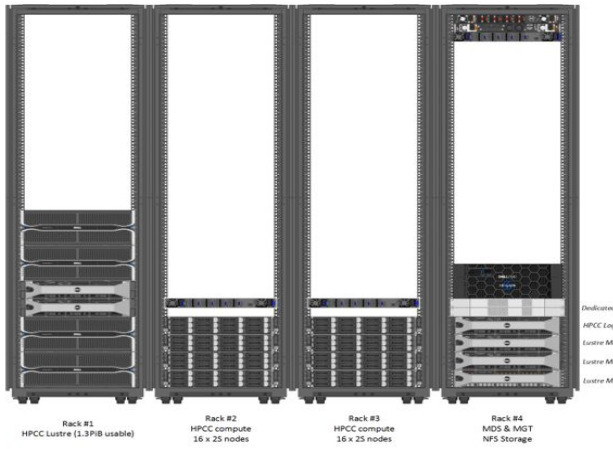
The infrastructure used for data analytics today

INFRASTRUCTURE

DATA CENTRES - GOTCHA!

Most of our data centres today are not designed/ready for HPC/AI/Big Data workloads!!!

1) Max power supply per rack - Max each C-form Amp: 32 amp (threshold safe practise: 12.8 amp for each power source)
2) Max floor loading per rack - Max Load 550kg



VS

	Rack #1	Rack #2	Rack #3	Rack #4
Power (Watts)	5,576	7,302	7,302	3,269
Cooling (BTU/Hr)	19,022	24,500	24,500	11,102
Weight (Kg)	614	351	351	368
Amps (@240V)	23	30	30	14

DATA ENGINEERING & INFRASTRUCTURE

- Often, people forget about this piece. Without proper infrastructure and tools, difficult to do good and scalable data science work. Some of the popular stacks:

Tools	Purpose
Kakfa, Flume	Streaming data collection. Collects data in real-time and aggregate the data and feed it to a SQL or NoSQL database like Hadoop/Spark
Hadoop	Hadoop is a framework and ecosystem of tools to store and process data. Batch based and MapReduce framework from 2004 Google paper. Often used with Pig, Hive (SQL like query languages) instead of lower level Java to query the data.
Spark	Spark is a memory based data processing framework slowly replacing Hadoop as the preferred framework for big data analytics workloads. Interfaces via Scala, Python and R.
Cassandra, HBase	NoSQL data stores for large Hadoop scale datasets.

A REFERENCE BIG DATA ANALYTICS ARCHITECTURE

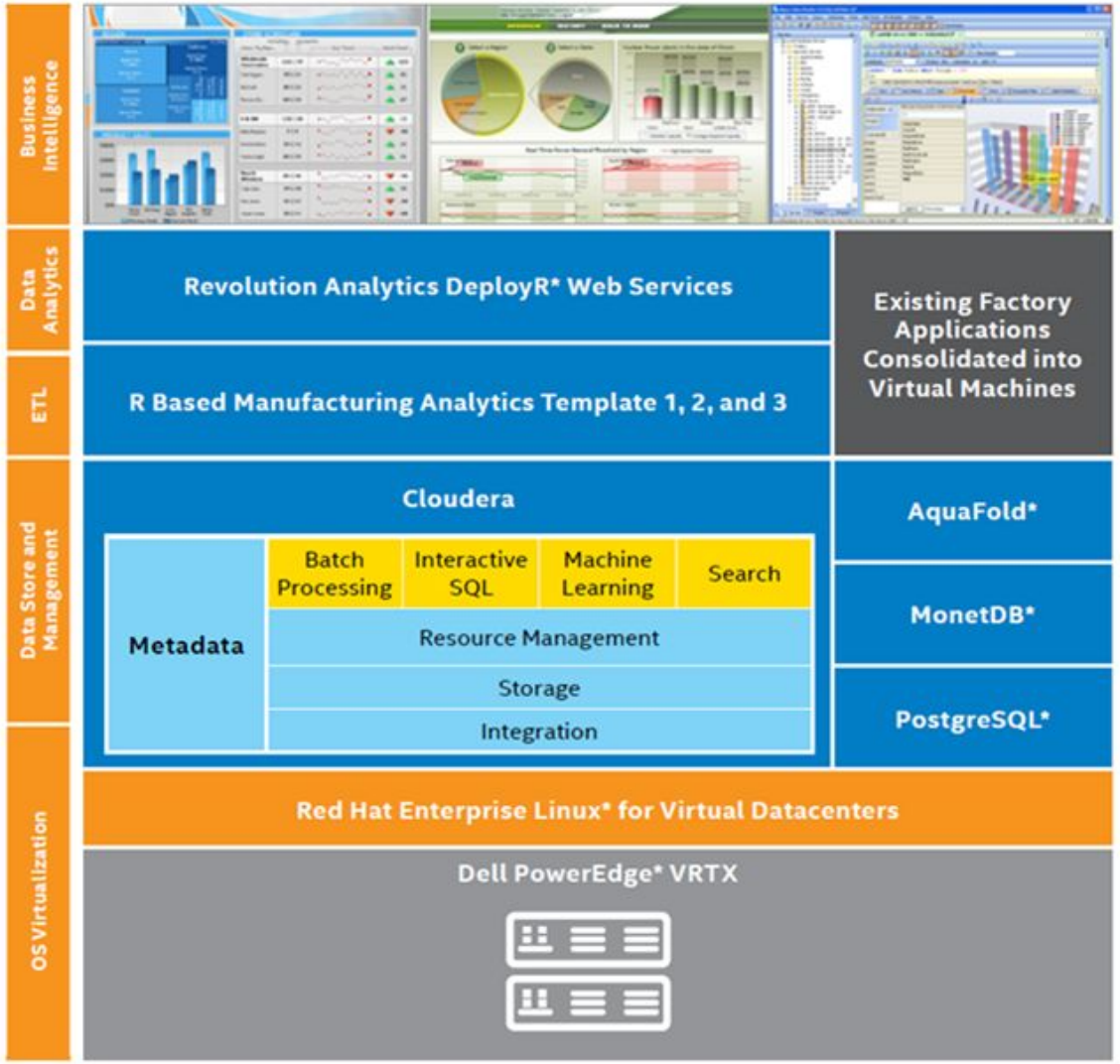
BI/Reporting

Analysis

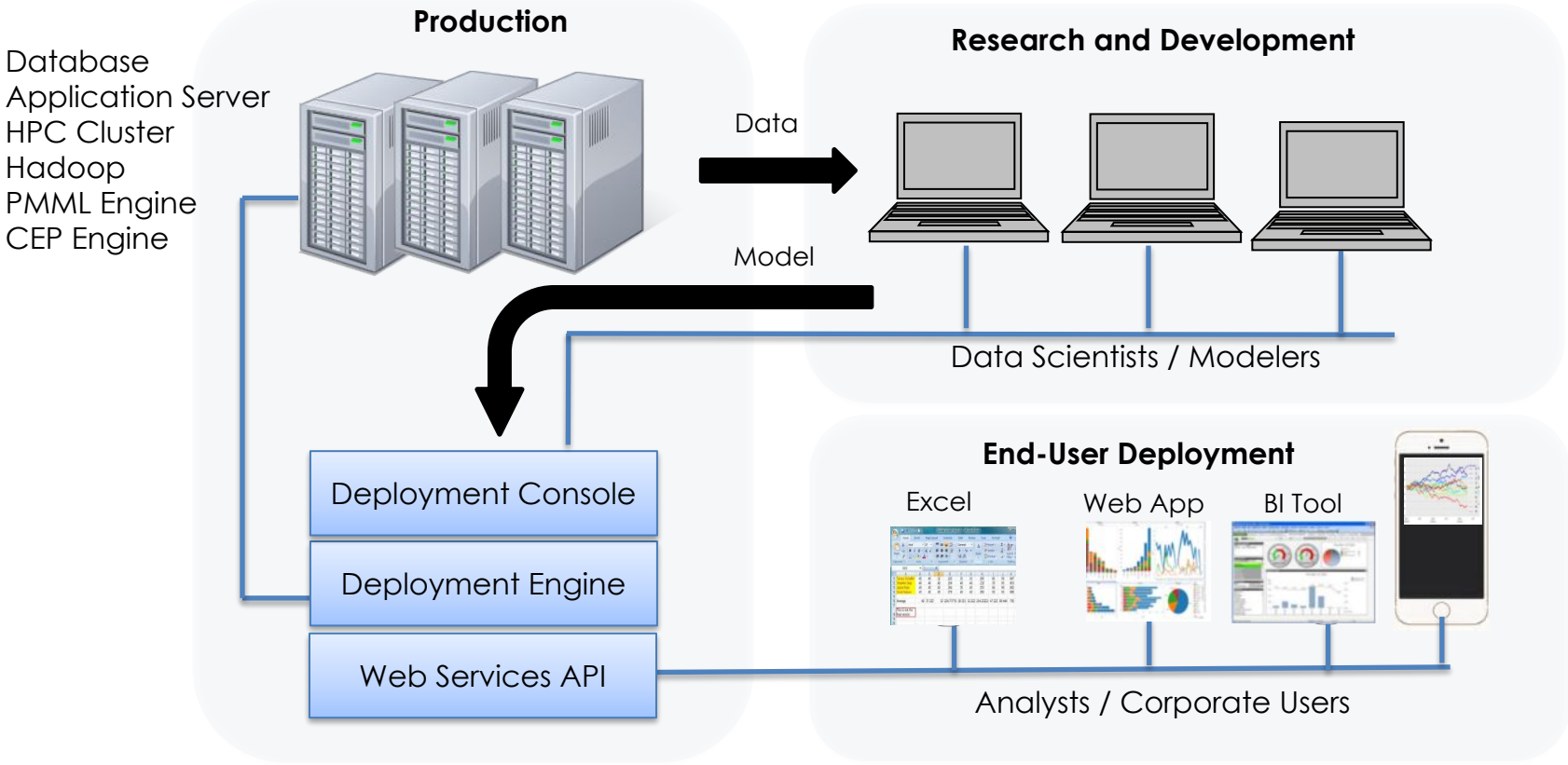
ETL

Data
SQL, Hadoop/Spark, NoSQL

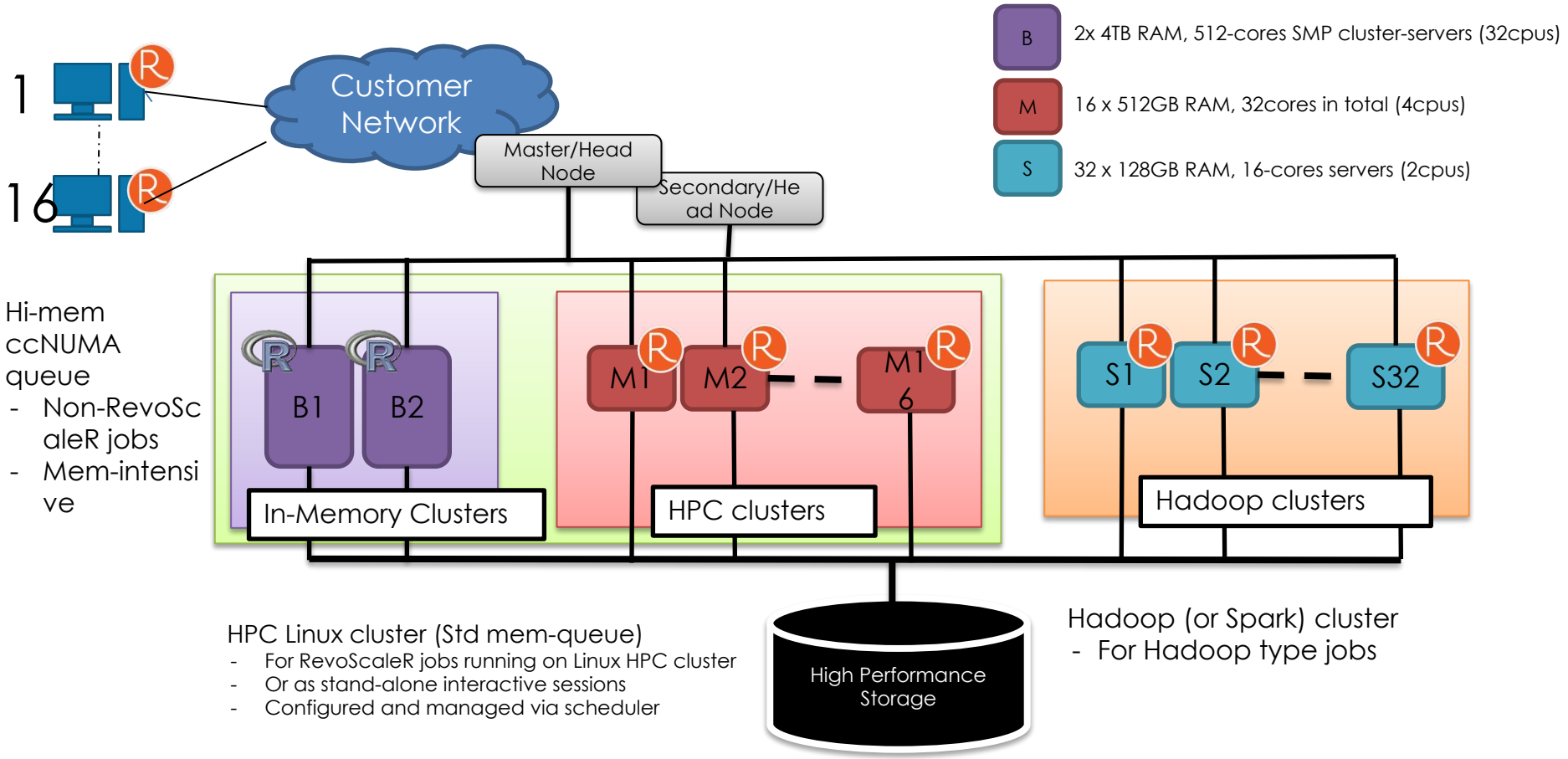
Hardware Infrastructure



TYPICAL ANALYTICS SETUP IN AN ENTERPRISE

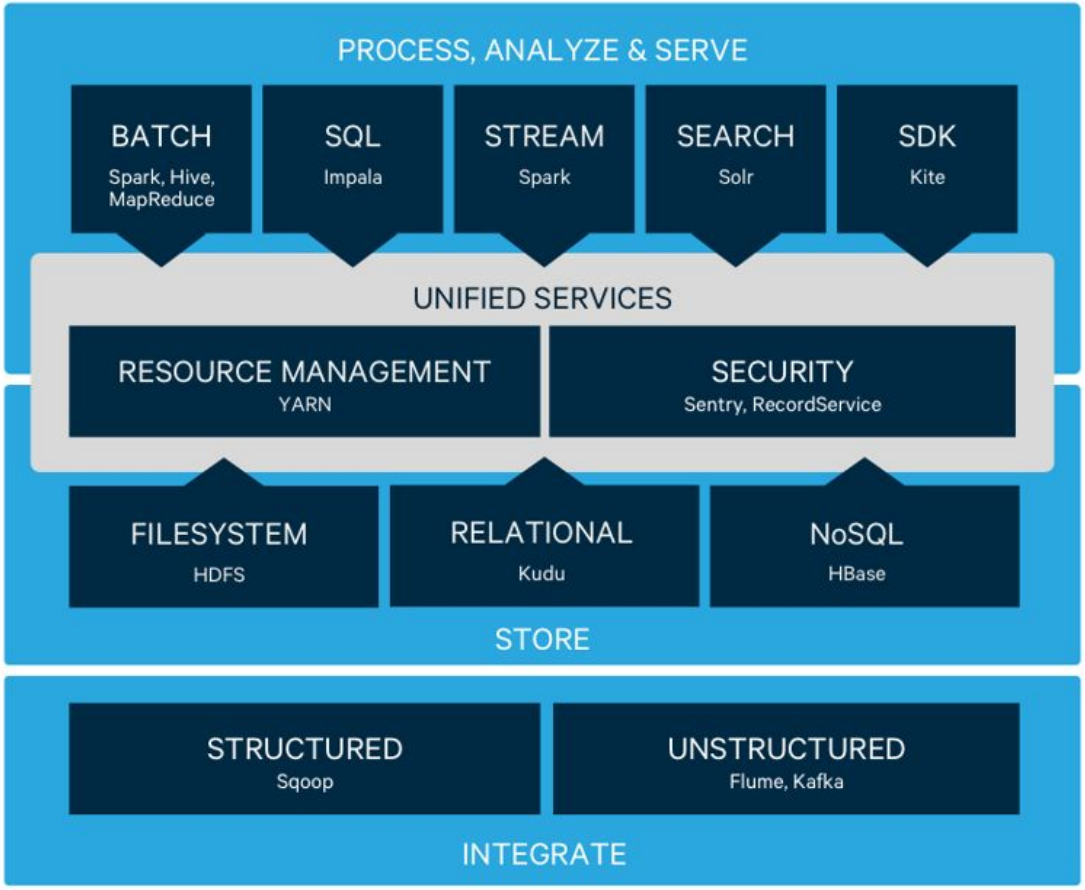


ACTUAL PROPOSED INFRASTRUCTURE TO AN AUSTRALIAN GOVT ORG

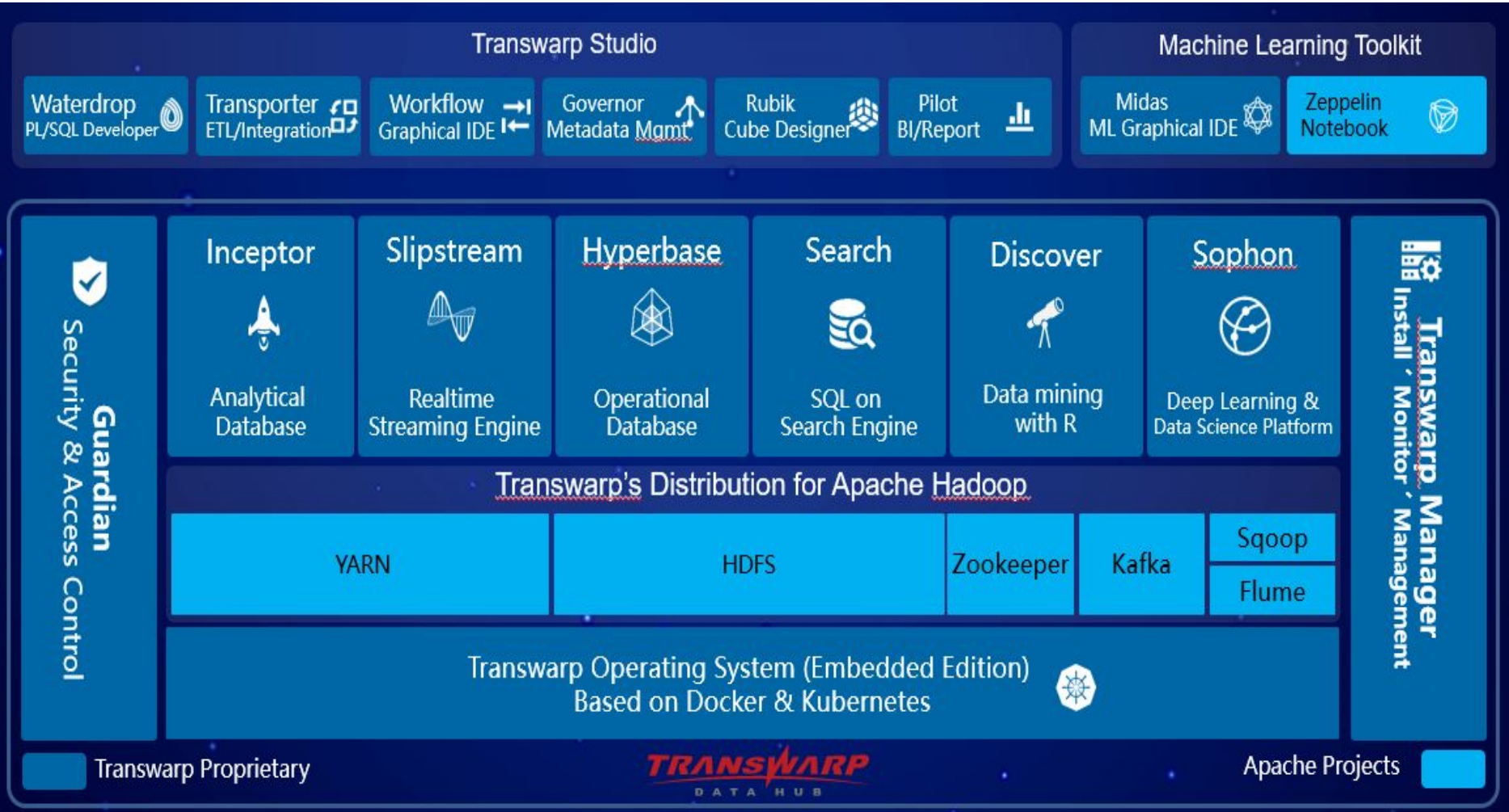


HADOOP

- Hadoop is an **ecosystem** of open source components
- Hadoop enables multiple types of analytic workloads to run on the same data, at the same time, at **massive scale** on **industry-standard x86 hardware**

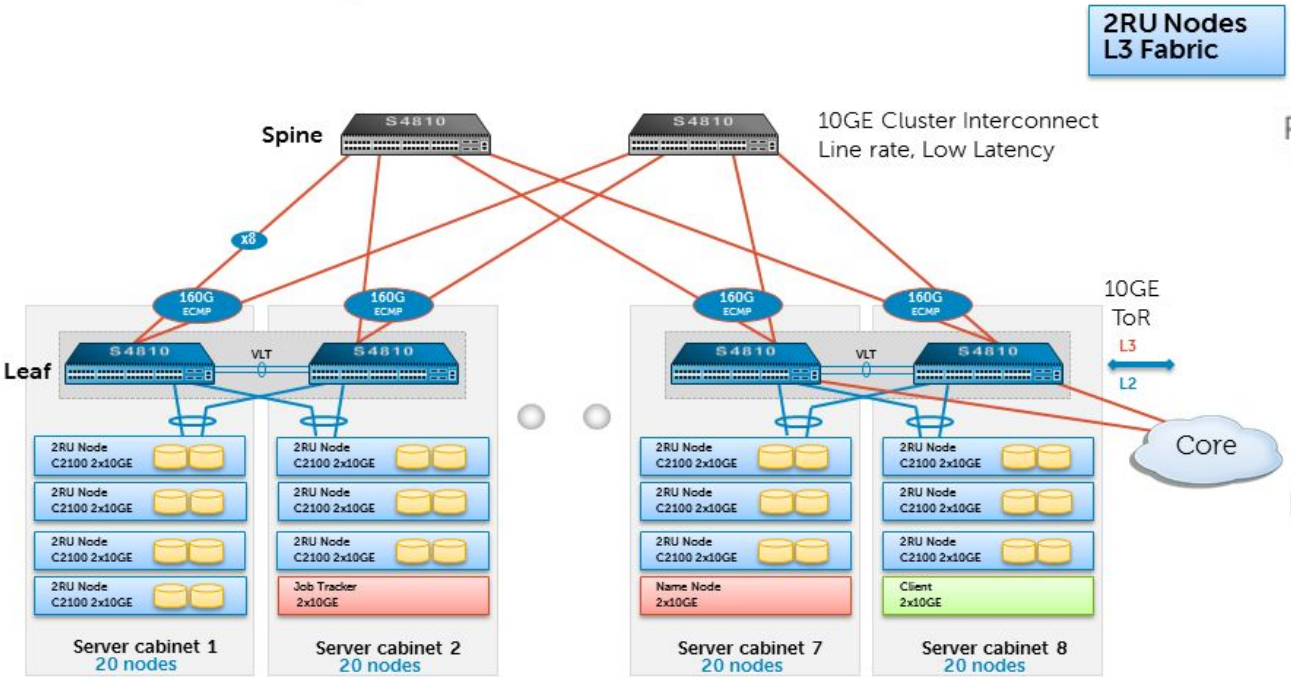


A MODERN END TO END ANALYTICS STACK



HADOOP CLUSTER HW

10GE Hadoop Cluster – 160 2RU node Starter



- **2RU Nodes** 2 x 10GE (C2100)
- 20 Nodes per rack
- 8 racks, **160 nodes**
- Scales to 64 racks, 1280 nodes

- Auto switch provisioning
- 2 rack pairs cross cabled
- 2.5:1 oversubscription @ ToR
- Core connected via Leaf w/ Layer 3

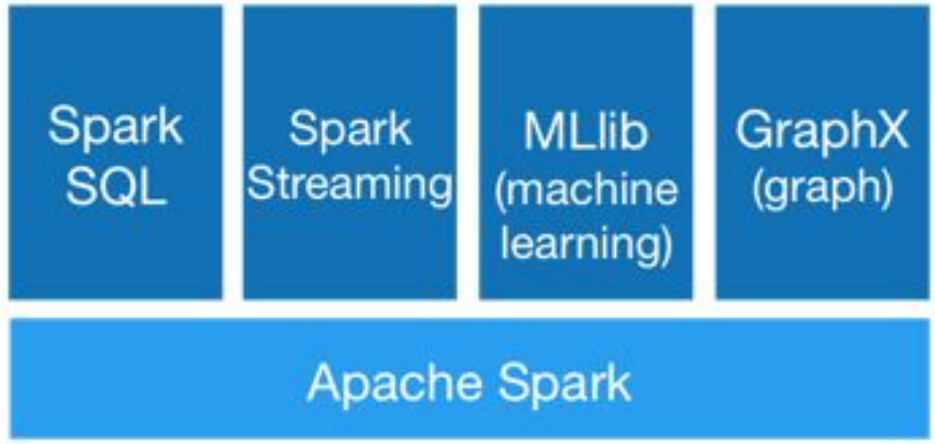
PowerEdge R720, R720XD
- Balanced Compute and Storage

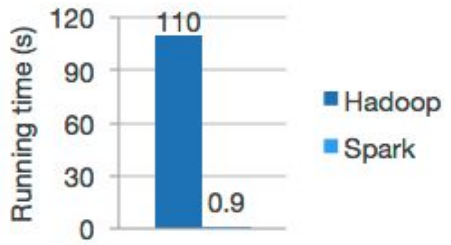
PowerEdge C8000
- Scale Out Computing
- Flexible Configuration

SPARK

APACHE SPARK™ IS A FAST AND GENERAL ENGINE FOR LARGE-SCALE DATA PROCESSING.

- Speed: 100x faster than Hadoop
- Ease of Use
  - Java, Python, R, Scala
- SQL + Streaming + ML
- Runs Everywhere
  - Hadoop, Mesos, standalone, Cloud
  - HDFS, Cassandra, Hbase, S3





Logistic regression in Hadoop and Spark

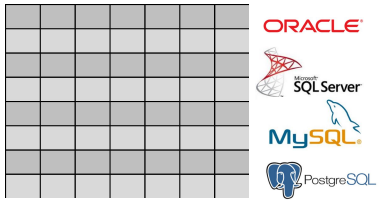
NOSQL

NO OR NEW SQL

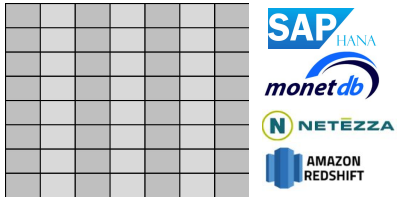
- Is No/New SQL better than traditional SQL databases?

NO !

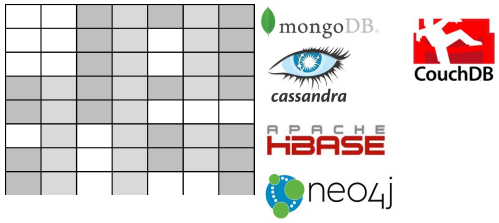
RDBMS/OLTP/Real-time



DW/OLAP/DSS/Batch

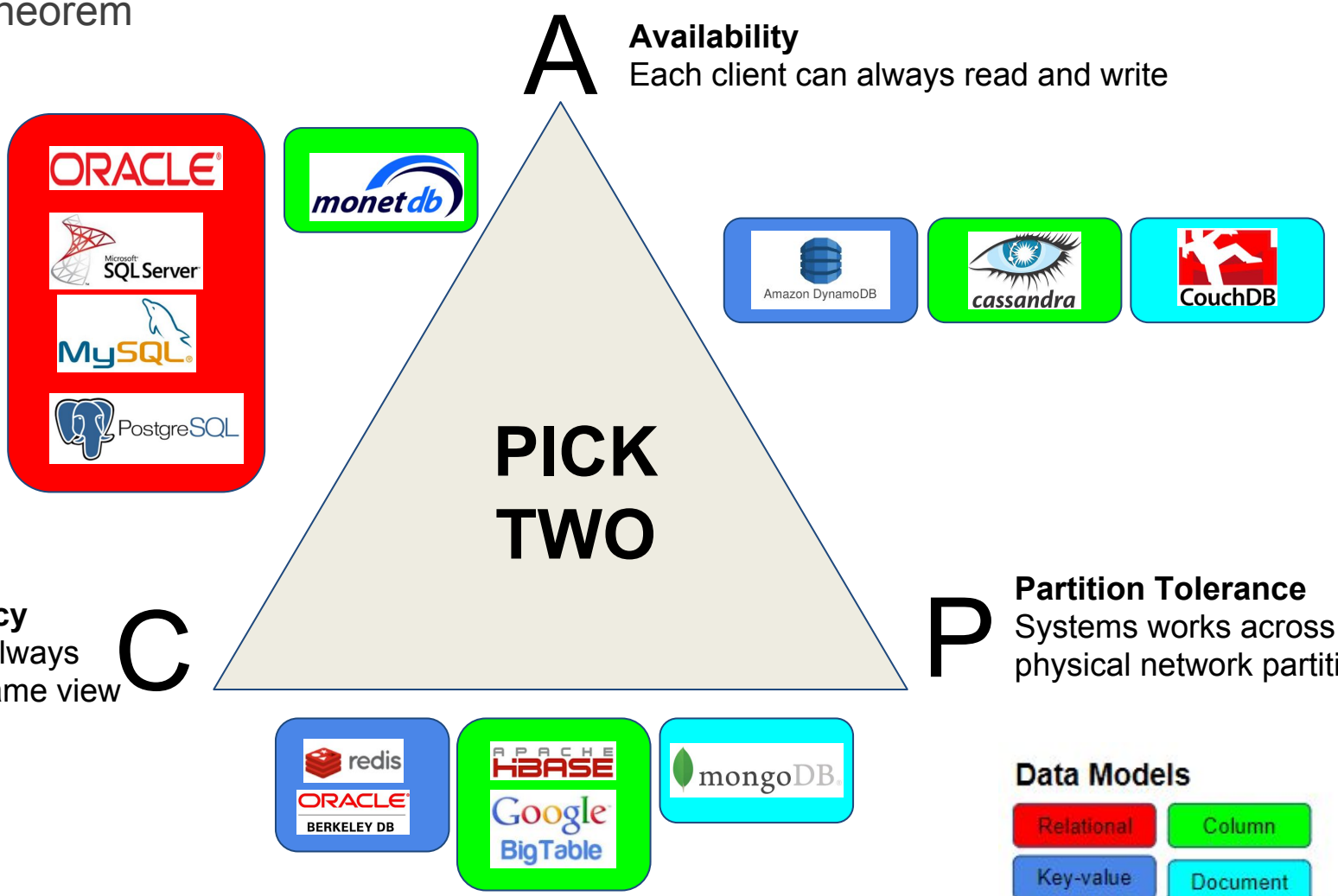


No/New SQL/BigData/Real/Batch



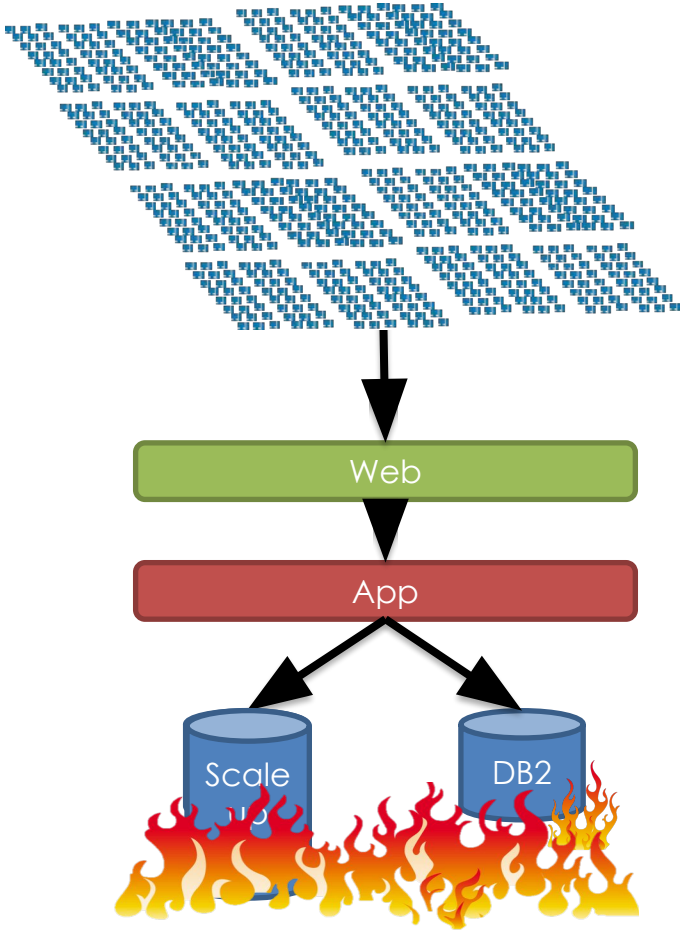
SQL vs NoSQL

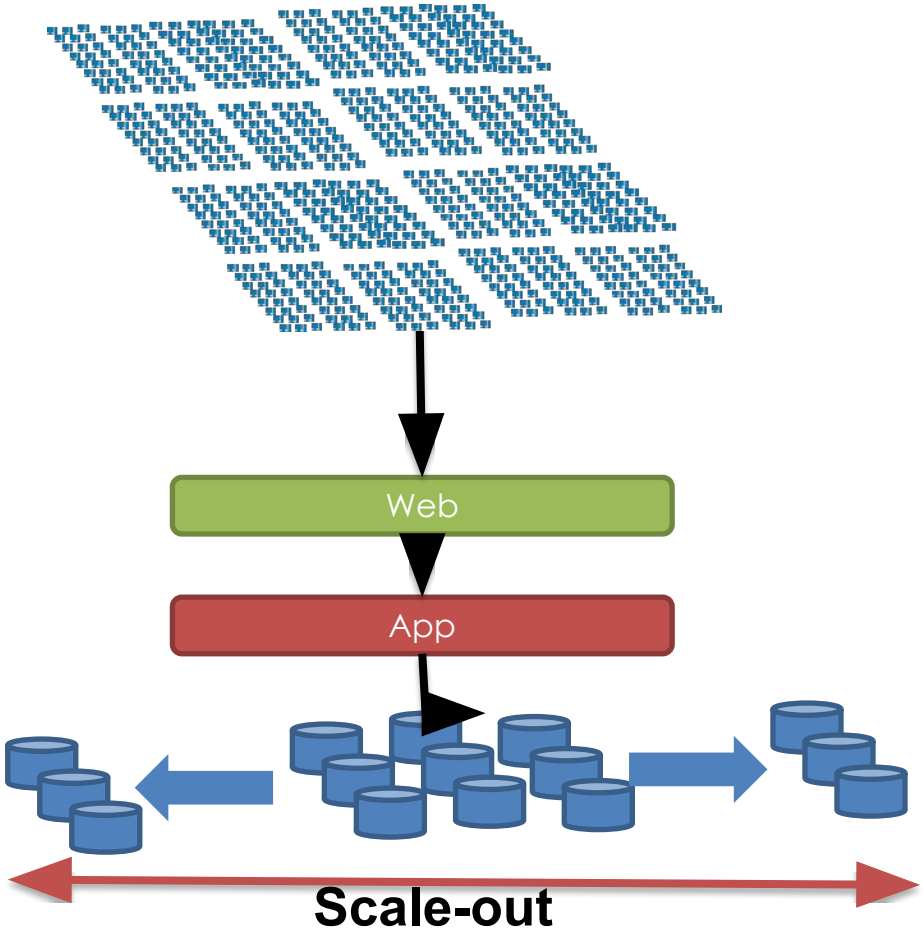
Cap Theorem



NOSQL

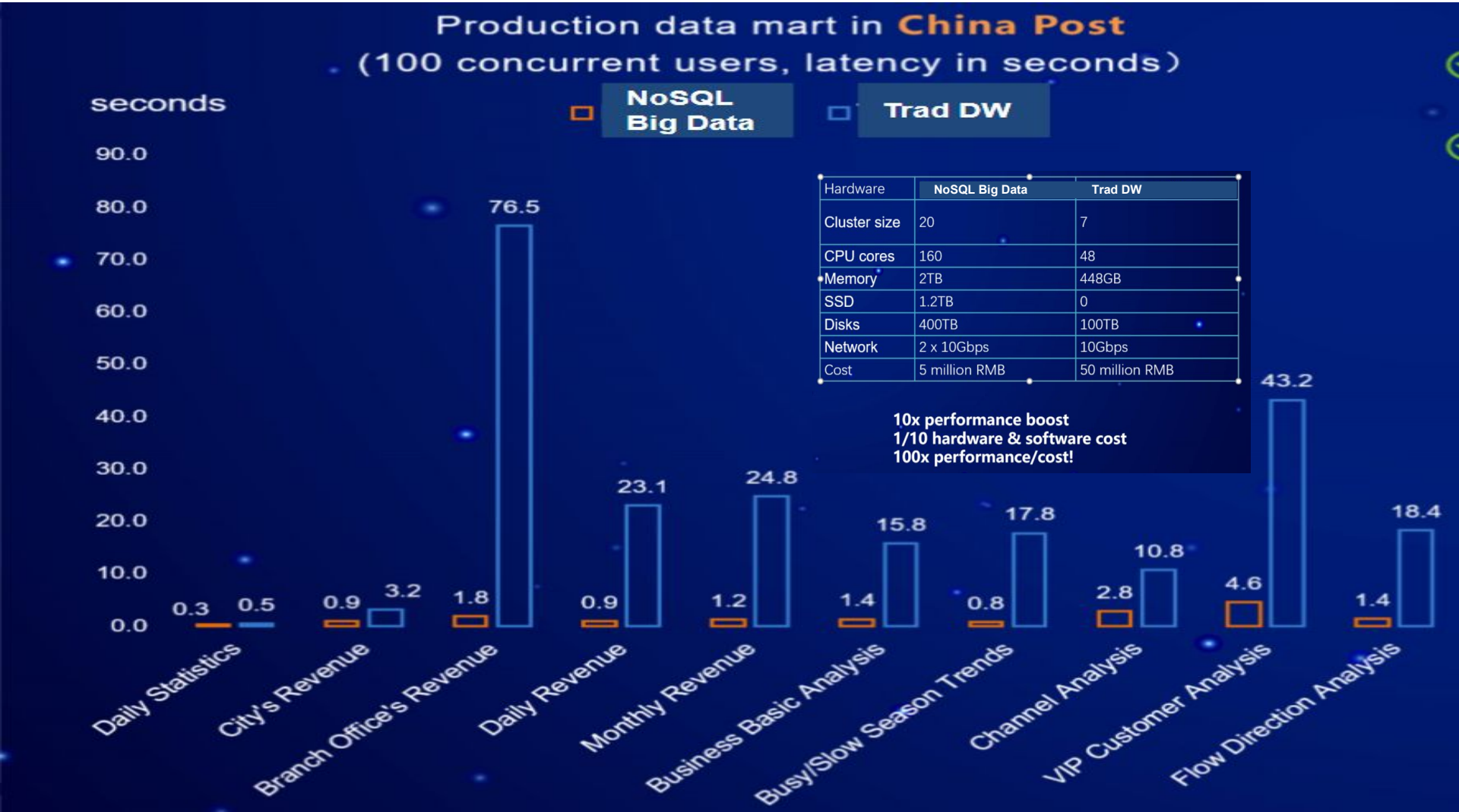
HORIZONTALLY SCALABLE DATABASES





HORIZONTAL SCALING IN ACTION

中国邮政 CHINA POST



Hardware	NoSQL Big Data	Trad DW
Cluster size	20	7
CPU cores	160	48
Memory	2TB	448GB
SSD	1.2TB	0
Disks	400TB	100TB
Network	2 x 10Gbps	10Gbps
Cost	5 million RMB	50 million RMB

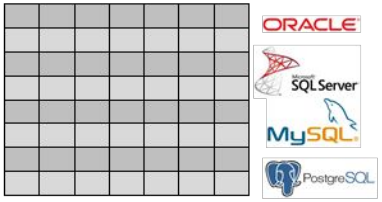
10x performance boost
1/10 hardware & software cost
100x performance/cost!

Reference

NOSQL

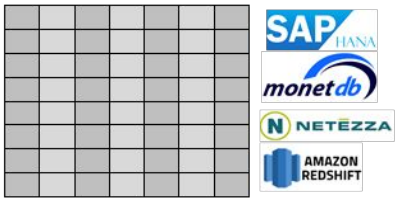
CHARACTERISTICS OF SQL VS NOSQL

RDBMS/OLTP/Real-time



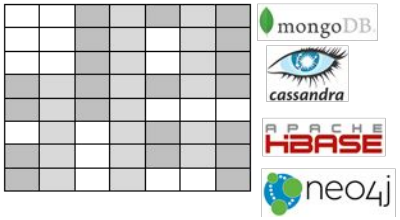
- Schema
- Data stored in rows
- Each select retrieves the whole row
- Scale-up (Add more cpus, raw) to process bigger queries

DW/OLAP/DSS/Batch



- Schema
- Data stored in columns
- Each select retrieves the whole column
- Its what you want for summaries (avg, count, mean) and other analytics
- Scale-up or scale-out (limited)

No/New SQL/BigData/Real/Batch



- Schema-free
- Data stored in various data structures:
  - Documents, JSON, XML
  - Rows, Columns, Graphs
- Typically access via APIs, some have SQL-like language queries
- “Unlimited” Scale-out (horizontally add more servers).. 100s – 1000s are typical
- Non ACID (except latest Neo4j)

Reference

NOSQL

DB-ENGINES.COM (1/4)

	MonetDB	Cassandra	Couchbase	MongoDB	Neo4j
Specific characteristics	Column-store In-memory optimized GeoSpatial support JSON document support	Continuous availability High scalability Performance Strong security Operational simplicity Lower overall cost of ownership.	Support the agile development and scalable deployment of enterprise web, mobile, and IoT applications. Distributed cache for low latency reads Key/value store for high performance reads and writes Document database for powerful querying and lightweight analytics	MongoDB is the next-generation database that helps businesses transform their industries by harnessing the power of data. MongoDB was also named a leader in the Forrester Wave™: Big Data NoSQL, Q3 2016.	Neo4j is a native graph database that is built to store, query and manage highly connected data more efficiently than other databases. Data relationships are first class citizens and can be traversed in constant time without index lookups that allows even complex queries to deliver results in milliseconds than in minutes. Architected for the property graph model, it provides optimized mechanisms to work with todays complex domains and use-cases.

Reference

NOSQL

DB-ENGINES.COM (2/4)

	MonetDB	Cassandra	Couchbase	MongoDB	Neo4j
Competitive advantages	Scalable architecture - vertical data organization and in-memory optimization guarantee fast query responses for very large datasets, without substantial hardware investments. In-database analytics with embedded R - combine the flexibly of statistical software with the performance of database management system for no-hassle analytics. Access all statistical analysis function with no copying data between system for maximum performance and no data transformations. Automated & adaptive indices - no extra DBA work for index management. Standards compliant - easy to integration in most ETL, BI and analytics stacks.	No single point of failure ensures 100% availability. Operational simplicity for lowest total cost of ownership. Best-in-class scalability of NoSQL platforms.	Powerful Querying: An expressive query language that extends SQL and supports joins (nest/unnest, left outer, inner) Consistent High Performance: A managed object cache and streaming replication, persistence, and indexing Global Deployment: An optimized replication protocol to support multiple data centers in active/active configurations Mobile Platform: An embedded database (Couchbase Lite) for iOS, Android, and others with automatic syncronization Spark Integration: A connector that supports Spark, Spark SQL, and Spark Streaming (and Kafka)	Best of traditional databases as well as the flexibility, scale, and performance required by today's applications. Relational databases Strong consistency Expressive query language Secondary indexes MongoDB provides the data model flexibility, elastic scalability, and high performance and availability of NoSQL databases.	Neo4j is the only transactional database that combines everything you need for performance and trustability in applications that bring data relationships to the fore: Native graph storage Native graph processing Graph scalability High availability Built-in ETL Integration support plus Cypher, a powerful and expressive language for queries using vastly less code than SQL.

Reference

NOSQL

DB-ENGINES.COM (3/4)

	MonetDB	Cassandra	Couchbase	MongoDB	Neo4j
Typical application scenarios	Analytical database Data warehouse OLAP Data mining Scientific database	Internet of Things (IOT) Fraud detection Recommendation engines Product catalogs Messaging applications.	Real-time Big Data Profile Management Mobile Applications Content Management Customer 3600 View Fraud Detection Internet of Everything Catalogs Digital Communication Personalization	Internet of Things (Bosch, Silver Spring Networks) Mobile (The Weather Channel, ADP, O2) Single View (MetLife) Real Time Analytics (Buzzfeed, City of Chicago, Crittercism) Personalization (Expedia, eHarmony, Gilt) Catalogs (Under Armour, Otto) Content Management (eBay, Forbes)	Real-Time Recommendations Master Data Management Identity and Access Management Network and IT Operations Fraud Detection Graph-Based Search
Key customers	Numascale Spinque SecurActive CHS NOZHUP	Barracuda Networks, NY Times, Outbrain, BazaarVoice, Best Buy, Comcast, eBay, Hulu, Sky, Pearson Education	AOL, AT&T, Bally's, BSkyB, Cisco, Comcast, Concur, DIRECTV, Disney, eBay, KDDI, Nordstrom, Neiman Marcus, Orbitz, PayPal, Ryanair, Rakuten / Viber, Tencent, Verizon, Wells Fargo, Willis Group	ADP, Adobe, AstraZeneca, BBVA, Bosch, Cisco, CERN, Department of Veteran Affairs, eBay, eHarmony, Electronic Arts, Expedia, Facebook's Parse, Forbes, Foursquare, Genentech, MetLife, Pearson, Sage, Salesforce, The Weather Channel, Ticketmaster, Under Armour, Verizon Wireless	eBay, Walmart, Cisco, UBS, HP, CenturyLink, Telenor, TomTom, Telenor, The National Geographic Society

Reference

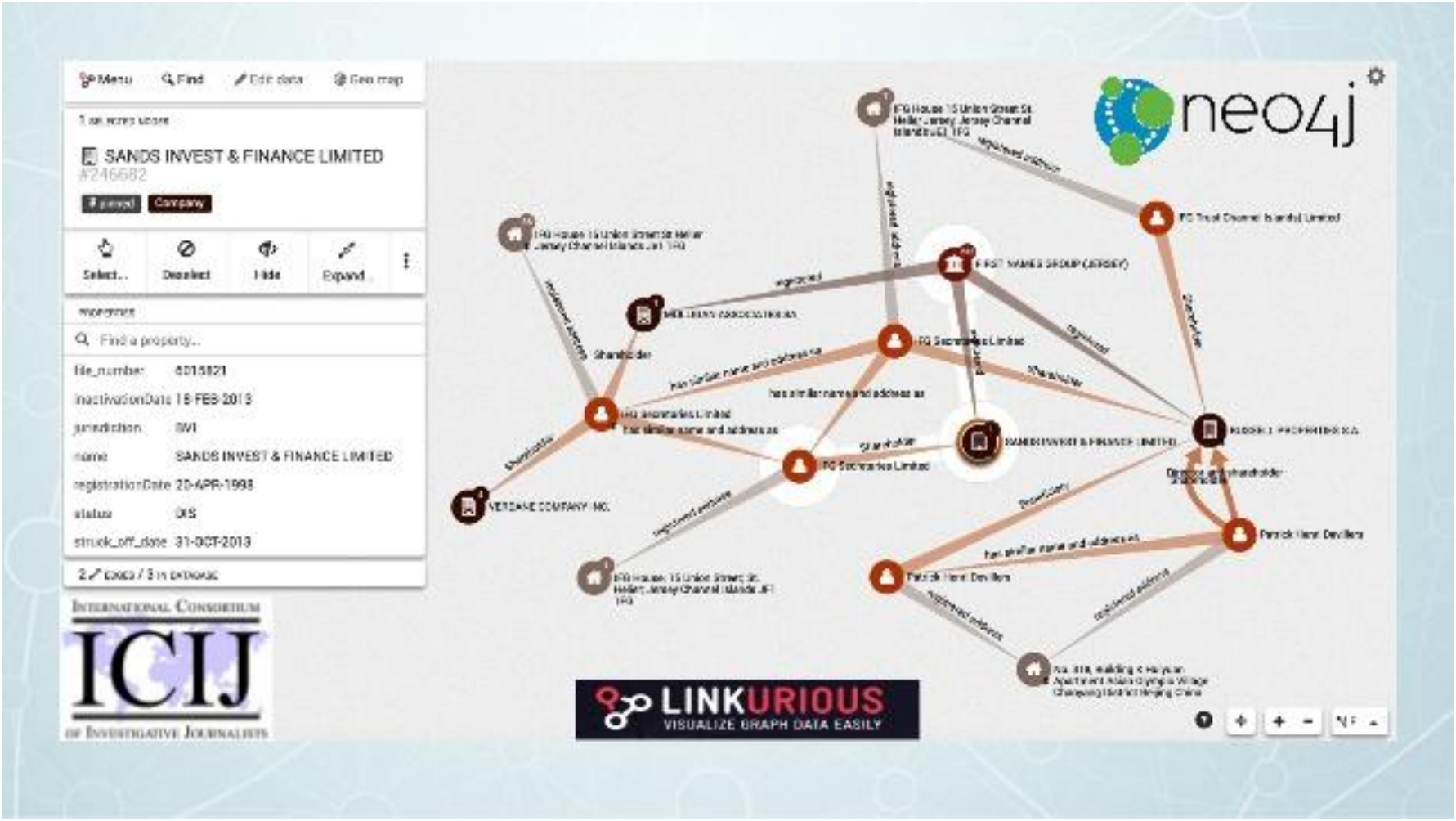
NOSQL

DB-ENGINES.COM (4/4)

	MonetDB	Cassandra	Couchbase	MongoDB	Neo4j
Market Metrics		Cassandra is used by a quarter of the Fortune 100		20 million downloads (growing at thousands downloads per day). 2,000+ customers including over one third of the Fortune 100. Named a leader in the Forrester Wave™: Big Data NoSQL, Q3 2016. Highest placed non-relational database in DB Engines rankings	Neo4j boasts the world's largest graph database ecosystem with more than a million downloads, and 200+ enterprise customers, including 50 Global 2000 companies.
Licensing and pricing models	Free and open-source software, Mozilla Public License 2.0 Commercial support available	Apache license Pricing for commercial distributions provided by DataStax and available upon request.	All software is licensed under the Apache License, Version 2.0	Free Software Foundation's GNU AGPL v3.0. Commercial licenses are also available from MongoDB, Inc.	GPL v3 license that can be used all the places where you might use MySQL. Neo4j Commercial license is offered by Neo Technology, Inc

GRAPH DATABASE

FIND AND EXPOSE RELATIONSHIPS

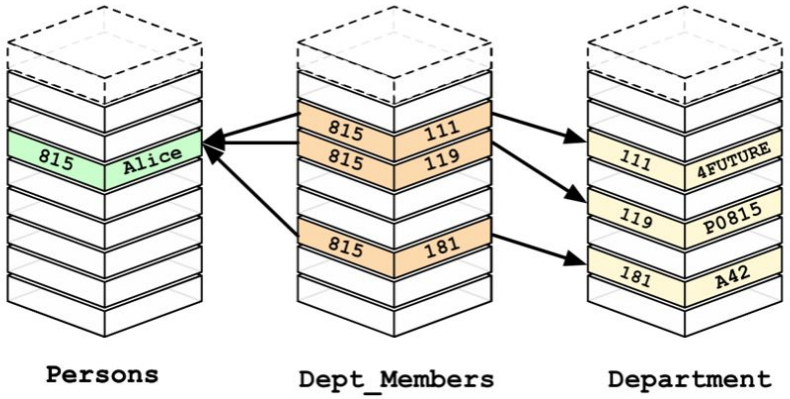


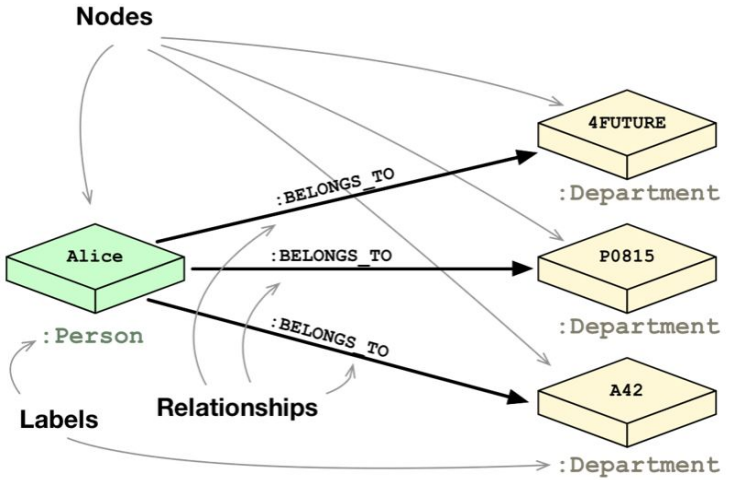
Reference

GRAPH DATABASE

WHY GRAPH DATABASE

- Why graph?
  - From whiteboard to database
  - Little or no impedance
  - More natural query language
  - Fast
    - A few seconds vs hours compared to SQL (too many complex joins and multiple tables)

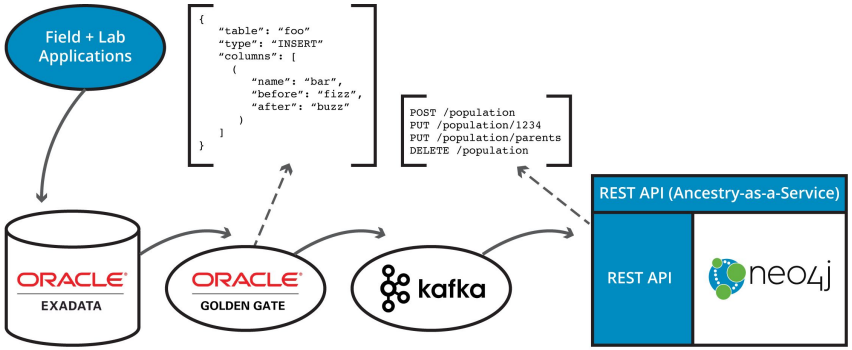




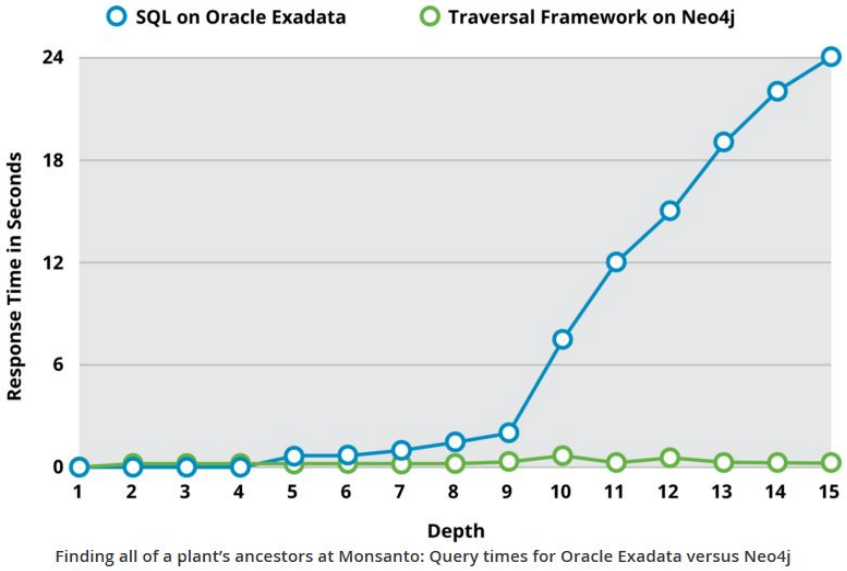
Reference

GRAPH DATABASE

PERFORMANCE



- Not replace but complement
- Use the right tool for the right job

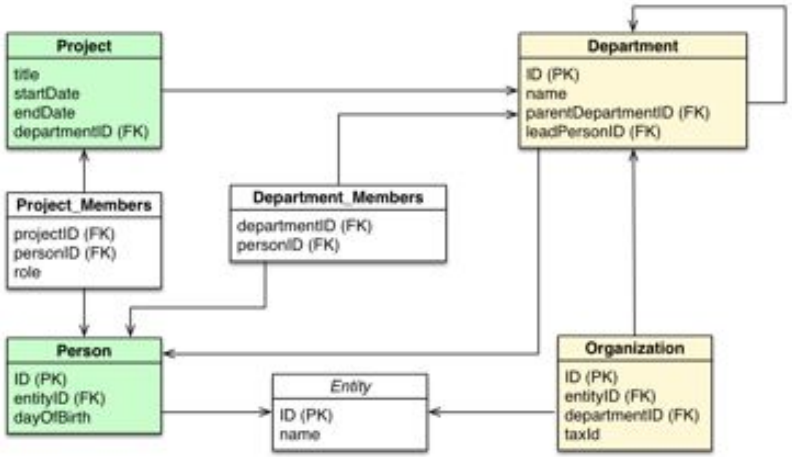


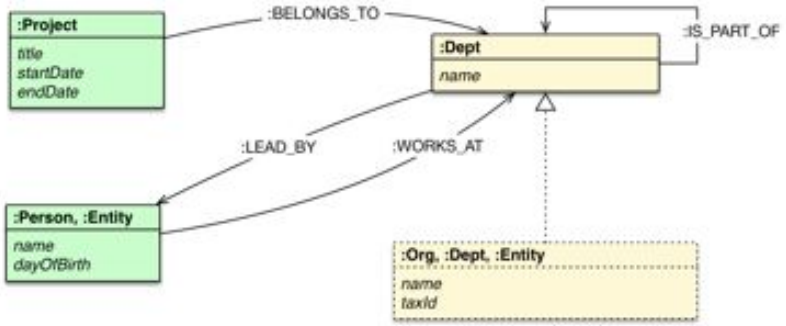
Finding all of a plant's ancestors at Monsanto: Query times for Oracle Exadata versus Neo4j

Reference

GRAPH DATABASE

SIMPLER QUERY STATEMENTS





Query: Lists the *employees in the "IT Department"*

SQL Statement

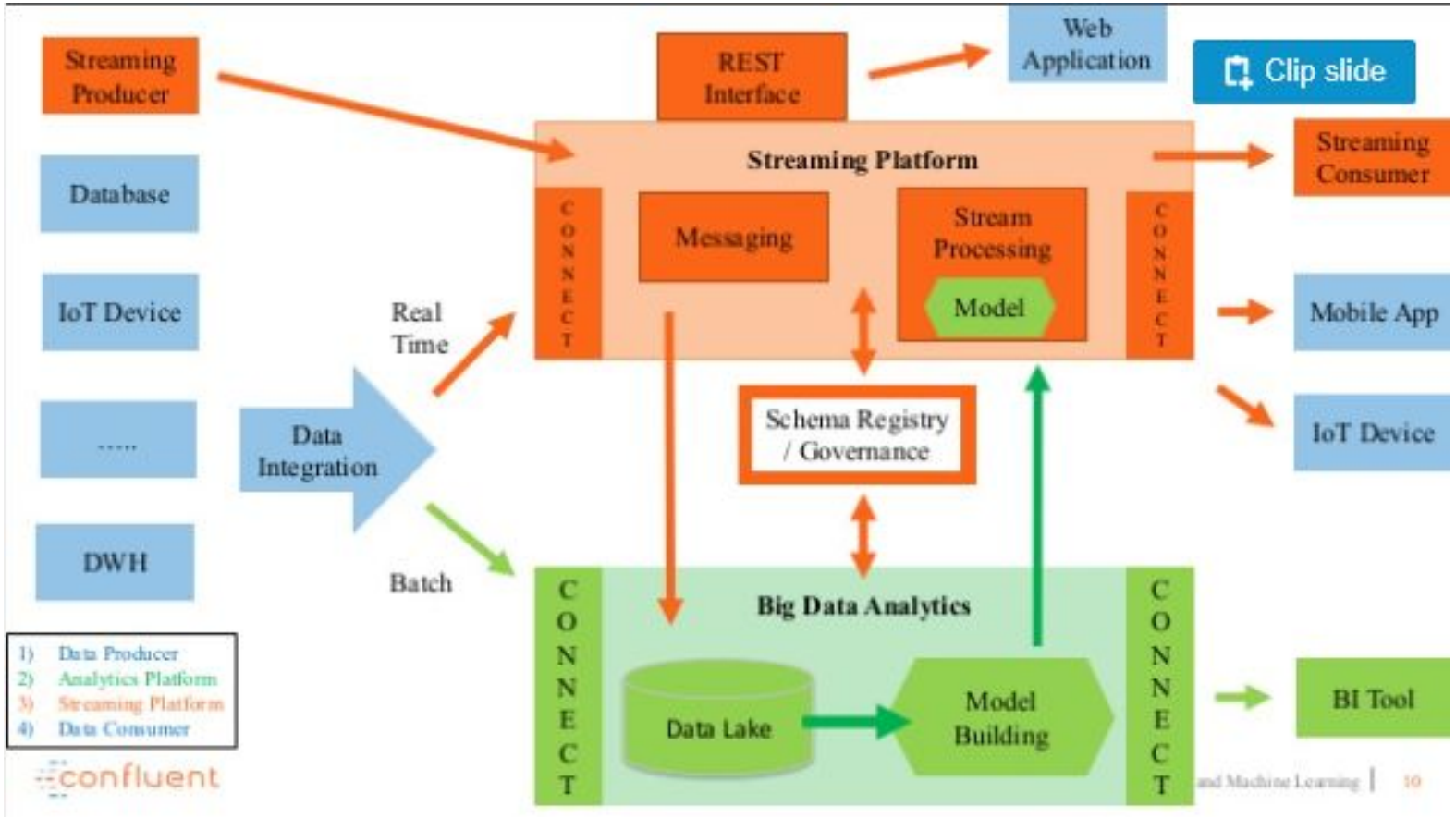
```
SELECT name FROM Person
LEFT JOIN Person_Department
  ON Person.Id = Person_Department.PersonId
LEFT JOIN Department
  ON Department.Id = Person_Department.DepartmentId
WHERE Department.name = "IT Department"
```

Cypher Statement

```
MATCH (p:Person)<-[:EMPLOYEE]-(d:Department)
WHERE d.name = "IT Department"
RETURN p.name
```

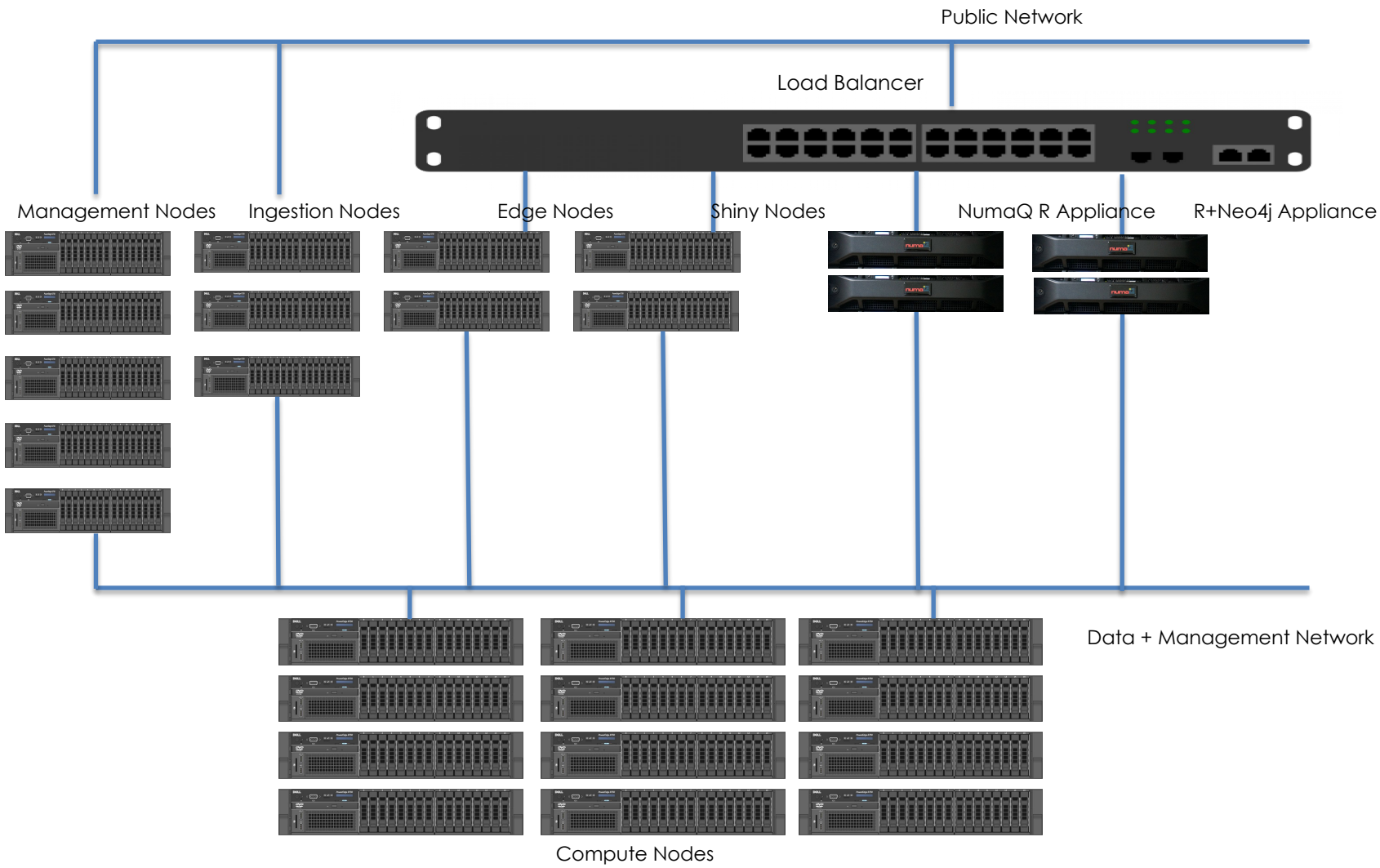
IOT ARCHITECTURE

APACHE KAFKA STREAMS

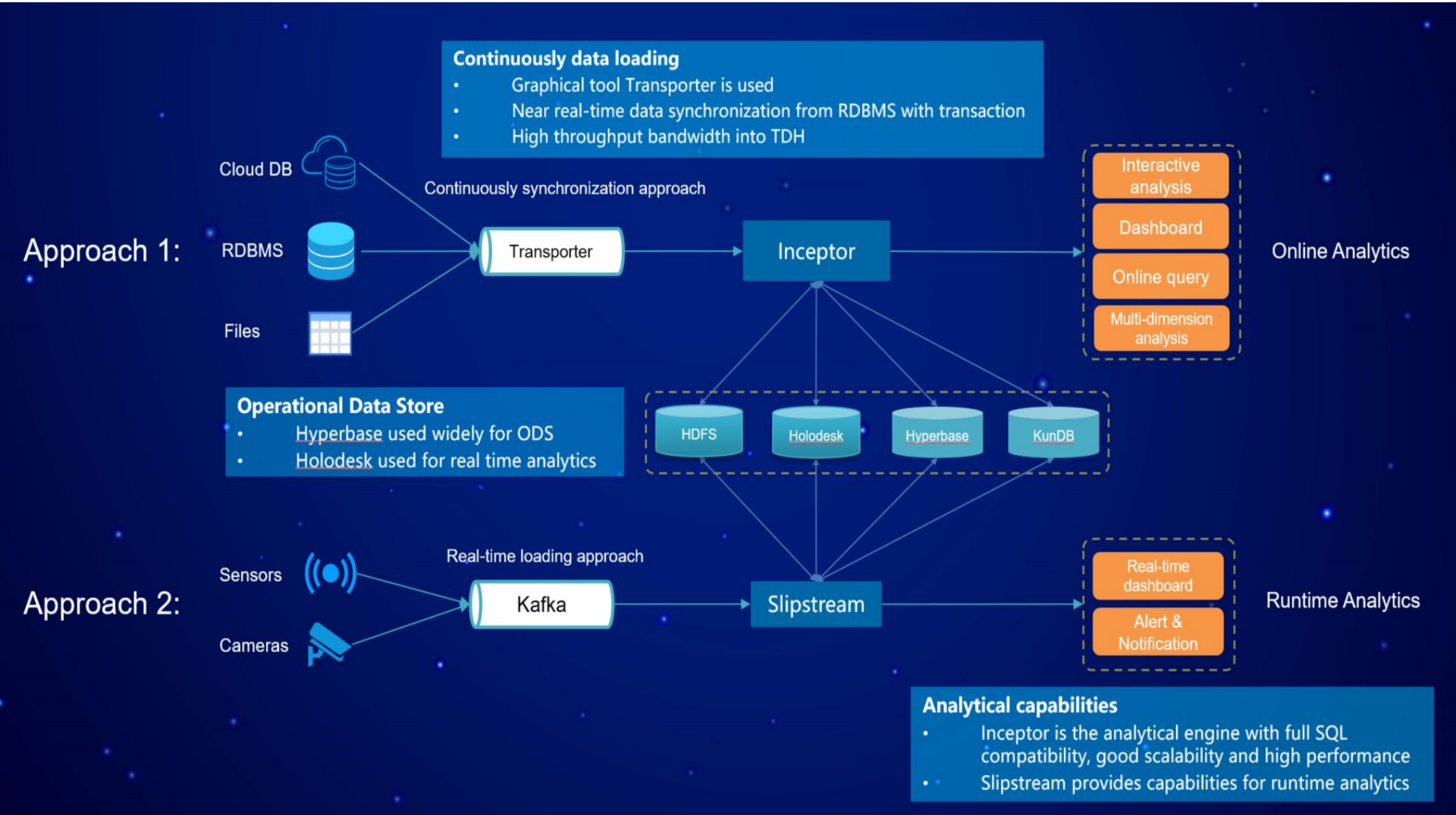


Source: https://www.slideshare.net/KaiWaehner/apache-kafka-streams-machine-learning-deep-learning

ON-PREMISE IOT ARCHITECTURE

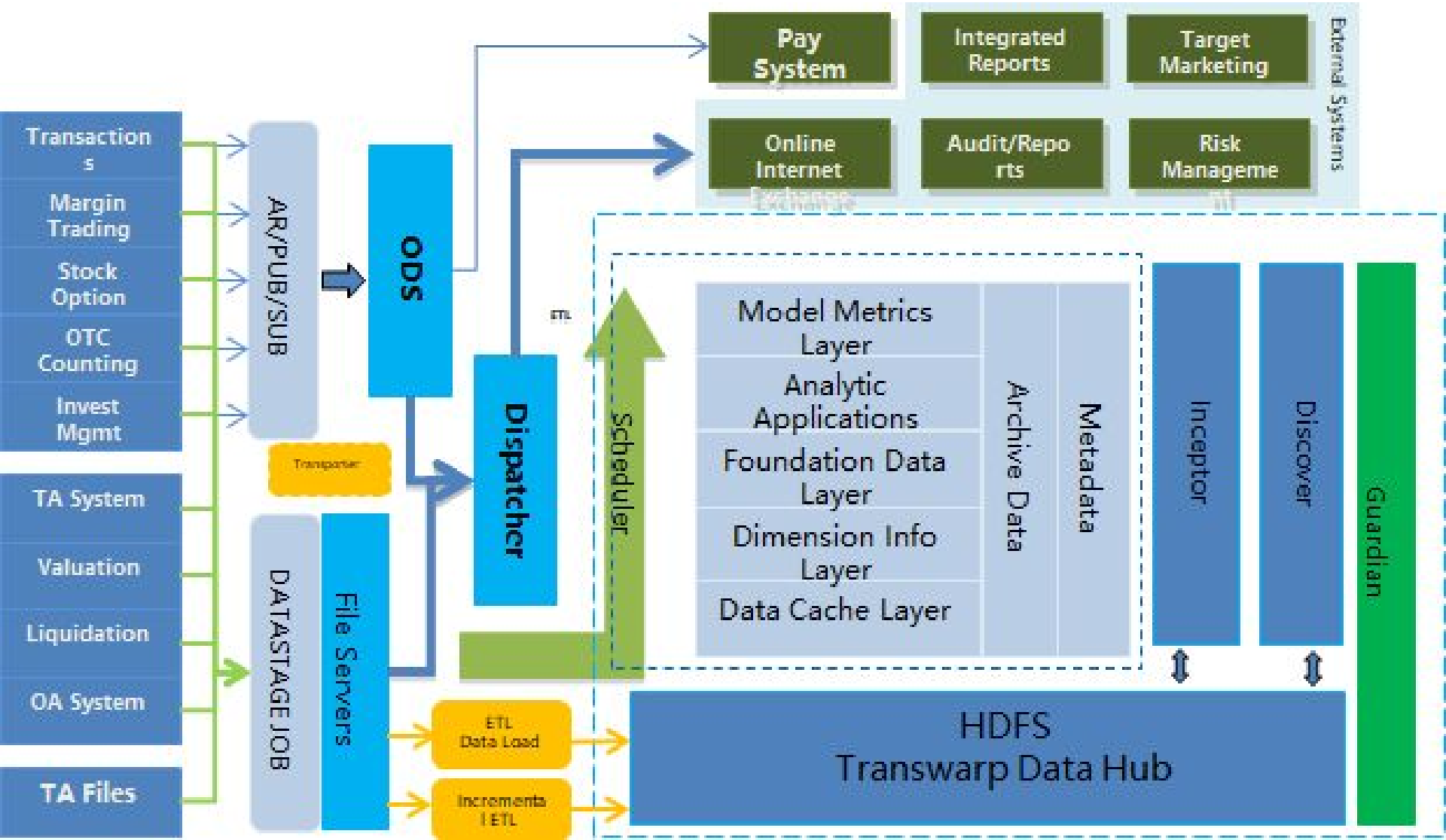


LARGE SCALE NEAR REALTIME DATA PROCESSING

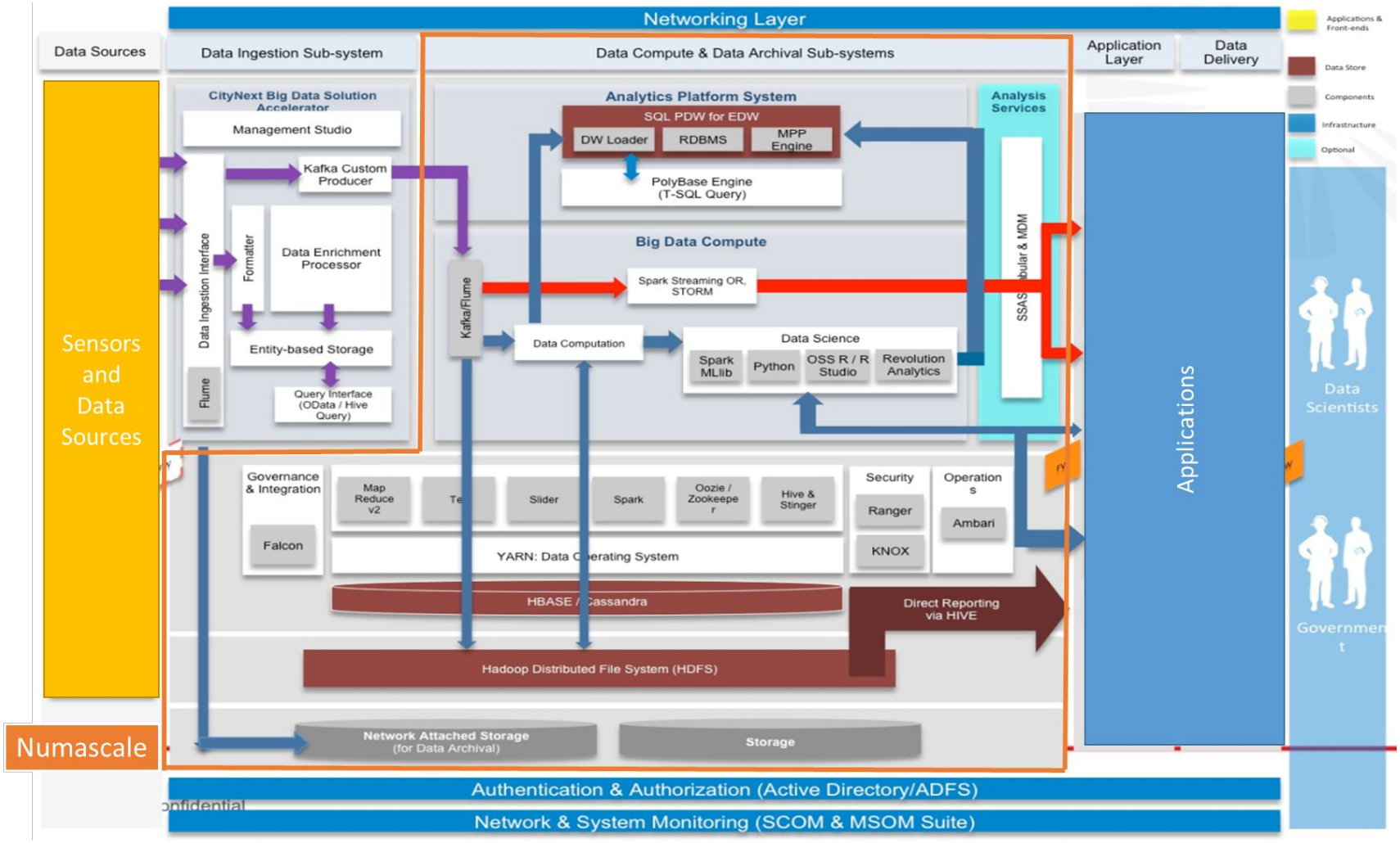


BIG DATA - DATA WAREHOUSE

First Capital Bank

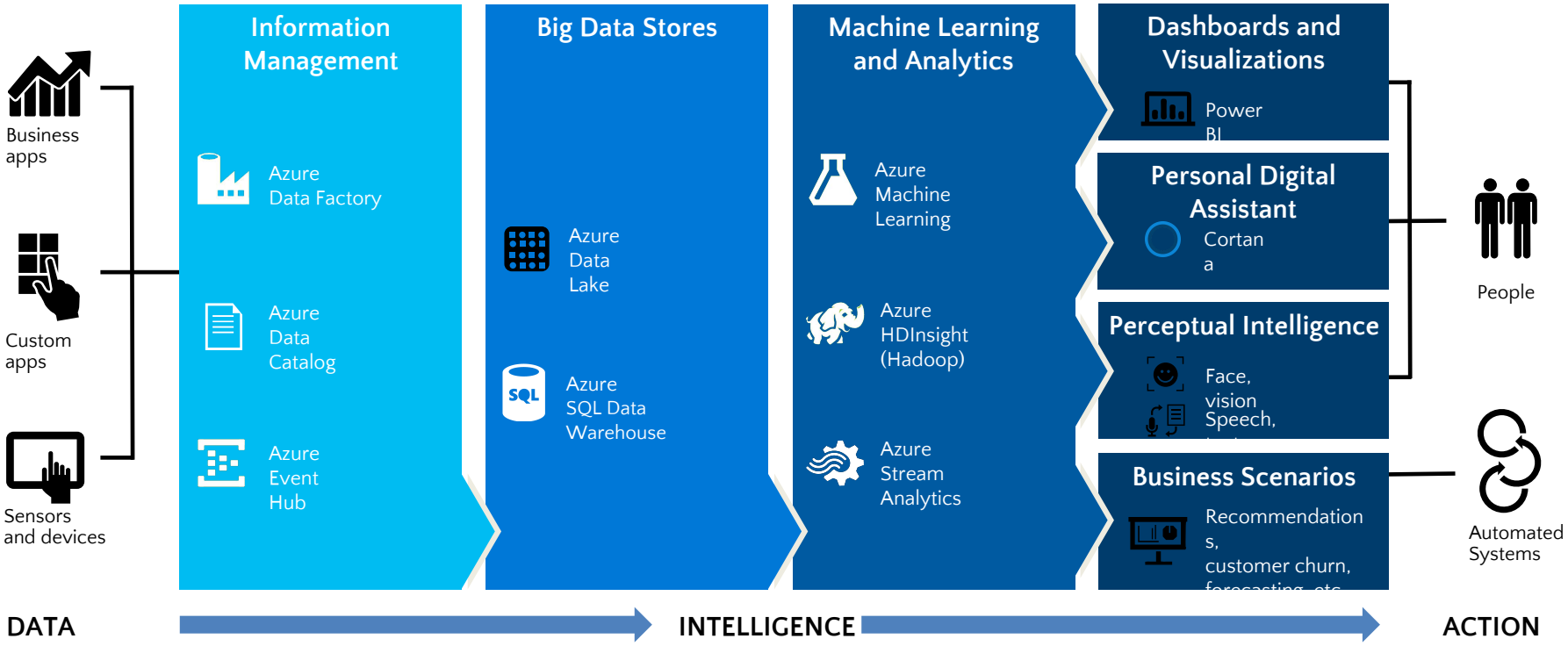


IOT ARCHITECTURE IN CLOUD



CORTANA INTELLIGENCE SUITE

What would it take your organization to build the same infrastructure?



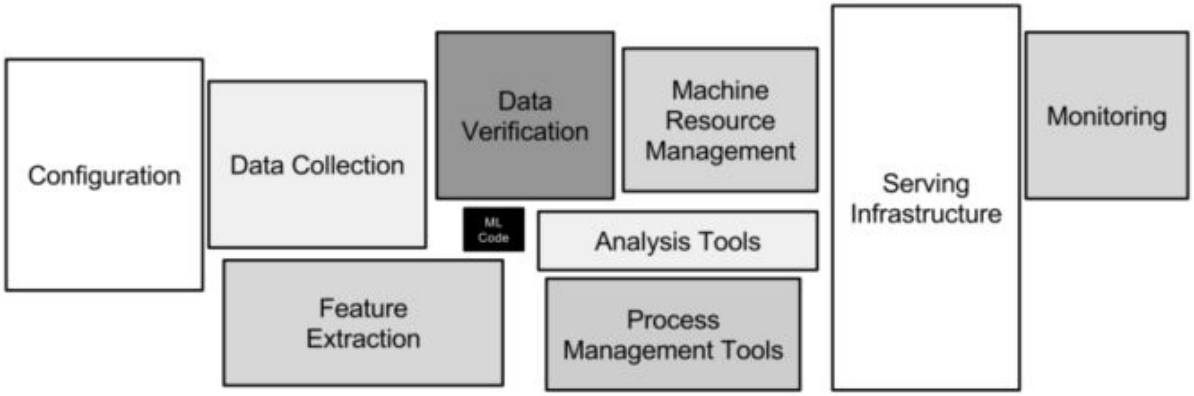
Source: Microsoft Cortana Analytics Training

TECHNICAL DEBT

THERE IS A LOT MORE OF OTHER SUPPORTING CODE AND INFRASTRUCTURE IN A ML PROJECT!

Hidden Technical Debt in Machine Learning Systems

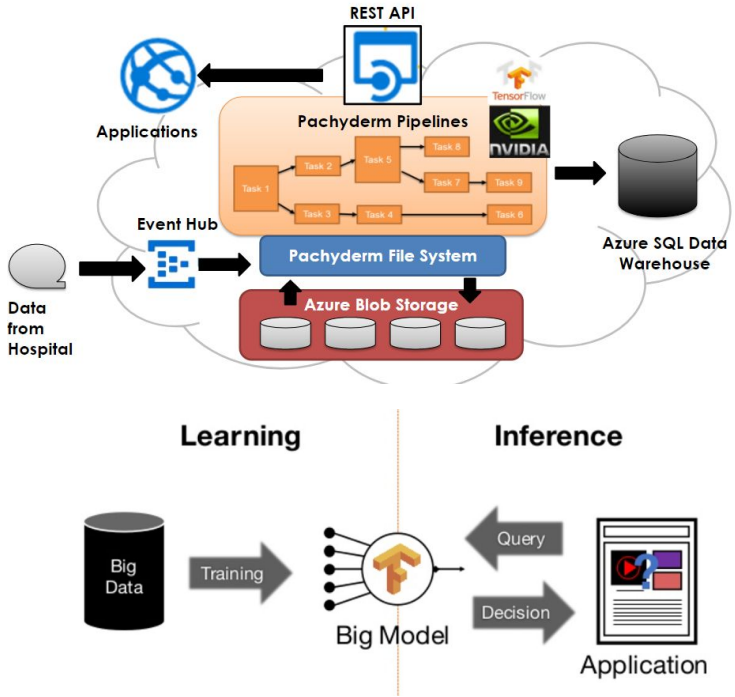
D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.



"Only a fraction of real-world ML systems is composed of ML code"

SUMMARY

- Open source frameworks!
- Commodity hardware (x86), SATA/SAS HDD/SSDs. No SAN!
- Choose the right tool for the workload
  - It means you need to know your workload
- No one size fits all
- You will be gluing many/different frameworks together for a solution
- Expect to refresh you architecture every 2 years!

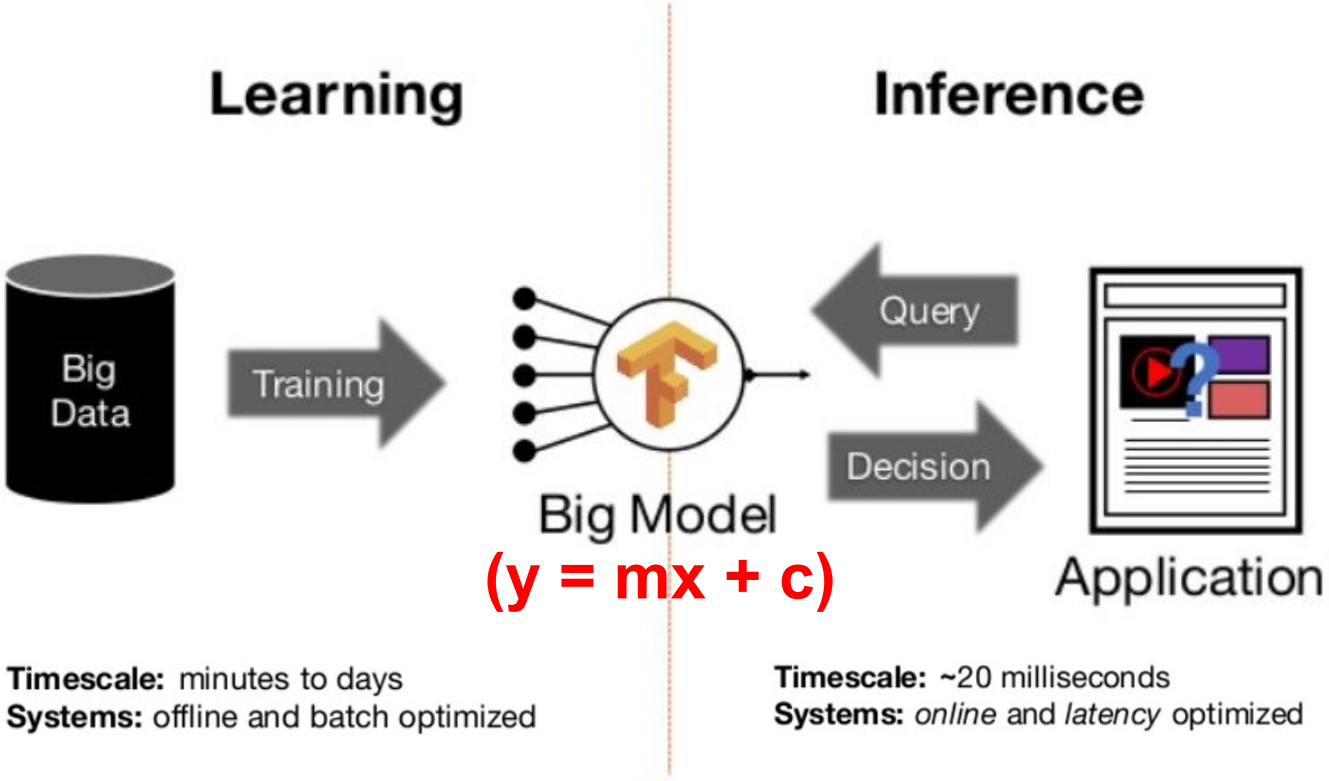


Data Analytics for Managers

Reproducible Analytics Production Pipeline

DEPLOYMENT

LEARNING VS PREDICTING



Source:
https://www.slideshare.net/SparkSummit/clipper-a-lowlatency-online-prediction-serving-system-spark-summit-east-talk-by-dan-crankshaw

A/B TEST

1. Define your goal and form your hypotheses.
2. Identify a control and a treatment.
3. Identify key metrics to measure.
4. Identify what data needs to be collected.
5. Make sure that appropriate logging is in place to collect all necessary data.
6. Determine how small of a difference you would like to detect.
7. Determine what fraction of visitors you want to be in the treatment
8. Run a power analysis to decide how much data you need to collect and how long you need to run the test.
9. Run the test for AT LEAST this long.
10. First time trying something new: run an A/A test (dummy test) simultaneously to check for systematic biases.

PRODUCTION DEPLOYMENT

Previously

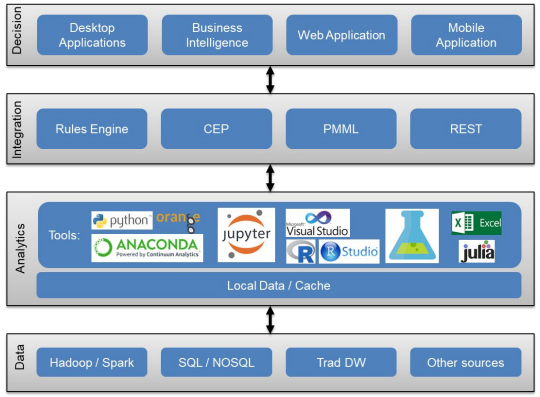
1. Prototype in R and/or Python
2. Re-write in C/C++ or Java

3- 12 months "delay"

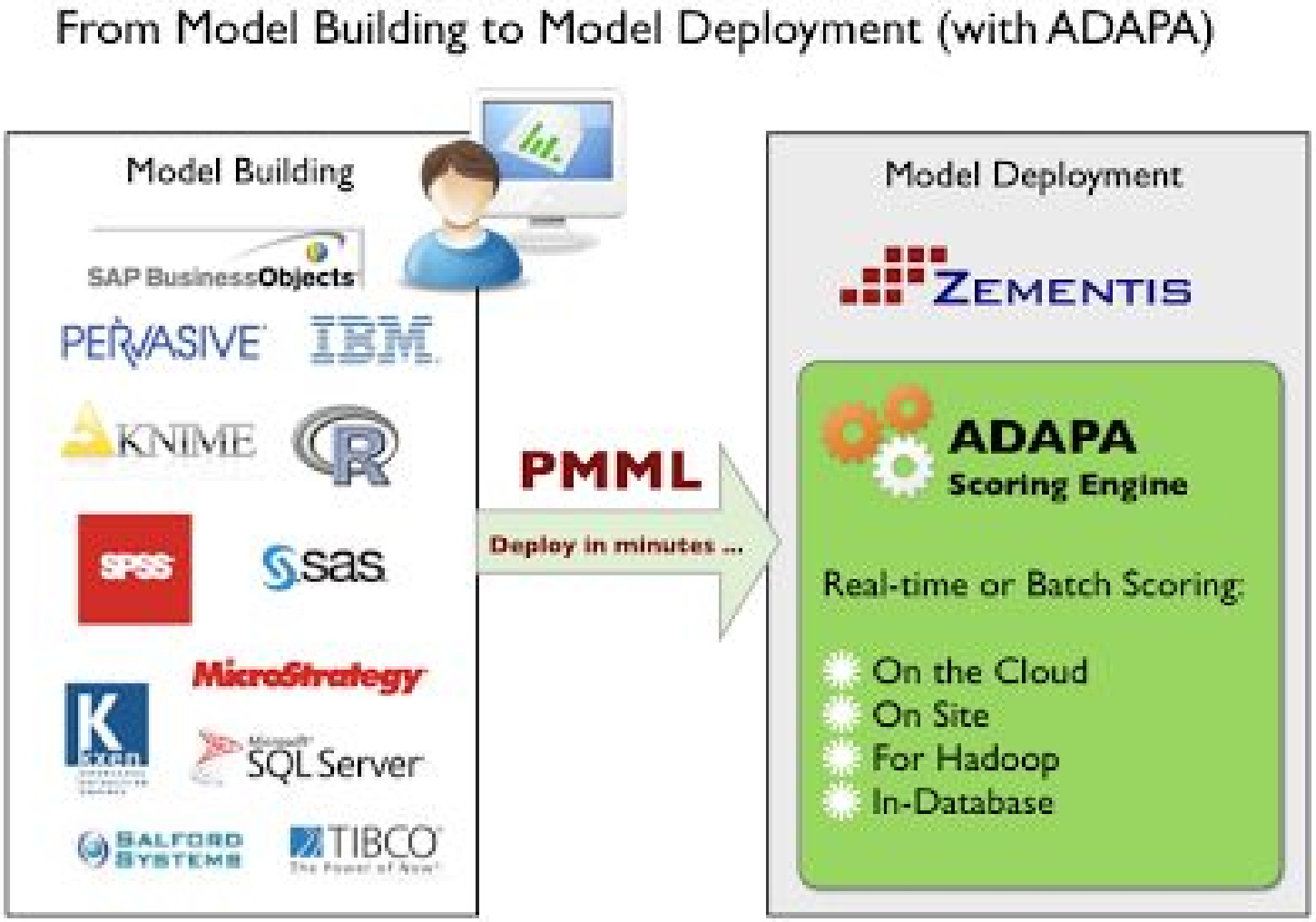
Now

Nearly immediately!

1. Prototype in R and/or Python
2. Deploy in R and Python

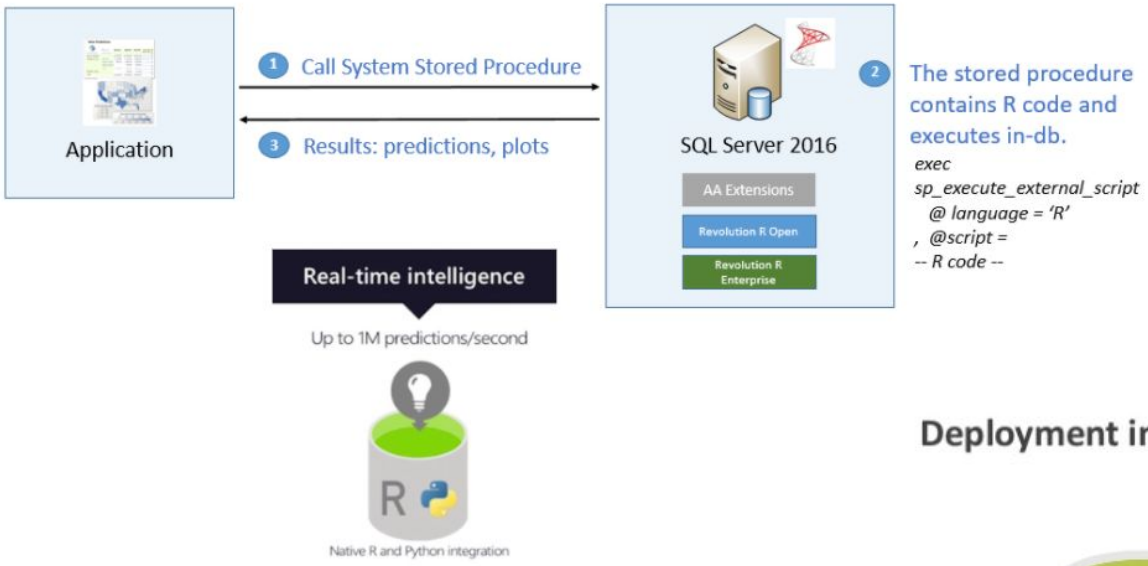


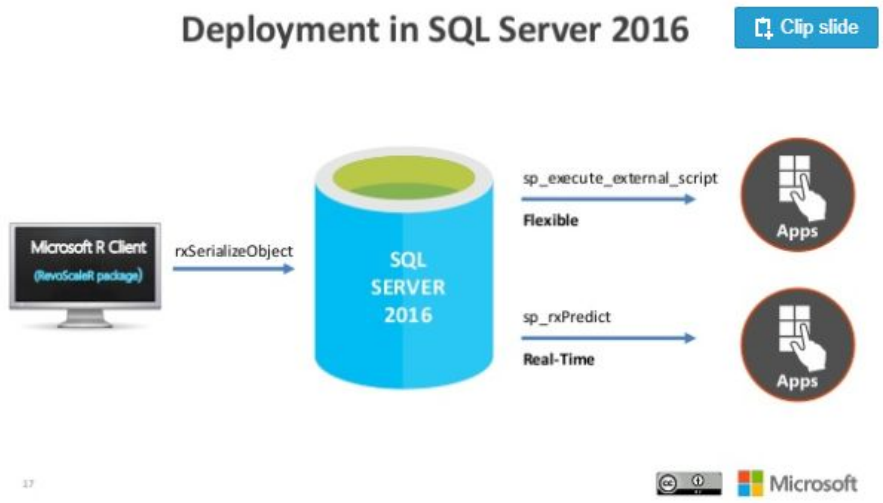
PMML



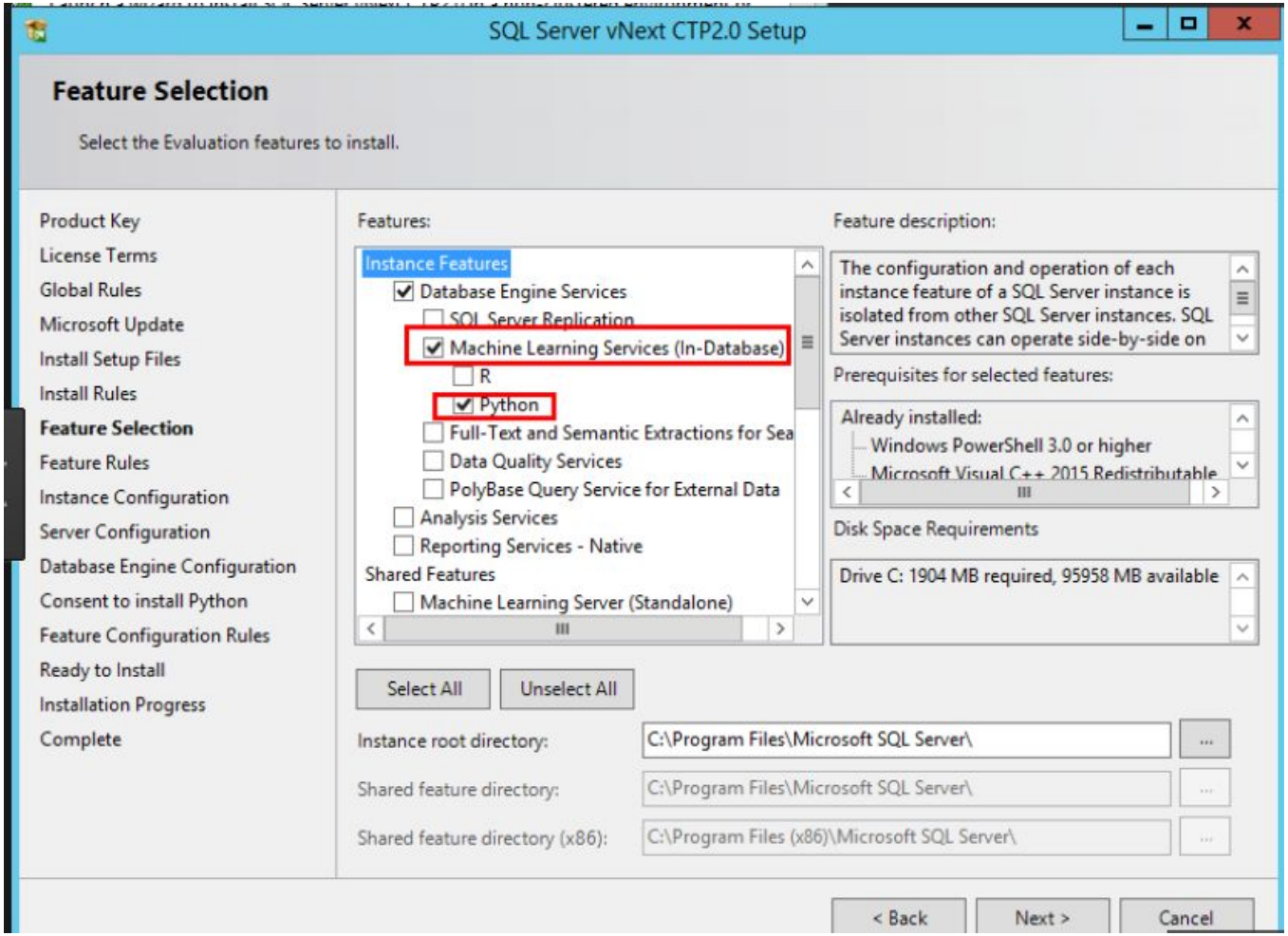
SQL SERVER 2017

I can call a T-SQL System Stored Procedure from my application and have it trigger R script execution in-database. Results are then returned to my application (predictions, plots, etc.).





SQL SERVER 2017



REAL TIME FRAUD DETECTION

September 20, 2016

Using R to detect fraud at 1 million transactions per second

In Joseph Sirosh's keynote presentation at the Data Science Summit on Monday, Wee Hyong Tok demonstrated using R in SQL Server 2016 to detect fraud in real-time credit card transactions at a rate of 1 million transactions per second. The demo (which starts at the 17:00 minute mark) used a gradient-boosted tree model to predict the probability of a credit card transaction being fraudulent, based on attributes like the charge amount and the country of origin.

Then, a stored procedure in SQL Server 2016 was used to score transactions streaming into the database at a rate of 3.6 billion per hour. If you'd like to try this yourself, a step-by-step tutorial with code to implement the model and scoring is available here.

20

Flexible vs Real-Time
1M predictions/sec
Same benchmark
One-sixth the resources

SQL Server 2017
8 sockets, 192 cores
6 TB RAM
Flexible operationalization

blog.revolutionanalytics.com/2016/09/fraud-detection.html

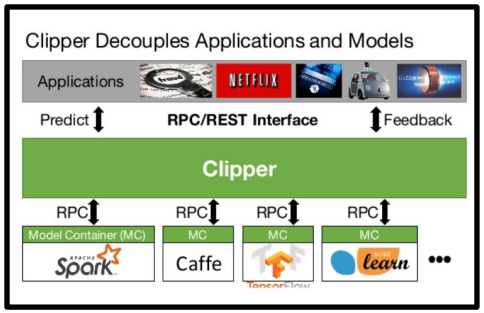
Microsoft

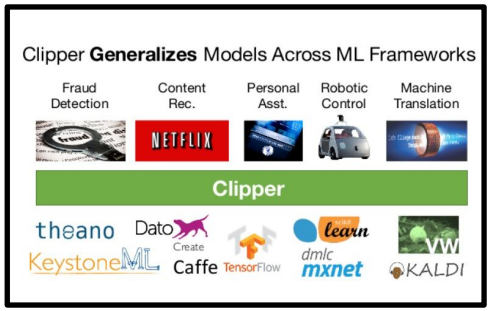
CLIPPER - A GENERIC DEPLOYMENT FRAMEWORK

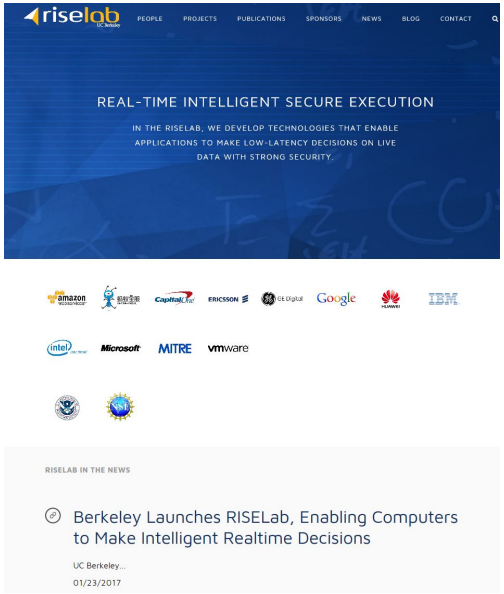
https://clipper.ai

What Does Clipper Do?

- Clipper **simplifies integration of machine learning techniques** into user facing applications by providing a simple standard REST interface for prediction and feedback across a wide range of commonly used machine learning frameworks. *Clipper makes product teams happy.*
- Clipper **simplifies model deployment** and **helps reduce common bugs** by using the same tools and libraries used in model development to render live predictions. *Clipper makes data scientists happy.*
- Clipper **improves throughput** and ensures **reliable millisecond latencies** by introducing adaptive batching, caching, and straggler mitigation techniques. *Clipper makes the infra-team less unhappy.*
- Clipper **improves prediction accuracy** by introducing state-of-the-art bandit and ensemble methods to intelligently select and combine predictions and achieve real-time personalization across machine learning frameworks. *Clipper makes users happy.*

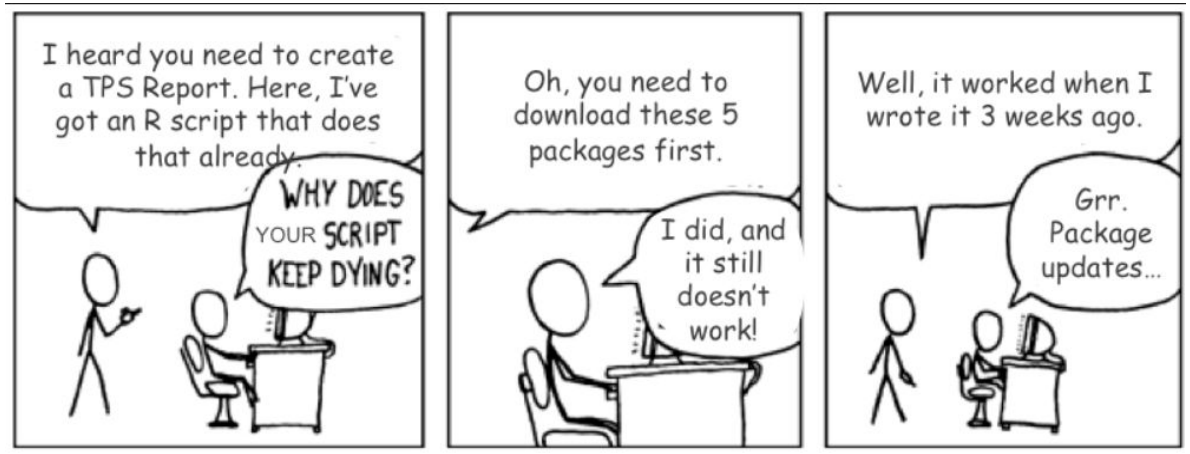


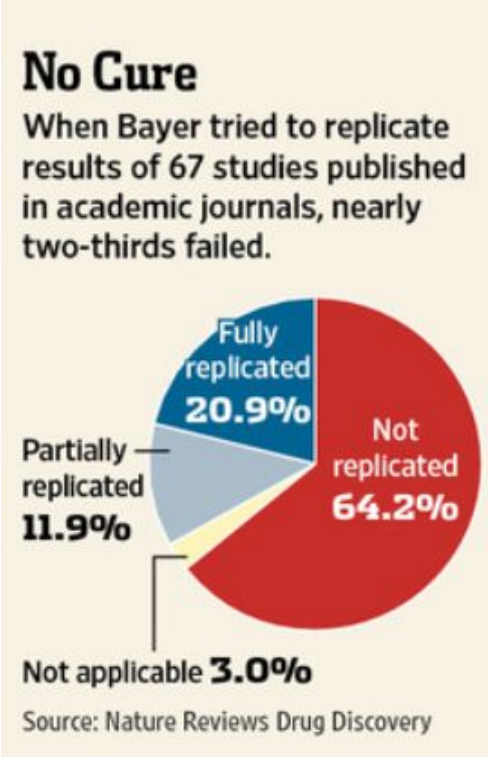




100% Open source – would you use it?

REPRODUCIBILITY MATTERS



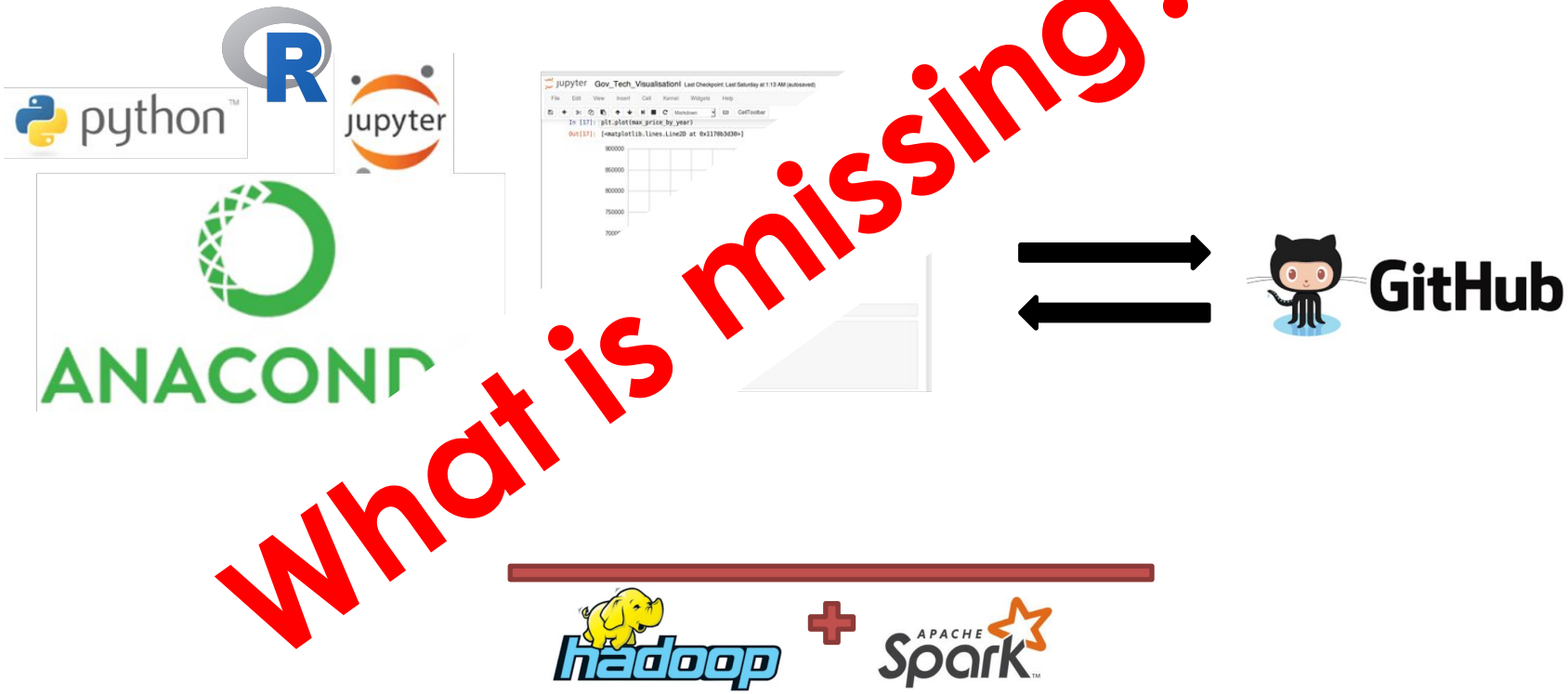


REPRODUCIBLE DATA SCIENCE

- Why
  - Analysis becomes repeatable and auditable -> Reproducible!
  - So others can repeat the analysis, verify the findings and build on the investigation
- How
  - Steps in data analyses, data and software code are documented and **codified**

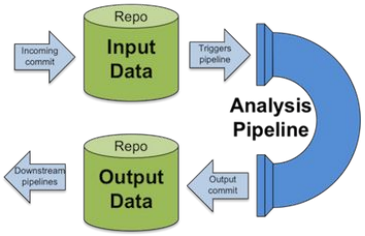
REPRODUCIBLE DATA SCIENCE

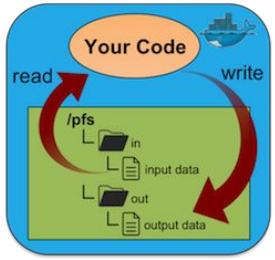
Current state of the art/best practices

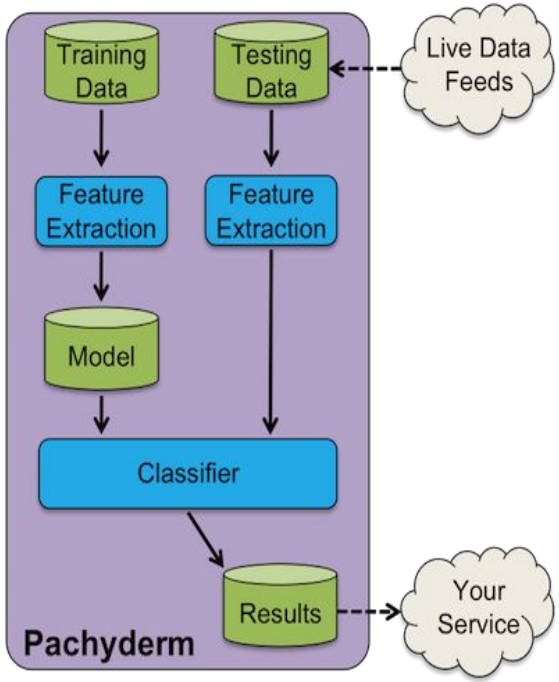


PACHYDERM

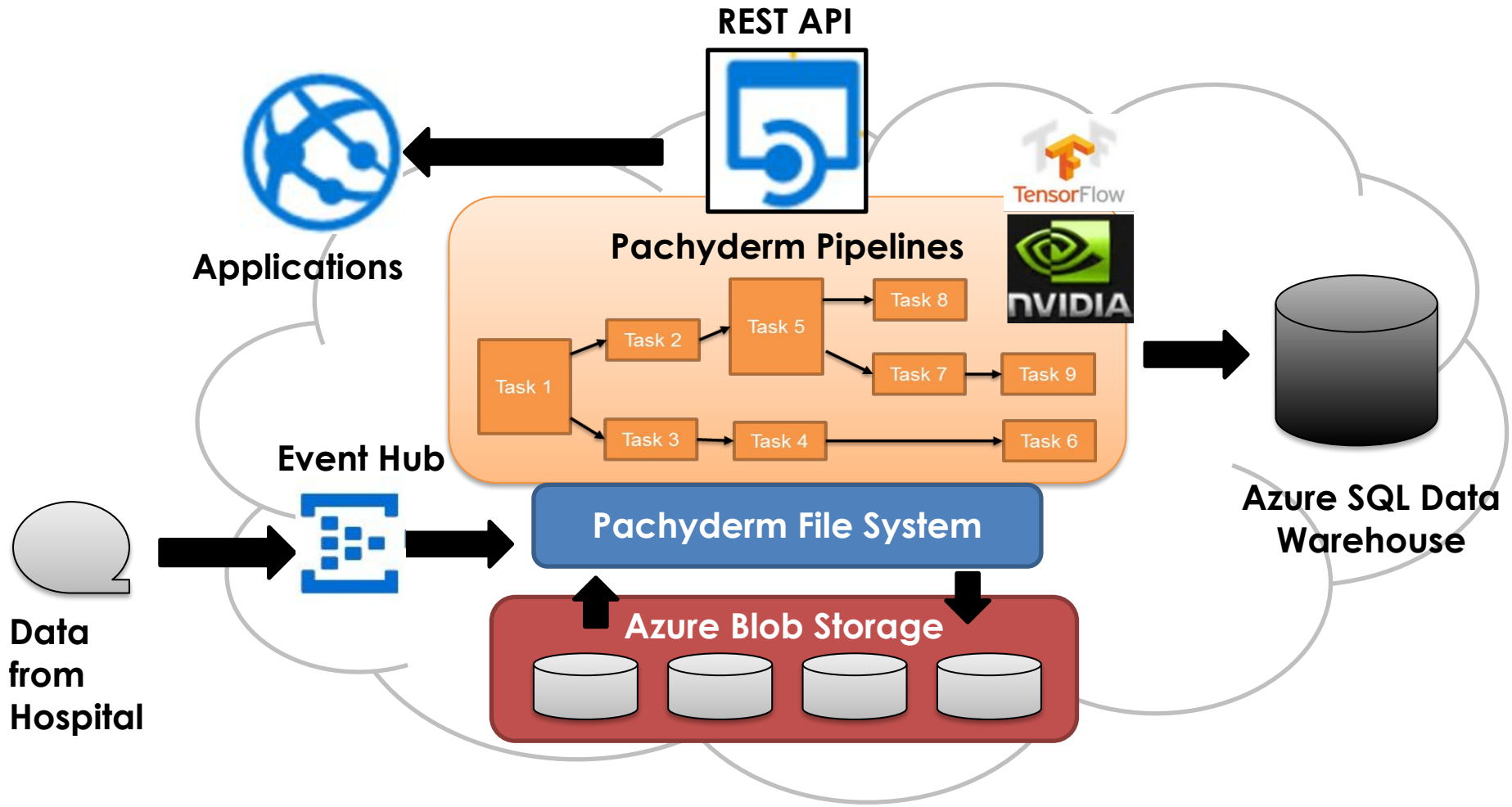
- “Git for Data Science”
- Containerized Analysis, Unified Data+Code
- Runs on Kubernetes/Docker, Language/tooling agnostic
- Focuses on Data provenance, reproducibility and collaboration
- “Data Versioning” + “Data Processing/Pipelining”







POC HOSPITAL PROJECT



PRODUCTION DEPLOYMENT
GUIDELINES

- Specify the required performance, eg response time, accuracy.
- Reproducibility of the model (code + data), regression tests + git of (code+data), eg Pachyderm pipeline
- Think about automated tests and update of the model: model drift, underlying assumptions have changed etc
- Conduct A/B Tests to validate the performance of the model in the field!!

SUMMARY

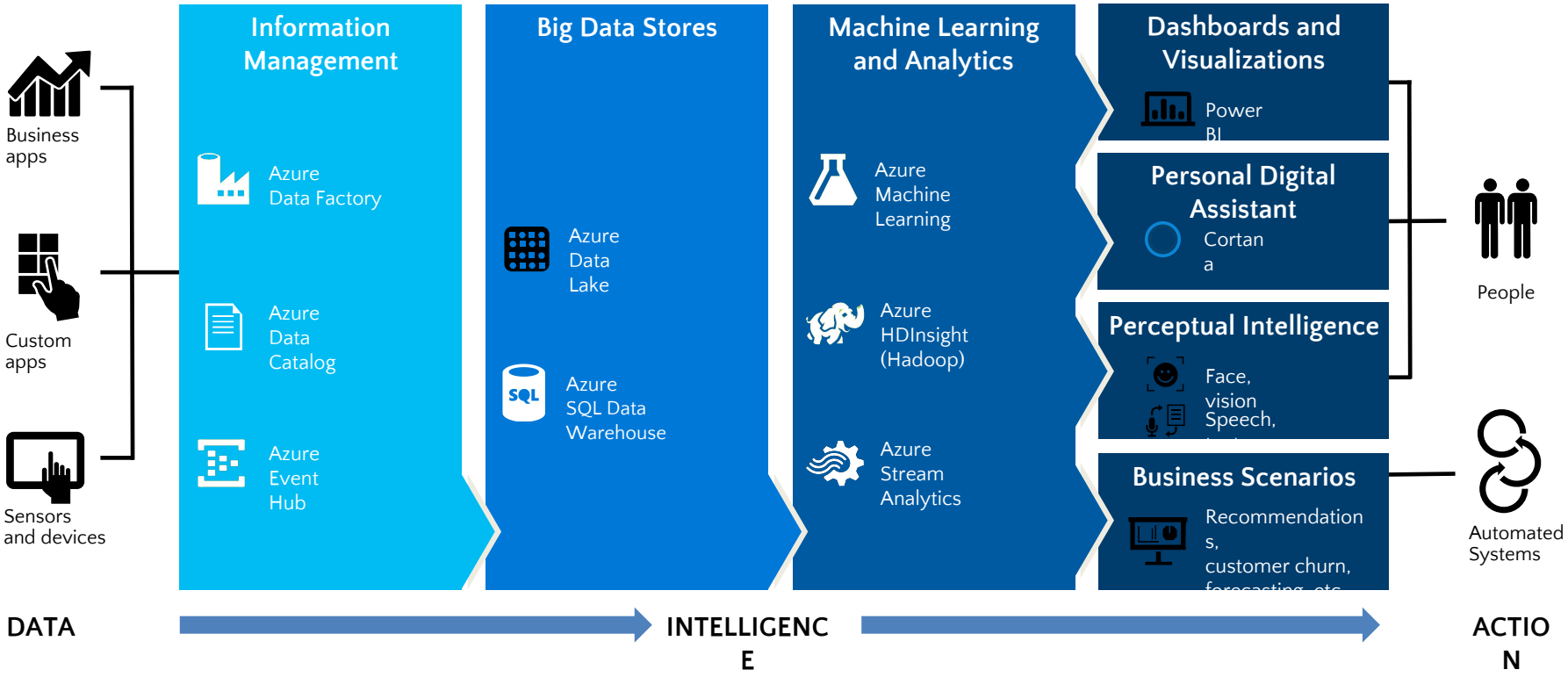
- Model building is just one small part of the story
- You need to put the model into a reproducible analytics production pipeline
- Deployment strategy
- Reproducible strategy

If possible – choose cloud over on premise (unless you have very good DevOps team!)

CLOUD

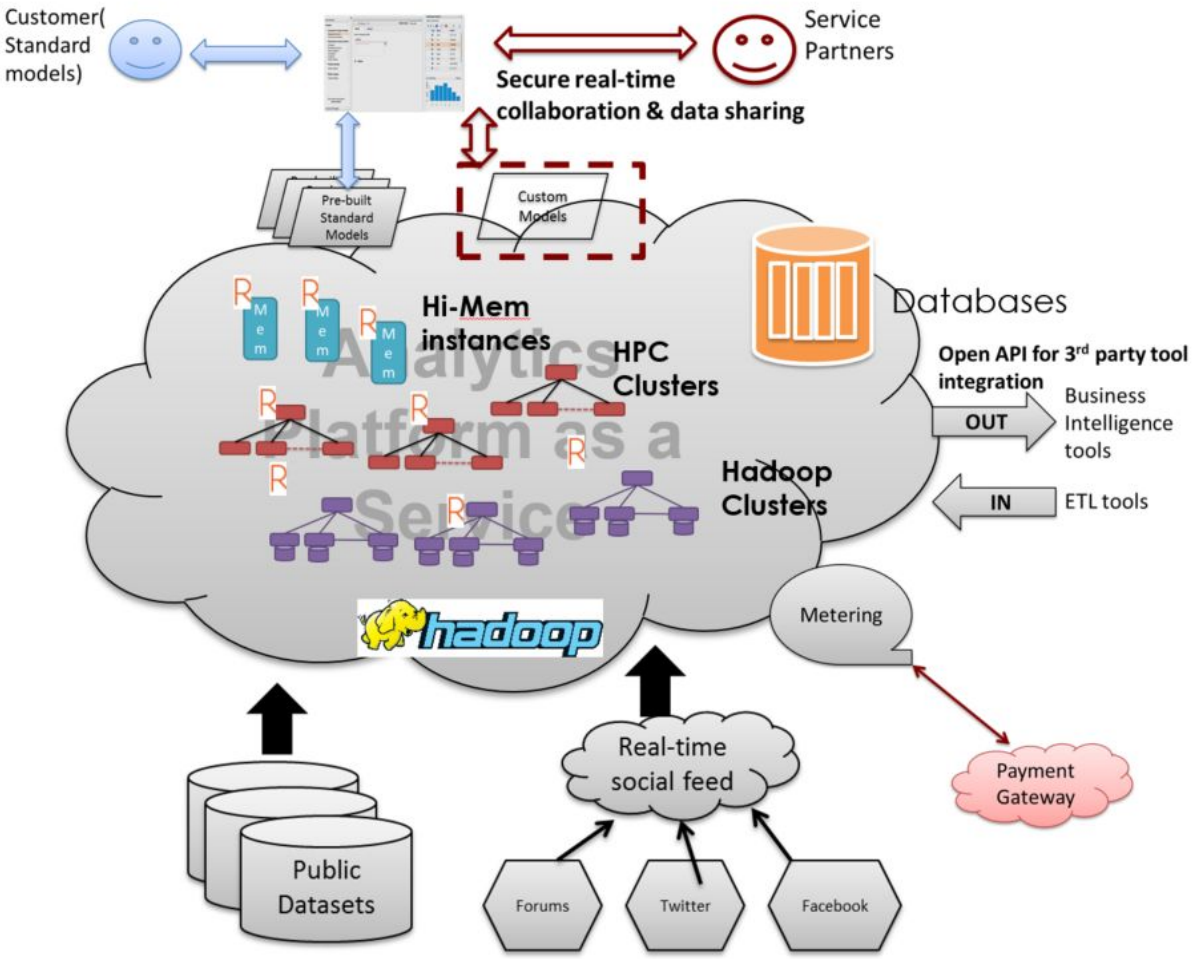
A TYPICAL ANALYTICS CLOUD

What would it take your organization to build the same infrastructure?



Source: Microsoft Cortana Analytics Training

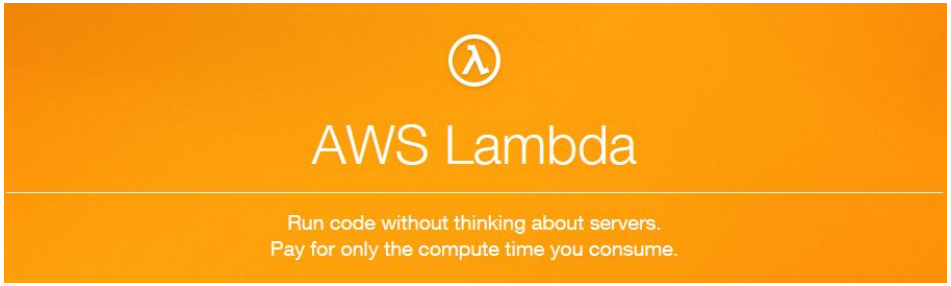
ANALYTICS “CLOUD”

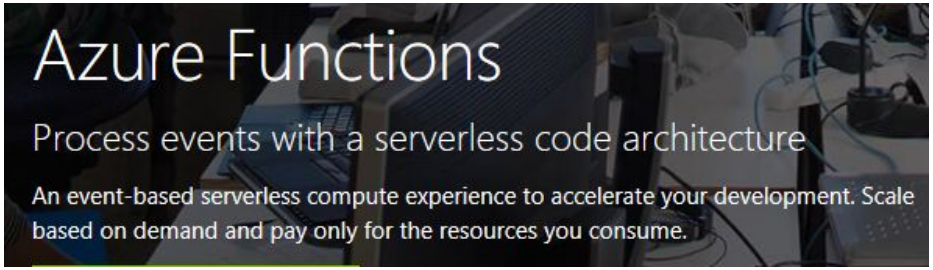


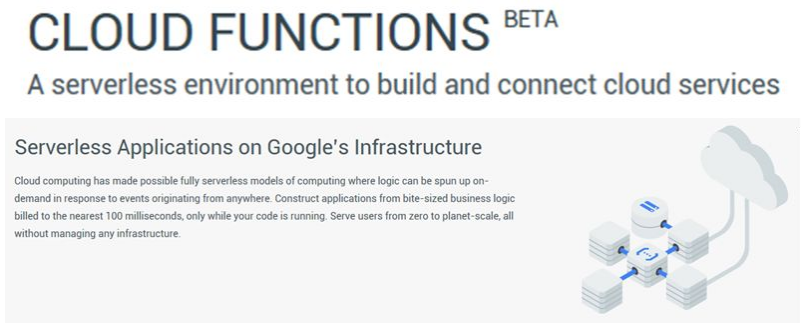
SPEED + COSTS

	Legacy Analytics SW	R running on premise	R running in the Cloud
Rows of data	1 billion	1 billion	1 billion
Parameters	"just a few"	7	7
Time	80 seconds	44 seconds	95 seconds
Data location	In memory	On disk	On disk
Nodes	32	5	8
Cores	384	20	24
RAM	1,536 GB	80 GB	120GB
	US$2.5M	US$20K	US$26
	Months	Weeks	10 minutes

SERVERLESS



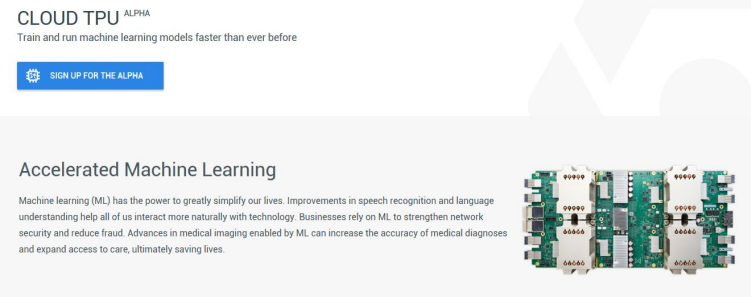


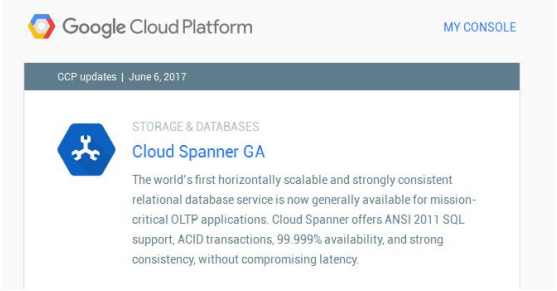


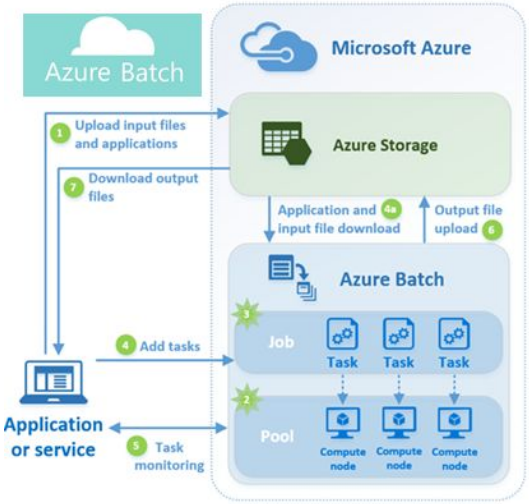
A major bank:
From **$100K** per month to **$60** per month!

EARLY INNOVATIONS

LATEST AND GREATEST







Amazon Lex Now Generally Available

Posted On: Apr 19, 2017

Amazon Lex is now generally available for all customers. Amazon Lex is a service for building conversational interfaces into your application using voice and text. With Amazon Lex, the same deep learning technologies that power Amazon Alexa are now available to developers, enabling you to quickly and easily build sophisticated, natural language conversational bots.

Introducing Amazon Connect

Posted On: Mar 28, 2017

Amazon Connect is a simple to use, cloud-based contact center service that makes it easy for you to deliver better customer service at lower cost. This new service from Amazon Web Services is based on the same contact center technology used by Amazon customer service associates around the world to power millions of customer conversations. Setting up a cloud-based contact center with Amazon Connect is as easy as a few clicks in the AWS Management Console, and agents can begin taking calls within minutes.

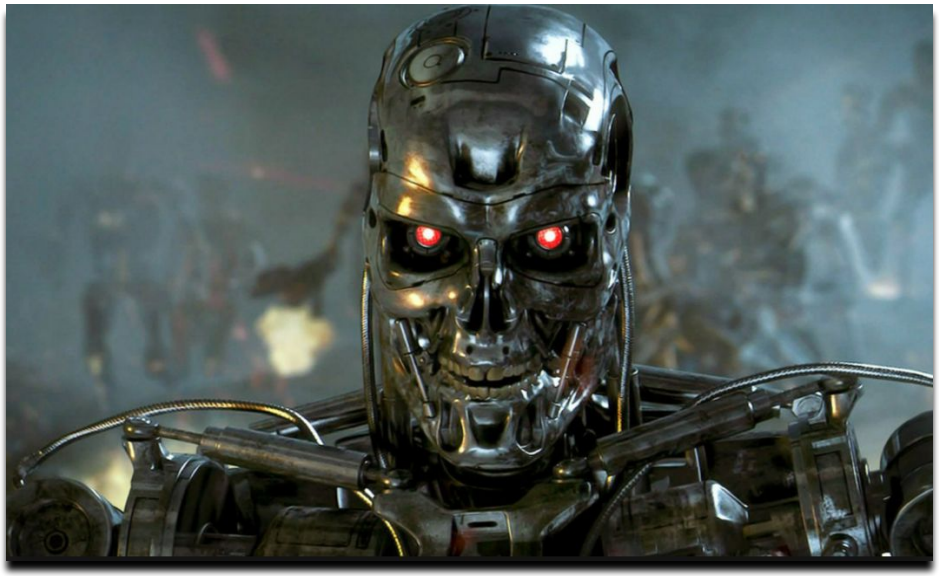
BUY OR CLOUD

- If today, you are able to – please go and use a cloud like Azure, Google or AWS for your analytics workload
  - Ready platform to start immediately with a swipe of the credit card
  - Start fast -> minutes/days vs months!
  - Succeed or fail fast! -> Know fast!
  - No tender, no RFQ
  - Spend in tens – hundreds of $ vs hundreds of thousands $$$$$$$$
  - Focus on your analysis, not infrastructure
    - No need for big highly experienced Devops team!
    - Just a smaller, experienced team
  - Early innovations
- You buy when
  - Company policies
  - You need the air-gap
  - Your workload is 24x7
  - Your have a big in-house team to support your big data infrastructure requirements
    - Assuming you can hire and retain the talent!
- Some things to think about
  - Security (a misconfigured server in the cloud or on-premise is just as vulnerable)
  - Attacks are often within
  - DDOS can bring down your system whether they are on-premise or on the cloud

SUMMARY

- Use public Cloud when you can
- If cannot, demand your IT provide a private cloud with similar self-service capabilities as a public cloud
- The days of waiting 3 days/weeks/months to get a server or database provisioned is over!

In ONE word – what is your idea of AI?



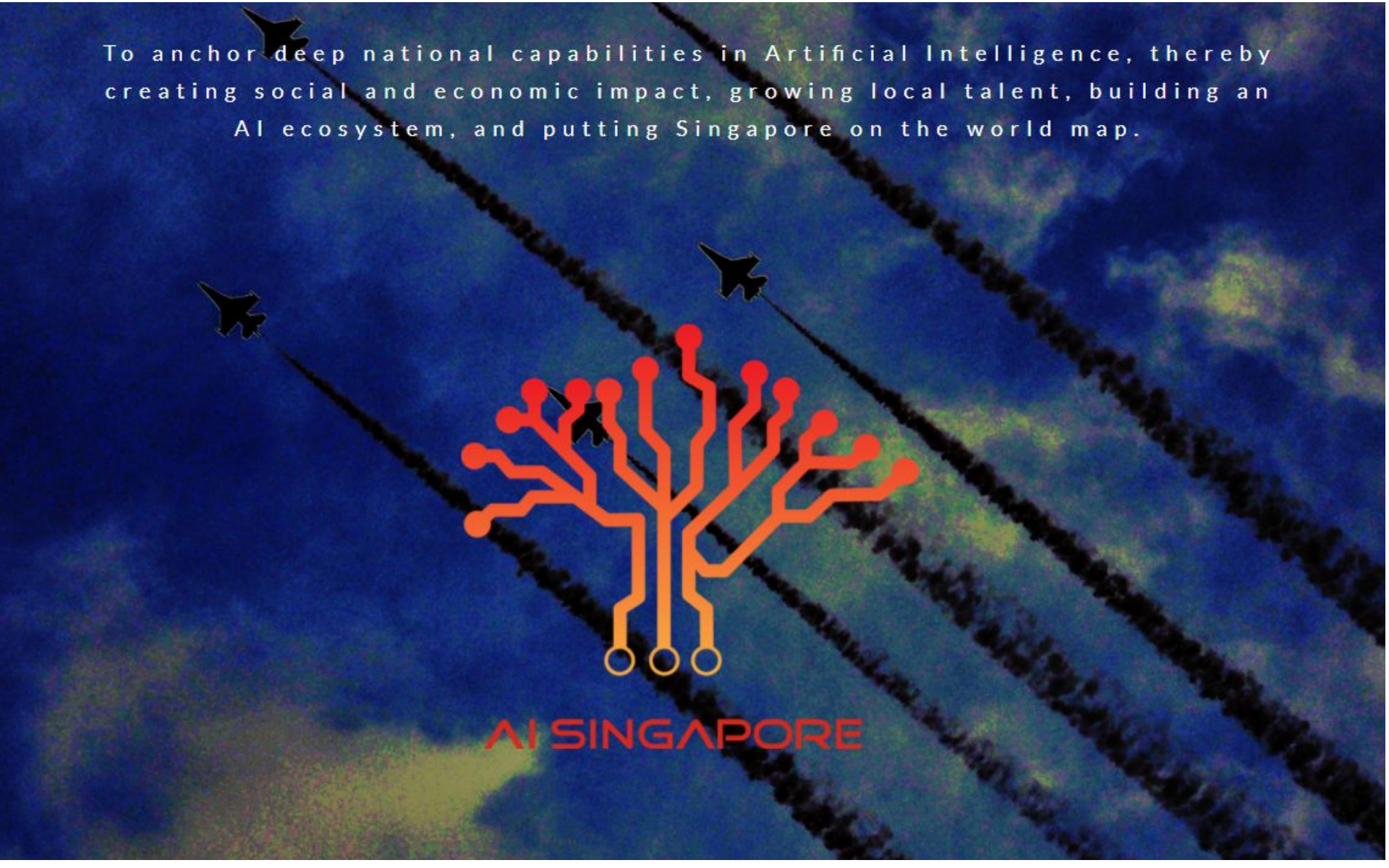
Myths, Hype and Reality

ARTIFICIAL INTELLIGENCE

AI Singapore

SINGAPORE'S ARTIFICIAL INTELLIGENCE CAPABILITIES TO GET S$150M BOOST

About AI Singapore (AISG)



AI SINGAPORE

A KEY PILLAR IN SUPPORT OF SMART NATION

Artificial intelligence (AI) can lead to substantial economic growth and vastly improve our standard of living. AI Singapore (AISG) is a national initiative to nurture talents in AI, increase AI industry translation and commercialisation, drive innovation and position Singapore at the forefront of the global AI industry.

Objectives of AISG

- Create impactful solutions to major societal and industrial challenges
- Seed AI innovation in homegrown companies
- Achieve breakthroughs in AI research

National Investment

$150 million

Through AISG, the **National Research Foundation (NRF)** will invest $150 million over 5 years in building top-tier AI capabilities and other game-changing solutions.

- 100 Experiments — AI Projects + Proof of Concept
- Grand Challenges — Affecting Society + Industry
- Emerging Research — The next wave of scientific innovation

Focus Application Areas

AISG will perform use-inspired research to solve **global** challenges and improve Singaporeans' lives.

1 Health

Seamlessly integrated care environments powered by humans and machines, personalised treatments and accelerated drug discovery.

2 Smart Cities

Using technology, real-time monitoring and data acquisition to manage urban ecosystems, infrastructure and tackling inefficiency.

3 Finance

Automation and democratisation of financial services, machine learning-powered trading, risk modelling and compliance.

AISG Partners

- National Research Foundation
- Smart Nation & Digital Government Office
- Economic Development Board
- Infocomm Media Development Authority
- Integrated Health Information Systems
- SGInnovate
- Monetary Authority of Singapore

Government Agencies

Large Local Enterprises

Institutes of Higher Learning

- Nanyang Technological University Singapore
- Singapore University of Technology and Design
- SMU Singapore Management University
- NUS National University of Singapore

SMEs & Start-ups

MNCs

Individuals

Research Institutions

Agency for Science, Technology and Research Singapore

AISG will leverage on the universities' areas of expertise in multidisciplinary capabilities and technologies that will aid research and innovation in AI.

Likewise, access to valuable databases and strengths in areas like machine learning, natural language processing (NLP) and robotics are crucial to the success of the AISG programme.

NUS National University of Singapore

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

SMU SINGAPORE MANAGEMENT UNIVERSITY

SUTD SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

A*STAR SINGAPORE Agency for Science, Technology and Research

AI APPRENTICESHIP PROGRAMME

Training real-world AI/ML Engineers

AI Apprenticeship Program (AIAP)

3-months
Hybrid classroom, online, projects

6-months
100E projects as part of capstone projects

Back to industry

<3

30th Dec 2017

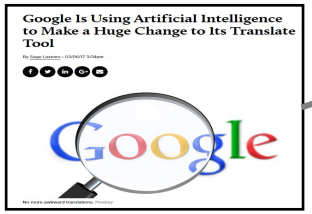
WHAT IS YOUR IDEA OF AI

AI TODAY

STATE OF ROBOTICS

What is AI?

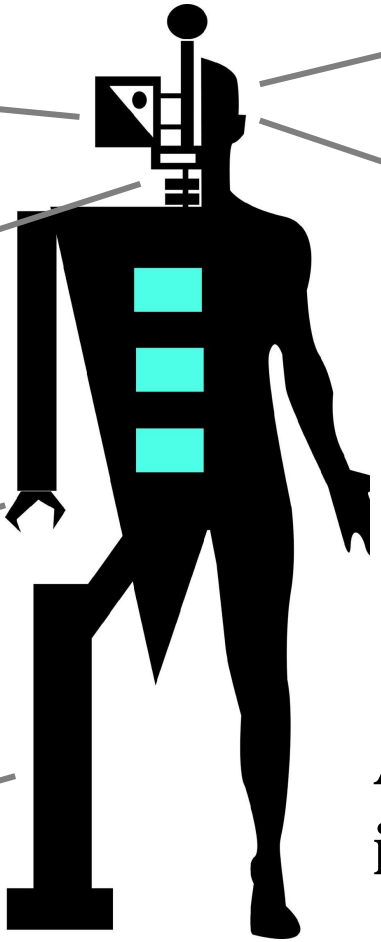
See



Speak

Feel

Move

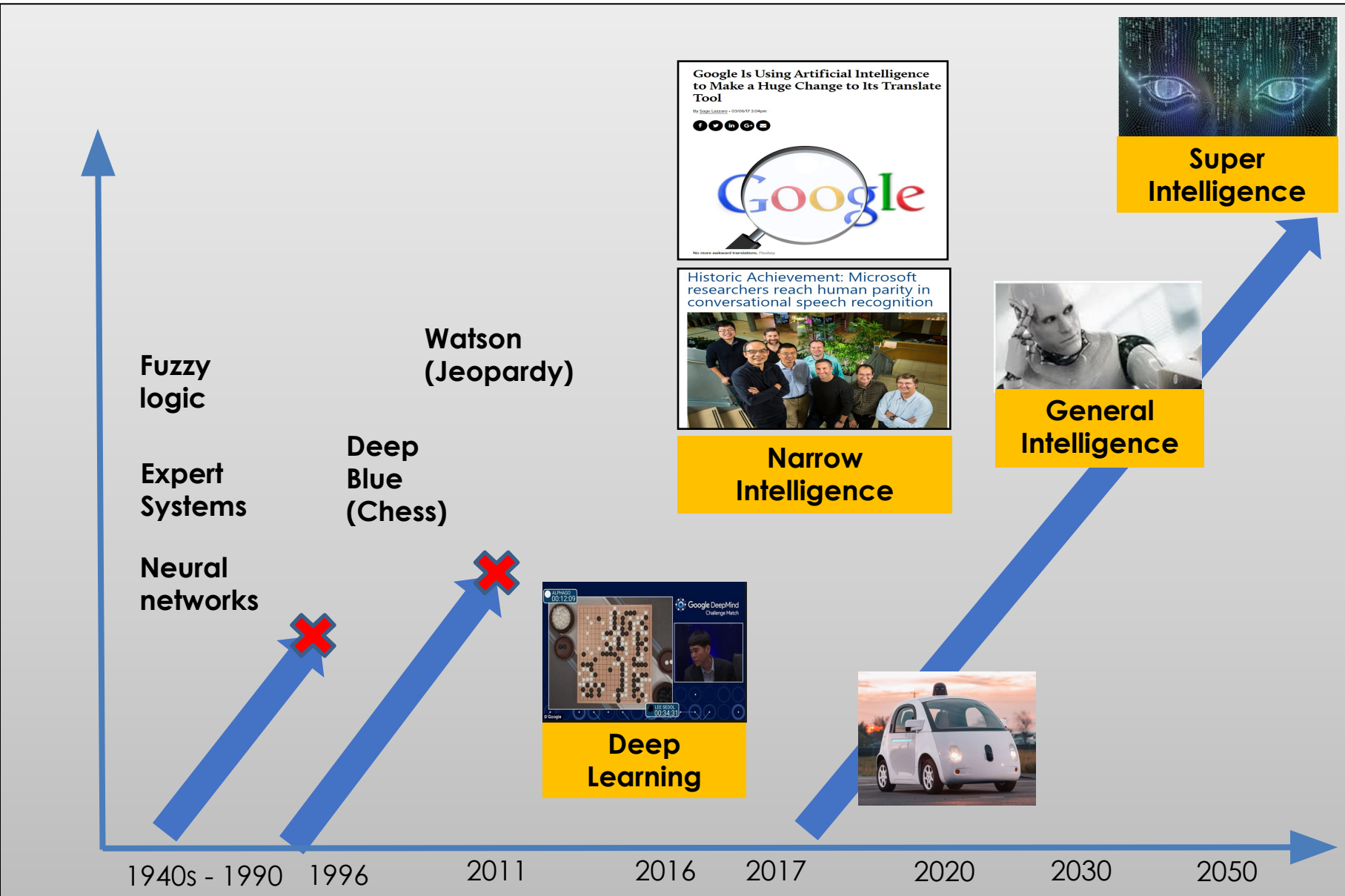


Learn

Hear

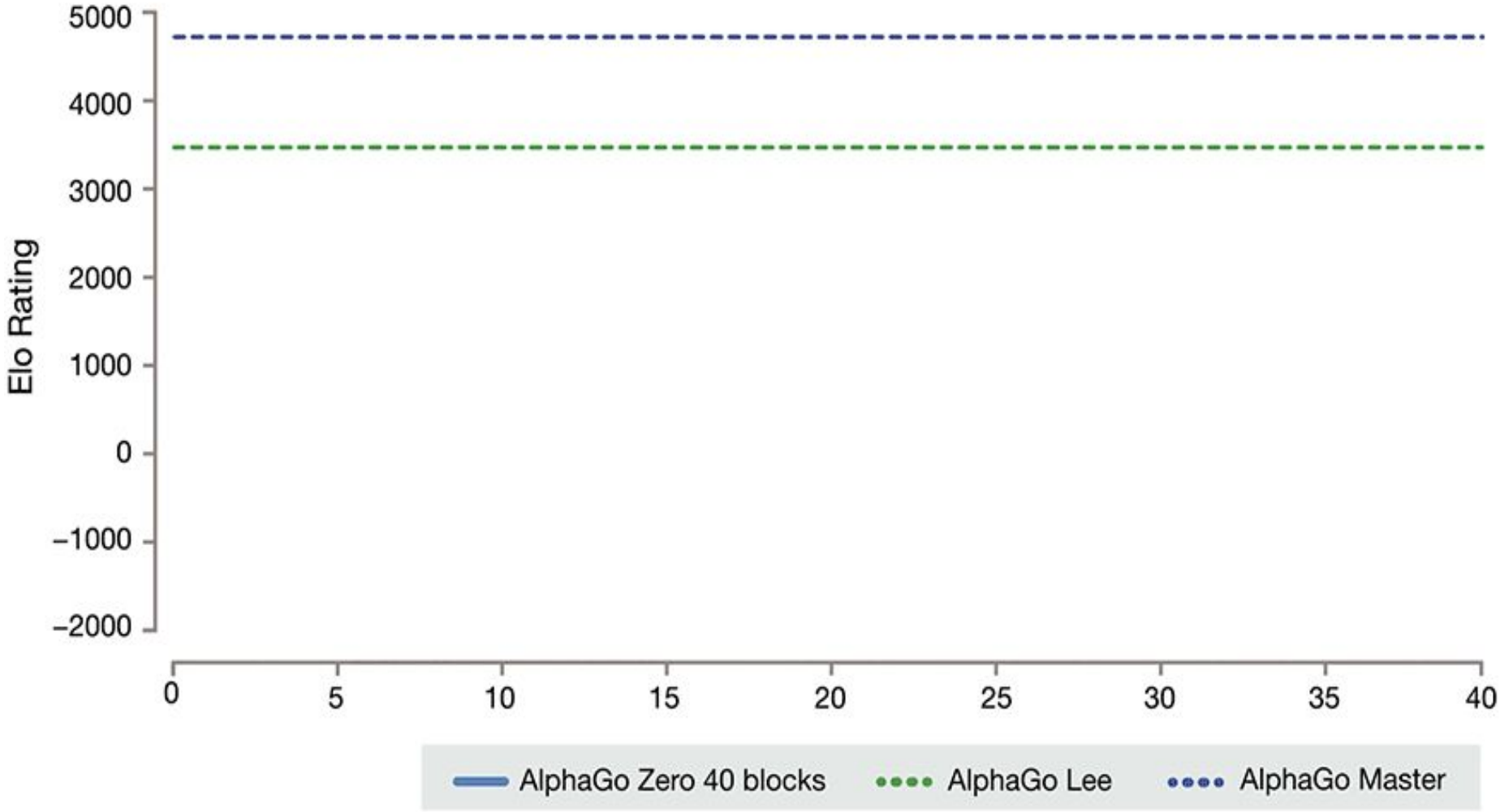
Artificial intelligence

AI TRENDS


Google Is Using Artificial Intelligence to Make a Huge Change to Its Translate Tool
Google
Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition
Super Intelligence
General Intelligence
Narrow Intelligence
Deep Learning
Watson (Jeopardy)
Deep Blue (Chess)
Fuzzy logic
Expert Systems
Neural networks
1940s - 1990
1996
2011
2016
2017
2020
2030
2050

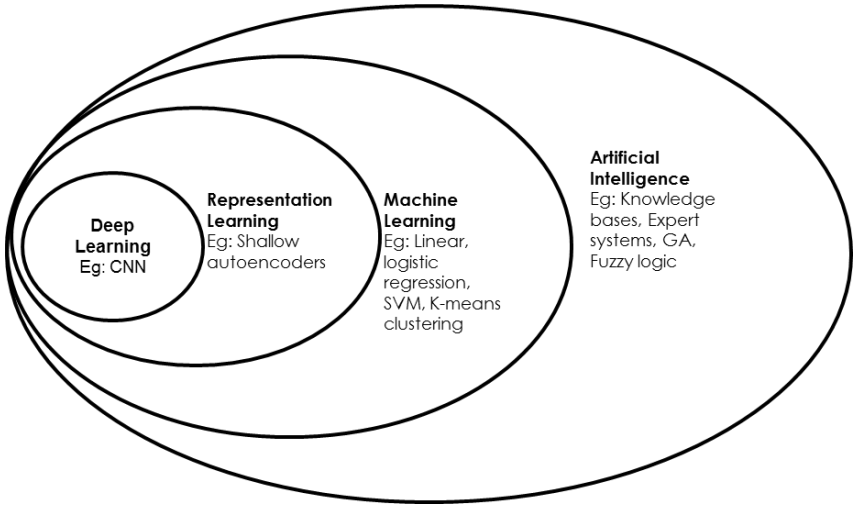
SELF-LEARNING AI

AlphaGo Zero beats AlphaGo



AI IS NOT JUST DEEP LEARNING!

- Rules based Expert Systems
- Fuzzy Logic
- Genetic Algorithms
- Case-Base Reasoning
- Neural Networks/Deep Learning
- Hierarchical Temporal Memory (HTM)

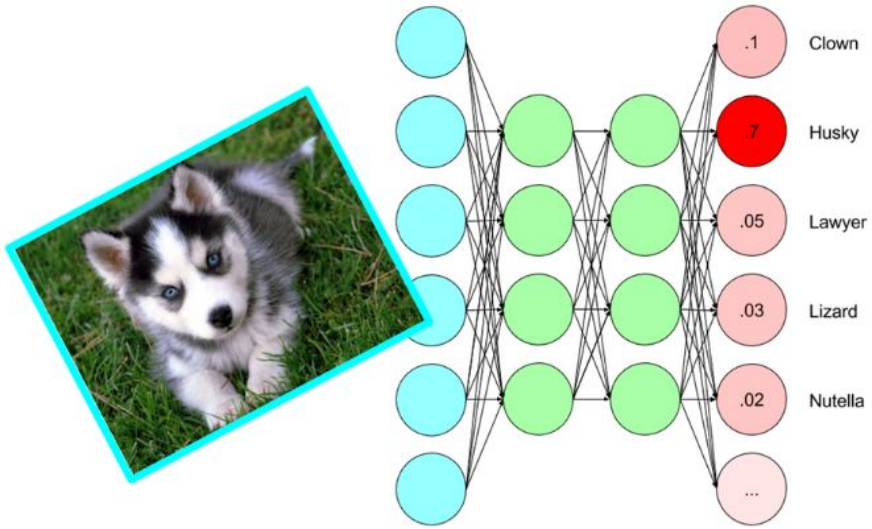


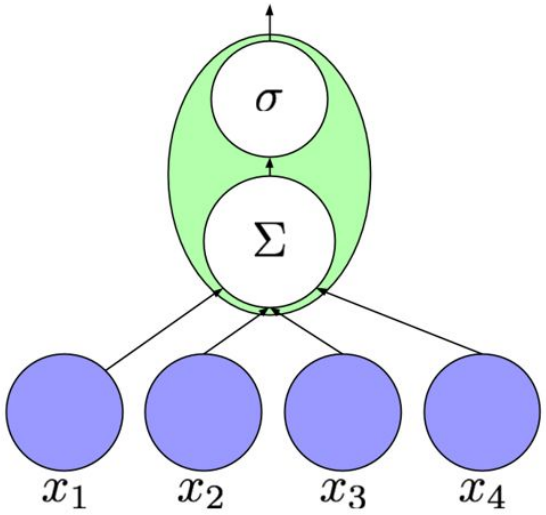
THE AI STACK


INTEL® NERVANA™ PORTFOLIO
EXPERIENCES
PLATFORMS
Intel® Nervana™ Cloud & Appliance
Intel® Nervana™ DL Studio
Intel® Computer Vision SDK
Movidius Fathom
intel Saffron
FRAMEWORKS
Spark
MLLIB
BIGDL
neon
TensorFlow
theano
mxnet
Caffe
torch
Caffe2
Chainer
LIBRARIES
python
Intel Python Distribution
Intel® Data Analytics Acceleration Library (DAAL)
Intel® Nervana™ Graph*
Intel® Math Kernel Library (MKL, MKL-DNN)
HARDWARE
ATOM
CORE i7
ARRIA 10
XEON
XEON PHI
More
Compute
Memory & Storage
Networking
*Future
INSIDE AI
intel Nervana

DEEP LEARNING

NEURAL NETWORKS





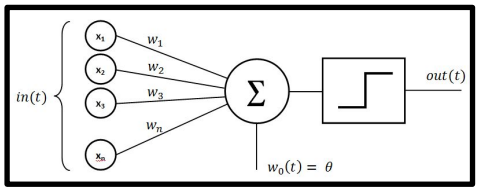
Source: http://mxnet.io/get_started/why_mxnet.html

Think of neural networks as functions for transforming input arrays X into output arrays Y.

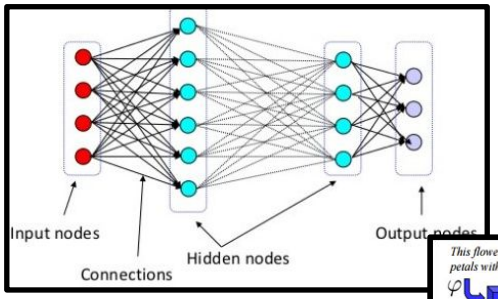
For example, X might be the pixel values of an image, and Y might represent the corresponding probabilities that the image belongs to a cat (or dog, tiger etc)

DEEP LEARNING - EVOLUTION

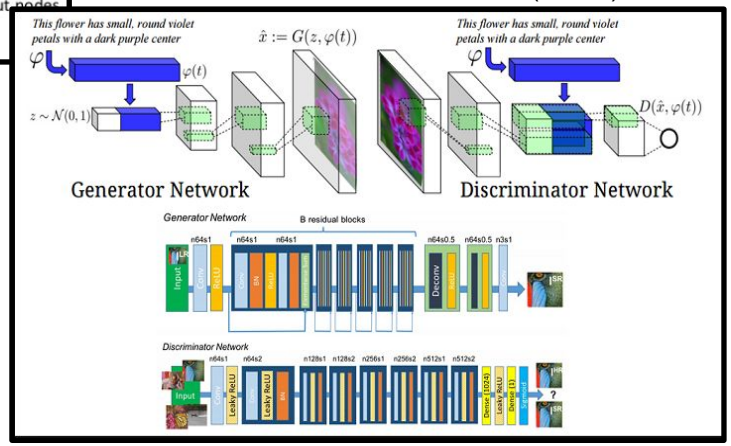
Perceptron 1958



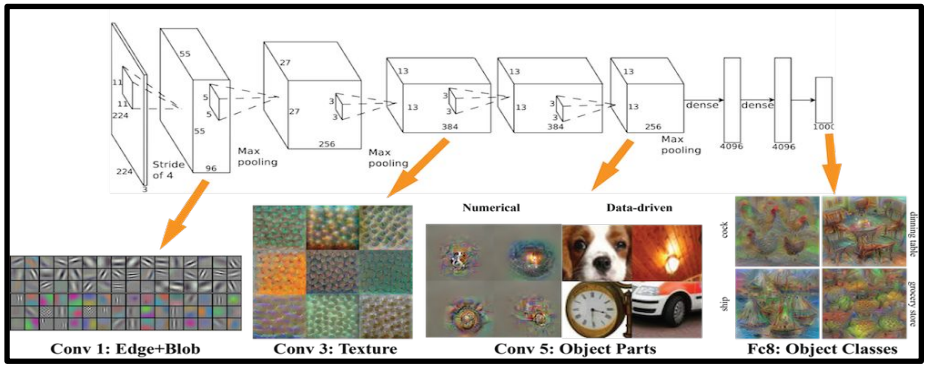
Neural Networks in 1970s-1990s



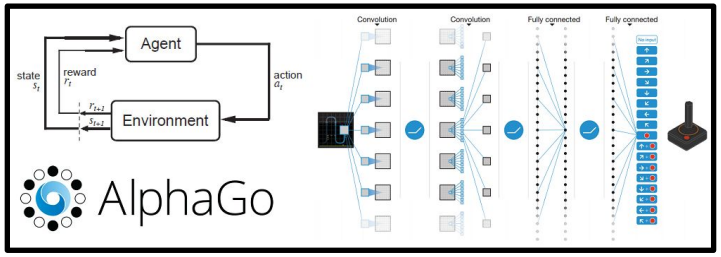
Generative Adversarial Network (GANs)



Convolution Networks (CNN) 2012



Reinforcement Learning



AI VS TRADITIONAL PROGRAMMING

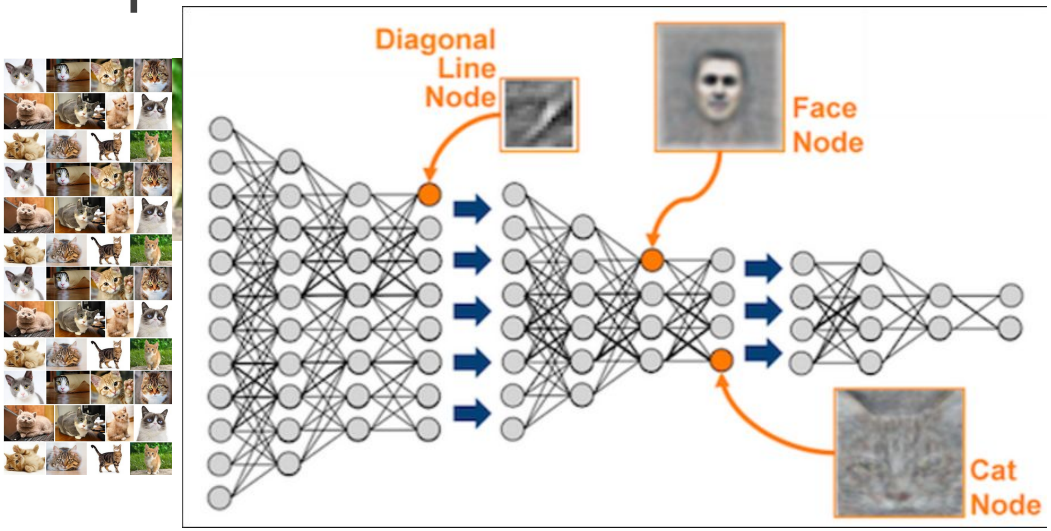
1) Handle complexity
   - no programmer have managed to program code to recognized a cat or do language translation!

2) Easier to show and tell then to code
   - Faster! More efficient.

3) More accurate!
   - latest Google/Microsoft speech recognition is on par with human and sometimes better!

AI 101

- No magic… yet…
- You need to either tell it what to do
  - Rules (expert systems) – was not very successful in the 1980s-1990s
- Show it enough examples
  - This is a cat!
  - Neural networks
  - Deep Learning



CLOSING THOUGHTS..