Big Data Architectures

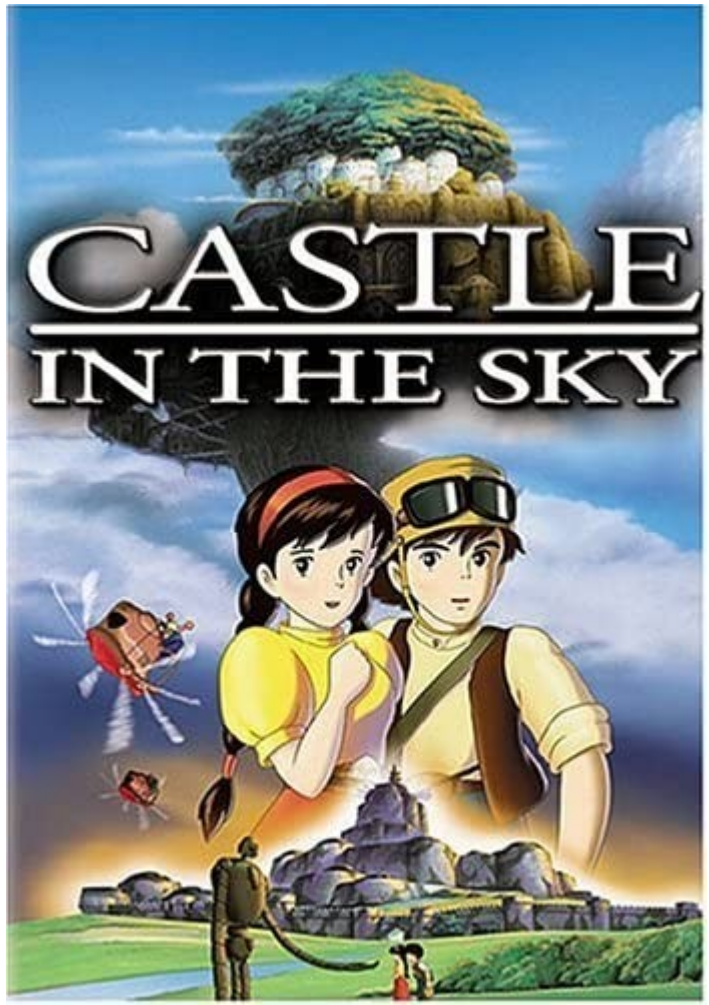
and the News Hunter case

Marc Gallofré Ocaña

marc.gallofre@uib.no

University of Bergen

What do they have in common with Twitter?



August 3, 2013

What do they have in common with Twitter?

Made Twitter crash

Achieved one of the all-time records of tweets in a second.

August 3, 2013

How many tweets are sent every day?

How many tweets are sent every day in avg?

- A) less than 1 million
- B) more than 500 million
- C) between 10 and 50 million
- D) around 100 million

“New Tweets per second (TPS) record: 143,199 TPS. Typical day: more than 500 million Tweets sent; average 5,700 TPS.”

- Raffi Krikorian, VPE Twitter, 2013

Recommended lecture: https://blog.twitter.com/engineering/en_us/a/2013/new-tweets-per-second-record-and-how

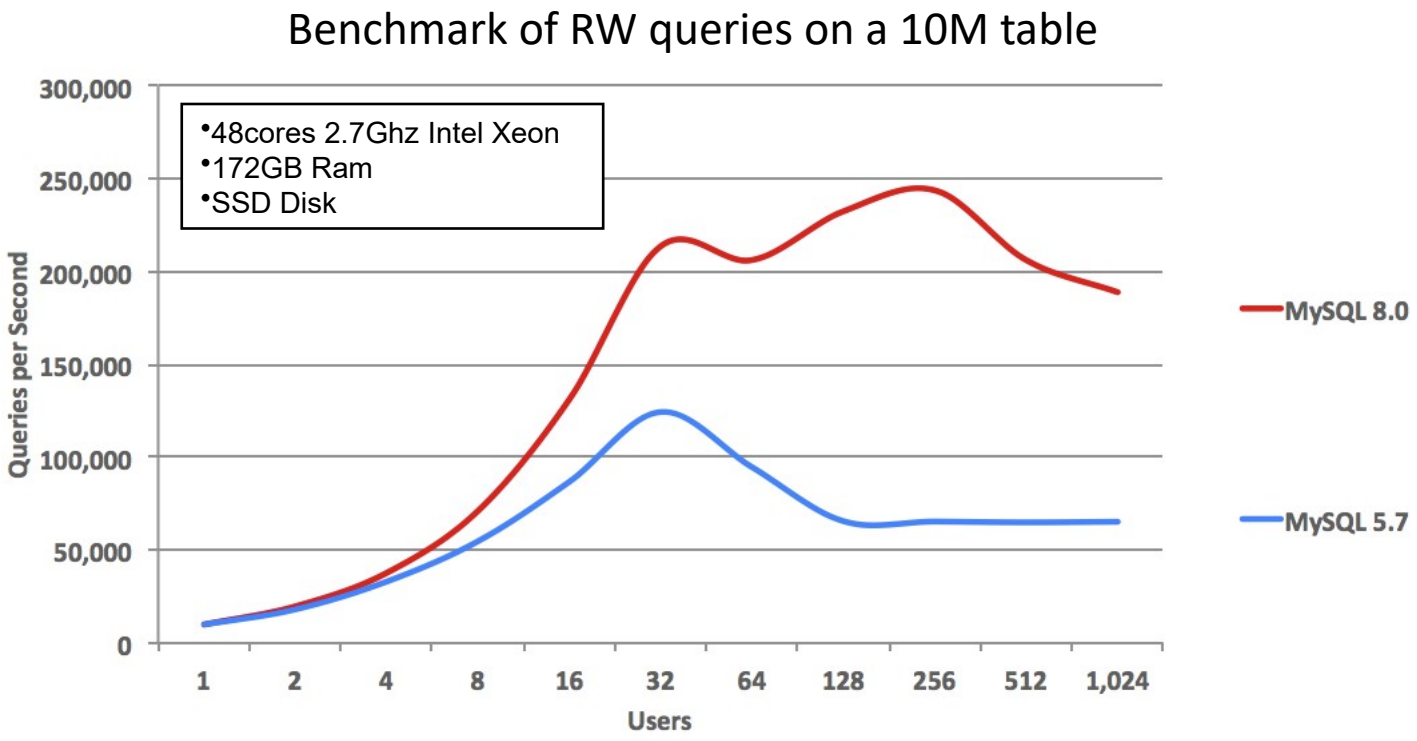
What may happen to a non-big data system when it reaches these volumes?

What may happen to a **non**-big data system when it reaches these volumes?

- A) Nothing, system are made to handle large amounts of data.
- B) Systems may crash, become unavailable and not respond.
- C) The engineering team has work overnight to fix the problems.
- D) The system needs to be redesign with new architecture and technology.

What happens when data goes big?

MySQL limits:
- Tablespace: 256TB
- Row limit: 65556 bytes
- Max of ca. 3-4 Billion (10^9) records



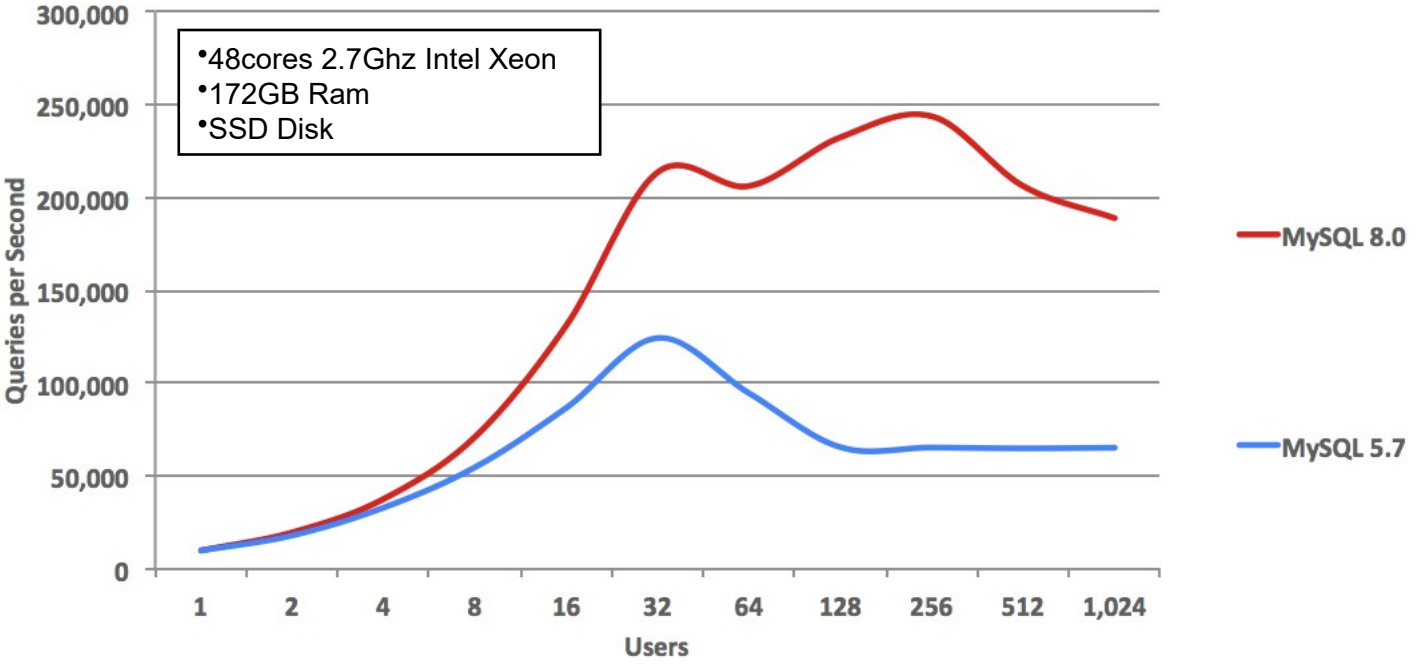
https://dev.mysql.com/doc/refman/8.0/en/innodb-limits.html
http://dimitrik.free.fr/blog/posts/mysql-performance-80-and-sysbench-oltp_rw-updatenokey.html

What happens when data goes big?

MySQL limits:
- Tablespace: 256TB
- Row limit: 65556 bytes
- Max of ca. 3-4 Billion (10^9) records

Twitter numbers:
- 500 M tweets x 7 days = 3.5 B
- It crashes in less than week

Benchmark of RW queries on a 10M table



https://dev.mysql.com/doc/refman/8.0/en/innodb-limits.html
http://dimitrik.free.fr/blog/posts/mysql-performance-80-and-sysbench-oltp_rw-updatenokey.html

What happens when data goes big?

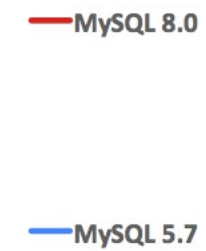
Benchmark of RW queries on a 10M table

MySQL lim
- Tablespac
- Row limit
- Max of ca

Limits exist beyond databases.

Computational resources like Memory, CPU, GPU have limits too!

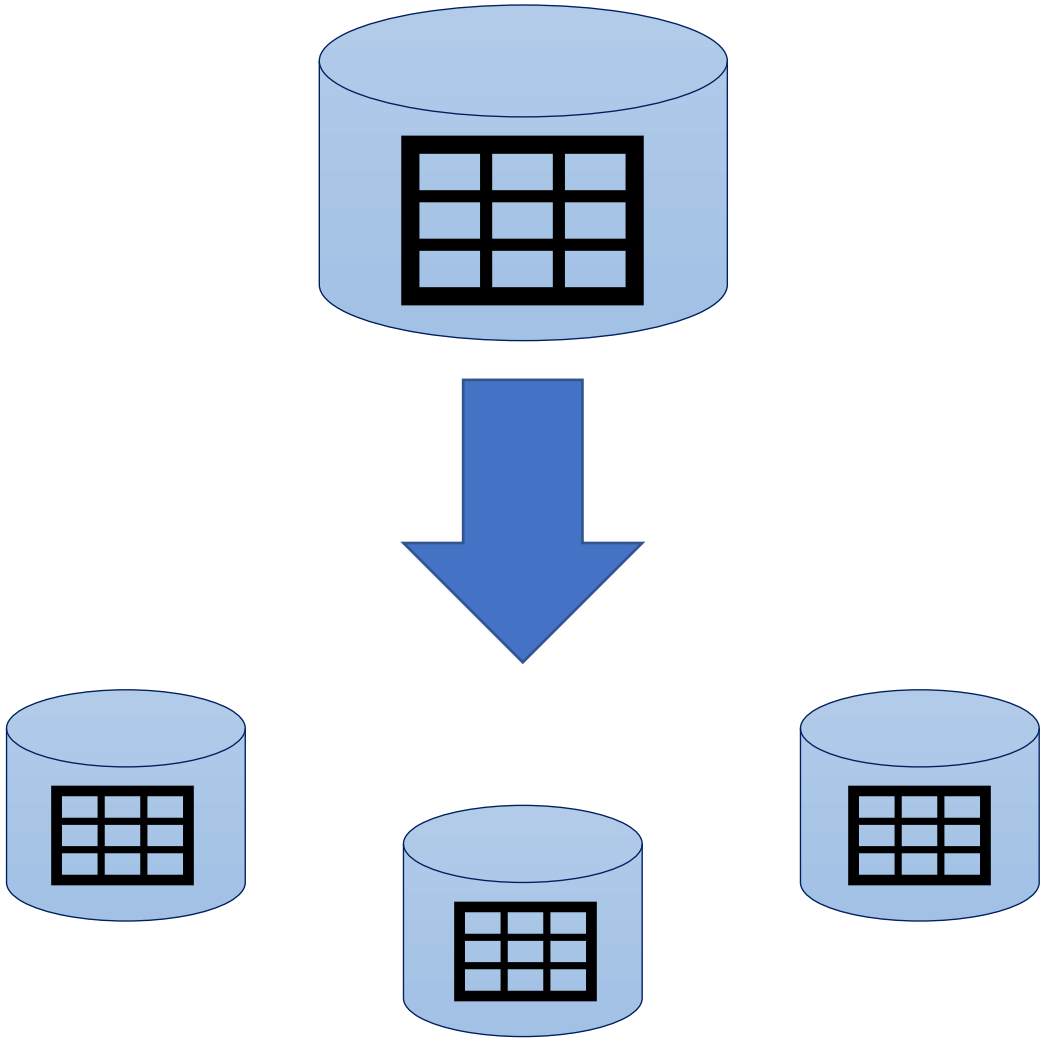
Twit
- 500
- It crashes in less than week



https://dev.mysql.com/doc/refman/8.0/en/innodb-limits.html
http://dimitrik.free.fr/blog/posts/mysql-performance-80-and-sysbench-oltp_rw-updatenokey.html


DIVIDE
AND
CONQUER!
ONAPARTE

Distribute data and system

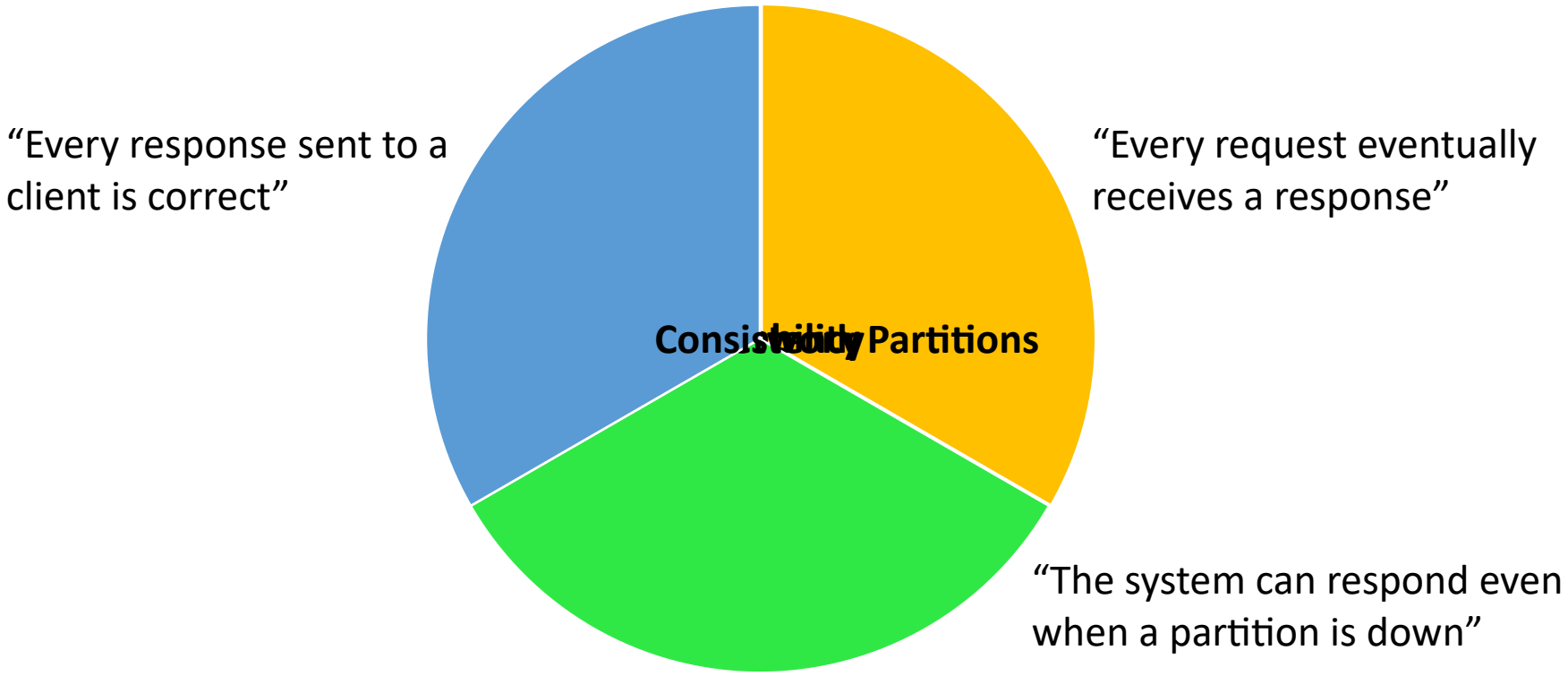


What can happen when your system is distributed?

What can happen when your system is distributed?

- A) You can lose the communication between your components.
- B) Data may not be available sometimes or have long response waiting times.
- C) You can get different results.
- D) Nothing, distributing the system only improves the performance.

CAP theorem (a tradeoff for distributed systems)



Brewer, E. A. (2000, July). Towards robust distributed systems. In *PODC* (Vol. 7, No. 10.1145, pp. 343477-343502).

S. Gilbert and N. Lynch, "Perspectives on the CAP Theorem," in *Computer*, vol. 45, no. 2, pp. 30-36, Feb. 2012, doi: 10.1109/MC.2011.389.

Software Architectures

Monolithic vs Microservices architectures

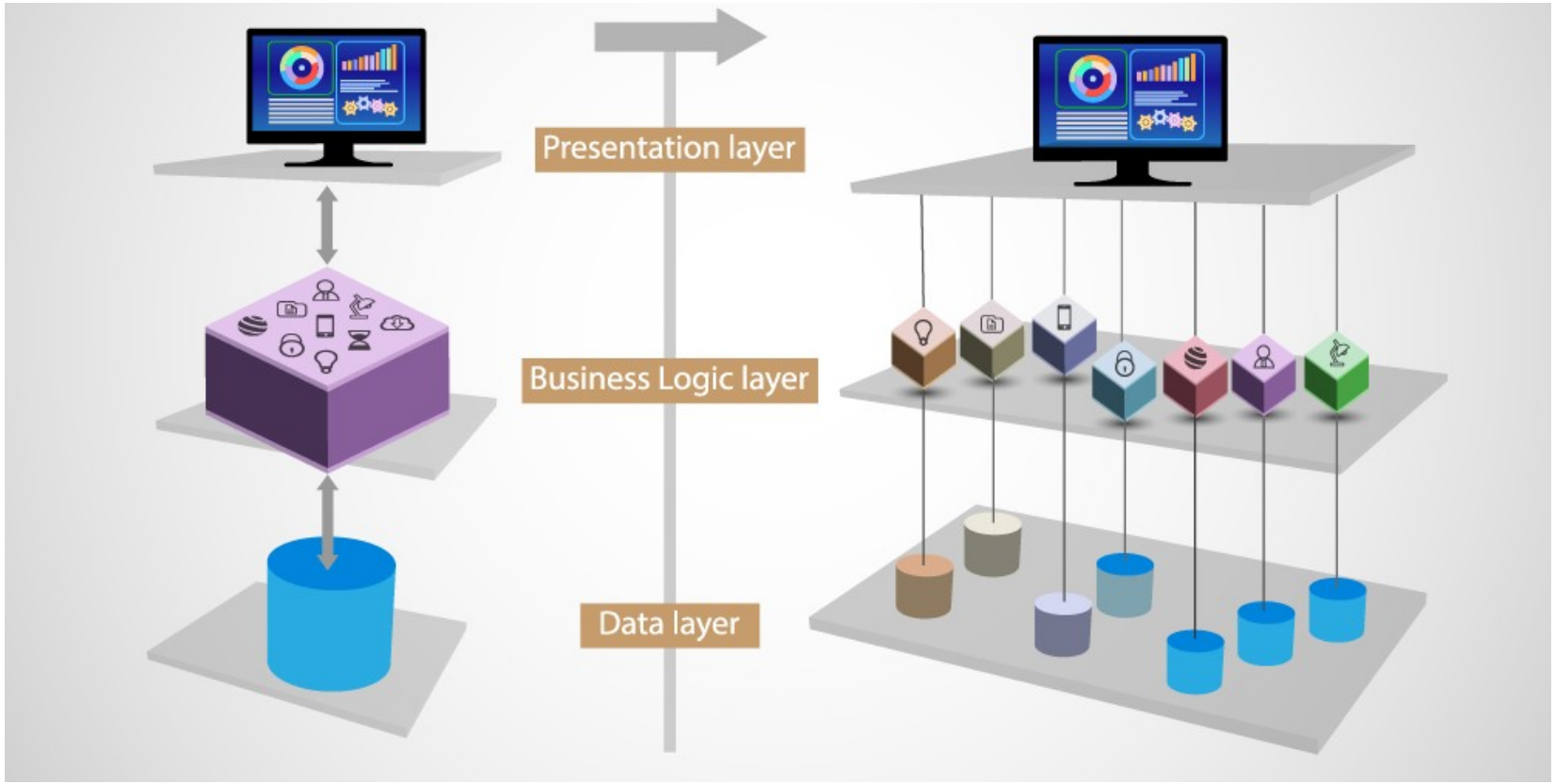


Image modified from: https://www.flagship.io/migrating-from-monolith-to-microservices/

Monolithic vs Microservices

Monolithic

Microservices

Monolithic vs Microservices

Monolithic

- Easier and faster to deploy, all code in the same codebase.
- Difficult to scale and distribute.
- Difficult to maintain.
- Tightly coupled processes.
- Maybe good for small applications.
- Use a single technology stack, difficult the integration.
- Easier to secure.
- An error effect the whole system.

Microservices

- More effort to develop.
- Easier to distribute and scale.
- Easier to maintain.
- Lousy coupled components.
- Can easily combine and integrate different technologies.
- Many independent components and API that need to be secured.
- An error effect one service.

So far, we know how to split our systems and the consequences.
But how do we process the data?

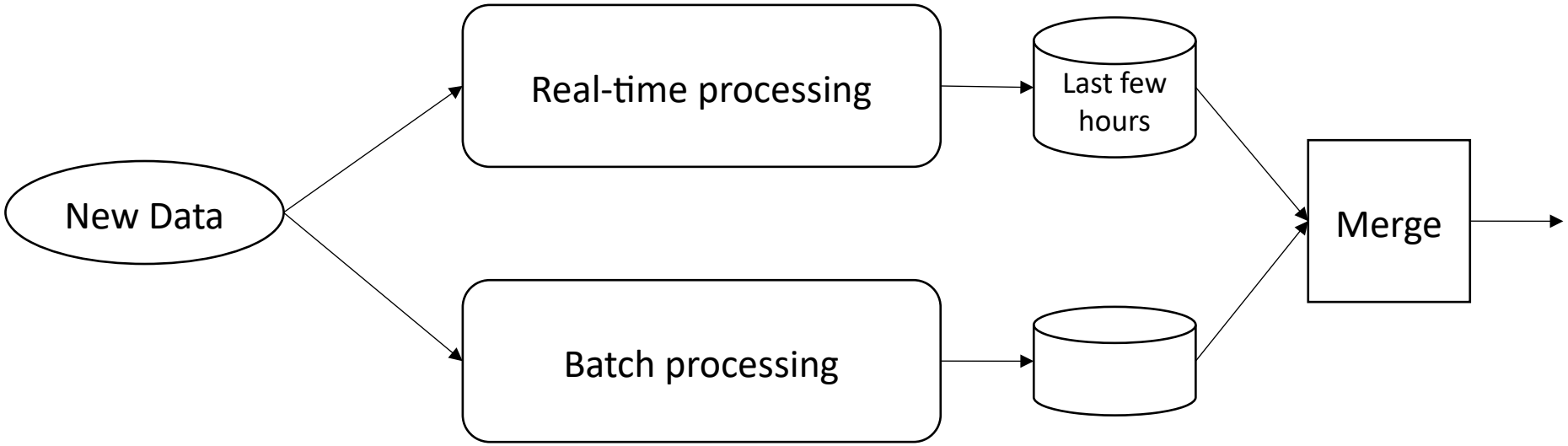
Big Data processing architectures

Since the CAP theorem was announced different Big Data architectures has been proposed to structure data processing. Most of them are based on:

1. Lambda (λ) architecture (from Nathan Marz, Twitter, 2011)
2. Kappa architecture (from Jay Kreps, Linkedin, 2014)

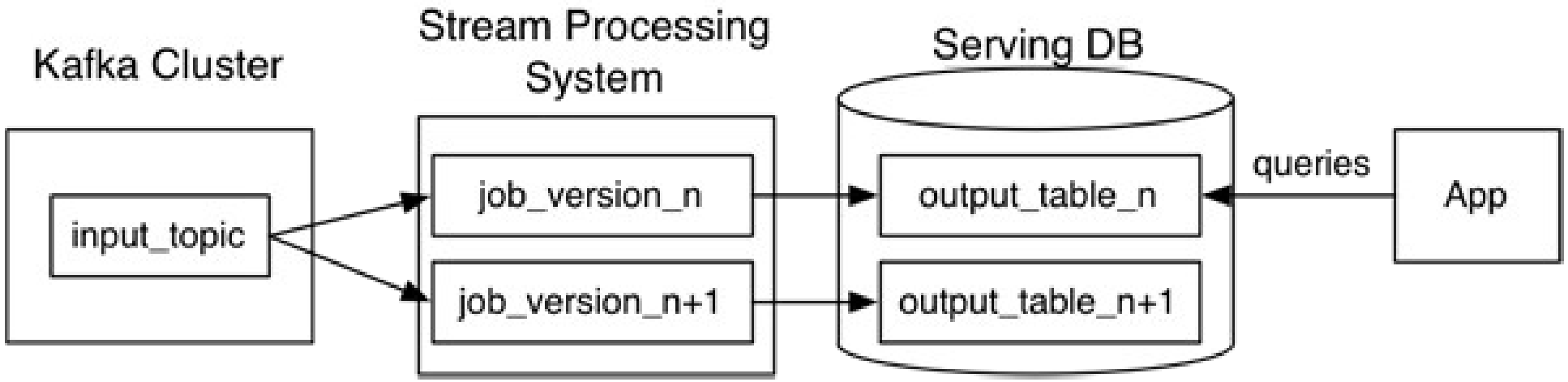
Lambda: http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html
Kappa: https://www.oreilly.com/radar/questioning-the-lambda-architecture/

Lambda



- Data is immutable, there are no “updates” (past data does not change in the future).
- Batch layer processes the whole dataset of raw data (few hours). ←HDFS, MapReduce, Spark + Indexing stored data (ElasticSearch, Redis)
- Real-time does stream processing over the most recent data using incremental algorithms. ← Storm, SparkStream + Cassandra/Druid/Scylla.
- Results from all the dataset except the last few hours (batch layer) are merged with the last few hours (real-time layer) to serve the queries.
- Data is sent to both real-time and batch processing computation processes.

Kappa



- No batch, everything is stream. If you have a bigger dataset, just increment the parallelization of your system.
- It uses systems like Kafka that allows to retain the data log for longer periods (e.g., 30 days and couple of months).
- If something change (code, algorithm) reprocess the data log and when the data log is ready, change the view and discard the old.
- Needs more data space.

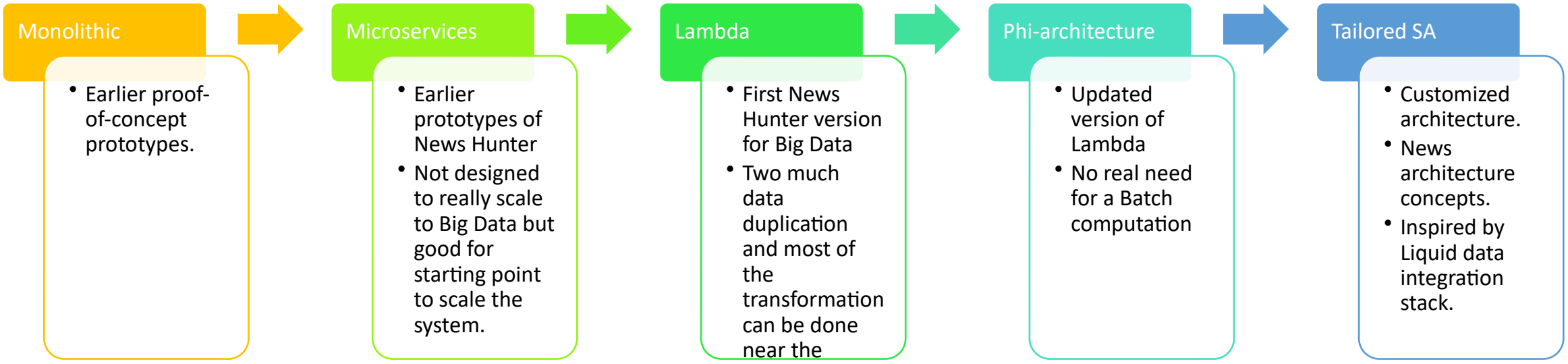
The News Hunter

News Hunter (News Angler project)

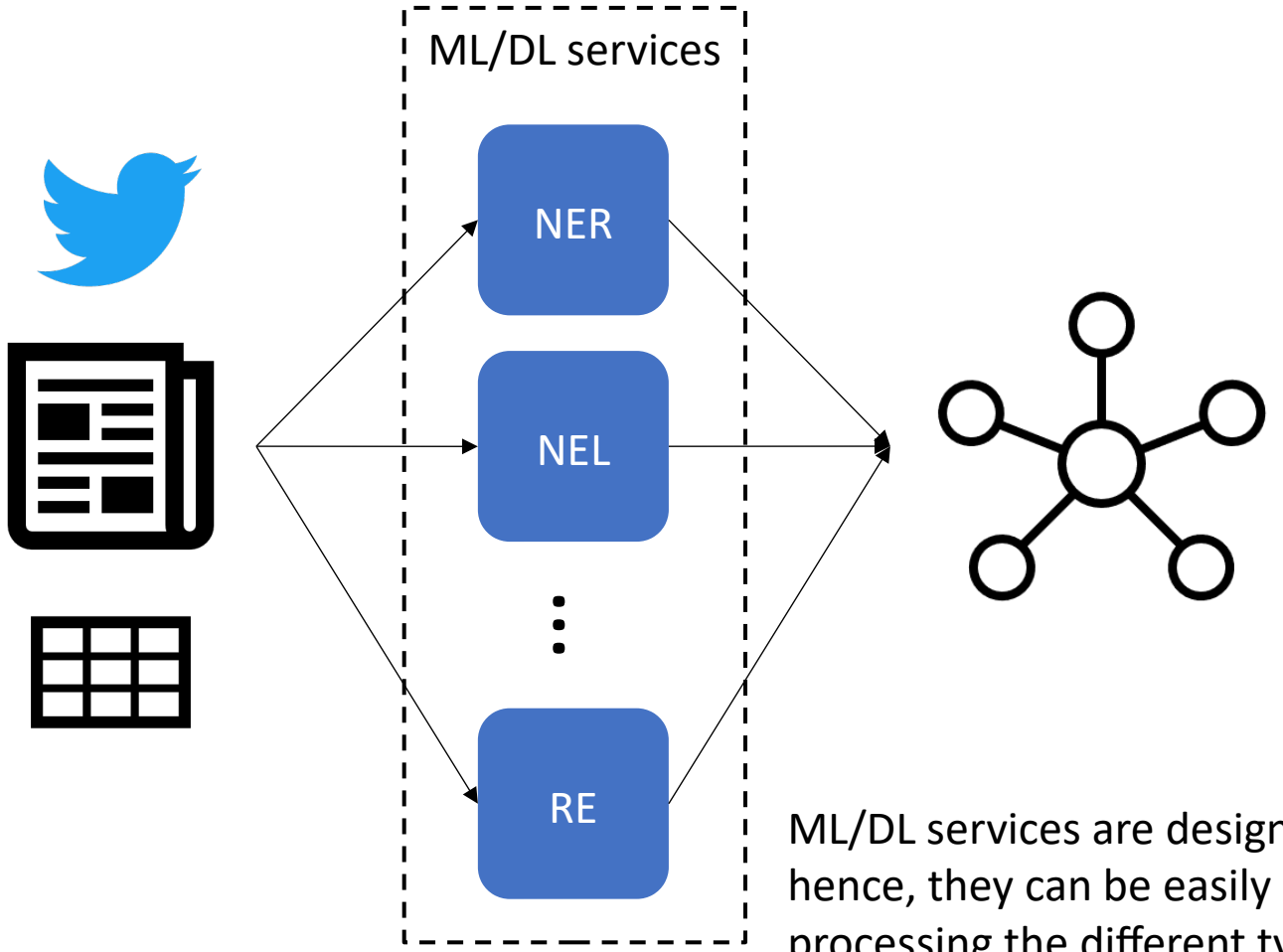
- We work with news to support journalists!
- More than 100.000 news are published daily.
- We also use real-time social media (Twitter) and semi-structured data (GDELT).
- We want to explore semantic connections and relations between news. To do so, we use Knowledge Graphs and semantic data and technologies.
- As the world changes, our information too.

→ We need a system to fulfil our goals.

The evolution (simplified version)

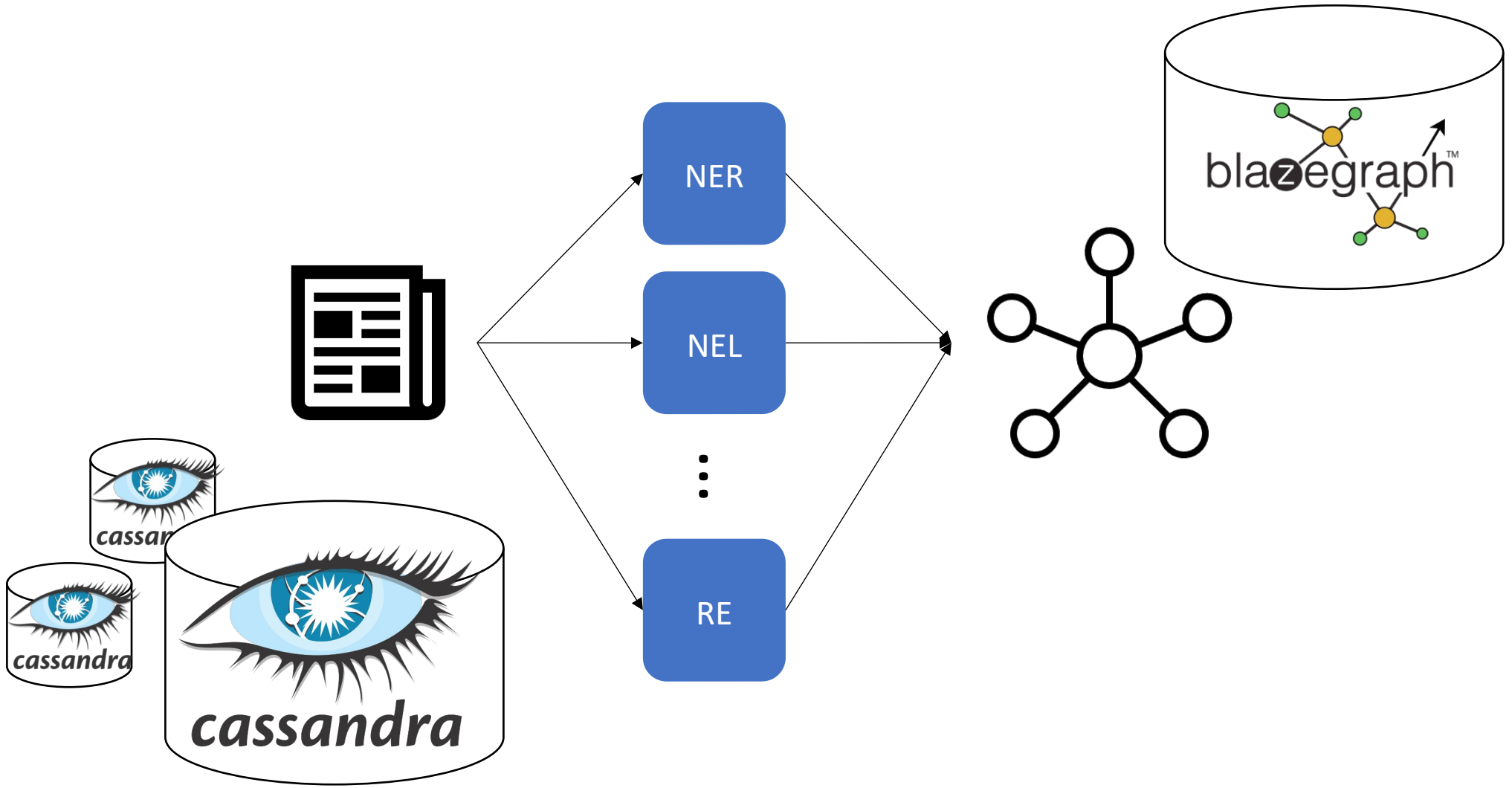


Data transformation

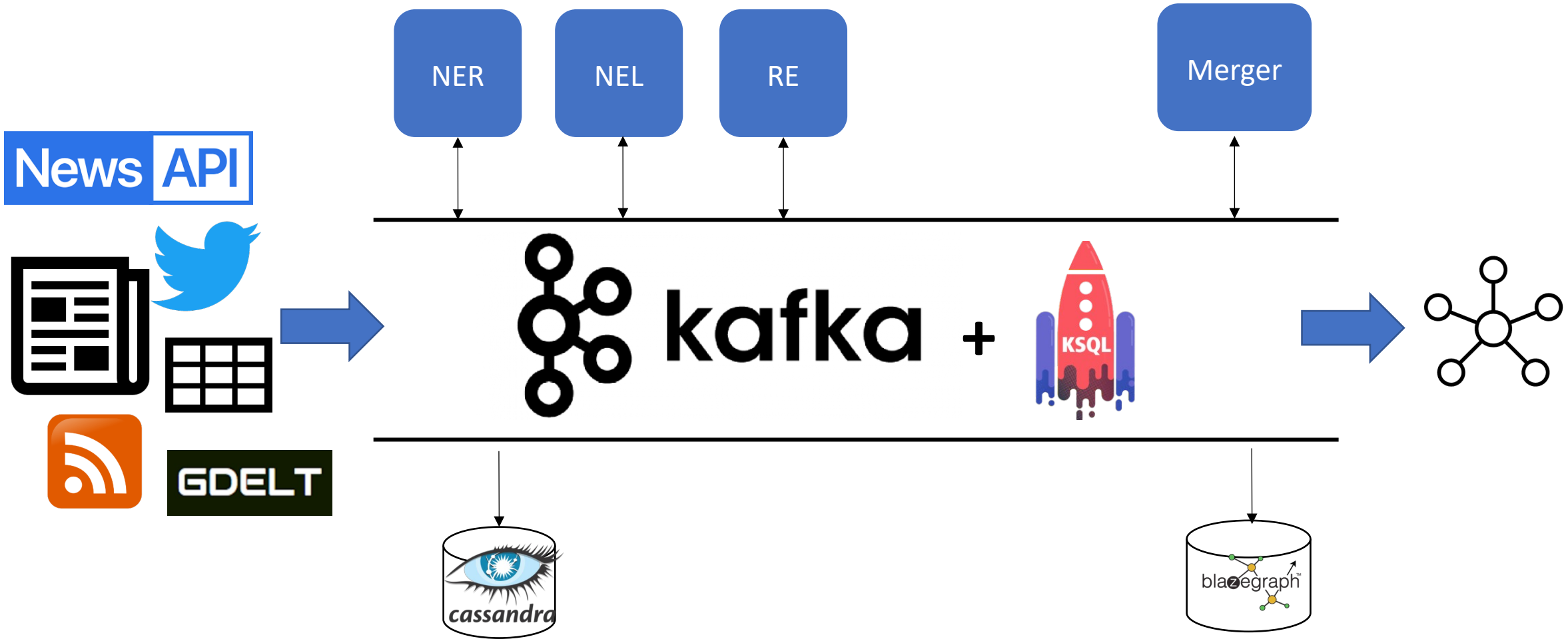


ML/DL services are designed as microservices, hence, they can be easily scaled and reused for processing the different types of data.

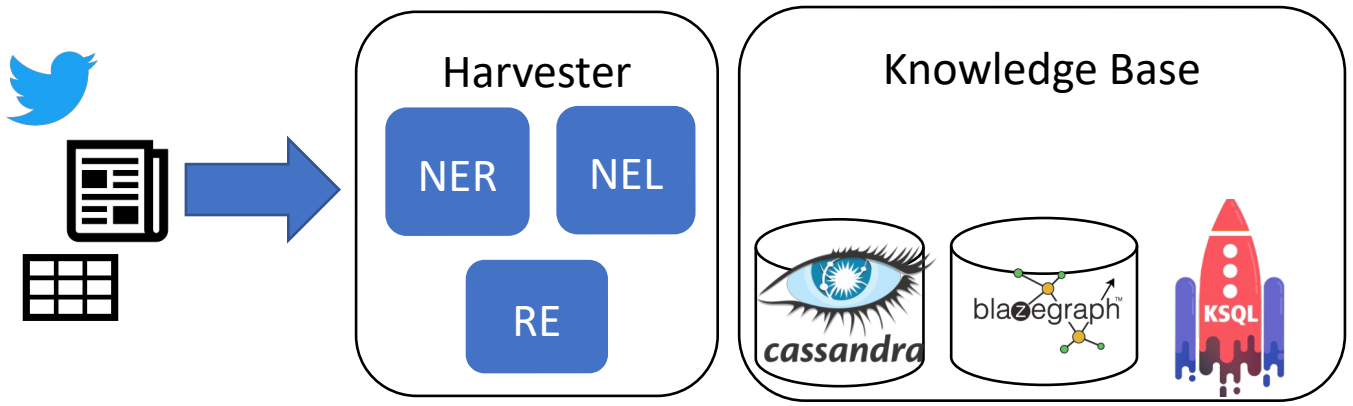
Data storage



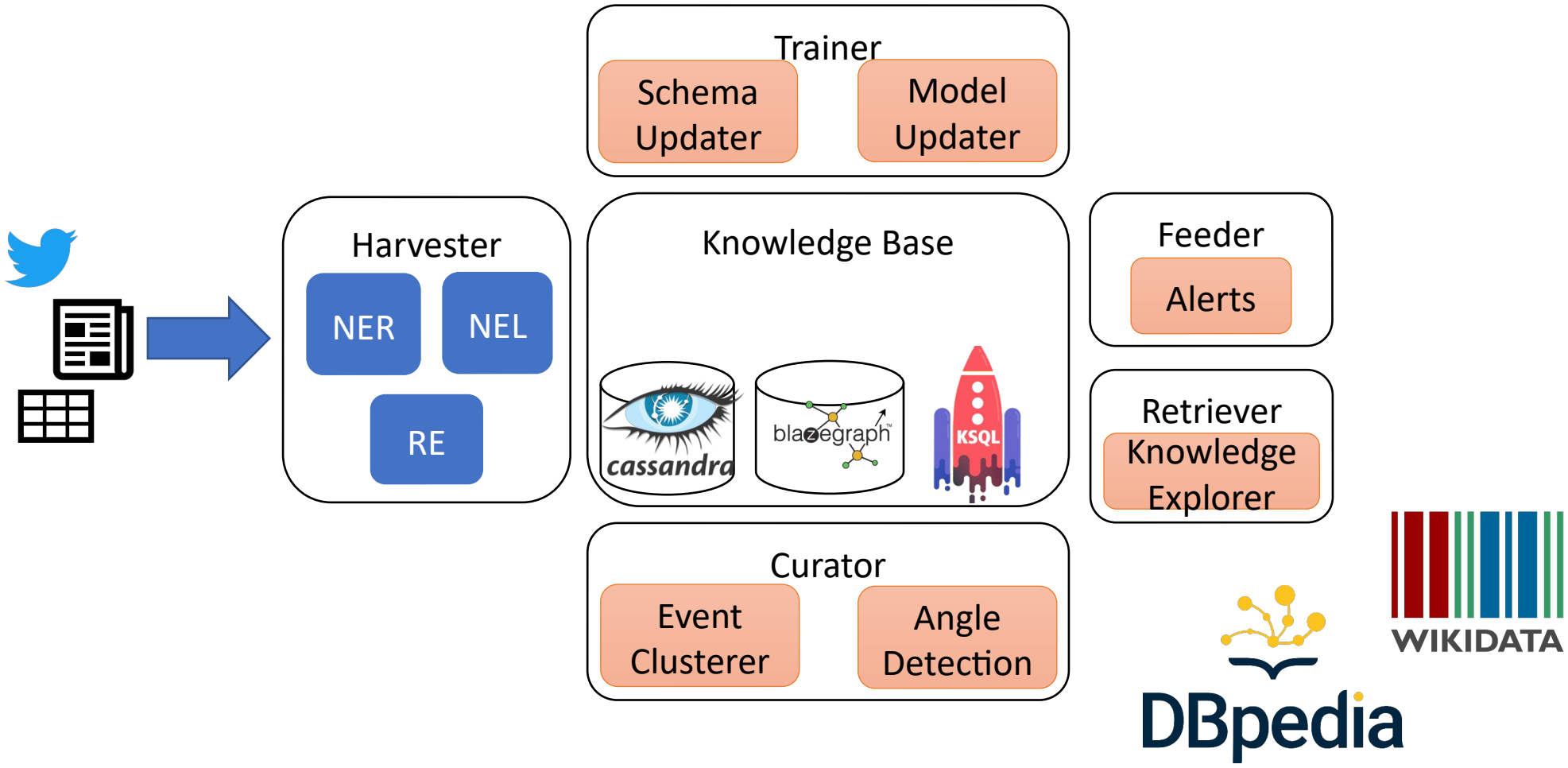
Data integration stack (inspired by Liquid)



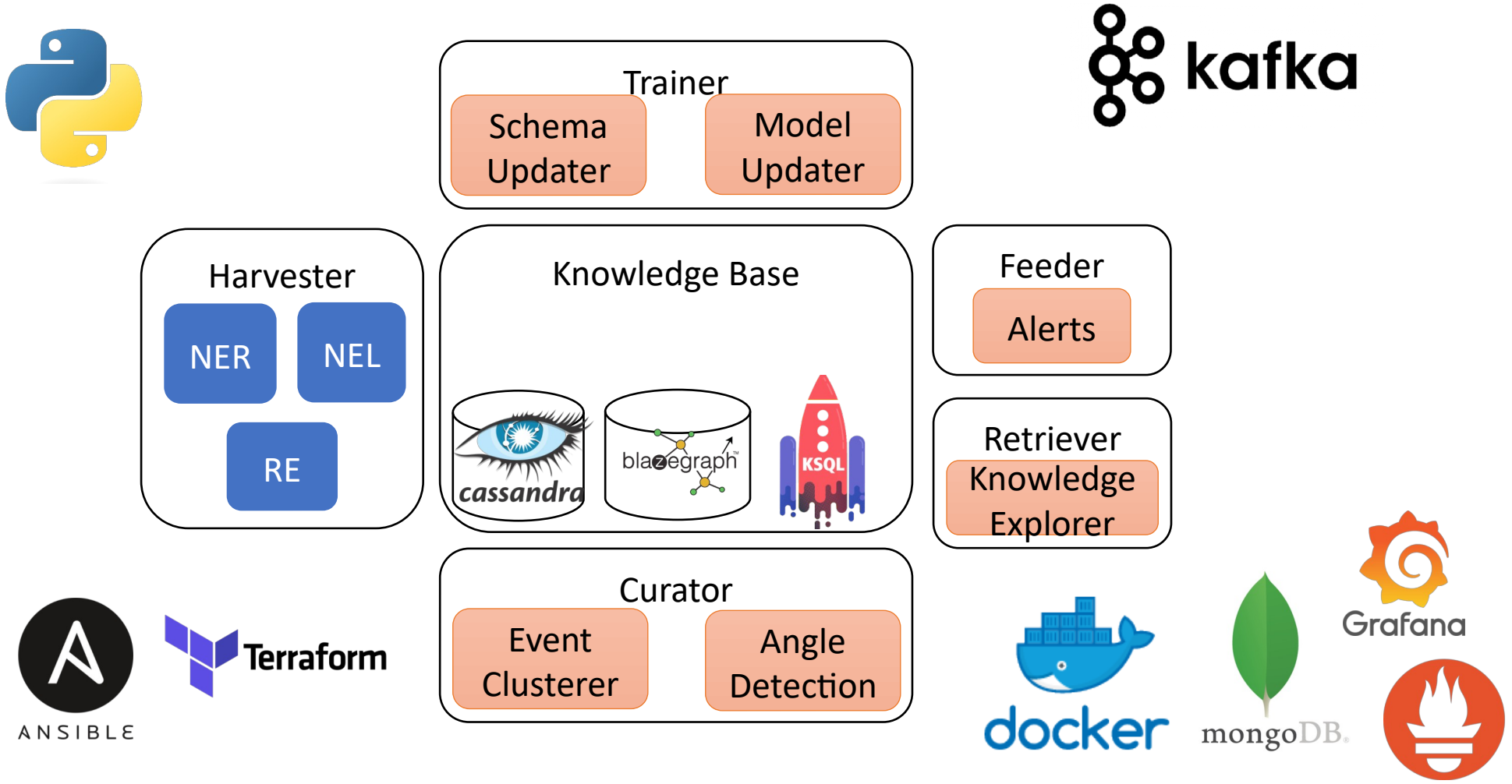
The big picture



The big picture



The big picture



The big picture

- 22 cloud instances (with a total of 55 vCPU, 220GB RAM and 20TB disk)
- 20.000-30.000 daily news + 40.000-50.000 events from GDELT
- Daily ingest of ca. 22M triples.
- The ingestion rate is not homogenous, hence, on the peaks we process 1 item in less than 0.2 seconds.

What could be added?

- Tweets alone do not provide enough information. Perhaps pre-processing them in small clusters.
- Event detection or clustering needs to be implemented as a windowed process.
- Alerts on breaking or hot news can process a chuck of the last 20 minutes of news.