**Benefits of autonomous industry**

Comfort of travel

Process of cars into autonomous vehicle

All decisions were made by driver before autonomation

Using sensors ,lessors and cameras to navigate the road

Sensors-where to drive and how fast to drive

**Driving in the road**

Traffic lights

Pedestrians

This type of architecture is therefore needed when fast, low-latency decisions are required, in particular, in case of autonomous vehicles which should calculate the most optimal route as fast as possible. Amongst the most important stream processing tools there are: Kafka streams, Flink, Storm, or Spark streaming.

All of them could be considered when building architecture for fast route selection for autonomous vehicles, nevertheless in this article, **Kafka** streaming platform has been chosen for 3 reasons:

* Thanks to the fact that different partitions of each topic can reside on different servers it enables horizontal scaling. In practice, it means that additional load of data can easily be accommodated.[13](https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.36#itl236-bib-0013)
* It guarantees at least one delivery of data from producers to consumers.[9](https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.36#itl236-bib-0009)
* Kafka can be a source of data and therefore it can be easily integrated with other streaming platforms such as Storm or Spark (which is often the case).[13](https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.36#itl236-bib-0013)

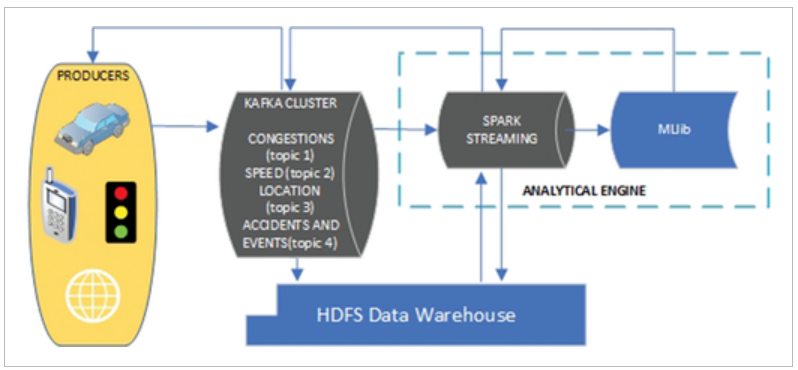
To summarize, both batch and stream processing are important and the most efficient architectures normally mix them. That is also something that should be implemented in case of autonomous vehicles. The first purpose is of course a fast, low-latency analysis based on the data gathered in real time. Nonetheless, we should not give up on storing historical data for any further, predictive analyses and forecasting that can be accomplished through batch processing.

We need to identify

Speed,location,congestions,accidents,

In this topic our producers are all types of vehicles

According to incoming data separate topic has to be created



What about real time?



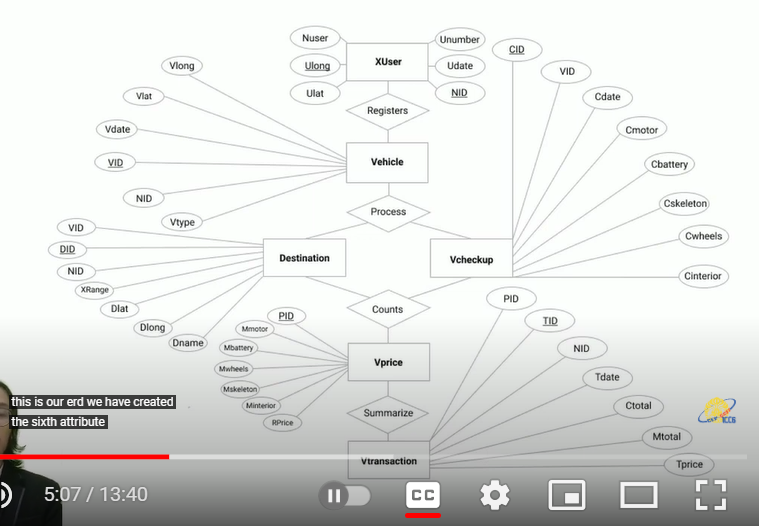
AVERSOT

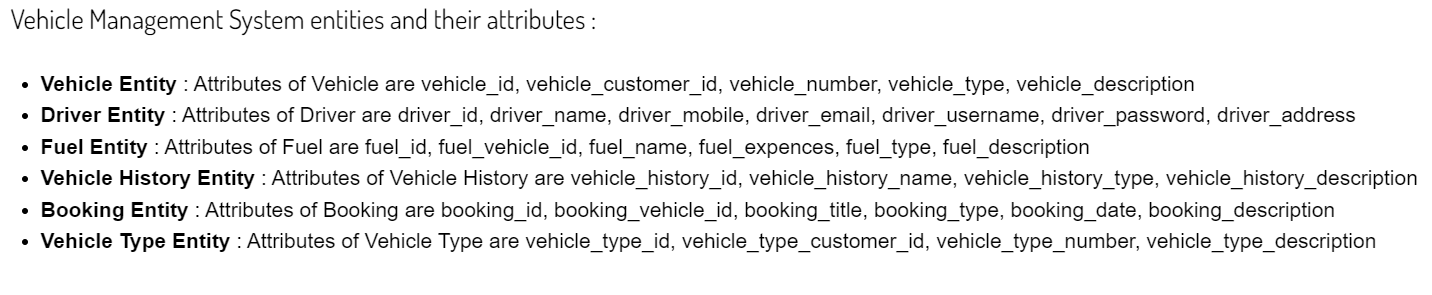
Backend system of autonomous vehicles in online transportation,handling the autonomous system in an autonomous vehicle system based on SQL database.

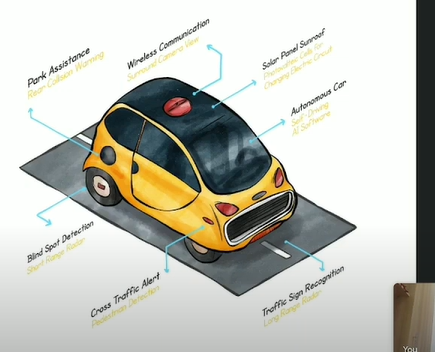
I’s based on SQL database

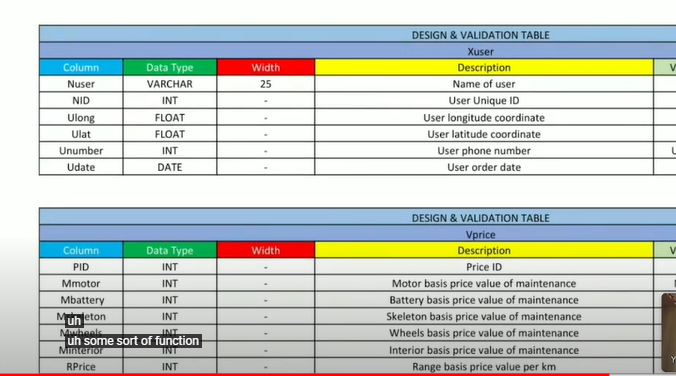
Autonomous vehicle works without driver or driverless vehicle.

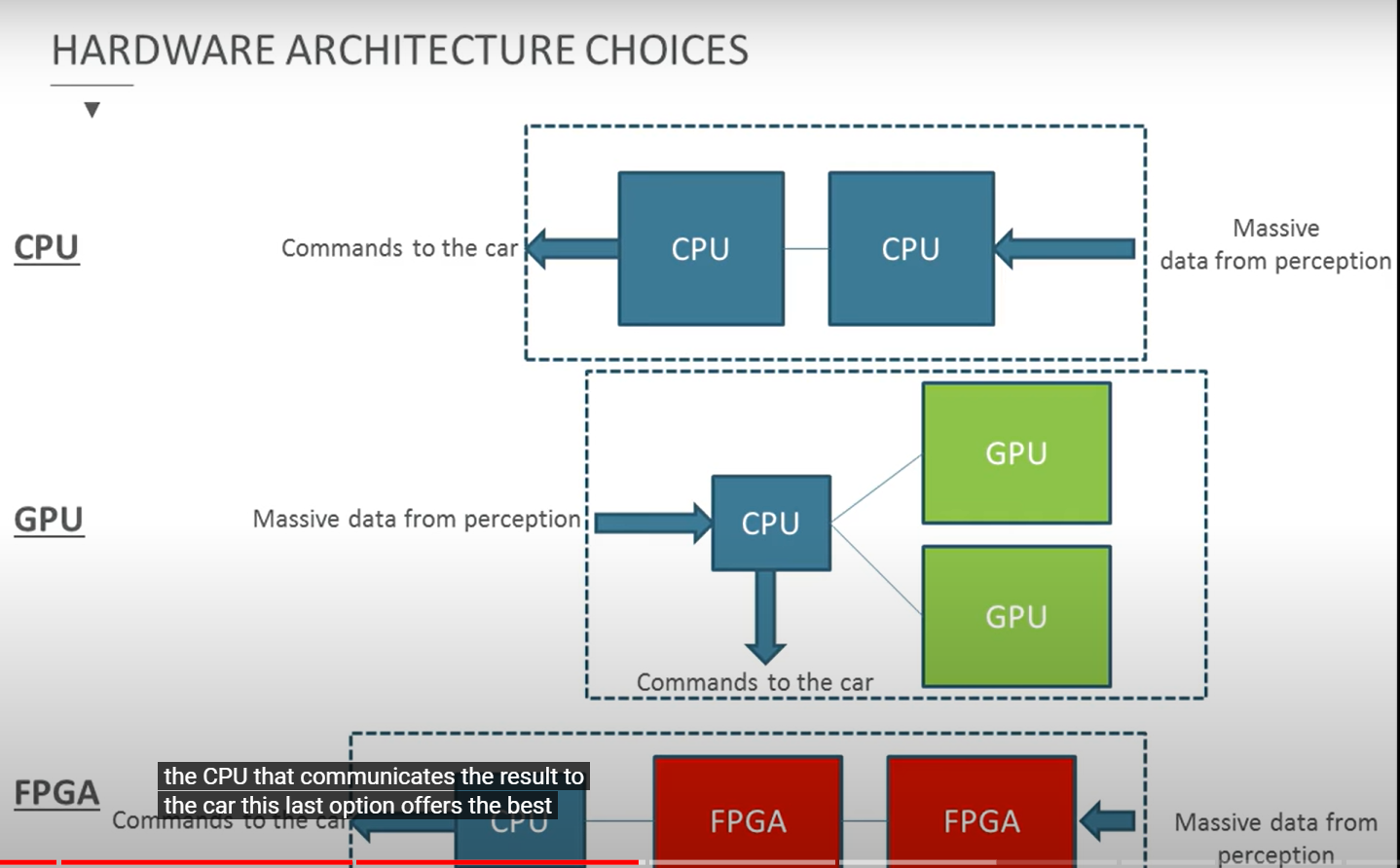
This system is handling data management,processing and storage of information.

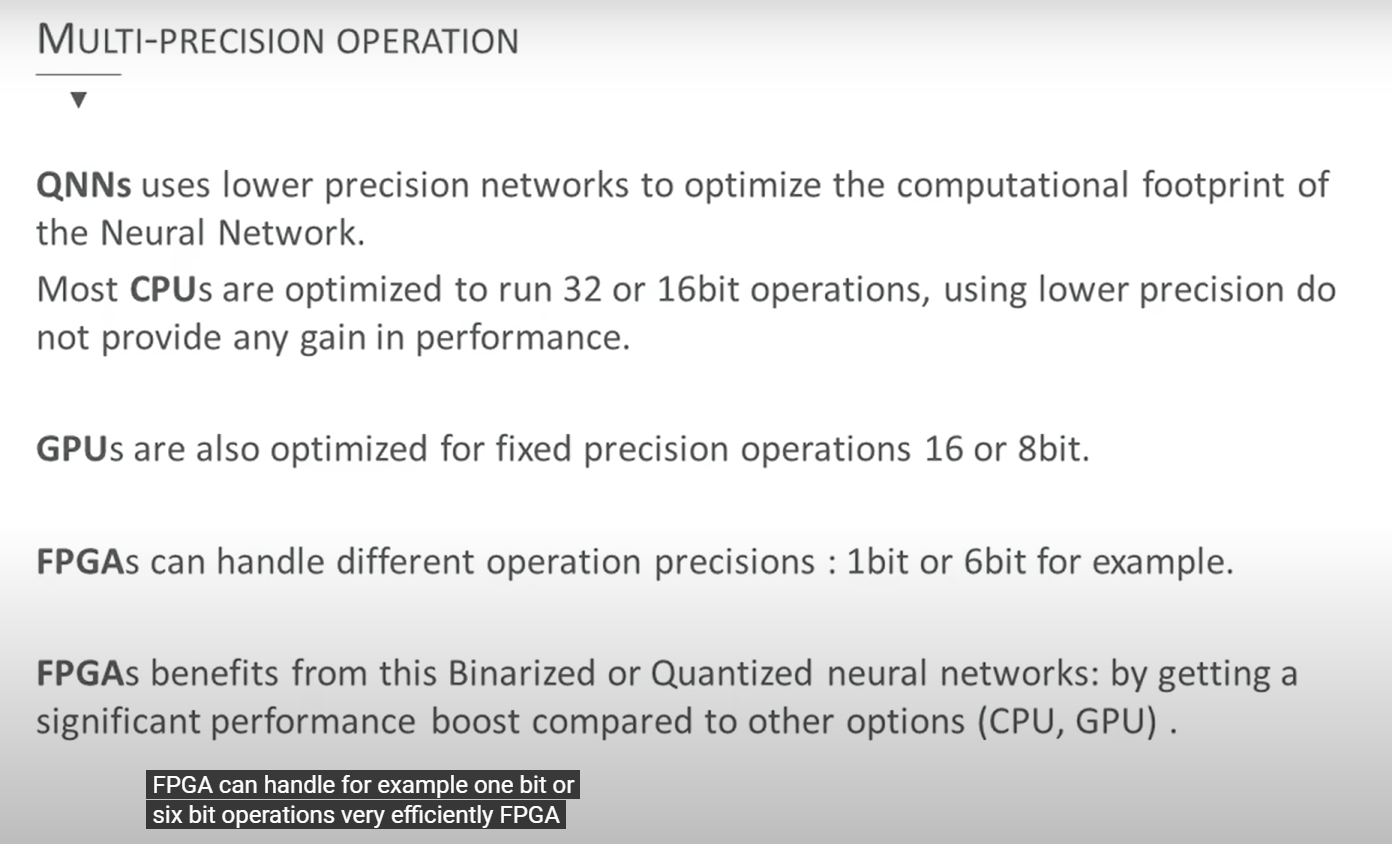


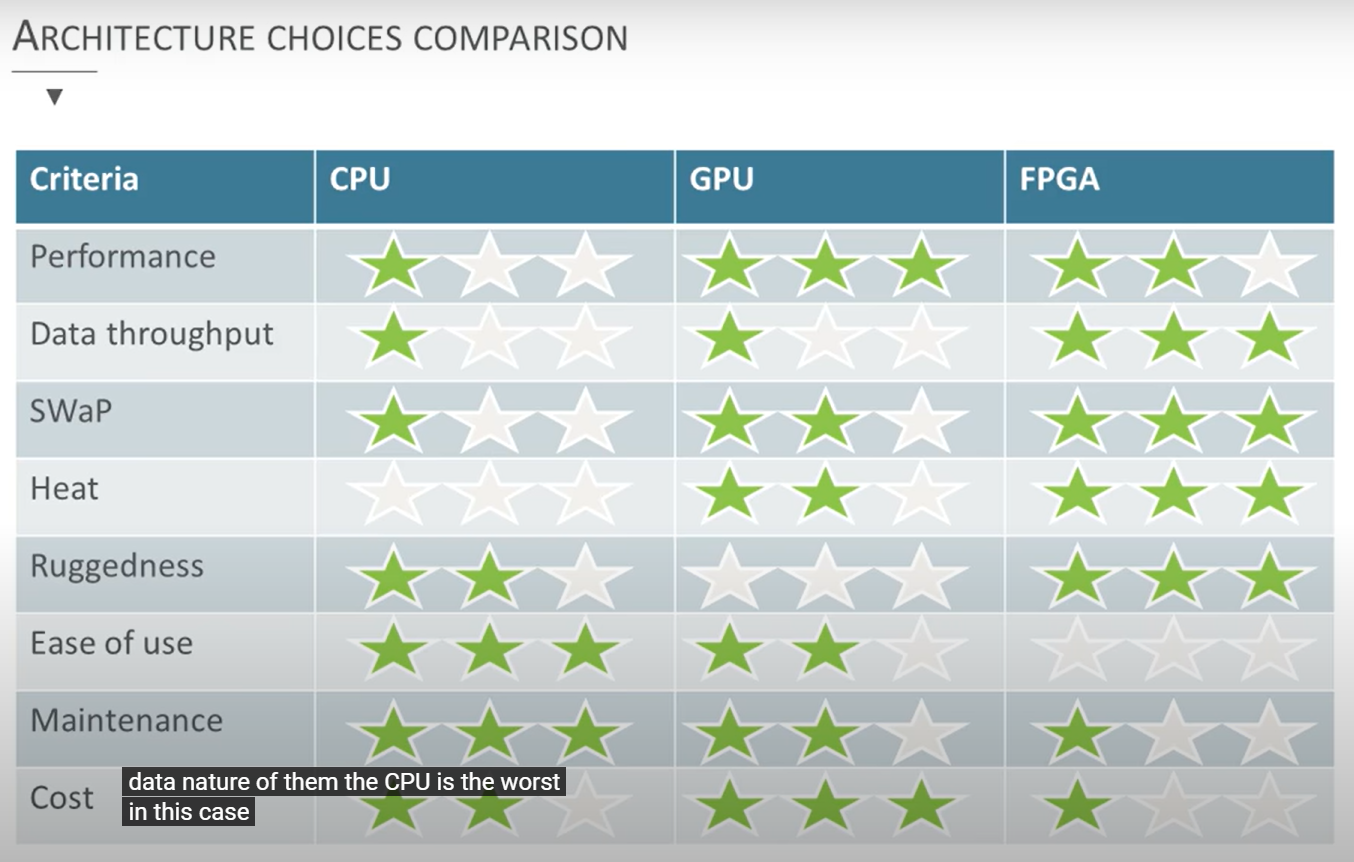












**Why big data for autonomous cars**

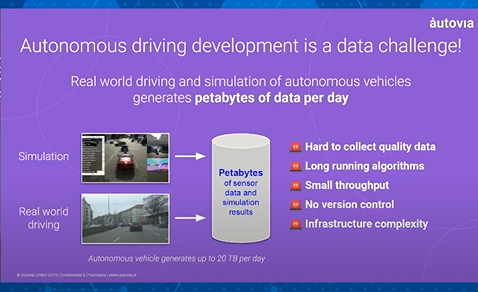
In this cloud, big data concept is used to handle large collection of datasets that cannot handle by conventional systems. These systems can find patterns, associations, trends, etc. Autonomous vehicles require access to vast amount of data. For example, sensors network data, maps, images, videos, weather forecasts, programs, algorithms, etc.

Solution for Storage of Data

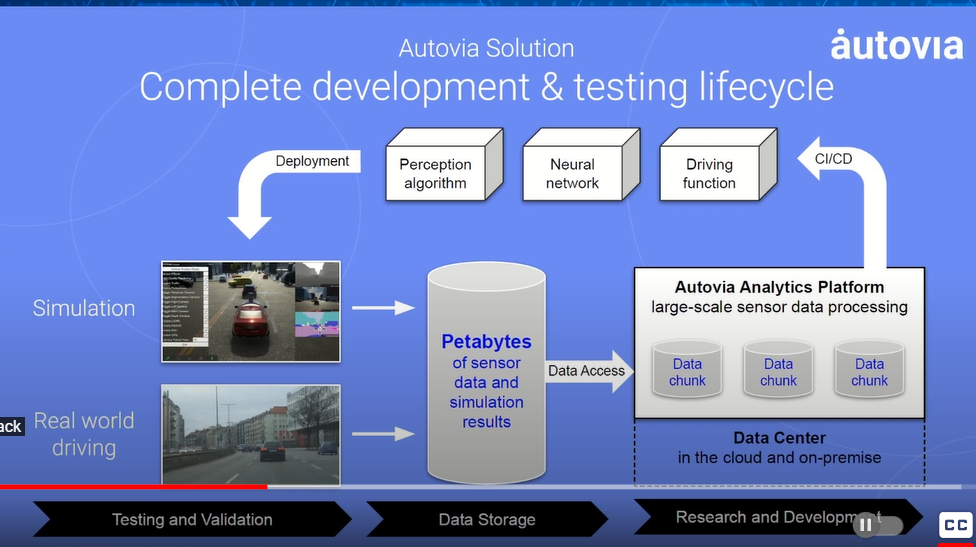
It has been observed that googles self-driving car gathers 1 gigabyte of sensor data per second. This is a big challenge for today's scientists. It is because sensors are capturing every single thing that sees on the road like cars, buses, birds, rolling balls etc. and combine all that data to make decisions while driving. Therefore, it is a big problem or say challenging for the IT industry to deal with this huge amount of data. The solution for this problem is a Technology that is provided by Apache Foundation Software. It is a big data (Hadoop) platform that is specially designed to handle massive amount of data that cannot be handle by traditional system. The architecture of Hadoop provides HDFS (Hadoop distributed file system) that is used for storage purpose.

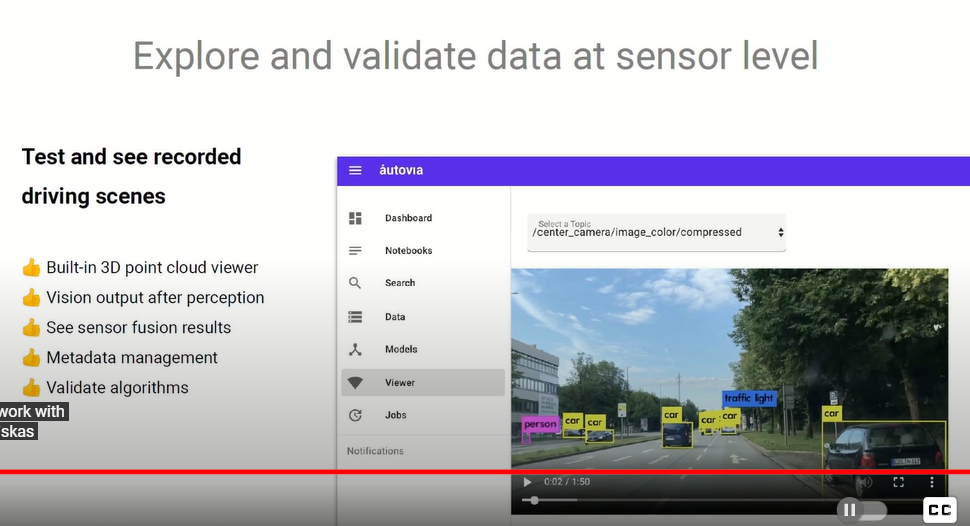
Autonomous car normally generates up to 20 TB per day

Real world driving and simulation of autonomous vehicles generates petabytes of data per day.



Normally an autonomous vehicle generates more than 20 TB per day. When it comes to the process of an autonomous car it can be categorized in to three steps as testing and validation,





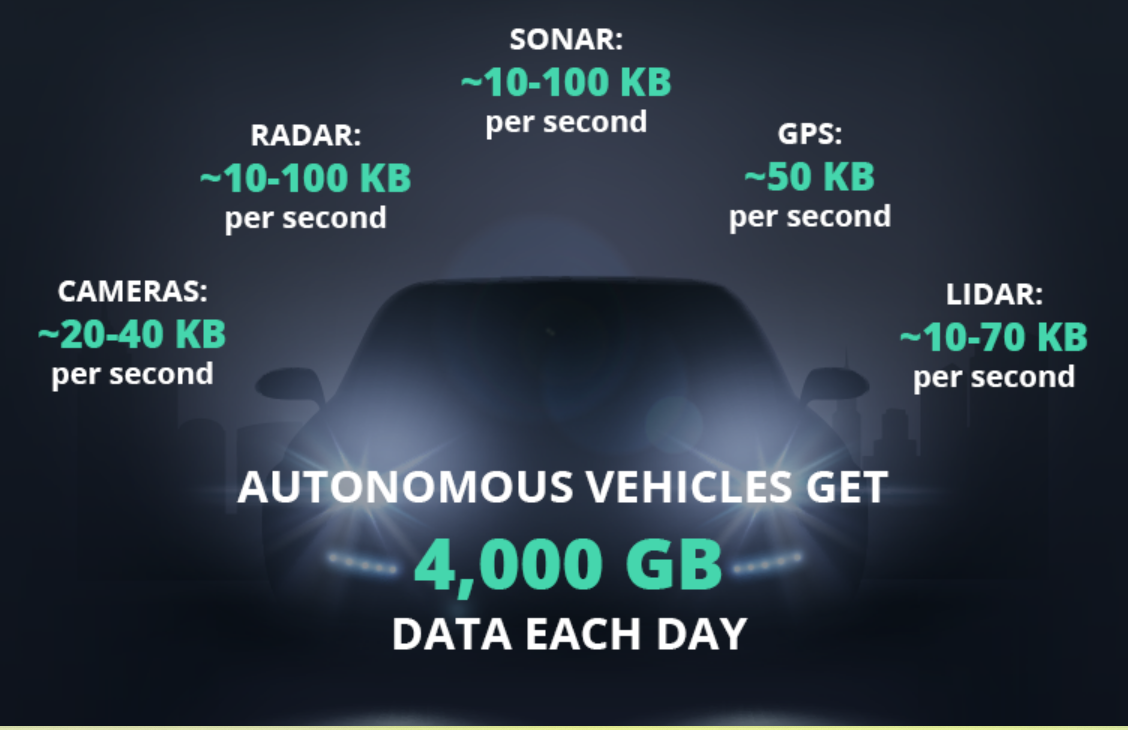
# Using Polygot Persistence with NoSQL Databases for Streaming Multimedia, Sensor, and Messaging Services in Autonomous Vehicles**2020-01-0942**

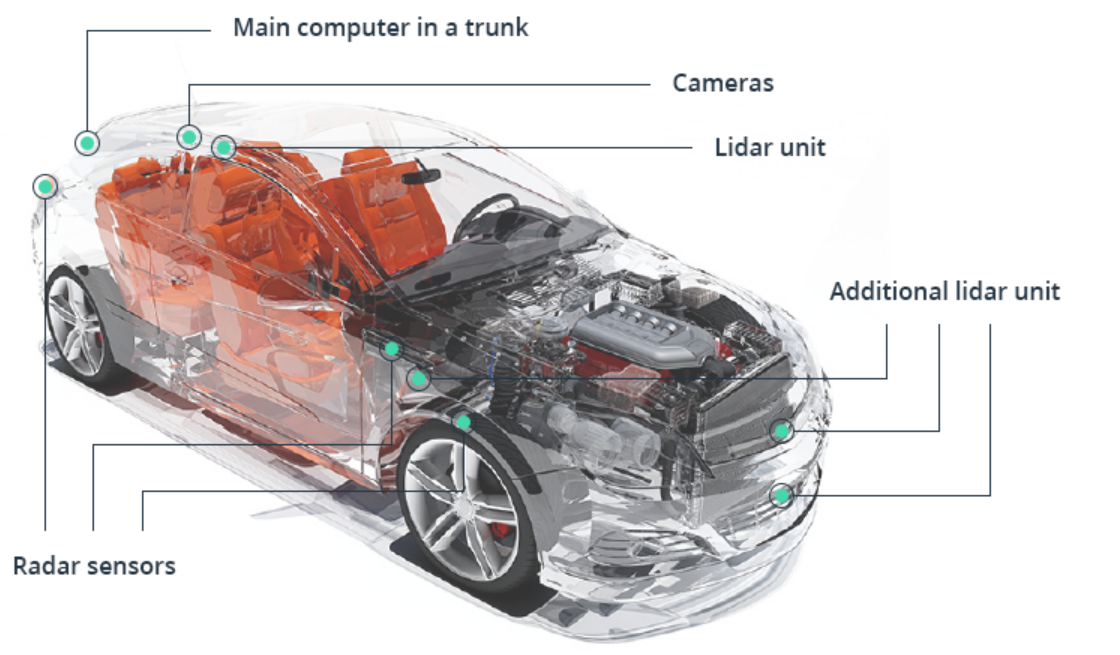
The explosion of big data has created challenges for both cloud-based *systems* and *Autonomous Vehicles (AVs) in data collection and management*. The same challenges are now being realized in developing databases for integrated sensors, streaming, real-time and on-demand services in AVs. With just one AV expecting to generate over 30 Terabytes of data a day, modern NoSQL databases provide opportunities to horizontally scale AV data seamlessly. NoSQL provides solutions designed to accommodate a wide variety of data models such as, key-value, document, column and graph databases. Key-value stores are by nature scalable, fast processing, and distribute horizontally. These databases are tasked with handling several data types including IoT, radar, lidar, ultra-sonic sensors, GPS, odometry, and sensor data while providing streaming and real-time services. NoSQL can store and utilize structured, semi-structured, and unstructured data necessary for multimedia storage needs. NoSQL databases such as Graph databases support big data necessary for the demands of modern software development. Graph databases can scale AV data by using geospatial and geolocation coordinates as entities for flexible queries and pattern recognition. This paper addresses the development of an autonomous platform to process structured, unstructured, semi-structured, and polymorphic data using NoSQL databases built on a hybrid framework. Using Polygot Persistence for processing multimedia, social media, GPS data, audio, fleet diagnostics, and messaging services will be incorporated into the Platform as a Services (PaaS). Integration of NoSQL’s toolboxes and horizontal scalability through cloud deployments will continue the development of an integrated database that’s scalable for the PaaS application.

* Apache Hadoop is based mainly on HDFS and map-reduce concepts.[**11**](https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.36#itl236-bib-0011) HDFS (Hadoop distributed file system) is responsible for storing the data and map-reduce paradigm is a programming model aiming to distribute large amounts of data across many servers and consists of “map” and “reduce” functions.
* On the other hand, there are the tools that apply so called stream processing. Stream processing, as opposed to batch processing enables to gather and process the data dynamically as it appears.[**10**](https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.36#itl236-bib-0010) This type of architecture is therefore needed when fast, low-latency decisions are required, in particular, in case of autonomous vehicles which should calculate the most optimal route as fast as possible. Amongst the most important stream processing tools there are: Kafka streams, Flink, Storm, or Spark streaming.
* The results do not leave any illusions: the first iteration works faster in Hadoop (127 vs 174 seconds) but then all the subsequent iterations are performed much faster in Spark as it reuses cached data and each iteration takes only 6 seconds which allows to run a job approximately 10 times faster. Obviously, no system is fully reliable and as safety in the context of autonomous vehicles plays an unimaginably important role the system should be extra protected.

**Weaknesses**

The most significant weak points of transportation in present days that can be eliminated by autonomous vehicles are traffic congestions. Traffic congestions occur when too many infrastructure users want to use the same road at the same time.[**15**](https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.36#itl236-bib-0015) It seems obvious that the solution of this problem would make self-driving vehicles have an absolute knowledge about traffic concentration on every possible route.



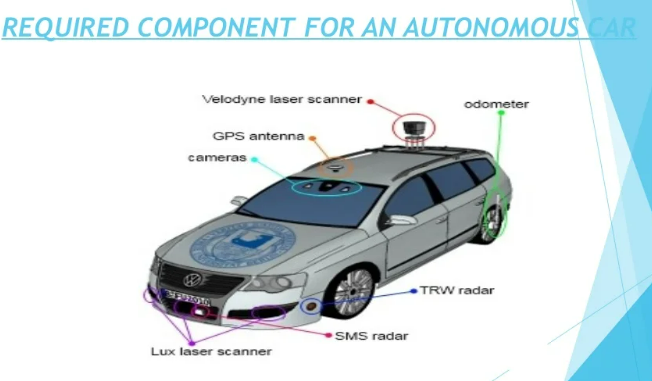


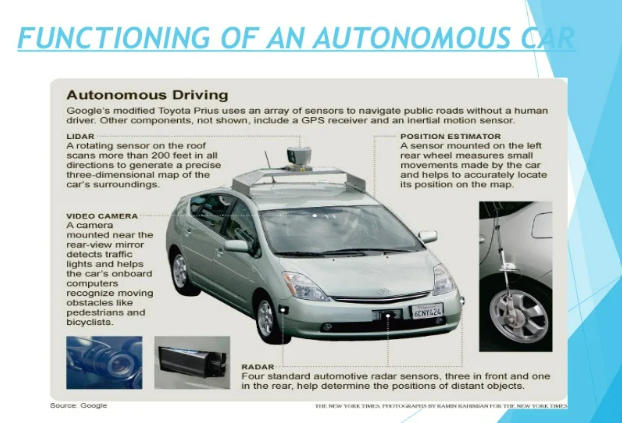
Importance of collected data

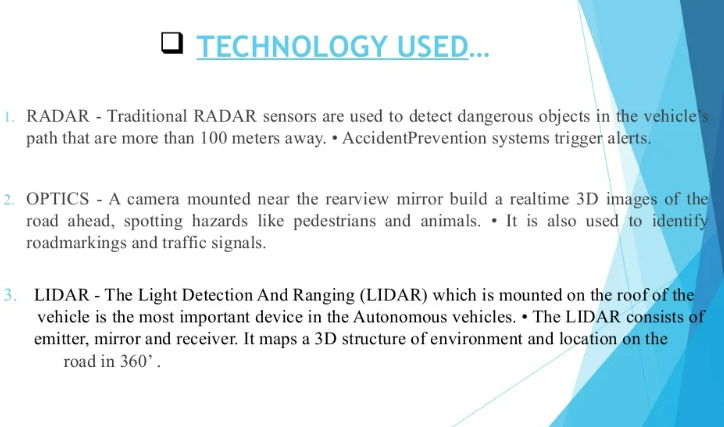
Using collected data, a driverless car can build strategies for many possible situations on the road. Data sharing between autonomous vehicles will aid in avoiding traffic jams, taking into account weather conditions and reacting to emergencies.

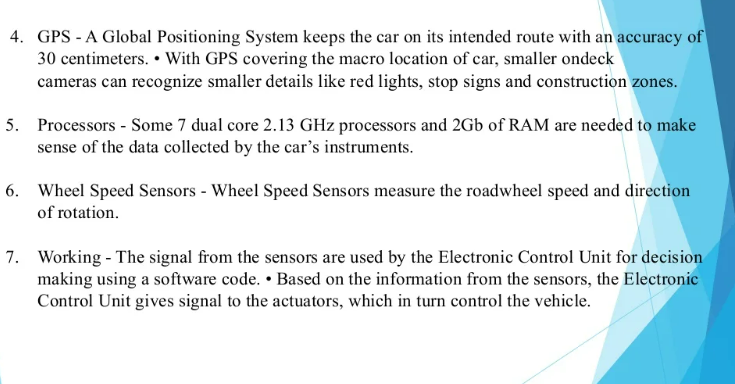
## What are the use cases of big data autonomous vehicles?

* An automaker can remotely see a problem with your car and immediately inform you through your vehicle.
* Big data in connected cars can help to develop custom insurance plans for drivers based on their performance.
* City planning and engineering will become more accurate with big data from connected vehicles: more efficient road planning, early warning systems on dangerous spots, and safer pedestrian walkways are just a **few possible outcomes.**









* **Intel estimates that autonomous vehicles may generate 4 terabytes of data each day. These vehicles will collect and share important data, including GPS, video, sensor movements, and more. The vehicles rely on this data to provide automatic braking, collision protection, emergency assistance, and other safety features.**

References

<https://ieeexplore.ieee.org/abstract/document/8473347>

<https://www.sae.org/publications/technical-papers/content/2020-01-0942/>

<https://intellias.com/how-big-data-in-autonomous-vehicles-defines-the-future/>

<https://www.scribd.com/archive/plans?slideshare=true>

<https://www.udacity.com/blog/2021/03/how-self-driving-cars-work-sensor-systems.html>

- sonar

<https://towardsdatascience.com/autonomous-driving-dataset-visualization-with-python-and-vizviewer-24ce3d3d11a0#:~:text=Within%20the%20context%20of%20autonomous%20driving%2C%20there%20are,current%20road%2C%20traffic%20signs%20and%20traffic%20lights%2C%20etc>. – data structure/types,…

<https://insidehpc.com/2022/02/edge-nvme-storage-for-autonomous-vehicles/#:~:text=Effective%20storage%20within%20autonomous%20vehicles%20must%20meet%20three,design%20to%20operate%20in%20any%20autonomous%20vehicle%20conditions>. -data storage

<https://www.unleashedsoftware.com/blog/5-major-consequences-poor-quality-data-how-avoid>

-poor quality data

<https://becominghuman.ai/high-quality-training-data-for-autonomous-cars-22d542d62cbf>

-high quality data requirement

<https://report.asam.net/test-map-report-1>

-Testing AV

<https://www.infoq.com/articles/validation-autonomous-systems/>

-validation

<https://pelotondb.io/>

-relational data base management system for AV

Comparison between SQL and NO SQL

No SQL Databases have high performance compared with relational databases. Poor reliability and consistency can be taken as the disadvantages of using No SQL while SQL has more reliability and consistency. Both SQL and No SQL databases have high availability, scalability but it’s more expensive when increasing scalability for SQL.

Before discussing the

The solution makes use of spatial big data and combined Kafka and Spark architectures that would enable gathering unstructured data from different sources and processing them in real time.

## BATCH AND STREAM PROCESSING IN BIG DATA

Batch processing-volume

The first and the most important distinction when considering big data architecture in general shall be the distinction between batch and stream analytics.[**9**](https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.36?casa_token=HPZO2S_37VkAAAAA%3AIJ1Mxjm6BqGBhIcMsopi9-Zi9W0fARVck7Vt9BlUDzG82iO0emi1KJ4zXwFK39ti9t1JZNgLpt2qdA#itl236-bib-0009) Batch processing is used for the analysis of large amount of static data that is being collected over time.[**10**](https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.36?casa_token=HPZO2S_37VkAAAAA%3AIJ1Mxjm6BqGBhIcMsopi9-Zi9W0fARVck7Vt9BlUDzG82iO0emi1KJ4zXwFK39ti9t1JZNgLpt2qdA#itl236-bib-0010) It can be identified mostly with the Hadoop ecosystem and NoSQL databases (eg, MongoDB or Cassandra). Apache Hadoop is based mainly on HDFS and map-reduce concepts.[**11**](https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.36?casa_token=HPZO2S_37VkAAAAA%3AIJ1Mxjm6BqGBhIcMsopi9-Zi9W0fARVck7Vt9BlUDzG82iO0emi1KJ4zXwFK39ti9t1JZNgLpt2qdA#itl236-bib-0011) HDFS (Hadoop distributed file system) is responsible for storing the data and map-reduce paradigm is a programming model aiming to distribute large amounts of data across many servers and consists of “map” and “reduce” functions. Together with Hadoop ecosystem NoSQL (or not only SQL), databases constitute batch processing. Not only NoSQL stores chosen pieces of data (eg, user transaction data or customer profile data) but also it provides an end user with interactivity for example, in web applications. It can read and write certain segments of data immediately.

How to handle various types of data sets-variability

**Apache Spark: It** is an open source framework for performing cluster computing. This framework is used for the processing of large scale data analytics applications. It can process various type of data such as data from Hadoop Distributed File System (HDFS), relational datasets like from Apache Hive and NO-SQL database. Apache Spark framework consists of four libraries that are listed below [15]:

No sql importance

NoSQL (Not Only SQL) is data storage technique where Big heterogeneous data is saved by organizations. Researchers have proposed query languages for different types of NoSQL data storage. The limitations of existing solutions is that no standard query language is proposed and different query syntax are available for different data storage. If user asks in the form of natural language query, QA system should map this to respective NoSQL data store using Artificial Intelligenc

Chart, bubble chart

Description automatically generated

### **3.1. What’s wrong with traditional RDBMSs?**

Traditional relational database management systems (RDMS) are simply not capable of handling big data. The data is too big, too fast, and too diverse to store and manipulate. Relational databases require a schema before writing to the database, which is too rigid to manipulate volumes of real-time data with diverse [data structures](https://www.sciencedirect.com/topics/computer-science/data-structure). The ACID properties (atomicity, consistency, isolation, and durability) are too strict for some applications. The RDBMS clusters are SPOF (Single Point of Failure) and expensive, and have [impedance mismatch](https://www.sciencedirect.com/topics/computer-science/impedance-mismatch) (aggregate versus atomic data). This all led to requirements for new architectures and new transaction management such as BASE (**B**asically **A**vailable, **S**oft state, **E**ventual consistency), which relaxes the ACID properties in distributed [data management systems](https://www.sciencedirect.com/topics/computer-science/data-management-system). BASE is common in NoSQL systems.

Graphical user interface, text, application, email

Description automatically generated

## NoSQL databases

NoSQL refers to open-sourced, distributed, and non-relational databases. Horizontal-scaling enables scalability and fault tolerance. It is schemaless, allowing new data type to be dynamically added to the database, increasing write-performance. Most NoSQL systems adopt BASE (Basically **A**vailable, **S**oft state, **E**ventual consistency), as opposed to ACID (properties, atomicity, consistency, isolation, and durability), for transaction management to increase availability and performance rather than strict consistency.

NoSQL was specifically designed to address the needs of big data, big users, and cloud computing. It supports unstructured or non-relational data types (nested structure, column families, document, JSON (JavaScript Object Notation), BSON (binary serialization of JSON), and graph). NoSQL exhibits several other important characteristics. It is schema-less (schema-on-read, implicit schema). It can scale out massively at low cost and with fast retrieval (elastic scaling), providing low cost operational management for a large number of users. It supports scalability, high performance, and fault-tolerance, and is designed for real-time, non-uniform big data.

Activities important for dealing with big data issues include: scalability, schema flexibility, ease of development, cost, and availability of [deployment options](https://www.sciencedirect.com/topics/computer-science/deployment-option) [[9]](https://www.sciencedirect.com/science/article/pii/S0169023X17300277?casa_token=JbN39aXCNmwAAAAA:kbnmEIN4YrBOf8q3BG-k1m71GZG1Kvf-7pn-h26dlqI2K6oMmG9uIt1CrUMaKchNuI23fb9P" \l "bib9). The shift from relational databases to NoSQL Databases is spurred as well by the need for flexibility both in the scaling model and the data model. In terms of scaling, in relational databases, scaling up is accomplished by adding a bigger server when additional capacity is needed. In NoSQL, scale out means that, instead of acquiring a bigger server, one can add more commodity servers.

**Useful Link**

[**https://www.sciencedirect.com/science/article/pii/S0169023X17300277?casa\_token=JbN39aXCNmwAAAAA:kbnmEIN4YrBOf8q3BG-k1m71GZG1Kvf-7pn-h26dlqI2K6oMmG9uIt1CrUMaKchNuI23fb9P**](https://www.sciencedirect.com/science/article/pii/S0169023X17300277?casa_token=JbN39aXCNmwAAAAA:kbnmEIN4YrBOf8q3BG-k1m71GZG1Kvf-7pn-h26dlqI2K6oMmG9uIt1CrUMaKchNuI23fb9P)

[**https://www.sae.org/publications/technical-papers/content/2020-01-0942/**](https://www.sae.org/publications/technical-papers/content/2020-01-0942/)

[**https://www.datavail.com/blog/how-to-manage-a-very-large-database-in-sql-server/#:~:text=The%20most%20recommended%20and%20best,the%20source%20of%20the%20standby**](https://www.datavail.com/blog/how-to-manage-a-very-large-database-in-sql-server/#:~:text=The%20most%20recommended%20and%20best,the%20source%20of%20the%20standby)**. -SQL**

[**https://www.geeksforgeeks.org/how-nosql-system-handle-big-data-problem/**](https://www.geeksforgeeks.org/how-nosql-system-handle-big-data-problem/)

[**https://ieeexplore.ieee.org/abstract/document/8457745?casa\_token=HVh49NpbSk8AAAAA:RXvopGncFt51kuw2EGunylDIxfWMzuEOKiiRhDiybIHS4lb6BPdAjciJaXGbpj\_otijEYmxL**](https://ieeexplore.ieee.org/abstract/document/8457745?casa_token=HVh49NpbSk8AAAAA:RXvopGncFt51kuw2EGunylDIxfWMzuEOKiiRhDiybIHS4lb6BPdAjciJaXGbpj_otijEYmxL) **-nosql security**

[**https://ieeexplore.ieee.org/abstract/document/5600026?casa\_token=C4oWARqnsPkAAAAA:0F\_1YgQ1VLsz21W5Na-zUDUK5Dx3Q7\_LWEf9IVRzZHszRFZ-f6sl19njVxylEfG7Gm5PnGmX**](https://ieeexplore.ieee.org/abstract/document/5600026?casa_token=C4oWARqnsPkAAAAA:0F_1YgQ1VLsz21W5Na-zUDUK5Dx3Q7_LWEf9IVRzZHszRFZ-f6sl19njVxylEfG7Gm5PnGmX) **-sql attacks**

* **volume**: horizontal scalability, elastic or not, is an issue for DBMSs; the advocated vertical scalability is too expensive
* **velocity**: consequent to being outbound [[9](https://link.springer.com/chapter/10.1007/978-3-319-94845-4_13#ref-CR9)] systems, SQL databases do not perform well on streaming data for real-time analytics
* **variety**: heterogeneous data are anathema to SQL databases that deal well with structured data (e.g. number, boolean, varchar), partially well with semi-structured data (e.g. XML, JSON) but are inadequate with unstructured data (e.g. tweets, text, video, audio)

**NoSQL Data Storage**

In relational database, data can be stored in rows and columns format. Furthermore, entity integrity, referential integrity, joins, keys and normalizations are the essential features of SQL based databases. The difference in SQL database and NoSQL database is in supported data models and query languages [12]. In column-oriented data storage, data is stored column-based instead of row-based. Table’s column reading is faster [13]. The main advantage is that dynamic schema is used in this data storage. There are a lot of sparse entries in row-oriented data stores as null value signifies the values that are unknown. In this data store, unknown values are don’t included in flexible schema. HBase [14], Google BigTable [15] and Cassandra are examples of column-oriented database.In document-based data store, record is stored in document.

**A picture containing logo

Description automatically generated  
Chart, bubble chart

Description automatically generated**

Data Quality –

For data storage it is handled for Big Data quality through data distribution and replication. For example, storage using Hadoop ecosystem relies on several nodes that duplicate data to avoid any catastrophic data loss and ensure continuity when failure happens. Moreover, data storage quality relies on the storage medium I/O performances for several types of Big Data. Reading and writing ratios must follow the I/O requirements for each data type, e.g., HD Video data, and stream processing.

Many data pre-processing tools has emerged, in addition others existing tools used in the database domain have been updated to handle Big Data. Mostly, Hadoop and spark based preprocessing tools and framework are widely used (e.g. Talend Open Studio for Big Data [36], and Open Refine