# Azure Big Data

## **Exercises**

Research some examples/use cases of big data and think about how you might architect a solution using architectures shown above, or your own solution.

7 Big Data Examples: Applications of Big Data in Real Life

<https://intellipaat.com/blog/7-big-data-examples-application-of-big-data-in-real-life/>

Big Data has totally changed and revolutionized the way businesses and organizations work. In this blog, we will go deep into the major Big Data applications in various sectors and industries and learn how these sectors are being benefitted by these applications.



* Updated on: 13th Jul, 16
* 42779 Views

In this era where every aspect of our day-to-day life is gadget oriented, there is a huge volume of data that has been emanating from various digital sources.

Needless to say, we have faced a lot of challenges in the analysis and study of such a huge volume of data with the traditional data processing tools. To overcome these challenges, some [big data](https://intellipaat.com/blog/big-data-tutorial-for-beginners/) solutions were introduced such as Hadoop. These big data tools really helped realize the applications of big data.

More and more organizations, both big and small, are leveraging from the benefits provided by big data applications. Businesses find that these benefits can help them grow fast. There are lots of opportunity coming in this area, want to become master in Big Data check out this [Big Data Hadoop Training](https://intellipaat.com/big-data-hadoop-training/)?

### **Check out this insightful video on Hadoop Tutorial for Beginners:**

In this blog, we are going to look at some of the famous applications of big data in detail.

So, let’s begin.

## **Big Data in Education Industry**

****

Education industry is flooding with huge amounts of data related to students, faculty, courses, results, and what not. Now, we have realized that proper study and analysis of this data can provide insights which can be used to improve the operational effectiveness and working of educational institutes.

Following are some of the fields in the education industry that have been transformed by big data-motivated changes:

* **Customized and Dynamic Learning Programs**

Customized programs and schemes to benefit individual students can be created using the data collected on the bases of each student’s learning history. This improves the overall student results.

* **Reframing Course Material**

Reframing the course material according to the data that is collected on the basis of what a student learns and to what extent by real-time monitoring of the components of a course is beneficial for the students.

* **Grading Systems**

New advancements in grading systems have been introduced as a result of a proper analysis of student data.

* **Career Prediction**

Appropriate analysis and study of every student’s records will help understand each student’s progress, strengths, weaknesses, interests, and more. It would also help in determining which career would be the most suitable for the student in future.

The applications of big data have provided a solution to one of the biggest pitfalls in the education system, that is, the one-size-fits-all fashion of academic set-up, by contributing in e-learning solutions.

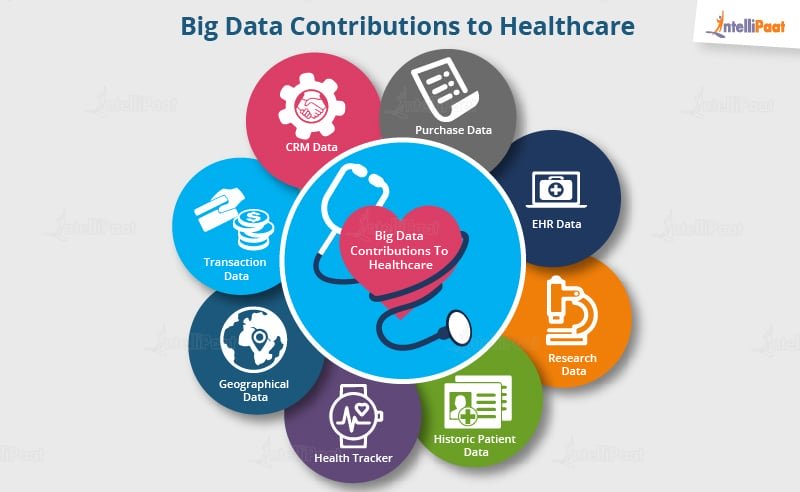
* **Example**

The University of Alabama has more than 38,000 students and an ocean of data. In the past when there were no real solutions to analyze that much of data, some of them seemed useless. Now, administrators are able to use analytics and data visualizations for this data to draw out patterns of students revolutionizing the university’s operations, recruitment, and retention efforts.

***Prepare yourself for the industry by going through this*** [***Top Hadoop Interview Questions And Answers***](https://intellipaat.com/interview-question/big-data-hadoop-interview-questions/)***!***

******

## **Big Data in Healthcare Industry**

****

Healthcare is yet another industry which is bound to generate a huge amount of data. Following are some of the ways in which big data has contributed to healthcare:

* Big data reduces costs of treatment since there is less chances of having to perform unnecessary diagnosis.
* It helps in predicting outbreaks of epidemics and also in deciding what preventive measures could be taken to minimize the effects of the same.
* It helps avoid preventable diseases by detecting them in early stages. It prevents them from getting any worse which in turn makes their treatment easy and effective.
* Patients can be provided with evidence-based medicine which is identified and prescribed after doing research on past medical results.
* **Example**

Wearable devices and sensors have been introduced in the healthcare industry which can provide real-time feed to the electronic health record of a patient. One such technology is from Apple.

Apple has come up with Apple HealthKit, CareKit, and ResearchKit. The main goal is to empower the iPhone users to store and access their real-time health records on their phones.

***Become a master of Hadoop by going through this online*** [***Hadoop training in London***](https://intellipaat.com/big-data-hadoop-training-london/)***!***

## **Big Data in Government Sector**

****

Governments, be it of any country, come face to face with a very huge amount of data on almost daily basis. The reason for this is, they have to keep track of various records and databases regarding their citizens, their growth, energy resources, geographical surveys, and many more. All this data contributes to big data. The proper study and analysis of this data, hence, helps governments in endless ways. Few of them are as follows:

**Welfare Schemes**

* In making faster and informed decisions regarding various political programs
* To identify areas that are in immediate need of attention
* To stay up to date in the field of agriculture by keeping track of all existing land and livestock.
* To overcome national challenges such as unemployment, terrorism, energy resources exploration, and much more.

**Cyber Security**

* Big Data is hugely used for deceit recognition.
* It is also used in catching tax evaders.
* **Example**

Food and Drug Administration (FDA) which runs under the jurisdiction of the Federal Government of USA leverages from the analysis of big data to discover patters and associations in order to identify and examine the expected or unexpected occurrences of food-based infections.

***Go through the*** [***Hadoop Course in New York***](https://intellipaat.com/big-data-hadoop-training-new-york/) ***to get a clear understanding of Big Data Hadoop!***

## **Big Data in Media and Entertainment Industry**

****

With people having access to various digital gadgets, generation of large amount of data is inevitable and this is the main cause of the rise in big data in media and entertainment industry.

Other than this, social media platforms are another way in which huge amount of data is being generated. Although, businesses in the media and entertainment industry have realized the importance of this data, and they have been able to benefit from it for their growth.

**Some of the benefits extracted from big data in the media and entertainment industry are given below:**

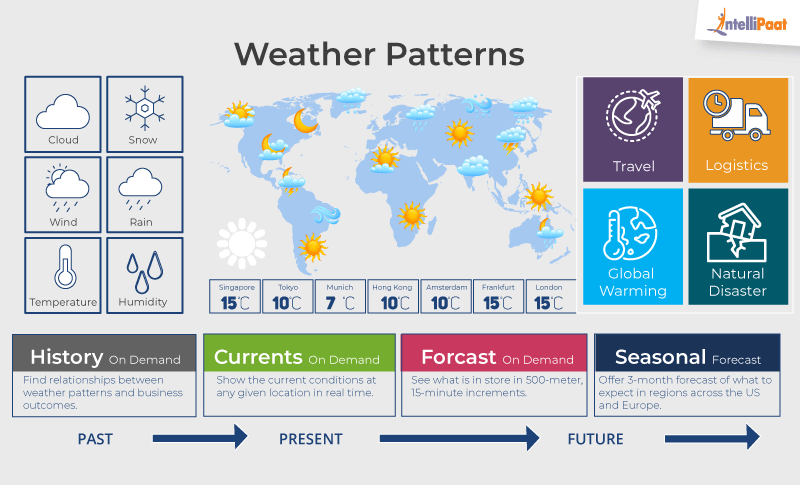
* Predicting the interests of audiences
* Optimized or on-demand scheduling of media streams in digital media distribution platforms
* Getting insights from customer reviews
* Effective targeting of the advertisements
* **Example**

Spotify, an on-demand music providing platform, uses Big Data Analytics, collects data from all its users around the globe, and then uses the analyzed data to give informed music recommendations and suggestions to every individual user.

Amazon Prime that offers, videos, music, and Kindle books in a one-stop shop is also big on using big data.

***Learn more about Hadoop from this*** [***Hadoop Training in Sydney***](https://intellipaat.com/big-data-hadoop-training-sydney/) ***to get ahead in your career!***

## **Big Data in Weather Patterns**

****

There are weather sensors and satellites deployed all around the globe. A huge amount of data is collected from them, and then this data is used to monitor the weather and environmental conditions.

All of the data collected from these sensors and satellites contribute to big data and can be used in different ways such as:

* In weather forecasting
* To study global warming
* In understanding the patterns of natural disasters
* To make necessary preparations in the case of crises
* To predict the availability of usable water around the world
* **Example**

IBM Deep Thunder, which is a research project by IBM, provides weather forecasting through high-performance computing of big data. IBM is also assisting Tokyo with the improved weather forecasting for natural disasters or predicting the probability of damaged power lines.

# The emerging big data architectural pattern

<https://azure.microsoft.com/en-gb/blog/the-emerging-big-data-architectural-pattern/>

Posted on 28 June, 2018

[Pratim Das](https://azure.microsoft.com/en-gb/blog/author/pratim-das/) Head of Solutions Architecture (Data & AI), Customer Success Unit

## **Why lambda?**

Lambda architecture is a popular pattern in building Big Data pipelines. It is designed to handle massive quantities of data by taking advantage of both a [batch](https://en.wikipedia.org/wiki/Batch_processing) layer (also called cold layer) and a [stream-processing](https://en.wikipedia.org/wiki/Stream_processing) layer (also called hot or speed layer).

The following are some of the reasons that have led to the popularity and success of the lambda architecture, particularly in big data processing pipelines.

### **Speed and business challenges**

The ability to process data at high speed in a streaming context is necessary for operational needs, such as transaction processing and real-time reporting. Some examples are fault/fraud detection, connected/smart cars/factory/hospitals/city, sentiment analysis, inventory control, network/security monitoring, and many more.

Typically, batch processing, involving massive amounts of data, and related correlation and aggregation is important for business reporting. This is to understand how the business is performing, what the trends are, and what corrective or additive measure can be executed to improve business or customer experience.

### **Product challenges**

One of the triggers that lead to the very existence of lambda architecture was to make the most of the technology and tool set available. Existing batch processing systems, such as data warehouse, data lake, Spark/Hadoop, and more, could deal with petabyte scale data operations easily but couldn’t do it fast enough that was warranted by the operational needs.

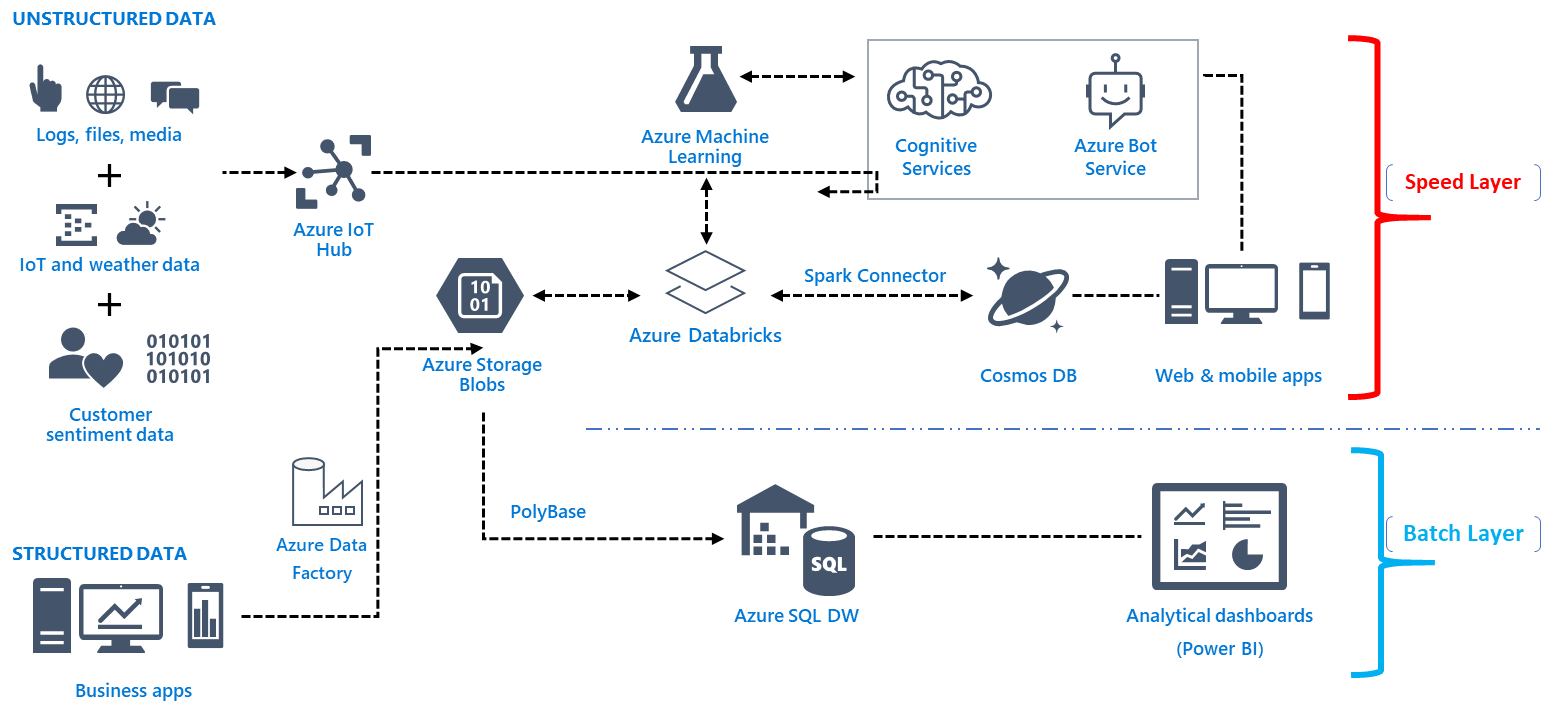
Similarly, very fast layers such as cache databases, NoSQL, streaming technology allows fast operational analytics on smaller data sets but cannot do massive scale correlation and aggregation and other analytics operations (such as [Online Analytical Processing](https://en.wikipedia.org/wiki/Online_analytical_processing)) like a batch system can.

### **The skills challenge**

Additionally, in the market you will find people who are highly skilled in batch systems, and often they do not have the same depth of skills in stream processing, and vice versa.

### **Lambda on Azure**

The following is one of the many representative Lambda architecture on Azure for building Big Data pipelines.



*Figure 1: Lambda architecture for big data processing represented by Azure products and services. Note, other Azure and (or) ISV solutions can be placed in the mix if needed based on specific requirements.*

## **What problems do lambda solve vs. what problems does it introduce?**

As stated in the previous section, lambda architecture resolves some business challenges. Various parts of the business have different needs in terms of speed, level of granularity and mechanism to consume data. It also resolves the challenge of the choice of technology, by using the best of the speed layer and batch layer together, and not stretching one product to do both which it isn’t comfortable in doing. Finally, it ensures people with skills dealing with transaction and speed layer can work in parallel and together with people with skills in batch processing.

Although immensely successful and widely adopted across many industries and a defacto architectural pattern for big data pipelines, it comes with its own challenges. Here are a few:

* **Transient data silos:** Lambda pipelines often creates silos that could may cause some challenges in the business. The reporting at the speed layer that the operations team is dealing with, may be different for the aggregate batch layer that the management teams are working with. Such creases may eventually iron out, but it has the potential of causing some inconsistencies.
* **More management overhead:** It also increases the number of subsystems, as a result during maintenance time, many needed to be managed and maintained. This could potentially mean one may need bigger teams with deep and wide skill sets.

## **The emerging big data design pattern**

If there was a way that utilized the right mix of technologies that didn’t need a separate speed or batch layer, we could build a system that has only a single layer and allows attributes of both the speed layer and batch layer. With the technological breakthrough at Microsoft, particularly in [Azure Cosmos DB](https://azure.microsoft.com/en-gb/services/cosmos-db/), this is now possible.

Azure Cosmos DB is a globally distributed, multi-model database. With Cosmos DB you can independently scale throughput and storage across any number of Azure's geographic regions. It offers throughput, latency, availability, and consistency guarantees with comprehensive [service level agreements](https://aka.ms/acdbsla) (SLAs).

Here are some of the key features that renders Cosmos DB as a suitable candidate for implementing the proposed reference architecture where the speed later and the batch layer merges into a single layer.

### **Cosmos DB change feed**

* Most importantly, the key feature that is pivotal in building this emerging big data architectural pattern is the [Cosmos DB change feed](https://docs.microsoft.com/en-us/azure/cosmos-db/change-feed). Change feed support works by listening to an Azure Cosmos DB collection for any changes. Then, it outputs the sorted list of documents that were changed in the order in which they were modified. The changes are persisted, can be processed asynchronously and incrementally, and the output can be distributed across one or more consumers for parallel processing.

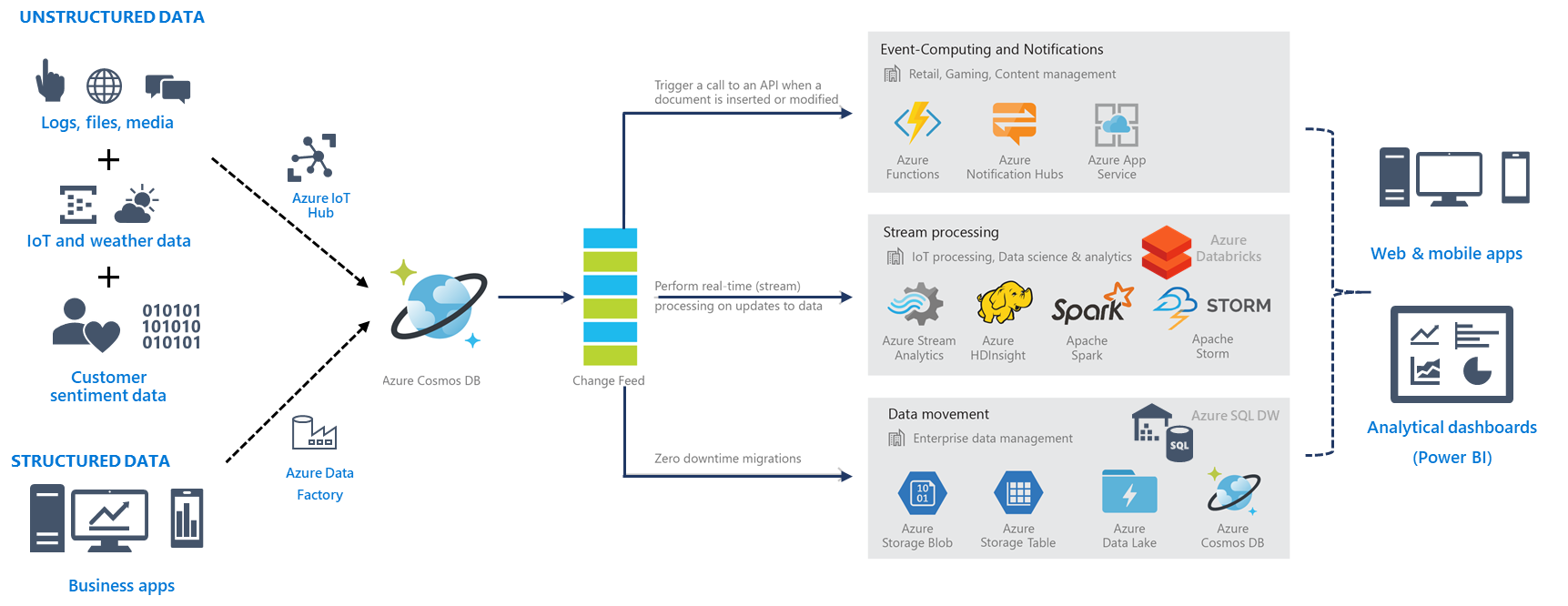
### **Features for speed/hot layer**

* Azure Cosmos DB’s database engine is fully schema-agnostic – it automatically indexes all the data it ingests without requiring any schema or indexes and serves blazing fast queries.
* Cosmos DB allows you to easily scale database throughput at a [per-second](https://docs.microsoft.com/en-us/azure/cosmos-db/request-units) granularity, and change it anytime you want.
* You can [distribute your data](https://docs.microsoft.com/en-us/azure/cosmos-db/distribute-data-globally) to any number of [Azure regions](https://azure.microsoft.com/regions/), with the [click of a button](https://docs.microsoft.com/en-us/azure/cosmos-db/tutorial-global-distribution-sql-api). This enables you to put your data where your users are, ensuring the lowest possible latency to your customers.
* Azure Cosmos DB guarantees end-to-end low latency at the 99th percentile to its customers. For a typical 1KB item, Cosmos DB guarantees end-to-end latency of reads under 10 ms and indexed writes under 15 ms at the 99th percentile, within the same Azure region. The median latencies are significantly lower (under 5 ms).

### **Features for batch/cold layer**

* You can access your data by using APIs of your choice, like the [SQL](https://docs.microsoft.com/en-us/azure/cosmos-db/documentdb-introduction), [MongoDB](https://docs.microsoft.com/en-us/azure/cosmos-db/mongodb-introduction), [Cassandra API](https://docs.microsoft.com/en-us/azure/cosmos-db/cassandra-introduction), and [Table](https://docs.microsoft.com/en-us/azure/cosmos-db/table-introduction) APIs, and graph via the [Gremlin API](https://docs.microsoft.com/en-us/azure/cosmos-db/graph-introduction). All APIs are all natively supported.
* You can also scale storage size [transparently and automatically](https://docs.microsoft.com/en-us/azure/cosmos-db/partition-data) to handle your size requirements now and forever.
* Five well-defined, practical, and intuitive [consistency models](https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels) provide a spectrum of strong SQL-like consistency all the way to the relaxed NoSQL-like eventual consistency, and everything in-between.
* Rapidly iterate the schema of your application without worrying about database schema and/or index management.
* Using the features described above, the following will be an implementation of the emerging architectural pattern.

The following is a diagrammatic representation of the emerging big data pipeline that we have been discussing in this blog:



*Figure 2: Emerging architectural pattern implemented using Cosmos DB for Big Data pipelines as an evolution of the traditional lambda architecture.*

Hence, by leveraging Cosmos DB features, particularly the change feed architecture, this emerging pattern can resolve many of the [common use-cases](https://docs.microsoft.com/en-us/azure/cosmos-db/use-cases). This in turn, gives all the benefits of the lambda architecture, and resolves some of complexities that lambda introduces. More and more customers adopting this and resulting in a successful community, and success of this new pattern and increased adoption of Azure Cosmos DB.

# From Lambda to Kappa: A Guide on Real-time Big Data Architectures

<https://www.talend.com/blog/2017/08/28/lambda-kappa-real-time-big-data-architectures/>

When it comes to real-time big data architectures, today… there are choices. Today, there is more than just Lambda on the menu of choices, and in this blog series, I’ll discuss a couple of these choices and compare them using relevant use cases. So, how do you select the right architecture for our real-time project? Let’s get started.

## **Real-time requirements**

Before we dive into the architecture, let’s discuss some of the requirements of real-time data processing systems in big data scenarios.

The most obvious of these requirements is that data is in motion. In other words, the data is continuous and unbounded. It’s really about **when** you are analyzing this data that matters. If you are looking for answers against the current snapshot of data or have specific low-latency requirements, then you’re probably looking at a real-time scenario.

**See how Beachbody** [**modernized their data architecture**](https://www.talend.com/uncategorized/2018/03/06/beachbody-delivers-a-cloud-data-lake-talend-and-aws/) **and mastered big data with Talend.**

In addition, there are very often business deadlines to be met. After all, if there were no consequences to missing deadlines for real-time analysis, then the process could be batched. These consequences can range from complete failure to simply degradation of service.

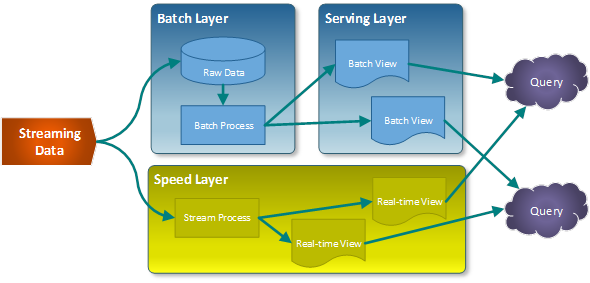
Since we are talking about big data, we also expect to push the limits on volume, velocity and possibly even variety of data.

Real-time data processing often requires qualities such as scalability, fault-tolerant, predictability, resiliency against stream imperfections, and must be extensible.

## **New architectures for the New Data era**

To address this need, new architectures were born… or in other words, necessity is the mother of invention.

The Lambda Architecture, [attributed to Nathan Marz](https://www.manning.com/books/big-data), is one of the more common architectures you will see in real-time data processing today. It is designed to handle low-latency reads and updates in a linearly scalable and fault-tolerant way.



The data stream entering the system is dual fed into both a **batch and speed layer**.

The **batch layer** stores the raw data as it arrives, and computes the batch views for consumption. Naturally, batch processes will occur on some interval and will be long-lived. The scope of data is anywhere from hours to years.

The **speed layer** is used to compute the real-time views to compliment the batch views.

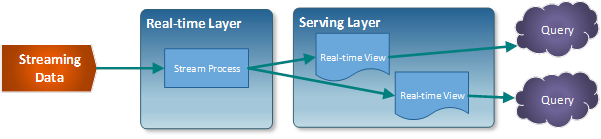
Any query may get a complete picture by retrieving data from both the batch views and the real-time views. The queries will get the best of both worlds. The batch views may be processed with more complex or expensive rules and may have better data quality and less skew, while the real-time views give you up to the moment access to the latest possible data. As time goes on, real-time data expires and are replaced with data in the batch views.

One additional benefit to this architecture is that you can replay the same incoming data and produce new views in case code or formula changes.

The biggest detraction to this architecture has been the need to maintain two distinct (and possibly complex) systems to generate both batch and speed layers. Luckily with Spark Streaming (abstraction layer) or Talend (Spark Batch and Streaming code generator), this has become far less of an issue… although the operational burden still exists.

Next, we’ll discuss the **Kappa Architecture**.

The [Kappa Architecture was first described by Jay Kreps](http://milinda.pathirage.org/kappa-architecture.com/). It focuses on only processing data as a stream. It is not a replacement for the Lambda Architecture, except for where your use case fits. For this architecture, incoming data is streamed through a real-time layer and the results of which are placed in the serving layer for queries.



The idea is to handle both real-time data processing and continuous reprocessing in a single stream processing engine. That’s right, reprocessing occurs from the stream. This requires that the incoming data stream can be replayed (very quickly), either in its entirety or from a specific position. If there are any code changes, then a second stream process would replay all previous data through the latest real-time engine and replace the data stored in the serving layer.

This architecture attempts to simplify by only keeping one code base rather than manage one for each batch and speed layers in the Lambda Architecture. In addition, queries only need to look in a single serving location instead of going against batch and speed views.

The complication of this architecture mostly revolves around having to process this data in a stream, such as handling duplicate events, cross-referencing events or maintaining order- operations that are generally easier to do in batch processing.

## **One size may not fit all**

Many real-time use cases will fit a Lambda architecture well. The same cannot be said of the Kappa Architecture. If the batch and streaming analysis are identical, then using Kappa is likely the best solution. In some cases, however, having access to a complete set of data in a batch window may yield certain optimizations that would make Lambda better performing and perhaps even simpler to implement.

There are also some very complex situations where the batch and streaming algorithms produce very different results (using machine learning models, expert systems, or inherently very expensive operations that must be performed differently in real-time) which would require using Lambda.

So, that covers the two most popular real-time data processing architectures. The next articles in this series will dive deeper into each of these and we’ll discuss concrete use cases and the technologies that would often be found in these architectures.