



# Data Architecture Explained: Components, Standards & Changing Architectures

Data architecture is a framework for how [IT infrastructure](#) supports your data strategy. The goal of any data architecture is to show the company's infrastructure how data is acquired, transported, stored, queried, and secured.

A data architecture is the foundation of any [data strategy](#). It is the “how” when implementing a data strategy.

In this article, we'll look at:

- Business agility
- Data architecture
- Architecture components
- Data standards
- The shift to new architecture

Let's get started.

## **Data architecture supports agility**

In today's world, [the key is agility](#).

Agility allows your company to adapt quickly to the business environment and industry. A particular kind of data architecture can actually enable agility, so you can meet these business demands.

Data architecture is critical to the success of a business (and why we've written extensively on each data component). The world is jumping aboard this framework. The writings and how-to's and best practices are there to share the architecture and help get organizations moving towards it.

## **What is data architecture?**

Data architectures will define a company's livelihood. If a company were a chess piece, the data architecture defines the moves the company can make on the board.

A primitive architecture allows your company to move like a pawn. An advanced architecture can make that pawn a queen.



Picture these different data architectures:

- Storing a file as a .csv on a local hard drive and reading the file into [Tableau](#) on a person's computer for analysis is a very simple kind of data architecture.
- Streaming data from a set of point-of-sale registers to accounting is another kind of architecture.

The data architecture is 100% responsible for increasing a company's freedom to move around the world.

If agility is what is needed to avoid collapse during slow seasons or to capitalize on the spontaneous popularity of a new product, the more advanced the data architecture is, the more capable the company is to take action.

Explicitly, the data architecture:

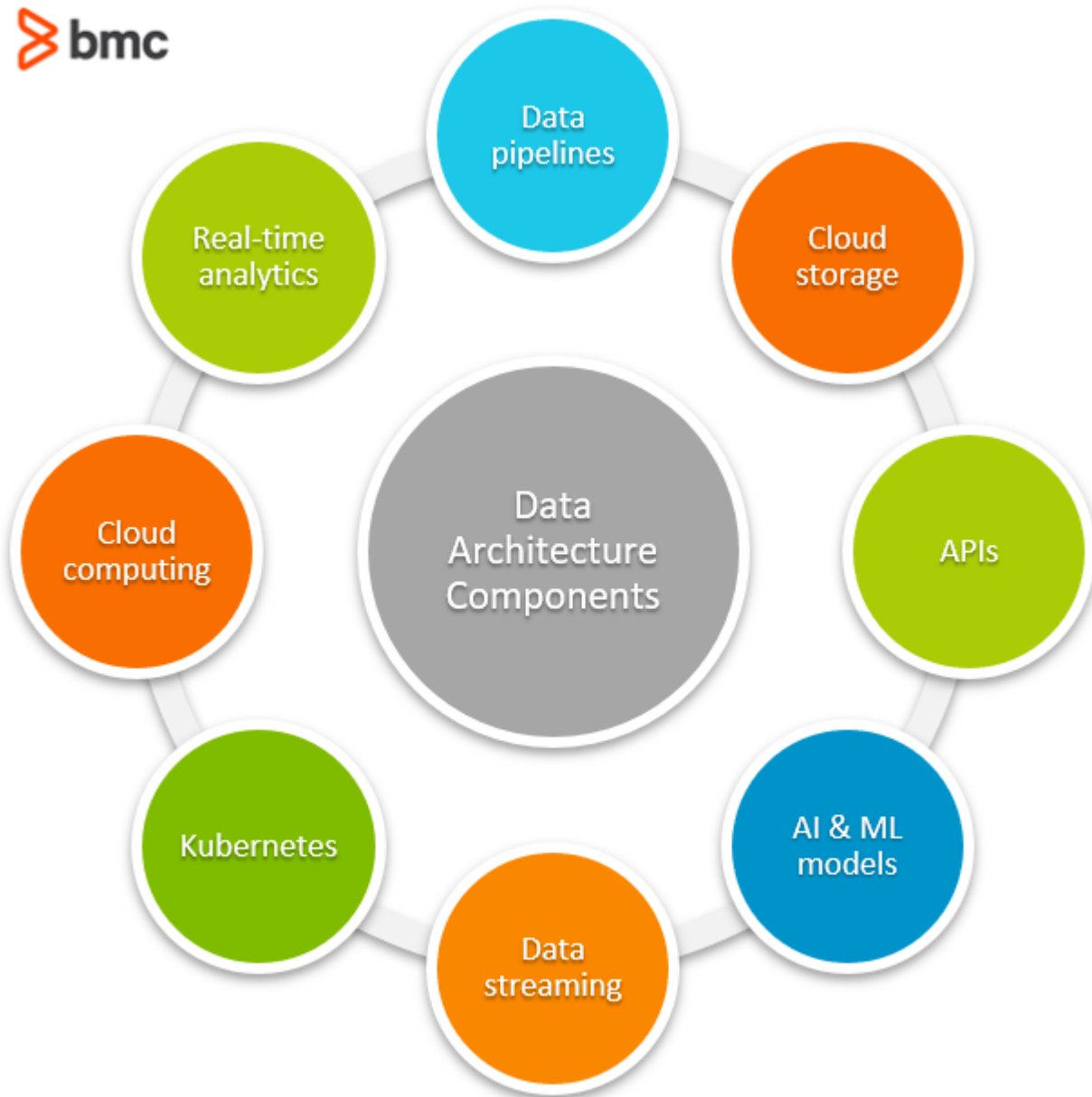
- Gives a fuller picture of what is happening in the company
- Creates a better understanding of the company's data
- Offers protocols by which data moves from its source to being analyzed and consumed by its destinations
- Ensures a system is in place to secure the data

- Grants all teams the ability to make [data-driven decisions](#)

## Components of data architecture

The architectural components of today's data architectural world are:

- [Data pipelines](#)
- [Cloud storage](#)
- [APIs](#)
- [AI & ML models](#)
- [Data streaming](#)
- [Kubernetes](#)
- [Cloud computing](#)
- [Real-time analytics](#)
- And more...



## Data standards

Data standards are the overarching standards of a data architecture, which you apply to areas such as data schemas and security.

## Data schemas

The architecture is responsible for setting the data standards that define what kinds of data will pass through it.

These standards can be achieved by creating a data schema. The data schema defines:

- **Each entity that should be collected.** Schema for contact info, for example, might include name, phone number, email, and place of work.
- **The type of data each piece should be.** For example, name is text data, phone number is integer data, email is text data, place of work is text data.
- **The relationship of that entity to others in the database,** such as where it comes from and where it's going.

Most companies will version their data schema. As [data becomes increasingly pervasive](#), companies will begin using relational databases over more traditional SQL databases.

[Relational \(NoSQL\) databases](#) allow you to easily add data and piece data together more like a network of entities rather than a strict hierarchy of entities. Plus, these relational databases can grow much larger and handle adding data dynamically to the database, where traditional SQL databases could not (or was strongly advised against).

That's why versioning is so vital. Versioning the data schema helps standardize:

- What to find where
- The ability to ask *when* a data was where

(Explore data storage [from database to warehouse to lake](#) and [from hot to cold](#).)

## Data security

Data standards also help set the security rules for the architecture. These can be visualized in the architecture and schema by showing what data gets passed where, and, when it travels from point A to point B, [how the data is secured](#).

Security protocols can include:

- Encrypting data during travel

- Restricting access to individuals
- Anonymizing data to decrease the value of the information upon receipt by receiving party
- Additional actions

## Shifting to new architecture

McKinsey published a great article about [six important changes](#) to consider when building a data architecture in today's world. It highlights the older architectural components, and how it has been updated to the distributed, agile architecture for today's companies.

Here is the short version of these six changes:

1. From on-premise to cloud-based data platforms
2. From batch to real-time data processing
3. From pre-integrated commercial solutions to modular, best-of-breed platforms
4. From point-to-point to decoupled data access
5. From an enterprise warehouse to domain-based architecture
6. From rigid data models toward flexible, extensible data schemas

When thinking about anything related to data—which is arguably everything—you should always consider the data architecture.

## Related reading

- [BMC Business of IT Blog](#)
- [BMC Machine Learning & Big Data Blog](#)
- [BMC Guides](#), offering multi-part tutorials on a variety of data products
- [Data Ethics for Companies](#)
- [3 Keys to Building Resilient Data Pipelines](#)
- [5 Questions to Ask About Data in Your SaaS Environment](#)