**Database-centric Architecture** or **data-centric architecture** has several distinct meanings, generally relating to [software architectures](https://en.wikipedia.org/wiki/Software_architecture) in which [databases](https://en.wikipedia.org/wiki/Database) play a crucial role. Often this description is meant to contrast the design to an alternative approach. For example, the characterization of an architecture as "database-centric" may mean any combination of the following:

معماری پایگاه داده محور یا معماری داده محور دارای معانی مختلفی است که عموماً مربوط به معماریهای نرم افزاری است که در آن بانکهای اطلاعاتی نقش اساسی دارند. غالباً این توصیف برای تضاد طراحی با رویکرد جایگزین است. به عنوان مثال ، توصیف یک معماری به عنوان "پایگاه داده محور" ممکن است به معنای ترکیبی از موارد زیر باشد:

* using a standard, general-purpose [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system), as opposed to customized in-[memory](https://en.wikipedia.org/wiki/Memory_(computers)) or [file](https://en.wikipedia.org/wiki/Computer_file)-based [data structures](https://en.wikipedia.org/wiki/Data_structures) and [access methods](https://en.wikipedia.org/wiki/Access_method). With the evolution of sophisticated [DBMS](https://en.wikipedia.org/wiki/Database_management_system) software, much of which is either free or included with the [operating system](https://en.wikipedia.org/wiki/Operating_system), application developers have become increasingly reliant on standard database tools, especially for the sake of [rapid application development](https://en.wikipedia.org/wiki/Rapid_application_development).
* استفاده از یک سیستم مدیریت پایگاه داده رابطه ای استاندارد ، با اهداف کلی ، برخلاف ساختارهای داده های مبتنی بر حافظه یا فایل های مبتنی بر سفارشی و روش های دسترسی. با تکامل نرم افزار پیشرفته DBMS که بخش اعظم آن رایگان یا همراه با سیستم عامل است ، توسعه دهندگان برنامه به طور فزاینده به ابزارهای استاندارد بانک اطلاعاتی وابسته شده اند ، خصوصاً به منظور توسعه سریع برنامه.
* using dynamic, [table](https://en.wikipedia.org/wiki/Table_(database))-driven logic, as opposed to logic embodied in previously [compiled](https://en.wikipedia.org/wiki/Compiled) [programs](https://en.wikipedia.org/wiki/Computer_program). The use of table-driven logic, i.e. behavior that is heavily dictated by the contents of a database, allows programs to be simpler and more flexible. This capability is a central feature of [dynamic programming languages](https://en.wikipedia.org/wiki/Dynamic_programming_language). See also [control tables](https://en.wikipedia.org/wiki/Control_table) for tables that are normally coded and embedded within programs as [data structures](https://en.wikipedia.org/wiki/Data_structures) (i.e. not compiled statements) but could equally be read in from a [flat file](https://en.wikipedia.org/wiki/Flat_file), [database](https://en.wikipedia.org/wiki/Database) or even retrieved from a [spreadsheet](https://en.wikipedia.org/wiki/Spreadsheet).
* استفاده از منطق پویا و جدول محور برخلاف منطق موجود در برنامه های که قبلاً تدوین شده اند. استفاده از منطق محور جدول ، یعنی رفتارهایی که به شدت توسط محتویات یک بانک اطلاعاتی دیکته شده است ، اجازه می دهد تا برنامه ها ساده تر و انعطاف پذیر تر باشند. این قابلیت یک ویژگی اصلی زبانهای برنامه نویسی پویا است. جدولهای کنترلی را برای جداول که بطور معمول کدگذاری شده و درون برنامه ها به عنوان ساختار داده ها مشاهده می شوند (به عنوان مثال بیانیه های کامپایل نشده) را مشاهده کنید ، اما به همان اندازه می توانید از یک فایل صاف ، بانک اطلاعاتی یا حتی از یک صفحه گسترده بازیابی کنید.
* using [stored procedures](https://en.wikipedia.org/wiki/Stored_procedure) that run on [database servers](https://en.wikipedia.org/wiki/Database_server), as opposed to greater reliance on logic running in middle-tier [application servers](https://en.wikipedia.org/wiki/Application_server) in a [multi-tier architecture](https://en.wikipedia.org/wiki/Multi-tier_architecture). The extent to which [business logic](https://en.wikipedia.org/wiki/Business_logic) should be placed at the back-end versus another tier is a subject of ongoing debate. For example, Toon Koppelaars presents a detailed analysis of alternative [Oracle-based](https://en.wikipedia.org/wiki/Oracle_Database) architectures that vary in the placement of business logic, concluding that a database-centric approach has practical advantages from the standpoint of ease of development and maintainability.[[1]](https://en.wikipedia.org/wiki/Database-centric_architecture#cite_note-1)
* با استفاده از روشهای ذخیره شده که روی سرورهای پایگاه داده اجرا می شوند ، بر خلاف اعتماد بیشتر به منطقی که در سرورهای برنامه رده متوسط در معماری چند لایه کار می کند. میزان قرار دادن منطق تجارت در بخش انتهایی در مقابل ردیف دیگر ، موضوعی است که در حال بحث و بررسی است. به عنوان مثال ، Toon Koppelaars یک تجزیه و تحلیل دقیق از معماریهای جایگزین مبتنی بر اوراکل ارائه می دهد که در جای منطق کسب و کار متفاوت است ، نتیجه می گیریم که یک رویکرد پایگاه داده محور دارای مزایای عملی از نظر سهولت توسعه و نگهداری است.
* using a shared database as the basis for communicating between [parallel processes](https://en.wikipedia.org/wiki/Parallel_computing) in [distributed computing](https://en.wikipedia.org/wiki/Distributed_computing) applications, as opposed to direct [inter-process communication](https://en.wikipedia.org/wiki/Inter-process_communication) via [message passing](https://en.wikipedia.org/wiki/Message_passing) functions and [message-oriented middleware](https://en.wikipedia.org/wiki/Message-oriented_middleware). A potential benefit of database-centric architecture in [distributed applications](https://en.wikipedia.org/wiki/Distributed_application) is that it simplifies the design by utilizing DBMS-provided [transaction processing](https://en.wikipedia.org/wiki/Transaction_processing) and [indexing](https://en.wikipedia.org/wiki/Index_(database)) to achieve a high degree of reliability, performance, and capacity.[[2]](https://en.wikipedia.org/wiki/Database-centric_architecture#cite_note-2) For example, [Base One](https://en.wikipedia.org/wiki/Base_One) describes a database-centric distributed computing architecture for [grid](https://en.wikipedia.org/wiki/Grid_computing) and [cluster](https://en.wikipedia.org/wiki/Computer_cluster) computing, and explains how this design provides enhanced security, fault-tolerance, and [scalability](https://en.wikipedia.org/wiki/Scalability).[[3]](https://en.wikipedia.org/wiki/Database-centric_architecture#cite_note-3)
* استفاده از یک پایگاه داده مشترک به عنوان مبنای برقراری ارتباط بین فرآیندهای موازی در برنامه های محاسباتی توزیع شده ، بر خلاف ارتباط مستقیم بین فرایند از طریق توابع انتقال پیام و واسطه پیام محور. یک مزیت بالقوه معماری پایگاه داده محور در برنامه های کاربردی توزیع شده این است که این طرح را با استفاده از پردازش و نمایه سازی معاملات DBMS تأمین می کند و به درجه بالایی از قابلیت اطمینان ، عملکرد و ظرفیت دسترسی پیدا می کند. [2] به عنوان مثال ، پایگاه اول یک معماری محاسباتی توزیع شده در مرکز پایگاه داده را برای شبکه و محاسبات خوشه ای توصیف می کند ، و توضیح می دهد که چگونه این طراحی باعث افزایش امنیت ، تحمل در برابر خطا و مقیاس پذیری می شود.
* an overall [enterprise architecture](https://en.wikipedia.org/wiki/Enterprise_architecture) that favors shared data models[[4]](https://en.wikipedia.org/wiki/Database-centric_architecture#cite_note-4) over allowing each application to have its own, idiosyncratic data model.
* یک معماری کلی شرکت که از داده های مشترک [4] پشتیبانی می کند و اجازه می دهد هر برنامه مدل داده خاص و خاص خود را داشته باشد.

**Data-centric Architecture — A Different Way of Thinking**

Published: December 12, 2018 | Updated: September 13, 2019  
Posted in: [Data-Centric Engineering](https://www.vistaprojects.com/tech-innovation/data-centric-engineering/), [Technology and Innovation](https://www.vistaprojects.com/tech-innovation/)

Data-centric architecture can have many benefits for industrial projects. Executing [large capital projects](https://www.vistaprojects.com/markets/) is a difficult and challenging process and involves many parts that all need to come together to achieve a common goal.

معماری داده محور می تواند مزایای بسیاری برای پروژه های صنعتی داشته باشد. اجرای پروژه های بزرگ سرمایه ای فرایندی دشوار و چالش برانگیز است و بخش های زیادی را شامل می شود که همه آنها برای دستیابی به یک هدف مشترک باید جمع شوند.

[Project teams](https://www.vistaprojects.com/project-engineering/) need to balance:

* different systems
* complex and, sometimes, competing requirements
* the participation of multiple disciplines and organizations
* تیم های پروژه باید تعادل برقرار کنند:  
  • سیستم های مختلف  
  • نیازهای پیچیده و ، گاهاً رقابتی  
  • مشارکت چندین رشته و سازمان

However, most large capital projects are fraught with issues that can result in a project being late, over budget, or both.

با این حال ، بیشتر پروژه های بزرگ سرمایه مملو از موضوعاتی است که می تواند منجر به تأخیر یک پروژه ، بیش از بودجه یا هر دو شود.

**There is a Better Way**



Data-centric architecture (or model) is a solution that addresses the issues of conventional capital project methodology and delivers positive results. Data-centric execution architecture has been around for a few years and has been used in a limited way [by some owners and operators in Alberta](https://blueprintdea.com/category/case-studies/).

معماری داده محور (یا مدل) راه حلی است که به مباحث روش شناسی پروژه سرمایه معمولی می پردازد و نتایج مثبتی را ارائه می دهد. معماری اجرای مرکز محور چند سالی است که توسط برخی از دارندگان و اپراتورها در آلبرتا به صورت محدود مورد استفاده قرار گرفته است.

Adopting data-centric architecture does not change a capital project’s development phases—feasibility, concept and design basis, front-end engineering design, detailed engineering, construction, and start-up and commissioning—but radically changes how work is performed in these phases.

اتخاذ معماری داده محور ، فازهای توسعه یک پروژه سرمایه را تغییر نمی دهد - امکان پذیری ، مفهوم و مبنای طراحی ، طراحی مهندسی جلو ، مهندسی دقیق ، ساخت و ساز و راه اندازی و راه اندازی - اما اساساً نحوه عملکرد در این فازها را تغییر می دهد.

The most important aspect of data-centric architecture is data. Without accurate and timely data and the means to analyze it, there is no valid basis for making decisions.

مهمترین جنبه معماری داده محور ، داده است. بدون داده های دقیق و به موقع و ابزار تجزیه و تحلیل آن ، هیچ مبنای معتبری برای تصمیم گیری وجود ندارد.

**What does data-centric mean?**

A data-centric outlook is a core concept in **data-centric architecture**, where data is viewed as a critical and perpetual asset used in support of applications to produce deliverables. Within data-centric architecture, the data model precedes the implementation of a given application and remains valid long after the application is gone.

چشم انداز محور داده یک مفهوم اصلی در معماری داده محور است ، جایی که داده ها به عنوان دارایی مهم و همیشگی مورد استفاده قرار می گیرند که در پشتیبانی از برنامه ها برای تولید تحویل مورد استفاده قرار می گیرد. در معماری داده محور ، مدل داده پیش از اجرای یک برنامه معین است و مدت طولانی پس از اتمام برنامه معتبر است.

In a data-centric approach, data must drive the development of projects, designs, business decisions, and culture. The emergence of cloud computing and storage enables organizations to remotely access and analyze large databases to make more **objective, risk-mitigating, and profitable decisions**.

در یک رویکرد محور داده ، داده ها باید توسعه پروژه ها ، طرح ها ، تصمیمات تجاری و فرهنگ را هدایت کنند. ظهور رایانش ابری و ذخیره سازی ، سازمانها را قادر می سازد تا از راه دور به پایگاه داده های بزرگ دسترسی پیدا کرده و آنالیز کنند تا تصمیمات عینی ، کاهش ریسک و سودآوری بگیرند.

**Document-Driven Execution**

The traditional project execution method relies on document-based deliverables produced at certain times throughout a project’s development.

روش اجرای سنتی پروژه به تحویل های مبتنی بر اسناد متکی است که در زمان های مشخصی در طول توسعه پروژه تولید می شوند.

Producing deliverables are important events, as they are used to measure project progress. More importantly, these documents are a physical embodiment of the project data and knowledge: they are the means by which data and knowledge transfer between project disciplines, project tasks, and between project phases.

تولید مواد تحویل رویدادهای مهمی هستند ، زیرا از آنها برای اندازه گیری پیشرفت پروژه استفاده می شود. مهمتر از همه ، این اسناد تجسم فیزیکی از داده ها و دانش پروژه هستند: آنها وسیله ای هستند که داده ها و انتقال دانش بین رشته های پروژه ، وظایف پروژه و بین مراحل پروژه انتقال می دهند.

Data-knowledge transfer can also occur between different engineering companies, as it is common practice for owner-operators to use different engineering companies for different phases of a project.

انتقال دانش داده همچنین می تواند بین شرکتهای مختلف مهندسی اتفاق بیفتد ، زیرا این معمول است که صاحبان اپراتورها از شرکتهای مختلف مهندسی برای مراحل مختلف یک پروژه استفاده کنند.

The document-driven model has several issues:

* errors and misalignment of data
* slow response to change
* the data is static
* information silos with multiple data instances

مدل محور اسناد دارای چندین موضوع است:  
• خطاها و سوء استفاده از داده ها  
• پاسخ آهسته برای تغییر  
• داده ها ثابت است  
• سیلوهای اطلاعاتی با چندین نمونه داده

These issues lead companies to make decisions using poor-quality information.

این مسائل باعث می شود تا شرکت ها با استفاده از اطلاعات بی کیفیت تصمیم گیری کنند.

**Data Silos**

All too often, for a variety of reasons, data contained on a deliverable does not align with data on other deliverables or across the various information systems.

اغلب اوقات ، به دلایل مختلف ، داده های موجود در یک تحویل با داده های سایر تحویل ها یا سیستم های اطلاعاتی مختلف مطابقت ندارند.

Slowly and progressively, **data discordance** builds within the project. This situation results in a low level of confidence in project information and data, forcing users to verify the accuracy of the data before being confident enough to use it.

به تدریج و به تدریج ، عدم توافق داده ها در پروژه ایجاد می شود. این وضعیت منجر به اطمینان کم در اطلاعات و داده های پروژه می شود و باعث می شود کاربران قبل از اطمینان کافی برای استفاده از آن ، صحت داده ها را بررسی کنند.

This verification activity is especially important when there is a changeover in engineering contractors between project phases.

این فعالیت تأیید به ویژه هنگامی مهم است که تغییر پیمانکاران مهندسی بین مراحل پروژه وجود داشته باشد.

Change is inevitable in projects and being able to manage change orders effectively is crucial. Nowhere is this more important than in the latter project phases, where change can have a significant effect on cost and schedule.

تغییر در پروژه ها اجتناب ناپذیر است و توانایی مدیریت سفارشات تغییر به طور مؤثر ضروری است. هیچ جا این مهمتر از مراحل بعدی پروژه نیست ، جایی که تغییر می تواند تأثیر قابل توجهی در هزینه و برنامه داشته باشد.

In practice, implementation of changes in the document-based method is slow, complicated, and open to error.

در عمل ، اجرای تغییرات در روش مبتنی بر اسناد ، کند ، پیچیده و برای خطا است.

Documents are static things that present information at a particular point in time. This is not necessarily bad, it’s just not optimal, particularly in dynamic and complex systems, such as capital asset projects.

اسناد چیزهای ایستا هستند که اطلاعات را در یک نقطه خاص از زمان ارائه می دهند. این امر لزوماً بد نیست ، به ویژه در سیستم های پویا و پیچیده ، مانند پروژه های دارایی سرمایه ، مطلوب نیست.

In the early project phases, a design can be quite fluid and undergo many iterations to find the best design option. Data changes trigger document updates, which will in turn trigger changes in the dependent documents.

در مراحل اولیه پروژه ، یک طرح می تواند کاملاً روان باشد و برای یافتن بهترین گزینه طراحی ، چندین بار تکرار شود. تغییرات داده باعث بروزرسانی اسناد می شوند که به نوبه خود باعث تغییر در اسناد وابسته می شود.

It takes time for the information to work its way through the various project deliverables and systems, and some documents might not be updated, while others are incorrectly updated.

زمان لازم است تا اطلاعات از طریق سیستم های مختلف تحویل پروژه و برخی از اسناد به روز شود ، و برخی از اسناد ممکن است به روز نشوند ، در حالی که برخی دیگر به اشتباه به روز نمی شوند.

Project execution has evolved to use a plethora of information systems (e.g. applications such as CAD) to manage the many tasks involved in a project. While essential to the execution of complex capital projects, these systems can lead to information silos.

اجرای پروژه با استفاده از مجموعه ای از سیستم های اطلاعاتی (به عنوان مثال برنامه هایی مانند CAD) برای مدیریت بسیاری از وظایف درگیر در یک پروژه تکامل یافته است. در حالی که برای اجرای پروژه های سرمایه پیچیده ضروری است ، این سیستم ها می توانند به سیلوهای اطلاعاتی منجر شوند.

Traditional document-centric project execution often leads to inefficient data silos.

اجرای پروژه متداول سند محور سنتی اغلب منجر به سیلوهای داده ناکارآمد می شود.

Each system has its source of data, which results in multiple instances of data occurring across the project. Managing **data congruence** across these silos is time-consuming and inefficient.

The article [Top 10 Engineering Documents and Data Management Challenges](http://news.aucotec.com/top-10-engineering-document-data-management-challenges/) explores the issues of the document-driven model further.

هر سیستم منبع داده خود را دارد که منجر به چندین مورد داده در پروژه می شود. مدیریت هماهنگی داده ها در این سیلوهای وقت گیر و ناکارآمد است.  
مقاله 10 اسناد مهندسی برتر و چالش های مدیریت داده ، موضوعات مدل محور اسناد را بیشتر مورد بررسی قرار می دهد.

**Data-Centric Architecture**

Data-centric execution methodology is based on the premise that data is the primary and permanent asset of a project and everything else revolves around the data. The data-centric execution method has two key advantages over the conventional document-driven method:

1. a single source of truth (SSOT)
2. up-to-date data

روش اجرای محور داده ها مبتنی بر این فرض است که داده ها دارایی اصلی و دائمی یک پروژه هستند و هر چیز دیگری حول داده می چرخد. روش اجرای داده محور دارای دو مزیت اساسی نسبت به روش متداول اسناد محور است:  
1. یک منبع حقیقت واحد (SSOT)  
2. داده های به روز

The SSOT involves creating a single data model that is used by all project personnel and information systems. Using a single project-wide data source eliminates information silos created by information systems and their attendant data sources and structures, thus eliminating multiple instances of data. All systems and project participants use **data that originates from the SSOT**.

SSOT شامل ایجاد یک مدل داده واحد است که توسط کلیه پرسنل پروژه و سیستم های اطلاعاتی مورد استفاده قرار می گیرد. با استفاده از یک منبع داده در طول پروژه ، سیلوهای اطلاعاتی ایجاد شده توسط سیستمهای اطلاعاتی و منابع و ساختارهای داده آنها را از بین می برد ، بنابراین چندین نمونه از داده ها را از بین می برد. همه سیستم ها و شرکت کنندگان در پروژه از داده هایی استفاده می کنند که از SSOT سرچشمه می گیرد.

There is no data discordance in a data-centric environment, as there is only one instance of the data. As data is created or changed, it is made available to all connected information systems—there is no lag waiting for someone to take action and update a deliverable or system and the accuracy of the data is very high.

در یک محیط داده محور هیچ گونه اختلاف نظر وجود ندارد ، زیرا فقط یک نمونه از داده ها وجود دارد. با ایجاد یا تغییر داده ها ، در اختیار همه سیستم های اطلاعاتی متصل قرار می گیرد - هیچ انتظاری در انتظار اقدام و بروزرسانی یک سیستم تحویل یا سیستم وجود ندارد و صحت اطلاعات بسیار بالا است.

The value of the SSOT is significant, as it **provides decision-makers with the most accurate and timely data** on which to act.

A project data model must be created to establish the SSOT. The data model is a graphical representation of the attributes of things, the relationships between entities, and the flow of data.

The model comprises several layers of increasing detail and specificity. Each functional application, such as a CAD or procurement system, will read and write through the shared data model.

ارزش SSOT قابل توجه است ، زیرا اطلاعات دقیق و بهینه ای را برای تصمیم گیری در اختیار تصمیم گیرندگان قرار می دهد.  
برای ایجاد SSOT باید یک مدل داده پروژه ایجاد شود. مدل داده ها گرافیکی از ویژگی های چیزها ، روابط بین موجودات و جریان داده ها است.  
این مدل شامل چندین لایه از افزایش جزئیات و ویژگی ها است. هر برنامه کاربردی ، مانند CAD یا سیستم تهیه ، از طریق مدل داده اشتراکی می خواند و می نویسد.

Data modelling for an engineering design project involves defining:

* the types of assets (tags), such as mechanical and electrical equipment and instrumentation
* the attributes that make up each asset type, such as pump attributes for each pump type (the number of attributes assigned to an item can be in the hundreds)
* the entity relationships to ensure the flow of data between information systems (which entities need to interact with each other)
* who (the commercial entity and specific person) is responsible for populating each attribute
* who requires each attribute and when

مدل سازی داده ها برای یک پروژه طراحی مهندسی شامل تعریف:  
• انواع دارایی ها (برچسب ها) مانند تجهیزات مکانیکی و برقی و ابزار دقیق  
• ویژگیهایی که هر نوع دارایی را تشکیل می دهند ، مانند ویژگی های پمپ برای هر نوع پمپ (تعداد ویژگی های اختصاص داده شده به یک مورد می تواند در صدها نفر باشد)  
• روابط موجودیت برای اطمینان از جریان داده ها بین سیستم های اطلاعاتی (که نهادها نیاز به تعامل با یکدیگر دارند)  
• چه کسی (واحد تجاری و شخص خاص) مسئولیت جمعیت هر ویژگی را دارد  
• چه کسی به هر ویژگی و چه زمانی نیاز دارد

**The Data-Centric Project**

A data-centric organization understands the importance of data as a project driver and collecting data is an important first step. The data-centric project must:

* eliminate silos and analyze data simultaneously
* centrally manage data
* improve efficiencies based on new information learned

یک سازمان داده محور از اهمیت داده به عنوان یک درایور پروژه می فهمد و جمع آوری داده ها اولین گام مهم است. پروژه محور داده باید:  
• سیلوهای را از بین ببرید و داده ها را همزمان تجزیه و تحلیل کنید  
• داده ها را بطور مرکزی مدیریت کنید  
• بهره وری را بر اساس اطلاعات جدید آموخته شده بهبود بخشد

Although data is of primary importance to the project, data-centric execution architecture requires people and systems to create a holistic project execution model. Adopting digital execution architecture and data centricity will transform the nature of project execution for oil and gas capital assets.

اگرچه داده ها از اهمیت اولیه پروژه برخوردار هستند ، اما معماری اجرای داده محور به مردم و سیستم ها نیاز دارد تا یک الگوی اجرای پروژه جامع را ایجاد کنند. اتخاذ معماری اجرای دیجیتالی و محوریت داده ، ماهیت اجرای پروژه را برای داراییهای سرمایه نفت و گاز تغییر خواهد داد.

<https://en.wikipedia.org/wiki/Database-centric_architecture>

https://www.vistaprojects.com/blog/data-centric-architecture/