Model Based Systems Engineering (MBSE) on AWS: From Migration to Innovation

Model Based Systems Engineering (MBSE) on AWS: From Migration to Innovation: AWS Whitepaper

Table of Contents

This whitepaper is for historical reference only. Some content might be outdated and some links might not be available.

Model Based Systems Engineering (MBSE) on AWS: From Migration to Innovation

Publication date: **September 20, 2021** (*Document history*)

Model Based Systems Engineering (MBSE) is a modern approach to the conventional practice of document-based systems engineering. MBSE benefits from modern cloud computing technologies, microservices, AI/ML, advanced analytics and others.

These technologies not only enable broad adoption of MBSE by engineering organizations, but also go beyond the current prospects of MBSE and bring innovation, flexibility, scalability and cost optimization. MBSE has been recently adopted by aerospace, energy, and automotive customers and growing in other industries where complex products - made of multitude of engineering disciplines and collaboration - are required to design, build, test, sustain and monitor the whole product lifecycle through their lifecycle.

AWS provides both building block technologies and solutions tailored to your needs. This whitepaper addresses both MBSE *developers* who develop MBSE technologies and MBSE *users* who use MBSE tools. It also provides introductory information about MBSE and its challenges for newcomers to this technology.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console, you can review your workloads against these best practices by answering a set of questions for each pillar.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

Topics

- [What is MBSE and why do industries start to use?](#)
- [MBSE to Model Based Enterprise / Engineering (MBE)](#)
- [Challenges with MBSE: Engineering and IT](#)
- [Addressing Challenges with the Cloud](#)
- [Digital Engineering: Cloud Based Organizational Transformation for IT and Engineering Agility](#)

## What is MBSE and why do industries start to use?

Today's multi-disciplinary, multi-supplier, and multi-application/tool product development environment encourages customers to adopt **agile practices** under firm deadlines. Agility is especially important in more traditional engineering environments such as **aerospace**, **energy**, and **automotive** industries, where *waterfall product development* has been long practiced and perfected.

As these products are highly complex, following a build-based agile iteration is expensive. Moreover, these industries are highly regulated and exceptionally quality-driven. Hence, the agility should not bear "haste creates waste" but should bear technologies that allow data-driven decisions, visibility, traceability, simplicity, and preferably automation to "go fast" while "sticking to highest standards".

Complex product development requires inter-disciplinary and inter-supplier/company work. Hence, customers look for ways to be agile in this complex and dynamic environment. At the center of the complexity, we see **Systems Engineering (SE).**

According to the [INCOSE (International Council on Systems Engineering)](#), [Systems Engineering](#) is *a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods.*

The term **system** here is a wide-range definition, including hardware, software, information, processes, people, configurations, supply chains, regulators and geographies. One of the fundamental functions of SE is to conduct the **V**erification and **V**alidation **(V&V)** of the product development and overall lifecycle.
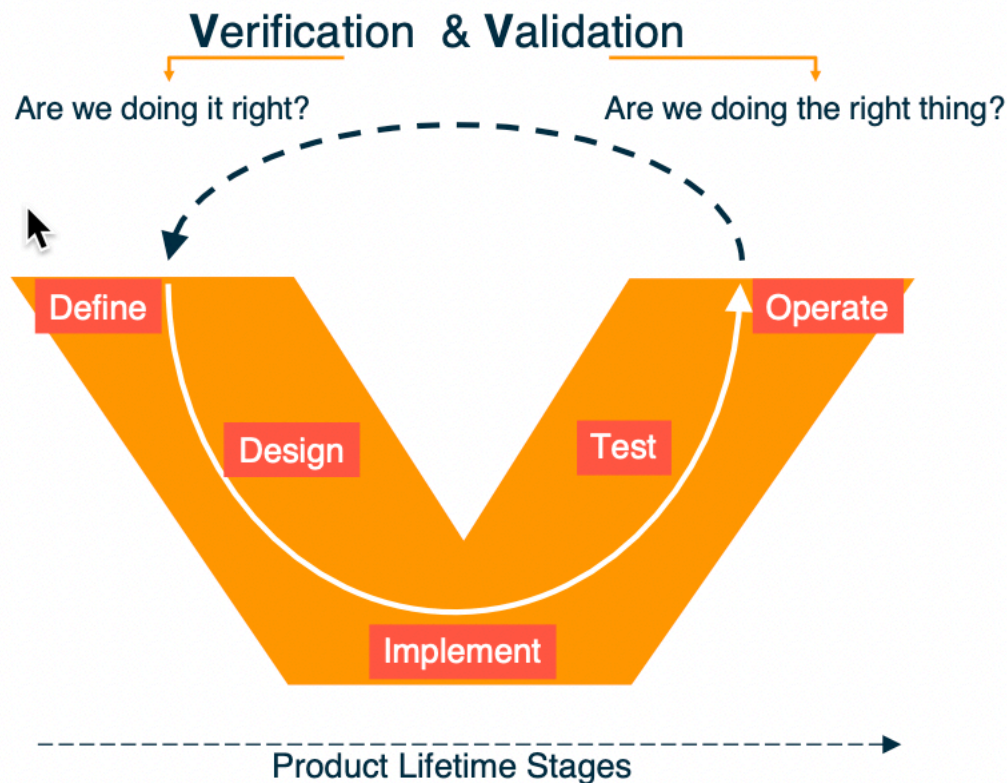
Verification and Validation through a product development, MBSE, traditional SE cost and cost of defects to the project

(V&V figures are adapted from [Clarus Concept of Operations](#), [Archived on 2009-07-05](#) at the [Wayback Machine](#), Publication No. FHWA-JPO-05-072, Federal Highway Administration (FHWA), 2005.)

Above you can see the stages of a waterfall product development method with the V&V diagram.

V&V is designed to follow the product development and the product itself in compliance with the **requirements**. Verification asks "are we building the product right?", while Validation asks "are we building the right product?".

V&V helps minimize late defect discovery by engineering and manufacturing functions orchestrated by systems engineering costs more with issue discovery near the end. It is good to know about serious issues sooner, rather than later.

Traditionally, Systems Engineers follow a **document-based approach**. The challenge is the inability to adapt to changes and complications with collaboration due to a large number of stakeholders and moving parts in the projects that overall slows down the product development process.
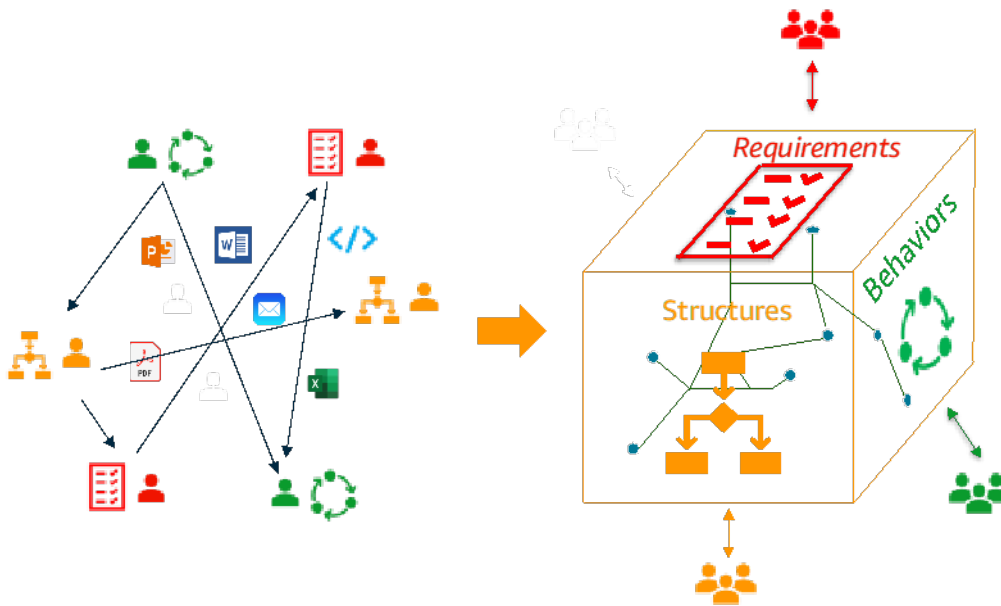
Document Based SE MBSE

Meanwhile, in the diagram above, you can see on the left side the challenge of a traditional approach.

It often involves lots of meetings, sharing of documents, emails with attachments, spreadsheets, pdfs and models "flying" between stakeholders in a document-based systems engineering setting. It can also mean that defects are prone to be discovered near the end of the project.

As we established earlier, late defect discovery costs more. It is far better to become aware of serious issues sooner than later.

Without this kind of approach, you could end up with multiple "sources of truths" for engineers - creating complications such as conflicting data, using inaccurate data, or late-minute correction of source data. This not only damages product quality, but also delays projects as it slows down overall product development.

Finally, with the IoT/Industry 4.0 transition in these industries, there are - and will be -more data. The size of data further creates a burden on an already-complicated situation.

At this triangle of complexity, speed and quality, systems engineers bring the "modeling" to replace documents. **Model-Based Systems Engineering (MBSE)** handshakes "**behaviors**", "**requirements**" and "**structures & relationships**" expressed with "**models"**.

INCOSE describes **MBSE** as "*the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.*"

The **model** here corresponds to an abstracted concept such as a phenomenon, relationship, structure, or system which can be represented graphically, mathematically, or physically.

The model facilitates concepts to aid decision-making processes by explaining the behaviors and simulating events. Models are low-fidelity *system models*, rather than high-fidelity engineering discipline models such as CAD models, Finite Element Models, etc. We will talk about system models in the "MBSE Under the Hood" section.

The other component of MBSE is the structures. **Structures** are any information (data or metadata) that exhibits a tree-like representation made of semi-static components. For example, the structures can be abstract concepts such as **Product Variations** and **Configurations,** or it can be as physical as a Bill-of-materials (BOM) or an assembly of components and moving systems such as an engine.

Behaviors define the relationship between structures or the parts inside the structures. The whole validation process in MBSE is automated by low-fidelity simulations like this one. Those bounding parameters are based on **requirements**. The diagram on page 7 shows how structures, behaviors, and requirements are connected under MBSE.

In this whitepaper, we have incorporated "**people**" into the MBSE framework. This is because the projects adopting MBSE are working in a highly multi-disciplinary environment in multi-stakeholder (internal & external) settings. Since we talk about automation and agility, people/user responsibilities, access and identity management, provided functionalities, work scope, and more are embedded parts of the MBSE workflows.

The term, "*Model-Based*", varies by its scope and domain. For example, in software engineering, **Model-Based '*Software*' Engineering (MB'*S*'E)** has its unique features such as **automatic code generation** for system behavior and testing**.** MBSE's auto code generation feature is especially important for highly regulated and safety critical industries such as aerospace industries.

On the other hand, the 'model-based' approach can be extended to a **Model Based Enterprise (or Engineering)** which covers the complete lifecycle of the product, not limited to virtualization but including physical space such as products, manufacturing, and more. MBE will be discussed in the following sections.

From the public sector perspective, US DoD 2018 Digital Engineering Strategy can be referenced. That document describes that MBSE is a core enabler of the strategy, whose first of five pillars is *to formalize the development, integration, and use of models to inform enterprise and program decision making*. Hence, AWS customers in complex discrete manufacturing are looking to MBSE to:

- Reduce time to field capabilities (time-to-market)

- Improve efficiency and reuse across life cycle phases and programs with digital traceability

- Enable early risk identification (quality) and cost reduction

- Leverage data and models as an authoritative source of truth

- The goal of an integrated digital ecosystem where stakeholders can access the data, functions, and elements needed to work in a purely digital manner

MBSE under the Hood: SysML, Digital Threads and Ontology

The main purpose of MBSE is to replace documents with reusable models. Therefore, the models should be *system (i.e. discipline, type)* and *platform (i.e. tools, routines) agnostic*. Hence, this creates the complication while connecting these into a common language and schema.

To do that, MBSE follows **Unified Modeling Language (UML)** standard. Some examples are SysML (see *Flexible Views for View-based Model-driven Development* By Burger, Erik. KIT Scientific Publishing, Nov 14, 2014. Pg. 250), OPM (see Dori, Dov. "Object-process analysis: maintaining the balance between system structure and behaviour." *Journal of Logic and Computation* 5, no. 2 (1995): 227-249), ARCADIA, and others where the popular is **SysML (System Modeling Language)**. SysML is an extension to UML specialized for SE applications and originally created as open-source. Currently, AWS does not have native SysML libraries. However, you can bring your own library (BYOL) and build SDKs for your needs. AWS Solutions Architects, AWS Partners, and/or AWS Professional Services will help you in every step.
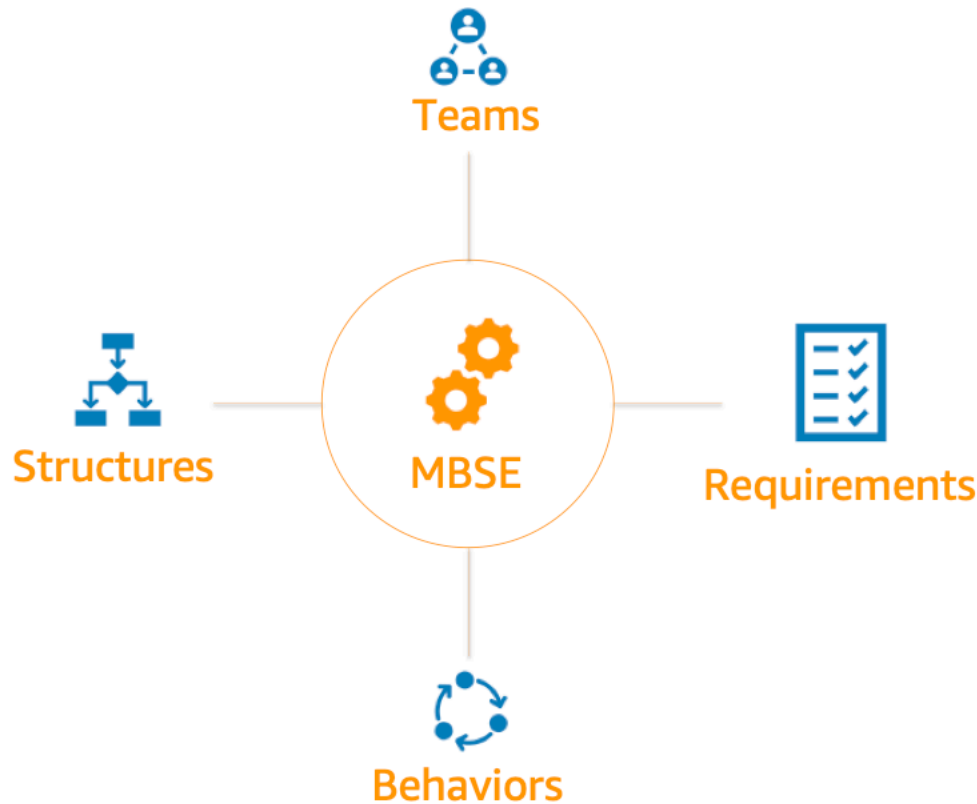
Harnessing requirements, structures and behaviors with MBSE

The next important concept in MBSE is **Ontology.** Ontology establishes well-defined domain
concepts in terms of the *objects*, *definitions*, and *relationships* inside and among the systems by
bringing **formal semantics**. It is especially the case since these are **"systems of systems"** where
each system is made of multitude of systems.

The meaning of the models, not the details but the emergent properties and metadata, needs to
be expressed and understandable without calling their subject domain experts. Hence, ontology is
a necessary step towards standardization and enabling data interchange which enables *flexibility*
and *interoperability*.

This formalized application of modeling and simulation improves system requirements definition,
design, verification and validation, operation, and overall performance.
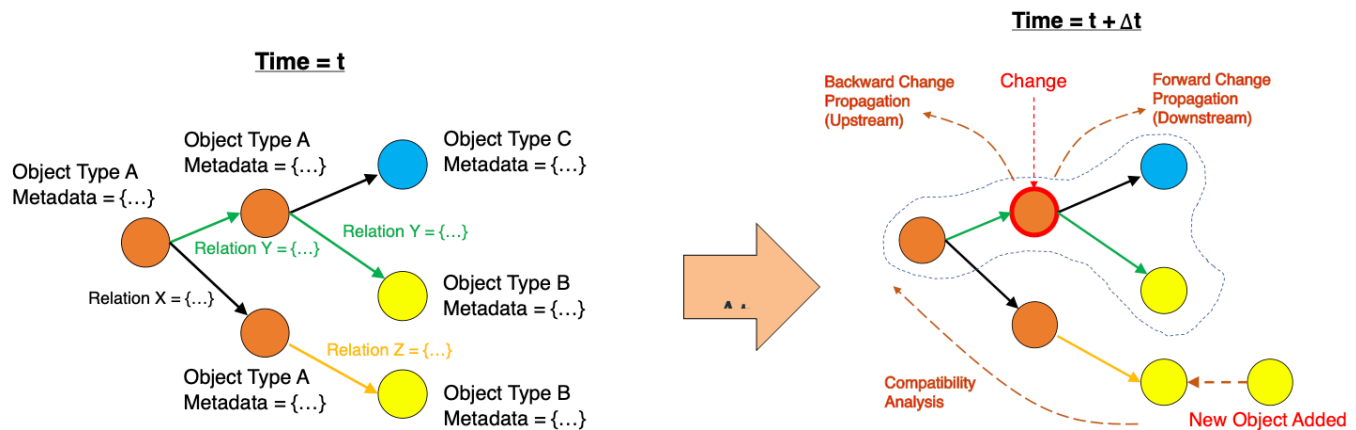
A graph model made of different types of objects (nodes) and relations with each has different metadata changing in time (left), a change in the object propagating changes backwards and forwards and new object addition compatibility analysis (right).

The other concept that enables an MBSE model is called **Digital Thread**. The "thread" should be a standardized way to "connect" or "plug" the different types of models into a system of systems model. You may notice the overall features of the MBSE forms a "**graph**" or "network topography" as shown in the diagram above.

It shows a few examples for the impact of an engineering change in an object or a newly added object and relation.

In this model, MBSE can simulate the propagation of changes forwards and backwards as well as perform compatibility analysis based on the requirements sets. Hence, MBSE is a key to the **traceability** and **change management** as well as automation to discover defects and issues earlier.

The other feature of the MBSE is to provide **Integrated (System) Design Environments (ISDEs)** that provides a visual environment for the stakeholders to build, interact and work with system models.

The above discussion and concepts will be critical to see the benefit of cloud and AWS technologies to enable MBSE and MBSE related technologies as discussed in the following sections.

MBSE to Model Based Enterprise / Engineering (MBE)

Today, industries work on extending the MBSE being a "capable systems engineering tool" towards being a **complete digital transformation** so called **Model Based Enterprise or Engineering (MBE)**. In this whitepaper, we did not differentiate MBSE with MBE but discussed MBSE under the light of

MBE where it is a broader and more capable approach made of multiple technologies to harness
physical and virtual aspects of complete product lifecycle.
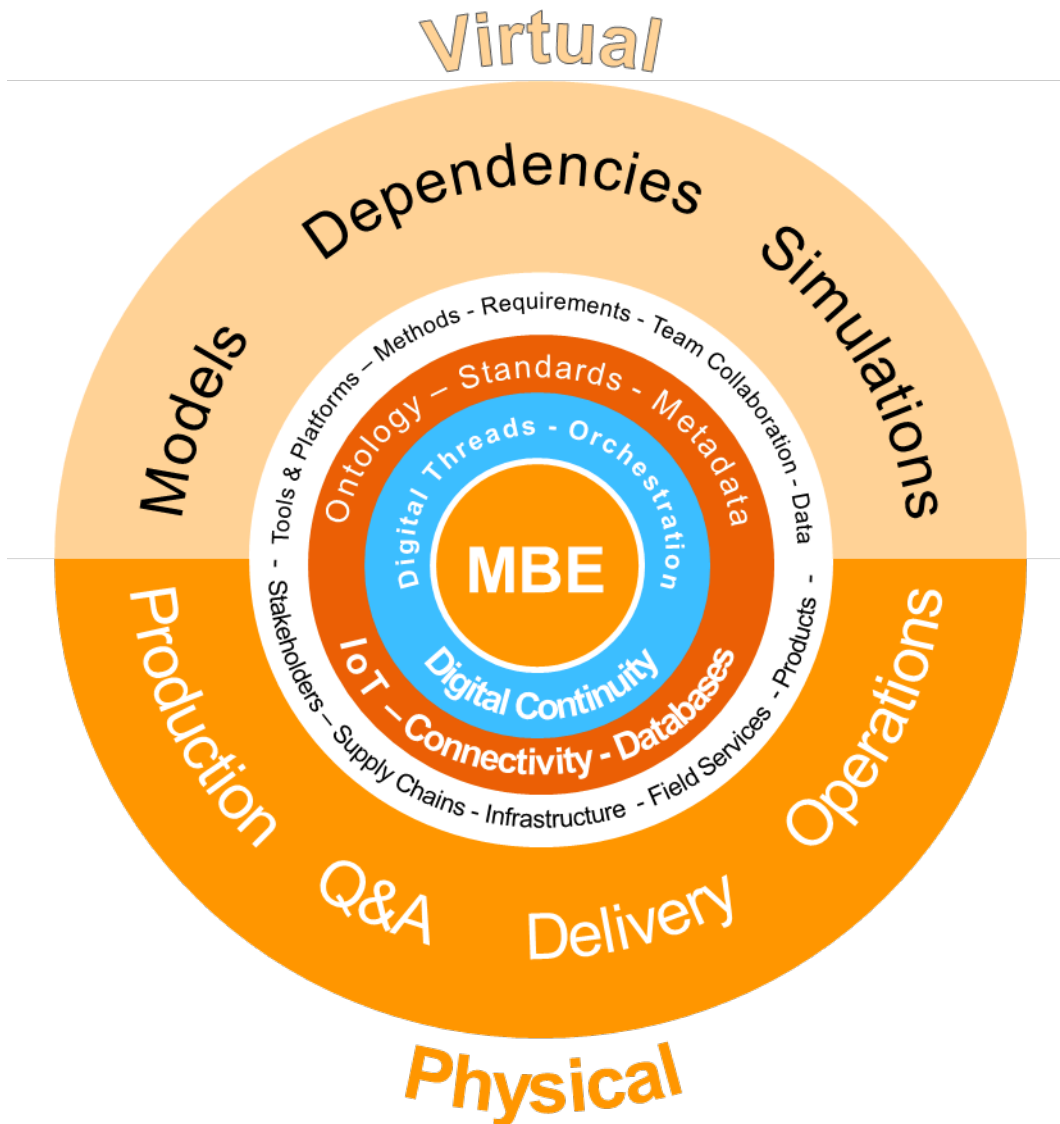
MBE - Model Based Enterprise/Engineering at the center of a product lifecycle

The diagram above attempts explain the MBE approach under a single setting divided into physical
and virtual aspects of a complete product lifecycle. In virtual, it includes abstract concepts such
as "dependencies" implying "relationships" or "configurations" both dynamic and static, whereas
models and simulations are from all disciplines of engineering, management and operations.

At a high level, the physical part is about production, testing/Q&A, delivery, operations including
product disposal and recycling. Under this level, there are plenty of variables, which are relatively
independent such as stakeholders, services, products in physical or tools, infrastructure,
requirements and data in the virtual part.

Especially, the "data" piece is critical as it is both "big data" and very diverse data including structured and non-structured data.

Under the abstracted definitions of ontologies, including semantics and agnostic relationships, and metadata standardizing the virtual connectivity. In the physical "realm", we can talk about connectivity in global scale; high throughput ingestion and streaming, databases; relational, non-relational or object storage.

All of those aspects are connected via Digital Continuity with the help of Digital Threads and Orchestration with APIs, messaging, events and business logics as to be discussed in the following chapters.

Challenges with MBSE: Engineering and IT

MBSE come with its own challenges; both for IT department enabling the technology and the engineering teams using it. Throughout this whitepaper, we will address challenges and solutions to those two team challenges.

Perhaps the main challenge of MBSE is its relatively early stages of Technology Readiness Level as an overall technology. Scholars, universities, industries and practitioners work on envisioning and developing the technology. As MBSE functionalities develop, more capabilities are expected, this causes more expectations that requires more development in the technology. In parallel, digitalization (digital transformation) of product development, especially waterfall type of product development creates extra demand on MBSE and MBSE related technologies by the MBSE adopters. In addition, MBSE developers works on implementing new technologies on the IT space; such as database types, architectures and IoT.

One of the challenges of implementing MBSE in IT space is the **dynamic scalability**. Since MBSE grows exponentially, the storage and database for MBSE should be flexible and therefore dynamically scalable.

Therefore, you may be looking for **virtually unlimited scalability**, **reliability**, **durability**, **interoperability**, **flexibility,** and **cost-effectivity** for your MBSE solution.

MBSE does not work alone. It works in a complex and dynamic engineering environment. There are a wide variety of configurations that MBSE needs to support. It should be operable in a multitude of devices, software, APIs, endpoints, environments, engineering practices, mechanical sympathy, and others. The challenge is to put the MBSE at the center of engineering. This is referred as **interoperability**.

The next challenge is **extending MBSE to the edge** or**,** vice-versa, **connecting the edge to the MBSE**. With more and more devices are connected, there are more data and shadow devices required to build models. This is similar to **Digital Twins**. To keep the MBSE at the center of the "correct information", the MBSE should reach the edge and other types of internal and external resources.

Another challenge is the **data sovereignty** and **access management of model sharing** since MBSE is to be used by multiple-suppliers distributed all around the world. Hence, granular access management and model sharing with minimum privileges are still a challenge of which failure poses security risks for MBSE practitioners.

Having MBSE at the center of processes and information in the product lifecycle, there would be **other functionalities attached to MBSE**. For example, Multi-Disciplinary Optimization (MDO) or Design Space Exploration (DSE) could all depend on MBSE.

The final challenge is the overall human hesitancy for the adoption of new technologies. MBSE shakes the existing practices of engineering for the good. However, like everything new, it has resistance. Therefore, MBSE needs to be **flexible to adapt to the current engineering practices**.

MBSE on-premises face extra challenges as IT challenges are added on the top of technological and organizational challenges mentioned above. MBSE requires considerable compute power, storage, wide variety of databases. It also needs to be flexible; hence, monolithic architecture approach would slow down the functionality of the MBSE tools in the long run.

Most of the time**, MBSE users are not IT companies**. Hence, IT infrastructure management, guessing capacities, planning for infrastructure retirement, keeping the infrastructure up-to-date, including extra technologies such as IoT, all yield more work and therefore, **overhead costs**.

Moreover, engineering teams practicing MBSE may discover more **incompatibilities** or **missing functionalities** as they use it over time. Hence, IT teams may need to work on these missing functionalities- or provide extra work-around solutions that create **technical debt for their MBSE** solution. Therefore, customers will work on finding simple, but versatile, solutions to start with an MBSE solution or solution built on MBSE.

In the next chapter, we first talk about how cloud would help MBSE and go beyond.

Addressing Challenges with the Cloud

MBSE is here to bring agility with high quality products by providing models and single source of truth. The aforementioned functionalities and challenges of MBSE require wide variety of technologies.

Hereby, cloud brings the fundamental value propositions such as **on-demand resources**, **stop guessing the capacity with elasticity**, **cost effectivity with pay-as-you-use and economics of mass**, **wide-selection of up-to-date technologies**, **experimentation culture with IaS** (Infrastructure as Software), **going global** and **elasticity**.

The *cloud-first* approach benefits not only **MBSE developers** also the **engineering companies who adopt MBSE** as a tool – staying at the center of engineering operations. MBSE developers can provide solutions with the state-of-the-art technologies, as going to be mentioned in this whitepaper, such as using **serverless technologies** to compute and databases.

A serverless approach can remove administrative tasks and provide a granular way of offering "pay-as-you-use". This is especially useful for MBSE, which is mostly transactional where an event driven architecture that can provide potential cost savings.

Similarly, a **microservices approach** can further decouple functions and services to bring **reliability** and **high availability**. For example, standardization of messaging and ontology can be performed using cloud native standards such as **Restful APIs** and **JSON** payloads, respectively.

Moreover, an **IaS approach** can further provide agility to your IT and even engineering teams who can "get the functionality whenever necessary".

MBSE practitioners are in industries with complex product developers, where many suppliers work on the same project joining all around the world. **Globally distributed edge locations globally while focusing on data residency requirements in specific geographies** is easy with cloud.

You can deploy edge locations to bring low-latency MBSE applications while dedicating data residency locations in your desired locations.

With AWS, you can even employ **AWS Outpost** to leverage your existing offices and local datacenters while using cloud technologies. This approach brings elasticity, speed and reliability compared to a silo-ed MBSE solution.

The **virtually unlimited storage** of cloud brings scalability to MBSE. You can also deploy hybrid solutions with **primary storage being the cloud (**or **the secondary being the cloud)** while still using your on-premises infrastructure.

The final advantage of the cloud is **sustainability**. According to the Amsterdam-based GRI (Global
Reporting Initiative), the disclosures of Environmental, Social and Governance (ESG) have now
become the mainstream regarding assessment of organizational performance.

As a part of **Amazon Sustainability Data Initiative (ASDI)**, at AWS, we are committed to running
our business in the most environmentally friendly way possible.

As of 2018, AWS had already received 50% of its energy from renewable resources and AWS is
committed to get 100% of energy from renewable energy by 2025. So you can reduce your carbon
footprint by moving to cloud and AWS.

Complex discrete manufacturing Industries are facing increasing global regulation including
sustainability targets, and increased pressure to decrease time to market for new product
introductions.

Hence, increasing productivity while complying with sustainability goals can be applied to scale
applications, such as MBSE/MBE, which spans over manufacturing, design, edge locations, devices
and more. Leveraging renewable energy resources with AWS for your MBSE/MBE infrastructure can
help you to commit your sustainability goals.

Digital Engineering: Cloud Based Organizational Transformation for IT and Engineering Agility

Most MBSE solutions are based on on-premises siloed solutions - or partially use the benefits of
the cloud. Indeed, using cloud technologies is a journey, rather than a binary approach.
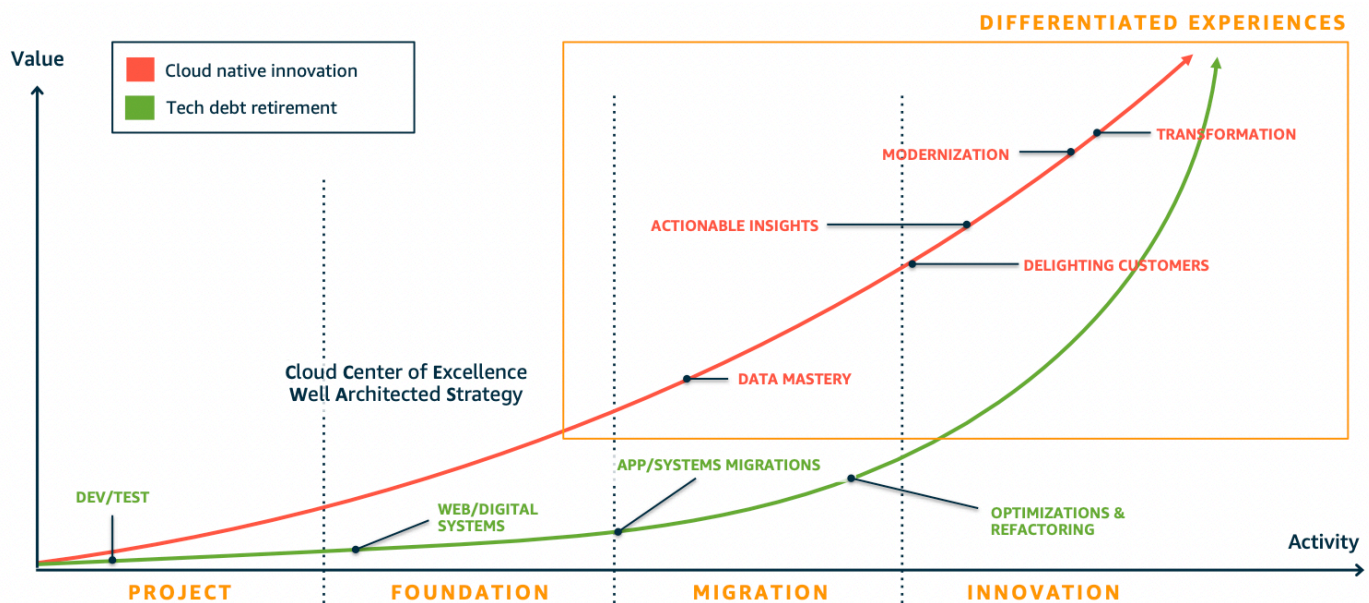
Enterprise Transformation Journey with Cloud

The above figure shows a typical journey for an enterprise. We typically see customers start with a *project phase* where they start testing the cloud implementation. Then, the foundation stage has been setup with employing frameworks such as **Cloud Adoption Framework** such as building **Cloud Center of Excellence (CCoE)** and working on **Well-Architected Strategies**.

The next stages of *migration* and *innovation* are when the value of the cloud becomes evident. These are the stages where our customers benefit from differentiated experience.

With migration, customers start utilizing scalability, elasticity and cost savings; generally, via hybrid solutions gradually moving applications from on-premises to cloud. In time, customer start to utilize cloud native technologies via the **experimentation culture** which is enabled via the IaS and IT agility of the cloud.

This kicks in the innovation phase, where customers modernize their applications and add extra features such as **AI/ML technologies, analytics, serverless**, and more.

Hence, in the following section, we divide the MBSE with AWS into two categories; **Migration of MBSE** and **Innovation with MBSE.**

MBSE on AWS: From Migration to Innovation

Topics

Overview

AWS brings solutions to the above challenges. You can benefit from:

- **Scalability and Elasticity**. The following technologies and solutions bring scalability object storage and databases, compute and supporting essential technologies that provide virtually unlimited space for your MBSE application. We focus on AWS Managed Services that provide dynamic scaling and minimize the management of your MBSE IT workloads. AWS provides both relational and non-relational, including graph databases and object storage. You can even bring serverless databases such as **Aurora Serverless** for MySQL or PostgreSQL, non-relational (key-value) as **Amazon DynamoDB**, graph native database such as **Amazon Neptune** or **Amazon S3** as a data-lake with AWS Glue to catalog your model data or **Amazon Kinesis** for data warehouse. Since MBSE is a critical cornerstone, any technology backing MBSE must also be **highly durable** and **reliable**.

  Another aspect of scalability is the "**scalability of engineering**" attained by enhancing the interoperability of engineering tools, engineering collaboration, and managing the complexity that arises from multi-stakeholder environments and bringing low-latency applications online globally. You can more easily adopt new technologies to your MBSE - or augment value-delivering technologies such as AI/ML, HPC, Analytics - as a "plug-in" through APIs with the size desired of resources managed by AWS. This whitepaper also proposes augmenting micro-process level engineering collaboration based on events and messages as an extension to MBSE.

- **Agility for IT and Engineering**. You can bring agility not only to IT operations, but also to engineering operations. For the most, the agility offered by cloud to the IT operations might be apparent. On the other hand, the agility of the engineering through the cloud can be hidden. Indeed, agility is brought to you by **automation**, **"experimentation" culture** that automatically deploys and dismantles resources as desired, ease of engineering collaboration enabled by

workflow management, **transparency of operations based on events and messages** and **harnessing single source of data/information cataloged by AWS services**. AWS's data lake (we call it **Artifact Store** and **Data Catalog** for MBSE) can ease the file and data collaboration among the teams and suppliers where you can assign **granular access management** to relevant parties with desired privileges. AWS's tools to catalog that data would further reduce the process of data query by engineers. Moreover, **event-based architectures** that we will be discussing would provide visibility to micro-level engineering transactions. This brings more transparency in real-time, which means fewer emails, presentations, and meetings – allowing for a focus on the engineering activities themselves. You can also bring **Shared Services Platform (SSP)** to centrally manage, govern and deploy necessary resources for your engineering and IT teams. SSP can also act as the basis for the all mentioned modules in this whitepaper. Similarly, you can have **Service Catalog** to create a selection of technologies that can be deployed on-demand. The technologies and approaches explained here opens **Continuous Integration (CI)** and **Continuous Deployment (CD)** capabilities for your organization.

- **Interoperability and Flexibility**. You can bring flexibility and interoperability to your existing MBSE and MBSE related functions. You can augment new capabilities now or over time, as the technology progresses, with a **microservices approach**. You can **abstract data continuity layers** and **bring orchestration tools using AWS Application Integration Services** to enhance the interoperability of your MBSE. You can employ a **Digital Continuity Layer** as discussed in the following sections. AWS breadth and depth of application integration services - mostly based on serverless technologies such as messaging, message broker services, queuing, workflow management, Restful APIs and GraphQL APIs, and more - can enable flexibility and interoperability for your MBSE solution. The additional flexibility supports hybrid architectures, where you can build solutions partially using the cloud. You can employ local AWS technologies in your offices/data centers or to the edge using **AWS Outposts** or build high-speed, reliable private connections between your design offices/datacenters/edge locations and AWS cloud such as **AWS Storage Gateway** and **Amazon Direct Connect**.

- **Experiment More.** With 200+ services in AWS, you can promote innovation through experimentation for your IT and engineering teams**.** The main idea is to minimize your time spent maintaining and managing infrastructure and repeatable activities. Solutions and services such as **Shared Services Platform (SSP)**, **Service Catalog**, **AWS CodePipeline** can be used for custom build and on-demand and ephemeral resources for your teams to experiment with. With more than **400+ different types of chips as of August 2021** and automatic deployment, your engineering teams can also perform **on-demand deployments for high-fidelity simulations requiring large compute power**. Then you can bring automation to catalog that data and store the results in Artifact Store (discussed above). In this way, MBSE **can also embrace high-fidelity**

simulations, which may not be possible with on-premises infrastructure or manual processes. The outcome is to increase efficiency and give you back more time to focus on value delivering activities such as innovation and experimentation.

- **MBSE in Global Scale.** [230+ points of AWS presence serving 245+ countries and territories](#), and AWS global network backbone would help you go global in MBSE and MBSE related services to bring low latency applications. This is especially important for addressing **data sovereignty, geographical access management, while providing low-latency MBSE service**. Moreover, MBSE's value in bringing usable models, product and process can be performed globally using **read-replications**, **global tables/databases** and **primary/secondary connections** and **serverless IaS deployments**. This means, you can keep your data in the primary location that you decide makes sense, while providing read-access or API aces optimized for the edge such as **Amazon CloudFront**, **AWS Lambda@Edge** or **API Gateway** optimized at the edge. In addition, you can use **AWS Global Accelerator** to **employ static IPs always available globally**, even though a fail-over happens in your MBSE backend.

- **Extending MBSE to Edge.** With more processing power embedded into the devices, vehicles and manufacturing, it becomes almost a necessity to extend existing (on-cloud or on-premises) technologies to the edge. this is especially important for **traceability of changes** from design to production or in the reverse order; production (ie. Defects or issues) to the design. MBSE would be fed by data representing the overall lifecycle of a product. Hence, MBSE extends to **full product lifecycle** which requires a native, reliable, offline mode capable, light weight processing power, storage, AI/ML and connection between **your MBSE** and **the devices/vehicles/machines at the edge** and **the production lines**. In addition, the design process would be extended to the production to speed up experimentation to bring in more agile practices or just react quickly to production defects or changes stemmed from either side. Hence, you may be looking for **extending automation from design with MBSE to production**. For example, you can **automate 3D printing prototyping from the design to the edge** and get **feedback information from the production to the design team** through AWS IoT.

- **AI/ML in every layer of MBSE.** We especially see significant value of NLP (Natural Language Processing) technologies in MBSE. For example, you can use **Amazon Textract to automatically read your hand-written legacy documents/tables** or **Amazon Comprehend to establish formal semantics among models of disciplines for your ontology using** [Amazon Comprehend custom entities](#). **Amazon Translate** can be used to provide better **international collaboration**. The other level of AI/ML is through the **Amazon Neptune ML**. Since the graph database is dynamic, you can employ large variety of **ML algorithms for networks** such as finding patterns, anomaly detection, shortest distance, hub finding, unused branches and more. Since these data change in time, you can get insights of your operations, supply chain, design cycles and other processes.

For example, **Amazon Forecast** can be used for such purpose. Finally, you can bring large amount of AI/ML algorithms using **Amazon SageMaker AI**. Especially, for the NLP, you can use **Hugging Face with Amazon SageMaker AI**. We believe the AI/ML part would make the different in MBSE applications and will be the future in this technology field.

- **Serverless Technologies**: Another key, promising technology to enhance MBSE services is **serverless**. This is due to the transactional operations of MBSE - based on changes, commits, simulation, overall CRUD activities. Hence, serverless can not only take over all of management from your shoulders, also bring cost effectiveness. As mentioned before, **serverless is not limited to compute** such as AWS Lambda, but databases, workflow managements, messaging, APIs, and more.

- **Granular Access and Identity Management and Security**: Our customers would like to enable agility with international, multi-supplier collaboration - all while ensuring that the data is not accessed outside of the defined setting and secured in transit and at-rest. Identity and Access management and Security is such a large and deep topic that this whitepaper provides high-level information regarding MBSE and provides links on this topic.

- **Experience.** Our customers come with different needs and targets regarding MBSE. Since 2006, AWS has helped millions of customers from diverse industries.

Enterprise Transformation for your MBSE powered by AWS

Following the enterprise journey shared below, we divided this section into two groups:

- MBSE Application Migration to AWS

- MBSE Innovation with AWS
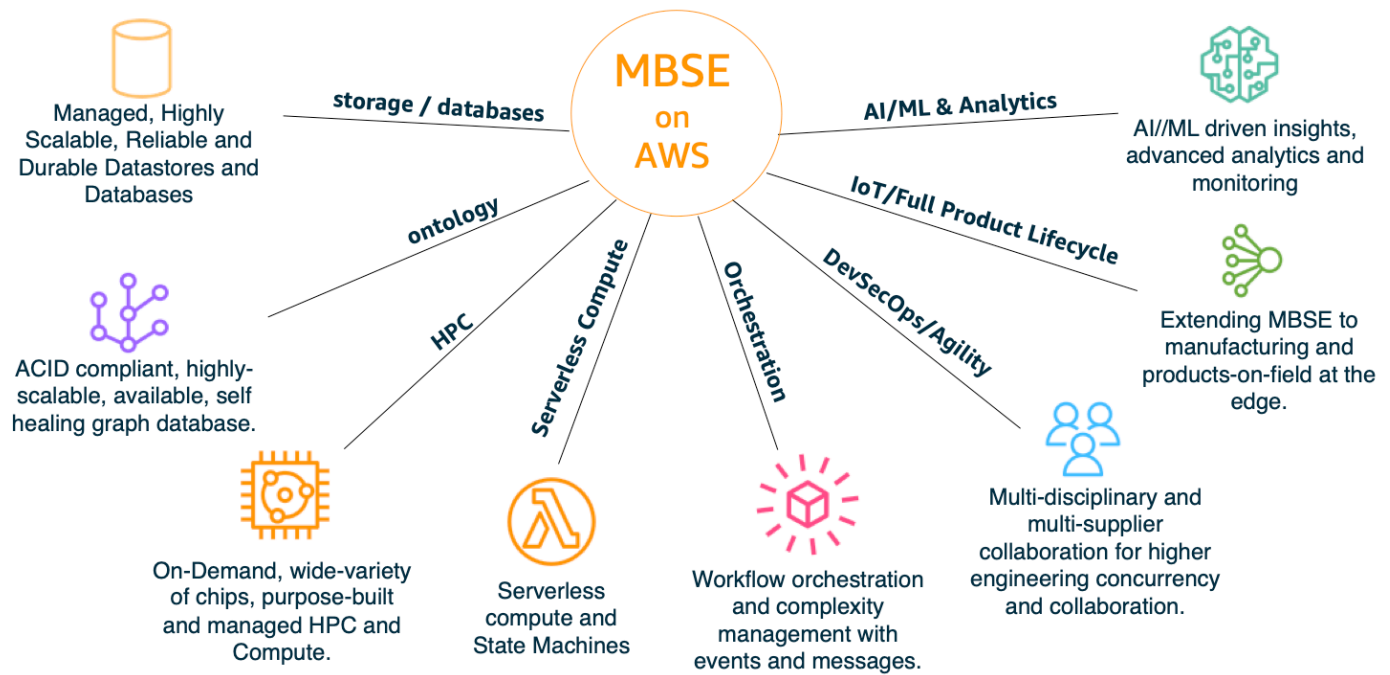
Single view of MBSE on AWS and the value delivering technology domains.

Migration

You can *rehost* or *re-platform* MBSE on AWS. This section follows the 6-Rs of migration strategy in AWS. We will start with *re-hosting* and then directly move to the other end of "Rs", *refactoring*. The other 4-Rs in between these two extremes would be a combination of these.

Rehosting MBSE Application

Rehosting MBSE means lifting and shifting your existing MBSE services to the cloud. You can use **AWS Application Migration Service** to migrate your workloads, with a wide selection of chip and cloud storage types available.

You can still bring flexibility, agility, scalability, and availability without changing your existing setup. You can build your own MBSE **Amazon Machine Image (AMI)** and deploy your applications on **EC2** instances, Kubernetes with **EKS,** or containers with **ECS**. You can also employ **Amazon AppStream 2.0** to virtually deploy a non-persistent desktop and MBSE application virtualization such as IDEs on demand.

On the other hand, you can also lift & shift your MBSE databases to AWS. You can re-host the databases on **Amazon EBS** and **Amazon FSx for Windows File Server.** You can go slightly beyond

simple rehosting and re-platform your databases on **Amazon RDS** to bring a cost-efficient and resizable capacity while automating time-consuming administration.

Amazon RDS is available on several database instance types - optimized for memory, performance or I/O - and provides you with six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server. You can use the **AWS Database Migration Service** to migrate easily or replicate your existing MBSE and MBSE-related databases.

You can use infrastructure automation by providing **Service Catalog for MBSE and Related Services**. Once you have the AMIs and design infrastructure configuration, **AWS CloudFormation** templates can be built for desired MBSE stack(s). Then, these stacks can be deployed in a CI/CD pipeline by your IT or through a self-service setting such as using **AWS CodePipeline**. For example, a MDO (Multi-Disciplinary Optimization) or DSE (Design Space Exploration) type of compute-heavy (but parallel processing) jobs can be included as CloudFormation Template that can deploy **ephemeral compute deployments, but with persistent storage**.

Therefore, the whole process of deploying the lift-and-shift MBSE can be automated where the configuration can be following the best practices (security, cost optimization, etc.). With appropriate **Identity and Access Management** (IAM) using **AWS Identity and Access Management (IAM)**, these configurations are only built/modified by authorized users.

Refactoring MBSE Application

We recommend refactoring towards breaking the monoliths into microservices through APIs and AWS Managed Services. The microservices approach can further bring flexibility to your MBSE workloads as new technologies can be augmented through Restful APIs (**Amazon API Gateway**) and GraphQL APIs (**AWS AppSync**). This chapter is illustrated as *MBSE Backend* diagram further below.

You can use any of the below recommendations - or adopt them all at once - during the refactoring. You may notice that you can bring **serverless technologies** for cost-effectiveness and to remove infrastructure management tasks altogether.

You can start refactoring/rearchitecting **your Digital Thread with APIs**. One of the refactoring of MBSE Digital Threads is employing **Amazon API Gateway**, a fully-managed service that allows you to create, publish, maintain, monitor, and secure APIs at scale.

This is especially the case for SysMLv2.0, enabling API-based communication for the new MBSE applications. You can also consider using **AWS AppSync** to build APIs with GraphQL since it is easier to develop applications with GraphQL, by giving the ability to query multiple databases, microservices, and APIs with a single GraphQL endpoint.

The other refactoring is employing **AWS Lambda,** a serverless compute service that lets you run your code with zero administration. You can directly use **CRUD (Create, read, update, delete)** operations to MBSE directly with AWS Lambda functions. These are especially useful for MBSE since the operations are transactional and rather event-based, such as in a CRUD form.

AWS Lambda can work natively with Amazon API Gateway to bring a serverless business logic to your Digital Thread when triggered by the API Gateway. Therefore, the **millisecond pricing granularity** of AWS Lambda allows you to run your business logic only when needed - and only with the time used (measured in milliseconds).

You can even perform **low-fidelity MBSE simulations** with AWS Lambda as long as the simulation does not exceed the maximum 15 minutes of runtime duration, which is the limit of AWS Lambda, as of today. AWS Lambda can support 6 vCPus and up to 10 GB of memory, which provides a decent amount of computing power for most low-fidelity simulations.

AWS Lambda can perform most of the MBSE simulations, not limited to CRUD operations. This can make **MBSE serverless**; therefore, you benefit from having 'millisecond' cost granularity, high-scalability, reliability, and zero-administration business logic can be achieved.

Along with AWS Lambda, **AWS Step Functions** can also help you to create complex business logic with a **State Machine** to perform stateful operations. If you prefer an open-source approach for the workflow, you can consider using **Amazon Managed Workflows with Apache Airflow (AMWAA)**, which a managed service for Apache Airflow on AWS. You can keep temporary and permanent data on **Amazon S3** with virtually unlimited storage space, cost-effective storage with 11 nines of durability (99.999999999%).

You can keep your costs under control for the stored metadata using **Amazon S3 Lifecycle Management,** which automatically moves colder or cost-effective tier storage as defined by you or determined by an embedded AI.

If you would like to go with containers and Kubernetes instead of AWS Lambda, you can select **AWS Fargate** to stick to a serverless approach that fits into MBSE operations. Please see Serverless on AWS to learn more about other resources.

The next refactoring step is for the databases. MBSE typically uses both relational and non-relational databases. **Amazon DynamoDB** can serve as a low-latency key-value and document database for the MBSE. **Amazon DynamoDB** takes the burden of server management from you while providing a highly scalable and reliable key-value and document database required for MBSE.

Amazon DynamoDB brings virtually unlimited throughput and storage with 11 9's of durability. Globally shared multi-tenant and multi-region MBSE applications can benefit technologies such as **Global Tables**. In addition, your product development teams benefit from low-latency using the caching on Amazon DynamoDB with **DynamoDB Accelerator (DAX)**.

MBSE requires relational databases. As mentioned before, you can employ Amazon Aurora for your MySQL or PostgresSQL databases on the top of Amazon RDS. **Amazon Aurora** features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 128TB per database instance. Similar to Global Tables, you can use **Amazon Aurora Global Databases** to employ an easily scalable relational database in MBSE globally read by consumers.

Global Databases enables fast local reads - typically with less than one second latency - in multiple regions, as well as providing disaster recovery in a region wide outage. These features are especially important for critical applications consumed globally such as MBSE.

Finally, you can also benefit from serverless technology in your MySQL and PostgresSQL databases with **Amazon Aurora Serverless**. Another advantage of Aurora is its direct integration of Amazon Sagemaker for incorporating machine learning (ML) and AI algorithms. With this feature, you can speed up your ML inference by up to x100.

All of the multi-level relationships in different domains and the same domains require a **graph database for ontology**. Here, you can benefit from using **Amazon Neptune**. The core of **Amazon Neptune** is a purpose-built, high-performance graph database engine optimized for storing billions of relationships and querying the graph with milliseconds latency. Amazon Neptune supports Property Graph and W3C's RDF, and their respective query languages such as Apache TinkerPop Gremlin and SPARQL.

Amazon Neptune is highly available, with read replicas, point-in-time recovery, **continuous backup to Amazon S3**, and replication across Availability Zones. Neptune is secure with support for HTTPS encrypted client connections and encryption at rest. Neptune is fully managed, so you no longer need to worry about database management tasks and activities.

Look for the best balance between **open-shared data to better collaboration** and **access controls to ensure** that unintentional data is secure and not shared, mainly in the aerospace and defense industries.

Using AWS, you can **provide identity and access management tools** that allow you to define
granular access controls and guardrails in multiple levels including resources, end-points and users.
You can also incorporate solutions to consider data residency while giving minimum privilege,
federated access to your MBSE data globally in the way you define. To learn more about identity
and access management, please visit the AWS IAM site..

# Innovation

In this chapter, we go beyond and above conventional MBSE software and applications of which
high-level architecture is shown in the diagram below. Please note that you see "AWS … Services"
icons in the figure to indicate a family of services rather than individual service not to confine the
architecture into a specific service and to simplify the figure.

Since the needs for MBSE are diverse for our customers, the following technologies are **modular**
where each module is **mutually inclusive** and **independent**. Therefore, we recommend that the
reader investigates each option separately, while keeping the overall perspective unified in case all
of the solutions can be applied at the same time.

This chapter explains how you can:

- Bring **Digital Continuity** layer to connect to your offices, edge, and vehicle devices – as well
  as ingest data from thousands of internal and external resources in petabyte scale. Manage
  complexity with an **Orchestration layer for Digital Thread** as a part of digital continuity.

- Extend multi-disciplinary/multi-supplier collaboration in micro-process scale with **Extended
  Workforce Collaboration built on MBSE**.

- Adopt **Intelligent Datalake for MBSE ("Artifact-Store with Artifact Catalog")** as single source
  of data where engineers can put their excels, pdfs, model data, etc.

- Bring **AI/ML for MBSE** to provide NLP for MBSE for ontology, legacy document integration,
  obtain valuable insights and analytics based on overall operations/events and data mentioned
  before.

- Employ **Shared Services Platform (SSP)** to centrally manage, govern your applications as well
  provide secure environment for IT and engineering experimentation.
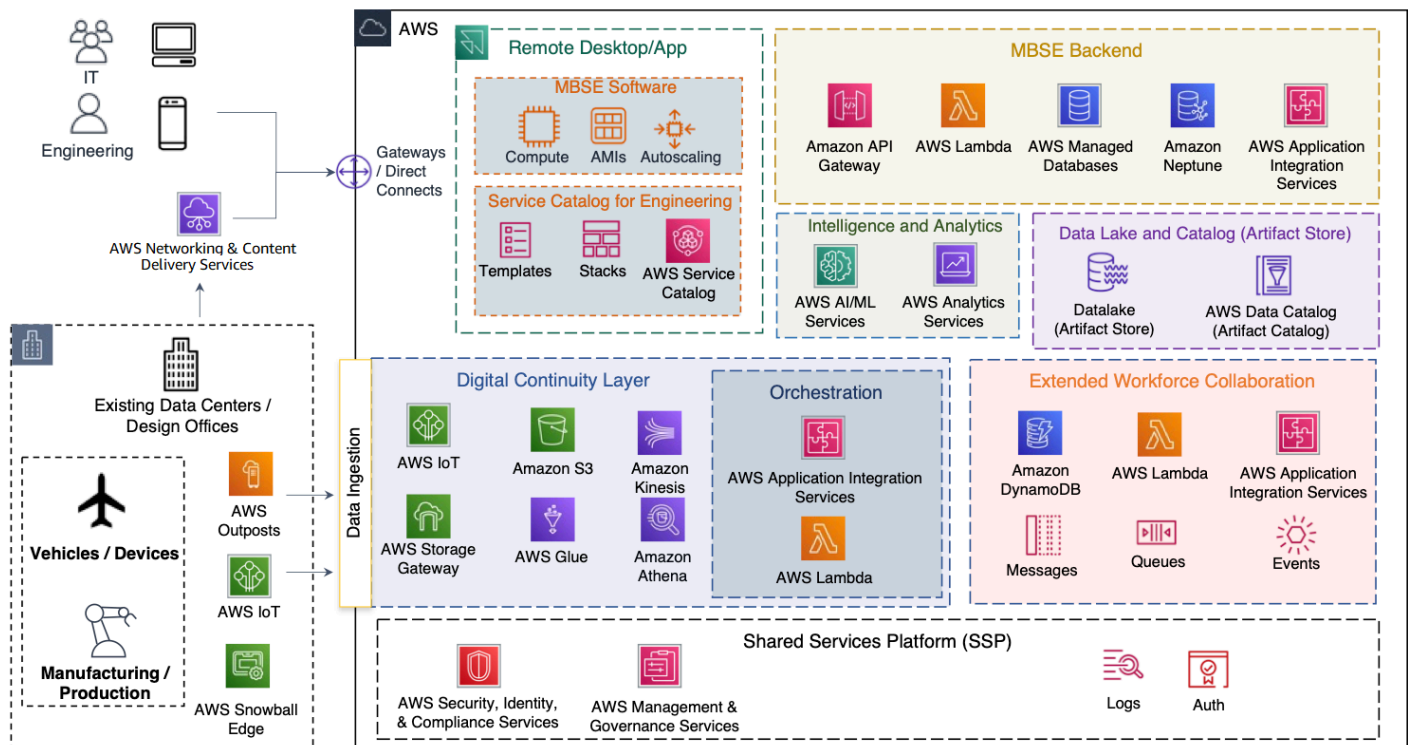
Modules and AWS services for MBSE on AWS (High-level Architecture)

Digital Continuity Layer

MBSE works in a complex environment under different expectations. This fact requires any
MBSE environment to exhibit interoperability in a wide variety of software, workflow, and
environments. The "environment" would cover a wide variety of APIs to work in hybrid on-prem/
cloud configurations. The digital continuity layer brings simplification with standardization and
orchestration that enhance interoperability and flexibility to extend the MBSE. One of the enablers
for Digital Continuity is the orchestration of APIs. Standardizing digital threads with APIs and
message payloads with JSON can be performed.

As shown in the diagram above, one of the first components is **data gateway**. MBSE can
be extended into the field via **AWS for the Edge** technologies such as **AWS Outposts**, **AWS
Wavelength**, **AWS IoT** and **AWS Storage Gateway** and **Amazon SageMaker AI** that can bring
compute, storage, business logic, contend delivery, offline-mode or machine learning to the edge.

One of the applications would be **data ingestion**. There could be a wide variety of data sources
ingested. **Amazon Kinesis** would bring petabyte-scale data ingestion capability with real-time
processing power as part of digital continuity on the cloud.

If you would like to use an open-source platform for building real-time streaming data pipelines for your MBSE such as **Apache Kafka** but still would like to use AWS, you can consider using **Amazon MSK**. The other part of the digital continuity layer is the **ETL (Extract – Transform - Load)** operations.

AWS Glue can perform ETL in a serverless mode. You can create **AWS Glue Workflows** and end to end data-pipelines, encompassing other AWS services such as Amazon SageMaker AI or external endpoints including on-premises. The digital continuity would also cover existing data centers or design offices where the hybrid cloud environment can be supported by the Digital Continuity, such as via **AWS Storage Gateway**.

Moreover, as discussed before, you can leverage more than **230 global points of presence** of AWS to further decrease the latency and increase the reliability of the interactions. Technologies such as **Amazon CloudFront** and **AWS Global Accelerator** can take over the traffic from the public internet through these edge locations.

The simulations in MBSE are high-level (low-fidelity) and discreet. Moreover, the processes are transactional, rather than running continuously. For example, a change in a product design or a change in a requirement would correspond to an "event: change" with the message content of "change: metadata".

The model data can be standardized as well, following UML (Unified Modeling Language) compliance. Hence, the overall processes can be expressed in terms of events and short simulations based on transactions (messages and events) and attached metadata (body). Hence, your MBSE can transform model metadata into machine-readable lightweight interpretable representations. *Simplification* and *standardization* enable **event-driven architectures** for the MBSE.

You can use **Amazon EventBridge** as a serverless event-bus without using any code. Amazon EventBridge would be one of the fundamental parts of an event driven architecture that manages complexity through events chorography.

Along with Amazon EventBridge, **AWS Step Functions** or **AWS Managed Apache Airflow** for an open-source approach are AWS managed technologies to build orchestration and manage the complexity of MBSE and derivative workloads as the orchestration.

Digital Continuity will act as a "hub" location of orchestration MBSE and MBSE-related services, which would, otherwise, look like "point-to-point" as shown in the diagram below. However, this approach would not allow flexibility, such as any change in the "points" (services, APIs, edge-locations, etc.) would hinder the overall performance and availability of the solution. As a

result, every new service or replacement of an existing service ("point") would be plugged into
the **orchestrator**.
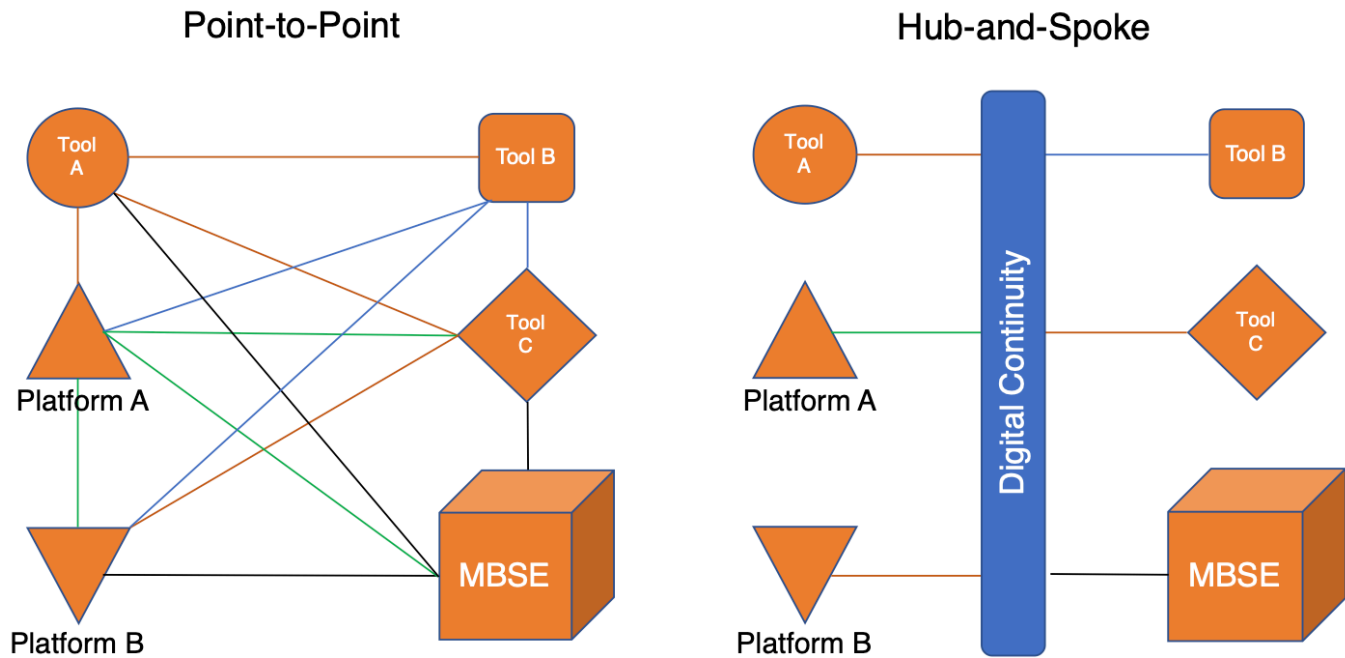
Orchestration and continuity of tools, locations and platforms via Digital Continuity.

Under the hood, **message broker/event orchestrator** acts as the single source - like a "hub" in the
hub-and-spoke architecture. The orchestrator is made of **messaging/queuing** and **translator**.

Message broker is required to distribute the messages to the recipients or conduct pub/sub
relationship in the software and API orchestration. **Amazon Simple Notification Service (Amazon
SNS)** and **Amazon Simple Queue Service (SQS)** provides managed serverless messaging and
queuing services, respectively, with no-administration required from you. If you would like to
employ a managed service but with an open source message broker such as Apache ActiveMQ
or RabbitMQ, you can choose **Amazon MQ**. The translator part is required to complete the
interoperability piece. Different types of applications and services require different formats.

Standardization comes with JSON, but effort is needed to build the translator for the digital
continuity. You can always work with your AWS Solutions Architect and other resources such as
AWS Partner Network (APN) and **AWS Professional Services** who can help you on tasks that
require hands-on development.

Extending Workforce Collaboration built on MBSE

Even though MBSE is slowly getting adopted by industries, existing engineering practices persist. In other words, engineers and practitioners may resist employing MBSE thoroughly. Of course, one of the reasons for this is the limitations of MBSE on interoperability and flexibility to adapt existing engineering workflows already being practiced and perfected by companies.

At this point, you can extend the activities as mentioned earlier to stage **engineering workflows**. People from different disciplines and different suppliers can cooperate in micro-level engineering transactions. For example, suppose the team would like to iterate on a design before committing anything to MBSE or PLM. In that case, they can perform local optimizations or "trial-and-errors" by using traditional ways of working.

Even though there are tools for almost every activity, they are not always used by all engineers, so this is the place where you can bring more macro to micro-level activities. Eventually, this will bring **automation** to repeatable processes, enable **transparency in daily activities**, employing single-source of data resources for those daily outputs **without worrying about cataloging them**.
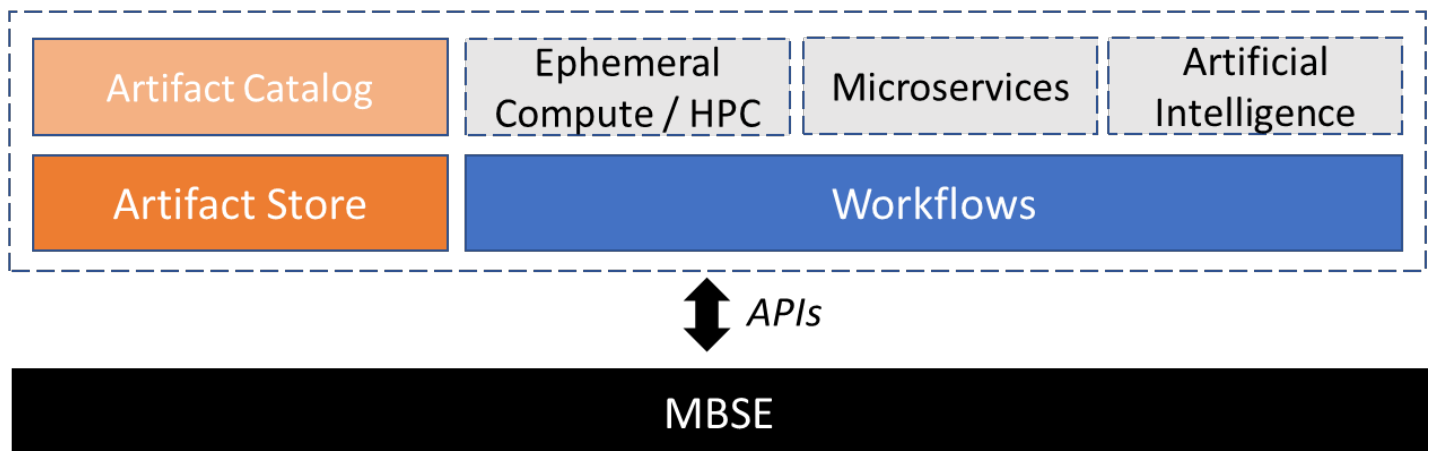
Workflow Management built on MBSE

The diagram above explains the concept of engineering collaboration based on MBSE. Here, MBSE is treated as the "single source of truth". Then, the **workflows** are a combination of **automation** and **human workflows**.

For example, if a team works on an FEA (finite element analysis) operation and asks for a supplier to quickly do a test, they can collaborate over an extension of workflow platform where an MBSE is

called when necessary. In this case, the team receives the requirements from the MBSE or simulate
the requirements through the MBSE, while performing their optimizations.

Meanwhile, the interim data (e.g. spreadsheets, presentations and documents) can be stored in
an **Artifact Store** (i.e. datalake). The diagram below offers examples of some "artifacts". AWS Glue
Crawler can then catalog these objects (artifacts) to form a data catalog that can be accessed
via **Amazon Athena**. Technologies such as **Amazon Textract** may be needed if these artifacts
include hand-written documents or "pictures" of tables.

One of the most important contributions to such an approach would be the insights derived
from old designs (and the analysis of their strengths and weaknesses). This concept is called
"**commonality**" in industries where the common parts and strategies (analysis, simulations
included) can be employed by other teams, future projects, and other departments. Thanks to the
artifact store working with MBSE ontology, commonality in design can be enhanced.

For example, a team of structural analysis engineers can query the stress results of an older
design to see how they can use the lessons learned from it. The overall workflow can be defined
using **AWS Step Functions** or **AWS Managed Apache Airflow**, if open-source is preferred. Please
note that you can specify the authorization and authentication in that process if you would like to
provide **granular access management and control** to engineers due to confidentiality issues.
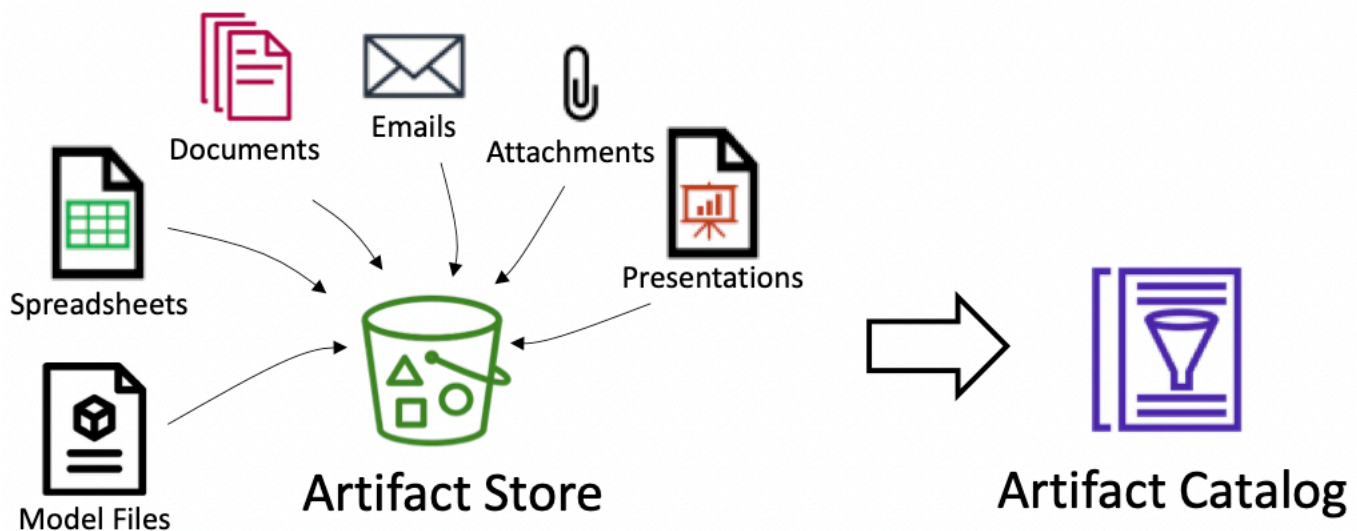
Artifact Store and Artifact Catalog

You can bring almost any workflow in this setting. For example, if the team requires an
automatically-deployed HPC cluster to perform the simulation, the deployment can be integrated

into this workflow. The results and the operation status can alert the engineering or relevant
stakeholders. The human workflow then kicks in and the operation continues like this.

The overall **visibility** of transactions, statuses and artifacts cataloged during the process can
increase the speed of product development.

Another application would be **auto-document (report) generation stemming from MBSE**. This
process can also be automated and can be further enhanced, especially using serverless computing.
This can bring more **agile practices** to waterfall types of engineering practices, such as aerospace
and automotive industries. Automation also improves the quality of work due to the reduction of
human errors in the processes.

The workflows can bear other AWS technologies and microservices. For example, suppose the team
would like to embed an automatic HPC simulation in batches, store the simulation results into an
Amazon S3 bucket (Artifact Store) and get cataloging automatically so that they can directly query
the results. Here the process can be performed automatically and on-demand.

Meanwhile, the HPC deployed to perform the simulation can be "dismantled," and you only pay for
the compute time you used. The cataloging part performed by AWS Glue has its serverless step-up
that also allows you to pay only the part being used (e.g. x minutes of cataloging).

The process is automated and **granulated so that you only pay for the activities being used**, and
the team **does not spend time on processes that do not create direct value for your business**.
The key to success in making such a process happen for you is to embed these scenarios using AWS
workflow management tools (e.g. AWS Step Functions, AWS managed Apache Airflow, etc.) and
AWS infrastructure stacks (e.g. Cloudformation, Terraform, etc.).

As discussed in the next section, NLP AI services such as Amazon Textract, Amazon Comprehend,
Amazon Translate and Amazon SageMaker AI with Hugging Face algorithms especially provide
value in the Artifact Store. You can put your old documents and the afore mentioned AI services
will help you to read the content. You can also create your own

The final part is to bring this monitoring and transparency to engineers and all stakeholders. Since
the overall process explained above is based on artifacts (i.e. objects, documents, files, etc.) and
metadata (events, messages, etc), visibility would be possible.

A **mobile application** using **AWS mobile technologies** that is asynchronous with the **activity
tracker** on the cloud would enable engineers to go faster. Moreover, if there is an issue
happening, **the engineering teams and management can react more quickly to the
problems** due to better visibility.

We mean "**visibility**" here as the **transparency in micro-processes**, for which you would define
the details based on your needs. We believe the "devil is in the details" in activities; therefore, we
recommend granular visibility. Yet, all of the components explained above are defined by you.

AI/ML Services for MBSE

MBSE is designed to be at the center of engineering operations. With that in mind, it offers data
about products, engineering activities, project activities, requirements, human resources, telemetry
data, and more during the months and years of operations. Hence, MBSE and the supporting
functions described in the previous sections can provide **unique insights about your products,
operations,** and **projects**.

You can use AWS embedded Analytics such as **Amazon Redshift** and **Amazon Quick Suite** to
monitor, observe and receive business intelligence and analytics.

Machine Learning and AI can deliver even more insights. These are the operational data that can
predict delivery dates, possible bottlenecks in the products and projects, and detection of potential
defects based on history.

- **Amazon SageMaker AI** to provide insights about anomaly detection in activities and products,
  pattern detection, and advanced machine learning applications.
- **Amazon Forecast** to perform forecasting using historical data. The activities in the MBSE and
  MBSE-related workloads would build up a historical dataset. This can be used to forecast delivery
  dates, timelines and estimation of engineering resources.
- **Amazon Neptune ML** to perform graph-based machine learning on your data ontology. Possible
  applications include pattern detection in ontology, patterns in design architectures, shortest-
  path and single-point-of-failure identification in processes.
- **Amazon Kendra** to extend data reach to the overall enterprise database on the hands of your
  engineers.
- **Amazon Lookout for Metric** to get AI-driven insights about metrics (any quantitative data)
  related to or correlated to each other. This is especially important to identify anomalies, edge
  case detection and pattern detection, and defects.

We especially see value in NLP technologies harnessed with data cataloging (such as AWS Glue) to
incorporate your legacy documents and build your own semantics.

- **Amazon Textract** extracts text, handwriting, and data from scanned documents beyond simple
  optical character recognition (OCR) to identify, understand, and extract data from forms and

tables. This is especially useful for scanning old design documents, reports, syntaxed drawings and pdf files, which can be a useful tool for your artifact store.

- **Amazon Comprehend** provides insight about your text. Amazon Comprehend Custom Entities can be used to create formal semantics and used in ontology.

- **Amazon Translate** can be used to translate text for 71+ languages. This service can be harnessed on the top of Amazon Comprehend and Amazon Textract to bring better collaboration in multi-language collaboration environments.

Shared Services Platform (SSP)

MBSE must have uniform standards to make it secure, compliant, and highly available. A centralized **Shared Service Platform (SSP)**, sometimes also referred to as a **Platform-as-a-Service (PaaS)**, can be used as a self-service interface streamlined for deploying code. You can also use Service Catalog in this section to make a PaaS available to your engineers.

With SSP, operators will have complete control on defining the standards on security, software delivery, monitoring, and networking related to MBSE and beyond. This allows your development teams to be more productive since they do not need to configure and manage the underlying cloud resources by themselves.

An SSP is typically built by your central IT team, or through **Cloud Center of Excellence (CCOE)** team or cloud infrastructure team. The SSP is composed of multiple AWS or open source products.

It typically supports multiple targets for running applications (e.g. Amazon EC2, containers, serverless), CI/CD pipelines, capturing logs/metrics, and security enforcement. The SSP packages these tools into a cohesive whole and makes them available to development teams via a simplified interface(typically a command line interface, graphical user interface or manifest file).

Conclusion

In this whitepaper, we first discussed MBSE as an industry-transforming technology by itself and highlighted the possible challenges with today's MBSE adoption and technology.

Then, we explained how AWS can help you with those challenges and provided possible technologies strategies to do so - from simple lift-and-shift to implementing add-on technologies such as digital continuity or AI-driven insights for your organization and projects. Finally, we extensively highlighted AWS services that will be part of such a solution customized for your business needs.

Contributors

Contributors to this document include:

- Dr. Burak Gozluklu, Solutions Architect, Amazon Web Services

Document history

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Minor update	Fixed broken link.	May 6, 2022
Initial publication	Whitepaper first published.	September 20, 2021

> ⓘ **Note**
>
> To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser that you are using.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.