# Abstract

This document details a system architecture designed to power self-learning Artificial Intelligence (AI) technology integrated with digital twins. The architecture focuses on core components such as data acquisition, middleware integration, the AI core engine, digital twin integration, and system monitoring, governance, and security. Mathematical formulations and code snippets are provided to illustrate key elements of the system. Additionally, the report outlines targeted Dassault products for integration, the rationale behind these choices, and details of the integration approach.

# 1. Introduction

Advances in AI are reshaping engineering and manufacturing, particularly through the use of digital twins. Our goal is to deploy a self-learning AI system that continuously evolves using real-time data and integrates seamlessly with digital twin simulations. This document outlines the architecture for achieving this vision, focusing on robust data pipelines, adaptive algorithms, and reliable integration. It also details our targeted Dassault products to maximize strategic value and operational efficiency and provides insights into Dassault’s work in Michigan.

# 2. System Architecture Overview

The system architecture is organized into the following layers:

1. Data Acquisition and Ingestion Layer
2. Integration and Middleware Layer
3. AI Core Engine
4. Digital Twin Integration
5. Monitoring, Governance, and Security

# 3. Detailed Architecture Components

## 3.1 Data Acquisition and Ingestion Layer

Data is gathered from various sources, processed, and fed into the AI engine.

Data Sources:

* Operational Sensors and IoT Devices: Provide real-time data.
* Historical Databases and Simulation Logs: Supply past data for training purposes.
* External Data Feeds: Enhance contextual data for robust decision-making.

Ingestion Pipeline:

* Data is ingested using API connectors and message brokers (e.g., Apache Kafka), ensuring reliable and real-time streaming.

## 3.2 Integration and Middleware Layer

This layer ensures seamless communication between internal systems and external platforms like Dassault’s digital twin environments.

Key Components:

* API Gateway: Exposes RESTful APIs to enable data exchange.
* Data Transformation Services: Convert data formats as needed.
* Security Measures: Enforce authentication (e.g., OAuth2, JWT) and data encryption both in transit and at rest.

## 3.3 AI Core Engine

At the heart of the system lies the self-learning AI engine, which continuously evolves from incoming data.

**Self-Learning Algorithm:**

The system leverages reinforcement learning techniques. A basic Q-learning update rule is as follows:

Where:

* is the learning rate
* is the discount factor
* is the reward for action in state
* is the subsequent state

## 3.4 Digital Twin Integration

The AI outputs are integrated into digital twin simulations that mirror the real-world environment.

Key Aspects:

* Simulation Environment: A digital replica that receives real-time updates from the AI engine
* Feedback Loop: Simulation outcomes are fed back into the AI for continuous model refinement
* Visualization: Interactive dashboards provide real-time insights and performance metrics

## 3.5 Monitoring, Governance and Security

This layer ensures system reliability and security.

Features:

* Operational Monitoring: Real-time dashboards to track system performance
* Audit Trails and Logging: Comprehensive logging of data interactions and AI decisions for compliance
* Access Management: Role-based access control (RBAC) secures sensitive data and functions

# 4. Dassault Product Integration

This section outlines the targeted Dassault products, the rationale behind these choices, and the integration approach:

**SIMULIA:**

* Why: Dassault’s advanced simulation solution can benefit from AI-enhanced predictive analytics to improve simulation accuracy
* How: Integration is achieved via RESTful APIs connecting the AI engine’s outputs with SIMULIA’s simulation data, enabling continuous refinement

**DELMIA:**

* Why: DELMIA, focused on manufacturing operations, can leverage our AI to offer real-time insights into production efficiency and predictive maintenance
* How: The integration leverages middleware that feeds operational data into the AI and returns actionable insights to optimize processes

**3DEXPERIENCE Platform:**

* Why: As an overarching ecosystem that integrates design, simulation, and manufacturing, embedding our AI here offers cross-functional data integration and a unified digital twin environment
* How: Using dedicated microservices and API gateways, AI outputs are embedded within the 3DEXPERIENCE interface

# 6. System Flow Diagram

Note: A visual diagram would represent the following flow:

1. Data Ingestion: Data is collected, preprocessed, and streamed to the AI engine.
2. Integration: Secure APIs and middleware route data to both the AI Core Engine and digital twin platforms.
3. AI Processing: The self-learning AI processes data, updates its model, and generates predictions.
4. Digital Twin Update: The simulation is refreshed with new predictions, mirroring the physical system in real time.
5. Monitoring: Dashboards and logging systems continuously track system health and performance.

# 7. Discussion/Analysis

The phased approach minimizes technical risks while maximizing integration value. Collaborative pilots with Dassault’s teams will help refine the system, ensuring it meets operational standards. The modular design allows for incremental improvements and scalability over time.

# 8. Conclusions and Recommendations

The proposed system architecture offers a scalable, secure, and effective solution for integrating self-learning AI with digital twins. The design aligns with current industry practices and enhances simulation accuracy, manufacturing efficiency, and real-time decision-making. Initiating a pilot project is recommended to validate integration approaches and further refine the system.

# 9. References

Apache Kafka Documentation. (n.d.). *Apache Kafka*. Retrieved April 25, 2025, from <https://kafka.apache.org/documentation>

Dassault Systèmes. (n.d.). *3DEXPERIENCE Platform*. Retrieved April 25, 2025, from <https://www.3ds.com/3dexperience/3dexperience-platform/>

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press

.

# 10. Appendix A: Detailed Code Listings



Figure 1: Example Python Code for Self-Learning Algorithm



Figure 2: Example Python Code for Ingestion Pipeline