David McNabb

david@aiapplied.uk

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

Document Title

Subtitle

Smart Manufacturing Data Hub (SMDH)

**Architecture Design Options**

Comprehensive Analysis of Four Architecture Approaches

|  |  |
| --- | --- |
| Document Version | 2.0 FINAL |
| Date | October 2025 |
| Status | For Review |
| Classification | Internal Use |
| Author | AI Applied Technical Team |
| Intended Audience | Technical Architects and Business Stakeholders |

# 1. Executive Summary

This document presents a comprehensive analysis of four distinct architectural approaches for the Smart Manufacturing Data Hub (SMDH) platform. The SMDH is designed as a cloud-native, multi-tenant Internet of Things (IoT) platform that empowers small and medium-sized manufacturing enterprises with real-time visibility into their operations.

## 1.1 Document Purpose

The purpose of this document is to provide both technical architects and business stakeholders with a clear understanding of four viable architecture options for the SMDH platform. Each option has been analysed in detail, considering factors such as cost implications, technical complexity, performance characteristics, scalability, implementation timelines, and operational considerations.

## 1.2 The Four Architecture Options at a Glance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Option A: Flink | Option B: Snowflake | Option C: SiteWise | Option D: Timestream |
| Monthly Cost (30 tenants) | £2,500-4,200 | £2,170-3,450 | £6,334 | £3,965 |
| Complexity | Very High | Low | Medium-High | Medium |
| Timeline | 24 weeks | 20 weeks | 28 weeks | 22 weeks |
| Best Suited For | Sub-second latency required | Most scenarios (Recommended) | SiteWise expertise available | AWS-native mandate |

## 1.3 Key Finding: All Options Meet Budget Requirements

An important finding from our analysis is that all four options meet the budget constraint of £200-300 per tenant per month. A previous cost calculation error for Option C has been corrected, reducing its estimated monthly cost by 70% (from £21,150 to £6,334 for 30 tenants). This means the decision should be based on factors such as technical requirements, team capabilities, and strategic preferences rather than basic affordability.

## 1.4 Recommended Approach

**🏆 Recommended: Option B (Snowflake-Leveraged Architecture)**

For the majority of manufacturing IoT scenarios, Option B (Snowflake) provides the best balance of:

* Lowest cost: £2,170-3,450/month (£72-115 per tenant)
* Lowest complexity: Only 5 core services to manage
* Fastest time-to-market: 20 weeks to production
* SQL-first development: Familiar to most engineering teams
* Native multi-tenancy: Secure row-level security built-in
* Proven in manufacturing: Used by major industrial companies

The default configuration provides 60-65 second dashboard update latency, which can be improved to <5 seconds for critical alerts by adding a Lambda "fast-path" (additional £100-150/month and 2 weeks implementation).

## 1.5 Alternative Options

Whilst Option B is recommended as the default choice, the other options may be preferable in specific circumstances:

**Option D (AWS-Native):** Select this if your organisation has a mandate for AWS-only services (no Snowflake). Provides excellent multi-tenancy, Grafana dashboards, and unified time-series storage. Cost: £3,965/month (15% premium over Option B).

**Option A (Flink-Based):** Select this ONLY if sub-second latency is legally or regulatorily required for all data processing. Provides advanced stream processing but requires Flink expertise and accepts highest complexity. Cost: £2,500-4,200/month.

**Option C (SiteWise):** Select this only if your team has deep AWS IoT SiteWise expertise and you're serving <10 tenants (multi-tenancy limitations). Provides managed asset modelling but requires custom dashboard development. Cost: £6,334/month (most expensive).

## 1.6 How to Use This Document

This document is structured to support both detailed technical review and high-level business decision-making:

* Technical Architects: Review Sections 4-7 for detailed component descriptions, data flows, and technical trade-offs for each option
* Business Stakeholders: Focus on Section 1 (this summary), Section 8 (comparisons), and Section 9 (recommendations)
* Decision-Makers: Review the comparison matrices in Section 8 and the decision framework in Section 8.5
* All Readers: Refer to Section 10 (Glossary) for explanations of technical terms. Terms are explained in plain English throughout the document.

Each architecture option (Sections 4-7) includes a placeholder for inserting the corresponding architecture diagram created in Draw.io. Please insert these diagrams before final distribution.

# Table of Contents

1. 1. Executive Summary

* 1.1 Document Purpose
* 1.2 The Four Architecture Options at a Glance
* 1.3 Key Finding: All Options Meet Budget Requirements
* 1.4 Recommended Approach
* 1.5 Alternative Options
* 1.6 How to Use This Document

1. 2. System Overview and Requirements

* 2.1 What is the Smart Manufacturing Data Hub?
* 2.2 Key System Requirements
* 2.3 Use Cases Supported
* 2.4 Budget and Constraints

1. 3. Architecture Option A: Flink-Based AWS-Heavy Architecture

* 3.1 Overview and Cost Summary
* 3.2 Architecture Diagram [PLACEHOLDER]
* 3.3 Detailed Component Description
* 3.4 Data Flow and Processing
* 3.5 Strengths
* 3.6 Limitations and Challenges
* 3.7 When to Choose This Option

1. 4. Architecture Option B: Snowflake-Leveraged Architecture (RECOMMENDED)

* 4.1 Overview and Cost Summary
* 4.2 Architecture Diagram [PLACEHOLDER]
* 4.3 Detailed Component Description
* 4.4 Data Flow and Processing
* 4.5 Strengths
* 4.6 Limitations and Considerations
* 4.7 When to Choose This Option

1. 5. Architecture Option C: AWS IoT SiteWise-Based Architecture

* 5.1 Overview and Cost Summary (CORRECTED)
* 5.2 Architecture Diagram [PLACEHOLDER]
* 5.3 Detailed Component Description
* 5.4 Data Flow and Processing
* 5.5 Strengths
* 5.6 Limitations and Challenges
* 5.7 When to Choose This Option

1. 6. Architecture Option D: Pure AWS-Native Architecture (Timestream + Grafana)

* 6.1 Overview and Cost Summary
* 6.2 Architecture Diagram [PLACEHOLDER]
* 6.3 Detailed Component Description
* 6.4 Data Flow and Processing
* 6.5 Strengths
* 6.6 Limitations and Considerations
* 6.7 When to Choose This Option

1. 7. Architecture Comparison

* 7.1 Cost Comparison
* 7.2 Complexity and Skills Required
* 7.3 Performance and Latency Characteristics
* 7.4 Multi-Tenancy Approaches
* 7.5 Decision Framework

1. 8. Recommendations

* 8.1 Primary Recommendation: Option B with Lambda Fast-Path
* 8.2 When to Consider Alternatives
* 8.3 Implementation Roadmap

1. 9. Glossary of Terms

# 2. System Overview and Requirements

[Insert detailed system overview from comprehensive documentation]

NOTE TO AUTHOR: This section should explain in plain English:  
• What the SMDH platform does and who it serves  
• Key technical requirements (data volume, latency, availability)  
• Multi-tenancy requirements and budget constraints  
• The four use cases: machine utilisation, air quality, energy monitoring, job tracking

# 3. Architecture Option A: Flink-Based AWS-Heavy Architecture

*[INSERT: SMDH-Option-A-Flink-Architecture.drawio diagram here]*

[Insert detailed Option A content from comprehensive documentation]

NOTE TO AUTHOR: This section should explain:  
• Why Flink? What makes this "AWS-Heavy"?  
• Each component in plain English with technical details  
• How data flows from sensors to dashboards  
• Strengths: Sub-second latency, advanced ML, unlimited flexibility  
• Limitations: Very high complexity, 24-week timeline, Flink expertise required  
• When to choose: Only if sub-second latency is legally mandated

# 4. Architecture Option B: Snowflake-Leveraged Architecture (RECOMMENDED)

*[INSERT: SMDH-Option-B-Snowflake-Architecture.drawio diagram here]*

[Insert detailed Option B content from comprehensive documentation]

NOTE TO AUTHOR: This section should explain:  
• Why Snowflake? What makes this the simplest approach?  
• How Snowpipe Streaming, Streams, and Tasks work  
• Native Row-Level Security for multi-tenancy  
• Strengths: Lowest cost, simplest, fastest timeline, SQL-first  
• Limitations: 60s dashboard latency (fixable with Lambda fast-path)  
• When to choose: Default choice for most scenarios  
• How to add Lambda fast-path for <5s alerts (£100-150/month, +2 weeks)

# 5. Architecture Option C: AWS IoT SiteWise-Based Architecture

*[INSERT: SMDH-Option-C-SiteWise-Architecture.drawio diagram here]*

[Insert detailed Option C content from comprehensive documentation]

NOTE TO AUTHOR: This section should explain:  
• What is SiteWise? Purpose-built for industrial IoT  
• Asset models, compute expressions, time-series storage  
• Why dual storage? (SiteWise + Timestream)  
• Cost correction: Was £21,150, now £6,334 (still most expensive)  
• Strengths: Managed asset modelling, OPC-UA/Modbus support  
• Limitations: NOT designed for multi-tenant SaaS, tag-based isolation risk, custom dashboards required  
• When to choose: Only if deep SiteWise expertise and <10 tenants

# 6. Architecture Option D: Pure AWS-Native Architecture (Timestream + Grafana)

*[INSERT: SMDH-Option-D-Timestream-Architecture.drawio diagram here]*

[Insert detailed Option D content from comprehensive documentation]

NOTE TO AUTHOR: This section should explain:  
• Why Timestream? Unified storage for time-series and events  
• Why Grafana? Best-in-class dashboards, fully white-labelable  
• Native partition-based multi-tenancy (better than SiteWise tags)  
• Strengths: Fully AWS-native, excellent dashboards, good multi-tenancy  
• Limitations: 15% more expensive than Option B, custom asset modelling needed  
• When to choose: If AWS-native is mandated (no Snowflake allowed)

# 7. Architecture Comparison

[Insert comparison matrices and decision framework]

NOTE TO AUTHOR: This section should include:  
• Cost comparison table (all 4 options)  
• Complexity ranking with explanations  
• Performance and latency comparison  
• Multi-tenancy security comparison  
• Decision tree: "If AWS-native required → Option D; If cost priority → Option B; etc."

# 8. Recommendations

## 8.1 Primary Recommendation: Option B with Lambda Fast-Path

For the Smart Manufacturing Data Hub platform, we recommend Option B (Snowflake-Leveraged Architecture) with the addition of a Lambda fast-path for critical alerts. This combination provides:

* Lowest total cost: £2,270-3,600/month including Lambda fast-path
* Lowest complexity: Only 5 core services plus Lambda alert handler
* Fastest time-to-market: 22 weeks (20 weeks baseline + 2 weeks for fast-path)
* Meets all requirements: Including <10 second alert latency after fast-path addition
* Lowest operational overhead: Minimal ongoing maintenance
* Best data governance: Unified platform simplifies compliance
* Proven at scale: Battle-tested by major manufacturers

## 8.2 When to Consider Alternative Options

**Choose Option D:** If your organisation requires AWS-native only (no Snowflake). Accept 15% cost premium for fully AWS-managed solution with excellent Grafana dashboards.

**Choose Option A:** ONLY if sub-second latency is legally or regulatorily mandated for all data (not just alerts). Accept very high complexity and 24-week timeline.

**Choose Option C:** ONLY if team has deep SiteWise expertise, serving <10 tenants, and can accept highest cost (£6,334/month). Multi-tenancy risk must be carefully mitigated.

# 9. Glossary of Terms

This glossary provides plain English explanations of technical terms used throughout this document.

**Apache Flink:** An advanced software framework for processing continuous streams of data in real-time. Used in Option A for sub-second data processing.

**AWS (Amazon Web Services):** Amazon's cloud computing platform providing on-demand services like storage, computing power, and databases.

**API (Application Programming Interface):** A way for different software applications to communicate with each other. Like a menu at a restaurant—defines what you can request and what you'll receive.

**Asset Model:** A template defining the structure and properties of industrial equipment in SiteWise. For example, "CNC Machine" asset model defines what data all CNC machines will provide.

**Latency:** The delay between when something happens (sensor reading) and when the system responds (dashboard update or alert sent). Measured in milliseconds (ms) or seconds.

**MQTT (Message Queuing Telemetry Transport):** A lightweight messaging protocol designed for IoT devices with limited bandwidth. Sensors use MQTT to send data to the cloud efficiently.

**Multi-Tenancy:** A software architecture where a single system serves multiple customers (tenants) whilst keeping their data completely isolated. Like an apartment building—shared infrastructure, private apartments.

**OPC-UA (Open Platform Communications Unified Architecture):** An industrial communication protocol widely used in manufacturing equipment. Allows machines to share data in a standardised format.

**Partition Key:** A database technique for organising data into separate sections. In multi-tenant systems, using tenant\_id as partition key physically separates each customer's data.

**Row-Level Security (RLS):** Database feature that automatically filters query results based on who is asking. Ensures users only see data they're authorised to access.

**Snowflake:** A cloud data platform (third-party service) that stores and analyses large amounts of data. Known for simplicity and SQL-based interface. Used in Option B.

**Snowpipe:** Snowflake's automated data loading service that continuously ingests new data as it arrives.

**SQL (Structured Query Language):** A standard language for interacting with databases. Used to retrieve, filter, and analyse data. Widely known and relatively easy to learn.

**Stream Processing:** Analysing data as it arrives in real-time, rather than storing it first and analysing later (batch processing).

**Time-Series Data:** Data points collected over time from sensors. Examples: temperature readings every minute, machine status every second.

**Timestream:** AWS's time-series database service, optimised for storing and querying sensor data with timestamps. Used in Options C and D.