Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

AI-Native Business Intelligence for Form-Driven Loan Origination Systems: A Cost-Efficient Architecture Using API Integration and Webhook-Driven Analytics

Sudheer Reddy Narra
Department of Information Technology
University of North Texas, Denton, Texas, USA
sudheerreddynarra@my.unt.edu

ARTICLE INFO

Received: 30 Dec 2024

Revised: 19 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Loan origination systems (LOS), including commercial platforms such as Encompass and custom enterprise implementations, are architecturally form-driven: their data is organized within rigid field-based schemas that are largely inaccessible to conventional business intelligence tooling without costly middleware, proprietary connectors, or enterprise BI licensing agreements. This paper proposes the AI-Native BI (ANBI) framework — a novel architecture for delivering analytics, dashboards, key performance indicators (KPIs), and export capabilities directly from LOS data without reliance on third-party BI platforms. The ANBI framework exposes LOS data through standard GET API endpoints and webhook integrations, applies a schema-agnostic field mapping and calculation normalization layer, and stores consolidated data in a single-table structure optimized for AI-assisted analytics development. AI coding tools including Cursor and Claude are used to implement the BI layer, significantly reducing development time and enabling continuous iterative improvement. The framework eliminates enterprise BI licensing costs — estimated at approximately $24,000 annually for Power BI at organizational scale — while delivering equivalent or superior analytical coverage through AI-native development patterns. The architecture draws on foundational work in AI-integrated BI systems (Pathoori, 2025) and extends it to the domain-specific constraints of regulated financial origination environments. Results demonstrate measurable reductions in implementation cost, time-to-dashboard, and operational dependency on proprietary analytics vendors.

Keywords: Loan Origination Systems, AI-Native BI, Business Intelligence Architecture, API Integration, Webhook Analytics, Encompass, LOS Data, KPI Dashboard, LLM-Assisted Development, BI Cost Optimization, Financial Technology Analytics, Field Mapping, Single-Table Architecture

1. INTRODUCTION

Loan origination systems represent one of the most data-rich and analytically underserved environments in financial services technology. Platforms such as Ellie Mae Encompass, BytePro, and custom enterprise LOS implementations collectively govern the mortgage application, underwriting, processing, and closing workflows for millions of loan transactions annually. Each of these platforms organizes its data within form-driven, field-based schemas — structured representations of paper loan documents that have been digitized but not fundamentally reconceived from an analytical perspective.

The analytical challenge this creates is well-documented in practice but underexplored in the academic literature. Conventional business intelligence deployments assume a relational database architecture with normalized tables, well-defined foreign keys, and a data model amenable to dimensional modeling or analytical query engines. LOS schemas frequently violate these assumptions: fields are named according to form conventions rather than analytical conventions, calculated values are stored as rendered strings rather than numeric types, and the same conceptual data element — loan amount, interest rate, closing date — may appear under different field identifiers across different LOS implementations or even across different loan products within the same platform.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The prevailing institutional response to this challenge has been to procure enterprise BI platforms — primarily Microsoft Power BI, Tableau, or Qlik — and invest in custom connector development to bridge LOS schemas to BI data models. This approach carries substantial costs: enterprise Power BI licensing at organizational scale approximates $24,000 annually before accounting for connector development, data warehouse infrastructure, and the ongoing engineering resources required to maintain schema mappings as LOS platforms evolve. For community banks, regional mortgage lenders, and independent mortgage brokers operating on constrained technology budgets, this cost profile is prohibitive.

The emergence of large language model (LLM)-assisted software development tools — including Cursor, GitHub Copilot, and Claude — creates a new architectural possibility that has not been systematically evaluated in the peer-reviewed literature: the construction of complete, production-quality BI layers through AI-native development patterns that eliminate the dependency on third-party BI platforms entirely. Pathoori (2025) demonstrated that AI integration with enterprise BI infrastructure through Retrieval-Augmented Generation fundamentally changes the analytical capability profile of BI systems, enabling natural language querying and context-aware analytics that static dashboards cannot provide. The present work extends this architectural insight to a different but related problem: rather than augmenting an existing BI layer with AI, we propose using AI development tools to construct the BI layer from scratch, directly against LOS API endpoints, in a manner that makes the resulting system inherently AI-native.

This paper makes three primary contributions. First, we propose the AI-Native BI (ANBI) framework — a five-layer architecture for connecting LOS platforms to a self-contained analytics environment through API and webhook integration. Second, we describe the schema-agnostic field mapping and normalization layer that enables the ANBI framework to operate across heterogeneous LOS implementations without platform-specific customization at the BI layer. Third, we report implementation results demonstrating the cost, time, and operational advantages of the ANBI approach relative to conventional enterprise BI deployments in mortgage origination environments.

The remainder of this paper is organized as follows: Section 2 reviews relevant literature on LOS analytics, BI architecture, and AI-assisted development. Section 3 describes the ANBI framework in detail. Section 4 presents the implementation methodology. Section 5 reports results. Section 6 discusses findings and concludes.

2. LITERATURE REVIEW

2.1 Business Intelligence in Financial Services

The application of business intelligence to financial services has been extensively studied. Chaudhuri et al. (2011) described the fundamental architecture of BI systems as encompassing data warehousing, reporting, and analytical query processing, establishing the foundational framework within which most enterprise BI deployments still operate. In the lending and mortgage domain, BI has historically focused on pipeline management, pull-through rate analysis, and compliance reporting — analytical use cases that are well-suited to the batch reporting paradigms of conventional BI architecture.

However, the field-based schema architecture of LOS platforms creates integration challenges that conventional BI approaches do not address directly. Kimball and Ross (2013) established the dimensional modeling framework that underlies most enterprise data warehouse design, but the LOS field schema — with its form-driven naming conventions, mixed data types, and calculated string fields — does not map naturally to fact-dimension architectures without substantial data engineering effort. The integration gap between LOS data models and BI query models represents a persistent cost driver that has not been systematically resolved in either commercial products or academic literature.

2.2 API-Driven Data Integration

The emergence of RESTful API standards in enterprise software platforms has created new possibilities for data integration that bypass traditional ETL and data warehouse architectures. Fielding (2000) established the Representational State Transfer (REST) architectural principles that now underpin most enterprise API design, enabling programmatic data access through standard HTTP verbs and structured response payloads. Modern LOS platforms, including Encompass, expose their data through REST APIs that support field-level GET requests and

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

event-driven webhook notifications — capabilities that create the technical foundation for the ANBI architecture proposed in this paper.

Webhook-based data synchronization — in which platform events trigger outbound POST notifications to registered endpoints — represents a significant advancement over polling-based API integration patterns. Rather than querying a source system at fixed intervals to detect state changes, webhook architectures allow the source system to push change notifications in real time, eliminating latency and reducing API call volume. In the LOS context, loan status changes, document receipt events, and underwriting decisions all generate webhook notifications that can be used to maintain a current analytical data store without continuous polling.

2.3 AI-Integrated Business Intelligence

The integration of artificial intelligence with business intelligence infrastructure has emerged as an active research area over the past several years. Early work focused on natural language query interfaces that allowed non-technical users to interact with BI systems through conversational language rather than structured query syntax (Affolter et al., 2019). More recent work has examined the integration of large language models with enterprise data systems at the architectural level.

Pathoori (2025) introduced the RAG-BI framework — a Retrieval-Augmented Generation architecture for enterprise BI platforms that enables natural language querying of live enterprise data through LLM integration. This work established that the fundamental architecture of enterprise BI could be reconceived around AI capabilities rather than treating AI as an add-on to existing dashboard infrastructure. The RAG-BI framework demonstrated that context-aware analytics — in which the AI system retrieves relevant data in response to natural language queries and synthesizes actionable responses — represented a genuine paradigm shift from passive data visualization to active decision intelligence. The ANBI framework proposed in the present paper builds on this architectural insight, applying AI-native development patterns to the construction of a BI layer rather than to the querying of an existing one.

The use of LLM-based coding tools in software development has also received growing research attention. Chen et al. (2021) introduced Codex, demonstrating that large language models pre-trained on code corpora could generate functional code from natural language descriptions with high accuracy across a range of programming tasks. Subsequent tools including GitHub Copilot, Cursor, and Anthropic's Claude have operationalized these capabilities in production development environments, enabling rapid generation of dashboard components, API integration code, and data transformation logic from natural language specifications.

2.4 Cost Optimization in Enterprise Analytics

The cost profile of enterprise BI deployment has received attention in both practitioner and academic literature. Gartner (2023) reported that enterprise BI platform licensing represents a substantial and growing line item in technology budgets, with total cost of ownership frequently exceeding license fees when integration, administration, and training costs are included. For smaller financial institutions — community banks, credit unions, and independent mortgage lenders — enterprise BI licensing costs represent a proportionally larger budget impact than at larger institutions, creating a systematic analytical disadvantage relative to well-capitalized competitors.

The potential for open-source analytics frameworks and custom development to close this gap has been discussed in practitioner contexts, but peer-reviewed research on AI-assisted development as a cost reduction mechanism in BI deployment is limited. The present work addresses this gap by providing a systematic architectural framework and quantitative implementation analysis for AI-native BI development in the mortgage origination domain.

3. THE AI-NATIVE BI (ANBI) FRAMEWORK

3.1 Framework Overview

The ANBI framework consists of five integrated layers: the LOS Integration Layer, the Field Normalization and Mapping Layer, the Single-Table Data Store, the AI-Native Analytics Layer, and the Export and Distribution Layer.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Each layer is designed to operate independently of third-party BI platform licensing while maintaining the analytical coverage — KPIs, dashboards, trend analysis, and export capabilities — that mortgage origination operations require.

3.2 LOS Integration Layer

The LOS Integration Layer establishes the data connection between the source LOS platform and the ANBI data environment through two complementary mechanisms: scheduled GET API calls for historical data loading and webhook subscriptions for real-time event-driven updates.

Historical data loading uses the LOS platform's REST API to retrieve loan records in configurable batch sizes. The GET request specifies the field set required for analytical coverage — loan amount, product type, origination date, loan officer identifier, status code, calculated rate, and applicable compliance fields — and the response payload is parsed and loaded into the Single-Table Data Store. For Encompass implementations, this corresponds to the Loan Pipeline API endpoint with field projection parameters. For custom LOS implementations, equivalent functionality is available through standard REST query endpoints.

Real-time synchronization uses webhook subscriptions registered against LOS loan lifecycle events. When a loan status changes — from application to processing, processing to underwriting, underwriting to conditional approval, and through to closing or denial — the LOS platform posts a webhook notification to the ANBI endpoint. The webhook handler extracts the relevant field values from the notification payload, applies the normalization logic described in Section 3.3, and upserts the loan record in the Single-Table Data Store. This architecture ensures that the analytical environment reflects current loan pipeline status within the webhook delivery latency of the source platform, typically under thirty seconds.

3.3 Field Normalization and Mapping Layer

The Field Normalization and Mapping Layer is the architectural element that enables the ANBI framework to operate across heterogeneous LOS implementations. Different LOS platforms use different field naming conventions for conceptually identical data elements. Encompass refers to the loan amount as field 1109; a custom LOS implementation may use loan_amount, loanAmt, or principal_balance for the same value. Without a normalization layer, the analytics code would require platform-specific implementations for every calculated metric and KPI.

The normalization layer implements a JSON-based mapping configuration that defines a canonical field schema — a set of standardized field names and data types that represent all analytically relevant LOS data elements — and maps each canonical field to its corresponding source field identifier in each supported LOS implementation. The mapping configuration is loaded at initialization and applied during both historical data loading and webhook event processing, ensuring that all data entering the Single-Table Data Store conforms to the canonical schema regardless of source platform.

Calculated fields — such as days-in-process, rate spread, loan-to-value ratio, and debt-to-income ratio — are standardized through a calculation normalization layer that evaluates the canonical field values and produces derived metrics in a consistent format. This separation of source field mapping from calculated metric derivation ensures that changes to LOS field schemas require updates only to the mapping configuration, not to the analytical logic layer above it.

3.4 Single-Table Data Store

The ANBI framework stores all normalized LOS data in a single flat table — a deliberate architectural choice that diverges from dimensional modeling conventions but is optimized for the specific requirements of AI-native analytics development. The single-table structure enables LLM-based coding tools to generate accurate query logic without requiring awareness of join relationships, foreign key constraints, or dimensional schema conventions that introduce complexity in AI code generation contexts.

Each row in the Single-Table Data Store represents a loan record at a point in time, with columns corresponding to all canonical fields in the normalization layer plus system metadata fields including ingestion timestamp, source platform identifier, and data freshness indicator. For analytical use cases requiring historical trend analysis, a

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

separate event log table records all status change events with timestamps, enabling time-series queries without denormalizing the primary loan record.

3.5 AI-Native Analytics Layer

The AI-Native Analytics Layer implements the KPI calculations, dashboard visualizations, and analytical logic that constitute the BI deliverables of the ANBI framework. This layer is implemented using AI coding tools — specifically Cursor and Claude — which generate the dashboard component code, KPI calculation logic, and visualization rendering from natural language specifications provided by the analyst or developer.

The use of AI coding tools at this layer produces several structural advantages. First, development velocity is substantially higher than conventional dashboard development: a KPI card displaying pull-through rate by loan officer, calculated from the Single-Table Data Store with drill-down capability, can be specified in natural language and generated as functional code in a matter of minutes rather than hours. Second, the generated code is inherently modifiable through natural language iteration — the analyst can request changes to the calculation methodology, visualization type, or filter behavior without writing code directly. Third, the AI development pattern scales naturally: adding new KPIs, new dashboard tabs, or new export formats requires the same AI-assisted specification process regardless of the complexity of the addition.

3.6 Export and Distribution Layer

The Export and Distribution Layer provides Excel export capability for all dashboard views and KPI tables in the ANBI framework. Excel export is implemented through a server-side generation function that queries the Single-Table Data Store with the same filter parameters applied to the dashboard view and writes the results to a formatted xlsx file using standard open-source libraries. This capability addresses the reporting workflow requirement that mortgage origination operations frequently have for compliance documentation, investor reporting, and executive distribution — requirements that dashboard visualization alone does not satisfy.

4. IMPLEMENTATION METHODOLOGY

The ANBI framework was implemented and evaluated in a pilot deployment at a mid-sized independent mortgage lender operating an Encompass LOS environment with approximately 850 active loans in pipeline at any given time and a team of twelve loan officers. The implementation proceeded in four phases: API integration and data loading, field mapping configuration, AI-native dashboard development, and operational validation.

Phase one established the GET API connection to the Encompass Loan Pipeline endpoint and configured the webhook subscription for loan status change events. The API integration code was generated using Cursor with natural language specifications describing the required field set and response parsing logic. The historical data load retrieved 2,400 closed loan records spanning eighteen months of origination activity, loading them into the Single-Table Data Store in approximately forty minutes through batched API calls.

Phase two developed the field mapping configuration for the Encompass field schema, mapping 47 canonical ANBI fields to their corresponding Encompass field identifiers. Calculated fields — including days-to-close, pull-through rate, average loan amount by product type, and lock expiration risk indicator — were defined in the calculation normalization layer using the canonical field names.

Phase three used Claude as the primary AI coding tool to generate the analytics layer. Dashboard components including pipeline summary, loan officer performance scorecard, product mix visualization, pull-through funnel, and compliance status tracker were specified in natural language and generated as functional React components consuming the Single-Table Data Store through a lightweight API layer. The complete dashboard implementation required approximately 14 hours of AI-assisted development across three days.

Phase four conducted operational validation over a thirty-day period in which the ANBI dashboard was used in parallel with the organization's existing Power BI deployment. Analytical coverage, data accuracy, and user workflow satisfaction were assessed through structured evaluation against the existing Power BI implementation.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5. RESULTS AND DISCUSSION

5.1 Cost Analysis

The primary cost advantage of the ANBI framework relative to the existing Power BI deployment was confirmed through direct cost comparison. The organization's Power BI licensing cost totaled $23,760 annually across the analytics user population. The ANBI implementation incurred no ongoing licensing cost. The one-time AI-assisted development cost, calculated at an estimated $85 per hour for development time, totaled approximately $1,190 for the 14-hour implementation. The ANBI framework achieves full licensing cost recovery within three weeks of deployment, with zero ongoing licensing cost thereafter.

Additional cost reductions were identified in connector maintenance and data engineering overhead. The existing Power BI deployment required a dedicated ETL process maintained by an external consultant at a cost of approximately $6,000 annually. The ANBI webhook architecture eliminates this cost entirely by maintaining data currency through event-driven updates without a scheduled ETL pipeline.

5.2 Analytical Coverage

The thirty-day parallel operation evaluation confirmed that the ANBI dashboard provided equivalent analytical coverage to the existing Power BI deployment across all primary use cases: pipeline status monitoring, loan officer performance tracking, product mix analysis, and pull-through rate calculation. Three additional analytical views were implemented during the validation period at the request of loan officers — rate lock expiration alerts, processing bottleneck identification, and investor commitment utilization tracking — each requiring less than two hours of AI-assisted development.

Data accuracy validation against the Encompass source system confirmed a 99.3% field-level accuracy rate across the normalized data store, with the 0.7% discrepancy rate attributable to a subset of calculated string fields in the Encompass schema that required additional parsing logic in the normalization layer. This issue was resolved through a mapping configuration update without changes to the analytics layer code.

5.3 Development Velocity and Scalability

The AI-assisted development pattern demonstrated substantially higher velocity than conventional dashboard development. The fourteen-hour implementation timeline for a complete five-dashboard analytics environment compares favorably with industry benchmark estimates of 80-120 hours for equivalent Power BI implementations inclusive of data modeling, connector configuration, and dashboard development. The AI-native approach achieved a 6-8x velocity improvement, attributable to the elimination of data modeling complexity through the single-table architecture and the natural language specification capability of the AI coding tools.

The scalability advantage of the ANBI architecture was demonstrated during the validation period when three new dashboard components were added in response to user requests. Each addition required natural language specification and AI-assisted code generation without modifications to the underlying data architecture, confirming that the single-table design and AI development pattern scale well to incremental analytical requirements.

5.4 Implications for Mortgage Technology

The ANBI framework has implications beyond cost reduction for the mortgage origination technology sector. The systematic disadvantage that smaller lenders face in analytical capability relative to large institutional lenders is driven substantially by the cost profile of enterprise BI deployment. By demonstrating that AI-native development patterns can deliver equivalent analytical coverage at a fraction of the licensing and development cost, the ANBI framework establishes a pathway for analytical capability democratization in the mortgage origination domain.

The architectural principles of the ANBI framework — API-driven data access, schema-agnostic normalization, single-table analytical storage, and AI-assisted development — are not specific to Encompass or to mortgage origination. They apply equally to any form-driven enterprise system with REST API access, suggesting that the ANBI pattern has broader applicability across regulated industry verticals including healthcare claims processing, commercial insurance underwriting, and public sector grant management.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6. CONCLUSION

This paper introduced the AI-Native BI (ANBI) framework — a five-layer architecture for delivering complete business intelligence capabilities against loan origination system data through API integration, webhook synchronization, and AI-assisted analytics development, without dependency on third-party BI platform licensing. The framework addresses a persistent and underexplored challenge in mortgage origination technology: the analytical inaccessibility of form-driven LOS schemas to conventional BI tooling, and the prohibitive cost of enterprise BI licensing for smaller lending organizations.

The ANBI framework demonstrated a full annual licensing cost elimination of approximately $24,000, a 6-8x development velocity improvement over conventional BI implementation, and equivalent analytical coverage across all primary mortgage origination reporting use cases. The field normalization and mapping layer enables the framework to operate across heterogeneous LOS implementations, and the AI-native development pattern provides a scalable and low-cost mechanism for iterative analytical expansion.

Future work should pursue empirical validation across multiple LOS platforms and lender sizes, comparative analysis of AI coding tool performance in this architectural context, and extension of the ANBI pattern to include natural language querying capabilities consistent with the RAG-BI architectural advances documented by Pathoori (2025). Integration of the ANBI data layer with conversational analytics interfaces — enabling loan officers and managers to query pipeline status and performance metrics in natural language — represents a natural and achievable next step that would further reduce the analytical skill barrier in mortgage origination operations.

REFERENCES

[1] Affolter, K., Stockinger, K., & Bernstein, A. (2019). A comparative survey of recent natural language interfaces for databases. The VLDB Journal, 28(5), 793-819. https://doi.org/10.1007/s00778-019-00567-8

[2] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., & Zaremba, W. (2021). Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.

[3] Chaudhuri, S., Dayal, U., & Narasayya, V. (2011). An overview of business intelligence technology. Communications of the ACM, 54(8), 88-98. https://doi.org/10.1145/1978542.1978562

[4] Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation, University of California, Irvine).

[5] Gartner. (2023). Magic quadrant for analytics and business intelligence platforms. Gartner Research.

[6] Kimball, R., & Ross, M. (2013). The data warehouse toolkit: The definitive guide to dimensional modeling (3rd ed.). Wiley.

[7] Pathoori, M. R. (2025). Retrieval-augmented dashboards: Enabling context-aware analytics through LLM integration with BI platforms. Journal of Information Systems Engineering and Management, 10(60s). https://doi.org/10.52783/jisem.v10i60s.13128

[8] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J. Y., & Wen, J. R. (2023). A survey of large language models. arXiv preprint arXiv:2303.18223.