Journal of Information Systems Engineering and Management
2018, 3(1)

e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Practical Reference Architecture for Data Consolidation in Multi-Domain MDM Using EBX5 Technical Research Article - Spring 2018

Kinshuk Dutta
Principal Solutions Engineer
Orchestra Networks, New York, USA
Email: kinshuk.dutta@orchestranetworks.com; dutta.kinshuk@gmail.com
Corresponding Author: Kinshuk Dutta

ARTICLE INFO

Received: 01 Feb 2018

Revised: 18 Mar 2018

Accepted:28 Mar 2018

ABSTRACT

Enterprise master data management programs frequently fail not because master data concepts are misunderstood, but because implementation teams lack a repeatable architecture for moving heterogeneous source data through profiling, validation, standardization, matching, stewardship, and publication. This paper presents a practical reference architecture for data consolidation using ON EBX5. The architecture organizes consolidation into four primary zones: pre-landing, landing, staging, and mastering. It also defines the operational roles, transformation responsibilities, data quality checkpoints, matching and survivorship patterns, stewardship interventions, and publishing mechanisms required to create governed golden records from multi-source enterprise data. The reference model is intentionally implementation-oriented: it is designed for professional services teams, solution architects, data stewards, and business stakeholders who must convert complex customer data landscapes into auditable, repeatable, and scalable master data programs.

Keywords: Master Data Management; EBX5; data consolidation; data governance; data quality; stewardship; survivorship; golden record; data exchange; matching; MDM architecture.

I. Introduction

Master data consolidation is the process of collecting records from multiple source systems, preserving provenance, standardizing data, identifying duplicate or related entities, and producing an enterprise-approved representation of a business object. In multi-domain MDM programs, this process becomes significantly more complex because the data is rarely homogeneous. Source systems differ in structure, terminology, quality, operational ownership, trust level, and lifecycle state.

ON EBX provides a model-driven platform for governing master, reference, and metadata. However, successful data consolidation requires more than a configured data model. It requires an implementation pattern that separates raw capture from source alignment, separates source alignment from transformation, separates transformation from matching, and separates automated matching from human stewardship. The reference architecture described in this paper was developed to address that practical implementation challenge.

The objective of this paper is to define a repeatable consolidation architecture for EBX5 implementations that supports source-system onboarding, controlled data movement, transparent quality checks, stewardship-driven exception handling, and auditable golden-record creation. The design is intentionally pragmatic: it favors implementation repeatability, traceability, and governance over theoretical purity.

Copyright © 2018 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

e-ISSN: 2468-4376
https://www.jisem-journal.com/

II. Problem Statement

In many enterprise environments, master data is fragmented across customer relationship management systems, enterprise resource planning systems, legacy applications, operational databases, spreadsheets, and externally supplied reference feeds. These systems may represent the same party, product, organization, location, or reference value differently. Even when the business entity is the same, the record shape, identifier scheme, attribute completeness, language, character normalization, and address or contact representation may vary substantially.

A direct load from source systems into master tables creates operational risk. It can pollute governed data with incomplete or unverified values, make duplicate detection harder, hide provenance, and force data stewards to resolve issues without context. Conversely, a heavily customized integration layer can become brittle and difficult to audit. The reference architecture therefore introduces a zone-based pattern that provides controlled separation of concerns.

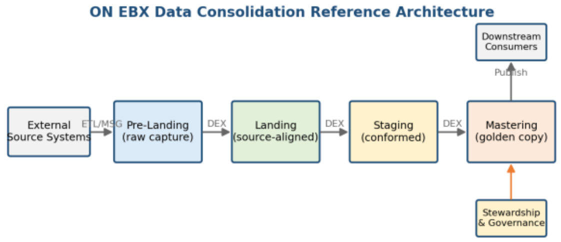

Fig. 1. ON EBX zone-based consolidation reference architecture.

III. Architectural Principles

The proposed architecture is based on seven principles.

First, source fidelity must be preserved at the point of ingestion. The initial capture of source data should reflect the structure and values received from upstream systems as closely as possible. This allows auditability, replay, and source-level diagnostics.

Second, transformation should be staged and explainable. Data should not be transformed directly into mastered structures without intermediate validation and mapping. Each transformation should have a clear business or technical purpose.

Third, matching and survivorship must be governed, not hidden. Automated matching can reduce manual effort, but stewardship is required when scores fall into ambiguous ranges or when regulatory and business consequences require human approval.

Fourth, data quality is not a single operation. Profiling, validation, cleansing, standardization, matching, and survivorship are separate activities that must be coordinated across the consolidation lifecycle.

Fifth, the architecture must support multiple implementation styles. Some customers use EBX as a golden copy while source authoring remains external. Others use EBX as a golden source where records are authored and governed centrally. The zone model supports both styles.

Sixth, source-system onboarding must be repeatable. A new source should be evaluated, profiled, mapped, tested, and approved through a standard procedure rather than handled as an isolated custom integration.

Seventh, business and IT responsibilities must be explicit. Data stewards, architects, administrators, developers, and business users participate in different stages of the consolidation lifecycle; the architecture must make these responsibilities visible.

Copyright © 2018 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

IV. Primary Data Zones

The architecture uses four primary data zones. Each zone has a distinct purpose and is represented in EBX through dedicated data spaces, datasets, tables, and workflow controls where appropriate.

A. Pre-Landing Zone

The pre-landing zone captures data as received from external systems. It is intentionally close to raw source format and is useful for initial profiling, completeness checks, and source-specific diagnostics. In many implementations, the pre-landing zone may be optional; however, it is valuable when external feeds are highly variable or when a customer requires auditability of the originally received payload.

B. Landing Zone

The landing zone provides a standardized structure that receives source data while preserving source identity. It may use a common union structure across sources or maintain separate landing formats for different source families. The landing zone is the first point where data can be consistently validated within EBX while still retaining strong alignment with the source representation.

C. Staging Zone

The staging zone performs conformance and preparation. Reference value translation, structural normalization, type conversion, cleansing, validation, and match-preparation logic are typically applied here. Staging tables may resemble landing tables, master tables, or a hybrid structure depending on the transformation strategy. The primary design goal is to make the final transition into mastered records as deterministic and transparent as possible.

D. Mastering Zone

The mastering zone contains the governed representation of the entity. This is where deduplication, cross-system matching, survivorship, golden-record creation, stewardship review, and publication control are applied. The mastered record may be consumed by downstream applications or used as the authoritative representation for enterprise governance.

V. Operational Consolidation Pipeline

The zone architecture is implemented through an operational pipeline. A source feed is loaded into the appropriate raw or landing structures, validated, profiled, transformed, matched, reviewed, and finally promoted into the mastered representation. The pipeline allows errors to be identified early and routed to either technical correction or business stewardship.

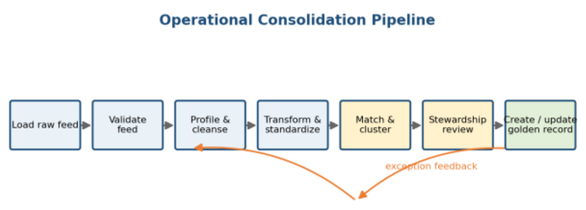

Fig. 2. Operational pipeline from raw feed to governed golden record.

The pipeline supports both automated and manual checkpoints. System tasks can load files, create draft areas, run validation, execute data exchange mappings, generate profiling summaries, and run matching policies. User tasks are required when the data requires interpretation, when a source system must correct a rejected feed, or when a steward must decide whether records should be merged, excluded, or retained as separate entities.

Copyright © 2018 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2018, 3(1)

e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

VI. Data Exchange And Transformation Strategy

Data movement between zones is typically configured using EBX Data Exchange patterns. The transformation layer should be treated as a governed mapping model rather than a collection of isolated scripts. Source and target applications, data models, semantic models, table mappings, and field mappings together define how data moves through the pipeline.

Typical transformation operations include concatenating first and last names into a full-name field, translating prefixes through cross-reference lookup tables, converting string dates into typed dates, splitting addresses into atomic components, and converting source-specific flags into Boolean or enumerated values. These transformations should be documented and attached to the implementation model because they directly affect match quality and downstream trust.

VII. Data Quality And Profiling

Profiling should occur before records are promoted into mastered structures. It helps determine whether a source feed is structurally valid, whether fields are complete enough for matching, whether candidate keys exist, whether values require standardization, and whether the source contains systematic data quality issues. Profiling results should be drillable so that stewards and implementation teams can inspect field-level distributions, distinctness, completeness, and duplicate indicators.

Data quality is also a governance feedback loop. When a source consistently produces invalid values, the solution should not simply compensate downstream. The issue should be surfaced to process data stewards or source owners so that upstream correction can occur where possible.

VIII. Matching, Survivorship, And Stewardship

The mastering zone applies matching policies to determine whether incoming records correspond to existing mastered entities. Matching may use exact comparison, fuzzy full-text comparison, weighted field rules, relation-aware matching, or context-specific policies. Match configuration should reflect both technical similarity and business semantics.

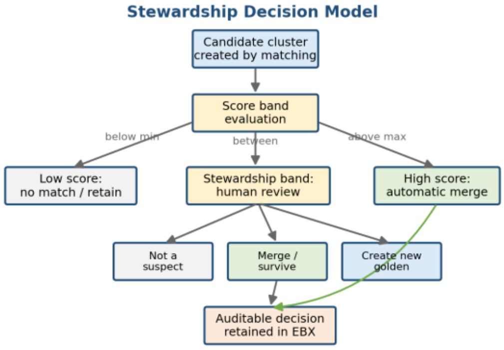

Fig. 3. Stewardship decision model for match-score evaluation.

A common pattern is to divide match outcomes into three categories. Low-confidence candidates are not treated as matches. High-confidence candidates may be automatically merged where business policy permits. Ambiguous candidates are routed to stewardship. The stewardship band is especially important because false positives in master data can be more damaging than false negatives; an incorrect merge can corrupt the enterprise view of a customer, supplier, product, or location.

Stewards must be able to inspect clustered records, compare attributes, determine whether records are incorrectly grouped, mark records as not suspects, remove records from clusters, add records into clusters, and select survivorship values. The design must preserve auditability so that human decisions become part of the governed record history rather than informal side effects.

Copyright © 2018 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2018, 3(1)

e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

IX. Party Data Design Considerations

Party data consolidation illustrates the importance of context. Individuals and organizations require different matching logic. A person may be matched using name, address, communication, and date-of-birth attributes. An organization may require legal name, trade name, registered address, tax identifier, domain, or other organization-specific attributes. A single generic party matching policy can therefore produce weak results unless it is constrained by party type or context.

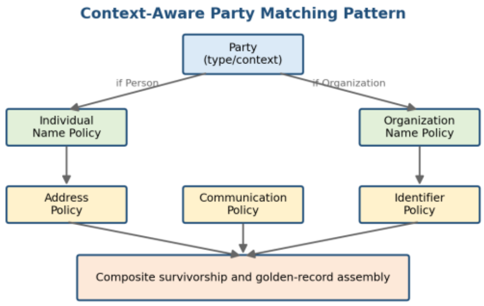

Fig. 4. Context-aware party matching pattern with related child policies.

The recommended design is to configure matching policies bottom-up. Child entities such as names, addresses, communications, and identifiers should have their own matching logic where appropriate. The parent party table can then use relation-based matching to incorporate child-table results. Context constraints, such as person versus organization, prevent policies from comparing semantically incompatible records.

X. Governance Roles And Responsibilities

The reference architecture requires explicit role separation. Architects design the data model, zone structure, match policies, workflow boundaries, and integration patterns. Data stewards review quality issues, suspect clusters, survivorship outcomes, and source-system anomalies. Steward managers monitor throughput, escalation, and quality performance. System administrators operate environments, jobs, security, and operational controls. Business users consume and maintain governed data according to permissions and business process design.

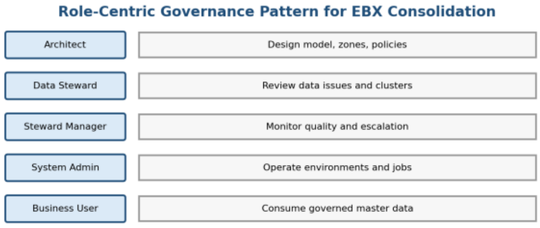

Fig. 5. Role-centric governance model for EBX consolidation programs.

Role	Primary responsibility in consolidation architecture
Architect	Designs data zones, master model, policies, permissions, workflows, and integration architecture.
Data Steward	Reviews validation issues,

Copyright © 2018 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

	profiling anomalies, suspect records, merge decisions, and survivorship values.
Data Steward Manager	Manages stewardship workload, escalation, data quality goals, and operating procedures.
System Administrator	Maintains EBX environments, deployment configuration, schedules, security integration, and operational monitoring.
Business User	Uses governed master data and participates in approval or correction workflows as configured.

TABLE I. Representative role allocation for ON EBX consolidation programs.

XI. Source-System Onboarding

A scalable consolidation solution must include a controlled process for accepting new source systems. Source onboarding should begin with provider validation: the source file or interface is inspected, a sample model is generated where useful, a sample dataset is loaded, profiling is executed, and candidate identifiers or matching fields are reviewed. The outcome of this process determines whether the source can be accepted as-is, accepted with transformations, or rejected pending upstream correction.

The onboarding process should also include system modification analysis. If a new source introduces new attributes, new reference values, new hierarchy structures, or new source-specific identifiers, the implementation team must assess whether the master model, staging structure, mapping configuration, validation rules, or match policies require change. This prevents uncontrolled schema drift and protects the governed master model from source-specific pollution.

XII. Iterative Delivery Method

The implementation approach should be iterative. Each iteration should include business workshops, design refinement, data modeling, governance design, data quality configuration, workflow design, integration configuration, implementation, demonstration, user feedback, and UAT. The goal is not to configure every domain in a single pass, but to build a prioritized backlog and progressively onboard domains or source systems through repeatable increments.

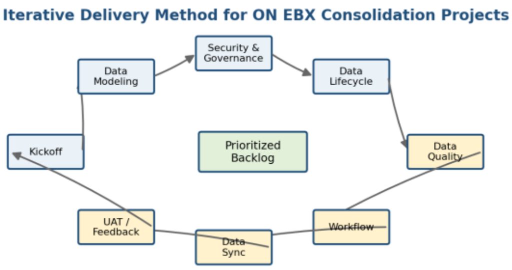

Fig. 6. Iterative delivery model for EBX consolidation implementation.

Copyright © 2018 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2018, 3(1)

e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

XIII. Implementation Patterns

Several implementation patterns are recommended. First, landing tables should remain close enough to the source to simplify traceability. Second, staging tables should be designed around the transformation process, not merely copied from source systems. Third, matching policies should be tested with realistic data and reviewed by business stewards before operational use. Fourth, survivorship rules should be explicit and should account for trusted-source hierarchy, recency, completeness, and steward overrides. Fifth, workflows should be used for exceptions and approvals rather than forcing every record through manual review.

Another important pattern is to avoid embedding complex join logic in service code when the same logic can be expressed in configuration or data movement design. Configuration-driven transformations are easier to review, migrate, and reuse across implementations. Service-level customization should be reserved for requirements that cannot be reasonably expressed through EBX configuration, DEX mapping, validation rules, or workflow design.

XIV. Benefits And Expected Outcomes

The reference architecture provides several benefits. It improves implementation repeatability because teams can reuse the same conceptual zones and responsibilities across domains. It improves auditability because source values, transformations, validation errors, match decisions, and stewardship actions are separated and traceable. It improves data quality because profiling and validation occur before records are promoted into mastered structures. It improves business trust because stewards have controlled intervention points rather than informal correction paths.

The architecture also improves scalability. New sources can be onboarded through a defined method, new domains can be added through the same zone pattern, and matching policies can evolve independently of raw ingestion. Finally, it improves communication between business and IT teams because each zone and each role has an understandable purpose.

XV. Limitations

The architecture is not a substitute for domain-specific modeling. Each implementation must still define the appropriate master entities, reference data, hierarchy structures, identifiers, permissions, and governance processes. The architecture also assumes that the organization is willing to formalize stewardship responsibilities. Without clear data ownership, even the best technical consolidation pattern will degrade into ad hoc exception handling.

The model also requires disciplined environment management. Data models, DEX mappings, match policies, workflows, and permissions must be migrated and versioned carefully. Poor release management can create inconsistency between zones or environments.

XVI. Conclusion

This paper presented a practical reference architecture for data consolidation in ON EBX5. The architecture organizes consolidation into pre-landing, landing, staging, and mastering zones and defines the operational controls required to move records from raw source capture to governed golden-record publication. It emphasizes source fidelity, staged transformation, data quality profiling, context-aware matching, stewardship intervention, and repeatable source-system onboarding.

The central argument is that enterprise MDM success depends on implementation architecture as much as product capability. EBX provides the modeling, governance, workflow, and data management foundation; the zone-based consolidation architecture provides the repeatable method for applying those capabilities to complex multi-source enterprise data landscapes.

Copyright © 2018 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Acknowledgment

This article was developed from enterprise implementation experience with Orchestra Networks EBX5. The author acknowledges the broader EBX engineering, professional services, presales, and customer delivery ecosystem whose practical implementation challenges informed the architecture described in this paper.

References

[1] K. Dutta, EBX Data Consolidation Solution, technical presentation, Orchestra Networks, Spring 2018.

[2] K. Dutta, EBX Standard Operating Procedure: Data Consolidation, technical operating procedure, Orchestra Networks, version 1.0, June 25, 2018.

[3] K. Dutta, Data Exchange Guide to Configure Data Transfer, technical configuration guide, Orchestra Networks, 2018.

[4] Orchestra Networks, EBX5 Project Methodology, implementation methodology presentation, 2016.

[5] K. Dutta, Add-on DAQA configuration artifacts and matching policy examples, EBX configuration assets, 2018.

[6] Orchestra Networks, EBX5 product and add-on implementation practices, field enablement materials, 2016-2018.

APPENDIX A: SUPPORTING ARTIFACT MAPPING

Artifact	Date / period	How it supports this paper
EBX Data Consolidation Solution	Spring 2018	Defines the primary zones, stewardship flow, matching considerations, and consolidation visuals.
EBX Standard Operating Procedure (Data Consolidation)	June 25, 2018	Provides operating procedures for loading, staging, consolidation, stewardship, and source-system onboarding.
Data Exchange Guide to Configure Data Transfer	2018	Provides DEX concepts, application registration, semantic model, data model, mapping, and transformation details.
EBX5 Project Methodology	2016	Provides broader implementation activities: data modeling, governance, lifecycle, data quality, workflow, synchronization, and user experience.
DAQA configuration artifacts	2018	Provides concrete matching-policy, process-policy, threshold, and stewardship configuration examples.

Copyright © 2018 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.