ARCHITECTURAL SOVEREIGNTY

The MedSyncAI Split-Plane and Agentic Workflow Framework

> ***Secure, AI-native interoperability without surrendering data sovereignty.***

KELENO TECHNICAL WHITEPAPER

Contents

Update this table of contents in Microsoft Word if page references are not shown.

Contents 2
1. The Modern Interoperability Paradigm: Addressing the Legacy "Hidden Tax" 3
2. The Split-Plane Architecture: Engineering PHI Sovereignty 3
The Metadata-Only Boundary 3
3. The 11-Node Agentic Workflow: From Natural Language to Deterministic Execution 4
Phase 1: Intent & Clarification (The LLM Layer) 4
Phase 2: Deployment, Validation, and Self-Healing 4
Workflow Logic Summary 4
4. Deterministic Guardrails and Self-Healing Mechanisms 5
Error Correction Logic 5
The Flow Registry & Versioning 5
5. Security, Compliance, and Audit-Ready Operations 5
The Security Stack 5
Operational Visibility 5
6. Technical Implementation Use Cases 5
Solution Matrices 6
Contact & Usage Policy 7
Contact 7
Restricted Usage & Confidentiality 7

Architectural Sovereignty: The MedSyncAI Split-Plane and Agentic Workflow Framework

1. The Modern Interoperability Paradigm: Addressing the Legacy "Hidden Tax"

In the contemporary healthcare landscape, integration remains the primary friction point for clinical innovation. Legacy interface engines impose a "hidden tax" on health systems, characterized by a reliance on a scarce pool of specialist talent, proprietary IDE lock-ins, and annual licensing fees often exceeding $200,000. These architectural bottlenecks force strategic initiatives into a 6–12 week development cycle per interface, creating a specialist-driven model that is inherently unscalable. This "tax" is not merely financial; it is a tax on clinical agility, where the latency of integration prevents the timely deployment of life-saving digital tools.

MedSyncAI establishes a new transformation mandate by shifting the integration paradigm from manual coding to intent-driven AI orchestration. By leveraging an agentic engine capable of a 30-minute deployment from a plain-English prompt, organizations can move from requirements to production in minutes. This shift fundamentally alters the ROI profile of the IT department, enabling a high-velocity environment that maintains a **38ms p95 latency** floor—matching the performance of traditional engines while removing the technical debt of legacy middleware. However, this velocity is only viable if it is underpinned by a rigorous architecture that guarantees clinical data sovereignty.

2. The Split-Plane Architecture: Engineering PHI Sovereignty

For the Healthcare Security Strategist, the primary concern of AI adoption is the protection of Protected Health Information (PHI). MedSyncAI addresses this through a Split-Plane Architecture, physically and logically isolating the Control Plane (governance) from the Data Plane (execution). This ensures that PHI remains entirely within the customer's trust boundary and never persists within the MedSyncAI core.

The following table details the component distribution across the sovereign boundary:

Plane	Component Services	Managed Data Types
Control Plane (MedSyncAI Cloud)	Identity & SSO (OIDC/MFA), Frontier LLMs (Models), Metadata DB, API Gateway, Metering	Identity signals, audit tokens, flow metadata, usage event streams
Data Plane (Customer Cloud)	Flow Runtime, Managed Containers, LLM Processing (Inference), Flow Registry, Secrets Vault, Log Analytics	PHI payloads, HL7/FHIR/X12 data, customer secrets, per-tenant audit logs

The Metadata-Only Boundary

A critical architectural nuance lies in the separation of model "intelligence" from payload "processing." While the Control Plane hosts the **Frontier LLMs** (the model weights and orchestration logic), the actual **LLM Processing**—including the inference on PHI and the **Pre-LLM Credential Redactor**—occurs within the Managed Containers of the Data Plane. The MedSyncAI Control Plane is stateless regarding PHI; the two planes communicate exclusively via "Private Endpoints" and "Delegated Access" using metadata and identity signals. This design ensures that the system satisfies CISO requirements for data sovereignty while allowing the 11-node engine to orchestrate high-stakes clinical data flows safely.

3. The 11-Node Agentic Workflow: From Natural Language to Deterministic Execution

Unlike generic chatbots, MedSyncAI utilizes an orchestrated state machine. This multi-agent approach breaks down integration requests into 11 discrete nodes to ensure that AI creativity is always bound by deterministic healthcare requirements.

Phase 1: Intent & Clarification (The LLM Layer)

The first phase focuses on translating human intent into a structured technical plan through seven specialized nodes:

- **analyze_intents** and **parse user request:** The system extracts source/target protocols and clinical mapping requirements.
- **match_components:** A deterministic node that identifies necessary building blocks (e.g., MLLP listeners or FHIR transforms).
- **gen_components:** The LLM configures each node based on the parsed request.
- **gen_clarifications** and **wait_for_answers:** If the intent is ambiguous (e.g., a missing destination IP), the system initiates a **Human-in-the-Loop (HITL)** pause. This state is persisted and resumable, ensuring the architect can provide missing context without restarting the process.
- **apply_answers:** Once the user provides clarification, the configuration is finalized.

Phase 2: Deployment, Validation, and Self-Healing

The final four nodes transform the plan into a production-grade runtime:

generate_dsl: The LLM emits a formal flow definition (Domain Specific Language).

deploy: The system pushes the definition to the runtime cluster within the customer's data plane.

validate: The system tests the flow against real schemas (HL7 v2.6, FHIR R4, etc.).

handle_errors and **finalize:** If validation fails, the system enters a self-healing loop where errors are fed back to the agents for correction, eventually moving to finalize for publication and notification.

Workflow Logic Summary

LLM-driven (Creative/Mapping): Nodes like gen_components and generate_dsl that handle complex logic and mapping.

Deterministic (Deployment/Registry): Nodes such as deploy and match_components that execute fixed code paths.

Self-healing: A bounded loop where the handle_errors node automates refinements based on deterministic validation failures.

This orchestrated approach ensures that the output is not a "suggestion" but a validated, production-ready interface that satisfies the most demanding clinical safety standards.

4. Deterministic Guardrails and Self-Healing Mechanisms

In 24/7 clinical environments, interface failure is not an option. MedSyncAI builds reliability into the runtime through automated self-correction and rigorous versioning.

Error Correction Logic

When the **validate** node detects a failure—such as a schema mismatch in a Meditech ADT feed—it triggers the **handle_errors** node. This node provides the specific error context back to the LLM to refactor the logic. To prevent infinite loops or non-deterministic behavior, this process is governed by strict **retry caps** and **deterministic guardrails**. The AI is tasked with solving a specific, bound problem within the context of the error logs, ensuring that only verified code is promoted.

The Flow Registry & Versioning

Every integration is **Versioned by Default** within the Data Plane's Flow Registry. This provides a strategic safety net for the clinical enterprise:

Rollback: Revert to any prior known-good state instantly.

Diff: Compare revisions to identify exactly how a mapping changed between versions.

Replay: Reprocess historical transactions through updated flow definitions to ensure data consistency after a logic change.

5. Security, Compliance, and Audit-Ready Operations

At MedSyncAI, security is the architecture. The platform is designed for **HITRUST alignment** and operates within a **SOC 2 Type II** environment, employing continuous vulnerability scans to maintain a defensible posture.

The Security Stack

Identity & Access: Governance is enforced through **RS256 JWT validation**, OIDC, and mandatory MFA. Role-Based Access Control (RBAC) ensures workspace isolation.

Privacy-First AI: A **Pre-LLM credential redactor** resides in the Data Plane to strip secrets from prompts. AI agents have no ambient access to data; every interaction is explicit and authorized per-use.

Transaction Lineage: The platform provides end-to-end provenance lineage. Auditors can track the journey of every HL7 or X12 message with full visibility into who modified the flow and when.

Operational Visibility

The MedSyncAI Studio surfaces real-time metrics, currently supporting over **128,420 daily transactions** across its global footprint. With **38ms p95 latency**, the engine provides the performance of a low-level C++ engine with the visibility of a modern SaaS dashboard, including anomaly alerting and real-time status updates on flow health.

6. Technical Implementation Use Cases

The split-plane architecture and agentic workflow manifest as tangible ROI across diverse healthcare scenarios.

Solution Matrices

Outpatient Networks: MedSyncAI allows health systems to onboard new locations in **5 days** rather than 12 weeks. Using site-specific overrides and shared templates, organizations achieve an **~80% reduction in per-site integration costs**.

Partner Integration: The hub-and-spoke architecture manages bidirectional feeds across **MLLP, SFTP, and REST/FHIR streams**. This streamlines X12 (837/835) workflows, significantly reducing claim rejection rates by validating data against payer-specific rules before submission.

EHR Migration: Dedicated **Vendor Packs** for **Meditech, Cerner, Allscripts, NextGen, and eCW** bootstrap mappings. The engine generates per-record reconciliation reports, ensuring 100% data integrity during the move to Epic or other modern platforms.

Unstructured Data: The **OCR + AI pipeline** converts faxes and PDFs into FHIR R4. The system extracts fields with **bounding boxes and confidence scores**, auto-routing **95% of documents** without manual entry and reducing transcription errors by 70%.

The MedSyncAI architecture represents the future of secure, high-velocity healthcare interoperability. By isolating PHI in a customer-controlled Data Plane and utilizing a self-healing 11-node agentic workflow, it provides the clinical enterprise with the agility of AI and the deterministic reliability of a traditional interface engine.

Contact & Usage Policy

Contact

For partnership and technology enquiries: info@keleno.com | www.keleno.com

Restricted Usage & Confidentiality

This case study is intended for Keleno website publication and approved stakeholder communication. It describes a proposed technology and operating model; it does not constitute audited production, financial or performance results. The customer is intentionally anonymised. No part may be reproduced, adapted, distributed or used for commercial purposes without prior written permission from Keleno.

Any capacity, cost or savings figures are illustrative source assumptions and must be independently validated for a specific site, species, operating environment and commercial model.