PRD v1 Document

AI-Powered Medical Referral and Scheduling System

Version: 1.0

Date: October 22, 2025

Project: Automated Medical Referral Processing

Team: Artera Healthcare AI Initiative

1.1 Problem Statement

Healthcare referral and scheduling workflows remain one of the most fragmented and error-prone aspects of modern clinical operations. Even in digitized environments, the pathway between a doctor's recommendation and a patient's confirmed appointment is still burdened by manual coordination, redundant data entry, and system-level silos. A single referral often requires multiple humans to transcribe information, validate insurance coverage, and coordinate scheduling across disconnected platforms. This process can extend what should be a same-day task into a multi-day—or even multi-week—exchange of emails, faxes, and phone calls.

Beyond inefficiency, this fragmentation introduces risks. Miscommunication between staff can delay care, lead to untracked referrals, or result in missed follow-ups for critical cases. From a patient's perspective, these friction points diminish trust, create financial uncertainty, and amplify anxiety during already sensitive medical situations. The burden is shared across providers and patients alike: clinicians lose valuable time to administrative tasks, while patients wait longer for the care they need.

1.2 Innovation

The Artera AI Handoff System represents a paradigm shift in how medical referrals and scheduling are handled. Rather than automating individual steps in isolation, the system reconceptualizes the entire workflow as a network of intelligent, communicative agents capable of reasoning, coordination, and action.

At its core, this innovation lies in replacing linear human handoffs with dynamic agent collaboration. Each AI sub-agent specializes in a domain of the care continuum—scribing, referral creation, insurance verification, or scheduling—but all operate within a unified orchestration fabric that allows them to exchange context and delegate decisions autonomously. This design mirrors human teamwork, yet executes with machine precision and speed.

The system embodies several novel dimensions of innovation:

- Workflow Intelligence, not Task Automation: Instead of narrowly automating repetitive actions, the system interprets medical context and intent. It transforms natural conversation—what a doctor says aloud—into structured, actionable medical events that can propagate downstream.
- Human-in-the-Loop Transparency: Automation does not replace medical judgment. The design emphasizes collaboration between human providers and AI subagents, allowing doctors to oversee, intervene, and approve as needed. The system is thus assistive, not autonomous—it augments human expertise while preserving control.
- End-to-End Patient-Centered Flow: By connecting data across referral, insurance, and scheduling pipelines, patients receive appointment options and estimated costs within minutes, not days. This transparency addresses one of healthcare's most persistent pain points: uncertainty around access and affordability.
- Composable Multi-Agent Architecture: The agents communicate via shared protocols and structured memory rather than bespoke integrations. This architecture allows the ecosystem to grow modularly—new agents or workflows can be added without redesigning the core. It transforms the system into a living, extensible infrastructure rather than a static application.

In essence, the Artera Handoff System introduces a new kind of operational intelligence to healthcare—one that blends natural language understanding, contextual reasoning, and direct EMR interaction to create a seamless bridge between clinical intent and logistical execution.

1.3 Core Technical Advance

Under the hood, the system is realized through several intertwined technical pillars:

- Multi-Agent Orchestration: Agents communicate through A2A (Agent-to-Agent) protocols, sharing structured context, intent, and state updates without human mediation. This allows asynchronous execution and recovery if one component fails or requires human input.
- AthenaHealth Integration: Each medical action—creating a referral, checking insurance, scheduling an appointment—is implemented as a tool callable via AthenaOne's REST APIs. Agents authenticate using OAuth2 and operate within the EMR's secure data schema, ensuring accuracy and auditability.
- Bedrock Model Reasoning: Each agent runs atop Amazon Bedrock, leveraging Claude Haiku 4.5 for structured reasoning, confidence scoring, and adaptive prompting. This allows the system to interpret ambiguous statements ("Let's have you see cardiology soon") and translate them into codified EMR actions.
- Cost Transparency as a Technical Product: The referral and scheduling chain is enriched with insurance metadata, enabling real-time cost estimation. The system computes copay, deductible, and total expected patient responsibility, merging medical data with financial visibility—a crucial trust component in patient-facing automation.

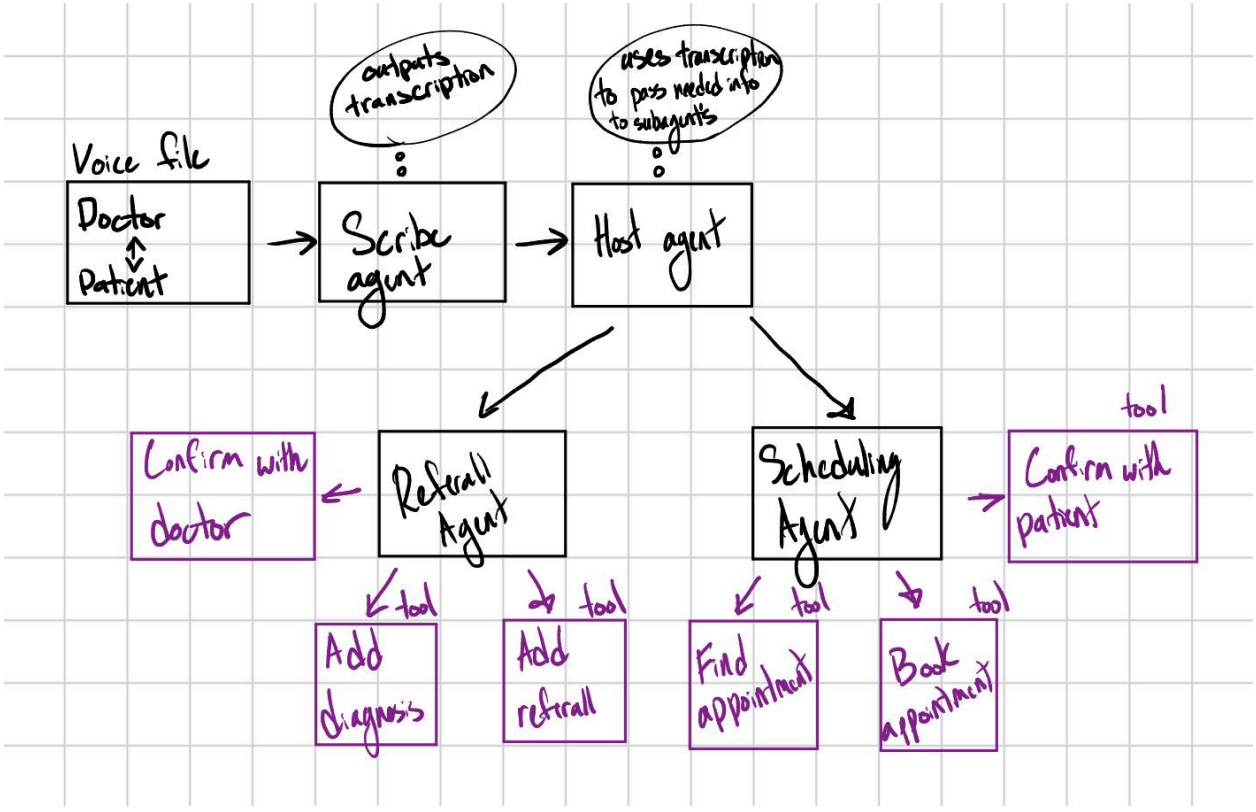
1.4 Background and Team Objectives

The LinkrAI team's mission is to:

1. Build an interoperable chain of AI subagents that can accurately process real patient interactions end-to-end, from transcript to appointment.
2. Integrate with the AthenaHealth EMR sandbox for live testing of referral creation, encounter management, and scheduling APIs.
3. Ensure HIPAA-compliant handling of sensitive data and audit trails across all agent communications.
4. Quantitatively demonstrate reductions in referral processing time (target: from 7 days → <2 minutes) and improvements in patient engagement and administrative throughput.

2. System Architecture Overview

2.1 High-Level Diagram



2.2 User Interaction and Design

Primary Users and Touchpoints

Doctor / Provider

- Input: Speaks naturally during consultation.
- Touchpoint: Indirect; the Scribe Agent listens passively.
- Feedback Loop: The Referral Agent may request doctor confirmation if diagnostic or referral details fall below confidence thresholds.
- Outcome: Receives confirmation through EMR interface that a referral has been created and, once verified, that scheduling is complete.

Patient

- Input: None during transcription; becomes active once Scheduling Agent sends options.
- Touchpoint: Receives appointment options (via SMS, email, or Artera message) with estimated cost and preparation instructions.
- Feedback Loop: Confirms preferred time; system updates EMR automatically.

Administrative Staff

- Input: Oversight and manual override for flagged or failed automations.
- Touchpoint: Dashboard view showing workflow state: Listening → Parsed → Referral Created → Appointment Booked → Completed.
- Feedback Loop: Can manually resume or edit a workflow at any point.

Interaction Flow

Doctor–Patient Conversation (Voice Input)

- The Scribe Agent continuously listens to the session and generates structured text in real time.
- The Host Agent is triggered once the session ends or a referral cue is detected.

Host Agent Reasoning

- Host Agent parses the transcription, identifies intent (e.g., “refer to cardiology”), and determines which sub-agents to call.
- It constructs a structured task message containing all relevant fields: specialist type, urgency, diagnosis keywords, patient ID.

Referral Agent Execution

- Receives referral task from Host.
- Calls the appropriate Athena tools:
    - Add_diagnosis
    - Add_referral
    - If required fields are missing or confidence is low, it pauses execution and sends a confirmation request to the doctor.

Scheduling Agent Execution

- Receives scheduling task from Host (once referral confirmed).
- Calls:
    - Find_appointment
    - Once optimal options are found, sends confirmation and cost breakdown to the patient via a secure communication channel.
    - Book_appointment

Feedback Confirmation

- Doctor confirms referral accuracy.
- Patient confirms appointment.
- System updates EMR with final status and logs the full workflow trace.

3. Requirements

3.1 Functional Requirements

The Artera AI Handoff System automates the clinical handoff pipeline for referrals and scheduling, focusing on seamless collaboration between the Scribe Agent, Host Agent, Referral Agent, and Scheduling Agent.
Each sub-agent performs distinct responsibilities within the Athena-integrated referral-to-appointment lifecycle.

3.1.1 Scribe Agent Requirements

FR-1.1: Referral Detection
The system SHALL detect when a provider verbally indicates a referral or follow-up during a patient encounter, including implicit referral intent (e.g., “Let’s have you see cardiology next week”).

FR-1.2: Structured Data Extraction
The system SHALL extract the following structured data from transcribed consultations:
- Specialist type (e.g., cardiology, dermatology)
- Reason for referral or follow-up
- Urgency level (emergency, urgent, moderate, routine)
- Relevant clinical context and symptoms

FR-1.3: Real-Time Processing
The system SHALL process doctor–patient conversations in near-real time, outputting structured transcripts ready for Host Agent ingestion before the visit ends.

FR-1.4: Confidence Scoring
The system SHALL assign confidence levels (high, medium, low) to each detected referral or scheduling intent and route low-confidence detections for doctor verification via the Host Agent.

3.1.2 Host Agent Requirements

FR-2.1: Intent Routing
The Host Agent SHALL receive structured transcripts and determine the correct downstream action, routing tasks to the Referral or Scheduling subagents.

FR-2.2: Context Assembly

The Host Agent SHALL generate a unified data package including:

- Patient MRN
- Provider ID
- Referral intent metadata
- Temporal context (timestamp, urgency, etc.)

FR-2.3: Agent-to-Agent Messaging
The Host Agent SHALL communicate via A2A protocol, maintaining message correlation IDs for traceability between subagents.

FR-2.4: Error Recovery
If any sub-agent call fails, the Host Agent SHALL queue the task and retry automatically, ensuring the workflow resumes once external dependencies (e.g., EMR or network) recover.

3.1.3 Referral Agent Requirements

FR-3.1: EMR Integration
The Referral Agent SHALL create a referral record in the AthenaOne EMR with all required patient and provider fields populated automatically.

FR-3.2: Data Validation
Before record creation, the system SHALL verify that the following data are present and valid:

- Patient MRN (Medical Record Number)
- Primary insurance data
- Referring provider ID
- Diagnosis or referral reason

FR-3.3: Referral Tracking
Each referral record SHALL have a unique referral ID and persistent state (pending, approved, scheduled, denied) to track progress across subagents.

FR-3.4: Doctor Confirmation Loop
The Referral Agent SHALL request confirmation from the provider when referral details have low confidence or when required data (e.g., diagnosis codes) are incomplete.

FR-3.5: Missing Data Handling
If the EMR rejects a referral request or data is incomplete, the system SHALL alert staff and pause execution until corrected.

3.1.4 Scheduling Agent Requirements

FR-4.1: Appointment Search
The Scheduling Agent SHALL search for available appointments matching the specialist type, urgency, and patient preferences (location, time, provider).

FR-4.2: Appointment Booking
The agent SHALL book appointments directly into the EMR using the appropriate AthenaOne API endpoints once the patient confirms their preferred time.

FR-4.3: Patient Communication
The agent SHALL send appointment options and confirmations via the patient's preferred contact method (SMS, email, or Artera message), including:

- Specialist name and address
- Appointment time options
- Estimated cost (from insurance data)
- Preparation or pre-visit instructions

FR-4.4: Alternative Scheduling
When no slots are available, the system SHALL query alternate in-network specialists and re-offer available times to the patient.

FR-4.5: Confirmation Sync
Upon patient confirmation, the Scheduling Agent SHALL update the appointment status in Athena and notify both doctor and patient automatically.

3.1.5 System Integration Requirements

FR-5.1: Secure Agent Communication
All agents SHALL communicate via authenticated, encrypted A2A channels with correlation IDs for workflow traceability.

FR-5.2: EMR Tool Invocation
All clinical actions (add_referral, book_appointment, add_diagnosis) SHALL be implemented as callable tools mapped to AthenaOne REST endpoints.

FR-5.3: Fault Tolerance
If the Athena API or insurance endpoints are unavailable, the system SHALL queue pending operations and retry based on exponential backoff policies.

FR-5.4: Logging and Auditability
Every tool call SHALL be logged with:

- Agent origin

- Input parameters
- Timestamp
- API response
- Workflow correlation ID

3.2 Non-Functional Requirements

3.2.1 Performance

- NFR-1.1: End-to-end referral → appointment confirmation time SHALL not exceed 2 minutes under normal conditions.
- NFR-1.2: Scribe Agent transcription latency SHALL be under 3 seconds for spoken referral cues.
- NFR-1.3: The system SHALL support at least 50 concurrent patient workflows per cluster without degradation.
- NFR-1.4: Deployment SHALL scale horizontally to support 1,000+ daily workflows per clinic instance.

———

3.2.2 Reliability

- NFR-2.1: System uptime SHALL be at least 99.5% during operating hours (6AM–8PM).
- NFR-2.2: No data loss for referral or appointment records is permissible. All transactions must be recoverable.
- NFR-2.3: On external failure (e.g., EMR outage), the system SHALL degrade gracefully and resume queued operations automatically.
- NFR-2.4: Referral detection and classification accuracy SHALL reach ≥ 90% overall and ≥ 95% for high-confidence cases.

3.2.3 Security

- NFR-3.1: All operations SHALL comply with HIPAA and relevant state-level data protection laws.
- NFR-3.2: All data in transit SHALL use TLS 1.3+, and all stored data SHALL use AES-256 encryption.
- NFR-3.3: Role-based access control (RBAC) SHALL restrict EMR actions to authorized system roles.
- NFR-3.4: Agents SHALL authenticate using OAuth 2.0 credentials and scoped API keys per Athena practice.

3.2.4 Usability

- NFR-4.1: Both clinicians and patients SHALL have transparent visibility into referral and scheduling progress.
- NFR-4.2: Human-in-the-loop review SHALL be possible at any stage without disrupting downstream automation.
- NFR-4.3: Patient-facing notifications SHALL be written at an 8th-grade reading level, with clear cost and next-step explanations.
- NFR-4.4: Error messages SHALL be actionable and traceable to their source agent and tool.

3.2.5 Maintainability

- NFR-5.1: All agent activity SHALL be logged and correlated across subagents for end-to-end debugging.
- NFR-5.2: System dashboards SHALL visualize workflow status, throughput, and failure rates in real time.
- NFR-5.3: Configuration changes (thresholds, model prompts, retry intervals) SHALL be updatable without redeployment.
- NFR-5.4: Developer documentation SHALL include API references, prompt structures, and data flow diagrams kept current with each PR release.

4. System Architecture

4.1 Overview

The Artera AI Handoff System operates as a modular, agent-based microservice architecture deployed on AWS EKS.
Each agent container (Scribe, Host, Referral, and Scheduling) communicates through the Model Context Protocol (MCP) and interacts directly with AthenaHealth REST APIs.
Agents use AWS Bedrock (Claude Haiku 4.5) for reasoning and tool orchestration, while workflow data and state persistence are maintained in DynamoDB.

This architecture enables:

- Clear separation of responsibilities between agents
- State continuity across encounters, referrals, and appointments
- Auditable, secure communication via logged MCP tool invocations
- Low-latency operation, achieving an end-to-end referral-to-appointment workflow in under two minutes

4.2 Referral Architecture

The referral subsystem begins immediately after the Host Agent identifies referral intent from the conversation transcript.
Its design mirrors AthenaOne's internal workflow, ensuring each downstream action—encounter creation, diagnosis addition, and referral order creation—maps correctly to Athena API structures.

Referral Workflow

1. Initialize Encounter
    - The Host Agent initiates or retrieves an active encounter using the Athena endpoint get_active_encounter.
    - This ensures a valid encounter ID is available for subsequent diagnosis and referral linkage.
2. Add Diagnosis to Encounter
    - The Referral Agent calls the MCP tool add_diagnosis.
        - Input: chosen diagnosis and optional provider notes.
        - SNOMED CT codes are retrieved from a cached mapping file named DIAGNOSIS_MAPPINGS.
        - Optionally, ICD-10 codes may be added for insurance linkage.
        - Example: "Irregular heartbeat" → SNOMED 164865005 ("Cardiac arrhythmia") → ICD-10 I49.9.
3. Select Referral Order Type
    - The Referral Agent uses the MCP tool list_referral_types, which queries the Athena endpoint for available referral categories.
    - The agent selects the appropriate order based on detected specialty (e.g., cardiology, dermatology).
4. Create Referral Order
    - The Referral Agent executes create_referral_order.
    - Inputs: referral type, reason for referral, and optional provider notes.
    - The agent auto-fills these fields using structured data extracted from the transcript (symptoms, urgency, encounter ID).
    - The resulting referral ID is stored and shared back to the Host Agent.
5. Doctor Confirmation
    - If the Referral Agent's confidence score is low or required data are missing, it pauses execution and sends a confirmation prompt to the doctor through the EMR interface or dashboard.

Referral MCP Tools

- list_diagnosis: retrieves SNOMED CT codes from cache or external service
- add_diagnosis: attaches coded diagnosis to encounter
- list_referral_types: retrieves valid referral types from Athena

- create_referral_order: submits completed referral order to EMR

Prompt Design
Each Referral Agent prompt explicitly defines its objective:

“Your role is to create a validated medical referral using Athena tools. Ensure the referral details, diagnosis, and specialist information align with the intended appointment.”

All relevant tool definitions are injected into the prompt context to reduce hallucination and improve reliability.

4.3 Scheduling Architecture

After a referral is successfully created and validated, the Scheduling Agent takes control of the workflow.
It manages provider lookup, appointment discovery, booking, and patient communication using AthenaHealth’s Online Appointment Scheduling Workflow (see: https://docs.athenahealth.com/api/workflows/online-appointment-scheduling).

Scheduling Workflow

1. Get Departments
    - The agent calls the MCP tool list_departments.
    - For this deployment, all scheduling operations assume a single department (Department 162), simplifying routing and provider search.
2. Get Providers
    - The agent calls get_providers, filtered by department 162 and the specialty derived from the referral.
    - This returns a list of available providers with their IDs, specialties, and locations.
3. Find Appointment Slots
    - The agent uses find_appointment_slots.
    - Input includes provider ID, department (162), and reason ID (-1, indicating general-purpose appointments).
    - The tool returns available time slots, which may be truncated to the top three options to prevent model overloading or hallucination.
4. Book Appointment
    - The agent calls book_appointment with the chosen slot, patient ID, and department ID.
    - Athena returns an appointment confirmation object that includes the appointment ID and metadata.
5. Patient Notification and Confirmation
    - The Scheduling Agent then sends a confirmation message to the patient via the Artera Messaging API, including:

- Provider name and specialty
- Appointment date, time, and location
- Estimated cost (if provided by insurance data)
- Preparation or pre-visit instructions

When the patient confirms (“YES” reply), the appointment status is updated in Athena and logged in the workflow.

Scheduling MCP Tools

- get_providers: retrieves providers by department or specialty
- find_appointment_slots: retrieves available time slots for a given provider
- book_appointment: finalizes appointment in Athena and returns confirmation

4.4 Prompt and MCP Integration

Each agent operates through a consistent MCP (Model Context Protocol) schema that defines available tools, expected inputs, and validation rules.
MCP allows large language models to act as orchestrators, invoking real-world actions through Athena APIs while maintaining structured context.

The Host Agent manages this orchestration, injecting the proper MCP context into each downstream subagent and maintaining message correlation IDs across tools.

Prompt Example for Scheduling Agent

“Your goal is to locate and book a patient appointment based on referral details. Use get_providers to list available providers in department 162, find_appointment_slots to retrieve open times, and book_appointment to finalize the appointment once the patient confirms.”

All prompt outputs are validated JSON objects that conform to Athena’s API schemas, ensuring EMR compatibility and deterministic tool behavior.

5. Use Cases, User Stories, and Prototyping Tests
5.1 Overview
This section defines the primary user interactions and agent workflows that demonstrate the functionality of the Artera AI Handoff System, focusing on the Referral and Scheduling subagents.
Each use case is supported by user stories and acceptance tests that link directly to implementation tasks on Trello and GitHub.

At least 10 of these user stories are backed by active prototyping code and test coverage in the current sprint.

5.2 Use Cases

Use Case 1 — Successful Automated Referral and Scheduling
Primary Actor: Doctor / Scribe Agent
Goal: Complete a patient referral and schedule a follow-up appointment automatically.
Preconditions:

- Patient is in an active encounter.
- Doctor's statement includes a referral intent.
- EMR (AthenaOne) and insurance systems are available.

Main Flow:

1. Doctor says: "Let's refer you to cardiology for your irregular heartbeat."
2. Scribe Agent transcribes and detects referral intent.
3. Host Agent extracts structured fields: specialty = cardiology, reason = irregular heartbeat, urgency = routine.
4. Referral Agent initializes encounter, adds diagnosis, and creates a referral order.
5. Scheduling Agent finds available cardiology appointments in department 162.
6. Patient receives SMS: "Appointment scheduled with Dr. Smith (Cardiology) on Nov 4 at 2:00 PM. Estimated cost: $50. Reply YES to confirm."
7. Patient replies YES.
8. Appointment is booked, EMR updated, and workflow logged as complete.

Success Guarantee:
Referral record, insurance verification, and appointment confirmation are all completed within two minutes.

Use Case 2 — Insurance Denial and Escalation

Primary Actor: Insurance Verification Subprocess
Goal: Handle coverage denial gracefully while allowing provider intervention.

Flow:

1. Referral and diagnosis created successfully.
2. Insurance check returns denial ("Specialist not in network").
3. Referral Agent logs denial and alerts referring doctor.
4. Doctor receives options: switch to in-network specialist, request peer-to-peer review, or offer patient self-pay option.

5. Doctor chooses alternate provider.
6. Scheduling Agent resumes workflow with the new provider.

Success Guarantee:
Workflow recovers within one hour and provides clear feedback to both doctor and patient.

Use Case 3 — Prior Authorization Handling

Primary Actor: Insurance Verification AI Agent
Goal: Automate prior authorization submission when required.

Flow:
1. Referral detected and verified.
2. Insurance policy flags “prior authorization required.”
3. Insurance Agent extracts clinical documentation and submits prior auth.
4. System checks authorization status every two hours.
5. Once approved, workflow resumes to scheduling.
6. Patient receives notification and confirms appointment.

Success Guarantee:
Authorization handled asynchronously, no human intervention required.

Use Case 4 — Doctor Confirmation Loop

Primary Actor: Doctor / Referral Agent
Goal: Validate low-confidence referral before EMR submission.

Flow:
1. Scribe detects potential referral (“You might want to see an ENT”).
2. Host Agent marks low confidence (< 0.7).
3. Referral Agent sends confirmation request to doctor: “Confirm ENT referral?”
4. Doctor selects YES in EMR dashboard.
5. Workflow continues with validated referral.

Success Guarantee:
Human-in-the-loop review completed in under 30 seconds.

Use Case 5 — Appointment Rebooking After Patient Decline

Primary Actor: Scheduling Agent / Patient
Goal: Rebook appointment when patient declines or requests alternate time.

Flow:

1. Scheduling Agent offers three time slots.
2. Patient replies: “Can I do Friday instead?”
3. Agent searches next available Friday slots.
4. New appointment booked and EMR updated.

Success Guarantee:
Patient receives alternate confirmed appointment without manual scheduler intervention