

Agentic Infrastructure for the Democratization of Care

MCP, FHIR and accountable coordination across accessible care systems

RESEARCH LENS	SCOPE
International / Technical / Cross-specialty	Reference architectures, inclusive integration, trust boundaries and evaluation
Central question	How can this field advance the democratization of care?
Evidence cutoff	October 2, 2026

Democratization of care means expanding people's practical ability to obtain appropriate, affordable care, exercise informed choice and participate in decisions that affect them.

An AI-assisted academic working paper. General subject-matter research; no original participant data or independent peer review. Empirical evidence, mathematical results, assumptions and proposed evaluations are distinguished in the text.

Abstract & research position

Public-interest focus. This paper examines agentic infrastructure as a means of advancing democratization of care: wider practical access, affordability, informed choice and equitable participation. Claims of benefit require evidence about who gains and who remains excluded.

Question. How can programmable agents coordinate heterogeneous care systems without confusing the ability to invoke a tool with the authority or competence to perform a consequential action? This paper develops a reference architecture for programmable care-coordination systems in which agent reasoning, clinical semantics, authorization and workflow execution remain distinct but connected layers.

Evidence base. The technical analysis draws on HL7 FHIR, SMART App Launch, MCP authorization and security documentation, NIST's generative AI risk profile and primary vendor materials.^{[1][2][3][4][7]} These establish interface concepts and risk-management guidance. They do not establish that an implementation exists or that an agent has demonstrated clinical safety.

Finding. MCP can expose discoverable tools to an agent; FHIR can structure health information; SMART can support scoped application authorization. None of these alone resolves patient identity, clinical appropriateness, cross-organization accountability or the economics of implementation. A useful care hub must compose these mechanisms around a defined workflow with explicit control of writes, disclosure and escalation.

Proposed contribution. The paper specifies a bounded execution architecture, an interoperability contract, a threat model, a synthetic evaluation suite and a staged developer program. Neural networks and transformer models are treated as probabilistic components. Clusters provide computational capacity and isolation strategies; they are not evidence of general intelligence or reliability.

Design thesis: Let an agent propose and interpret; let policy determine permission; let a controlled executor perform authorized actions; let independently observable evidence determine success.

The term “agentic” is used here for a system that can select and sequence tools within an assigned task. It does not imply AGI, autonomous medical practice or unlimited delegation. The proposed research and analytics function monitors evidence and performance, but cannot silently broaden its own authority.

Paper map

Sections 1-3 define the standards and architecture. Sections 4-5 examine safety and the emerging vendor landscape. Sections 6-7 describe evaluation and the developer program. The appendix provides a concrete review contract for one supervised referral workflow.

Research method & evidence appraisal

Technical evidence and conformance boundaries

The review selected primary standards and vendor documentation relevant to agent-facing tools, health-data exchange, authorization and risk management. Specifications were read as normative interface descriptions; vendor materials were treated as claims about an offering. The review did not execute conformance tests, inspect implementation source code or access a production care environment.

Evidence class	Extraction focus	Interpretation limit
Protocol specification	Version, transport and authorization scope	Defines contract, not deployed correctness
Health-data standard	Resource and profile boundaries	Does not prove semantic equivalence
Vendor announcement	Date, availability and outcome status	Commercial claim, not independent evaluation
Risk guidance	Control objective and applicability	Reference framework, not certification

The architectural synthesis is a proposed design argument. It decomposes a consequential task into interpretation, authorization, execution and verification, then identifies failures at each boundary. The design is assessed for testability and accountability, not claimed empirical superiority. A comparison with a simpler non-agent workflow remains necessary.

Future evaluation should freeze model, prompt, adapter, policy and dataset versions. A conformance result should name the tested operations and negative cases. Generalization should be assessed on held-out workflows and independent environments. Architectural plausibility is insufficient evidence for clinical or security claims.

Transparency statement

This is an AI-assisted, targeted narrative synthesis. Source selection was purposive rather than exhaustive; no registered systematic-review protocol is claimed. The reference list identifies the material used, including access limitations. No original participant data were collected. Proposed studies, capabilities and economic scenarios are labeled and have not been independently validated.

1. Standards solve different problems

FHIR supplies a resource-oriented approach to representing and exchanging healthcare information. SMART App Launch supplies authorization and launch conventions around applications using health data. MCP supplies a protocol through which a model-enabled application can discover and invoke tools. Their roles overlap at integration boundaries but are not interchangeable.^{[1][2][3]}

Layer	Principal concern	What it does not establish
FHIR	Data structure and exchange semantics	A patient's permission for every use
SMART	Scoped application access and context	Clinical appropriateness of an action
MCP	Agent-facing tool interaction	Universal healthcare compatibility
Workflow policy	Allowed state transitions and responsibility	Correctness of every external system

The architectural implication is a translation boundary. An agent-facing tool should present a narrow, typed operation such as retrieving an authorized referral status. An adapter should translate that operation into the target system's supported API and validate the response. The agent should not construct arbitrary network calls or infer undocumented write operations from prose.

Versioning is essential. A connection that works against one FHIR profile or vendor implementation does not prove compatibility with another. The integration record should name the specification version, required profiles, authentication method, supported operations and known omissions. A capability statement should describe tested behavior, not an aspiration to support every endpoint.

Semantic alignment is a separate workstream

Two systems can exchange valid JSON while disagreeing on identifiers, code systems, units or the meaning of a status. The proposed semantic layer records source terminology, transformation rules and uncertainty. A mapping that cannot preserve meaning should produce an explicit unresolved state rather than a plausible but unsupported interpretation.

The product should expose this uncertainty to the responsible operator in useful terms: which field is missing, which decision it blocks and what evidence would resolve it. A generic confidence score is insufficient. Confidence in language generation, confidence in identity matching and confidence in source freshness are different quantities and should not be collapsed into one reassuring percentage.

2. A proposed execution architecture

The proposed design separates planning from execution. A transformer-based model can interpret a request, identify missing context and propose tool calls. A deterministic workflow controller maintains task state. A policy service checks the actor, organization, subject, purpose and permitted operation. A typed adapter performs the authorized call. An event record captures what actually happened.

- 01 User or staff request with verified context
- 02 Agent proposes a bounded operation
- 03 Policy checks identity, purpose and authority
- 04 Typed adapter validates and executes
- 05 Workflow records receipt, result and next owner

The execution path should not rely on the model remembering security instructions. Authorization belongs outside the model and is reevaluated at the point of action. A change in patient, tenant, recipient or purpose should trigger a new check even if the conversation appears continuous. Delegating to another agent does not enlarge the originating actor's permissions.

A write requires an explicit operation schema and an idempotency mechanism where the target supports it. The controller should distinguish proposed, approved, submitted, acknowledged and reconciled states. A timeout after submission is an ambiguous outcome, not proof that the operation failed. Retrying blindly may create duplicate appointments, messages or financial events.

Recovery as a first-class feature

The workflow should preserve enough information to determine whether a remote action occurred. If the state cannot be reconciled, the system should stop automated retries and assign a human owner. Recovery may require a compensating action rather than deletion: canceling an appointment is not the same as pretending it was never booked.

The proposed architecture therefore produces an action receipt with a correlation identifier, target system, authorized purpose, input version, submission time and disposition. Sensitive content should not be copied into a general analytics log. The receipt establishes operational accountability while the underlying data remains in the appropriate governed store.

3. Secure networks and bounded delegation

Secure networking is an implementation discipline, not a decorative claim. The proposed care hub should restrict outbound destinations, isolate tenants, encrypt traffic, manage secrets outside model context and limit which services can reach sensitive systems. Network controls should be tested against the actual deployment topology, including development environments and support access.

MCP's authorization specification addresses HTTP-based authorization, while its security guidance identifies risks such as token passthrough and confused-deputy behavior.^{[3][4]} The proposed policy is stricter than merely supporting a protocol: sensitive remote tools should require authenticated, audience-appropriate access and least-privilege scopes. A tool being discoverable should never mean it is available to every agent or user.

Threat	Proposed control	Verification evidence
Prompt injection in retrieved text	Treat content as data; policy outside model	Adversarial fixtures cannot widen scope
Cross-tenant request	Tenant-bound credentials and storage checks	Negative tests deny foreign identifiers
Duplicate consequential write	Idempotency and reconciliation	Retry tests produce one final action
Secret disclosure	Secret broker and output filtering	No credential appears in logs or prompts
Unbounded delegation	Explicit tool and budget limits	Nested calls retain original constraints

Agent-to-agent communication adds another identity boundary. A receiving agent needs a verifiable request envelope, not merely a natural-language assertion that another agent has permission. The envelope should identify the originating principal, allowed operation, expiration and intended audience. Returned instructions should be evaluated under the receiving system's policy rather than accepted as superior authority.

Clusters and operational resilience

Compute clusters can separate workloads and support scaling, but replication alone does not prevent correlated model or policy failures. A flawed prompt or adapter can propagate across every worker. Proposed safeguards include version-pinned deployments, canary cohorts, bounded queues, circuit breakers and a tested rollback path. Availability metrics should be paired with correctness and containment metrics.

Security evidence should include adversarial testing and incident exercises. A high uptime figure does not establish safe data handling, and encryption does not establish appropriate use. External statements should identify implemented controls and their verification scope rather than imply that the architecture is unconditionally secure.

4. Neural systems, interpretation and evidence

A neural language model is useful for interpreting varied requests and generating structured proposals. Its output is still an inference. The proposed design should preserve source evidence alongside extracted fields so that a reviewer can distinguish what a record says from what the model inferred. Missing evidence should remain missing rather than being completed with a plausible narrative.

An “interpretability agent” could assist by comparing a proposed action with the evidence and policy that support it. That agent should be described as a checking component, not a proof engine. If it shares the same model family, prompts or retrieval errors as the acting agent, its mistakes may be correlated. Agreement between two agents is not independent validation.

Output type	Required representation	Review consequence
Direct extraction	Source location and original value	Verify faithful transfer
Terminology mapping	Mapping version and ambiguity flag	Review unresolved semantic differences
Inference	Explicit inference label and rationale summary	Do not substitute for missing facts
Proposed action	Preconditions and expected effect	Evaluate authority and consequence

The system should expose concise, evidence-linked explanations appropriate to the operator. It need not reveal internal model reasoning to provide accountability. A useful explanation identifies the relevant source, the policy applied, the uncertainty and the next action. A long generated justification can increase confidence without increasing correctness.

Research and monitoring function

The proposed analytics function can track drift in task mix, extraction errors, refusal rates, latency, review burden and outcome disparities. It can identify clusters of failures for investigation. It should not retrain production behavior or change permission rules without a controlled release process. Monitoring observations need their own provenance and retention rules.

This is an innovation agenda grounded in bounded systems. The aim is to make previously impractical coordination possible while retaining clear human and institutional responsibility. Terms such as neural network, transformer and cluster should describe real components when deployed; they should not be used as substitutes for measured capability, nor as claims that the system has achieved general intelligence.

5. What the emerging ecosystem demonstrates

Cognizant announced Workflow Agentic Processing and Enterprise MCP Servers for TriZetto Facets and QNXT on September 28, 2026, describing a library of more than 100 MCP tools. The announcement also states that early-adopter performance results would be shared later.^[5] This is evidence of a vendor offering and direction of travel, not independent proof of operational benefit.

Stedi's MCP documentation provides another example of an agent-facing healthcare interface.^[6] These developments support the proposition that tools can expose specialized administrative capabilities to agents. They do not demonstrate universal compatibility, unrestricted data access or an operational partnership. Integration feasibility must be established with each vendor and customer environment.

Evidence class	What it supports	What remains unproven
Standards specification	Interface and behavior requirements	Quality of a specific implementation
Vendor documentation	Advertised product surface	Local configuration and permissions
Vendor announcement	Availability or roadmap claim	Independent effectiveness and safety
Controlled deployment study	Outcomes in a defined setting	Transfer to different populations or systems

The number of tools is an especially weak proxy for value. One reliable tool that closes a referral loop may be more useful than many overlapping read operations. Tool count should be accompanied by supported workflows, permission boundaries, failure handling, version stability and the cost of maintenance. A catalogue is not an outcome study.

An integration strategy for inclusive participation

Begin with a capability matrix rather than a promise of universal integration. For each candidate partner, identify supported operations, commercial access terms, data-handling restrictions, sandbox availability, rate limits and review requirements. Record whether a capability is documented, tested in a sandbox, tested with authorized production data or validated in routine operations.

The proposed portal should make these statuses visible to developers. A demonstration can use synthetic data and still be compelling, provided it is labeled as a demonstration. Product communication should reserve “integrated” for an observed end-to-end result and specify its scope. This prevents a broad innovation narrative from outrunning the evidence required for trustworthy deployment.

6. Evaluation before consequential autonomy

The proposed evaluation unit is a complete task, including failed and abandoned attempts. Tool-call success is a useful diagnostic but can be misleading: every call may return a valid response while the patient is matched incorrectly or the workflow remains unresolved. Evaluation should therefore connect technical traces to independently adjudicated task outcomes.

Test family	Example fixture	Passing evidence
Identity	Similar names and conflicting identifiers	Correct match or safe unresolved state
Semantics	Mixed units and ambiguous code mapping	No silent conversion or invented certainty
Permissions	Revoked access and wrong recipient	Action denied before disclosure
Recovery	Timeout after remote submission	Reconciliation without duplicate write
Injection	Malicious instructions in tool output	No policy or destination expansion
Equity	Language and accessibility variants	Error and escalation rates reported by group

Start with synthetic fixtures containing known answers and deliberately difficult cases. Then test in a shadow mode where the system proposes actions without executing them. Compare proposals with the actual workflow and independent review. Only after the error pattern is understood should a narrowly bounded supervised execution phase begin.

A proposed primary endpoint is the proportion of eligible tasks completed correctly within the allowed time and authority boundary. Secondary endpoints include human review minutes, unresolved cases, latency, cost and user burden. Safety endpoints include unauthorized disclosure, wrong-subject action and duplicate consequential writes. Report both rates and absolute counts; rare severe events should not disappear inside a composite success score.

Statistical and operational caution

A zero-event pilot does not demonstrate zero risk. Sample size and exposure determine what can be inferred. Predefine acceptable risk thresholds with qualified owners and state the uncertainty around observed rates. Evaluate on a held-out set and preserve failures for regression testing. Repeatedly tuning against the same benchmark can create apparent progress without generalizable reliability.

NIST's generative AI profile offers a risk-management reference, not certification of the proposed system.^[7] Implementing organizations should translate relevant risks into concrete tests, owners and release criteria, then document which controls were verified and which remain assumptions.

7. A developer portal built around contracts

“Developers welcome” should mean that a developer can understand a capability, obtain appropriate access, test it safely and know what evidence is required for release. The proposed portal should prioritize readable contracts, synthetic examples, typed schemas, version history and observable error behavior. A gallery of impressive demos is useful only if it leads to reproducible implementation.

Portal artifact	Required content	Why it matters
Tool catalogue	Input/output schema, side effects, scope	Prevents ambiguous capability discovery
Sandbox	Synthetic tenants and resettable fixtures	Supports experimentation without real records
Integration guide	Versions, authentication and limits	Makes compatibility claims testable
Evaluation pack	Normal, adversarial and recovery cases	Encourages consistent release evidence
Change log	Deprecations and migration windows	Reduces unplanned downstream failure

Each tool should state whether it is read-only, reversible or consequential. A tool description should avoid persuasive language that encourages the model to use it unnecessarily. It should identify preconditions, prohibited uses and the meaning of empty or partial responses. Documentation should be tested against the actual adapter behavior, not generated independently and assumed correct.

The proposed release process has four states: experimental, sandbox-verified, supervised pilot and operationally validated. Promotion requires evidence; time in the catalogue is not enough. Breaking changes should trigger compatibility review and regression tests. Emergency disablement should be possible for one tool or tenant without shutting down unrelated safe functions.

Economics of the developer ecosystem

Measure developer success by time to a verified useful workflow, not signups alone. Record onboarding obstacles, support demand, abandoned integrations and maintenance effort. A broad tool ecosystem can create a long tail of brittle connectors; a smaller set of well-maintained contracts may offer more dependable value. Partner incentives should reward sustained interoperability and clear responsibility for failures.

This approach gives innovation a credible institutional form. Programmable coordination systems should expand their capabilities through evidence and accountable oversight. The portal becomes a place where ambitious ideas are translated into bounded, testable and maintainable workflows.

Appendix. Supervised referral contract

The following is a proposed design artifact, not an implemented endpoint. It illustrates the specificity needed before a care workflow should be exposed to an agent. The intended action is to prepare a referral handoff and obtain a human-approved submission, not to diagnose, select treatment or guarantee appointment availability.

Contract element	Proposed requirement
Inputs	Verified subject reference, originating organization, service request, permitted recipient, purpose and source versions
Preconditions	Active authority, minimum required fields, recipient capability and current workflow state
Read stage	Retrieve only permitted context; preserve source provenance and unresolved fields
Proposal stage	Produce a structured draft with missing-data flags and expected recipient
Approval stage	Authorized reviewer confirms content and consequential action
Execution stage	Submit once with correlation and idempotency controls where supported
Closure	Record acknowledgement, acceptance/rejection and responsible next owner
Failure	Stop on identity conflict, permission failure or ambiguous prior submission

The review packet should contain the proposed payload, evidence references, permission decision and a plain-language description of what will happen. Human approval must be meaningful: a reviewer should have enough time, context and authority to reject or correct the action. A mandatory click without those conditions is weak oversight.

Limitations and research agenda

This paper is a targeted technical synthesis. It does not benchmark model families, test real APIs, verify security controls or establish a deployed architecture. Vendor evidence is identified as vendor evidence. Specifications were consulted at explicit versions, and future revisions may alter implementation details.

The next research deliverable should be a runnable synthetic prototype with a published failure report, followed by an authorized shadow evaluation. The central question is whether the proposed separation of reasoning, policy and execution reduces consequential errors without transferring excessive burden to human reviewers. That question can be answered empirically; it should not be settled by architectural language alone.

Democratization of care: an evaluation framework

Agentic systems as accessible coordination infrastructure

Programmable agents can be studied as an accessibility mechanism: they may help people interpret administrative language, move between services and express preferences through different communication modes. These possibilities do not establish that an agent is safe, accurate or beneficial. The democratization question is whether the entire service expands effective choice and successful care completion across groups.

An agent should not become a compulsory gatekeeper. People need understandable routes to human assistance, correction and withdrawal from an automated interaction. The care organization must retain responsibility for consequential decisions and ensure that failures do not strand a patient between systems. A service that performs well for fluent, digitally confident users may still exclude people with limited connectivity, disability or uncommon communication needs.

Developer participation also has distributional consequences. Documentation, affordable testing environments, accessible conformance tools and transparent connector terms can enable community organizations and smaller developers to contribute. A nominally open protocol can coexist with inaccessible commercial conditions. Research should measure practical participation rather than count published tool definitions.

Dimension	Proposed measure	Interpretation safeguard
Language access	Task completion and error correction by preferred language	Evaluate intended use with community input
Human assistance	Time to an effective human handoff	Do not count a displayed phone number as a successful handoff
Provider participation	Integration burden by organizational size	Include maintenance and incident response
Accountability	Ability to inspect, challenge and reverse appropriate actions	Evaluate real disputes and recovery pathways

These measures are a proposed evaluation framework, not established findings about an existing service. Report baseline conditions, uncertainty, excluded populations and adverse results. A credible study can conclude that an intervention is useful, ineffective or inequitable; democratization is the question being tested, not a benefit assumed in advance.

References & evidence access

Numbered references identify source material supporting nearby factual statements. Interpretation, proposed architecture and labeled scenarios are original synthesis. Web sources were accessed for this review; the evidence cutoff is October 2, 2026. Report dates and underlying observation years are retained separately. No exhaustive database search or formal risk-of-bias meta-analysis was performed.

[1] HL7 International. FHIR Release 4: Overview. FHIR R4, version 4.0.1.

Primary technical specification. Version is explicit; this paper does not assert it is the newest FHIR release.

<https://hl7.org/fhir/R4/overview.html>

[2] HL7 International. SMART App Launch Implementation Guide. Version 2.2.0 consulted.

Primary technical specification. Authorization and application-launch context; deployed versions vary.

<https://hl7.org/fhir/smart-app-launch/>

[3] Model Context Protocol. Authorization. Specification revision 2025-11-25.

Primary protocol specification. Describes authorization for HTTP-based transports; not healthcare law or a clinical semantic standard.

<https://modelcontextprotocol.io/specification/2025-11-25/basic/authorization>

[4] Model Context Protocol. Security Best Practices. 2025-11-25 documentation path.

Primary security documentation. Protocol-related threats and mitigations; implementation must be tested locally.

https://modelcontextprotocol.io/docs/2025-11-25/tutorials/security/security_best_practices

[5] Cognizant. Cognizant Brings Agentic AI and MCP tool library to Core Claims Operations with Workflow Agentic Processing for TriZetto. September 28, 2026.

Primary vendor announcement, not an independent outcome evaluation. Announces 100+ MCP tools and availability for Facets and QNXT; performance results were described as forthcoming.

<https://news.cognizant.com/2026-09-28-Cognizant-Brings-Agentic-AI-and-MCP-tool-library-to-Core-Claims-Operations-with-Workflow-Agentic-Processing-for-TriZetto>

[6] Stedi. MCP server documentation. Product documentation accessed for 2026 review.

Primary vendor documentation. Evidence of an interface offering, not proof of Careverse integration or operational effectiveness.

<https://www.stedi.com/docs/healthcare/mcp-server>

[7] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. July 26, 2024.

NIST AI 600-1. DOI: 10.6028/NIST.AI.600-1. Voluntary risk-management guidance; not a certification.

<https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence>