IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 9, September 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.524

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

High-Throughput Clinical Context Extraction: Optimizing Straight-Through Ingestion via Unified Entity APIs and Asynchronous Caching

Neeraj Gaddam

Independent Researcher, Leander, Texas, USA

ABSTRACT: Before a specialty prescription can be cleared, the pharmacy must assemble its clinical context: patient, coverage, prescriber, drug, clinical documents and an eligibility determination, usually from separate systems of record. When the assembly is slow, incomplete or wrong, prescriptions fall out of straight-through processing and technicians look up and re-key data, which delays the time to fill, especially for multi-therapy regimens whose prescriptions must be ready together. This paper presents an ingestion architecture with boundary validation that repairs transcription errors only when a unique repair is supported by reference data; a unified entity API that resolves each case's dependency graph with deduplication, batching, deadlines and hedged requests; and an asynchronous cache with freshness classes, versioned change-data-capture invalidation, stale-while-revalidate for reference data and prefetch. A two-stage simulation, verified against closed-form results, compares it with per-prescription sequential and parallel access, a unified API without cache and one with a 60-minute TTL cache. At the reference load the proposed design needed 86% fewer backend calls per prescription than per-prescription fan-out, cut the 95th-percentile case latency from 795 to 350 ms, and sustained 3.8 times the event rate of fan-out before that latency exceeded one second. Its decision-critical stale reads were 3.9 per 10,000 prescription evaluations, against 155 for the TTL cache, at a hit ratio of 91%. Validators run on synthetic identifiers with injected errors detected every NPI error and repaired 98% of single substitutions without a wrong repair when the prescriber's name was checked; without that check, wrong repairs appeared. Binary schema-based encodings halved payload size. Workload and backend parameters are assumed rather than measured; in every sensitivity scenario the design made 4.3 to 9.8 times fewer backend calls per prescription than fan-out.

KEYWORDS: Specialty pharmacy, clinical context, entity API, request batching, hedged requests, caching, change data capture, stale-while-revalidate, data validation, serialization, straight-through processing

I. INTRODUCTION

Specialty pharmacies dispense medications for complex and often rare conditions, and much of their work happens before the first fill. Benefits must be verified, prior authorization obtained where the plan requires it, the prescription reviewed clinically and financial assistance arranged. The Pharmacy Quality Alliance measures specialty pharmacy turnaround time as the number of days from receipt of a new prescription to the point at which it is ready for pickup or scheduled for delivery [1]. In focus groups, specialty pharmacy staff named prior authorization and third-party payer issues, together with patient-communication and prescriber issues, as the main impediments to turnaround [2], and in a national survey of 1,001 physicians conducted in December 2022, 94% reported that prior authorization delays access to necessary care [3].

Each of these steps is a decision over data that the pharmacy must first assemble. This paper calls that data the clinical context of a prescription: the patient's demographics, the coverage on file, the prescriber, the drug product, the diagnosis and supporting clinical documents, and an eligibility and benefit determination for the drug under the patient's plan. The pieces usually live in different systems of record, each behind its own service. When the context can be assembled quickly, completely and correctly, rules can clear the prescription or route it to the right queue without human work, which is called straight-through processing (STP). When it cannot, a technician looks up the missing data, telephones, or re-keys fields from the referral image. Pharmacy technicians do most of this work and are hard to retain: in workforce surveys reported in 2022, a majority of pharmacy administrators described technician turnover of at least 21% in the previous year [4].

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

Regulation is moving these lookups toward real time. Since 2021, Medicare Part D plans have had to offer prescribers a real-time benefit tool that returns patient-specific formulary and cost information [5], and the Interoperability and Prior Authorization final rule (CMS-0057-F) requires impacted payers to support electronic prior authorization interfaces [6] based on HL7 FHIR [7] and the Da Vinci Prior Authorization Support guide [8]. Prescriptions themselves arrive as NCPDP SCRIPT messages [9], faxes and portal submissions. These standards make individual transactions electronic. They do not tell a pharmacy how to assemble the many transactions that one case needs, how to keep the assembled context fresh while coverage and authorization status change during benefits investigation, or how to stop transcription errors at the boundary before they reach adjudication.

Multi-therapy regimens make each of these problems larger. A patient starting a combination regimen is referred with several prescriptions, and the regimen can start only when all of them are ready. If context is extracted per prescription, patient-level data is fetched several times, and the case waits for the slowest of many independent calls, which is the tail-at-scale effect of fan-out [10]. Every later event on the case, such as an authorization decision or a corrected insurance card, triggers re-evaluation and repeats the work. Errors in identifiers compound the delay: data-entry errors in e-prescriptions are usually caught by technicians at data entry [11], and errors that escape are found later, when a claim or an eligibility check is rejected.

This paper presents and evaluates a vendor-neutral reference architecture for clinical context extraction. It generalizes design patterns used in large-scale specialty pharmacy ingestion and describes them without reference to the systems or data of any organization. Its contributions are:

- a formulation of clinical context extraction as the evaluation of a dependency graph over entity services under a time budget, with metrics for latency, backend load, freshness and straight-through completion within the budget (Section III);
- an architecture that combines boundary validation with minimal-edit repair, a unified entity API that resolves each case's dependency graph with deduplication, batching, deadlines and hedged requests, and an asynchronous cache with freshness classes, versioned change-data-capture invalidation, stale-while-revalidate for reference data and prefetch (Section IV);
- a two-stage simulation that separates cache dynamics over days from queueing over milliseconds, verified against closed-form results, which compares five access policies over a load sweep with ablations and a sensitivity analysis (Sections V and VI);
- measurements with working validators on synthetic identifiers with injected errors, using a published ICD-10-CM code set, and a benchmark of six encodings on clinical-context payloads (Section VI).

Section II reviews related work and Section III formulates the problem. Section IV describes the architecture and Section V the experimental method. Section VI reports the results, Section VII discusses implications and limitations, and Section VIII concludes.

II. BACKGROUND AND RELATED WORK

A. Specialty fulfillment and real-time eligibility

Turnaround time is a central quality measure for specialty pharmacies [1], and the barriers to it are mostly informational: missing or incorrect insurance data, authorization requirements and incomplete referrals [2]. Real-time benefit tools [5] and electronic prior authorization [6], [8] move eligibility information closer to the point of decision, so that a pharmacy can increasingly determine eligibility while the case is being assembled instead of days later. This raises the value of fast and fresh context: an eligibility result is useful only if the coverage it was computed from is still current.

B. Entity services and their composition

Data-integration theory distinguishes virtual integration, in which a query over a mediated schema is reformulated against the sources at run time, from materialized integration [12], [13]. Microservice architectures implement virtual integration through API composition, often behind an API gateway that offers each client a single endpoint [14], and derived data systems such as caches and search indexes are kept up to date from the systems of record [15]. GraphQL [16] makes the composition declarative: a query names the fields it needs and the server resolves them, and its semantics and complexity have been formalized [17]. How resolvers batch and deduplicate calls to the services behind them is left to the implementation, and that is where latency and load are decided.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

C. Tail latency

When a request fans out to many services, its latency is the maximum of theirs, so rare slow responses become common slow requests. Dean and Barroso [10] describe this effect and techniques to tolerate it, among them hedged requests, which send a second copy of a request after a delay and use whichever response arrives first. In context extraction the fan-out has two dimensions: across the entity types of one prescription, which have dependencies, and across the prescriptions of a case, which the regimen needs together.

D. Caching and freshness

HTTP caching defines a freshness lifetime after which a stored response must be revalidated [18], and the stale-while-revalidate extension allows a cache to serve a stale response while it revalidates in the background [19]. Jung, Berger and Balakrishnan derived the hit ratio of TTL-based caches under Poisson requests [20]. At large scale, Facebook's memcache deployment used invalidations driven from the database and leases that prevent stale sets and thundering herds [21], and TAO caches a social graph in front of a database with read-through and write-through semantics [22]. Change data capture streams the committed changes of a system of record to consumers [23], which can then invalidate or update derived copies [15]. Clinical context adds a twist: entities differ by orders of magnitude in how often they change and in the harm a stale copy can do, so one freshness rule cannot fit all of them.

E. Serialization

JSON [24] is the common format of web APIs and of FHIR resources [7]. MessagePack [25] and CBOR [26] are schemaless binary encodings of the JSON data model, while Protocol Buffers [27] and Avro [28] are schema-based and omit field names from the payload. Viotti and Kinneer [29] survey JSON-compatible binary formats and compare their sizes on a set of documents. Encoding affects processor time per request and, because caches hold encoded values, the memory a cache needs for a given hit ratio.

F. Validation of identifiers and codes

The National Provider Identifier (NPI) is a 10-digit number whose last digit is an ISO standard check digit computed with the Luhn algorithm as if the number were preceded by the prefix 80840 [30], [31], [32]. A National Drug Code (NDC) has 10 digits in one of three segment configurations, 4-4-2, 5-3-2 or 5-4-1, and HIPAA standard transactions use an 11-digit 5-4-2 form obtained by adding a leading zero to the appropriate segment; the Food and Drug Administration has proposed a uniform 12-digit format [33]. NDCs carry no check digit. ICD-10-CM diagnosis codes have three to seven characters, only the most specific codes of a category are billable, and the code set is revised every year [34], so a code that is billable in one release may be split into more specific codes in a later one. Faxed referrals add errors from optical character recognition, whose correction is the subject of a large literature [35]. Check digits detect many keying errors, but no single layer detects all of them, and the cost of an undetected error is paid later, at adjudication.

G. Positioning

Each line of work supplies a part: composition interfaces, tail-tolerance techniques, cache consistency mechanisms, compact encodings and check digits. This paper combines them for clinical context, where dependencies between entities, the per-case structure of multi-therapy regimens, entity-specific freshness requirements and noisy identifiers interact, and evaluates the combination quantitatively. Table I summarizes the capabilities each approach provides.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

TABLE I. CAPABILITIES PROVIDED BY RELATED APPROACHES

Approach	Per-case batching	Tail tolerance	Freshness control	Entity-specific policy	Input validation
API composition and gateways [14]	possible	not addressed	not addressed	no	no
GraphQL [16], [17]	implementation-defined	not addressed	not addressed	no	schema types
Hedged requests [10]	n/a	yes	n/a	n/a	n/a
HTTP caching and SWR [18], [19]	n/a	partly (stale serving)	TTL	per response	no
Invalidation and CDC [21], [23]	n/a	n/a	event-driven	possible	no
Check digits [30], [31]	n/a	n/a	n/a	n/a	detection only
This paper	yes (dependency graph)	hedging, deadlines	CDC + safety TTL, SWR	freshness classes	layered, with repair

III. PROBLEM FORMULATION

A case c holds the prescriptions of one patient's referral, indexed $j = 1, \ldots, K$. Its clinical context consists of entity instances of six types: the patient p; the coverage and the clinical documents of p; the prescriber r_j and the drug product d_j of each prescription; and an eligibility determination for each pair (p, d_j). Each type e is served by a backend with service time S_e and m_e parallel servers. Instances depend on one another: coverage and clinical documents are found through the patient, and eligibility needs the coverage and the drug. These dependencies form a directed acyclic graph G, and a context request completes when every node of its graph instance is resolved. If node v takes D_v once its predecessors are resolved, the request completes at

$$T = \max_{v \in G} \tau_v, \qquad \tau_v = \max_{u \in \mathrm{pred}(v)} \tau_u + D_v \tag{1}$$

where D_v is a cache-hit latency η if the instance is cached, and otherwise the waiting time W_e plus the service time S_e of a backend call, plus network time. The maximum over parallel branches is what makes tails grow with fan-out. If n independent calls have latency distributions F_i, then

$$P\left(\max_i X_i > t\right) = 1 - \prod_{i=1}^{n} F_i(t) \tag{2}$$

and if each prescription of a K-prescription case is processed separately and misses a budget B with probability q, the case misses it with probability $1 - (1 - q)^K$. For $q = 2\%$ and $K = 4$ this is 7.8%.

Load matters as much as latency. Let λ be the rate of context requests, which include the evaluation at release and every re-evaluation, let n_e be the number of backend calls to type e per request, after deduplication and caching, and let b_e count background calls (prefetch and revalidation). The utilization of backend e is

$$\rho_e = \frac{\lambda\,(\mathbb{E}[n_e] + \mathbb{E}[b_e])\,\mathbb{E}[S_e]}{m_e} \tag{3}$$

and waiting times grow without bound as ρ_e approaches one. Throughput is therefore limited by the busiest backend, and anything that reduces n_e raises the sustainable request rate.

Caching trades load for freshness. A cached instance fetched at time f carries the source version v_f; read at time t, it is stale if the source has changed since:

$$\mathrm{stale}(t) = \mathbb{1}[V(t) > v_f] \tag{4}$$

where $V(t)$ is the version of the system of record at t. Stale reads of coverage, clinical documents and eligibility are called decision-critical, because they can change an automated decision. The objective is to maximize the share of prescriptions whose context is complete within the budget,

$$S(B) = P(T \le B \ \wedge \ \text{context complete}) \tag{5}$$

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

subject to the backend capacities and to a bound on decision-critical stale reads. A context is incomplete when a backend call fails twice or misses its deadline twice; the prescription then needs manual lookup. Errors in the identifiers themselves are handled at the boundary (Section IV-A) and are evaluated separately. Table II lists the notation.

TABLE II. NOTATION

Symbol	Meaning
K, G	prescriptions in a case; dependency graph of entity types
S_e, W_e, m_e	service time, waiting time and servers of backend e
η, T, B	cache-hit latency; request latency; latency budget
λ, ρ_e	context-request rate; utilization of backend e
n_e, b_e	critical-path and background backend calls per request
τ, γ, δ	freshness lifetime (TTL); stale-while-revalidate grace; CDC delivery delay
v, v_f, $V(t)$	update rate of a source instance; cached version; source version
$S(B)$	share of prescriptions with complete context within B

IV. PROPOSED ARCHITECTURE

Fig. 1 shows the architecture. Cases released by intake, and every later event that requires re-evaluation, enter through a boundary-validation layer. Validated requests go to a unified entity API, which evaluates the case's dependency graph against an asynchronous cache and calls the systems of record only on misses. A change-data-capture (CDC) stream from the systems of record invalidates cached entries by version, and first-document events, which precede the release of a case, trigger prefetch. Eligibility and STP rules then decide on the assembled context whether the case proceeds without human work. Fig. 2 illustrates the effect of these choices on one three-prescription case.

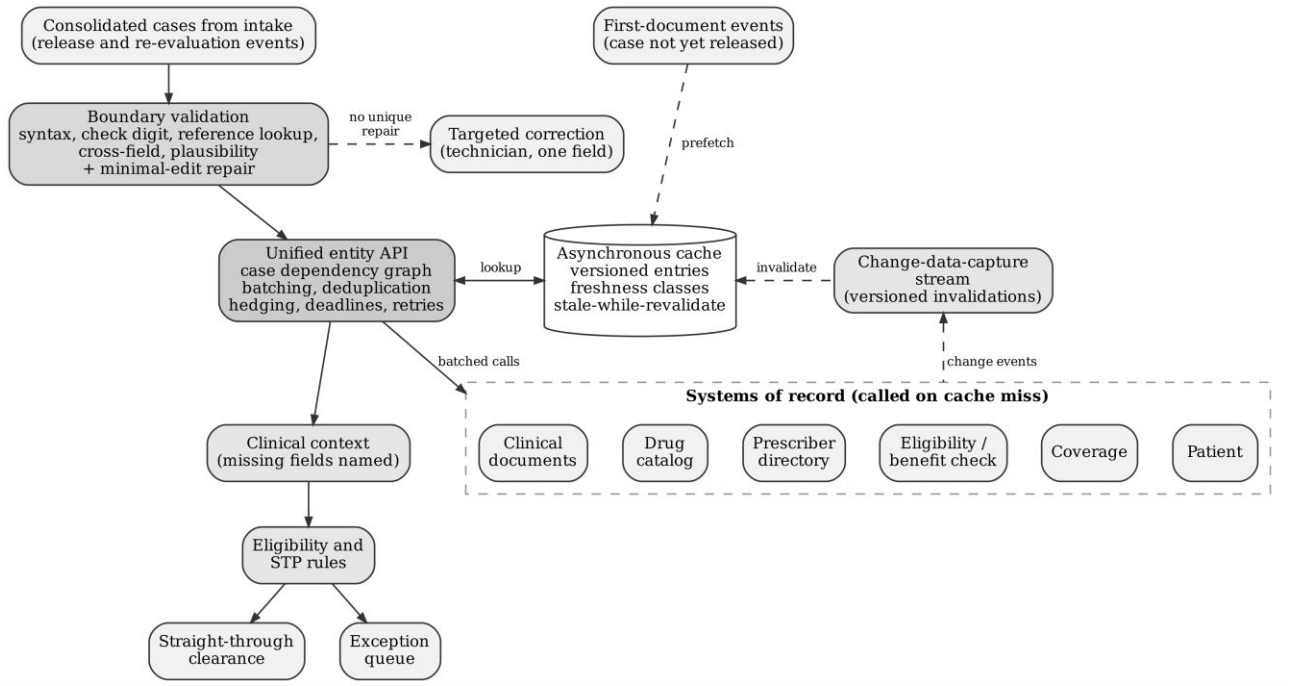


Fig. 1. Reference architecture for clinical context extraction. Dashed arrows show asynchronous paths: prefetch, invalidation and routing to targeted correction.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

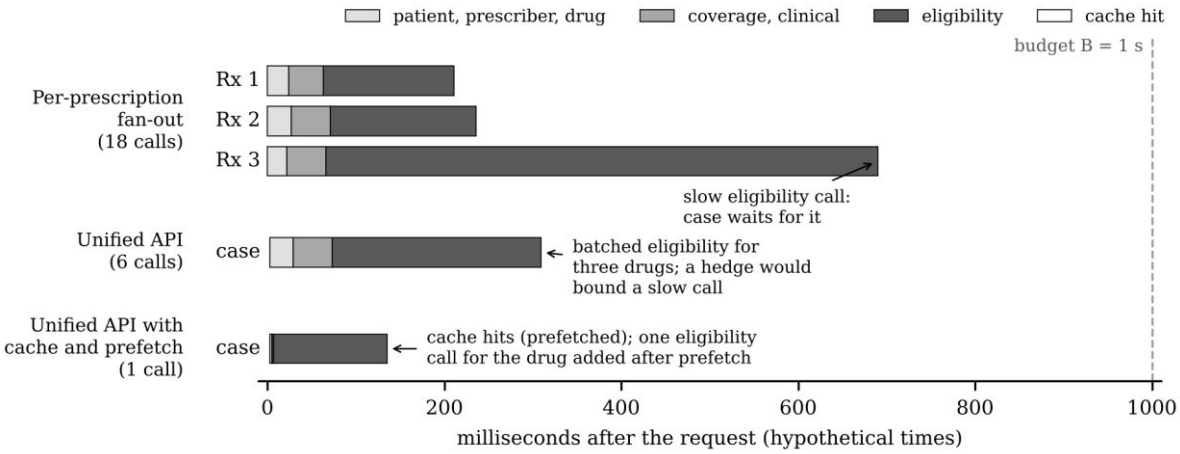


Fig. 2. Schematic execution of one three-prescription case with hypothetical times. Per-prescription fan-out issues 18 calls and waits for the slowest; the unified API issues six; with cache and prefetch, one call remains.

A. Boundary validation and repair

Every identifier and code is checked in five layers before any lookup uses it. Syntax checks enforce length, character classes and segment patterns, and normalize representations: an NDC written as 10 digits with hyphens is converted to the 11-digit 5-4-2 form, and the decimal point of an ICD-10-CM code is removed. Check digits are verified where they exist, as for the NPI. Reference checks confirm that the value exists in an authoritative list: the provider registry for an NPI, the product catalog for an NDC, and the code set of the applicable release for a diagnosis, including whether the code is billable. Cross-field checks compare the reference entry with other fields of the same document: the prescriber's name for an NPI, and the drug name and strength for an NDC. Plausibility checks test the combination, for example whether the diagnosis belongs to the therapy class of the drug.

When a value fails, the layer tries to repair it before routing it to a person. Candidates are generated in order of edit cost: first other readings of the characters as written (an unhyphenated 10-digit NDC has three possible 11-digit forms), then single-character substitutions, adjacent and jump transpositions, insertions and deletions. A candidate survives if it passes every layer, including the cross-field check against the document. With $C_k(x)$ the surviving candidates at edit cost k, the repair rule is

$$\hat{x} = \begin{cases} y & \text{if } C_{k^*}(x) = \{y\},\ k^* = \min\{k : C_k(x) \neq \emptyset\} \\ \text{route to a person} & \text{otherwise} \end{cases} \tag{6}$$

so a repair is applied only when exactly one candidate survives at the lowest cost that has any. For diagnosis codes, where the document offers little independent evidence, repairs are proposed to the technician instead of applied. Every repair is logged with the original value. Routed values go to a targeted-correction task that shows the field, the image and any candidates, which takes far less time than re-keying the prescription. Algorithm 1 summarizes the procedure.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

Algorithm 1. Boundary validation with minimal-edit repair

```
validate(x, doc, cls):
    for y in normalizations(x, cls):          # zero edits
        if passes_all(y, doc, cls): return accept(y)
    for k in 1 .. k_max:
        C <- {y in edits(x, k, cls) : passes_all(y, doc, cls)}
        if |C| == 1 and cls.auto_repair: return repair(x, y in C)   # logged
        if |C| >= 1: return route(x, candidates = C)
    return route(x, candidates = {})
passes_all(y, doc, cls):
     return syntax(y) and check_digit(y) and in_reference(y)
            and cross_field(y, doc) and plausible(y, doc)
```

B. Unified entity API

The unified API accepts one request per case and builds the case's graph instance: one patient, coverage and clinical node; one prescriber node for the distinct prescribers; a drug node for the distinct drug products; and one eligibility node for the *K* patient-drug pairs. Deduplication removes repeated patient-level lookups, and batching turns the drug and eligibility lookups of a regimen into one call each. A batched call takes longer than a single one, but less than the calls it replaces, and batches above a configured size are split into chunks issued in parallel, so that the expected latency of every call stays well inside its deadline. Nodes start as soon as their predecessors resolve, so independent branches run in parallel.

Each backend call has a deadline. A call that fails or misses its deadline is retried once after a short backoff; a second failure marks the node and its dependants missing, and the context is returned with explicit gaps instead of waiting indefinitely. A call that has not returned after the 95th percentile of its expected service time, which grows with the batch size, is hedged: a second copy is sent and the first response wins [10]. Copies still queued when the call resolves are canceled, and queued calls whose deadline has passed are dropped, so a backlog does not consume capacity on work nobody waits for. Table III lists the entity types and their treatment.

TABLE III. ENTITY TYPES, DEPENDENCIES AND FRESHNESS CLASSES IN THE PROPOSED DESIGN

Entity	Key	Depends on	Freshness class	Cache rule
Patient	patient	none	patient-scoped	CDC, safety TTL 12 h
Coverage	patient	patient	patient-scoped, decision-critical	CDC, safety TTL 12 h
Clinical documents	patient	patient	patient-scoped, decision-critical	CDC, safety TTL 12 h
Prescriber	NPI	none	reference	TTL 24 h, SWR 24 h, CDC
Drug product	NDC	none	reference	TTL 24 h, SWR 24 h, CDC
Eligibility	patient, NDC	coverage, drug	decision-critical	CDC on coverage, drug and authorization events; safety TTL 12 h

C. Asynchronous cache with freshness classes

The cache distinguishes two classes of data. Reference data, the drug catalog and the prescriber directory, is shared across patients and changes rarely; it is kept for 24 hours and served stale for up to another 24 hours while it is revalidated in the background [19], so that expiry never adds a backend call to the critical path. Patient-scoped data, including coverage, clinical documents and eligibility, changes during the life of a case, for instance when benefits investigation finds a different plan or an authorization is decided. It is kept until invalidated by the CDC stream, with a safety TTL of 12 hours that bounds the damage of a lost invalidation. Eligibility entries, keyed by patient and drug, are invalidated by coverage changes, drug-catalog changes and authorization decisions.

Entries carry the version of the source record they were read from. An invalidation for version *v* evicts an entry only if the entry's version is older than *v*, so a delayed or reordered invalidation cannot evict a fresher entry. Concurrent misses

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

on the same key are coalesced into one backend call, which prevents thundering herds after an eviction [21]. When intake sees the first document of a case, before the case is released, the cache prefetches the patient-level entities and those of the first prescription in the background. Algorithm 2 gives the lookup.

Algorithm 2. Versioned cache lookup with CDC invalidation and stale-while-revalidate

```
lookup(key, t):
    ent <- cache.get(key)
    if ent is None: return fetch(key, t)          # miss: one coalesced call
    if cls(key).cdc and invalidated_after(key, ent.version, t):
        return fetch(key, t)                      # evicted by a newer version
    age <- t - ent.fetched
    if age <= cls(key).ttl: return ent.value      # fresh by policy
    if age <= cls(key).ttl + cls(key).grace:
        refresh_async(key)                        # stale-while-revalidate
        return ent.value
    return fetch(key, t)
on_change(key, v):                                # from the CDC stream
    if cache.get(key) and cache.get(key).version < v: cache.evict(key)
```

The freshness of a TTL cache can be predicted. For an entry whose lifetime starts at a miss and is not extended by hits, with requests arriving as a Poisson process of rate λ and source updates as a Poisson process of rate ν, the hit ratio [20] and the probability that a hit is stale are

$$h = \frac{\lambda\tau}{1+\lambda\tau}, \qquad P(\text{stale} \mid \text{hit}) = 1 - \frac{1-e^{-\nu\tau}}{\nu\tau} \tag{7}$$

The second expression follows because the hits of one lifetime are uniformly distributed over $(0, \tau]$. A longer TTL raises the hit ratio and the staleness together, which is the dilemma CDC removes: with invalidation, an entry is stale only between a change and the delivery of its invalidation, so for small $\nu\delta$ the stale fraction is roughly $\nu\mathrm{E}[\delta]$, independent of τ, plus a term for lost invalidations that the safety TTL bounds.

A. Serialization and payloads

Inside the architecture, between the unified API, the cache and the backends, payloads use a schema-based binary encoding; at the external boundary, where FHIR and partner interfaces apply, they use JSON. A schema gives every field a stable identifier and a default, so producers and consumers can evolve independently, and it omits field names from each payload, which shrinks cached values. Section VI-G quantifies the choice.

B. Eligibility rules and missing data

Eligibility and STP rules read the assembled context: coverage active on the date of service, the drug covered under the plan, authorization on file when the plan requires it, the diagnosis consistent with the therapy, and quantity within limits. A context with missing fields is never cleared automatically. It is routed with the missing fields named, so that the technician knows exactly what to look up instead of re-checking the whole case. Decision-critical fields are read from the cache only under the CDC rule above; Section VI-A measures how often such a read is nevertheless stale.

C. Correctness and fault tolerance

Three properties keep the design safe under failure. The cache is never authoritative: an entry can be lost or evicted at any time without changing any decision, only its latency. Invalidations are idempotent and versioned, so redelivery and reordering are harmless, and a lost invalidation is bounded by the safety TTL. Retries, hedges and refreshes are idempotent reads. The validation layer never changes a value silently: repairs are applied only under the uniqueness rule of Eq. (6), logged with the original, and limited to identifier classes for which the experiments in Section VI-H found no wrong repairs.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

V. EXPERIMENTAL METHOD

A. Two-stage simulation

The behavior of the architecture depends on processes at two time scales. Whether a lookup hits the cache, and whether the cached value is stale, depends on request and update patterns over minutes to days. Whether a backend call waits, times out or is hedged depends on queueing over milliseconds. Simulating both in one model would require days of simulated time at millisecond resolution. Because the two scales are far apart, the system is nearly decomposable [36], and the study uses two stages. Stage A replays eight days of synthetic case activity against each cache policy at the level of individual keys and versions, discards the first day as warm-up [37], and records for every request which lookups hit, which were stale and which went to a backend. Stage B replays these request records, in their original order, through a discrete-event model of the backends at a chosen request rate and measures latency, failures and load. Stage B therefore inherits the cache outcomes of stage A, including the correlation between lookups of the same request, while the request rate can be varied freely.

In stage A, cases arrive as a Poisson process at 3,000 per day, with the episode-size distribution of Table IV. Sixty percent of cases belong to one of 60,000 existing patients; prescribers and drug products are drawn from Zipf popularity distributions over 15,000 prescribers and 1,500 products. Each case generates a first-document event, a release event and a Poisson number of re-evaluations, and each record in the systems of record changes according to its own process: rarely for reference data, and more often during the life of a case for coverage, clinical documents and eligibility, which also changes when an authorization is decided. Every change publishes a versioned invalidation with a log-normal delivery delay, and a small fraction of invalidations is lost.

Table IV and Table V list the baseline inputs of the two stages. They are assumed values chosen to be plausible, not measurements from any pharmacy or production system, and Section VI-F varies them.

TABLE IV. STAGE A WORKLOAD AND CHANGE PROCESSES (BASELINE)

Parameter	Value
Cases per day; days per replication	3,000; 8 (1 warm-up)
Prescriptions per case $K = 1, \ldots, 6$	0.45, 0.25, 0.15, 0.08, 0.04, 0.03
Existing patients; share of cases	60,000; 60%
Prescribers; products (Zipf exponent)	15,000 (1.0); 1,500 (1.1)
Same prescriber for all prescriptions	85%
First document before release	Exp(mean 15 min)
Re-evaluations per case; gaps	Poisson(3); Exp(mean 2 h)
Background change rates	drug 1/30 d, prescriber 1/180 d, patient 1/120 d, coverage 1/60 d, clinical 1/30 d
Changes during a case	patient 20% (mean 4 h); coverage 35% (mean 6 h); clinical documents 1 per 8 h for 48 h
Authorization decisions	30% of prescriptions, LogN(median 24 h, $\sigma = 0.8$)
CDC delivery delay; lost invalidations	LogN(median 2 s, $\sigma = 1.0$); 0.1%

In stage B, each entity type has a backend with a fixed number of servers and log-normal service times; one call in 100 is 8 times slower, which produces the occasional slow responses that dominate tails in practice [10]. A batched call with n items takes $1 + 0.3(n - 1)$ times as long as a single call, and the unified API splits eligibility batches into chunks of at most 2 items. Calls fail transiently with probability 0.2%, have a deadline of 1 s and are retried once after 10 ms. Network hops take 1 ms, a cache hit 0.5 ms and the unified API adds 2 ms of processing. Table V lists the values. Server counts are scaled down so that the simulation remains tractable; only ratios between policies, and loads relative to capacity, are meaningful, so loads are reported in requests per second of this scaled system.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

TABLE V. STAGE B BACKEND MODEL (BASELINE)

Backend	Median service time (ms)	σ (log-normal)	Servers
Patient	20	0.5	4
Prescriber	15	0.5	2
Drug	10	0.5	2
Coverage	35	0.5	6
Clinical	60	0.5	8
Eligibility	150	0.7	22

B. Policies

Five policies are compared. The first two extract context per prescription without a cache, as integrations that treat the prescription as the unit of work do: sequentially, one entity after another, or in parallel by dependency level. The third uses the unified API per case, with deduplication, batching and hedging, but no cache. The fourth adds a cache with a single TTL of 60 minutes for all entity types, a common default. The fifth is the proposed design, with the freshness classes of Table III, CDC invalidation and prefetch. Ablations remove hedging, prefetch or stale-while-revalidate from the proposed design, remove hedging from the unified API, or replace CDC by short per-class TTLs (patient 1 h; coverage and clinical documents 15 min; eligibility 5 min; reference data 24 h with stale-while-revalidate).

C. Metrics and statistics

For each prescription, latency is measured from the request to the moment its context is complete or known to be incomplete; for a case, it is the latest of its prescriptions. The budget is $B = 1$ s. Backend calls per prescription count every call, including retries, hedges and background calls. The reference load is 30 requests per second, at which the busiest backend of per-prescription fan-out is about 60% utilized. Each configuration at the reference load is replicated 20 times with independent stage-A worlds, and each point of the load sweep five times. Policies are compared on the same request records and with paired Wilcoxon signed-rank tests [38] with Holm correction [39]. Means are reported with 95% confidence intervals.

D. Verification

Both stages were checked against closed-form results. For stage A, a single key with Poisson requests and updates was simulated with the cache code for 12 combinations of $\lambda\tau$ and $v\tau$; the hit ratio and the stale share of hits agree with Eq. (7) (Table VI), with a largest hit-ratio difference of 0.0016. For stage B, the backend model itself, with exponential service, no errors, no hedging and zero network time, was run as an M/M/c queue and compared with the Erlang C formula for the mean wait [40]; in all three cases the formula lies within the 95% confidence interval of ten replications.

TABLE VI. STAGE A CACHE MODEL AGAINST THE CLOSED FORMS OF EQ. (7) (200,000 REQUESTS PER CELL)

$\lambda\tau$	$v\tau$	Hit ratio, formula	Hit ratio, simulated	P(stale | hit), formula	P(stale | hit), simulated
0.25	0.1	0.2000	0.1984	0.0484	0.0475
0.25	1	0.2000	0.1992	0.3679	0.3634
1	0.1	0.5000	0.5005	0.0484	0.0478
1	1	0.5000	0.4997	0.3679	0.3655
4	0.1	0.8000	0.8002	0.0484	0.0471
4	1	0.8000	0.7997	0.3679	0.3652
16	0.1	0.9412	0.9413	0.0484	0.0490
16	1	0.9412	0.9411	0.3679	0.3731

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

TABLE VII. STAGE B QUEUE AGAINST THE ERLANG C FORMULA (EXPONENTIAL SERVICE, MEAN 10 MS; TEN REPLICATIONS)

Servers *c*	Utilization	Erlang C mean wait (ms)	Simulated mean wait (ms), 95% CI
2	0.7	9.608	9.408 ± 0.268
6	0.8	4.315	4.377 ± 0.087
22	0.9	2.422	2.467 ± 0.080

E. Validation experiment

The validators of Section IV-A were implemented and applied to synthetic identifiers with injected errors, 20,000 per error type. NPIs were generated with valid check digits. Because a real provider registry cannot be used here, registry membership of any other valid number was decided by a keyed hash so that a chosen share of all valid numbers, the registry density, is in use; densities of 0.1%, 1% and 4% were tested. Each registered number has a synthetic prescriber name drawn from a Zipf distribution over 100,000 names, and the name on the document carries a typing error in 10% of cases. A synthetic NDC catalog of 122,659 packages in the three FDA configurations [33] was generated, each with a drug name and strength. Diagnosis codes were drawn from therapy-specific blocks of the 2021 ICD-10-CM code set [34], read from the order file distributed with the simple_icd_10_cm package [41]; plausibility requires the code to belong to the therapy class of the drug. Error types are single substitutions, adjacent and jump transpositions, twin errors (two equal adjacent digits replaced by another equal pair), optical-character-recognition confusions between similar characters, deletions, insertions, shifted NDC hyphens, missing NDC hyphens and truncated or re-punctuated diagnosis codes.

F. Serialization benchmark

Six encodings were benchmarked on synthetic clinical-context records for cases with one, two and four prescriptions: JSON with the Python standard library and with orjson 3.9.10, MessagePack 1.0.5, CBOR (cbor2 5.4.6), raw Avro binary encoding without the object-container header (fastavro 1.8.2; the schema is known to both sides) and Protocol Buffers 4.21.12 with its C implementation. Every encoding was checked to round-trip the record exactly. Times are the minimum over 25 interleaved repetitions of the mean over 1,000 encodings or decodings of 50 different records, which limits interference from other processes; sizes are means over 200 records, also after zstd compression at level 3. The benchmark ran under Python 3.11.4 on an x86-64 Linux workstation (8-core, 32 GB RAM).

VI. RESULTS

A. Cache behavior and freshness

Table VIII reports stage A over ten worlds. Without a cache, the unified API needs 3.90 backend items per prescription, because patient-level lookups are shared by the prescriptions of a case; per-prescription extraction needs six. A single TTL of 60 minutes hits 43% of lookups, but 155 of every 10,000 prescription evaluations read stale coverage, clinical or eligibility data, mostly because coverage and authorization status change during benefits investigation, exactly when cases are re-evaluated. Short per-class TTLs reduce those stale reads to 9.1 per 10,000, but the volatile entities then hit only 9% (coverage) and 3% (eligibility) of the time. The proposed design hits 91% of lookups, needs 0.89 backend items per prescription and reads stale decision-critical data 3.9 times per 10,000 evaluations. Keeping its TTLs but removing CDC raises that figure to 2,893, which shows that the long lifetimes are safe only because of invalidation.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

TABLE VIII. STAGE A RESULTS (MEAN OF 10 WORLDS): HIT RATIO (%) OVERALL AND FOR PATIENT, COVERAGE, CLINICAL DOCUMENTS, ELIGIBILITY, PRESCRIBER AND DRUG; BACKEND ITEMS PER PRESCRIPTION; DECISION-CRITICAL STALE READS PER 10,000 EVALUATIONS

Policy	All	Pat.	Cov.	Clin.	Elig.	Presc.	Drug	Items / Rx	Critical stale / 10,000
No cache	0.0	0	0	0	0	0	0	3.90	0.0
Single TTL 60 min	43.0	26	26	26	26	57	77	2.22	155.3
Per-class TTL, no CDC	43.7	26	9	9	3	94	99	2.24	9.1
Proposed	91.2	94	93	84	80	99	99	0.89	3.9
Proposed without prefetch	78.2	69	68	59	68	93	99	0.89	3.9
Proposed without SWR	90.5	94	93	84	80	97	98	0.89	3.9
Proposed without CDC	94.7	97	97	97	84	99	100	0.76	2,892.6

Fig. 3 places these policies on the trade-off curve of a single TTL. Lengthening the TTL from 1 minute to 24 hours raises the hit ratio from 8% to 83% and the decision-critical stale reads from 0.07 to 3,418 per 10,000 evaluations. No single TTL reaches the hit ratio of the proposed design without far more stale reads: a TTL short enough to match its staleness, about 5 minutes, hits only 16% of lookups. Prefetch does not reduce the number of backend items, which stays at 0.89 per prescription, but it moves them off the critical path: without prefetch, the hit ratio seen by requests falls from 91% to 78%. Stale-while-revalidate matters only when reference data expires and raises the prescriber and drug hit ratios by one to two points.

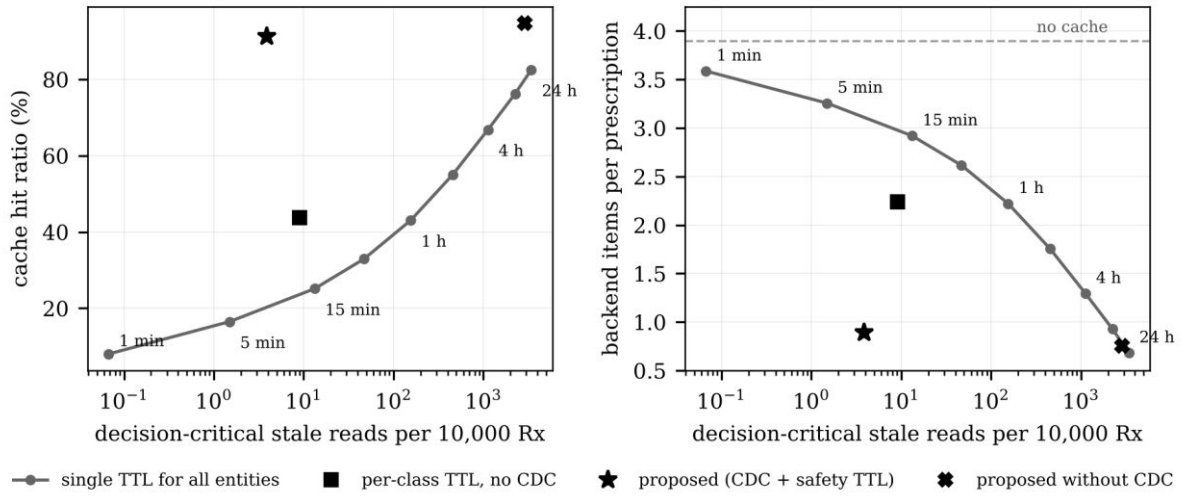


Fig. 3. Hit ratio (left) and backend items per prescription (right) against decision-critical stale reads for a single TTL from 1 minute to 24 hours, compared with per-class TTLs and with the proposed design with and without CDC. Means of 10 worlds.

B. Latency and throughput

Table IX compares the policies at the reference load. Per-prescription fan-out completes the median prescription in 243 ms, faster than the unified API without cache (309 ms), because a batched eligibility call for a whole regimen takes longer than a single one and every prescription of a case waits for the case. At the case level the order reverses: the 95th-percentile case latency is 795 ms for fan-out and 691 ms for the unified API, which also needs 46% fewer backend calls. The 60-minute TTL cache lowers calls further, to 2.25 per prescription, and the proposed design to 0.93, 86% fewer than fan-out. Its median prescription is served in 6 ms, because most re-evaluations find their entire context in the cache, and its 95th-percentile case latency is 350 ms. Sequential extraction is the slowest policy at every percentile.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

TABLE IX. STAGE B RESULTS AT 30 REQUESTS PER SECOND (MEAN OF 20 REPLICATIONS)

Policy	Rx p50 (ms)	Rx p95 (ms)	Rx p99 (ms)	Case p95 (ms)	Within 1 s (%)	Incomplete / 1,000 Rx	Calls / Rx	Busiest backend (%)
Sequential, per prescription	371	776	1,263	951	97.94	0.14	6.65	60
Parallel fan-out, per prescription	243	623	1,103	795	98.73	0.14	6.65	60
Unified API, no cache	309	759	956	691	99.33	0.06	3.58	46
Unified API + TTL 60 min	242	706	916	640	99.53	0.00	2.25	34
Proposed	6	425	723	350	99.90	0.00	0.93	16

Fig. 4 sweeps the request rate. Every policy keeps its latency almost constant until its busiest backend approaches saturation, and then fails abruptly: queued calls miss their deadlines, retries add load, and almost no context completes. The largest tested rate at which the 95th-percentile case latency stays within 1 s is 30 requests per second for sequential extraction, 40 for fan-out, 55 for the unified API, 70 with the 60-minute TTL and 150 for the proposed design, 3.8 times the rate of fan-out. The ratio follows the utilization of the busiest backend (Eq. (3)): at the reference load it is 60% for fan-out and 16% for the proposed design. The gain in throughput is smaller than the reduction in calls because the calls that remain are mostly batched eligibility calls, the most expensive kind.

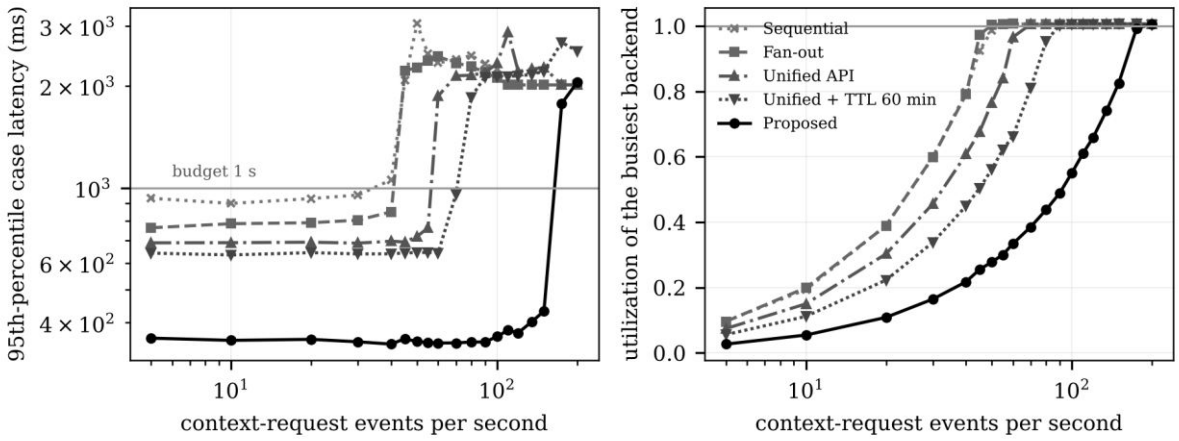


Fig. 4. 95th-percentile case latency (left, log scale) and utilization of the busiest backend (right) against the request rate. Means of five replications; the horizontal line on the left marks the 1 s budget.

C. Multi-therapy regimens

Fig. 5 (left) breaks case latency down by the number of prescriptions. Under fan-out, the 95th-percentile case latency rises from 615 ms for one prescription to 1,149 ms for 6, because the case waits for the slowest of its independent requests (Eq. (2)). Under the unified API it rises more slowly, from 580 to 864 ms, and under the proposed design from 181 to 570 ms. The gap between fan-out and the unified API widens from 36 ms for one prescription to 285 ms for 6, so regimens with several drugs, which must be ready together, gain the most from case-level extraction. Fig. 5 (right) shows the share of prescriptions whose context is complete within a budget B: within 250 ms it is 52% for fan-out, 34% for the unified API and 87% for the proposed design; within 1 s it is 98.7%, 99.3% and 99.9%.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

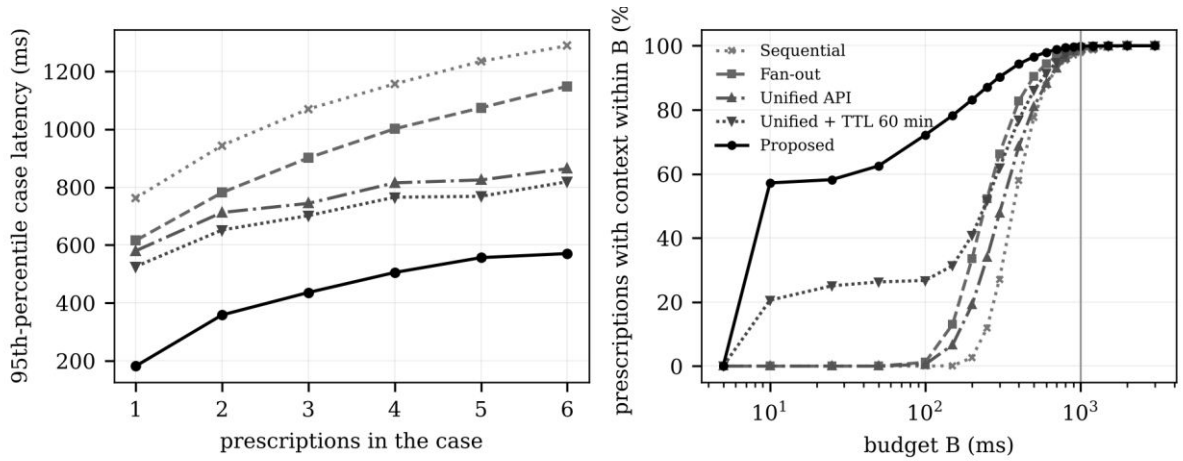


Fig. 5. Left: 95th-percentile case latency by number of prescriptions. Right: share of prescriptions with complete context within a budget B. Reference load, means of 20 replications.

D. Ablation

Table X removes one component at a time at the reference load. Hedging matters for the tail: without it the 99th-percentile latency of the proposed design rises from 723 to 838 ms, and that of the uncached unified API from 956 to 1,313 ms, at a cost of 0.019 and 0.258 extra calls per prescription. Prefetch matters for first evaluations: without it the 95th-percentile latency rises from 425 to 540 ms and the 95th-percentile case latency from 350 to 482 ms, because release events, whose entities have not been seen before, must then fetch everything on the critical path; the median, set by re-evaluations, is 6 ms either way. Replacing CDC by short per-class TTLs keeps staleness low (Table VIII) but raises calls per prescription to 2.25 and the 95th-percentile case latency to 678 ms.

TABLE X. ABLATION AT 30 REQUESTS PER SECOND (MEAN OF 20 REPLICATIONS)

Configuration	Rx p50 (ms)	Rx p95 (ms)	Rx p99 (ms)	Case p95 (ms)	Calls / Rx	Hedges / Rx	Within 1 s (%)
Proposed (all components)	6	425	723	350	0.93	0.019	99.90
Without hedging	6	435	838	358	0.92	0.000	99.36
Without prefetch	6	540	813	482	0.92	0.053	99.78
Without stale-while-revalidate	6	433	720	352	0.93	0.022	99.91
Per-class TTL instead of CDC	293	744	940	678	2.25	0.131	99.38
Unified API, no cache	309	759	956	691	3.58	0.258	99.33
Unified API, no cache, no hedging	314	848	1,313	764	3.34	0.000	96.85

E. Invalidation delay, lost invalidations and the safety TTL

Table XI varies the CDC stream. With no lost invalidations, the proposed design reads stale decision-critical data only 0.58 times per 10,000 evaluations, and the median stale read is 2 s old: the window between a change and its invalidation. At the reference loss rate of 0.1%, lost invalidations dominate, and their stale reads persist until the safety TTL expires. Losses of 1% and 5% raise the figure to 28 and 140, so the completeness of the CDC stream, not its speed, is the property to monitor. A median delay of 200 s raises stale reads to 68 per 10,000. The safety TTL trades hits for the residual staleness of lost invalidations: 1 hour gives a hit ratio of 65% and 0.36 stale reads per 10,000; 12 hours gives 91% and 3.35; beyond 24 hours little changes.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

TABLE XI. SENSITIVITY OF THE PROPOSED CACHE TO THE CDC STREAM AND THE SAFETY TTL (MEAN OF 5 WORLDS)

Variant	Hit ratio (%)	Backend items / Rx	Critical stale / 10,000 Rx	Median stale age (min)	p95 stale age (h)
Median delay 0.2 s	91.2	0.89	2.84	151.4	7.1
Median delay 2 s (reference)	91.2	0.89	3.35	114.5	7.0
Median delay 20 s	91.2	0.89	9.97	0.8	5.2
Median delay 200 s	91.3	0.89	67.99	3.8	0.7
No lost invalidations	91.2	0.89	0.58	0.0	0.0
1% lost	91.2	0.89	27.84	153.2	8.3
5% lost	91.4	0.89	139.88	156.4	9.0
Safety TTL 1 h	65.5	1.90	0.36	3.2	0.3
Safety TTL 4 h	82.6	1.23	1.38	21.2	1.9
Safety TTL 12 h (reference)	91.2	0.89	3.35	114.5	7.0
Safety TTL 24 h	92.3	0.84	3.91	175.5	10.8
Safety TTL 72 h	92.3	0.84	4.09	198.7	21.7

F. Sensitivity

Table XII repeats the main comparison under altered assumptions, ten replications each. The proposed design needs between 4.3 and 9.8 times fewer backend calls per prescription than fan-out in every scenario; the advantage is smallest with one re-evaluation per case and with mostly single-prescription cases, where fewer lookups can be shared or found in the cache. Its 95th-percentile case latency ranges from 281 to 678 ms, within the budget in every scenario. Fan-out exceeds the budget with three times as many slow calls, with eligibility median 300 ms and with 70% of backend servers; its busiest backend saturates with eligibility median 300 ms and with 70% of backend servers, where retries inflate its call count. The hit ratio of the cache ranges from 90% (one re-evaluation per case) to 92% (mostly single-prescription cases), and decision-critical stale reads from 1.7 to 4.3 per 10,000 evaluations.

TABLE XII. SENSITIVITY AT 30 REQUESTS PER SECOND (10 REPLICATIONS PER SCENARIO)

Scenario	Case p95, fan-out (ms)	Case p95, proposed (ms)	Calls / Rx, fan-out	Calls / Rx, proposed	Hit ratio, proposed (%)	Critical stale / 10,000, proposed	Busiest backend (%), fan-out / proposed
Reference assumptions	800	350	6.64	0.94	91	3.9	60 / 16
Three times as many slow calls	1,108	372	6.67	0.94	91	3.9	68 / 19
Eligibility median 300 ms	2,187	678	8.79	0.96	91	3.9	101 / 34
Transient errors 1%	797	355	6.75	0.94	91	3.9	60 / 17
70% of backend servers	2,282	353	9.22	0.94	91	3.9	100 / 24
One re-evaluation per case	794	408	6.64	1.55	90	1.7	60 / 28
Six re-evaluations per case	794	338	6.64	0.71	91	4.3	60 / 12
Mostly single-prescription cases	708	281	6.65	1.18	92	4.1	44 / 13
Three times the case volume	792	344	6.64	0.90	91	3.4	59 / 16
80% new patients	795	353	6.64	0.93	91	3.4	60 / 17

G. Serialization

Table XIII and Fig. 6 report the benchmark. For a two-prescription case, the schema-based encodings are about half the size of JSON: 1,161 bytes for Protocol Buffers and 1,039 bytes for Avro against 2,326 bytes, while MessagePack and CBOR save about a sixth. Compression narrows the gap, because it removes much of the redundancy that field names carry, but leaves the schema-based encodings about 29% smaller. Encoding and decoding take between 1.3 and 34 microseconds per record. Protocol Buffers and orjson are the fastest, Avro in this Python implementation the slowest. Converting Protocol Buffers messages from and to dictionaries costs 204 and 76 µs, far more than the encoding itself, so the benefit appears only if services use the generated types directly.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

TABLE XIII. SERIALIZATION OF A TWO-PRESCRIPTION CLINICAL CONTEXT (SIZES: MEAN OF 200 RECORDS; TIMES: MICROSECONDS PER RECORD)

Encoding	Size (bytes)	Relative to JSON (%)	zstd size (bytes)	Encode (µs)	Decode (µs)
JSON (stdlib)	2,326	100	1,106	15.3	14.7
JSON (orjson)	2,326	100	1,106	2.1	6.3
MessagePack	1,923	83	1,134	6.1	10.5
CBOR	1,928	83	1,097	26.5	14.6
Avro (raw binary)	1,039	45	707	34.5	30.6
Protocol Buffers	1,161	50	789	1.3	1.8
Protocol Buffers, from and to dictionaries	1,167	50	–	204	76

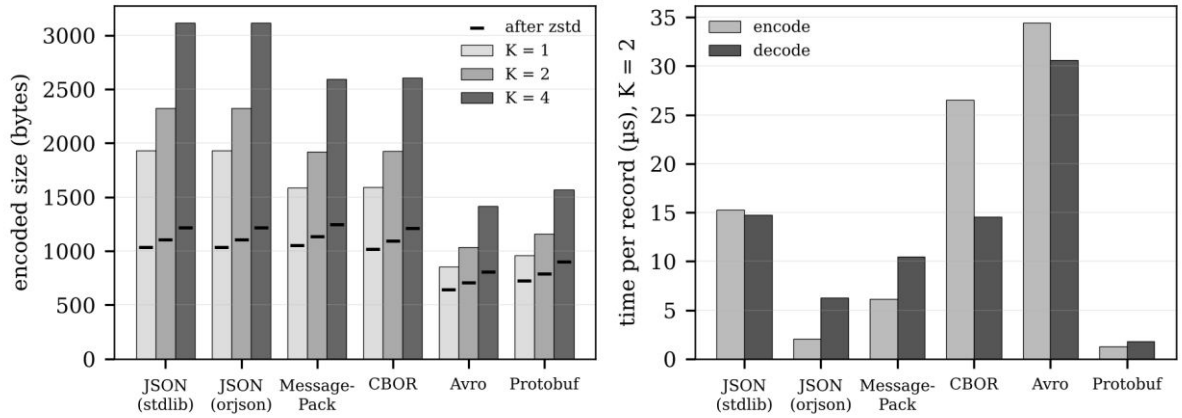


Fig. 6. Left: encoded size by encoding for cases with one, two and four prescriptions; ticks show the size after zstd compression. Right: encoding and decoding time per two-prescription record.

These costs are three orders of magnitude below backend latencies, so encoding does not affect the latencies of Table IX. It affects resources. A million cached two-prescription contexts occupy about 2.3 GB as JSON, 1.2 GB as Protocol Buffers and 0.8 GB compressed, before the overhead of the cache itself, and at a thousand requests per second the difference between the slowest and fastest encoding amounts to a few percent of one processor core.

H. Boundary validation and repair

Table XIV reports NPI validation at the highest registry density tested, 4%, where repair is hardest. Format and check digit together detected all single substitutions, OCR confusions, deletions and insertions; the check digit alone detected 98.0% of adjacent transpositions and 93.5% of twin errors, as the theory of the Luhn algorithm predicts (it misses only the transposition of 0 and 9 and the twin pairs 22/55, 33/66 and 44/77), and no jump transpositions, which leave the weighted sum unchanged. Of the jump transpositions, the format check caught 11.0% (those that moved a digit other than 1 or 2 into the first position), registry membership 85.5% and the name check the remaining 3.5%, so every injected error was detected by some layer. Repair with the name check corrected 98.0% of substitutions and similar shares of the other single-edit errors without a single wrong repair; the remaining 2.0% were routed because a typing error in the prescriber's name defeated the match. Twin errors lie outside the one-edit model and were always routed.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

TABLE XIV. NPI VALIDATION AND REPAIR AT 4% REGISTRY DENSITY (% OF 20,000 INJECTED ERRORS PER TYPE)

Error type	Format or check digit	Registry or name	Undetected	Repaired, name check	Wrong, name check	Ambiguous, no name check	Wrong, no name check
Substitution	100.0	0.0	0.0	98.0	0.0	27.5	0.0
Adjacent transposition	98.0	2.0	0.0	97.9	0.0	29.8	0.0
Jump transposition	11.0	89.0	0.0	98.0	0.0	18.1	0.0
Twin error	94.3	5.7	0.0	0.0	0.0	4.9	24.3
OCR confusion	100.0	0.0	0.0	98.0	0.0	26.8	0.0
Deletion	100.0	0.0	0.0	97.9	0.0	23.5	0.0
Insertion	100.0	0.0	0.0	98.0	0.0	2.4	0.0

The name check is what makes repair safe. Fig. 7 shows that without it, the share of single-edit errors with more than one registered candidate grows with registry density, from about 1% at 0.1% density to 28% at 4% for substitutions. More importantly, errors outside the edit model are then repaired wrongly: a twin error has a unique registered candidate at one edit that is not the true number in 0.8% to 24.3% of cases, depending on density. With the name check, no wrong repair occurred at any density.

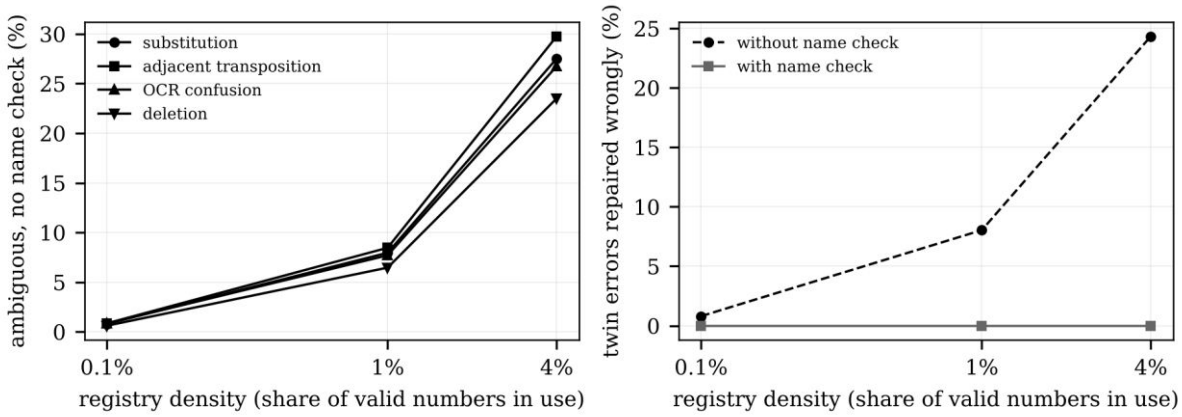


Fig. 7. NPI repair without the prescriber-name check: share of single-edit errors with ambiguous candidates (left) and share of twin errors repaired to a wrong number (right), against registry density; with the name check, no wrong repair occurred.

For NDCs (Table XV), which have no check digit, catalog lookup detected 99.5% of substitutions in the 11-digit form; the remaining 0.5% produced another package of the same product, which a quantity check rather than the drug check must catch. A shifted hyphen, which turns a 5-3-2 code into a 4-4-2 reading and produces a different 11-digit code, was always detected and always repaired by re-reading the digits in the other configurations, at zero edit cost. Unhyphenated 10-digit codes resolved to a single catalog entry in all but 0.01% of cases in this catalog; the ambiguity of such codes depends on how densely real labelers use their code space. Repairs matched the drug name and strength in every applied case, and between 2.3% and 5.1% of errors were routed because several package sizes of the same product matched.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

TABLE XV. NDC VALIDATION AND REPAIR (% OF 20,000 INJECTED ERRORS PER TYPE)

Form and error	Detected	Harmless after normalization	Same product, other package	Repaired	Routed, ambiguous	Wrong repair
11-digit, substitution	99.5	0.0	0.50	94.7	4.8	0.0
11-digit, transposition	99.9	0.0	0.10	97.6	2.3	0.0
11-digit, OCR confusion	99.5	0.0	0.52	95.0	4.5	0.0
Hyphenated, substitution	99.5	0.0	0.50	94.4	5.1	0.0
Hyphenated, OCR confusion	99.5	0.0	0.51	94.5	5.0	0.0
Hyphenated, hyphen shifted	100.0	0.0	0.00	100.0	0.0	0.0
10 digits, hyphens missing	0.0	100.0	0.00	0.0	0.0	0.0

Diagnosis codes are harder (Table XVI). The code set is dense within categories, so 4.5% of random substitutions and 6.7% of adjacent transpositions produced another billable code of the right therapy class, which no layer can detect without independent evidence such as the diagnosis text. Character-type rules (a letter where a digit must be) detected 55% of OCR confusions and allowed all of them to be repaired correctly; code-set membership caught most of the rest. Truncated codes were always caught by the billability check. Type-guided repair of random substitutions, however, produced wrong codes in 0.89% of cases, which is why diagnosis repairs are proposed to the technician rather than applied.

TABLE XVI. ICD-10-CM VALIDATION AND REPAIR (% OF 20,000 INJECTED ERRORS PER TYPE)

Error type	Format	Not in code set	Not billable	Wrong therapy class	Undetected	Harmless	Repaired	Wrong repair
OCR confusion	55.1	41.6	0.6	0.7	2.0	0.0	55.1	0.00
Substitution	38.4	47.7	6.6	2.8	4.5	0.0	2.0	0.89
Adjacent transposition	32.8	44.3	6.2	9.9	6.7	0.0	0.0	0.10
Truncation	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.00
Decimal misplaced	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.00

I. Effect on manual work

The detection and repair rates can be combined with assumptions about error incidence and handling costs to estimate the manual work that identifier errors cause. Assume per-field error rates of 0.1% for e-prescriptions, 0.5% for portal submissions and 1.0% for faxed referrals, the channel mix of 60%, 10% and 30%, three identifier fields per prescription, and error-type mixes dominated by substitutions and, for faxes, OCR confusions. Assume further that an error discovered at adjudication costs 15 technician minutes of rework, a targeted correction 2 minutes and the confirmation of a proposed diagnosis repair 1 minute. Under these assumptions there are 12.3 identifier errors per 1,000 prescriptions, and Table XVII gives the expected technician minutes. Format and check-digit validation alone saves 40% of the rework, the full set of layers 86%, and repair 95%. The absolute values depend on the assumptions; the ordering does not.

TABLE XVII. ESTIMATED TECHNICIAN MINUTES PER 1,000 PRESCRIPTIONS CAUSED BY IDENTIFIER ERRORS

Boundary treatment	Minutes / 1,000 Rx	Relative to none (%)
No boundary validation (errors found at adjudication)	184	100
Format and check digit only	111	60
All five layers, routed to targeted correction	27	14
All five layers with minimal-edit repair	10	5

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

J. Statistical significance

Table XVIII reports paired tests over the 20 replications at the reference load. All differences are significant after Holm correction (largest adjusted p = 1.7×10^{-5}). As in any simulation with many replications, the tests mainly show that differences are systematic; their size is better judged from the tables above and the sensitivity analysis.

TABLE XVIII. PAIRED WILCOXON SIGNED-RANK TESTS AT THE REFERENCE LOAD (TWO-SIDED, HOLM-ADJUSTED)

Comparison	Metric	Mean difference	p	Adjusted p
Proposed vs Fan-out	Case p95 latency (ms)	-445.05	1.9×10^{-6}	1.7×10^{-5}
Proposed vs Fan-out	Backend calls per Rx	-5.71	1.9×10^{-6}	1.7×10^{-5}
Proposed vs Unified API	Case p95 latency (ms)	-341.50	1.9×10^{-6}	1.7×10^{-5}
Proposed vs Unified API	Backend calls per Rx	-2.65	1.9×10^{-6}	1.7×10^{-5}
Proposed vs Unified + TTL 60	Case p95 latency (ms)	-290.21	1.9×10^{-6}	1.7×10^{-5}
Proposed vs Unified + TTL 60	Backend calls per Rx	-1.32	1.9×10^{-6}	1.7×10^{-5}
Unified API vs Fan-out	Case p95 latency (ms)	-103.55	1.9×10^{-6}	1.7×10^{-5}
Unified API vs Fan-out	Rx p50 latency (ms)	+66.40	1.9×10^{-6}	1.7×10^{-5}
Proposed vs Unified + TTL 60	Critical stale reads per 10,000 Rx	-153.04	1.9×10^{-6}	1.7×10^{-5}

VII. DISCUSSION

A. Operational implications

Three results matter most for practice. First, the unit of extraction should be the case. Deduplicating and batching the lookups of a regimen roughly halves backend calls before any cache is added, and it shortens case latency for multi-drug regimens even without a cache. Second, a cache in front of clinical systems is safe only with invalidation. A TTL long enough to give useful hit ratios also serves stale coverage and eligibility at the moment cases are re-evaluated, while a cache driven by versioned CDC events achieves a 91% hit ratio with stale decision-critical reads in the order of a few per 10,000 evaluations, most of them caused by lost invalidations that monitoring can detect. Third, deadlines and hedged requests decide the tail. Without hedging, a rare slow eligibility response delays the whole regimen, and near saturation, timeouts and retries add load until throughput collapses abruptly (Fig. 4).

The latencies themselves are seconds or less, while specialty turnaround is measured in days [1]. The contribution of fast extraction to the time to fill is therefore indirect: a context that is complete, fresh and validated when a case is released lets the case be cleared or routed correctly at once, instead of waiting for a technician to look up missing data, and makes it practical to re-evaluate a case every time something changes rather than in batches. Boundary validation attacks the same delay from the other side, by turning rework discovered at adjudication into targeted corrections or automatic repairs at the moment of ingestion.

B. Patient safety

The design introduces two hazards and controls both. A stale read of coverage or eligibility could clear a prescription that should not be cleared or hold one that could proceed. Versioned invalidation keeps such reads rare, the safety TTL bounds their age, and for the highest-risk decisions an organization can add a conditional read of the record version before final clearance, at the cost of one call per decision. A wrong repair could attach a prescription to the wrong prescriber, product or diagnosis. The repair rule applies a repair only when exactly one candidate survives every check, including the cross-field check against the document, and the experiments found no wrong repair under that rule for NPIs and NDCs. For diagnosis codes they did, so those repairs are suggestions. Every repair is logged with the original value and can be audited.

C. Privacy and security

A cache of clinical context is a store of protected health information and must be treated as one. Entries should be encrypted at rest and in transit, keyed by internal identifiers rather than by names or member numbers, and limited to the fields that the rules need, in line with the minimum-necessary standard [42]. The safety TTL doubles as a retention limit. The CDC stream carries changes to the same data and needs the same protection. Access to the cache, the stream and the repair log should be logged to support the audit controls of the HIPAA Security Rule [43].

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

D. Deployment and monitoring

A deployment can begin with the unified API alone, which needs no cache, and add the cache class by class, starting with reference data. Before decision-critical data is cached, the CDC stream should be shown to be complete, for instance by reconciling sequence numbers, because lost invalidations dominate residual staleness. Validators can run first in a reporting mode that flags errors without repairing them, so that repair rates and the absence of wrong repairs can be confirmed on local data. Table XIX lists indicators for operation.

TABLE XIX. OPERATIONAL INDICATORS FOR A DEPLOYMENT

Indicator	What it reveals	Typical response
Hit ratio by freshness class	whether lifetimes and prefetch suit the workload	adjust TTL or prefetch scope
Sampled stale reads (cache against source)	residual staleness of decision-critical data	investigate CDC gaps; shorten safety TTL
Invalidation lag and sequence gaps	speed and completeness of the CDC stream	repair the stream; replay from the log
Utilization of the busiest backend	headroom before saturation	add capacity; widen batching
Hedge and deadline-miss rates	tail behavior of each backend	tune hedge delay; escalate slow backends
Incomplete contexts by entity	where manual lookup is still needed	fix the source or the dependency
Repairs, routings and audited wrong repairs	data quality by channel and identifier	correct at the source; restrict repair

E. Generalization

The same structure appears wherever automated decisions need data from several systems of record that change at different rates: claims adjudication, prior authorization submissions, laboratory order validation and benefit verification for medical services. The design carries over with different entity types and dependencies, and the measurements of this paper, how much batching, invalidation and hedging contribute and how safe minimal-edit repair is, indicate what to expect.

F. Limitations and threats to validity

Construct validity. The metrics are proxies. Latency to a complete context does not include the time the rules take or the time a human needs for a routed case, and a stale read is counted as a risk whether or not it would have changed the decision. The estimate of manual work rests on assumed error rates and handling times.

Internal validity. Both stages were verified against closed-form results, policies were compared on identical request records, and each reference configuration was replicated twenty times. The two-stage decomposition assumes that cache outcomes do not depend on the request rate. This holds for patient-scoped data, whose request pattern is set by case events, but understates the hit ratio of shared reference data at high volumes, which makes the comparison conservative for the cached policies.

External validity. All workload, change and backend parameters are assumptions chosen to be plausible, not measurements from any pharmacy, so absolute values should not be read as predictions. The comparative conclusions held in every scenario of the sensitivity analysis, but other workloads, such as refill-dominated traffic with little re-evaluation, would benefit less from caching. The validation experiments use synthetic registries, a synthetic product catalog and synthetic error processes; real error distributions, especially for optical character recognition, and real registry and catalog densities should be measured before repair is enabled.

Conclusion validity. With twenty paired replications, small differences reach significance easily; the confidence intervals, the load sweep and the sensitivity analysis are the better guides to their size. A prospective evaluation on operational data, with appropriate privacy safeguards, is the natural next step.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

VIII. CONCLUSION

Clearing a specialty prescription without human work requires its clinical context to be complete, current and correct when the decision is made. This paper formulated context extraction as the evaluation of a dependency graph over entity services under a time budget and presented an architecture that validates and repairs identifiers at the boundary, resolves each case's graph through a unified entity API with deduplication, batching, deadlines and hedged requests, and serves entity data from a cache whose freshness is governed by class, by versioned change-data-capture invalidation and by prefetch. In a two-stage simulation, the design needed 86% fewer backend calls per prescription than per-prescription fan-out, cut the 95th-percentile case latency from 795 to 350 ms, sustained 3.8 times the request rate within a one-second budget and kept stale decision-critical reads to 3.9 per 10,000 evaluations. Validators with minimal-edit repair detected every injected NPI error and repaired most of them without a wrong repair when the prescriber's name was checked. Future work includes learning freshness classes from change logs, extending repair with document text, and validating the results prospectively.

Conflict of interest. None declared. The work relies exclusively on synthetic data and describes no specific organization's systems.

Data availability. No real patient or prescriber data were used. The simulation and benchmark code can be obtained from the author on request.

REFERENCES

[1] Pharmacy Quality Alliance, "Specialty Pharmacy Turnaround Time (SP-TAT): Measure rationale," PQA, 2021. [Online]. Available: https://www.pqaalliance.org/assets/docs/PQA_SP-TAT_Rationale.pdf
[2] M. H. Gabriel, C. M. Kotschevar, D. Tarver, V. Mastrangelo, L. Pezzullo, and P. J. Campbell, "Specialty pharmacy turnaround time impediments, facilitators, and good practices," *J. Manag. Care Spec. Pharm.*, vol. 28, no. 11, 2022, doi: 10.18553/jmcp.2022.28.11.1244.
[3] American Medical Association, "2022 AMA prior authorization (PA) physician survey," AMA, Chicago, IL, USA, 2023.
[4] American Society of Health-System Pharmacists, pharmacy administrator and pharmacy technician workforce surveys (Nov. 2021), reported in "Survey reveals severe shortage of pharmacy technicians in hospitals and health systems," *News-Medical*, Mar. 15, 2022. [Online]. Available: https://www.news-medical.net/news/20220315/Survey-reveals-severe-shortage-of-pharmacy-technicians-in-hospitals-and-health-systems.aspx
[5] Centers for Medicare & Medicaid Services, "Modernizing Part D and Medicare Advantage to lower drug prices and reduce out-of-pocket expenses; final rule," *Federal Register*, vol. 84, p. 23832, May 23, 2019.
[6] Centers for Medicare & Medicaid Services, "Advancing interoperability and improving prior authorization processes (CMS-0057-F)," *Federal Register*, vol. 89, no. 27, p. 8758, Feb. 8, 2024.
[7] HL7 International, "FHIR Release 4 (v4.0.1)," 2019. [Online]. Available: https://hl7.org/fhir/R4/
[8] HL7 International, "Da Vinci Prior Authorization Support (PAS) FHIR Implementation Guide." [Online]. Available: https://hl7.org/fhir/us/davinci-pas/
[9] National Council for Prescription Drug Programs, *SCRIPT Standard Implementation Guide, Version 2017071*. Scottsdale, AZ, USA: NCPDP, 2017.
[10] J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, 2013.
[11] O. K. Odukoya, J. A. Stone, and M. A. Chui, "How do community pharmacies recover from e-prescription errors?," *Res. Social Adm. Pharm.*, vol. 10, no. 6, pp. 837–852, 2014, doi: 10.1016/j.sapharm.2013.11.009.
[12] M. Lenzerini, "Data integration: A theoretical perspective," in *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS)*, 2002, pp. 233–246.
[13] A. Halevy, A. Rajaraman, and J. Ordille, "Data integration: The teenage years," in *Proc. 32nd Int. Conf. Very Large Data Bases (VLDB)*, 2006, pp. 9–16.
[14] C. Richardson, *Microservices Patterns*. Shelter Island, NY, USA: Manning, 2018.
[15] M. Kleppmann, *Designing Data-Intensive Applications*. Sebastopol, CA, USA: O'Reilly Media, 2017.
[16] GraphQL Foundation, "GraphQL Specification," October 2021 ed. [Online]. Available: https://spec.graphql.org/October2021/
[17] O. Hartig and J. Pérez, "Semantics and complexity of GraphQL," in *Proc. World Wide Web Conf. (WWW)*, Lyon, France, 2018, pp. 1155–1164, doi: 10.1145/3178876.3186014.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309132|

[18] R. Fielding, M. Nottingham, and J. Reschke, "HTTP Caching," RFC 9111 (STD 98), Internet Engineering Task Force, Jun. 2022.
[19] M. Nottingham, "HTTP Cache-Control Extensions for Stale Content," RFC 5861, Internet Engineering Task Force, May 2010.
[20] J. Jung, A. W. Berger, and H. Balakrishnan, "Modeling TTL-based Internet caches," in *Proc. IEEE INFOCOM*, vol. 1, 2003, doi: 10.1109/INFCOM.2003.1208693.
[21] R. Nishtala, H. Fugal, S. Grimm, M. Kwiatkowski, H. Lee, H. C. Li, R. McElroy, M. Paleczny, D. Peek, P. Saab, D. Stafford, T. Tung, and V. Venkataramani, "Scaling Memcache at Facebook," in *Proc. 10th USENIX Symp. Networked Systems Design and Implementation (NSDI)*, 2013, pp. 385–398.
[22] N. Bronson et al., "TAO: Facebook's distributed data store for the social graph," in *Proc. USENIX Annual Technical Conf. (ATC)*, 2013, pp. 49–60.
[23] S. Das et al., "All aboard the Databus! LinkedIn's scalable consistent change data capture platform," in *Proc. 3rd ACM Symp. Cloud Computing (SoCC)*, 2012, doi: 10.1145/2391229.2391247.
[24] T. Bray, Ed., "The JavaScript Object Notation (JSON) Data Interchange Format," RFC 8259, Internet Engineering Task Force, Dec. 2017.
[25] MessagePack, "MessagePack specification." [Online]. Available: https://github.com/msgpack/msgpack/blob/master/spec.md
[26] C. Bormann and P. Hoffman, "Concise Binary Object Representation (CBOR)," RFC 8949, Internet Engineering Task Force, Dec. 2020.
[27] Google, "Protocol Buffers documentation." [Online]. Available: https://protobuf.dev/
[28] Apache Software Foundation, "Apache Avro specification." [Online]. Available: https://avro.apache.org/docs/
[29] J. C. Viotti and M. Kinneer, "A survey of JSON-compatible binary serialization specifications," arXiv:2201.02089, 2022.
[30] U.S. Department of Health and Human Services, "HIPAA administrative simplification: Standard unique health identifier for health care providers; final rule," *Federal Register*, vol. 69, no. 15, pp. 3434–3469, Jan. 23, 2004.
[31] H. P. Luhn, "Computer for verifying numbers," U.S. Patent 2,950,048, Aug. 23, 1960.
[32] International Organization for Standardization, *ISO/IEC 7812-1: Identification Cards — Identification of Issuers — Part 1: Numbering System*. Geneva, Switzerland: ISO, 2017.
[33] U.S. Food and Drug Administration, "Revising the National Drug Code format and drug label barcode requirements; proposed rule," *Federal Register*, vol. 87, no. 141, pp. 44038–44048, Jul. 25, 2022.
[34] Centers for Disease Control and Prevention, National Center for Health Statistics, "International Classification of Diseases, Tenth Revision, Clinical Modification (ICD-10-CM), 2021 code tables and order file." [Online]. Available: https://www.cdc.gov/nchs/icd/icd-10-cm/files.html
[35] T. T. H. Nguyen, A. Jatowt, M. Coustaty, and A. Doucet, "Survey of post-OCR processing approaches," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–37, 2021, doi: 10.1145/3453476.
[36] P. J. Courtois, *Decomposability: Queueing and Computer System Applications*. New York, NY, USA: Academic Press, 1977.
[37] A. M. Law, *Simulation Modeling and Analysis*, 5th ed. New York, NY, USA: McGraw-Hill, 2015.
[38] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
[39] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Statist.*, vol. 6, no. 2, pp. 65–70, 1979.
[40] N. Gans, G. Koole, and A. Mandelbaum, "Telephone call centers: Tutorial, review, and research prospects," *Manufacturing & Service Operations Management*, vol. 5, no. 2, pp. 79–141, 2003.
[41] S. Travasci, "simple_icd_10_cm: A simple Python library for ICD-10-CM codes," version 1.2.0 (2021 code tables). [Online]. Available: https://github.com/StefanoTrv/simple_icd_10_CM
[42] U.S. Department of Health and Human Services, "Uses and disclosures of protected health information: General rules (minimum necessary)," 45 C.F.R. § 164.502(b).
[43] U.S. Department of Health and Human Services, "Security standards for the protection of electronic protected health information," 45 C.F.R. Part 164, Subpart C.

IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 9, September 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.524

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|

सवाई माधोपुर जिले में भूमि–उपयोग प्रारूप का भौगोलिक अध्ययन

विजयराम गुर्जर[1], डॉ. हेमंत मंगल[2]

शोधार्थी, भूगोल विभाग, राजकीय लोहिया महाविद्यालय, चूरु (राजस्थान), भारत[1]
आचार्य, भूगोल विभाग, राजकीय लोहिया महाविद्यालय, चूरु (राजस्थान), भारत[2]

सारांश

कृषि परम्परागत राष्ट्रों में विशेष कर भारत में कृषि एक अजीविका है, एक व्यवसाय है, एक क्रिया–उद्यम है, एक व्यवसाय है कृषि एक परम्परा है, जीवन पद्धति है, एक समाज, सभ्यता एवं संस्कृति है। इस प्रकार भारत में कृषि, कृषि विकास व भूमि उपयोग एक व्यवसाय ही नहीं वरन् एक जीवन पद्धति है। यह एक सांस्कृतिक विरासत है, जो भारत को सदियों से जीवंत रखे हुए है। इस सन्दर्भ में यह मानव जाति के सामाजिक, सांस्कृतिक एवं आर्थिक विकास के इतिहास का वह तथ्य है जो मानव सभ्यताओं के विकास क्रम के साथ–साथ धरोहर के रूप में पीढ़ी दर पीढ़ी हमें प्राप्त हुई है। आधुनिकीकरण ने प्राकृतिक संसासधनों का अधाधुंध उपयोग जैव तकनीकी एवं वैज्ञानिक अन्वेषण, आविष्कारों ने कृषि विकास के प्रारूप उत्पादन क्षमता , वितरण व्यवसाय आदि को एक नवीन रूप दिये है, वहीं इनमें जैविकीय तत्वों को बदल कर फसलों की परिस्थितिकी को ही बदल ड़ाला है। एक ओर बढ़ती जनसंख्या के कारण खाद्यानों की बढ़ती मांग को तो कृषि विकास के स्तर को बढ़ाकर कर लिए लेकिन दूसरी और पर्यावरणीय नियोजन को ध्यान में रखना हम सभी के लिए चिंता का विषय है। ''सवाई माधोपुर जिले में भूमि–उपयोग प्रारूप का भौगोलिक अध्ययन'' विषयक यह शोध–पत्र एक बहुआयामी भौगोलिक विश्लेषण है, जिसका उद्देश्य जिले की भूमि के विभिन्न उपयोगों की प्रकृति, वितरण, कारण एवं प्रभावों को समझना है। यह अध्ययन भूमि–उपयोग की विभिन्न श्रेणियों जैसे कृषि भूमि, सिंचित एवं असिंचित क्षेत्र, वनीय क्षेत्र, चारागाह, अनुपजाऊ–बंजर भूमि, आवासीय एवं शहरी विस्तार आदि का तुलनात्मक रूप से विश्लेषण करता है। शोध में यह भी स्पष्ट किया गया है कि किस प्रकार प्राकृतिक कारक जैसे स्थलाकृति, जलवायु, मृदा की गुणवत्ता, जल स्रोतों की उपलब्धता तथा मानवजनित कारक जैसे जनसंख्या वृद्धि, शहरीकरण, औद्योगीकरण एवं सरकारी नीतियाँ भूमि–उपयोग के स्वरूप को प्रभावित करते हैं।

मूल शब्द – भूमि उपयोग परिवर्तन, कृषि विस्तार, आधुनिकीकरण, औद्योगीकरण

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|

प्रस्तावना :

भारत एक कृषि प्रधान देश है, जहाँ भूमि उपयोग की प्रवृत्तियाँ न केवल आर्थिक विकास को प्रभावित करती हैं, बल्कि पारिस्थितिक संतुलन, जनसंख्या की आजीविका और प्राकृतिक संसाधनों के संरक्षण को भी प्रभावित करती हैं। राजस्थान राज्य के सवाई माधोपुर जिले की भौगोलिक संरचना, जलवायु परिस्थितियाँ तथा संसाधनों की उपलब्धता कृषि भूमि उपयोग की दिशा में विशेष भूमिका निभाती हैं। विगत तीन दशकों में जिले में भूमि उपयोग की प्रवृत्तियों में उल्लेखनीय परिवर्तन देखा गया है, जो कृषि तकनीकों में सुधार, सिंचाई सुविधाओं के विस्तार और जनसंख्या वृद्धि जैसे कारकों से प्रेरित है। कृषि भूगोल में कृषि विकास एवं नियोजन की संकल्पना विषय वस्तु के व्यवहारिक पक्ष को प्रस्तुत करती है। प्रायः कृषि विकास से तात्पर्य कृषि उत्पादकता वृद्धि से लिया जाता रहा है, कृषि उत्पादकता में यह वृद्धि वैज्ञानिक एवं तकनीकी विधियों खाद, बाजों, सिंचाई के आधुनिक साधनों व आधुनिक यन्त्रों के समावे'ा के फलस्वरूप सम्भव हुआ है। सवाई माधोपुर राजस्थान का एक महत्वपूर्ण जिला है, जहाँ भौगोलिक विविधता, वन क्षेत्र, रणथंभौर राष्ट्रीय उद्यान, कृषि, और बस्तियों की विशेष भूमिकाएँ भूमि–उपयोग पैटर्न को प्रभावित करती हैं। यहाँ पर कृषि वृद्धि और कृषि विकास में विभेद का ज्ञान आव'यक है। यान्त्रिक क्रान्ति व हरित क्रान्ति के पूर्व कृषि को उत्पादकता में वृद्धि का स्थानापन्न माना जाता रहा है। परन्तु आज उत्पादकता में होने वाली वृद्धि के अपेक्षाकृत कृषि विकास को अधिक विस्तृत अर्थों में प्रयोग करते है।1 विकास वृद्धि का पर्याय नहीं अपितु इसमें उत्पादकता वृद्धि के साथ ही उत्पादों का समान सामयिक वितरण तथा पर्यावरणीय सन्तुलन बनाये रखने पर भी विचार किया जाता है। इस प्रकार कृषि विकास का अभिप्राय उस उत्पादकता की वृद्धि से है जिसका लाभ समाज के सभी वर्गों को समान रूप से प्राप्त हो और पर्यावरण का स्वरूप भी विकृत न हो। कृषि भू–दृश्य में विकास तभी सम्भव हो सकता है जब कृषि के स्वरूप को निर्धारण करने वाले सभी कारकों को योजनाबद्ध ढ़ंग से प्रयोग किया जाये । अब तक केवल उत्पादकता की ही तरह कृषि विकास में सामजिक कल्याण और पर्यावरणीय नियोजन के सन्तुलन सम्बधी तथ्यों को भी प्राथमिकता प्रदान की जा रही है , ताकि किसी भी प्रकार का असंतुलन उत्पन्न न हो । इस अध्ययन का उद्देश्य इन परिवर्तनशील प्रवृत्तियों का विश्लेषण कर यह समझना है कि सवाई माधोपुर जिले में भूमि उपयोग की दिशा किस प्रकार विकसित हुई है, और भविष्य में इसे संतुलित व सतत विकास की ओर कैसे अग्रसर किया जा सकता है।

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|

शोध उद्देश्य :

प्रस्तुत शोध कार्य के उद्देश्य निम्नलिखित है–

- जिले में वर्तमान भूमि–उपयोग की विभिन्न श्रेणियों का वर्गीकरण और उनका भौगोलिक वितरण विश्लेषित करना।
- भूमि–उपयोग में समय के साथ हुए परिवर्तनों की पहचान करना तथा इसके प्रमुख कारणों का अध्ययन करना।
- प्राकृतिक एवं मानवजनित कारकों द्वारा भूमि–उपयोग पैटर्न पर पड़ने वाले प्रभावों का विश्लेषण करना।
- सतत भूमि प्रबंधन एवं पर्यावरणीय संतुलन की दृष्टि से जिले में उपयुक्त भूमि–उपयोग रणनीतियाँ सुझाना।

अनुसंधान पद्धति :

प्रस्तुत शोध पत्र मूलतः द्वितीयक आंकड़ों की सहायता से पूर्ण किया गया है इस हेतु आंकड़ों का संकलन जिला सांख्यिकी कार्यालय, सवाई माधोपुर, भू–अभिलेख कार्यालय, सवाई माधोपुर।, ब्लॉक सांख्यिकी कार्यालय, आदि से किया गया है। साथ ही इस संबंध में नवीन आंकड़ों के संकलन हेतु विभिन्न ऑनलाइन स्रोतों से भी सहायता ली गयी है।

अध्ययन क्षेत्र :

अध्ययन क्षेत्र राजस्थान के दक्षिण–पूर्वी भाग में स्थित सवाई माधोपुर जिला है। सवाई माधोपुर शहर की स्थापना 1763 ई. में सवाई माधोसिंह प्रथम ने की थी जो जयपुर के महाराजा थे। यह $24^{o}45'$ से $26^{o}41'$ उत्तरी अक्षांश तथा $75^{o}79'$ से $77^{o}00'$ पूर्वी देशांतरों के मध्य स्थित है। जिले की समुन्द्रतल से ऊँचाई 450 मी. से 600 मी. तक है। जिले का कुल भौगोलिक क्षेत्रफल 4498.00 वर्ग किमी है जिसमें 4378. 40 वर्ग किमी. ग्रामीण तथा 119.7 वर्ग किमी. शहरी क्षेत्र है। जिले में उपखंड 8, तहसीलें 8, पटवार सर्किल 267 व राजस्व गांव 803 है। अध्ययन क्षेत्र को उच्चावच की दृष्टि से तीन मुख्य इकाईयों में बांटा जा सकता है। जिले का पश्चिमी एवं उत्तरी पश्चिमी भाग जलोढ़ मैदान, मध्यवर्ती भाग पृथक्–पृथक् पहाड़ियों के साथ जलोढ़ी मैदानी भाग तथा दक्षिण व दक्षिण पूर्वी भाग पर्वतीय भू–भाग के अंतर्गत आता है। जिले में बारहमासी नदी चम्बल है तथा बनास, मोरेल, गंभीरी, ढील अन्य प्रमुख नदियाँ हैं। जिले में उपार्द्र जलवायु पायी जाती है। यहाँ मृदा को मुख्यतः तीन प्रकारों में विभाजित किया जा सकता है जिसमें पुरानी जलोढ़ मृदा, पर्वतीय लियोसोल व रिगोसोल मृदा तथा नवीन जलोढ़ मृदा सम्मिलित है।

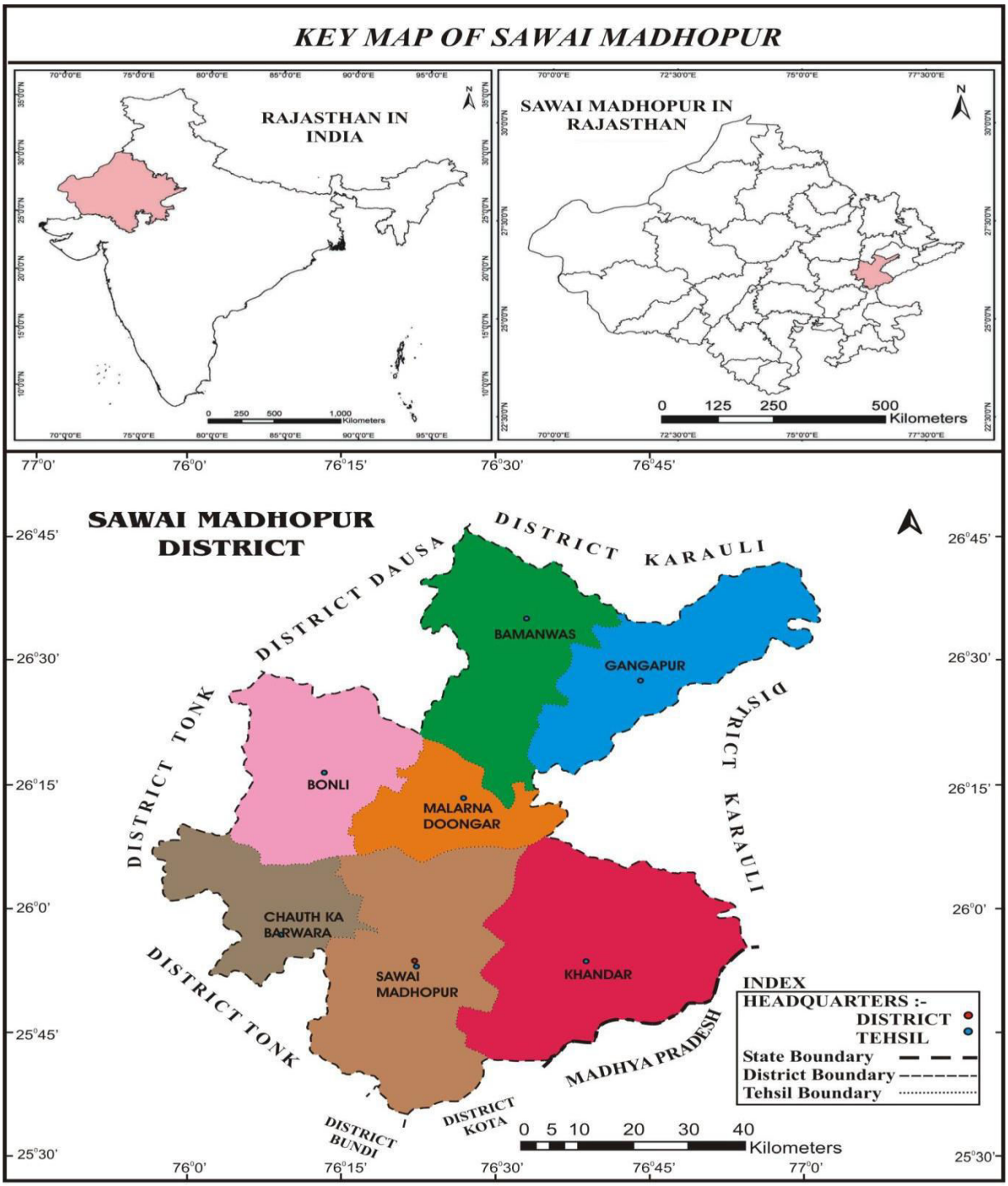
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|



International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|

कृषि भूमि उपयोग :

किसी भी क्षेत्र या प्रदेश के कृषि प्रतिरूप वहाँ के सिंचाई प्रतिरूप, भौतिक, सामाजिक एवं मानवीय कारकों द्वारा निर्धारित होता है। इसमें सिंचाई व जनसंख्या की आवश्यकता का योगदान अति महत्त्वपूर्ण होता है क्योंकि जनसंख्या की आवश्यकता की पूर्ति के लिए कृषि में विभिन्न परिवर्तन किये जाते है। जिले में जीविकोपार्जन का प्रमुख स्त्रोत भूमि ही है। अतः भूमि उपयोग का वि ंलेषण करने के पूर्व भूमि संसाधन आधार का मूल्यांकन करना परम आवश्यक जाने पड़ता है। क्योंकि इस क्षेत्र का अन्यत्र जीवन के उच्च स्तर का आधार निःसन्देह सम्पन्न भूमि संसाधन है। भारत एक कृषि प्रधान देश है, जहाँ कुल जनसंख्या का 70 प्रतिशत लोग कृषि कार्यों में संलग्न है। चीन को छोड़कर संसार के किसी भी देश में इतनी बड़ी जनसंख्या पूर्णतया कृषि पर आश्रित नहीं है। इस दृष्टि से भारत संसार के प्रमुख खेतीहर देशों में से एक है। वन तथा मत्स्य मिलाकर अकेल कृषि से लगभग 45 प्रतिशत राष्ट्रीय आय प्राप्त होती है। अतः भारत की अर्थव्यवस्था का आधार कृषि है।

आधुनिक युग की समस्या अभिमुख अनुसंधानीय एवं वैज्ञानिक अध्ययनों में भूमि एवं उसके सानुकुल उपयोग का अत्यन्त महत्त्वपूर्ण स्थान है। जिस देश की सम्पूर्ण अर्थव्यवस्था ही कृषि संसाधनों पर आधारित है, उस क्षेत्र के भौतिक अध्ययन में भूमि उपयोग से अपेक्षा अधिक महत्त्वपूर्ण सम्भवतः अन्य कोई विषय नहीं है। अतः यह आवश्यक हो गया है कि संसार के विभिन्न क्षेत्रों की भूमि का विभिन्न कार्यों के लिए सबसे लाभप्रद तथा अधिकतम उपयोग को ध्यान में रखते हुए भूमि का उपयोग आदर्श रूप में किया जाए। भूमि के सदुपयोग व दुरूपयोग के मूल्यांकन एवं उसके आधार पर आदर्श उपयोग के निर्माण हेतु यह आवश्यक है कि संसार के विभिन्न क्षेत्रों में जिस ढ़ग से विभिन्न उदेश्यों की पूर्ति के लिए भूमि का उपयोग किया जा रहा है, उसकी व्यवस्था एवं समीक्षा की जाए तभी वर्तमान एवं भूत के आधार पर भविष्य की योजना का निर्माण किया जा सकता है। भारतीय कृषि के समरूप ही जिले में भी स्वतंत्रता प्राप्ति के पश्चात ही योजनाबद्ध विकास से कृषि विकास हुआ है। जो देश में आई हरित क्रांन्ति के फलस्वरूप राजस्थान राज्य के इस क्षेत्र में भी तिव्र कृषि विकास हुआ है।

कृषि भूमि उपयोग का वर्गीकरण :

भूमि प्रकृति द्वारा प्रद्धत अमूल्य निधि है। भूमि संसाधन आर्थिक क्रियाओं का आधार है। किसी भी क्षेत्र में भूमि संसाधन की उपयोगिता से क्षेत्र का आर्थिक भू–दृश्य प्रभावित होता है। कृषि की दृष्टि से भूमि समतल होना ,उपजाऊपन पानी की उपलब्धता एवं अन्य अनुकूल भौगोलिक परिस्थितियों के आधार पर कृषि प्रारूप प्रभावित होता है। जब भू–भाग का प्राकृतिक स्वरूप लुप्त हो जाता है और मानवीय क्रियाओं के योगदान से एक नया भू–दृश्य जन्म लेता है, उसे भूमि उपयोग कहते है। भूमि उपयोग से तात्पर्य किसी भू–भाग का मानव द्वारा किसी कार्य के लिए प्रस्तुत किए जाने से है। यह प्रयुक्तिकरण कृषि, पशुपालन व उधोग आदि से सम्बन्धित हो

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|

सकता है। तो दुसरी ओर मानव की सांस्कृतिक क्रियाओं के लिए भी इसे प्रस्तुत किया जा सकता है। भूमि एक सीमित संसाधन है तथा इसके कुछ भाग में ही कृषि कार्य होते है, जो कृषि प्रारूप का आधार है।

तालिका संख्या 1 :
सवाई माधोपुर जिले में कृषि भूमि उपयोग
सन 1992–93 से 2021–22

क्र. सं.	भूमि उपयोग	क्षेत्रफल प्रतिशत में	क्षेत्रफल प्रतिशत में	1992 से 2022 में आया परिवर्तन (प्रतिशत में)
1	वन	22.74	16.38	–6.44
2	कृषि अनुपलब्ध भूमि	15.80	12.99	–2.81
3	कृषि आयोग्य भूमि	8.74	8.05	–1.28
4	परती भूमि	12.94	4.36	–8.58
5	वास्तविक बोया गया क्षेत्र	39.78	58.69	+18.85
6	कुल बोया गया क्षेत्र	47.50	80.42	+32.92
7	दो फसली	7.71	21.13	+14.06

स्त्रोतः– भू–अभिलेख कार्यालय, सवाई माधोपुर।

जिले में कृषि भूमि उपयोग (1992–93) :

वर्ष 1991–92 में सवाई माधोपुर जिला राजस्थान का एक प्रमुख कृषि प्रधान क्षेत्र था, जहाँ जनसंख्या की आजीविका का मुख्य आधार कृषि कार्य था। इस समय जिले में भूमि उपयोग की पारंपरिक प्रवृत्तियाँ स्पष्ट रूप से परिलक्षित होती थीं। प्राकृतिक संसाधनों पर अत्यधिक निर्भरता, सीमित सिंचाई साधन, परंपरागत कृषि पद्धतियाँ एवं असमतल भू–आकृति के कारण कृषि भूमि का समुचित उपयोग नहीं हो पाता था। इस कालखंड में जिले की भूमि का एक बड़ा भाग वन क्षेत्र (22.74 प्रतिशत) और कृषि अनुपलब्ध भूमि (15.80 प्रतिशत) के अंतर्गत आता था, जिससे स्पष्ट होता है कि बड़ी मात्रा में भूमि का कृषि में प्रत्यक्ष उपयोग नहीं हो रहा था। इसके अतिरिक्त कृषि योग्य परंतु अनुपयोगी भूमि (8.74 प्रतिशत) तथा परती भूमि (12.94 प्रतिशत) की अधिकता यह दर्शाती है कि जल संसाधनों की कमी, पूंजी की न्यूनता और तकनीकी जानकारी के अभाव के कारण किसानों की भूमि पर कृषि कार्य सीमित था। वास्तविक बोया गया क्षेत्र केवल 39.78 प्रतिशत था, जो यह इंगित करता है कि कृषि भूमि का पूर्ण रूप से उपयोग नहीं हो पा रहा था। वहीं दो फसली भूमि मात्र 7.71 प्रतिशत होने से यह अनुमान लगाया जा सकता है कि अधिकांश क्षेत्र वर्ष में केवल एक बार ही उपयोग में लाया जाता था। इस

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

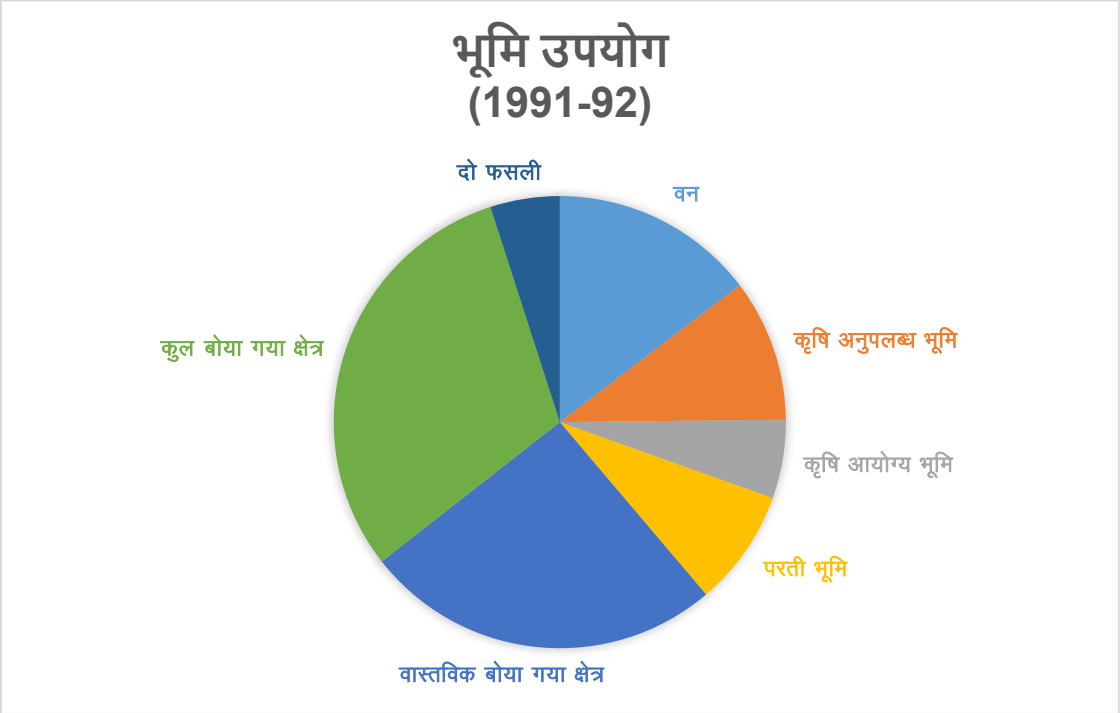
|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|

समय कृषि उत्पादन मानसून पर अत्यधिक निर्भर था, और सिंचाई सुविधाओं की सीमित पहुँच के कारण उत्पादन की निरंतरता और गुणवत्ता पर विपरीत प्रभाव पड़ता था। इस कालखंड का भूमि उपयोग परिदृश्य यह संकेत देता है कि जिले में कृषि क्षेत्र का विकास उस समय प्रारंभिक अवस्था में था, और विकास की अपार संभावनाएँ विद्यमान थीं। यह समय एक ऐसे संक्रमण काल का था, जिसमें परंपरागत कृषि व्यवस्था और आधुनिक कृषि तकनीकों के बीच संतुलन स्थापित करने की आवश्यकता थी।

आरेख संख्या 1 : जिले में कृषि भूमि उपयोग 1992–93



जिले में कृषि भूमि उपयोग (2021–22) :

वर्ष 2021–22 तक आते–आते सवाई माधोपुर जिले के कृषि परिदृश्य में उल्लेखनीय परिवर्तन परिलक्षित होता है। पिछले तीन दशकों में कृषि क्षेत्र में तकनीकी प्रगति, सिंचाई साधनों का विस्तार, सरकारी योजनाओं की प्रभावी क्रियान्विति और किसानों की जागरूकता में वृद्धि के कारण भूमि उपयोग की प्रवृत्तियों में सकारात्मक परिवर्तन हुआ है। अब भूमि का अधिक हिस्सा कृषि उत्पादन में सक्रिय रूप से सम्मिलित हो चुका है, जिससे जिले की कृषि उत्पादकता, आर्थिक संरचना और ग्रामीण जीवनशैली में व्यापक बदलाव आए हैं। इस अवधि में वन क्षेत्र घटकर 16.38 प्रतिशत रह गया है, जो शहरीकरण, खेती योग्य भूमि विस्तार और जनसंख्या दबाव का परिणाम है। वहीं कृषि अनुपलब्ध भूमि तथा कृषि योग्य लेकिन अनुपयोगी भूमि में क्रमशः गिरावट आई है, जिससे यह स्पष्ट होता है कि अब पहले अनुपयोगी मानी जाने वाली भूमि को भी कृषि योग्य बनाया गया है। परती भूमि में भारी कमी (4.36 प्रतिशत) यह दर्शाती है कि किसान अब अधिक भूमि पर खेती कर रहे हैं और कृषि संसाधनों

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

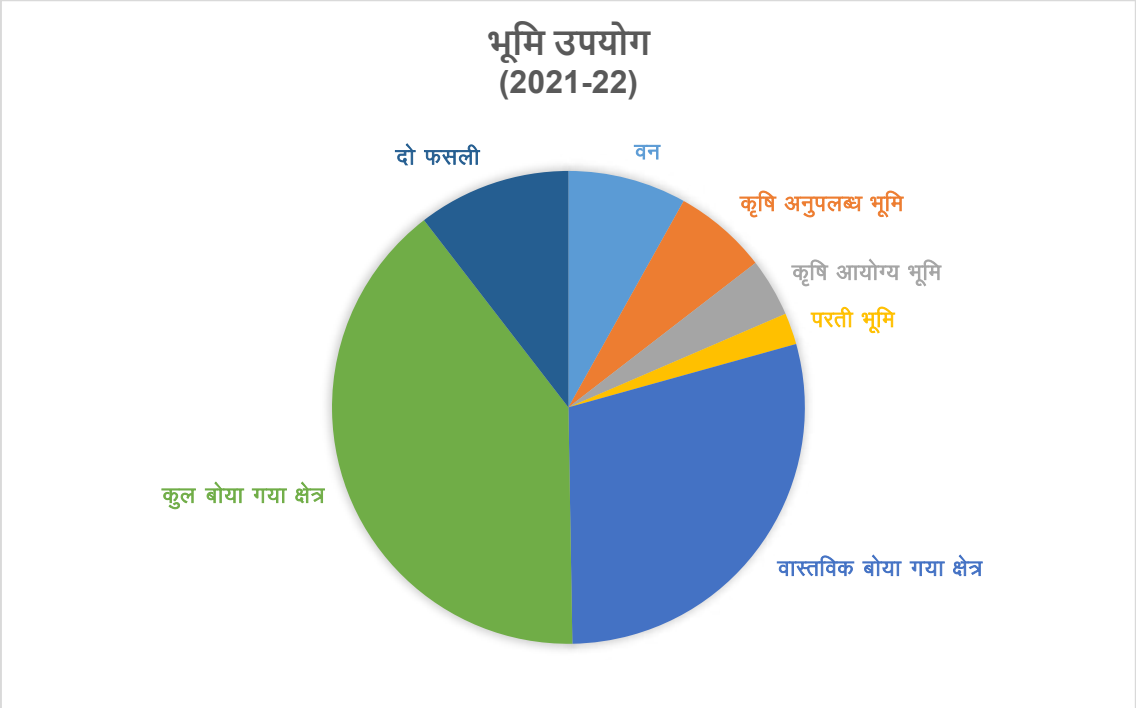
|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|

का उपयोग बढ़ा है। सबसे उल्लेखनीय परिवर्तन वास्तविक बोया गया क्षेत्र (58.69 प्रतिशत) तथा कुल बोया गया क्षेत्र (80.42 प्रतिशत) में देखा गया है। इससे यह स्पष्ट होता है कि जिले में कृषि गतिविधियाँ बड़े स्तर पर विस्तारित हुई हैं। दो फसली भूमि का बढ़कर 21.13 प्रतिशत होना यह संकेत देता है कि सिंचाई सुविधाओं और कृषि तकनीकों की बेहतर उपलब्धता ने किसानों को वर्ष में एक से अधिक बार खेती करने में सक्षम बनाया है। वर्ष 2021–22 का यह भूमि उपयोग परिदृश्य सवाई माधोपुर जिले की कृषि अर्थव्यवस्था में आए सकारात्मक बदलावों को प्रतिबिंबित करता है। यह समय आधुनिक कृषि प्रणाली, संसाधनों के बेहतर प्रबंधन और भूमि के अधिकतम उपयोग की दिशा में जिले की सफलता को दर्शाता है। यद्यपि इस प्रगति के साथ–साथ पारिस्थितिक संतुलन बनाए रखने की आवश्यकता भी अत्यंत महत्वपूर्ण हो जाती है।

आरेख संख्या 2 : जिले में कृषि भूमि उपयोग 2021–22



जिले में कृषि भूमि उपयोग में 1992–93 से 2021–22 में आया परिवर्तन :

वर्ष 1992–93 में जिले की भूमि का बड़ा भाग वन, कृषि अनुपलब्ध तथा परती भूमि के अंतर्गत आता था। उस समय वास्तविक बोया गया क्षेत्र 40 प्रतिशत से कम था तथा दो फसली क्षेत्र की स्थिति सीमित थी, जिससे कृषि संसाधनों की सीमित पहुँच का आभास होता था। परंतु वर्ष 2021–22 तक आते–आते यह परिदृश्य पूर्णतः परिवर्तित हो गया। वन क्षेत्र और परती भूमि में उल्लेखनीय कमी आई, जबकि वास्तविक बोया गया क्षेत्र और दो फसली भूमि में तीव्र वृद्धि हुई। कुल बोया गया क्षेत्र 47.50 प्रतिशत से बढ़कर 80.42 प्रतिशत तक पहुँच गया, जो इस बात का संकेत है कि जिले में कृषि भूमि का अधिकतम दोहन किया जाने लगा है। यह परिवर्तन भूमि

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

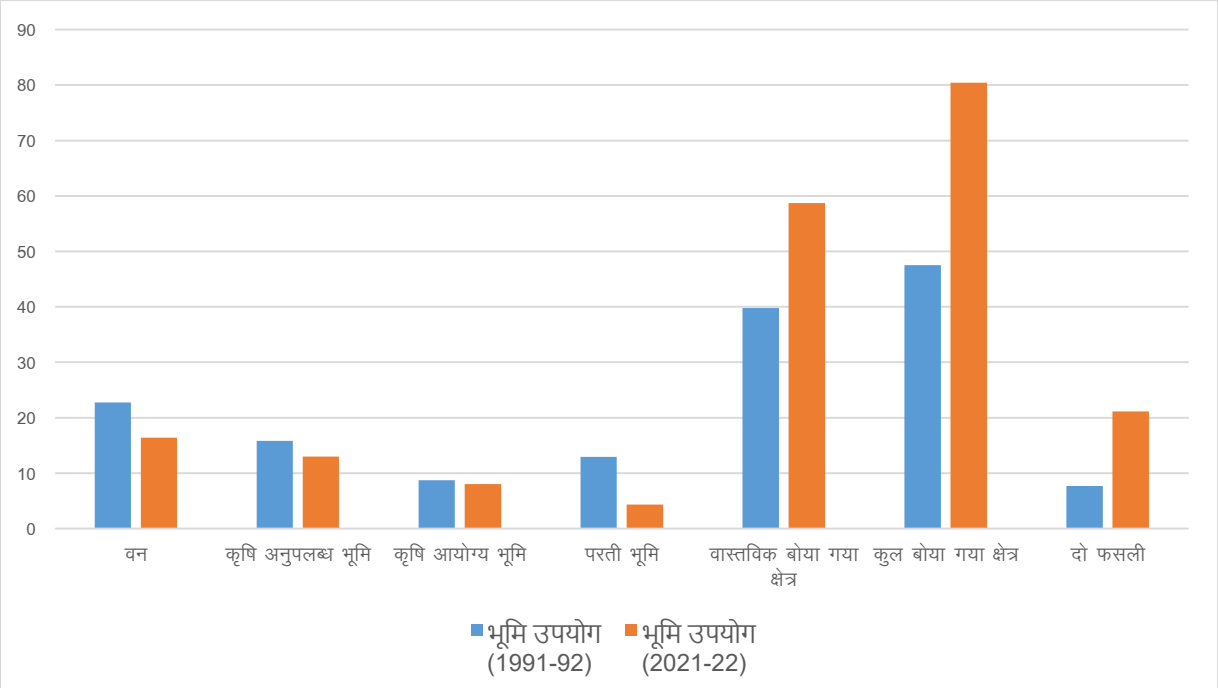
|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|

सुधार, सिंचाई योजनाओं, कृषि जागरूकता कार्यक्रमों और तकनीकी हस्तक्षेपों का प्रतिफल है। जहाँ एक ओर यह आर्थिक विकास और खाद्य सुरक्षा की दिशा में सकारात्मक संकेत है, वहीं दूसरी ओर वन क्षेत्र में कमी और पारिस्थितिक असंतुलन की संभावना भी चिंतन का विषय है।

आरेख संख्या 3 : जिले में भूमि उपयोग परिवर्तन – 1991 से 2022



निष्कर्ष :

सवाई माधोपुर जिले में वर्ष 1992–93 से 2021–22 के मध्य कृषि भूमि उपयोग में उल्लेखनीय परिवर्तन हुआ है। जहाँ एक ओर वन और परती भूमि में कमी आई है, वहीं बोया गया क्षेत्र और दो फसली भूमि में भारी वृद्धि हुई है। यह कृषि विकास, संसाधनों की बेहतर उपलब्धता, और आधुनिक तकनीकों की स्वीकार्यता को दर्शाता है, साथ ही पारिस्थितिक संतुलन बनाए रखने की आवश्यकता पर भी बल देता है।

संदर्भ सूची :

1. Central Ground Water Board. (2021). *Ground Water Year Book – Rajasthan (2020–21)*. Ministry of Jal Shakti, Government of India.
2. District Statistical Office, Sawai Madhopur. (1993). *Statistical Abstract of Sawai Madhopur District 1992-93*. Government of Rajasthan.
3. District Statistical Office, Sawai Madhopur. (2022). *Statistical Abstract of Sawai Madhopur District 2021-22*. Government of Rajasthan.
4. Government of Rajasthan. (2022). *Rajasthan Agricultural Statistics at a Glance 2021-22*. Directorate of Agriculture, Jaipur.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 9, September 2024

|DOI: 10.15680/IJIRSET.2024.1309130|

5. Jain, S. K., & Singh, V. (2016). Impact of Irrigation on Cropping Pattern in Rajasthan: A Study Based on Groundwater Usage. *Indian Journal of Agricultural Economics*, 71(2), 215–223.
6. Kumar, A., & Yadav, R. (2020). Analyzing Land Use Change and Agricultural Productivity in Rajasthan. *International Journal of Environmental Planning and Management*, 6(3), 140–148.
7. Ministry of Agriculture and Farmers Welfare. (2021). *Agricultural Statistics at a Glance 2021*. Government of India.
8. National Remote Sensing Centre (NRSC). (2021). *Land Use Land Cover Data (2005–2020)*. Indian Space Research Organisation (ISRO), Hyderabad.
9. Rao, M. S., & Meena, L. B. (2017). Temporal Analysis of Agricultural Land Use in Rajasthan. *Geographical Review of India*, 79(4), 356–368.
10. Sharma, H. S., & Meena, R. S. (2018). Land Use Dynamics in Eastern Rajasthan: A Case Study of Sawai Madhopur District. *Annals of Rajasthan Geographical Association*, 35(1), 65–72.



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details