A FHIR-Based Consent Management System with DS4P and Break-Glass for EHR: A Case Study on the SATUSEHAT Platform

1st Irfan Musthofa
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
im@irfanmusthofa.com

2nd Rinaldi Munir
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
rinaldi@informatika.org

Abstract*—The SATUSEHAT Platform, Indonesia's national health-data exchange built on the HL7 Fast Healthcare Interoperability Resources (FHIR) standard, currently relies on a general-consent model that does not let patients control access to their records at a granular level. The platform also lacks sensitivity-based data segmentation, a tamper-evident audit trail, and a structured emergency-access protocol. This paper presents the design, implementation, and evaluation of a prototype *Consent and Policy Gateway* that addresses these gaps as a FHIR-native reverse-proxy middleware. The gateway provides granular consent management based on the FHIR Consent Resource, letting patients authorize access by resource type, clinician role, and purpose of use. It enforces access through an Attribute-Based Access Control (ABAC) policy combined with Data Segmentation for Privacy (DS4P) security labels. For emergencies, it offers a break-glass protocol gated by step-up Time-based One-Time Password (TOTP) authentication and short-lived tokens, while every access decision is recorded in an immutable, hash-chained audit log. Black-box testing across fourteen functional and two non-functional test cases passed perfectly. Consent enforcement and versioning behaved correctly, label-V data stayed hidden on the normal path, break-glass tokens were issued only after valid TOTP, the hash chain detected tampering at the exact altered row, and mean access-decision latency was 1.9 s (max 2 s), well under the 10 s target. The results show that granular consent, DS4P segmentation, controlled emergency access, and tamper-evident auditing can be implemented as an interoperable gateway layer in front of a FHIR-native platform.

Keywords—SATUSEHAT Platform, Consent Management System, Break-Glass, FHIR, DS4P, ABAC.

I. Introduction (*Heading 1*)

Modern health information systems face a fundamental challenge, the fragmentation of electronic health records (EHRs) across facilities, which causes interoperability barriers, duplicated examinations, and inconsistent clinical information. To address this, HL7 developed the Fast Healthcare Interoperability Resources (FHIR) standard, representing health data as modular resources such as *Patient*, *Observation*, and *Condition* over a RESTful API with JSON/XML payloads, enabling integration across heterogeneous systems [1], [2]. In Indonesia, the Ministry of Health adopted FHIR through the SATUSEHAT Platform as the national milestone for EHR interoperability [3], [4].

Although adopting FHIR is a strategic step, SATUSEHAT's available consent mechanism remains a *general consent*, a single authorization with no granularity [3]. Patients therefore cannot specify who may access which data, when, or for what purpose, creating privacy risks for sensitive categories such as mental-health or communicable-disease records. Internationally, the HIPAA Privacy Rule's *minimum necessary* principle requires access to be limited to what is strictly needed for a given clinical or administrative purpose [5], demanding granular, selective control rather than blanket consent. The Patient-Accessible EHR (PAEHR) paradigm similarly strengthens the patient's role as an active manager of their data [6]. In Indonesia, comparable principles are embodied in Law No. 27/2022 on Personal Data Protection [7], Minister of Health Regulation No. 24/2022 on Medical Records [8], and Ministerial Decree No. 133/2023 on SATUSEHAT [4], all of which mandate confidentiality, integrity, and rights such as consent withdrawal.

To bridge the gap between regulatory mandate and technical implementation, prior work has proposed more granular access control. A systematic review of EHR access-control solutions found Attribute-Based Access Control (ABAC) to be the most popular authorization model, surpassing Role-Based Access Control (RBAC) [9], yet emergency-access and patient-consent management remain under-addressed. RBAC has significant limitations for dynamic care settings [10], whereas Acute-Care ABAC (AC-ABAC) applies contextual attributes to grant and revoke temporary access during emergencies [11]. Strict consent-based control, however, raises a tension in emergencies when a patient cannot consent in real time, this motivates a *break-glass* protocol, authorized documented emergency access that preserves accountability. To keep accountability intact, the audit trail must be tamper-evident, for which the hash-chaining concept derived from blockchain architectures is sufficient at this scope [12].

This paper presents a prototype *Consent and Policy Gateway* that implements a granular consent management module on the FHIR resources, enforces access automatically combining DS4P security labels with ABAC, provides an accountable break-glass module with step-up TOTP, time-limited tokens, and a hash-chained immutable audit log. The work is a proof-of-concept implementation on existing standards and not a new framework, scoped to functional correctness rather than production integration, full encryption, penetration testing, or scalability.

II. Related Works

A. FHIR and DS4P

FHIR exchanges data as discrete *resources*, each addressable via a unique URL over HTTP/REST, which

makes it the natural substrate for a vendor-agnostic access layer [1]. Not all patient data carry equal sensitivity, therefore HL7's Data Segmentation for Privacy (DS4P) attaches security labels to FHIR resources (e.g., in meta.security) so that a policy engine can enforce access without encrypting the payload itself [13]. DS4P labels derive from the HL7 *Confidentiality* code system, which defines six levels (U, L, M, N, R, V). The proposed confidentiality levels serve as guidelines and may be refined according to operational needs.

B. Access Control Models

Four access-control models are commonly discussed: Discretionary (DAC), Mandatory (MAC), Role-Based (RBAC), and Attribute-Based (ABAC) [17], [18]. DAC ties decisions to object owners through access-control lists, it is flexible but hard to audit centrally. MAC enforces centrally managed classification labels, it is strong for protecting sensitive data but rigid. RBAC grants access by role where easy to manage but static and context-blind. ABAC decides from subject, object, and environment attributes, it is the finest-grained and most context-adaptive, at higher design complexity. Ranked by granularity, RBAC is coarsest, DAC low-to-moderate, MAC moderate, and ABAC finest. ABAC is thus best suited to fine-grained consent, and DS4P conceptually borrows MAC's managed-label principle and pairs it with ABAC to obtain the highest granularity together with emergency-context adaptivity, a combination none of DAC, MAC, or RBAC reaches alone. ABAC applied directly to FHIR has been proposed where native OAuth/UMA mechanisms proved insufficient for fine-grained control [16], a key conceptual basis for this work.

C. Break-Glass and Tamper-Evident Audit

Conventional break-glass grants unrestricted emergency access and is frequently abused due to weak auditing [11]. A ciphertext-policy attribute-based encryption (ABE) break-glass protocol introduced temporary emergency tokens that are automatically revoked after the care session [14]. For accountability, the audit trail must resist tampering. Blockchain offers decentralization, persistency, anonymity, and auditability [15]. In healthcare, however, full records are not stored on-chain, only hashes or access events are, for integrity [12]. This work adopts only the persistency and auditability value through hash-chaining on a single immutable log, deliberately omitting decentralization and consensus as out of scope.

III. Proposed Solution

A. Design Rationale and Scope

The current SATUSEHAT general-consent model leaves four gaps: no fine-grained consent, no DS4P-based segmentation of sensitive data, no tamper-evident audit trail, and no structured break-glass for emergencies. The proposed *Consent and Policy Gateway* addresses these with one coherent design built on four pillars, granular consent on the FHIR Consent Resource, automatic policy enforcement combining DS4P security labels with ABAC, an accountable break-glass path with step-up TOTP and short-lived tokens, and a hash-chained immutable audit log.

DS4P is chosen for segmentation because it is the official HL7 security-labelling standard attached to FHIR resource metadata, needs no proprietary label scheme, and is interoperable across facilities [13]. As established in Section II, the finest access granularity comes from pairing ABAC with MAC-style managed labels; DS4P supplies exactly that labelling principle. The system uses the top three HL7 Confidentiality levels: N (normal), R (restricted), V (very restricted) because they map one-to-one onto three enforcement branches. N is allowed with an active consent, R requires a matching consent rule, and V is denied on the normal path.

The FHIR-based approach is a prerequisite rather than a preference, because SATUSEHAT is FHIR-native [4], a non-FHIR design would make consent non-portable, force ad-hoc labels, and forfeit the gateway's cross-facility value. For audit integrity, only the persistency and auditability of blockchain are adopted through hash-chaining on one immutable log, decentralization and consensus are deliberately omitted as out of scope [12], [15]. The scope is a proof of concept focused on the correctness of access-control logic, not production integration, full encryption, penetration testing, or scalability.

B. Conceptual Architecture

The gateway has two core components: a *Consent Management System* acting as the policy middleware before any request reaches the centralized data source, and a *Break-Glass Protocol* for emergencies. Architecturally it sits as an API gateway or reverse-proxy middleware in front of a FHIR-native platform such as SATUSEHAT, staying compatible with existing integration without intrusive changes to hospital EHRs.

A Patient Portal manages granular consent (grant/update/revoke) and a Clinician Portal issues data-access requests. All requests, including normal and break-glass, pass through the gateway to be verified, decided, and logged. Fig. 1 depicts the underlying infrastructure of the proposed system.

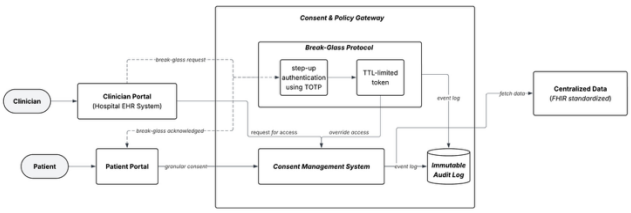

Fig. 1. Proposed conceptual solution

C. Business Process

As shown in Fig. 2, the flow begins when a patient registers at a facility. A clinician opens the patient list and requests the EHR, every request passes through the gateway for evaluation against the patient's consent and DS4P labels. If a matching consent exists, the normal path shows N- and R-labelled data while V is hidden. If no consent exists or the patient is in an emergency, the clinician activates the break-glass path with a written reason and TOTP verification. After obtaining the relevant data, the clinician performs the examination and records results with appropriate sensitivity labels. When the examination ends or the token expires, access is revoked automatically.

The solution's contribution as greyed-coloured box thus lies in access request, policy enforcement, controlled emergency access, labelling on write, access revocation, and audit logging, while registration and clinical actions follow existing processes.

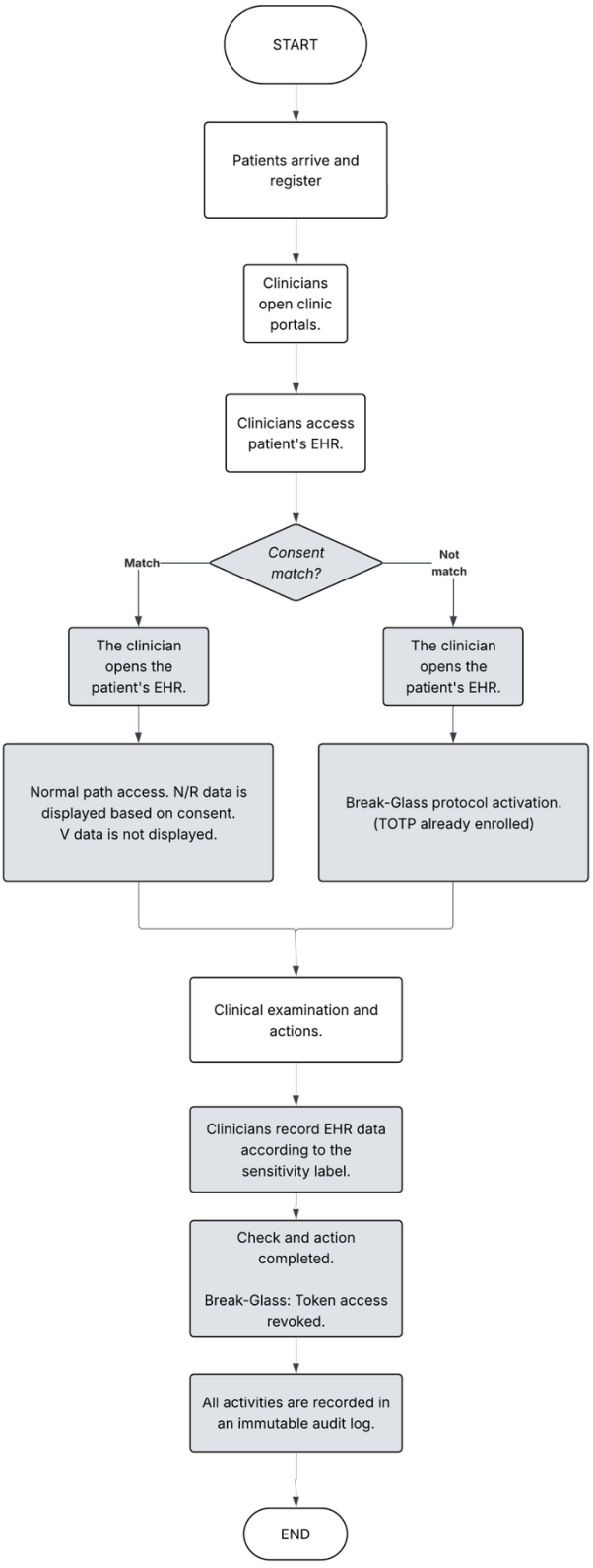

Fig. 2. Proposed business process

D. Use Cases

As shown in Fig. 3, three actors interact with the gateway: Patient (manages consent, views own data and access history), Clinician (browses patients, accesses/records EHR through the normal path, enrols TOTP, performs break-glass), and Admin (reviews audit log and verifies the hash chain, manages clinician TOTP, configures break-glass duration). Twelve use cases (UC-01–UC-12) capture these interactions.

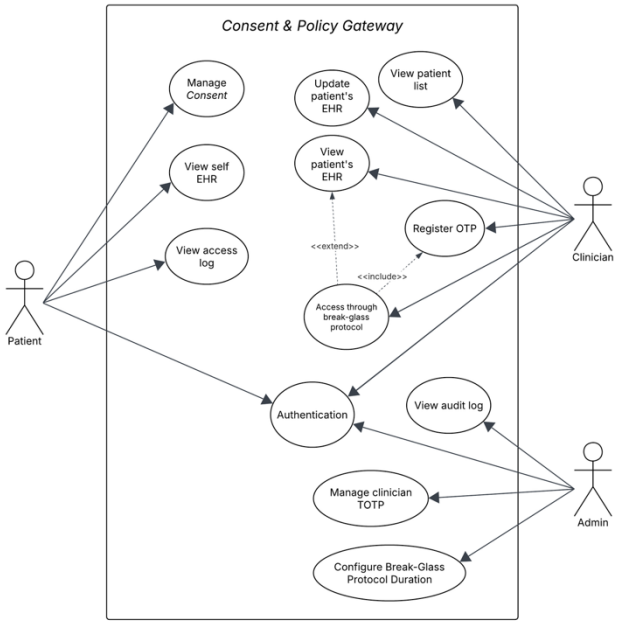

Fig. 3. System use cases

IV. Implementation

The prototype comprises three integrated components: a *FastAPI* (Python) backend acting as the Consent and Policy Gateway (modules for Consent Management, Break-Glass, the hash-chained Immutable Audit Log, and auth), a *Next.js* (React) frontend providing the Patient, Clinician, and Admin Portals, mobile-first (iPhone 14 Pro Max viewport) with a minimal, function-first UI, and a *PostgreSQL* database. PostgreSQL was chosen for JSONB storage of whole FHIR resources, generated identity columns for audit ordering, and advisory locks that keep the hash chain consistent. Access uses the SQLAlchemy 2.0 ORM with Alembic migrations. The schema has eight tables across four groups: identity (users), FHIR clinical data (fhir_patients, fhir_resources), consent (consents, consent_rules), and security with accountability (audit_events, break_glass_tokens, system_settings). The API uses FastAPI with Pydantic typing and auto-generated OpenAPI docs, with python-jose (HS256 JWT), pyotp (TOTP), and bcrypt. Endpoints are grouped into /auth, /consents, /fhir, /break-glass, /audit, and /admin/settings. The resulting schema and its table relations are shown in Fig 4.

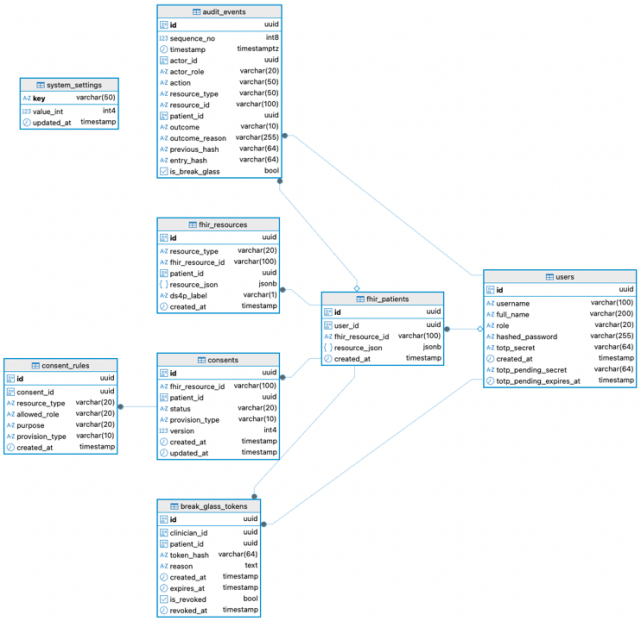

Fig. 4. ERD of the system

A. *Consent Management*

Policy enforcement triggers on the four clinical-read endpoints (observations, conditions, medications, specimens). The clinician read-path runs three sequential stages:

- Stage A (Preflight Consent): Fetches one active consent for the target patient. If none exists, it returns HTTP 403 with a *no-consent* reason without proceeding (fail-safe deny).
- Stage B (Loading Resources): retrieves the requested FHIR resources, sorted by clinical date.
- Stage C (Resource Classification): evaluates each resource by its DS4P label and writes exactly one audit row per resource. The three branches are:
    - label N: allow on active consent alone (ACCESS_GRANTED)
    - label R: allow only if a consent_rule matches the requester's attributes (role, resource type, purpose), else deny with rule_miss
    - label V: always deny on the normal path (label_v), accessible only through a 15-minute creator window for the clinician who just created it.

Two design properties make this path predictable and auditable. First, evaluation is fail-safe, the absence of an active consent in Stage A, or of a matching rule for a label-R resource, yields denial rather than a default allow. Second, Stage C writes exactly one audit row per evaluated resource, so every decision, allow or deny, carries an independent forensic trace tied to its reason code (ACCESS_GRANTED, rule_match, rule_miss, or label_v).

When several consents could apply to the same request, a precedence rule resolves the conflict by letting a more specific consent override a more general one, so the patient's narrowest intent prevails. The complete read-path is shown in Fig. 5.

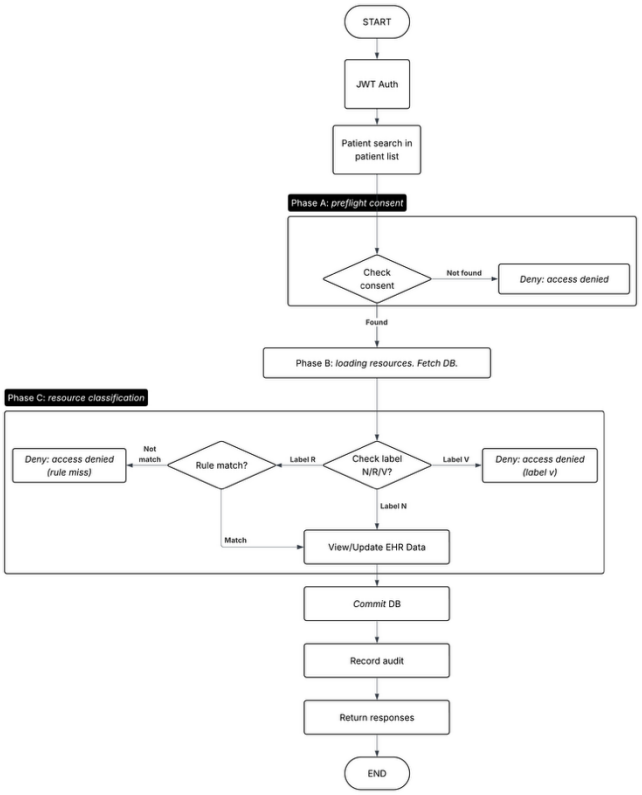

Fig. 5. Consent Management System flowchart

B. *Break-Glass Protocol*

Break-glass lets a clinician read patient data even when consent does not permit it, without sacrificing accountability. It has two phases, token issuance and token use when reading FHIR resources.

A X-Break-Glass-Token header changes gateway behaviour, it skips Stage A and allows classification of N- and R-labelled resources without a consent rule. V remains denied even with an active break-glass token. A token is issued only if all hold:

- The actor has a sub-clinician role (doctor, pharmacist, nurse, psychologist, dentist, therefore patient and admin are excluded).
- TOTP is enrolled.
- The six-digit TOTP is valid at request time
- A written reason is supplied.

The module emits four audit actions:

- BREAK_GLASS_REQUEST (issued, with a TTL snapshot)
- BREAK_GLASS_DENIED (invalid TOTP)
- BREAK_GLASS_EXPIRED (expired token)
- BREAK_GLASS_ACCESS (one row per resource read).

  All appended to the same hash chain, so emergency-path entries are as accountable and tamper-detectable as normal-path entries. Fig. 6 illustrates the break-glass authorization flow, including its main failure paths.

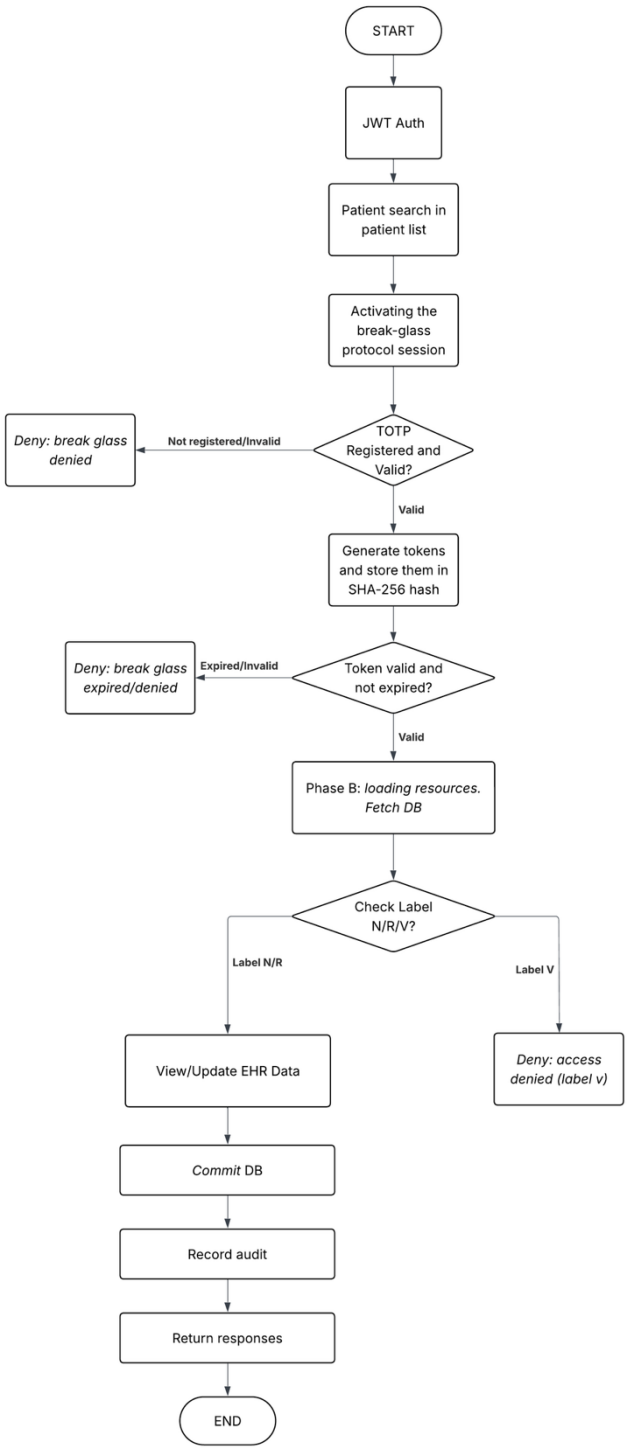

Fig. 6. Break-Glass Protocol flowchart

C. Immutable Audit Log

After classification, the audit module appends one entry per record read, computing entry_hash = SHA-256(previous_hash ‖ canonical_json(event)). Any insertion, deletion, or modification of a single entry breaks the chain and is detected at verification via GET /audit/verify, which reports the position of the broken entry, yielding blockchain-like tamper-evidence and auditability on one centralized log without a distributed ledger [12], [15].

D. Step-Up TOTP

OTP is an umbrella term. HOTP is counter-based [19]; SMS/email OTP relies on a delivery channel, while TOTP is time-based and expires roughly every 30 s [20]. TOTP is computed locally from a shared secret and the current time, needing no delivery channel.

TOTP was chosen for the Indonesian SATUSEHAT context for four reasons, it works offline with no SMS-gateway cost, suiting regional facilities with limited connectivity and budget, it is immune to SIM-swap and SS7 interception, SMS-OTP is even classified as a *restricted authenticator* by NIST SP 800-63B [21], its zero delivery latency fits time-critical break-glass, and it is an open standard (RFC 6238) interoperable with common authenticator apps, avoiding vendor lock-in.

TOTP is *not* used for ordinary login or consent-permitted normal access, it is required only when activating break-glass, as step-up authentication for enrolled sub-clinicians. Enrollment generates a 160-bit Base32 secret rendered as a QR code and verification tolerates a ±30 s server–device clock skew.

E. Dual TTL Rationale

Two short-lived tokens exist: the *break-glass token* and the *creator-window token*. Both defaulting to 15 minutes but configured differently. The break-glass duration is admin-tunable from 5 to 60 minutes (5-minute steps), because three compensating controls offset any extension, such as step-up TOTP at request time, an auditable is_break_glass flag, and revocation via is_revoked. So operational flexibility does not lower the security posture.

The 60-minute cap is far more conservative than the 8-hour-to-1-day emergency windows cited in industry guidance [22]. The creator-window duration is fixed at 15 minutes because it has no second authentication gate (it relies only on a signed, stateless JWT that cannot be revoked before expiry), so lengthening it would widen the attack surface with no compensating control. 15 minutes aligns with EHR session-timeout consensus [23], [24] and is stricter than the 30-minute NIST AAL2 reauthentication bound [21]. For longer access to V-labelled data, the correct path is break-glass with step-up TOTP.

F. User Interface

The frontend uses Next.js 16, Tailwind CSS, and shadcn/ui. Cross-role design choices include click-and-hold buttons for destructive actions (consent revocation, TOTP reset, chain verification), a six-digit InputOTP component, and DS4P sensitivity shown as colored badges on the resource viewer.

The Patient Portal exposes three pages, granular consent management, a personal record viewer in which the patient sees all data including label V without consent or break-glass, and an access log of who read their records.

The Clinician Portal centers on a patient list and a record-detail page where the clinician picks a purpose of use and is offered a break-glass banner when every resource is denied. The Admin Portal provides the filterable audit log, a click-and-hold chain-verification action, break-glass duration settings, and clinician TOTP management. Fig. 7 shows the clinician break-glass activation screen.

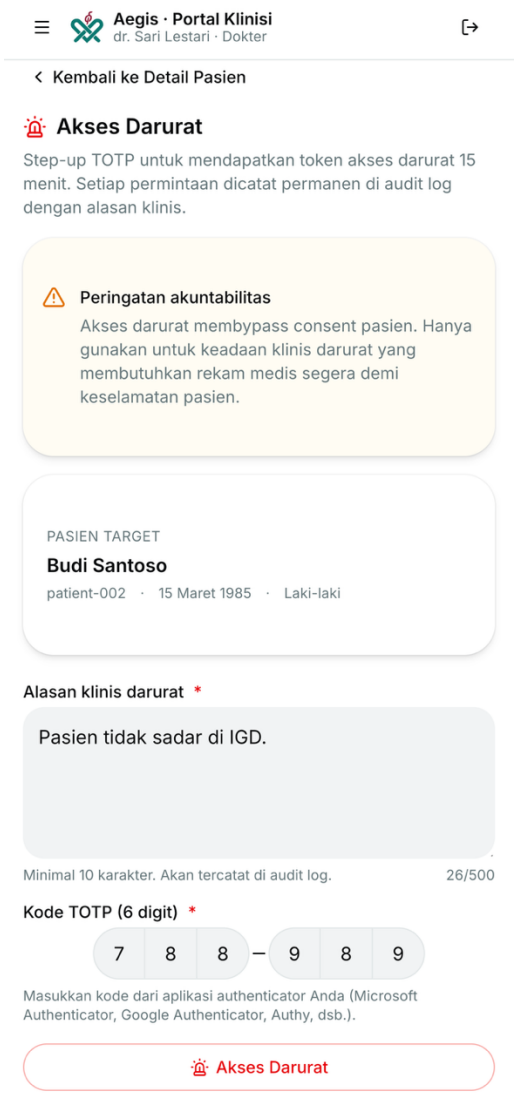

Fig. 7. Break-Glass activation screen

V. Evaluation

Evaluation verifies all functional and non-functional requirements using black-box testing, each functional capability exercised with positive and negative scenarios. All tests were run manually on a local development environment (Apple MacBook Pro 16″, M1 Pro, 16 GB), with verification at three layers: HTTP response code, database state, and audit-row completeness.

A. Functional Results

All fourteen functional test cases passed (Table III). For consent management, a consent is stored with version 1 and CONSENT_CREATED, updates apply versioning so the old version becomes inactive, and revocation closes access on the next request (TC-01–TC-03).

For granular consent, access is granted on rule_match and denied at preflight with a single no_consent row (TC-04–TC-05). For DS4P enforcement, N/R resources meeting the rule are accessible while V stays hidden, whereas invalid attributes are denied via rule_miss/label_v (TC-06–TC-07). On the emergency path, a break-glass token is issued only after valid TOTP.

TABLE I. Test Cases

ID	*Scenario*	*Status*
TC-01	Create new consent	Pass
TC-02	Update consent (versioning)	Pass
TC-03	Revoke consent	Pass
TC-04	Access per consent rule	Pass
TC-05	Access without active consent	Pass
TC-06	DS4P enforcement, valid attr. (N/R)	Pass
TC-07	DS4P enforcement, invalid attr. (rule-miss/V)	Pass
TC-08	Break-glass success	Pass
TC-09	Break-glass, invalid OTP	Pass
TC-10	Break-glass, expired token	Pass
TC-11	Audit completeness (chain valid)	Pass
TC-12	Tamper detection (chain broken)	Pass
TC-13	Patient portal functionality	Pass
TC-14	Clinician portal functionality	Pass
TC-01	Create new consent	Pass
TC-02	Update consent (versioning)	Pass
TC-03	Revoke consent	Pass

For audit integrity, chain verification is valid when intact and reports the broken_at position when a row is altered directly in the database, with both outcomes shown in Fig. 8 (TC-11–TC-12). Finally, both the patient and clinician portals function without error (TC-13–TC-14).

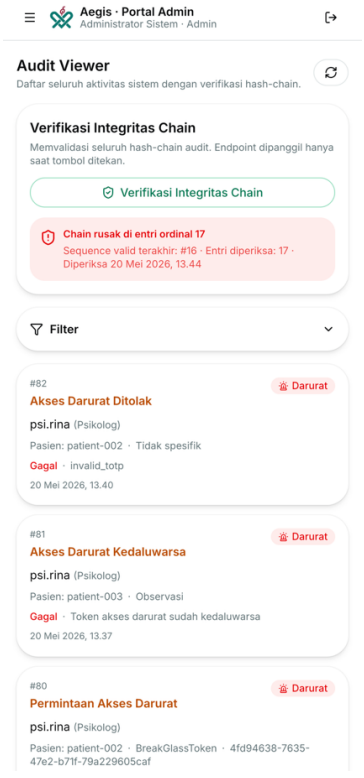

Fig. 8. Broken audit chain

B. Non-Functional Results

TOTP reliability shows that varying device time, time zone, and connectivity (Table IV) confirms that verification depends on absolute time accuracy versus the server, not on the displayed time zone: zone differences (scenarios 4–5), online or offline, do not affect the result as long as absolute time is correct, because TOTP is computed from Unix/UTC time and is therefore zone-independent. Conversely, drift beyond the ±30 s window (scenarios 2, 7) is rejected, while in-window drift (1, 6) is accepted. Verification also succeeds offline (3, 5). This validates correct integration and enforcement of TOTP in the break-glass context, not the TOTP algorithm itself.

TABLE II. TOTP TEST CASES

Scenario	***Device time***	***Connection***	***Status***
Within tolerance window	shift ≤ 30 s	Online	Accepted (correct)
Outside tolerance window	shift > 30 s	Online	Rejected (correct)
No network connection	absolute time correct	Offline	Verified (correct)
Different device time zone	zone differs, abs. time correct	Online	Accepted (correct)
Different device time zone	zone differs, abs. time correct	Offline	Accepted (correct)
Manual time change, in tolerance	phone clock ±≤ 30 s	Online	Accepted (correct)
Manual time change, out of tolerance	phone clock > 30 s	Any	Rejected (correct)

Across ten consecutive requests on the local environment, response time ranged 1–2 s with a 1.9 s mean, including reads that traverse the full ABAC and DS4P enforcement pipeline, well under the 10 s target. All fourteen functional and two non-functional test cases passed at 100% with no regressions or blocking defects, confirming that granular consent, DS4P segmentation, the break-glass protocol, and hash-chained auditing operate as designed. The limitations are no penetration or load testing and no integration with the live SATUSEHAT Platform, follow from the proof-of-concept scope.

VI. CONCLUSION

This work designed, implemented, and evaluated a FHIR-based *Consent and Policy Gateway* that closes four gaps in the current SATUSEHAT general-consent model. Granular consent on the FHIR Consent Resource lets patients control access by resource type, clinician role, and purpose (verified by TC-01–TC-05). DS4P security labels (N/R/V) combined with ABAC enforce access to sensitive data automatically, hiding V on the normal path (TC-06–TC-07).

An accountable break-glass module with step-up TOTP and short-lived tokens provides controlled emergency access (TC-08–TC-10), while a hash-chained immutable audit log makes every decision independently traceable and detects tampering at the exact altered row (TC-11–TC-12), both portals function correctly (TC-13–TC-14). Non-functionally, TOTP behaved correctly across time, zone, and connectivity changes (NFR-1) and access-decision latency averaged 1.9 s, far below the 10 s target (NFR-2).

All sixteen test cases passed demonstrating the prototype's feasibility as a vendor-agnostic middleware layer that can strengthen access control, sensitive-data protection, and auditability in front of a FHIR-native national platform. Future work includes integration with the live SATUSEHAT Platform, comprehensive security testing (pentest, SAST, DAST), end-to-end medical-data encryption, and load/scalability testing.

REFERENCES

[1] M. Ayaz, M. F. Pasha, M. Alzahrani, R. Budiarto, and D. Stiawan, "The Fast Health Interoperability Resources (FHIR) standard: Systematic literature review of implementations, applications, challenges and opportunities," *JMIR Med. Inform.*, vol. 9, no. 7, p. e21929, 2021.

[2] P. Tabari, G. Costagliola, M. De Rosa, and M. Boeker, "State-of-the-art FHIR-based data model and structure implementations: A systematic scoping review," *JMIR Med. Inform.*, vol. 12, p. e58445, 2024.

[3] Kementerian Kesehatan Republik Indonesia, "SATUSEHAT Platform — Keamanan dan akses data pasien," 2025. [Online]. Available: https://faq.kemkes.go.id/faq/bagaimana-satusehat-platform-menjaga-keamanan-data-pasien

[4] Kementerian Kesehatan Republik Indonesia, "Keputusan Menteri Kesehatan No. HK.01.07/MENKES/133/2023 tentang Integrasi Data Kesehatan Nasional melalui SATUSEHAT," 2023.

[5] U.S. Dept. of Health and Human Services, Office for Civil Rights, "Minimum necessary requirement [45 CFR 164.502(b), 164.514(d)]," HIPAA Privacy Rule Guidance, 2003.

[6] T. Kariotis, M. Prictor, K. Gray, and S. Chang, "Patient accessible electronic health records and information practices in mental healthcare contexts: A scoping review," *J. Med. Internet Res.*, vol. 27, 2025.

[7] Pemerintah Republik Indonesia, "Undang-Undang No. 27 Tahun 2022 tentang Pelindungan Data Pribadi," 2022.

[8] Kementerian Kesehatan Republik Indonesia, "Peraturan Menteri Kesehatan No. 24 Tahun 2022 tentang Rekam Medis," 2022.

[9] U. N. Cobrado, S. Sharief, N. G. Regahal, E. Zepka, M. Mamauag, and L. C. Velasco, "Access control solutions in electronic health record systems: A systematic review," *Inform. Med. Unlocked*, vol. 49, p. 101552, 2024.

[10] M. Carvalho and P. Bandiera-Paiva, "Health information system role-based access control current security trends and challenges," *J. Healthc. Eng.*, vol. 2018, pp. 1–8, 2018.

[11] M. T. de Oliveira, Y. Verginadis, L. H. A. Reis, E. Psarra, I. Patiniotakis, and S. D. Olabarriaga, "AC-ABAC: Attribute-based access control for electronic medical records during acute care," *Expert Syst. Appl.*, vol. 213, p. 119271, 2023.

[12] R. Sarode, Y. Watanobe, and S. Bhalla, "A blockchain-based approach for audit management of electronic health records," in *Proc. Int. Conf.*, 2023, pp. 86–94.

[13] HL7 International, "FHIR Data Segmentation for Privacy (DS4P) Implementation Guide," 2025. [Online]. Available: https://build.fhir.org/ig/HL7/fhir-security-label-ds4p/

[14] M. Tuler de Oliveira, A. Bakas, E. Frimpong, A. Groot, H. Marquering, A. Michalas, and S. Olabarriaga, "A break-glass protocol based on ciphertext-policy attribute-based encryption to access medical records in the cloud," *Ann. Telecommun.*, vol. 75, 2020.

[15] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Serv.*, vol. 14, no. 4, pp. 352–375, 2018.

[16] S. Mukherjee, I. Ray, I. Ray, H. Shirazi, T. Ong, and M. Kahn, "Attribute based access control for healthcare resources," in *Proc. ACM Workshop ABAC*, 2017, pp. 29–40.

[17] R. S. Sandhu and P. Samarati, "Access control: Principle and practice," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 40–48, 1994.

[18] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (ABAC) definition and considerations," NIST SP 800-162, 2014.

[19] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, "HOTP: An HMAC-based one-time password algorithm," IETF RFC 4226, 2005.

[20] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "TOTP: Time-based one-time password algorithm," IETF RFC 6238, 2011.

[21] P. A. Grassi et al., "NIST Special Publication 800-63B: Digital identity guidelines — Authentication and lifecycle management," NIST, Gaithersburg, MD, 2020.

[22] Yale University HIPAA Privacy Office, "Break glass procedure: Granting emergency access to critical ePHI systems," 2025. [Online]. Available: https://hipaa.yale.edu/security/break-glass-procedure-granting-emergency-access-critical-ephi-systems

[23] Accountable HQ, "Session timeout requirements for EHR systems: 2026 compliance standards and best practices," 2026. [Online]. Available: https://www.accountablehq.com/post/session-timeout-requirements-for-ehr-systems-2026-compliance-standards-and-best-practices

[24] Censinet, "HIPAA compliance: Session timeout rules," 2024. [Online]. Available: https://censinet.com/perspectives/hipaa-compliance-session-timeout-rules