V nadaljevanju vam pošiljam tehnično specifikacijo Better Auditorja.

1. Namen in arhitektura

Better Auditor je osrednja komponenta Better Platforme za zbiranje revizijskih sledi. Revizijske zapise prejema iz zalednih storitev in jih trajno shrani. Nato jih obdela v t. i. »use case«. En use case predstavlja eno uporabnikovo dejanje, na primer odprtje povzetka pacienta, urejanje zdravila ali ustvarjanje uporabnika. Use casei so na voljo za iskanje in analizo prek REST API-ja, po želji pa tudi za prevzem v zunanje sisteme.

Obdelava poteka v petih korakih:

1. zapis revizijskega zapisa v bazo (takoj),
2. uvrstitev v čakalno vrsto za obdelavo (takoj),
3. paketna obdelava (privzeto vsakih 5 minut, nastavljivo),
4. gradnja use casea in izluščenje identifikatorjev,
5. indeksiranje v OpenSearch in po želji objava v Kafko.

Revizijski zapis je v iskalnem API-ju običajno na voljo v 5–10 minutah od dogodka. Obdelava teče asinhrono, zato ne vpliva na odzivnost kliničnih aplikacij.

2. Viri revizijskih zapisov

- **EHR Server**: operacije nad kliničnimi podatki (openEHR) – branje, ustvarjanje, posodabljanje, brisanje in podpisovanje kompozicij, AQL poizvedbe, shranjene poizvedbe (views), operacije nad EHR, mapami in statusom EHR. Revidirajo se tudi operacije, ki ne posegajo v klinične podatke.
- **Demographics Server**: vse FHIR operacije (create, read, update, delete, search, transakcije in razširjene operacije) nad viri, kot so Patient, Practitioner, Organization ipd.
- **User Management**: upravljanje uporabnikov, skupin, vlog in članstev.
- **Simple Tasks**: upravljanje opravil.

3. Struktura revizijskega zapisa

Polje	Tip	Opis	Obvezno
application	String	Aplikacija, ki je ustvarila zapis	Da
methodName	String	Revidirana operacija	Da
requestTimestamp	ZonedDateTime	Čas začetka operacije	Da
arguments	String (JSON)	Argumenti operacije	Ne
response	String / JSON	Odgovor operacije (odvisno od konfiguracije)	Ne
errorMessage	String	Napaka; pri pisalnih operacijah pomeni, da spremembe niso bile shranjene	Ne
duration	Long	Trajanje operacije v ms	Ne
username / userId / userFullName	String	Uporabnik, ki je izvedel operacijo	Ne
patientId	String	Identifikator pacienta	Ne

Polje	Tip	Opis	Obvezno
ehrId	String	Identifikator EHR	Ne
sessionId	String	Identifikator seje	Ne
originIp	String	IP naslov, s katerega je prišla zahteva	Ne
requestId	String	Enolični identifikator zahteve za sledenje	Ne
useCaseId / useCaseName	String	Povezuje vse zahteve, ki izhajajo iz istega uporabnikovega dejanja	Ne
tokenClaims	Object	Atributi iz uporabnikovega žetona (JWT)	Ne
additionalInfo	Object	Dodatni atributi zaledne storitve in obogatitve Auditorja	Ne

Primer zapisa (ustvarjanje kompozicije):

json

Plain Text

```
{  "application": "Better Platform",  "requestTimestamp": "2024-01-
09T12:30:00Z",  "methodName":
"CompositionResource.createComposition(..)",  "arguments":
"{\"templateId\":\"medication.v1\",\"lifecycleStateString\":\"complete\"}",
  "username": "john.doe",  "userId": "user-123",  "userFullName": "John
Doe",  "duration": 245,  "ehrId": "550e8400-e29b-41d4-a716-
446655440000",  "response":
"{\"compositionUid\":\"composition::1\"}",  "additionalInfo":
"{\"organisationId\":\"CLN-1234\",\"organisationName\":\"Central Health
Clinic\"}"}
```

Posebnosti posameznih virov:

- **Demographics Server** v polju `arguments` vrne tip FHIR operacije, različico FHIR (DSTU2–R5) in tip vira. Pri operacijah nad več pacienti (iskanje, transakcije) navede vse pacientove reference. V `additionalInfo` je tudi identifikator najemnika (tenantId).
- **User Management** v polju `additionalInfo` vrne tip in opis dejanja, ciljnega uporabnika in njegov status, ime in e-pošto pred spremembo, skupino in dodeljene vloge ter lokacijo in delovišče.

Obogatitev zapisov: Auditor lahko ob prejemu zapis s konfiguracijo dopolni z atributi iz uporabnikovega žetona (npr. organizacija, izdajatelj žetona). Prav tako lahko pridobi dodatne podatke iz Demographics Serverja ali terminološkega strežnika (npr. ime organizacije na podlagi njenega identifikatorja).

4. Obdelava v use case

Zahteve se v isti use case združujejo prek HTTP glave `Use-Case-Id`, ki jo nastavi odjemalska aplikacija. Zapise brez tega identifikatorja lahko Auditor samodejno umesti v svoj use case. Tako nobena operacija ne ostane zunaj iskalnega API-ja.

Vsak klic znotraj use casea se zapiše v enotni obliki:

- **callType**: C (ustvarjanje), R (branje), U (posodobitev), D (brisanje),
- **name**: tip vira ali operacije (npr. COMPOSITION, PATIENT, EHR, VIEW),
- **info**: dodatne informacije (npr. predloga, identifikator, razlog brisanja).

Primeri:

- logično brisanje kompozicije → D, COMPOSITION, »comment: Entered in error, id: …«
- branje EHR → R, EHR
- ustvarjanje FHIR vira Patient → C, PATIENT

Vsakemu use caseu Auditor pripne identifikatorje, po katerih je mogoče iskati:

- identifikatorji pacientov in EHR,
- uporabnik, ki je izvedel operacijo (`user|`),
- ciljni uporabnik (`targetuser|`),
- organizacija (`organisationid|`),
- identifikator zahteve (`requestid|`),
- vloga uporabnika v organizaciji (`descriptiverole|`),
- identifikator načrta oskrbe (`careplan|`).

Za sisteme tretjih oseb je mogoče obdelavo nastaviti s konfigurabilnimi JSON preslikavami brez sprememb kode.

5. Vmesniki za dostop do podatkov

- **REST API** za iskanje use caseov:
  - `GET /rest/view/patient?patientId=…` – po pacientu,
  - `GET /rest/view/user?userName=…` – po uporabniku,
  - `GET /rest/view/search?ids=…` – po več identifikatorjih hkrati.
- **OpenSearch**: use casei so indeksirani za iskanje in analitiko.
- **Kafka** (po želji): objava obdelanih use caseov v Kafka topic, npr. za prevzem v SIEM ali drug zunanji sistem.
- **Nadzor**: metrike so na voljo v formatu Prometheus.

Administrativni del REST API-ja (upravljanje JSON preslikav) je zaščiten z avtentikacijo in posebno vlogo.