

TicketTriageAI

Scopo

Realizzare un sistema event-driven che, a partire da email di ticket in ingresso, effettui:

- ingestione strutturata del contenuto
 - validazione + idempotenza
 - classificazione AI (categoria, severità, confidence..)
 - decisione “auto vs human review”
 - persistenza su db
 - visualizzazione tramite web dashboard (readonly)
-

In-scope (MVP)

Ingresso ticket

- Fonte: Email (mailbox/shared mailbox)
- Filtro: mittente/oggetto (es subject contiene la parola “ticket”)
- Orchestrazione: Azure Logic App trigger su email
- Output: JSON **TicketIngested** inviato via HTTP alla function **Ingest**

Ingest API (Function HTTP Trigger)

- Endpoint: POST `api/tickets/ingest`
- Funzioni:
 - . validazione campi obbligatori
 - . idempotenza su messageld
 - . invio del messaggio su **Azure Service Bus queue**
 - . risposta 202 Accepted con correlationId

Processing (Function ServiceBus Trigger)

- Trigger su queue (es. tickets-ingest)
- Funzioni:
 - . normalizzazione testo (pulizia quote/firme)
 - . chiamata **Azure AI Foundry (GPT)** con output JSON
 - . calcolo flag needsHumanReview (confidence threshold + regole)
 - . persistenza su DB di payload originale, output AI e decisione finale stato

Dashboard Web (read-only)

Visualizza ticket triagiati

- Data, mittente, subject
- categoria, severità
- confidence
- stato: processed / needsReview / failed ecc

Dettaglio ticket:

- testo pulito
- estrazioni AI (entitites / summary)
- audit base (correlationId, timestamps)

Notifiche

Se needsHumanReview = true o severity alta:

- invio notifica (Teams/email) con link al dettaglio in dashboard
-

Out-of-scope (MVP)

- Crezione/modifica ticket su Jira/ServiceNow
 - Allegati: parsing contenuto allegati
 - Autorizzazioni avanzate
 - RAG con knowledge base
 - Workflow di approve/override direttamente in UI
-

Requisiti funzionali

RF-01 - Filtraggio email

Il sistema deve processare solo email conformi ai filtri configurati

RF-02 - Validazione input

Se mancano campi obbligatori (messageld, from, subject, body) l'ingest deve rifiutare con errore 400 BadRequest

RF-03 - Idempotenza

A parità di messageld, l'ingest deve evitare duplicati e non pubblicare più volte lo stesso ticket.

RF-04 - Classificazione AI

Il sistema deve produrre un risultato strutturato con

- category
- severity
- confidence

- summary
- entities (opt)
- needsHumanReview

RF-05 - Human review decision

Se confidence sotto soglia o regole di business lo richiedono, needsHumanReview dev'essere true

RF-06 - Persistenza

Ogni ticket dev'essere salvato con stato e audit minimo per tracciabilità

RF-07 - Dashboard

La dashboard deve permettere consultazione lista/dettaglio dei ticket

RF-08 - Dead-letter handling

Se un messaggio fallisce più volte, deve finire in DLQ e risultare failed nel DB o avere un log chiaro

RF-09 - Correlation / Trace

Ogni ticket deve avere un correlation Id propagato: ingest → ServiceBus → processing → DB → dashboard

RF-10 - Confidence threshold configurabile

La soglia (es 0.65) deve stare in config (app settings), non hardcoded

RF-11 - Stato ticket

Definire stati minimi

- Received
- Processed
- NeedsReview
- Failed

Requisiti non funzionali

Scalabilità: processing asincrono via ServiceBus

Affidabilità: retry automatici, DLQ per messaggi falliti

Osservabilità: correlationId end-to-end, logging strutturato (Application Insights)

Sicurezza: nessun segreto nel repo; config via appsettings/Key Vault

Dati e contratti

TicketIngested

- messageId (string, required)
- from (string, required)
- subject (string, required)
- body (string, required)
- receivedAt (datetime, required)
- source = "email"

TicketTriageResult

- category (enum/string)
- severity (P1...P4)
- confidence (double 0...1)
- summary (string)
- entities (object)
- needsHumanReview (bool)

Criteri di accettazione (MVP)

Inviando una email conforme ai filtri, viene generato un record visibile in dashboard.

Email duplicate non creano record duplicati

Ogni ticket ha categoria/severity/confidence valorizzati

I casi NeedsHumanReview sono marcati e (se attivo) notificati
