Trailgun

PACCHETTO DI CONFORMITÀ · DOCUMENTO P06



VERSIONE 2.0 · DICEMBRE 2026

Politica di sviluppo sicuro

Le regole con cui Trailgun progetta, scrive, prova, rivede e pubblica il software: punti unici per le operazioni sensibili, test sulle aree che proteggono i Clienti, nessun segreto nel codice.

CODICE	TITOLARE	REVISIONE	STATO
P06 · POL-06	Responsabile della sicurezza	Annuale	al 01/12/2026

POLITICA P06 · POL-06

Politica di sviluppo sicuro

Le regole con cui Trailgun progetta, scrive, prova, rivede e pubblica il software.

CODICE	TITOLARE	APPROVATA DA	VERSIONE
P06 (POL-06)	Responsabile della sicurezza	Organo amministrativo, [data della delibera]	2.0 del 01/12/2026

REVISIONE	DESTINATARI	CLASSIFICAZIONE
annuale	chi sviluppa per Trailgun; uffici ICT risk e CISO dei Clienti	Riservato · per il Cliente

DOCUMENTI COLLEGATI

P05, P07, P15; procedura 20 (Gestione dei cambiamenti); procedura 23 (Gestione delle chiavi e dei segreti); documento 57 (Deposito del codice: term sheet)

AMBITO

Tutto il codice di Trailgun (app, frontend, infrastruttura, script), le dipendenze, i prompt del registro e gli strumenti di sviluppo, compresi gli assistenti di programmazione basati su IA.

ISO/IEC 27001:2022, ALLEGATO A	NORME PRINCIPALI
8.4, 8.25-8.30, 8.33	Reg. del. (UE) 2024/1774 art. 16; DORA art. 25; GDPR art. 25

1 Scopo e ruoli

La sicurezza del servizio si decide quando si scrive il codice. Questa politica vale per chiunque scriva codice per Trailgun; il Responsabile della sicurezza ne verifica il rispetto e approva le eccezioni.

2 Requisiti e progettazione

2.1 Ogni funzione nuova nasce come opzione, attivabile dalle impostazioni e nel modulo dell'atto; spenta, il flusso resta identico, stesso percorso e stessi prompt; la scelta si congela sull'atto alla creazione.

2.2 Punti unici per le operazioni sensibili: ogni atto nasce da una sola funzione (`creaAtto`); il testo dell'atto si scrive da una sola funzione (`scriviTestoAtto`), che conserva le versioni con autore e motivo; la visibilità dei dati di un cliente passa da un solo predicato; le regole dei ruoli stretti stanno in un solo punto. Una funzione nuova riusa questi punti e non li aggira.

2.3 Ogni uscita nuova verso lo studio (JSON, stream, file, email, risposte degli agenti) rispetta le regole sui dati e sui costi: un campo nuovo con un dato riservato si aggiunge ai filtri esistenti e ha un test.

2.4 Un passo accessorio non fa mai fallire l'atto: a ogni guasto resta il risultato precedente.

2.5 Privacy fin dalla progettazione: si raccoglie e si conserva solo ciò che serve; nessun testo dei documenti nei log o nei registri dei costi; sui fascicoli penali nessun embedding (politica P16).

3 Scrittura del codice

3.1 TypeScript con controllo dei tipi obbligatorio, formattazione uniforme, schema del database cambiato solo con migrazioni numerate.

3.2 Nessun segreto nel repository, nell'immagine o nel template dell'infrastruttura. La scansione dei segreti (gitleaks, con le regole predefinite e le sole eccezioni motivate per falsi positivi) controlla l'intera storia del repository a ogni verifica della CI; un segreto trovato si ruota e si toglie dalla storia, mai si mette in eccezione.

3.3 I prompt stanno solo nel registro dei prompt, non sparsi nel codice; ogni prompt ha una sintesi obbligatoria.

3.4 Gli assistenti di programmazione basati su IA lavorano sul codice sotto il controllo dello sviluppatore, che ne risponde. Il repository non contiene dati dei Clienti e agli assistenti non si danno dati dei Clienti, documenti delle pratiche o segreti.

3.5 Le dipendenze si aggiornano con le richieste settimanali di Dependabot (npm dell'app e dell'infrastruttura, immagini Docker, GitHub Actions), integrate dopo la verifica; gli strumenti scaricati dalla CI sono a versione fissata e controllati con l'impronta SHA-256.

4 Prove

4.1 Ogni pull request verso un branch di ambiente e ogni rilascio superano: controllo dei tipi (server, frontend, infrastruttura), test unitari, test end-to-end, scansione dei segreti e prova di ripristino del database (copie ripristinate e verificate con controllo d'integrità, schema, righe e file).

4.2 Le aree sensibili hanno test dedicati: verifica dei token, isolamento fra studi, ruoli, API pubblica e chiavi, regole sui costi visibili allo studio, marcatura dei contenuti generati, esclusione delle regioni fuori dall'UE.

4.3 Le prove usano dati sintetici, mai dati di production.

4.4 Le prove di sicurezza indipendenti seguono il Percorso: scansione esterna mensile dell'applicazione (tappa R05) e penetration test annuale e dopo ogni cambiamento rilevante dell'architettura (tappa R06).

5 Revisione e pubblicazione

Ogni modifica entra con una pull request verso un branch di ambiente protetto: la verifica deve essere superata prima dell'integrazione, e la revisione del codice si fa con strumenti di revisione automatica e con la lettura dello sviluppatore responsabile. La pubblicazione segue la politica P05. Il codice sorgente e le istruzioni di ricostruzione si depositano presso un agente terzo con aggiornamento trimestrale (Percorso, tappa R09; documento 57).

6 Sviluppo affidato a terzi

Trailgun sviluppa il proprio codice. Se affida parte dello sviluppo a terzi, il fornitore riceve questa politica per contratto, non ha accesso alla produzione né ai dati dei Clienti e il suo codice passa dalla stessa verifica e dalla stessa pull request.

7 Evidenze e registri

CONTROLLO	EVIDENZA	DOVE E QUANDO
Verifica obbligatoria	IN ESSERE tipi, test unitari ed end-to-end, segreti, prova di ripristino	esito in ogni pull request; registro 45
Scansione dei segreti	IN ESSERE gitleaks sull'intera storia, eccezioni motivate nella configurazione	a ogni verifica
Dipendenze e immagini	IN ESSERE Dependabot settimanale; scansione delle immagini al caricamento	tempi della politica P07
Container non root	IN ESSERE utente non privilegiato nell'immagine	a ogni immagine

8 Eccezioni e riferimenti normativi

Eccezioni secondo P01 § 8; nessuna eccezione ammette dati dei Clienti nel repository o nelle prove. Riferimenti: Reg. del. (UE) 2024/1774 art. 16; DORA art. 25; GDPR art. 25; ISO/IEC 27001:2022 controlli 8.4, 8.25-8.30, 8.33.