BONKETIBA CERTIFIED - TRUST PACK PUBBLICO
==========

Documento : BKC-TRUST-1.0.0
Versione pack : 1.0.0 Dossier: 1.0 (2026-08-19)
Ambiente : PREPROD
Stato protocollo : Certified 1.0 - protocollo congelato, in attesa di verifica indipendente
Generato il : 2026-09-29T06:19:40.142Z

AVVERTENZA LEGALE

Bonketiba Certified e un sistema tecnico di evidenza: produce prove verificabili di integrita, sequenza temporale e firma. Non e PEC, non e un servizio fiduciario qualificato eIDAS, non e una firma elettronica qualificata e non attribuisce automaticamente valore legale ai messaggi. Il valore probatorio dipende dall'ordinamento applicabile e dalla valutazione dell'autorita competente.

1. PERIMETRO DELL'ASSESSMENT

In perimetro:
- Firma e verifica dei pacchetti Certified (ECDSA P-256 / SHA-256).
- Ciclo di vita delle chiavi: generazione, custodia, rotazione, ritiro, pubblicazione.
- Isolamento tenant su tutte le server function e sulla coda notifiche.
- Release Gate, catena di audit e riproducibilita delle prove.
- Posizionamento giuridico e comunicazione pubblica di 'Certified'.

Fuori perimetro:
- Sicurezza fisica dell'infrastruttura del fornitore cloud.
- Qualificazione eIDAS o accreditamento PEC (richiede un organismo di valutazione).
- Codice del browser desktop Bonketiba (assessment separato).

Metodo:
- Revisione del codice sorgente dei moduli crittografici e di governance.
- Analisi statica automatizzata della superficie server function (test CI).
- Ispezione delle policy RLS e dei GRANT su ogni tabella pubblica.
- Esecuzione del Release Gate e verifica del determinismo su due run consecutive.
- Verifica indipendente offline di un pacchetto firmato con la sola chiave pubblica.

2. SECURITY ASSESSMENT

Controlli: 20 totali - 12 implementati, 4 parziali, 2 gap, 2 verifica esterna.

2.1 CRITTOGRAFIA - Le prove sono verificabili da terzi senza fidarsi del server?

[IMPLEMENTATO] CR-1 Firma asimmetrica ECDSA P-256 / SHA-256
Manifest e sigilli di catena sono firmati con ECDSA su curva P-256 e digest SHA-256 tramite WebCrypto. La chiave privata non lascia mai il server.
Evidenza: src/lib/certified.server.ts (SIGNATURE_ALG) e verifica browser-side su /release-evidence/verify.

[IMPLEMENTATO] CR-2 Canonicalizzazione deterministica
Il payload firmato e serializzato in forma canonica: due esecuzioni con gli stessi input producono lo stesso hash normalizzato.
Evidenza: Prova pubblica di determinismo su /release-evidence/determinism.

[IMPLEMENTATO] CR-3 Catena di audit concatenata per hash
Ogni evento contiene prev_hash ed entry_hash: la rimozione o l'alterazione di un anello rompe la catena in modo rilevabile.
Evidenza: Verificatore offline HTML e /release-evidence/verify ricalcolano la catena localmente.

[PARZIALE] CR-4 HMAC legacy affiancato alla firma asimmetrica
La firma simmetrica storica resta calcolata per compatibilita con i pacchetti emessi prima della migrazione. Non e sufficiente per la verifica pubblica e non deve essere presentata come tale.
Evidenza: Il verdetto pubblico si basa sulla firma asimmetrica; l'HMAC e solo un controllo interno.

[VERIFICA ESTERNA] CR-5 Revisione crittografica indipendente
Nessun laboratorio esterno ha ancora rivisto lo schema. E il prerequisito dichiarato

prima di qualunque affermazione piu forte.
Evidenza: Aperto: da assegnare a un revisore terzo.
Rischi residui:
- Assenza di marca temporale qualificata di terza parte (RFC 3161 o TSA accreditata).
- Nessuna trasparenza append-only pubblica di tipo Certificate Transparency per i sigilli.
Perimetro per il revisore esterno:
- Correttezza della canonicalizzazione rispetto a input malevoli (unicode, chiavi duplicate).
- Assenza di malleabilita delle firme e corretto uso di IEEE P1363 vs DER.
- Resistenza a rollback della catena e a riordino degli eventi.

2.2 CUSTODIA DELLE CHIAVI - Chi puo firmare, con quale chiave, e cosa succede dopo una rotazione?
[IMPLEMENTATO] KM-1 Keyset versionato con key_id
Ogni firma dichiara il key_id usato. Le chiavi ritirate restano pubblicate: le prove storiche restano verificabili dopo la rotazione.
Evidenza: Keyset pubblico esposto ai verificatori; console admin mostra stato ed eta di ogni chiave.
[IMPLEMENTATO] KM-2 Rotazione periodica e manuale
Rotazione automatica alla scadenza configurata (default 90 giorni) e rotazione manuale amministrativa, entrambe tracciate come alert persistente.
Evidenza: ensureKeyRotation / rotateSigningKey, con evento di audit certified.key_rotated.
[IMPLEMENTATO] KM-3 Separazione ambienti staging e produzione
Prefissi di chiave distinti per PREPROD e produzione: un pacchetto di collaudo non puo apparire come produttivo.
Evidenza: Prefissi bkc-test- e bkc-cert- nel keyset pubblico.
[GAP] KM-4 Custodia in HSM / KMS gestito
La chiave privata e oggi custodita nel database sovrano dell'applicazione, non in un HSM certificato. E il gap piu rilevante dell'assessment.
Evidenza: KEY_CUSTODY riporta il custode effettivo; il valore attuale e dichiarato in console.
[PARZIALE] KM-5 Procedura di compromissione
La revoca di una chiave e tecnicamente possibile, ma la procedura formale di risposta a compromissione (chi decide, in quanto tempo, come si comunica) non e ancora approvata.
Evidenza: Da formalizzare come runbook firmato.
Rischi residui:
- Un compromesso del database di produzione consentirebbe firme arbitrarie fino alla revoca.
- Nessuna soglia multi-parte (m-di-n) per l'uso della chiave di produzione.
Perimetro per il revisore esterno:
- Percorso completo della chiave privata: generazione, riposo, uso, backup, distruzione.
- Efficacia della revoca: un pacchetto firmato con chiave revocata deve risultare NOT_VERIFIED o REVOKED.
- Segregazione dei ruoli tra chi ruota le chiavi e chi approva le release.

2.3 ISOLAMENTO TENANT - Un utente autenticato puo vedere o modificare dati di un altro?
[IMPLEMENTATO] TI-1 RLS su ogni tabella pubblica
Le tabelle applicative hanno RLS attiva e GRANT espliciti. Le server function autenticate operano con il client dell'utente: le policy si applicano come utente chiamante.
Evidenza: Policy ispezionabili; linter di piattaforma eseguito a ogni assessment.
[IMPLEMENTATO] TI-2 Superficie pubblica dichiarata e testata
Ogni server function non autenticata e elencata con motivazione ed esposizione. Un endpoint pubblico nuovo e non dichiarato fa fallire la CI.
Evidenza: src/lib/tenant-isolation.ts e src/lib/__tests__/tenant-isolation.test.ts.
[IMPLEMENTATO] TI-3 Adesione agli Spaces solo tramite invito
La policy permissiva di auto-adesione e stata rimossa: l'appartenenza si ottiene solo tramite redeem di un invito valido in funzione SECURITY DEFINER.
Evidenza: space_members non ha alcuna policy INSERT lato client.
[IMPLEMENTATO] TI-4 Coda notifiche admin-only e senza payload esposto
Tutte le operazioni sulla coda richiedono ruolo admin; la console non puo leggere il

payload degli elementi; ogni aggiornamento e scoped al singolo id e legato al run-id.
Evidenza: Test automatici sulle sei operazioni della coda e sulle colonne selezionabili.
[IMPLEMENTATO] TI-5 Uso del client service-role confinato
Il client privilegiato e importato solo dentro gli handler, mai a livello di modulo di una server function, e mai per decidere se il chiamante e admin.
Evidenza: Controllo automatico sul module scope di ogni file *.functions.ts.
[VERIFICA ESTERNA] TI-6 Test di penetrazione cross-tenant
I controlli sono statici e di policy; manca una verifica dinamica condotta da un terzo con due tenant reali.
Evidenza: Aperto: da assegnare a un revisore terzo.
Rischi residui:
- I link condivisi tramite token sono accessibili a chiunque possieda il token: la protezione e la non indovinabilita, non l'identita.
- Le funzioni SECURITY DEFINER concentrano la responsabilita: un errore in una di esse aggirerebbe la RLS.
Perimetro per il revisore esterno:
- Tentativi di lettura e scrittura cross-tenant su ogni endpoint autenticato.
- Enumerazione e forza bruta dei token di condivisione e di verifica.
- Comportamento delle funzioni SECURITY DEFINER con input ostili.

2.4 REQUISITI GIURIDICI - Cosa possiamo affermare pubblicamente, e cosa no?
[IMPLEMENTATO] LG-1 Disclaimer unico e non aggirabile
Ogni pagina pubblica, report firmato e comunicazione riporta lo stesso testo: verificabilita tecnica si, equivalenza PEC/eIDAS no.
Evidenza: LEGAL_DISCLAIMER in src/lib/trust-dossier.ts, riusato dalle pagine pubbliche.
[PARZIALE] LG-2 Motore di giurisdizione (OHADA, eIDAS)
Il sistema distingue i requisiti per giurisdizione e li mostra, ma la mappatura non e stata validata da un legale abilitato in ciascun ordinamento.
Evidenza: Motore giurisdizioni nel modulo Certified; validazione legale aperta.
[PARZIALE] LG-3 Conservazione e minimizzazione dei dati
I pacchetti conservano metadati e hash necessari alla prova; le politiche di retention per giurisdizione non sono ancora approvate formalmente.
Evidenza: Da formalizzare in una policy di conservazione approvata.
[GAP] LG-4 Qualificazione come servizio fiduciario
Nessuna qualificazione richiesta ne ottenuta. Qualsiasi affermazione di equivalenza legale sarebbe scorretta finche un organismo competente non si esprime.
Evidenza: Nessuna evidenza esiste: lo stato e dichiarato come assente.
Rischi residui:
- Rischio reputazionale se terzi interpretano 'Certified' come certificazione legale: mitigato dal disclaimer, non eliminato.
- Divergenza tra requisiti OHADA e eIDAS su marca temporale e identificazione del firmatario.
Perimetro per il revisore esterno:
- Revisione legale del wording pubblico in francese, inglese e italiano.
- Analisi di ammissibilita probatoria in almeno una giurisdizione OHADA e una UE.
- Valutazione degli obblighi di conservazione e protezione dei dati personali.

3. DOSSIER TECNICO

Algoritmo di firma..... ECDSA su curva NIST P-256 (secp256r1)
Digest..... SHA-256
Identificatore JOSE equivalente. ES256
Formato chiave pubblica..... JWK (kty EC, crv P-256), pubblicata nel keyset
Payload firmato..... Serializzazione canonica del manifest o del sigillo di catena
Implementazione..... WebCrypto SubtleCrypto, nessuna libreria crittografica di terze parti
Chiave privata..... Mai esportata verso il client, mai inclusa nei pacchetti
Concatenamento..... entry_hash = SHA-256(prev_hash contenuto evento canonico)

Keyset versionato:
- Ogni chiave ha un key_id stabile, un ambiente (staging o produzione) e uno stato (attiva

o ritirata).
  - Ogni firma dichiara il key_id: il verificatore risolve la chiave nel keyset pubblico, non indovina.
  - La rotazione crea una nuova chiave attiva e ritira la precedente, che resta pubblicata a tempo indeterminato.
  - Eta massima consigliata della chiave attiva: 90 giorni, configurabile e visibile in console.
  - Dopo una rotazione, i pacchetti storici restano VALID: cambia la chiave usata, non il verdetto.
  - Una chiave revocata per compromissione porta i pacchetti che ha firmato a NOT_VERIFIED, non a VALID silenzioso.

Procedura di verifica indipendente:
  1. Ottieni il pacchetto: Scarica audit HTML e manifest JSON firmato dalla pagina di Release Evidence o dal link pubblico.
  2. Ricalcola l'hash: Calcola SHA-256 del documento e confrontalo con il valore dichiarato nel manifest.
  3. Risolvi la chiave: Leggi il key_id nel manifest e recupera la chiave pubblica corrispondente nel keyset.
  4. Verifica la firma: Verifica la firma ECDSA P-256 sul payload canonico con la chiave pubblica risolta.
  5. Ricostruisci la catena: Ricalcola prev_hash ed entry_hash evento per evento: qualunque discontinuita invalida la sequenza.
  6. Controlla lo stato: Verifica revoca ed ambiente: il verdetto finale e VALID, REVOKED oppure NOT_VERIFIED.

4. THREAT MODEL E MAPPATURA MINACCIA -> CONTROLLO

T1 [Tampering] Alterazione di un documento gia emesso
    Mitigazione : Hash SHA-256 nel manifest firmato; qualunque modifica cambia il digest.
    Residuo   : Nessuno noto se il verificatore usa la chiave pubblica corretta.
    Controlli   : CR-1 (IMPLEMENTATO), CR-2 (IMPLEMENTATO)
T2 [Tampering] Rimozione o riordino di eventi nella catena di audit
    Mitigazione : Concatenamento per hash con sequenza: la catena si rompe in modo rilevabile.
    Residuo   : Troncamento dell'intera catena rilevabile solo confrontando con una copia esterna.
    Controlli   : CR-3 (IMPLEMENTATO)
T3 [Spoofing] Firma fraudolenta con chiave privata sottratta
    Mitigazione : Keyset versionato, rotazione periodica, revoca che porta a NOT_VERIFIED.
    Residuo   : ALTO finche la chiave non risiede in HSM: e il gap dichiarato KM-4.
    Controlli   : KM-1 (IMPLEMENTATO), KM-2 (IMPLEMENTATO), KM-4 (GAP), KM-5 (PARZIALE)
T4 [Repudiation] Retrodatazione di una prova
    Mitigazione : Timestamp server nella catena sigillata e sequenza monotona degli eventi.
    Residuo   : Senza TSA accreditata la data resta un'affermazione del sistema, non di un terzo.
    Controlli   : CR-3 (IMPLEMENTATO), LG-2 (PARZIALE)
T5 [Information disclosure] Accesso a pacchetti di altri tenant
    Mitigazione : RLS owner-scoped, verifica pubblica solo per token, controlli automatici in CI.
    Residuo   : Chi possiede il token di verifica accede al pacchetto: il token e la credenziale.
    Controlli   : TI-1 (IMPLEMENTATO), TI-2 (IMPLEMENTATO), TI-5 (IMPLEMENTATO)
T6 [Information disclosure] Enumerazione dei token di verifica
    Mitigazione : Token ad alta entropia, rate limit sulla verifica pubblica, alert su enumerazione sospetta.
    Residuo   : Rumore da scansioni automatiche: gestito come alert, non come blocco.
    Controlli   : TI-2 (IMPLEMENTATO), TI-6 (VERIFICA ESTERNA)
T7 [Denial of service] Indisponibilita di Slack o email esterna
    Mitigazione : Connettori isolati con timeout e coda con backoff: mai nel percorso critico del gate.
    Residuo   : Ritardo nelle notifiche opzionali; esito Certified invariato per

costruzione.
Controlli : TI-4 (IMPLEMENTATO)
T8 [Elevation of privilege] Un connettore esterno altera l’esito del Release Gate
Mitigazione : Il nucleo esegue e sigilla prima dei connettori; nessun risultato di connettore rientra nel report.
Residuo : Coperto da test automatici dedicati sull’isolamento dei connettori.
Controlli : TI-4 (IMPLEMENTATO), TI-2 (IMPLEMENTATO)
T9 [Elevation of privilege] Un utente autenticato assume privilegi amministrativi
Mitigazione : Ruoli in tabella dedicata con funzione SECURITY DEFINER has_role; mai ruoli nel profilo.
Residuo : Errore in una funzione SECURITY DEFINER aggirerebbe la RLS: superficie da rivedere in audit.
Controlli : TI-1 (IMPLEMENTATO), TI-3 (IMPLEMENTATO), TI-5 (IMPLEMENTATO)
T10 [Misrepresentation] Interpretazione di ’Certified’ come valore legale
Mitigazione : Disclaimer unico su ogni superficie pubblica e testi approvati per la comunicazione.
Residuo : Rischio comunicativo residuo: mitigato, non eliminato.
Controlli : LG-1 (IMPLEMENTATO), LG-4 (GAP)

5. ROTAZIONE DEL KEYSET

1. [staging] Annuncio e finestra: La rotazione e annunciata come alert persistente in console con ambiente, motivo (scadenza o compromissione sospetta) e finestra prevista. | Verifica: L’alert esiste ed e ispezionabile prima che la nuova chiave sia usata.
2. [staging] Generazione della nuova chiave di staging: Viene generata una coppia ECDSA P-256 con key_id nuovo e prefisso di ambiente staging. La chiave privata non lascia il server. | Verifica: Il keyset pubblico espone il nuovo key_id con stato attivo e ambiente staging.
3. [staging] Ritiro della chiave precedente: La chiave precedente passa allo stato ritirata ma resta pubblicata: le prove storiche restano verificabili a tempo indeterminato. | Verifica: Il key_id ritirato e ancora risolvibile nel keyset pubblico.
4. [staging] Firma di collaudo e verifica offline: Si emette un pacchetto di collaudo firmato con la nuova chiave e lo si verifica con il verificatore offline, senza rete. | Verifica: Verdetto VALID con la sola chiave pubblica nuova.
5. [produzione] Approvazione a due ruoli: La rotazione in produzione richiede approvazione Admin e Operator distinti; nessuna integrazione esterna puo sostituirsi a questa approvazione. | Verifica: Il log di audit riporta due attori distinti con timestamp.
6. [produzione] Rotazione e sigillo di audit: La rotazione emette l’evento certified.key_rotated concatenato alla catena di audit, con key_id uscente ed entrante. | Verifica: L’evento e presente nella catena e la catena resta continua (prev_hash -> entry_hash).
7. [post-rotazione] Verifica dei pacchetti storici: Si riverificano almeno tre pacchetti firmati con la chiave precedente: il verdetto deve restare VALID, perche il verificatore risolve il key_id dichiarato nel manifest. | Verifica: Nessun pacchetto storico regredisce a NOT_VERIFIED dopo la rotazione.
8. [post-rotazione] Gestione dei key_id ritirati: Un key_id ritirato non firma piu nulla, resta pubblicato per la verifica e puo essere marcato revocato solo in caso di compromissione accertata. | Verifica: Tentare una nuova firma con un key_id ritirato fallisce; la verifica storica no.
9. [post-rotazione] Comportamento su compromissione: Se la chiave e revocata per compromissione, i pacchetti che ha firmato passano a NOT_VERIFIED o REVOKED: nessun verdetto VALID silenzioso. | Verifica: Il verificatore pubblico restituisce un verdetto negativo esplicito, con motivo.
10. [post-rotazione] Pubblicazione dell’evidenza: Il keyset aggiornato e l’evento di rotazione sono pubblicati; il Trust Pack riporta la nuova numerazione documentale e il checksum aggiornato. | Verifica: Il checksum del Trust Pack cambia e la versione precedente resta scaricabile.

6. CHECKLIST PER REVISORI ESTERNI

1. Il keyset pubblico e raggiungibile senza autenticazione e contiene chiavi attive e ritirate.
2. Ogni manifest dichiara un key_id risolvibile nel keyset: nessuna chiave implicita.

3. La verifica di un pacchetto storico riesce dopo una rotazione, senza contattare il server Bonketiba.
4. La chiave privata non appare in nessun payload, log, PDF o risposta API.
5. Il gap KM-4 (custodia non HSM) e dichiarato apertamente nelle superfici pubbliche.
6. L’evento di rotazione e presente nella catena di audit e la catena resta continua.
7. Un key_id ritirato non puo firmare nuovi pacchetti.
8. Una chiave revocata porta i pacchetti a un verdetto negativo esplicito, non a VALID.
9. La rotazione in produzione richiede due approvazioni distinte e tracciate.
10. Nessun connettore esterno (Slack, email) puo alterare l’esito del Release Gate.

7. RISORSE PUBBLICHE

Assessment	https://navigator.bonketiba.com/security-assessment
Dossier tecnico	https://navigator.bonketiba.com/certified-dossier
Posizionamento	https://navigator.bonketiba.com/certified-positioning
Report CI	https://navigator.bonketiba.com/assurance
Rotazione keyset	https://navigator.bonketiba.com/keyset-rotation
Modulo auditor	https://navigator.bonketiba.com/auditor-access
Threat model JSON	https://navigator.bonketiba.com/api/public/threat-model
Keyset pubblico	https://navigator.bonketiba.com/api/public/certified-keys
Verificatore	https://navigator.bonketiba.com/release-evidence/verify

8. INTEGRITA’ DEL DOCUMENTO

doc_id	BKC-TRUST-1.0.0
sha256	0062171cc29e9a22d4bd3dc8f0ba619cf0764872a1c4bb470fe629d4b69c39c1
key_id	bkc-test-20260819-59c490
algoritmo	ECDSA_P256_SHA256
firma	ecdsa-p256-sha256:76b7bb6a574ff855980eea31bef2669c481add8514e0522ad013c6e6137fa1d56623b6c54bbc3b850330cb529cde9d4c83f4aad01ca03f4d4335629daa8fefc6

Verifica: ricalcola SHA-256 della forma canonica del testo, risolvi il key_id nel keyset pubblico e verifica la firma ECDSA P-256. Nessun accesso privilegiato richiesto.