BONAFIDE™ SPECIFICATION V1.0 — PART 12

Institutional Runtime & Data Governance

Runtime architecture, three-path key separation, third-party access framework (user sharing, institutional verification, legal access, emergency access), revocation, anti-duplication, and hardware abstraction layer

1. Purpose

This part specifies how institutions integrate with Bonafide at the infrastructure level: the runtime they deploy, the APIs their applications call, the key architecture that enables both institutional utility and user sovereignty, and the mechanisms for revocation, third-party access, and anti-duplication.

Third-party access—whether a friend viewing shared photos, a doctor receiving transferred medical records, a merchant verifying identity, or a court executing a warrant—is a first-class protocol operation with its own key path, authorization model, and lifecycle. Bonafide is a data sovereignty framework, not a payment system or messaging platform. Third-party access is scoped to data sharing, verification, legal process, and emergency scenarios.

2. Bonafide Runtime Architecture

2.1 What the Runtime Is

The Bonafide runtime is a library or service that institutions deploy on their own infrastructure. It is the sole interface between the institution's application code and the user's encrypted vault data. The runtime handles key derivation, encryption and decryption, access policy enforcement, audit logging, revocation processing, ghost quanta generation, and communication with the Bonafide network (validators, federation, relay).

From the institution's developer perspective, the runtime exposes a straightforward API: read a quantum, write a quantum, query a branch, authorize an operation. The underlying cryptography is invisible.

2.2 Deployment Models

Model	Description	Classification	Use Case
Embedded library	Bonafide runtime linked into the institution's application process	B or C	Simplest integration, software-only
Sidecar service	Separate process, communicating via local IPC	B	Better isolation, standard containers/VMs
TEE-hosted service	Runtime inside a hardware TEE, attested channel to application	A	Hardware-attested isolation
Enclave-hosted service	Runtime inside FPGA fabric or dedicated secure processor	S	Maximum isolation, ExaForge or equivalent
Cloud-managed	Bonafide Cloud operates the runtime as a managed service	A or B	No on-premises runtime needed

The runtime's API is identical across all models. An institution can start with the embedded library (Classification C) and migrate to enclave-hosted (Classification S) without changing application code.

2.3 Database Integration

The Bonafide database packages (PostgreSQL, Oracle, SQL Server, MySQL, MongoDB) provide transparent encryption beneath the institution's existing data model. SQL queries work normally. The Bonafide layer intercepts reads and writes, encrypts and decrypts through the runtime, and manages key material and audit logging. Institutions can adopt incrementally: encrypt one table, one column, one data category at a time.

3. Key Architecture

3.1 Three Key Paths

Bonafide defines three distinct key paths, each with a strictly bounded purpose. No key path can be used for another's purpose. This separation is enforced cryptographically.

Key Path	Holder	Purpose	Scope
User key path	User (Bio Root → branch key → DEK)	Sovereign access from any enrolled device	Full vault across all institutions
Institutional key path	Institution (master key → branch key → DEK)	Operational access for authorized services only	Only the institution's own branch
Third-party key path	Any authorized party (purpose-specific generated key)	Scoped access for sharing, verification, legal, or emergency	Only the scope specified in the authorization

3.2 Institutional Key — Sacrosanct

The institutional key is licensed exclusively for user-authorized institutional operations. No exceptions. The institution uses its key to process transactions, generate statements, run diagnostics, and deliver the services the user authorized during peering. The institution's key cannot be used to facilitate third-party access of any kind—not for law enforcement, not for regulators, not for partner institutions, not for the user's friends. Every third-party access scenario uses the third-party key path.

This is an architectural property. On Classification A and S deployments, the key exists only in hardware-isolated memory. The runtime enforces the scope. This protects institutions from coercion: the protocol does not allow institutional keys to serve third-party access. Any requesting party must use the third-party access protocol.

3.3 User Key — Sovereign

The user key path is derived from the Bio Root (Level 0) and provides the user with independent access to their data without institutional cooperation. The user key is never shared with any institution, any third party, or the Bonafide network.

3.4 DEK Wrapping

Every quantum's DEK is wrapped independently for each authorized key path:

- User wrapping: always present.
- Institutional wrapping: present while the peering relationship is active.
- Third-party wrapping(s): present only for active, validated authorizations. Multiple third-party wrappings can coexist for different authorized parties.

Each wrapping is independent. Revoking one does not affect the others. Adding a new wrapping does not require any other key holder's participation.

4. Third-Party Access Framework

4.1 Unified Model

Every third-party access scenario uses the same underlying mechanism: a purpose-specific key is generated by the Bonafide API, scoped to specific data, bound to a specific recipient, and limited to a specific duration. What varies across access types is the authorization source, the key lifetime, and the default data exposure.

4.2 Access Categories

Category	Authorization Source	Key Lifetime	Data Exposure	User Action
User sharing	User's explicit vault action	Minutes to permanent (user-configured)	User's choice (plaintext or ghost)	Explicit share action
Institutional verification	User pre-authorization rule	Seconds to minutes	Ghost quanta only (default)	None (pre-authorized) or real-time tap
Legal / audit	Validated legal instrument	Days to months	Per instrument specification	None (protocol-mediated)
Emergency	Pre-configured emergency profile	Hours	Medical/identity quanta only	None (pre-configured)

4.3 Key Properties (All Categories)

Regardless of category, every third-party key shares these properties:

- **Scope-bound:** The key decrypts only the authorized quanta. Attempting to use it on out-of-scope data produces nothing.
- **Time-bound:** The key incorporates a time component. After expiration, the key is non-functional. Expired wrappings are automatically removed from affected quanta.
- **Recipient-bound:** The key is bound to the recipient's certificate or bearer token. It cannot be transferred to another party.
- **Instrument-bound:** The key is derived from the authorization source's hash. A different authorization produces a different key, even for identical scope and recipient.
- **Audited:** Key generation, every access event, and key expiration are recorded in the immutable ledger.
- **Revocable:** The authorizing party (user for sharing, court for legal) can revoke the key before expiration.

5. User Sharing

5.1 Use Cases

- Share photos or a photo album with a friend or family member.
- Send medical records to a new doctor or specialist.
- Provide proof of address or identity documents to a landlord.
- Share a private article, document, or creative work with a colleague.
- Grant a family member ongoing access to specific vault branches for estate planning.
- Provide tax documents to an accountant for annual filing.

5.2 Flow

- User selects quanta, channels, or branches to share from their vault interface.
- User specifies the recipient: by Bonafide address (if the recipient has a vault), or by generating a share link (if they don't).
- User configures access parameters: duration, read-only or read-write, plaintext or ghost, number of views allowed.
- The Bonafide API generates a share key scoped to the selected data and bound to the recipient.
- DEK wrappings are created for the share key on each selected quantum.
- The recipient accesses the shared data through the Bonafide viewer (app, web, or link).

5.3 Share Modes

Mode	Key Behavior	Use Case
View-once	Key valid for a single access session, then invalidated and wrappings removed	Sensitive documents, ID verification, one-time record transfer
View-limited	Key valid for N access sessions (user-configured), then invalidated	Shared album the recipient can revisit a few times
Time-bounded	Key expires after configured duration (1 hour to 30 days), wrappings removed	Temporary access for a project, tax season, medical referral
Revocable	Key active until user explicitly revokes from any vault holder device	Ongoing sharing where user wants manual control
Permanent	Key does not expire but is still revocable	Family access to shared branches, estate planning, long-term collaboration

5.4 Share Links

For recipients who are not on the Bonafide network, the user generates a share link. The link opens the Bonafide viewer in a web browser. The recipient does not need a vault, biometric enrollment, or any Bonafide account to view shared data—they need only the link.

Share link properties:

- **Rendered, not downloaded:** Shared data is rendered in the Bonafide viewer. The viewer does not offer a download button, does not allow copy-paste of content, and applies forensic watermarking to the rendered display. This is not DRM—a determined recipient can always screenshot—but it prevents casual copying and ensures traceability.
- **Password-protected (optional):** The user can set a password on the share link, adding a knowledge factor. The recipient needs both the link and the password.
- **Combinable with share modes:** A share link can be view-once, view-limited, time-bounded, or any combination.
- **Onramp for new users:** After viewing shared data, the recipient is offered the option to create their own vault. User sharing is a natural acquisition channel for the Bonafide ecosystem.

5.5 Shared Data Rendering

All shared data is rendered through the Bonafide viewer, whether the recipient accesses it through the app (vault-to-vault sharing) or through a web link. The viewer enforces:

- Forensic watermarking on all rendered content (session-unique, invisible, traceable to the share key).
- No raw data export unless the user explicitly enabled download in the share configuration.
- Screen capture detection where the platform supports it (iOS, Android) with optional notification to the sharing user.
- Expiration enforcement: the viewer checks the share key’s validity before every render.

6. Institutional Verification

6.1 Use Cases

- Merchant needs the user's bank to confirm the user's identity or account standing.
- Landlord needs the user's employer to confirm income range.
- New doctor needs old doctor to transfer specific medical records.
- Insurance company needs hospital to confirm a diagnosis code.
- University needs to verify a transcript with another institution.
- Government agency needs employer to verify employment status for benefits eligibility.

6.2 Verification Rules

Users configure verification rules in their vault that pre-authorize specific institution-to-institution verifications:

- "Any institution I peer with may verify my identity with my bank using ghost quanta."
- "My insurance company may request diagnosis codes from my hospital using ghost quanta."
- "No institution may verify my employment status without my real-time approval."
- "My new doctor may request full medical records (plaintext) from my old doctor with my real-time approval."

Verification rules specify: the requesting institution or category, the holding institution, the data scope, whether ghost or plaintext, and whether pre-approved or requiring real-time user approval.

6.3 Verification Flow

Step 1: The requesting institution (Merchant A) initiates a verification request through the Bonafide API, specifying the data type needed and the holding institution (Bank B).

Step 2: The API checks the user's verification rules for this request pattern.

Step 3a (pre-approved): The API generates a short-lived verification key (default: 60-second lifetime). The Bonafide runtime on Bank B's infrastructure generates ghost quanta from the target data and wraps them with the verification key. The merchant receives the ghost response. The key expires immediately after delivery.

Step 3b (real-time approval): The user receives a notification on their vault holder device describing the request. One biometric tap approves. The flow continues as 3a. If the user does not approve within a configurable window (default: 5 minutes), the request expires.

Step 4: The verification event is recorded in the ledger. The user can see which institutions requested verification, when, what data was provided (in ghost or plaintext form), and which verification rule authorized it.

6.4 Ghost-Only Default

Institutional verification uses ghost quanta exclusively by default. The merchant receives a cryptographic proof that the user's account exists and is in good standing—not the account number. The landlord receives a proof that the user's income falls within a range—not the exact

figure. The insurance company receives a diagnosis code confirmation—not the full medical record.

Users can override the ghost-only default for specific verification rules if they want to share plaintext (e.g., full medical record transfer to a new doctor). This is the user's choice, configured in the verification rule, not the requesting institution's.

6.5 Institutional Separation

The holding institution (Bank B) uses its own institutional key for its normal purpose: accessing data in its branch to generate the ghost quanta or plaintext response. Bank B's key is never shared with Merchant A. Merchant A receives only the ghost quanta or plaintext wrapped with the short-lived verification key. The two institutions' keys never meet. The Bonafide network mediates the entire exchange.

7. Legal and Audit Access

7.1 Design Philosophy

Legal and audit access is the only third-party access category that does not require the user's authorization. It is mediated by the Bonafide API on the authority of a validated legal instrument. The institution is not involved in key generation or access mediation. The protocol handles it.

7.2 The Access Flow

Step 1 — Instrument submission: The authority presents a legal instrument (court order, warrant, regulatory audit authorization, subpoena) to the Bonafide API specifying the target, scope, access type (snapshot or monitoring), duration, and legal basis.

Step 2 — Validation: The instrument is hashed and submitted to the blind validation network. Validators verify the issuing authority, scope, uniqueness (no replay), jurisdiction, and cryptographic signature.

Step 3 — Key generation: A purpose-specific key is derived from the instrument's hash, scope, duration, and authority's certificate. New DEK wrappings are created for quanta within the authorized scope.

Step 4 — Key delivery: The key is delivered through an authenticated channel. It works only for the authorized scope and duration.

Step 5 — Ledger recording: The operation is recorded in the immutable ledger, visible to the data owner (subject to court-ordered non-disclosure deferral with mandatory eventual disclosure).

7.3 Snapshot vs. Monitoring

Snapshot (audit/forensic): A point-in-time copy of authorized data, encrypted with the third-party key. Immutable after capture. Changes to live data do not affect the snapshot. Snapshot carries its own Merkle root for integrity verification. Expires when the key expires.

Monitoring (surveillance order): Ongoing read access to specified data streams. New quanta are automatically wrapped with the third-party key. Hard expiration—extension requires a new instrument and new key.

7.4 Multiple Concurrent Authorizations

Multiple legal authorizations can coexist. Each authority receives a different key. Neither can see the other's access. Revoking one does not affect the others.

7.5 Export as Encrypted Vault

Data can be exported as a self-contained vault file. The legal instrument can specify constraints:

- **Minimum classification:** Receiving system must meet a privacy classification floor.
- **Enclave-only:** Decryption requires hardware enclave, preventing plaintext on general-purpose systems.
- **Time-to-live:** Exported vault self-destructs after TTL (key expiration prevents decryption).

- **Ghost-only export:** Export contains only ghost quanta—verifiable proofs, not raw data.

7.6 Instrument Validation Requirements

Check	Purpose	Failure Action
Authority registration	Requesting authority is registered for the jurisdiction	Reject
Instrument hash uniqueness	Prevent replay	Reject
Scope verification	Instrument specifies a valid, bounded scope	Reject
Digital signature	Instrument signature matches authority's published key	Reject
Jurisdiction check	Authority has jurisdiction over target data's residency	Reject
Duration reasonableness	Flag unusually long durations	Hold for manual review

8. Emergency Access

8.1 Use Cases

- Paramedic needs medical history for an unconscious patient.
- Emergency contact needs to reach someone after an accident.
- First responder needs to identify an unresponsive person.
- Designated agent needs vault access when user is incapacitated (legal, financial, medical decisions).

8.2 Emergency Profile

Users configure an optional emergency profile in their vault. Users who do not configure one have no emergency access path. The emergency profile specifies:

- **Emergency data scope:** Which quanta are accessible in an emergency. Typically: medical conditions, allergies, medications, blood type, emergency contacts, organ donor status, identity information. Never: financial data, personal communications, or other branches unless explicitly included.
- **Emergency recipients:** Who can trigger emergency access. Options include: any certified first responder (verified through institutional certification), specific named contacts (family members, attorney, physician), or a designated emergency agent.
- **Activation method:** How emergency access is triggered. Options: the user's vault device broadcasts an emergency beacon (medical alert); a certified first responder requests access through the Bonafide API with their institutional credential; a designated agent requests access with their personal vault credential.
- **Duration:** Emergency access keys default to 24-hour lifetime, renewable once for an additional 24 hours. Longer access requires the user to regain consciousness and authorize, or a legal instrument (power of attorney, guardianship order) to transition to legal access.

8.3 Emergency Access Flow

- Emergency is triggered (device beacon, first responder request, or designated agent request).
- The Bonafide API verifies the requester's credential against the user's emergency profile.
- A short-lived emergency key is generated, scoped to the emergency data profile.
- The requester receives the emergency data through the Bonafide viewer or API.
- The access is recorded in the ledger. The user is notified when they next access their vault.
- If the user does not access their vault within 48 hours, designated emergency contacts are notified that emergency access was triggered.

8.4 Emergency and Duress Interaction

Emergency access respects the persona architecture. If the user is authenticated into a duress persona when the emergency occurs, emergency access serves data from the duress

persona’s emergency profile—which may contain real medical data (allergies and medications are life-critical) but synthetic identity information. This is a configurable tradeoff that the user makes during emergency profile setup.

9. Institutional Key Lifecycle

9.1 Key Creation

The institutional master key is generated during peering within the highest-available enclave (FPGA PUF, HSM, TEE, or software keyring depending on Classification). The key is derived using HKDF from the peering handshake material and bound to the institution's certificate.

9.2 Key Use

The runtime derives branch-specific keys from the institutional master key and uses them to unwrap DEKs for authorized operations. The master key never leaves the trust anchor. Branch keys exist in runtime memory only during active sessions. DEKs exist in memory only for the duration of a single quantum operation.

9.3 Key Rotation

Rotation generates a new master key, re-wraps the institutional DEK wrappings, and invalidates the old master key. User key wrappings and active third-party wrappings are unaffected.

9.4 Key Revocation (User-Initiated)

- The institutional key wrapping is removed from all DEKs in the revoked scope.
- The institutional master key (or branch derivative) is invalidated in the runtime.
- Affected quanta are re-encrypted with new DEKs wrapped only with the user key path and any active third-party key paths (a legal authorization survives institutional revocation if the instrument is still valid).
- The revocation event is recorded in the immutable ledger.

9.5 Revocation Strength by Classification

Classification	Revocation Strength	Residual Risk
S — Sovereign	Absolute. No plaintext residue.	None (within enclave threat model)
A — Attested	Effective. TEE state purged.	Prior plaintext API responses may exist in application caches
B — Bounded	Operational. On-disk state changed.	Prior plaintext. Old DEKs potentially recoverable from memory
C — Compliant	Partial. Bonafide layer re-encrypted.	Institution may retain plaintext copies. Canary/watermark traceability.

10. Anti-Duplication Framework

Bonafide cannot cryptographically prevent copying after authorized access. The anti-duplication framework minimizes exposure, maximizes traceability, and makes duplication economically irrational.

10.1 Layer 1 — Minimize Exposure (Ghost Quanta)

Most interactions do not require raw data. Ghost quanta provide institutional utility without exposing plaintext: zero-knowledge proofs, redacted summaries, hashed confirmations, range proofs, and set membership proofs. Classification S uses ghost quanta for all operations by default.

10.2 Layer 2 — Traceability (Watermarking and Ledger)

Every plaintext release carries a session-unique forensic watermark. Leaks are traceable to source institution, session, timestamp, and employee. The immutable ledger corroborates.

10.3 Layer 3 — Detection (Canary Quanta)

Synthetic data unique to each peer, indistinguishable from real data. Canary appearance outside authorized context identifies the source with certainty.

10.4 Layer 4 — Economic Deterrence (Privacy Scoring)

Abnormal access patterns, unnecessary plaintext usage, and post-revocation data retention degrade privacy scores. Users choose competitors. Regulators increase scrutiny.

10.5 Layer 5 — Technical Enforcement (API Abstraction)

The Bonafide runtime API is the sole data interface. Institutions never handle DEKs or perform raw decryption. The easy path is the secure path.

10.6 Layer 6 — Contractual (Peering Agreements)

Machine-readable data handling terms in peering agreements: permitted use, retention duration, prohibition on secondary storage. Encoded in quantum access policies and auditable through the ledger.

11. Hardware Abstraction Layer

The Bonafide runtime abstracts underlying hardware through a HAL with pluggable backends. The same runtime binary operates on any supported platform. The backend determines the privacy classification.

11.1 HAL Operations

- **Key storage:** FPGA PUF, HSM, TPM, TEE secure storage, software keyring.
- **Key derivation:** FPGA fabric, HSM crypto engine, TEE execution, software.
- **Symmetric encryption:** FPGA fabric, GPU (CUDA), QAT accelerator, SmartNIC/DPU, TEE, software.
- **Merkle operations:** GPU, FPGA, software.
- **Attestation:** FPGA, TEE (SGX, TDX, SEV-SNP, Nitro, TrustZone), TPM, none.

11.2 Backend Mapping

Hardware	Key Storage	Bulk AES	Attestation	Classification
ExaForge FPGA	PUF	FPGA fabric	FPGA attestation	S
NVIDIA H100 CC	TEE	GPU (confidential)	TEE attestation	A
Intel SGX/TDX	TEE	TEE	SGX/TDX attestation	A
AMD SEV-SNP	TEE	TEE	SEV attestation	A
AWS Nitro Enclave	Enclave	Enclave	Nitro attestation	A
NVIDIA BlueField DPU	DPU cores	DPU crypto engine	DPU attestation	A
Intel QAT + HSM	HSM	QAT accelerator	None	B
TPM + GPU	TPM	GPU	TPM attestation	B
Software only	Encrypted keyring	Software	None	C

11.3 Trust Anchor and Acceleration Separation

The HAL separates the trust anchor (key secrecy) from the acceleration layer (throughput). Classification is determined by the weakest link. The HAL auto-detects available hardware and selects the highest-classification backend available.

V1.0 — February 2026