

# White Paper - Retyc

Encrypted File Transfer and Datarooms: Architecture, Security, and Independence

TripleStack SAS / Version 1.0 / April 2026

---

*This document is intended for technical decision-makers, information systems directors, chief information security officers (CISOs), data protection officers (DPOs), and compliance leaders within organizations operating in the European Union.*

---

**Original version takes precedence** — *This white paper was originally written in French. Although translations may be made available for the convenience of international readers, only the original French version is authoritative and shall prevail in the event of any difference in interpretation, contradiction, or technical nuance.*

---

## Contents

---

1. Executive summary
  2. The challenge: protecting sensitive data in a demanding regulatory environment
  3. Technical architecture and security model
  4. End-to-end encryption with mechanisms resistant to quantum threats
  5. Zero-knowledge architecture: confidentiality by design
  6. European technological independence
  7. Features: file transfer and datarooms
  8. Operational transparency
  9. Use cases and target sectors
  10. Plans and deployment
  11. Conclusion
-

## 1. Executive summary

---

Retyc is a European platform for secure file transfer and collaborative datarooms, developed and operated by TripleStack SAS, an independent French company. It is built on three inseparable pillars: **end-to-end encryption with mechanisms resistant to quantum threats, a zero-knowledge architecture, and European digital independence.**

In a context shaped by intensifying cyber threats, growing geopolitical tensions, the tightening of the European regulatory framework (GDPR, NIS2, DORA), and increasing awareness of risks linked to U.S. extraterritorial legislation (Cloud Act, FISA), organizations need solutions that do not rely solely on contractual or regulatory assurances, but on cryptographic guarantees\*\*.

By design, Retyc limits any access to your content in plaintext, including by its own teams. Files are encrypted locally on the user's device before being sent to our servers. Only authorized recipients hold the decryption keys.

This white paper presents Retyc's technical architecture, security model, regulatory compliance implications, and use cases for demanding organizations.

**Terminology** — *In this document, the term "password" refers, depending on the context, either to an authentication secret (login) or to a secret input used for client-side encryption (technical equivalent of a passphrase).*

## 2. The challenge: protecting sensitive data in a demanding regulatory environment

---

### 2.1 A strengthening European regulatory framework

The General Data Protection Regulation (GDPR, 2016/679) has imposed strict obligations on the processing of personal data since 2018. Compliance is no longer optional: penalties can reach 4% of annual global turnover.

Beyond the GDPR, several sector-specific regulations reinforce security requirements, including:

- **NIS2** (Directive on security of network and information systems, 2022): broadens cybersecurity obligations to many sectors
- **DORA** (Digital Operational Resilience Act, 2022): imposes digital resilience requirements on the financial sector
- **Professional secrecy** (lawyers, chartered accountants, doctors): ethical obligations reinforced by disciplinary and legal sanctions in case of breach.

### 2.2 The real threat facing file transfers

File transfer between organizations is one of the most exposed vectors in the security chain. Traditional solutions such as consumer cloud sharing, email attachments, or FTP servers present documented vulnerabilities:

- **Data exposed on the provider's servers:** consumer sharing platforms encrypt data in transit (HTTPS / TLS) and sometimes at rest on their servers, but they **retain the decryption keys**. Unauthorized access to the servers, a judicial request, or a data breach can fully expose your content.
- **Exposure to extraterritorial legislation:** providers operated by U.S. companies (even when data centers are physically located within the European Union) are subject to the *Cloud Act* (2018) and the *Foreign Intelligence Surveillance Act (FISA)*, which authorize U.S. authorities to access data stored by those companies, including data belonging to European clients, without necessarily informing the data subjects.
- **Lack of traceability:** informal solutions (email, instant messaging) do not provide any audit trail of access or any revocation of rights after sending (even if the file is deleted from the server, local copies remain accessible).
- **Future threats to current data:** the so-called *harvest now, decrypt later* strategy consists of capturing encrypted data today in anticipation of future hardware advances (notably quantum computers) that could make it possible to decrypt it in a few years. Sensitive data transmitted today with conventional encryption could be compromised tomorrow.

## 2.3 The limits of contractual approaches

Faced with these issues, many providers emphasize organizational and contractual guarantees: confidentiality clauses, security commitments, certifications, audits, or internal policies. These elements are useful, but **they are not sufficient on their own to guarantee the effective confidentiality of data.**

In practice, purely contractual protection does not change the provider's actual technical capabilities. If data is accessible in plaintext on its servers, it remains exposed in the event of a technical compromise, human error, internal abuse, or an access request from a competent authority.

In other words, a contractual commitment may govern use. It does not remove the technical possibility of accessing the data.

**The most robust protection therefore relies on the cryptographic architecture itself:** when content is encrypted client-side and the decryption keys are not accessible to the provider, confidentiality no longer depends solely on a promise, an internal procedure, or a contractual framework. It rests on a verifiable technical constraint.

This is the logic underlying Retyc's architecture.

*Some services reserve the right to change their terms of use, including on sensitive data-related matters. This dependence on an evolving contractual framework introduces **uncertainty about the real level of confidentiality** over time.*

*A cryptographic constraint does not depend on an internal policy. Retyc does not rely on a contractual promise, but on an architecture designed to make data inaccessible in plaintext to the provider.*

## 3. Technical architecture and security model

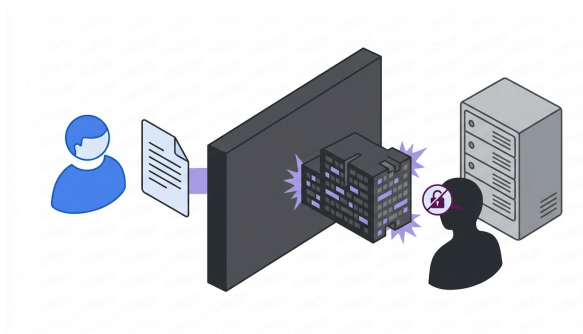
---

**Note** — *Two types of passwords are used in Retyc for each registered user: a login password (authentication) and an encryption password (protection of the private key). Retyc strongly discourages using the same password for both purposes.*

### 3.1 Core principle

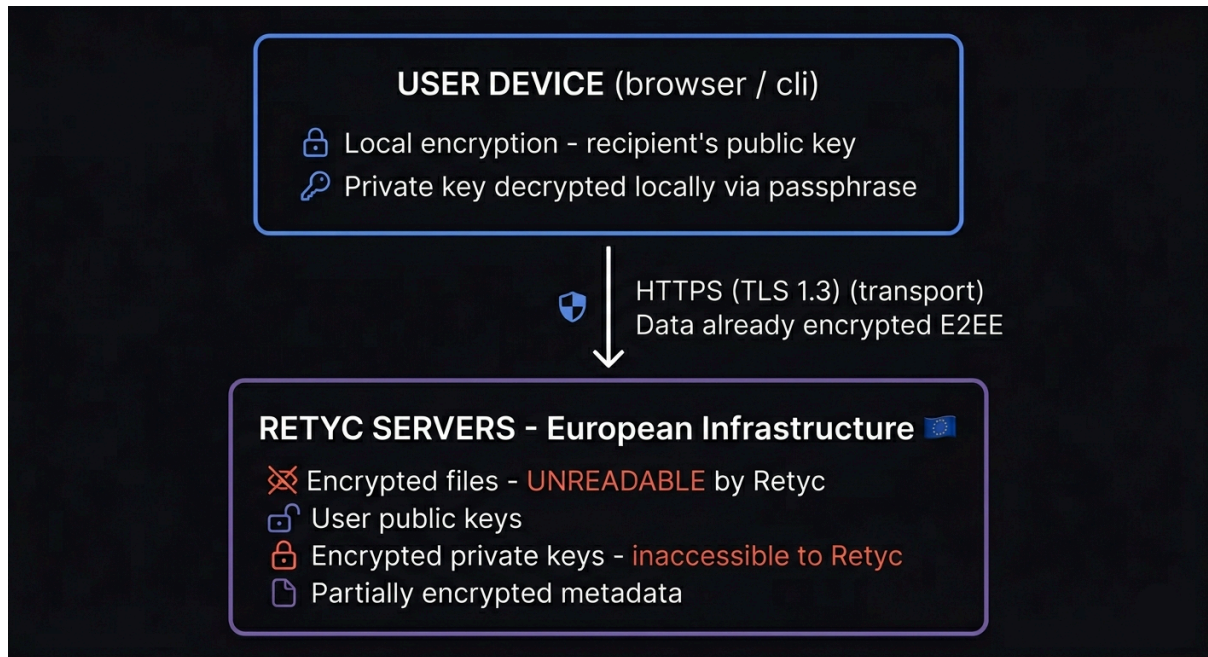
Retyc's architecture is designed to prevent any operator access to content in plaintext, including by the service provider itself.

This is not a matter of trusting our teams or the robustness of our internal policies. It is a cryptographic constraint imposed by our architecture: the decryption keys (in plaintext) for your content never pass through our servers and are never accessible to our infrastructure.



### 3.2 Architecture overview

Retyc's architecture combines several complementary layers:



When a user sends a file via Retyc:

1. The file is **encrypted locally in the browser** with the public key of the authorized recipient or recipients. If there is no registered recipient, an ephemeral key is generated for the transfer and encrypted with a password chosen by the sender.
2. The encrypted file is sent to our servers: we only see unreadable data.
3. The recipient retrieves the encrypted file and decrypts it **locally in their browser** with their private key.
4. Each user's private key is stored on our servers in **encrypted form with the user's encryption password**, which we do not know.

### 3.3 What Retyc can and cannot do

It is essential to be explicit about the technical limits of our access:

Data	Scope	Accessible to Retyc?
Email, first name, last name	Global	<b>Yes</b> — required for operation
File content (plaintext)	Retyc	<b>No</b> — encrypted before upload
File metadata (name, size, type)	Retyc	<b>Partially</b> — file size is not encrypted (necessary for storage management), but other metadata (type, name) is encrypted.
Special case: file names in a dataroom	Retyc	<b>No</b> — the name is encrypted; a <a href="#">sha256</a> hash with a 256-bit random salt (CSPRNG), itself encrypted E2EE with the session key, is used for version management. Since the salt is inaccessible to the server in plaintext, the server cannot infer file names.
Transfer messages	Retyc	<b>No</b> — E2EE encrypted
Activity log and messaging (dataroom)	Retyc	<b>Partially</b> — message content and sensitive metadata are encrypted. Information required for operation (e.g. users, event types) remains visible.
Transfer / dataroom title	Retyc	<b>Yes</b> — not E2EE encrypted
A user's private key	Retyc	<b>No</b> — stored only in encrypted form
Login password	IAM	<b>No</b> — hashed server-side with argon2 (version <a href="#">1.3</a> , <a href="#">type=id</a> , <a href="#">iterations=5</a> , <a href="#">parallelism=1</a> , <a href="#">memory=19 MiB</a> ). <b>If authentication is performed via an external provider, no password is stored.</b>
Recovery of the encryption password / private key	Retyc	<b>No</b> — if the encryption password is lost, access to encrypted data is permanently lost.
Email address (notifications)	Global	<b>Yes</b> — used to send all email notifications (transfer received, dataroom invitation, verification code, etc.). The email delivery provider does not receive files or encrypted content.

This transparency about our technical limits is a core part of our commitment to users.

*The "IAM" scope (Identity and Access Management) covers authentication and user management data (handled by the Keycloak v26.x application). The "Retyc" scope covers data related to transfers, datarooms, and files. The "Global" scope covers data present in both systems.*

### 3.4 Assumptions and limits of the security model

Retyc significantly reduces several risks related to the transfer and storage of sensitive files. However, it is important to acknowledge the limits inherent to any security solution:

Threat / Risk	Scope	Risk reduction by Retyc	Limits
User device compromise	Global	-	If the user's device is compromised by malware or physical access, locally encrypted data may be exposed. Retyc recommends strong client-side security practices (antivirus, MFA, access management).
Automated attacks (scans, brute force, etc.)	Global	<b>High</b>	Retyc implements application protection mechanisms (WAF) and intrusion detection, including solutions such as CrowdSec, to identify and block malicious behavior. Targeted or sophisticated attacks may still bypass these mechanisms.
Brute-force attack against the authentication system	IAM	<b>High</b>	Keycloak implements protections against brute-force attacks (rate limiting, account lockout).
Brute-force attack against the password protecting a private key	Retyc	<b>Partial</b>	The private key is encrypted with scrypt ( $N=2^{18}$ ), making each attempt very costly in time and resources. Actual resistance still depends on the strength of the password chosen by the user: a weak password significantly reduces this protection despite the derivation cost.
Attack against cryptographic algorithms	Retyc	<b>High</b>	Retyc uses modern and robust algorithms against known attacks, with a hybrid scheme integrating mechanisms resistant to quantum threats according to current knowledge.
Configuration or implementation error	Global	<b>Partial</b>	Although we follow secure development best practices, human error in configuration or implementation could introduce a vulnerability. We have rigorous code review and security testing processes in place to reduce this risk.
Insider threat (privilege abuse)	Global	<b>High</b>	Due to the zero-knowledge architecture, even a malicious employee or privilege abuse does not provide the technical means to access content in plaintext.
Judicial request	Global	<b>High</b>	In the event of an access request from a competent authority, Retyc and its file hosting provider can only provide unreadable encrypted data and connection logs. We cannot decrypt what we have no means to decrypt.
Data breach on the provider's servers	Global	<b>High</b>	In the event of a data breach on our servers, encrypted content remains unreadable. We do not store plaintext data that could be exposed. However, this does not provide access to plaintext content.
Breach of data stored in the database (email, first name, last name)	Global	<b>Partial</b>	In the event of a personal data breach, user identity information could be exposed.
Theft of session cookies or other	Global	<b>Partial</b>	In the event of session cookie theft, an attacker could potentially access a user account. We recommend using MFA to reduce this risk and not using Retyc on shared or unsecured devices. To limit the

Threat / Risk	Scope	Risk reduction by Retyc	Limits
authentication data			risk, session validity is time-limited and inactive sessions are automatically logged out.
Malicious script injection attack (XSS) on the user interface	Retyc (web only)	<b>High</b>	We apply strict security policies (CSP, HSTS) to reduce the risk of XSS attacks.
Theft of the transfer password	Retyc	<b>Low</b>	If an attacker obtains a transfer password, they can access the content as long as the transfer is active. The sender can disable the transfer at any time to revoke access. For sensitive data, we recommend using registered recipients (asymmetric keys).
Revoked user who already downloaded files before revocation	Retyc	<b>Low</b>	If a revoked user already downloaded the files before revocation, they may keep a local copy. Revocation prevents future access but cannot erase copies already downloaded. We recommend limiting link validity periods to reduce this risk.
Forgotten private key password	Retyc	-	In the absence of a "Master Key," forgetting the password by the sole holder of access results in data loss. However, multiple authorized members provide access redundancy.
Source code compromise	Retyc	<b>Partial</b>	In the event of source code compromise, an attacker could theoretically introduce a vulnerability or backdoor. However, the zero-knowledge architecture limits the risk of access to encrypted data. We follow strict code review and security practices to minimize this risk.
Compromise of a user's private key (exfiltration or targeted attack)	Retyc	<b>Partial</b>	In the event of compromise of a user's private key, data encrypted for that user could be exposed. However, other users and their data remain protected. In case of suspicion or compromise, key rotation and changing the encryption password help limit the impact.
Compromise of hosting infrastructure (Scaleway, Clever Cloud)	Hosting	<b>Partial</b>	In the event of hosting infrastructure compromise, encrypted data remains unreadable. However, such a compromise could lead to temporary service unavailability and possible exposure of unencrypted data.
Use of a revoked key to decrypt past data	Retyc	<b>Low</b>	During key rotation, the old encrypted private key is deleted server-side and the session key is re-encrypted for the relevant spaces. Residual risk is limited to the case where the user kept a local copy of their private key outside Retyc. Rotating a user's key only re-encrypts the session key for transfers they created and datarooms where they are an administrator. In other cases, rekeying must be triggered by a space administrator.

*Note — The security of mechanisms relying on a password depends on the strength of the chosen secret. We recommend using strong, unique passwords.*

### 3.5 Mechanisms minimizing residual risks

Mitigation mechanism	Description	Limits
Optional strong authentication (MFA)	We recommend and support two-factor authentication to strengthen the security of user accounts.	Does not protect against attacks directly targeting the user's device or client-side configuration errors.
Strict HSTS, CSP, CORS, and permissions policy	We apply strict HTTP security policies to protect against man-in-the-middle attacks and script injection.	Does not protect against attacks directly targeting the user's device.
WAF and anomaly monitoring	We use a web application firewall (WAF) and intrusion detection systems to monitor suspicious activity on our servers.	Does not protect against attacks directly targeting the user's device or client-side configuration errors.
Rate limiting and brute-force protection	We implement rate-limiting mechanisms to reduce the impact of automated abuse and certain denial-of-service attacks.	Hosting may become temporarily unavailable during a denial-of-service attack, but data remains protected.
Regular updates and vulnerability monitoring	We actively track security vulnerabilities and apply regular updates to fix potential flaws.	Does not protect against attacks directly targeting the user's device or client-side configuration errors.
No third-party tracking scripts and minimal collection of personal data	We do not use any third-party cookies that could be exploited for advertising tracking or behavioral analytics, and we limit personal data collection to what is strictly necessary.	Does not protect against attacks directly targeting the user's device or client-side configuration errors.
Authentication with Keycloak and role-based access management (RBAC)	We use Keycloak to manage authentication by limiting session duration and applying granular role-based access control.	Does not protect against attacks directly targeting the user's device or client-side configuration errors.
Encryption key rotation	We provide the ability to rotate encryption keys to limit the impact of a possible compromise.	Key rotation helps limit the impact of a compromise, but does not protect against any data exposed before rotation.
Account deactivation by an administrator (Business/Enterprise)	An administrator can deactivate a user account, immediately cutting off all access to resources: invalidation of active sessions (authentication) and revocation of access rights (ACLs).	Does not remove file copies already downloaded locally by the user before deactivation.

### 3.6 Identity and access lifecycle management

Retyc uses Keycloak for identity and access management (IAM).

Users can sign in with:

- an email address and password

- external identity providers (SSO) compatible with SAML 2.0 or OpenID Connect.

Sessions are **time-limited**, and inactive users are automatically logged out. In the event of suspected compromise or abnormal behavior, sessions can be manually invalidated by a Retyc administrator.

Access to the authentication service is **restricted to the strictly necessary scope** required for Retyc to operate. A dedicated reverse proxy limits public exposure to the authentication area used by the platform only, reducing the attack surface.

### Access management in organizations

In an organization-based setup (Business and Enterprise plans), administrators can manage collaborator access and revoke access rights.

**Invited but unregistered users** can be removed at any time by an administrator, resulting in the deletion of the data they own (transfers, datarooms).

By contrast, when a user **already has an account at the time of invitation**, removing their access to an organization ends their rights within that organization **without deleting their account or the data** they own.

### Integration with enterprise environments

Retyc offers a mechanism for automatically assigning users to an organization based on their email address (e.g. `@company.tld`).

Retyc also supports integration with identity providers (IdPs) owned by customer organizations, subject to compatibility with Keycloak, notably via standards such as SAML 2.0 or OpenID Connect.

This approach makes it possible to rely on the organization's existing authentication mechanisms (internal directory, SSO, MFA) without duplicating identities, while preserving consistent user affiliation management. The associated security policies, when delegated to the customer's IdP, remain under that customer's control.

---

## 4. End-to-end encryption with mechanisms resistant to quantum threats

---

### 4.1 Beyond conventional encryption

There are three levels of data protection in file sharing solutions:

**Level 1 — Encryption in transit (HTTPS / TLS):** protection during network transfer. Standard today, but insufficient: your data arrives decrypted on the provider's servers, which can access it.

**Level 2 — Encryption at rest (server-side encryption):** files are encrypted on the provider's servers. This protects against physical attacks, but the provider retains the decryption keys. It can access content for maintenance, analysis, or in response to legal requests.

**Level 3 — End-to-end encryption (E2EE):** the approach adopted by Retyc. Files are encrypted on the sender's device before any upload. Only authorized recipients can decrypt them. The provider does not hold the elements required to access content in plaintext.

These first two levels (transport and storage) are now industry standards and do not, by themselves, constitute a confidentiality protection mechanism against the provider.

*The file sharing market often maintains **frequent confusion**: some solutions marketed as "encrypted" actually rely on encryption in transit or at rest, standard mechanisms now widely used and **insufficient to guarantee data confidentiality**.*

*These approaches leave the provider with the **technical ability to access plaintext data** and are not equivalent to end-to-end encryption. Retyc adopts the latter approach.*

### 4.2 age-encryption cryptography

Retyc relies on the **age** encryption format, designed to provide a modern, simple, and robust approach to file encryption. Age can be considered a **trusted component**: its specification is public and formal, it relies on proven cryptographic primitives, and it is designed and maintained by [Filippo Valsorda](#), a recognized cryptographer (former security lead of Google's Go team).

On the web application side, Retyc uses [age-encryption](#), the TypeScript implementation of the [age specification](#). This approach makes it possible to execute cryptographic operations directly client-side, without exposing plaintext content to the server.

The transparency of the cryptographic code is an additional guarantee: the [age-encryption library](#) is open source and its code can be inspected and audited by independent third-party experts. The source code of the web application ( frontend and backend) is not public, but the **Retyc CLI is open source** and auditable, and the underlying cryptographic primitives are as well.

### 4.3 Encryption with post-quantum hybrid keys

When quantum computers reach sufficient power, they will be able to break the conventional asymmetric algorithms (RSA, ECC) currently used in most security solutions.

The *harvest now, decrypt later* strategy is a concrete threat: malicious or state actors capture encrypted data today in anticipation of decrypting it in the future. For data with a long lifetime (contracts, medical files, financial data), this risk is real.

Retyc uses by default a hybrid key scheme integrating mechanisms **resistant to quantum threats**, within the age format.

This hybrid approach aims to reduce the risk that a weakness affecting only one component could by itself compromise data confidentiality.

All keys are generated and encrypted exclusively on the user's device. **Private keys are not accessible in plaintext to the Retyc infrastructure.**

### 4.4 Key management and rotation

Each Retyc user has an asymmetric key pair (public/private):

- The **public key** is stored on our servers and used by senders to encrypt files for that user.
- The **private key** is encrypted with the user's encryption password and stored on our servers in encrypted form. We never have access to it in plaintext.
- **Key rotation** is available on all plans, allowing new keys to be generated and existing data to be re-encrypted.

#### No master key principle

*Unlike traditional "cloud" architectures, Retyc implements no master key ("Master Key") and no emergency recovery mechanism. **This architecture provides no administrator access to plaintext content.** Responsibility for password retention **lies exclusively with the user or their organization's secret management policy** (password vault, IAM).*

### 4.5 Detailed technical implementation

*This section is intended for technical teams and auditors who want to verify the platform's cryptographic properties.*

#### Cryptographic primitives

Retyc relies on [age-encryption](#), the TypeScript implementation of the [age format](#) for client-side encryption operations.

Depending on the mechanisms implemented by this library:

- **X25519**: for key exchange.
- **ML-KEM-768**: as part of hybrid keys integrating mechanisms **resistant to currently known quantum threats** ( MLKEM768-X25519 hybrid scheme). This scheme is implemented via an extension of the age format (key format `age1pq1...`), in addition to the standard age format (FIPS 203).
- **ChaCha20-Poly1305**: for authenticated data encryption.
- **scrypt** (  $N=2^{18}$  ): derivation of the wrapping key during passphrase-based encryption (age `scrypt` stanza).
- **HKDF-SHA-256**: derivation of the payload encryption key from the file key (internal age mechanism).

***Note** — `scrypt` is used by age for private key protection, an operation executed **client-side** in the browser. `Argon2id` is used independently by Keycloak for hashing authentication passwords server-side. These are two separate systems operating across two different scopes.*

## Generation of user key pairs

Key pairs are generated via `generateHybridIdentity()`, which produces a hybrid X25519 + ML-KEM-768 identity. This hybrid scheme aims to reduce the risk that a weakness affecting only one component could by itself compromise data confidentiality: a mathematical breakthrough against X25519 or a weakness discovered in ML-KEM-768.

Since all Retyc user identities are hybrid by default, there is no mix between post-quantum hybrid recipients and non-post-quantum recipients (an explicit constraint of the age specification).

The raw private key is **never transmitted or stored in plaintext** on our servers. It is encrypted client-side with the user's encryption password via age in passphrase mode (`scrypt`), which makes brute-force attacks prohibitively expensive in terms of machine cost.

## Multi-recipient encryption

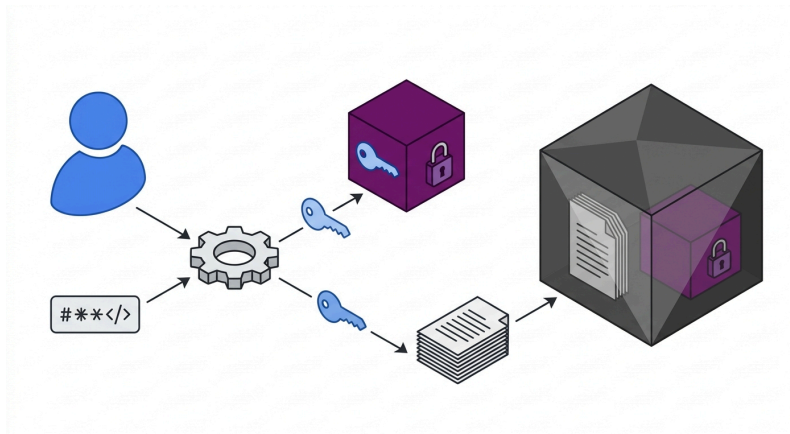
*The principle described below is the reference operation. Variants exist depending on the use case (recipient management, access modality), while preserving the same security guarantees.*

Transfers and datarooms rely on a **shared cryptographic foundation**:

1. A hybrid X25519 + ML-KEM-768 key pair (the **session key**) is generated client-side.
2. Files and **sensitive metadata** (file name, MIME type) are encrypted with the `session_public_key`. For large files, encryption is applied by chunk to limit memory impact.
3. The `session_private_key` is encrypted for each registered recipient via `encryptStringWithRecipients()`.

4. At download time, the recipient decrypts `session_private_key_enc` with their own age private key, then decrypts the files locally with the resulting `session_private_key`.

The server only receives encrypted elements: files, metadata, and the session key are inaccessible to it in plaintext.



### Isolation of cryptographic operations in the browser

All cryptographic operations run inside a dedicated **Web Worker**, isolated from the main UI thread. This architectural choice has two consequences:

- **Performance:** encryption operations, which can be lengthy for large files, do not block the user interface.
- **Isolation:** the cryptographic context is separated from the rest of the application, reducing the attack surface for malicious script injection into the main DOM.

Binary buffers are transferred between threads by ownership transfer rather than copying, limiting in-memory duplication of sensitive data.

## 5. Zero-knowledge architecture: confidentiality by design

---

### 5.1 Definition

**Terminology note** — The term “zero-knowledge” is used here in its common industry sense: the provider has no access to content in plaintext. In academic cryptography, the term refers to zero-knowledge proofs (Goldwasser-Micali-Rackoff), which is a distinct concept. Our usage is widespread in the industry but should be distinguished from its strict academic meaning.

A zero-knowledge architecture means that the service provider has no knowledge (*zero knowledge*) of the content of the data it processes. This property is guaranteed by cryptographic construction, not by an internal policy.

At Retyc, zero-knowledge means in concrete terms:

- We do not have the elements required to access in plaintext the contents of files, file names, associated messages, or users’ private keys.
- If a user forgets their encryption password, we can neither recover their private key nor decrypt the associated data.
- Cryptographic isolation: each space is independent. Compromise of one account or one administrator workstation does not provide any technical leverage to decrypt other collaborators’ data.

### 5.2 Implications for compliance

The zero-knowledge architecture has direct implications for regulatory compliance:

**For GDPR:** encrypted data for which we do not hold the keys is inaccessible to us in plaintext. We cannot exploit it for unauthorized purposes and do not hold the elements needed to access it in plaintext. Retyc nevertheless remains data controller\*\* under the GDPR for account data (email, first name, last name) and **processor** for encrypted content. The zero-knowledge architecture reduces our exposure; it does not exempt us from our regulatory obligations.

**For professional secrecy:** lawyers, chartered accountants, and healthcare professionals are subject to strict confidentiality obligations regarding entrusted data. A purely contractual guarantee is not enough to address this technically. Retyc’s zero-knowledge architecture provides cryptographic reinforcement that we cannot bypass.

**In response to judicial requests:** in the event of an access request from an authority, Retyc can only deliver unreadable encrypted data. We cannot decrypt what we have no means to decrypt.

### 5.3 Deliberate and transparent limits

Rigor requires being precise about what zero-knowledge does and does not cover in our architecture:

- The **title of a transfer or dataroom** is not E2EE-encrypted (this is a design choice to allow display in the interface and notifications).
- **File size** is not encrypted (necessary for storage management).
- The **member list** of a transfer or dataroom is not E2EE-encrypted (necessary for access management).
- **Connection metadata** (IP address, date/time) is collected for security purposes in our authentication system ( Keycloak).
- In a dataroom, a `sha256` hash of the file name is computed with a 256-bit random salt (CSPRNG) encrypted E2EE with the dataroom session key. Only this hash is transmitted to the server for version management. The file name and the salt remain inaccessible to the server in plaintext.

These limits are publicly documented on our transparency page. We prefer total transparency about our actual capabilities over unverifiable claims.

*Although Retyc cannot restore lost access to a private encryption key, we recommend the use of enterprise password managers when registering critical accounts.*

#### Structural limits of E2EE in a web browser

Two fundamental limits, common to any encryption solution operating in a browser, should be explicitly acknowledged:

1. **Public key substitution.** Retyc distributes recipients' public keys via its servers. A sender trusts that distribution. In the event of server compromise or internal abuse, an attacker could theoretically substitute a public key.
2. **Integrity of JavaScript code.** In a web context, the browser executes the encryption code delivered by the server each time a page is loaded. A compromise of the frontend code distribution server could therefore potentially affect cryptographic operations before they are executed. To limit this risk, Retyc applies a strict CSP that **forbids execution of any external script**: only scripts served directly by the Retyc frontend may run in the browser. No third-party, inline, or externally hosted scripts are allowed. This measure removes injection vectors (XSS, third-party supply chain), but does not protect against compromise of the original server itself (a limit inherent to any web-based E2EE model).

These two vectors are inherent to the web E2EE model and are recognized as such across the industry. They do not undermine the strength of the cryptographic model, but they must be taken into account in any assessment of residual risk.

## 6. European technological independence

---

### 6.1 100% European hosting

All data processed by Retyc is hosted **exclusively within the European Union**, with French providers subject to European law:

Provider	Role	Location
Scaleway SAS	Application hosting, storage	France (EU)
Clever Cloud SAS	Application hosting (Keycloak)	France (EU)
Brevo SAS	Transactional emails	France (EU)

Stripe Inc. is used only for payment processing. Data sent to Stripe is limited to the information strictly necessary for the transaction (amount, currency, customer identifier). No user content, file, or transfer metadata is communicated to Stripe. As a U.S. company, Stripe remains subject to the Cloud Act for this payment data — a residual exposure that is acknowledged and limited to that scope.

### 6.2 Hosting certifications

Scaleway and Clever Cloud are **HDS** (French Health Data Hosting) certified by bodies accredited by COFRAC.

These certifications attest to a high level of maturity in security and compliance among our infrastructure providers.

**Retyc itself is not HDS or ISO 27001 certified.** By choosing infrastructure providers subject to these demanding frameworks, we provide our customers with a high level of maturity across the hosting chain. Future evolutions of our architecture and processes may lead us, where appropriate, to consider suitable certification initiatives.

### 6.3 Replication and high availability

Your data (sent files) is **automatically replicated across three European availability zones** (Scaleway). This architecture ensures:

- High availability
- Resilience against failure of a data center.
- No single point of failure at the file storage level.

## 6.4 Independence from Big Tech

*Some solutions mistakenly equate data location with legal protection. Hosting data in a data center located in France or Europe through companies subject to **extraterritorial legislation** (notably U.S. legislation) does not guarantee the absence of exposure to those jurisdictions. This confusion can lead to **overestimating the actual level of protection** provided.*

Retyc uses **no hosting or infrastructure service from U.S. platforms** (such as Google Cloud, Amazon Web Services, or Microsoft Azure). All user data is hosted exclusively with European operators. Stripe Inc. (U.S.) is used only for payment processing (no content data is stored or hosted there).

This independence is not merely ideological: it significantly reduces exposure to extraterritorial risks. The **Cloud Act** (2018) compels U.S. companies to provide U.S. authorities with the data they host, including data belonging to European customers, regardless of where it is stored. By choosing an exclusively European infrastructure, Retyc significantly reduces the risks linked to such extraterritorial legislation.

This distinction between physical location and applicable jurisdiction is often misunderstood, including in commercial communications.

## 6.5 Decision-making sovereignty

TripleStack SAS is an independent French company, with no foreign shareholders and no dependence on international groups. Decisions regarding architecture, security, and data management are made in France, under the exclusive framework of European law.

---

## 7. Features: file transfer and datarooms

---

### 7.1 Secure file transfer

Retyc's file transfer is designed to combine ease of use with maximum security.

#### Transfer process:

1. **Automatic encryption:** the user drags and drops files into the interface. Upon confirmation, the files are encrypted locally in the browser with the public keys of the authorized recipients.
2. **Generation of a secure link:** a unique link is generated instantly. Only the authorized recipient or recipients can decrypt and access the encrypted content.
3. **Secure download:** the recipient downloads and decrypts the file directly in their browser. No software installation is required.
4. **Automatic expiration:** the link expires automatically after the configured period. The files are permanently deleted.

#### Control features:

- Configurable transfer expiration
- Optional password protection for recipients without an account
- Access revocation at any time (link deactivation)
- File reception: generation of links allowing third parties to send documents directly into your secure space
- Ability to generate custom file collection forms

#### Compatibility:

- Works from any modern web browser
- Support for all file types, including large files
- **No restriction on file type:** files are processed without filtering or format-based restrictions.
- No installation required for recipients

### 7.2 Encrypted collaborative datarooms

Retyc datarooms are secure collaborative spaces designed to centralize and manage confidential documents with teams or external partners. Unlike traditional shared storage solutions, security is not handled only through access control lists (ACLs) on a server, but through encryption itself.

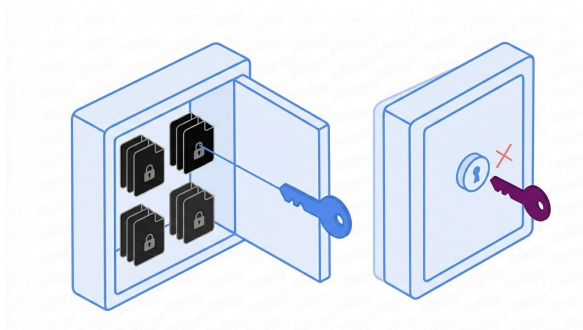
#### Dataroom security architecture:

- **Cryptographic isolation:** each dataroom is cryptographically isolated from the others

- **User-level cryptographic access management:** each dataroom relies on its own cryptographic isolation, and access to session keys is managed individually for each authorized user.
- **Zero Retyc access:** we cannot read documents or file names stored in a dataroom.
- **Cryptographic rekey:** when a member is added or removed, an administrator can trigger re-encryption of the session key for the exact list of authorized members. After this operation, a revoked member can no longer obtain the session key from the server. This operation is at the administrator's discretion and must be performed explicitly.
- **Access responsibility:** no "break-glass" access is maintained by Retyc. Data availability within a dataroom is ensured by the multiplicity of authorized users: as long as one active member can access it, the data remains available.

### Document management:

- Centralized file organization and management
- **Automatic versioning:** each update creates a new version without overwriting previous ones. Full history and restore capability
- Version history management: view, restore, or delete
- Integrated encrypted messaging



### Access management:

- **Granular role-based permissions:** read-only, document upload, full management per user
- Invitation of internal collaborators (organization) and external users (guests)
- Instant access revocation
- Full activity log: all access, downloads, and modifications are tracked

## 8. Operational transparency

---

### 8.1 Open-source cryptography

Retyc relies on the [age-encryption](#) library for client-side encryption operations. This library is open source, and its source code is publicly available for audit. The source code of the web application (frontend and backend) is not public, but the **Retyc CLI is open source** and the underlying cryptographic primitives are as well.

This transparency regarding the cryptographic foundations makes independent verification possible: the library can be inspected and audited by third-party experts, which reinforces confidence in the claimed security properties.

### 8.2 Public transparency page

Retyc publishes a detailed transparency page on its website, accessible to everyone, which exhaustively documents:

- The complete list of cookies used and their purpose
- The data stored in the user's browser
- Full details of all data stored in the database, its purpose, retention period, and level of encryption

This transparency about our practices is publicly accessible on our website.

### 8.3 Zero AI policy

Retyc uses **no artificial intelligence in its product**: no generative AI, no AI-assisted features, and no content analysis through machine-learning algorithms.

This deliberate choice responds to a demand from our professional users, for whom the use of their sensitive content to train AI models represents an unacceptable risk. At Retyc, your files are not used to train third-party models. This is a product commitment that is further constrained by the very nature of our application: we cannot analyze content we cannot read.

### 8.4 External security audit

Retyc integrates security audits carried out by independent third parties into its development process, covering in particular cryptography, infrastructure, and penetration testing. Significant findings, as well as the fixes implemented, are communicated transparently to our users.

## 8.5 Vulnerability disclosure

Retyc takes security very seriously. Our *responsible disclosure* policy is accessible via the [/.well-known/security.txt](#) file on our domain, in accordance with security community best practices (RFC 9116). Any discovered vulnerability must be reported according to the procedures described in that file.

## 8.6 Open-source tools and interoperability

Retyc also provides an open-source command-line interface (CLI), allowing interaction with the platform in an automated way or integrated into technical workflows.

The CLI source code is available on our GitHub repository: [github.com/retyc/retyc-cli](https://github.com/retyc/retyc-cli)

This CLI is intended in particular for technical teams wishing to:

- automate file transfers
- integrate Retyc into pipelines (CI/CD, scripts, internal tools)
- audit and control performed operations

The source code is public and auditable, as part of Retyc's transparency and openness approach.

---

## 9. Use cases

---

### 9.1 Law firms and legal departments

Legal professionals are subject to professional secrecy obligations that cannot rely on simple contractual commitments. Retyc's zero-knowledge architecture provides robust protection and reduces the risk of sensitive data leakage.

#### Typical use cases:

- Secure transmission of contracts, notarial deeds, and sensitive documents to clients.
- Datarooms for M&A transactions, due diligence, and litigation management.
- Secure collection of supporting documents without relying on unencrypted email.
- Collaboration on cases with fellow counsel, experts, or external stakeholders.

**Added value:** the full activity log provides evidence of due diligence regarding confidentiality in the event of litigation or regulatory review.

### 9.2 Accounting firms

Financial data (financial statements, bank statements, payslips, tax filings) is among the most sensitive data categories. Sending it by email represents a major risk.

#### Typical use cases:

- Secure collection of client accounting documents via a personalized unique link.
- Encrypted sending of tax returns, payslips, and closing documents.
- Datarooms for audits, due diligence, and firm transfers.
- Collaboration with peers or experts on complex cases.

**Added value:** client document collection through a unique link replaces unsecured email for day-to-day exchanges. Centralizing documents in secure datarooms facilitates document management and traceability.

### 9.3 Technical and engineering teams

Technical teams regularly handle highly sensitive information: API keys, database dumps, security reports, intellectual property.

#### Typical use cases:

- Secure sharing of penetration test reports and security audits.
- Transmission of configuration files containing secrets.
- Datarooms for confidential project documentation.

**Added value:** independence from Big Tech providers meets the security expectations of the most demanding organizations.

## 9.4 Companies and regulated sectors

For organizations subject to NIS2, DORA, or sector-specific regulations, Retyc can help address certain security and confidentiality requirements, particularly regarding encryption of data in transit and at rest, access traceability, and independence from extraterritorial providers.

**Scope** — *Retyc is an encrypted transfer and collaboration tool. It does not cover all NIS2 or DORA requirements (governance, resilience testing, ICT provider register, etc.). Its adoption should form part of a broader compliance effort led by the organization.*

### Typical use cases:

- Exchange of confidential documents with partners, service providers, or regulators.
- Datarooms for sensitive HR processes (contracts, evaluations).
- Secure transmission of data in due diligence processes or calls for tender.

**Added value:** depending on the selected plan, Retyc can integrate SSO (SAML/OIDC), custom branding, and on-premises deployment for organizations with specific integration requirements.

---

## 10. Plans and deployment

### 10.1 Freemium model with progressive plans

Retyc offers four plan levels allowing every organization to start for free and scale according to its needs:

	Free	Solo	Business	Enterprise
Outbound transfers	Limited	Extended	Extended	Customized
File reception	Yes	Yes	Yes	Yes
Datarooms	Yes	Yes	Yes	Customized
Internal members per dataroom	—	—	Unlimited	Unlimited
E2EE encryption	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>
Key rotation	Yes	Yes	Yes	Yes
SSO (SAML/OIDC)	No	No	On request	Yes
Premium support	No	No	No	Yes
Custom domain	No	No	No	Yes
On-premises deployment	No	No	No	Yes

Details of each plan are available on our website: [retyc.com/fr/pricing](https://retyc.com/fr/pricing).

*End-to-end encryption and key rotation are available on **all plans, including Free**.*

### 10.2 SaaS deployment

The standard Retyc deployment model is SaaS (Software as a Service): no client-side installation, accessible from any modern browser. The entire infrastructure is managed by Retyc and hosted by European providers.

### 10.3 On-premises deployment (Enterprise)

For organizations with regulatory or security constraints requiring data to remain within their own infrastructure, Retyc offers **on-premises deployment** as part of the Enterprise plan.

This deployment model allows the customer organization to run the entire Retyc platform on its own infrastructure while maintaining full control over its data.

## 10.4 Advanced integrations

The Enterprise plan includes the possibility of custom integrations, including:

- **SAML/OIDC SSO:** integration with the existing enterprise directory (Active Directory, Okta, etc.).
  - **Custom branding:** adapting the interface to the organization's colors and logo.
  - **Custom domain:** access through your own domain name.
-

## 11. Conclusion

---

Faced with growing cybersecurity, independence, and regulatory compliance challenges, European organizations need file transfer solutions that provide stronger security and do not rely solely on contractual commitments.

Retyc provides that guarantee by combining:

- **End-to-end encryption:** your files are unreadable on our servers, today and against future threats (hybrid scheme integrating mechanisms resistant to currently known quantum threats).
- **Zero-knowledge architecture:** without decryption keys, we cannot access your content in plaintext.
- **Exclusively European infrastructure:** hosting in France on European infrastructure, with very limited direct exposure to extraterritorial legislation.
- **Full transparency:** auditable open-source cryptography, public documentation of our practices, and honesty about the limits of our architecture.

Retyc is developed and operated by TripleStack SAS, an independent French company. Our mission is to provide organizations and professionals with a European, transparent, and technically rigorous alternative for handling sensitive content.

---

## Contact

---

### TripleStack SAS - Retyc

- Website: [retyc.com](https://retyc.com)
  - Security contact: see [/.well-known/security.txt](https://retyc.com/.well-known/security.txt)
  - GDPR / DPO contact: [privacy@retyc.net](mailto:privacy@retyc.net)
  - Sales contact: via [retyc.com/about/contact-us](https://retyc.com/about/contact-us)
- 

*This document is provided for informational purposes only. The information it contains reflects the state of the platform at the time of writing. Retyc reserves the right to evolve its architecture and features. The current version of the privacy policies and terms of use is the one published on [retyc.com](https://retyc.com).*

— TripleStack SAS, April 2026