



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Right to Erasure in Cloud and Distributed Systems: Can Data Ever be Truly Deleted?

Kushagra Tiwari, Sanidhya SS Chouhan, Priyanshu Danu

National Law Institute University, Bhopal 462044, India

ABSTRACT

The right to erasure presumes that deletion is an event. In cloud and distributed systems, it is a process, unfolding across replicas, tombstones, caches, backups, derived datasets and storage media, each running on its own clock. This article argues that the familiar question of whether data can ever be “truly” deleted is badly framed, because it treats physical annihilation as the only legitimate endpoint and so allows regulators and engineers alike to retreat into abstraction. Drawing on database-systems research, cloud provider documentation, European Union data protection law and India’s Digital Personal Data Protection Act, 2023, the article distinguishes logical, cryptographic and physical erasure; shows that the anonymisation threshold in Recital 26 of the General Data Protection Regulation supplies a workable legal standard for the second; and identifies the temporal weakness of that standard. It then proposes an “erasure horizon” model: a disclosed, tiered and auditable bound on how long any recoverable copy of personal data may persist after a valid request. The article concludes that erasure should be regulated as a verifiable engineering process rather than an instantaneous outcome.

Keywords: Right to Erasure; Right to Be Forgotten; Cloud Computing; Distributed Systems; Cryptographic Erasure; GDPR; Digital Personal Data Protection Act, 2023

I. INTRODUCTION

Consider a customer who presses “Delete my account” on a mid-sized e-commerce platform that runs on a public cloud. In the application, the profile vanishes instantly. In the primary database, however, nothing vanishes. The row is overlaid with a tombstone, a marker instructing the storage engine to treat the record as absent, and that marker must itself travel to replicas in three availability zones, one of which is briefly partitioned from the others. Meanwhile the same record has already been copied into a search index, a recommendation feature store, an analytics warehouse, a nightly snapshot and a disaster-recovery replica in another region, and a transcript of a support call sits in a vendor’s ticketing system. The tombstone will be compacted away in days; the snapshot will expire in weeks; the warehouse copy, never keyed to the customer’s identifier, may never be found at all. At what moment, if any, has the customer’s data been erased?

The law answers with a confident phrase. The General Data Protection Regulation (GDPR) entitles a data subject to obtain erasure of personal data “without undue delay” where, among other grounds, the data are no longer necessary for their original purpose or consent has been withdrawn. India’s Digital Personal Data Protection Act, 2023 (DPDP Act) likewise gives a Data Principal a right to erasure and requires a Data Fiduciary to erase personal data once consent is withdrawn or the purpose has been served. Neither instrument defines erasure. Both speak as though deletion were an event, a verb with a moment of completion, when the scenario above suggests that it is closer to a process with many clocks running at once.

The question in this article’s title is usually answered in one of two registers. The first is fatalism: nothing is ever really deleted, and everything lives on in a backup somewhere. The second is compliance theatre: a “delete” button, a confirmation email, and a privacy policy promising removal within a stated number of days, with little examination of what has been removed from where. Both registers share a hidden premise, namely that “true” deletion means the physical annihilation of every bit that ever encoded the personal data, and that anything short of this is either a failure or a fiction. This article argues that the premise is wrong as engineering and wrong as law.

Part II sets out what the GDPR and the DPDP Act actually demand. Part III dissects the deletion path in modern cloud and distributed systems to show where erased data lingers and why. Part IV separates logical, cryptographic and physical erasure, and argues that the anonymisation standard in Recital 26 of the GDPR supplies a defensible but inherently temporal legal threshold. Part V tests the framework against blockchains, machine-learning models and chains of processors. Part VI proposes an “erasure horizon” model, under which controllers disclose, and are held to, a bounded, tiered and auditable period of persistence.

II. WHAT THE LAW ACTUALLY PROMISES

A. The European Template: Delisting Before Deletion

Article 17 of the GDPR is the most developed statement of the right. A data subject may obtain erasure when the data are no longer necessary for the purposes for which they were collected, when consent is withdrawn, when the data subject successfully objects, when processing was unlawful, or when erasure is required by law. The right is qualified. Article 17(3) preserves processing necessary for freedom of expression, legal obligations, public-interest archiving and the defence of legal claims. Where the controller has made the data public, Article 17(2) requires it to take “reasonable steps, including technical measures,” having regard to “available technology and the cost of implementation,” to inform other controllers processing the data. Article 19 separately requires the controller to communicate erasure to each recipient unless this proves impossible or involves disproportionate effort.

Two features of this text matter for what follows. First, the obligations are expressed in the language of proportionality and technological feasibility: “reasonable steps,” “available technology,” “disproportionate effort.” The legislature thus anticipated that erasure across a chain of systems would not be frictionless, but did not say what the residual standard should be. Second, the surrounding provisions turn erasure into a matter of demonstrable process. The controller must respond within one month, extendable by two further months for complex requests, must implement data protection by design, and must be able to demonstrate compliance with the storage limitation principle. An accountable controller must therefore be able to prove what happened to the data, not merely assert that it happened.

The doctrine’s own history deserves attention because it is rarely mentioned in debates about technical deletion. In *Google Spain*, the Court of Justice recognised a right to have links to outdated or irrelevant information removed from search results, while the underlying web pages remained online. What the Court ordered was dereferencing: a functional, not physical, remedy that made information harder to retrieve without destroying it. The Court later held that dereferencing need not extend beyond the Union’s Member States, accepting a territorially bounded form of unavailability. The right to be forgotten thus began life as a right to be practically unfindable, which shows that European law has long treated the relevant question as accessibility, not the survival of bits.

B. The Indian Framework: Erasure Without a Definition

The DPDP Act is more compressed. Section 8(7) obliges a Data Fiduciary to erase personal data, and to cause its Data Processor to do so, when the Data Principal withdraws consent or as soon as it is reasonable to assume that the specified purpose is no longer being served, unless retention is necessary for compliance with law. Section 12(3) confers on the Data Principal a right to erasure, subject to applicable laws and the manner prescribed. Section 8(1) makes the Data Fiduciary responsible for compliance irrespective of any agreement with a processor, which is significant for the many Indian businesses that run on hyperscale cloud infrastructure they do not control. The Digital Personal Data Protection Rules, 2025 add operational detail, including time-based triggers under which certain large online platforms must treat the purpose as spent after a fixed period of user inactivity, following notice to the user.

Constitutional background points the same way. The Supreme Court’s recognition of informational privacy in *Puttaswamy* included separate reasoning that an individual should be able to control his or her presence on the internet, and High Courts have entertained requests to mask or remove names from published judgments. These decisions concern published records, not storage systems, and offer no test for when erasure is complete.

Read together, the European and Indian texts leave the central technical question open. Neither says whether a tombstoned record is erased, whether a backup that will expire in ninety days is an obstacle, or whether an encrypted blob whose key has been destroyed still counts as personal data. The answers must be built from engineering facts, to which this article now turns.

III. THE ANATOMY OF A DELETE

A. The Tombstone Problem

Most large-scale databases do not erase in place. Systems descended from Google’s Bigtable, including Cassandra, HBase and RocksDB, are built on log-structured merge trees, which achieve high write throughput by never modifying data on disk: new writes accumulate in memory, are flushed to immutable sorted files, and are periodically merged in a process called compaction. A delete in such a system is just another write. It inserts a tombstone that shadows the older record, and the record itself is physically removed only when compaction happens to merge the two files, which may be hours or months later depending on workload and configuration. Researchers examining this behaviour found that mainstream engines offer no upper bound on how long a deleted entry persists, and proposed an engine that guarantees deletion within a configurable threshold, motivated by GDPR-style requirements.

Distribution makes the problem worse. Cassandra retains tombstones for a configurable grace period, ten days by default, so that every replica has time to learn of the delete. A replica that is offline for longer than that period and is then returned to service without repair can reintroduce the deleted record, a phenomenon operators call “zombie” or resurrected data. The legal significance is stark: a controller that has dutifully executed an erasure request may find that its own fault-tolerance machinery has undone it, without any human act of re-processing.

B. Replicas, Caches and the Lineage Gap

Replication ensures that an erasure request is not a single write but a coordinated one. The CAP theorem formalises the constraint: when a network partition occurs, a system must choose between consistency and availability, and many cloud-scale stores choose availability, converging through mechanisms such as hinted handoff and anti-entropy repair. A delete issued during a partition is acknowledged by the replicas that can be reached and propagates to the others only after the partition heals. Eventual consistency suits a shopping cart, but it gives “without undue delay” a systems-level meaning that no statute has examined.

Replicas are the easy case, because they carry the same key. The harder problem is derived data. Personal data is routinely transformed into search indexes, cached fragments served from content delivery networks, event streams, feature stores, log lines, analytics aggregates and training sets, each held in a different system, under a different retention rule, and often without the identifier on which a deletion query would depend. Empirical work on retrofitting GDPR compliance into widely used data stores found that timely deletion was among the harder requirements: systems that expire data lazily or probabilistically could leave expired entries in place, and forcing prompt deletion imposed significant performance costs. This article calls the underlying difficulty the lineage gap. A controller cannot erase what it cannot find, and in distributed pipelines the map of where a person’s data has travelled is typically incomplete before any erasure request arrives.

C. Backups and the Immutability Paradox

Backups expose a direct conflict between two legal duties. Article 32 of the GDPR requires the ability to restore the availability of and access to personal data after an incident, and the DPDP Act requires reasonable security safeguards to prevent personal data breach. Modern practice meets these duties through immutable backups that deliberately cannot be altered or deleted for a fixed period, as a defence against ransomware and insider threats. Amazon S3’s Object Lock in compliance mode, for example, prevents an object version from being overwritten or deleted by any user, including the account’s root user, until the retention period has elapsed. A backup that an erasure request could edit is a backup that an attacker could edit. The security architecture and the erasure right pull in opposite directions.

Regulators have responded pragmatically, though informally. The United Kingdom’s Information Commissioner’s Office, for instance, accepts that where data cannot immediately be removed from a backup, the controller may delete it from live systems and put the backup copy “beyond use,” provided that it is not restored for any purpose without erasure being reapplied and that it is overwritten on its normal schedule. Cloud providers have built similar logic into their products: Google Cloud documents a deletion pipeline in which data is first scheduled for deletion, then removed from active systems, and only later from backup systems. Tolerance for backups is therefore already a bounded-time rule in all but name. What is missing is the bound.

D. The Last Mile: Physical Media

Beneath every logical layer lies physical storage, and here a cloud tenant has no visibility at all. On solid-state drives, a flash translation layer remaps logical addresses to physical pages and performs wear levelling, so that overwriting a file does not necessarily overwrite the cells that held its earlier contents. Researchers testing commercial drives found that software techniques designed for individual files frequently left recoverable remnants, and that even some drives’ built-in sanitisation commands were implemented incorrectly. For this reason, the leading United States sanitisation guideline recognises cryptographic erase, the destruction of the key protecting encrypted media, as a sanitisation technique for suitably encrypted devices. The shift from overwriting to key destruction is not a convenience. It is a concession that, at the hardware layer, the most reliable way to make data irrecoverable is to make it unintelligible.

IV. THREE CONCEPTS OF DELETION

The preceding survey suggests that “deletion” names at least three different operations. Logical erasure makes data unavailable through the system’s ordinary interfaces: a flag, a tombstone, an unlinked file, a revoked access path. Cryptographic erasure destroys the key that renders data intelligible, leaving ciphertext that may persist indefinitely but cannot be read. The idea is old. Boneh and Lipton proposed it for backups in 1996, observing that revoking a key can revoke every copy of the data encrypted under it. Physical erasure overwrites or destroys the medium. The three differ in speed, cost, verifiability and, crucially, in what an adversary with access to the underlying storage could still recover. Logical erasure leaves plaintext recoverable by anyone working beneath the interface. Cryptographic erasure leaves ciphertext recoverable but, if the key is truly gone, meaningless. Physical erasure, where achievable, leaves nothing.

The legal consequence turns on how the law classifies ciphertext whose key has been destroyed. Recital 26 of the GDPR excludes from the notion of personal data information that does not relate to an identifiable person, and tests identifiability by reference to “all the means reasonably likely to be used,” taking account of the cost and time required, available technology and technological developments. The Court of Justice has read this to exclude means that are prohibited by law or practically impossible, and the Article 29 Working Party’s opinion on anonymisation requires that anonymisation be irreversible in the sense that neither the controller nor any third party can reasonably re-identify the data. The General Court has since endorsed a relative approach, under which pseudonymised data may not be personal data in the hands of a recipient who lacks reasonable means of re-identification, a position the Court of Justice has been asked to review on appeal. On these authorities, ciphertext whose key has been irrecoverably destroyed is a strong candidate for treatment as anonymous data, and cryptographic erasure can satisfy the erasure right in substance even though the bytes remain on disk.

That conclusion carries three qualifications, each specific to distributed environments. First, key destruction is itself a distributed deletion problem. Keys are replicated for durability, cached in memory, escrowed in hardware security modules and captured in key-store backups. Managed key services build

in deliberate delay: AWS Key Management Service, for example, imposes a waiting period of seven to thirty days before a scheduled key deletion takes effect, to guard against accidental loss. Crypto-shredding therefore relocates the horizon problem rather than eliminating it, since the erasure is real only when the last copy of the key is gone. Second, cryptographic erasure reaches only what was encrypted under the destroyed key. Plaintext copies in caches, logs and derived stores sit outside the key hierarchy, so the technique works only where data has been architected around per-subject or per-cohort keys from the start.

Third, and most important, the legal standard is temporal. Recital 26 requires regard to technological developments, so an assessment that ciphertext is unreadable is a prediction about the future, and predictions can expire. Where a data key was ever wrapped or transported using public-key cryptography, an adversary who has captured those wrapped copies may hope to recover the key with a future quantum computer, a “harvest now, decrypt later” strategy that explains why standards bodies have begun publishing post-quantum key-establishment standards. Ciphertext whose key has been destroyed is therefore anonymous in the way a sealed envelope is anonymous: for as long as the seal holds. An erasure that satisfies the law today must be capable of being reassessed tomorrow.

Distributed systems also offer a cautionary tale about attempts to engineer forgetting without accountable key destruction. The Vanish system proposed that keys be split into shares scattered across a peer-to-peer network whose natural churn would erase them after a time, so that data would “self-destruct.” Subsequent research defeated it with low-cost attacks that harvested the shares before they vanished. The lesson is that forgetting cannot be delegated to the passive behaviour of an untrusted network. It must be an accountable act by an identifiable party, which is precisely what data protection law presupposes.

V. HARD CASES

A. Blockchains and Deliberate Immutability

Public blockchains are engineered so that history cannot be altered, which seems irreconcilable with erasure. The European Data Protection Board’s draft guidelines on blockchain processing take the position that technical difficulty does not excuse non-compliance, and that controllers should design systems so that personal data is not stored on-chain in the first place, keeping it in off-chain storage that can be erased and committing only references to the ledger. Even that pattern needs care. A hash of personal data may remain personal data where the underlying value can be guessed or re-derived, as the Working Party’s anonymisation opinion cautions in its discussion of hashing. The blockchain case thus confirms the central claim of this article: the legal question is not whether the chain survives, but whether what survives on it relates to an identifiable person by means reasonably likely to be used.

B. Models That Remember

Machine-learning models are derived data in an acute form. Training embeds information from individual records into millions of parameters, and removing a record’s influence by retraining from scratch is prohibitively expensive at scale. Research on machine unlearning has proposed ways to make systems forget specific training points, including by partitioning training so that only the affected shards need retraining. The European Data Protection Board has meanwhile opined that whether a model trained on personal data is itself anonymous must be assessed case by case and cannot be assumed. For erasure, the implication is that the controller’s inventory of systems must include the model, and that “derived data” is not a safe harbour.

C. Chains of Processors

Cloud erasure is a relationship, not an act. The GDPR requires processors to delete or return personal data at the end of the services and requires the controller to notify recipients of erasure, and the DPDP Act makes the fiduciary answerable for its processors. In practice, tenants cannot inspect the provider’s storage layers, cannot observe compaction, and typically receive only a contractual assurance. Responsibility without observability is the structural weakness of cloud erasure, and it is why verification, not merely promise, must become the centre of the regulatory standard.

Indian practice faces a further complication. Retention mandates elsewhere in the law, such as the CERT-In directions requiring service providers and other covered entities to maintain logs for a rolling period of 180 days, compete with erasure, and Section 8(7) of the DPDP Act expressly permits retention where necessary for compliance with law. Every such mandate extends the horizon for some class of data, and must therefore be visible in any honest account of how long erasure takes.

VI. A PROPOSAL: THE ERASURE HORIZON

If true deletion is neither achievable nor required, the regulatory task is to replace an unattainable absolute with an accountable bound. This article proposes the concept of an erasure horizon: the maximum interval, measured from a validated request, after which the controller commits that no copy of the personal data remains recoverable by any means reasonably likely to be used, together with a tier-by-tier schedule showing how that commitment is met. The model has four components.

First, the controller maintains a tier map identifying every class of system in which personal data may reside: live replicas, caches and edge nodes, derived stores such as indexes and feature tables, logs, backups and archives, vendor systems, and models. Each tier is assigned an erasure method (logical, cryptographic or physical), an accountable owner and a maximum sub-horizon. The map answers the lineage gap directly, and it converts the accountability principle from an abstraction into an inspectable artefact.

Second, tiers with long natural horizons, above all backups and archives, should be built around cryptographic erasure by default, using envelope encryption in which per-subject or per-cohort data keys are wrapped by master keys, so that destroying one data key neutralises every copy encrypted under it. Key destruction should itself be treated as a tier, with its own disclosed sub-horizon covering key-service waiting periods and key-store backups. Where per-subject keys are impractical, cohort keys (for example, one per customer segment per quarter) offer a coarser but still bounded alternative, trading granularity for cost.

Third, the model requires a restoration rule. Because backups are immutable, erased data will reappear whenever a backup is restored, so the controller keeps an erasure ledger of requests honoured, and no restored system may return to service until the ledger has been replayed against it. This operationalises the “beyond use” principle and also addresses the zombie-data problem for replicas returning from long partitions. The ledger is itself a privacy risk, since it records that particular individuals sought erasure. It should therefore store only keyed, non-reversible identifiers and carry its own retention limit. The tension is real, but smaller than the alternative of silent resurrection.

Fourth, the model demands disclosure and proof. The response to the data subject should state the horizon in plain terms, for example that live systems were cleared on a given date and that backup copies will cease to be recoverable by a stated later date. Internally, each tier should emit a dated completion record, and processors should be contractually required to supply equivalent attestations so that the chain of responsibility is documented end to end. Verification need not expose a provider’s infrastructure: signed attestations and key-destruction logs can supply the evidence.

For regulators, the model suggests a more precise enforcement standard than “truly deleted.” A controller should be treated as having erased data when, within a disclosed and proportionate horizon, the data are inaccessible in the ordinary course, are not used for any purpose, cannot be restored without erasure being reapplied, and are irrecoverable by means reasonably likely to be used, with the horizon reassessed as technology changes. Outer bounds could be set by presumption: active tiers within the one-month response period already found in Article 12(3) of the GDPR, and passive tiers within a period tied to the declared backup retention cycle, with any longer period requiring documented justification. These figures are policy choices rather than technical necessities, and a body such as the Data Protection Board of India, which will elaborate the DPDP framework through rules and guidance, would be well placed to calibrate them by sector and by firm size. Small organisations could be offered a safe harbour through managed services that provide cryptographic erasure and attestations as standard.

Three objections deserve a response. It may be said that the model legitimises persistence. It does the opposite, by replacing open-ended claims of “technical impossibility” with a bound that can be audited and breached. It may be said that it burdens controllers. It does, but the burden is work that data protection by design already implies, and it is cheaper to build in than to retrofit. And it may be said that long horizons will be abused. The answer is the presumption and justification requirement described above, backed by enforcement against horizons that cannot be explained by security or legal necessity.

VII. CONCLUSION

Can data ever be truly deleted? If “truly” means that no bit encoding the information exists anywhere, the answer in distributed systems is almost never, and the honest answer is that no one could prove otherwise. But that is the wrong test, and it has never been the legal one. European law began with delisting, defines identifiability by reference to means reasonably likely to be used, and already tolerates backups on a bounded basis. Indian law speaks of erasure without defining it and ties retention to purpose and legal necessity. Both are compatible with a different answer: data can be made, within a bounded and disclosed period and by methods that can be evidenced, irrecoverable by any reasonably likely means.

That answer has a cost. It requires abandoning the picture of deletion as an instantaneous event and accepting that erasure is a process with a duration, a method and a threat model, each of which can be wrong and must be revisited as technology and adversaries change. It requires controllers to know where personal data lives, to design key hierarchies before they need them, and to say, truthfully and in advance, how long the process takes. It requires regulators to stop asking whether data is truly deleted and to start asking how long it takes, who can prove it, and what has been done to ensure that it does not come back. A right that can be verified is worth more than a promise that cannot be falsified, and the erasure horizon is the form that verification should take.