



TOWARDS GDPR COMPLIANCE IN PUBLIC PERMISSIONLESS BLOCKCHAINS

European Blockchain Association

Reply to the European Data Protection Board (EDPB) Public Consultation

Erwin Voloder, Head of Policy EBA
Eugenio Reggianini, Head of Growth, EBA

We would like to express our appreciation to the European Data Protection Board (EDPB) for the timely release of the consultation draft Guidelines 02/2025. The evolving interplay between GDPR and blockchain technology (particularly in permissionless networks) remains one of the most pressing regulatory frontiers for both industry and public sector actors. The EDPB's effort to clarify how core data protection principles apply in decentralized contexts is commendable, and this consultation represents a pivotal step in fostering regulatory clarity and legal certainty.

This reply is structured to contribute meaningfully to that effort by offering a technically rigorous, yet legally grounded, assessment of the open issues identified in the draft. **It emphasizes the ongoing evolution of blockchain architecture from monolithic systems to modular and layered networks and focuses specifically on the Ethereum ecosystem as a reference model not for ideological reasons, but because Ethereum is both the most widely deployed public smart contract platform and the most advanced in terms of implementing modular privacy-preserving features at scale.** Moreover, Ethereum has a long history of protocol development with a wider development community. In doing so this provides a larger foundation to be taken as a plausible assumption for any regulatory considerations.

In particular, the reply analyzes how Ethereum's transition to modularity, via innovations such as proposer-builder separation (PBS), data availability sampling (PeerDAS), zk-SNARK-based execution, and other privacy-enhancing technologies, creates novel and sometimes ambiguous role allocations under GDPR, challenging traditional data controllership concepts.

The structure of this consultation reply is as follows:

1. Review of the EDPB Draft Guidelines: A summary and interpretive framing of the EDPB's position as it relates to blockchain-based data processing.

2. Open Points on Role Attribution in Modular Architectures: A legal-technological analysis of execution, consensus, and data availability layers, highlighting areas requiring further guidance.

3. Taxonomy Alignment via Ethereum Technical Innovations: A technical account of how Ethereum's protocol upgrades embed data minimization and reconfigure controllership responsibilities.

4. Harmonized GDPR Compliance Framework for Modular Chains: A proposed set of best practices for aligning modular blockchain systems with GDPR, with roles and obligations mapped before and after anonymization/pseudonymization.

5. Outstanding Questions: An identification of unresolved legal-technical gaps and suggestions for pragmatic compliance, including metadata erasure, separation of roles, and the need for continuing EDPB engagement.

6. Policy Recommendations: Aiming to balance regulatory considerations with the evolving nature of modular blockchain networks and the role of privacy preserving technologies within them.

In synthesizing these points, this submission seeks to align future-facing network architectures with foundational data protection principles. Our objective is to foster a compliance path that is both technically feasible and normatively robust, ensuring that decentralization and data protection evolve in mutual reinforcement, not contradiction.

1. Review of the EDPB Consultation Guidelines

The below points outline how the EDPB frames key concepts for GDPR applicability within its guidelines:

- The EDPB has affirmed that the GDPR applies to personal data on public, permissionless blockchains. In Guidelines [01/2022](#) and [05/2022](#), the EDPB emphasizes that fundamental definitions like data controller (Article 4(7) GDPR) retain their meaning in blockchain contexts. A controller is any party who determines the purposes and means of processing personal data. This means that simply because a blockchain is decentralized, it is not exempt from identifying controllers and allocating responsibility. Notably, node operators in public blockchains may be considered (joint) controllers if they exert significant influence on the purposes and means of processing. The act of writing transactions to a ledger or validating blocks constitutes data processing under GDPR, since it involves operations on information relating to identifiable individuals. The EDPB's 2025 draft guidance (Guidelines [02/2025](#)) underscores that blockchain's core features (like immutability and distribution) can conflict with GDPR obligations such as data minimization and storage limitation.
- **Pseudonymization vs. Anonymization (Recital 26):** Both the 2025 draft and prior guidelines draw a sharp line between pseudonymized and anonymized data. Recital 26 GDPR clarifies that data which have been pseudonymized but can be re-attributed by using additional information must still be treated as personal data. In other words, hashing or encrypting personal data does not remove it from GDPR's scope if the key or original reference can be obtained by someone. The EDPB explicitly notes that even modern encryption or hash functions do not make data anonymous per se – encrypted or hashed data is still considered personal data under GDPR if re-identification is reasonably possible. Truly anonymous information, by contrast, is data rendered irreversibly such that no individual is identifiable by any party. This distinction is critical for blockchain designs

that rely on cryptographic techniques: merely using addresses or hashes (a common practice for “pseudonymous” blockchain transactions) does not equate to anonymization if some entity (e.g., an off-chain KYC provider or analytic firm) holds a mapping of those pseudonyms to real identities. EDPB Guidelines 05/2022 (on facial recognition) reinforce that pseudonymization should be implemented at the earliest stages of processing as a safeguard, but even then the data remains personal throughout its lifecycle if reversibility is possible. The goal, aligned with Recital 26, is to push systems toward genuine anonymization wherever feasible, so that data protection principles would no longer apply.

- **Right to Erasure (Art. 17) in Immutable Ledgers:** Perhaps the thorniest issue is the “right to be forgotten” (erasure) in an immutable, append-only blockchain. The Guidelines 01/2022 and the 2025 draft acknowledge that the immutability of blockchain records is a fundamental obstacle to Article 17 GDPR. Once personal data is written to a public chain, it cannot be simply deleted or altered without undermining ledger integrity. The EDPB’s position is that this dilemma must be addressed by design. Concretely, storing personal data on-chain should be avoided wherever possible. Instead, personal data should reside off-chain (in databases or encrypted storage) with only references or commitments on-chain. If a deletion request is received, the controller can remove or unlink the off-chain data, thereby “erasing” the personal aspect of the on-chain reference. For example, if an on-chain transaction contains a hash pointer to a user’s data stored off-chain, deleting the off-chain record (and any keys) can render the on-chain hash meaningless (no longer attributable to an individual). This technique is sometimes called “metadata erasure”, and the EDPB explicitly recommends it: by removing the off-chain data such that on-chain data no longer has any personal reference, the right to erasure can be honored in substance. Additional on-chain transactions can also be used to flag that certain data is revoked or erroneous (though the original record remains, a new state can supersede it). It’s worth noting that the right to rectification (Art. 16) faces similar challenges – one cannot alter a past block, but one might append a correction or update off-chain and reference it going forward. The EDPB’s overall guidance is to engineer systems so that personal data can be effectively erased or corrected, even if the blockchain itself only allows additions. This is closely tied to the notion of data controllership: whoever introduces personal data into a blockchain must ensure they have a mechanism to fulfill deletion or correction requests, likely by controlling an off-chain component.
- **Data Controllership and Responsibility:** The guidelines repeatedly stress that decentralization does not absolve organizations of responsibility. Under GDPR’s broad definitions, multiple actors on a blockchain might be considered controllers, or even joint controllers, if they collaboratively determine how and why personal data is processed. The EDPB’s July 2021 guidance on controller/processor concepts (referenced in Guidelines 02/2025) is invoked to analyze blockchain roles. In a private or permissioned blockchain, typically a defined entity (or consortium) orchestrating the network will be the controller, making governance straightforward. However, in permissionless networks, role attribution is far more complex. The 2025 draft guidelines concede that participants in a public

blockchain do not all have equal roles; one must consider the factual influence each participant has. For instance, a smart contract developer who decides to record certain personal data on-chain for a given application might be a controller for that processing, whereas a miner/validator who merely includes the resulting transaction in a block might or might not be a controller, depending on their level of discretion. The EDPB makes a special mention that nodes “not acting on behalf of a controller” and who can independently decide to include or exclude transactions (or even fork the chain) may end up as controllers in their own right. Conversely, if nodes perform purely automated tasks under someone else’s direction (rare in public chains), they might be considered processors – but in a truly decentralized chain, there is usually no single entity instructing the others, so processor roles are uncommon. The Guidelines therefore encourage participants to establish governance frameworks or legal entities (e.g., a foundation or consortium) to assume the role of controller for the blockchain’s operations in order to clarify responsibilities. In fact, if a set of public nodes jointly determine protocol rules (like what data goes on-chain), the EDPB strongly encourages them to formalize a joint controllership arrangement – for example, via an association that can serve as the accountable entity. This is a pragmatic suggestion to avoid the situation where every single node operator is individually hit with full controller obligations in isolation.

The guidelines overall enumerate specific GDPR obligations in the context of monolithic network context. They stress data protection by design and default (Art. 25 GDPR) must be at the heart of blockchain solutions. This means before launching a blockchain service, the stakeholders should ensure that the system minimizes personal data on-chain, secures any necessary personal data with state-of-the-art measures, and makes data flows transparent. A Data Protection Impact Assessment (DPIA) is considered almost mandatory for blockchain projects, given the likely high risks. The DPIA should evaluate whether using a blockchain is necessary and proportionate, which data will be on-chain vs off-chain, the risks to individuals, and how international transfers will be managed (since nodes may be globally distributed).

On that note, the EDPB points out that public blockchains inherently involve international data transfers, so controllers using them should ensure a valid transfer mechanism (the draft hints at the use of standard contractual clauses or other Chapter V GDPR tools for nodes outside the EU). Additionally, data subject rights like the right of access, information, and data portability must be facilitated even if data is distributed. For example, an individual should be able to request from the controller what personal data of theirs is on the blockchain and get an intelligible answer, even though that data is replicated across many nodes.

The EDPB suggests that a “central point (e.g., the controller) must be accessible” to provide such information or to act on rectification/erasure requests.

This implies that application providers or blockchain consortia need to set up effective communication channels for data subjects, despite the decentralized backend. Transparency (Art. 13/14) is also vital: users should be clearly informed before putting data on a blockchain about how it will be processed, who will have access (potentially everyone, if public), and what rights or

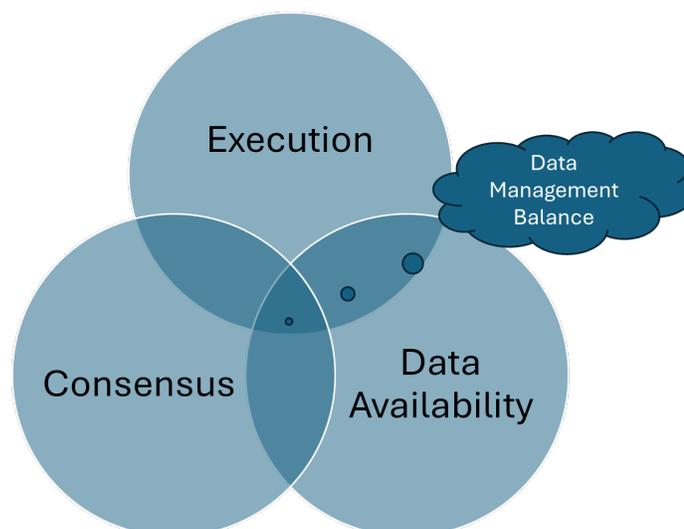
limitations exist. All these points from the EDPB's guidance condense into a clear principle: permissionless blockchain systems are not a lawless space. The GDPR fully applies, and compliance must be achieved through careful technical and organizational design choices. If a certain blockchain use-case cannot be achieved in a GDPR-compliant way (for instance, if it would require putting sensitive personal data immutably on-chain), the guidelines frankly suggest reconsidering the use of blockchain altogether or opting for a permissioned chain where governance and access can be controlled.

2. Open Points from the EDPB Guidelines in Role Attribution Across Modular Blockchain Layers

Despite the EDPB's guidance, there remain unresolved questions about how GDPR roles map to the increasingly modular architecture of permissionless blockchains that have shown real world implementation. In taking Ethereum as a reference, the post-Merge ecosystem is split into distinct layers:

(i) The execution layer where data is collected through application services and interfaces and proposed in the form of interchain hashed values (block), (ii) the consensus layer where the block of data is validated and finalized, (iii) and the data availability layer which is used for data storage. Each layer has different actors and functions. Assigning legal responsibility across these layers poses novel challenges.

Figure 1: This picture intends to represent the coordination between different layers in a permissionless blockchain network: execution collects data from apps, and participates in block origination, consensus propagates block information, and data availability offers data storage. Finding a balance between the three of them is key for data management.



Below are key open points in role attribution that both regulators and practitioners are still grappling with.

- **Execution Layer (Transaction Origination and Smart Contracts processing):** In Ethereum, the execution layer handles transaction execution and smart contract state changes. Open questions persist on who the controller for on-chain processing of personal data at this layer is. For example, consider a decentralized application (dApp) that processes personal data via a smart contract. The dApp developer or deploying entity may determine the purpose of that processing (e.g., storing user information on-chain), which suggests they may be a data controller. The end-user who submits their personal data in a transaction could also be a controller (for their own data) or at least a data subject with rights. But what about the node (execution client) that executes the transaction in a block? Is the node operator a mere facilitator (akin to a processor), or are they a joint controller because they decided to run an Ethereum node that inevitably processes all sorts of data in the mempool and blocks?

The EDPB draft hints that if nodes have no discretion about including a given transaction, they just execute whatever the consensus includes, their role might be more ministerial. However, in practice miners/validators do exert choice, they select transactions from the mempool (often prioritizing by fees, which is an economically driven purpose). This means a block builder on the execution layer might arguably determine “which personal data gets recorded on-chain” (by choosing transactions), thereby influencing the means of processing. That leans toward a controller role for block builders or proposers. Since there is no central authority instructing them on processing, many experts read the GDPR as treating them as independent controllers for the data they handle.

The open point is how to clearly delineate the dApp developer’s responsibility versus the blockchain infrastructure’s responsibility. Should the blockchain be viewed as just a platform (with the dApp as controller), and nodes as neutral intermediaries? Or are nodes co-determining the purpose (which is to maintain the ledger and include transactions) and thus jointly responsible? The EDPB suggests looking at factual influence. In Ethereum’s execution layer, influence is shared in a complex way, so this remains a gray area.

- **Consensus Layer (Block Validation & Finalization):** the consensus layer (e.g., [Ethereum’s Beacon Chain with Proof-of-Stake validators](#)) is responsible for agreeing on the canonical chain of blocks. Here the question is whether validators/attesters in the consensus process controllers of personal data are contained in blocks. Each validator receives blocks (which may contain personal data in transactions) and participates in voting on them and finalizing them. Do they “determine the purposes and means” of that data processing? On one hand, the purpose of a validator’s processing is to secure the blockchain and earn rewards, not necessarily to process any specific personal data. They might say they would process any payload in a block blindly, without purpose to handle personal information. On the other hand, by consenting to include a block in the chain, they are making a decision that results in personal data being immutably recorded and

replicated, which is arguably a decision about the means of processing that data (the means being a global broadcast via blockchain). The EDPB draft acknowledges that “validators in a public blockchain are not taking instructions on behalf of someone else and may have decisive influence on adding transactions or even changing protocol rules” (through forking network or consensus governance rules). This indicates they could be seen as joint controllers together with the execution client, collectively determining that transactions get processed on-chain.

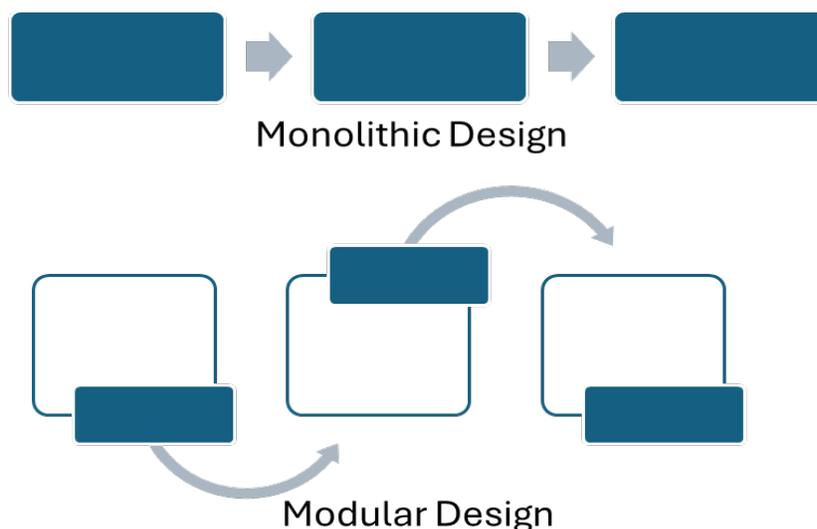
The unresolved point is the extent of joint controllership among hundreds or thousands of pseudonymous validators. Joint controllership under GDPR normally requires a clear arrangement between parties (per Article 26 GDPR), which is impractical in a permissionless setting. No legal agreement binds anonymous validators together in defining purposes; the “agreement” is essentially the blockchain protocol rules to which every node operator is bound to. There’s also a counterargument, perhaps validators are more analogous to “processors” of the dApp developers or users simply following the coded instructions (the protocol) to carry out data processing. But since no one entity can truly instruct a validator (they can always act independently, e.g., censor a transaction or not participate), the processor analogy is a weak one. This role ambiguity at the consensus layer is unresolved in current guidance.

- **Data Availability Layer (Data Storage):** with permissionless networks going into a more specialized and modular set of functions, data storage also called “availability” is being separated from execution. Solutions like sharding or external Data Availability (DA) layers mean that the bulk transaction data might be stored by a network of peers dedicated to data availability sampling (as in Ethereum’s upcoming Danksharding with PeerDAS). Here, nodes might only store fragments of encrypted or erasure-coded data rather than full plaintext transactions.

The open question is how to assign GDPR roles in this scenario. If a data availability committee or network stores chunks of potentially personal data such as pieces of rollup transaction blobs, are those nodes controllers for storing those pieces? Each node might not even be able to read the user data, for example, if the data is erasure-coded, a single chunk is meaningless by itself (it’s not identifiable information until reassembled). One could argue that storing a fragment that by itself is not identifiable might fall outside GDPR (as anonymous data) from that node’s perspective. But collectively, the network ensures the full dataset is available, so someone (any user who queries enough nodes) could reconstruct it. It’s unclear whether regulators would treat each DA node as responsible only for the fragment (which might be anonymous) or hold them jointly responsible for the whole dataset being available. If the DA layer is a separate chain or service (e.g., [Celestia](#) or another L1 providing DA), there may be a distinct controller for that layer’s operations. Another unresolved point is data retention on the availability layer. Ethereum’s plan (via EIP-4844 and sharding) is that nodes only keep data for a short period and after that it’s pruned. There is a general sense that minimizing data stored by any single party (via sampling) is good for compliance (aligning with data minimization).

In conclusion, as Ethereum and other permissionless chains evolve, a multi-layer, multi-actor environment emerges in which controllership may be distributed in unprecedented ways. The traditional GDPR framework assumes centralized decision-making over data. Mapping these concepts onto a decentralized, layered architecture remains an ongoing challenge. **The EDPB’s current guidance provides principles based on a typical monolithic approach to networks which does not represent reality and future network evolution, at least in the permissionless space.** These open points call for further clarification either through future EDPB guidance or perhaps jurisprudence down the line.

Figure 2: The picture illustrates the way in which permissionless blockchain network design evolved. So-called “monolithic” designs were the first to be developed. In this setting, data is fed into the block from a single off-chain source, with the resulting block data evenly redistributed. “Modular” designs, conversely, require the coordination of multiple network participants within the block, so technically they have a more complex set up and different block data distribution. At the same time, they aim to improve performance and provide clearer classification across data origination, processing, and storage.



3. Taxonomy Alignment Through Ethereum Technical Innovations

To address the above ambiguities and to better reconcile blockchain functionality with data protection, the Ethereum community is actively developing technical innovations. These innovations aim to delineate roles more cleanly and embed privacy protections at the protocol level. By adjusting “who does what” in processing transactions, Ethereum’s roadmap can help align with a data protection taxonomy (who is controller/processor) in a clearer way.

The following section analyzes several key innovations and how they could contribute to privacy and functional role specification within permissionless networks (taking Ethereum as a reference).

Proto-Danksharding & Blob Transactions

With the Dencun upgrade's [EIP-4844](#) “proto-danksharding,” Ethereum introduced blob-carrying transactions as special transactions that carry up to 128 KB of binary data called “blobs” alongside the usual call data. Crucially, raw blob data is not stored permanently on-chain; instead, blobs live in the consensus layer as SSZ “sidecars” and are retained by most clients for exactly 4 096 epochs (\approx 18 days) before being pruned to limit disk growth. After expiry, only a KZG commitment (a small on-chain fingerprint) remains, ensuring that the chain can still cryptographically attest to the blob’s prior existence without carrying its full contents.

Because blobs are transient and erasure-coded at the consensus layer, individual nodes no longer need to process or store terabytes of rollup call data indefinitely. Instead, most clients allocate tens of gigabytes (\approx 48 GB) for blobs during their retention window and then prune them. The on-chain footprint is likewise minimized. Verifiers see only small commitments and proof checks, never the full data. This shift embodies the GDPR’s data minimization and storage limitation principles by ensuring that most nodes process only the bare minimum and only temporarily of potentially personal payloads.

zk-SNARK-based Execution for On-chain Data Minimization

Ethereum is actively exploring zero-knowledge proofs (zk-SNARKs) to verify computations and even entire blocks without re-executing every transaction. A zkEVM is a ZK-compatible version of the Ethereum Virtual Machine, allowing for the generation of succinct proofs that a set of transactions were executed correctly and that the new state root is correct, without revealing all intermediate computations to every validator.

Several projects (Scroll, Polygon zkEVM, zkSync, etc.) are working on zkEVMs primarily for Layer 2. Vitalik Buterin has even [suggested](#) a radical long-term idea: replace the EVM on Layer 1 with a RISC-V based zkVM to make proving more efficient. RISC-V is a simple, open CPU architecture that is easier to translate into circuits for ZK proofs, potentially yielding huge scalability gains.

- **Verification without plain data:** In a zk-SNARK execution model, a block proposer (or a rollup sequencer) can generate a cryptographic proof that all transactions followed the rules and updated the state correctly. Other nodes then verify the proof instead of executing the transactions themselves. This means validators do not necessarily need to see or process each transaction’s details. They might still receive some data (especially for data availability, as discussed), but they could validate the state transition purely through the proof. If designed carefully, personal data within transactions could be hidden. A proof could confirm a transaction’s validity, balance, signature, etc., without revealing the user’s identity, reducing on-chain personal data exposure validators would essentially be dealing with mathematical attestations, not user data.
- **Data minimization through cryptography:** A zkEVM can enable scenarios like pseudonymous user credentials through zero-knowledge proofs to determine asset

transfers, where the chain only sees commitments and proofs, but never the raw personal data. If Ethereum L1 moves toward accepting zk-proofs of transactions natively, then much of the execution layer could be handling pseudonymous or even anonymous data from the perspective of the validators. The only entities seeing the actual personal data would be the ones constructing the proofs (likely the user's wallet or an application). This aligns with the GDPR principle of processing the least identifiable data possible at each step.

- **Examples of privacy features:** Layer 2 ZK-powered rollups are providing hints of what's possible. For example, some zk-rollups can have shielded transactions, but still produce a proof that everything balances out. If L1 accepts only the proof and some commitments, the L1 never sees personal data (like who paid whom). Instead, the L1 sees, for example, "a valid transfer proof for token X has been submitted" with no further information. That shifts the controller burden to the L2 operator or the user who had the information, whereas L1 might claim to be just a passive verifier of encrypted or abstract data. This again matches data minimization principles under GDPR.

zk-SNARK based execution shifts permissionless networks and in the example of Ethereum, toward a paradigm of "verify, don't see." By minimizing how much personally identifiable information is visible to network participants, it potentially allows most validators to operate without processing personal data (they process proofs, which ideally cannot be linked to identities). It also can support selective disclosure. Only those who need to know the data (perhaps the transacting parties, or a regulator with a viewing key) can see it, others cannot. This granular control is very much in spirit of GDPR's data minimization and confidentiality requirements (Art. 5(1)(c) and Art. 32).

Of course, a challenge remains, as someone must create the proofs, and those entities (like a rollup sequencer or the original user's client) will see the plaintext. But that is an easier situation to manage under GDPR. Those entities are clearly identifiable controllers (e.g., a company running a rollup can handle compliance for the data it processes in generating proofs). The base chain could remain relatively agnostic of personal details, acting almost like a notary that stamps "valid" on proofs. This could drastically simplify compliance on the base layer.

Fully Homomorphic Encryption to Run Private Computation

Fully Homomorphic Encryption (FHE) allows computation on encrypted data without decrypting it. An FHE-based VM (Virtual Machine) would let smart contracts execute on ciphertexts and produce encrypted outputs that only authorized parties can decrypt. The concept involves executing transactions in which inputs and potentially the contract state is encrypted.

Impact on privacy and roles:

- **In an FHE scenario:** operators do not see transaction details at all as they are encrypted. For example, if Alice wants to store a medical record on Ethereum via a smart contract,

she could encrypt the content and even encrypt any identifying tags. The contract's logic (if made FHE-compatible) could manipulate that data (verify a digital signature or update a record) in its encrypted form. The validator simply executes the EVM arithmetic on ciphertext. According to Zama's [description of their fhEVM](#), "the data (transaction inputs and on-chain state) is encrypted, no one can see it, not even validators." Validators process data that is unintelligible to them, thereby rendering it effectively anonymous from their perspective, in accordance with the standard set out in Recital 26

- **On-chain confidentiality with on-chain access control:** FHEVM approaches also introduce the notion that privacy can be programmable. Smart contracts can define who is allowed to decrypt what data. For instance, a contract could state that only a user with a certain private key (or a group of parties via threshold decryption) can decrypt the result of some computation. This aligns with GDPR's idea of access control and need-to-know principle. Data on-chain could remain encrypted to the public, but specific parties (perhaps the data subject themselves, or a regulator under certain conditions) hold decryption keys. Notably, solutions combine this with the use of threshold multi-party computation (MPC) for key management where the decryption key for the network's data can itself be split such that no single party (not even a validator or the contract deployer) can decrypt on their own. A group would need to collude to break privacy. This provides resilience as there is no central honeypot of keys.
- **GDPR compliance advantages:** With FHE, the entire network could be seen as processing personal data in an anonymized/encrypted form by default. If everything on-chain is encrypted by design, one might argue the blockchain is just a secure computation platform, not revealing personal information. In terms of roles, the application developer and user remain controllers (they decide to put the encrypted data and hold keys), whereas the node operators might be considered akin to data processors. It's worth noting that fully homomorphic encryption is very computationally heavy. Early implementations for Ethereum could only do a couple of TXs per second and needed specialized nodes. So, fhEVM is not yet ready to handle large scale permissionless network traffic now. But it can be introduced for specific use-cases (e.g., privacy-centric rollups or certain contract operations requiring confidentiality). Over time, improvements (and maybe hardware acceleration) could make it more viable. Meanwhile, combining FHE with zk-SNARKs might yield the best of both proving encrypted computations were done right toward so-called "verifiable computation". While still experimental, FHE represents the direction for maximizing privacy at the execution layer without sacrificing the verifiability and composability that make blockchains useful.

Trusted Execution Environments (TEEs) as Interim Confidential Computing

Trusted Execution Environments are hardware-based secure enclaves that allow code to run in isolation such that the host cannot see the data being processed, only the result. TEEs have been used in some blockchain contexts as a stopgap to achieve confidential computing before cryptographic methods are fully practical.

How TEEs help and their limitations:

- **Confidential execution:** With a TEE, transactional data could be executed inside an enclave where the input data is decrypted, do the computation, and produce an output and maybe a proof that the correct code was run, then re-encrypt outputs for the ledger. The operator system can't access the plaintext as it's shielded by hardware. This means the validator technically doesn't "see" the personal data in the clear, much like with FHE or ZK, the reliance is on hardware trust. For GDPR, if one trusts the TEE, one could argue the organization running the node isn't actually processing personal data in a meaningful way (the enclave is, and its a black box to them). However, legally the node operator is still in possession of the data (just in a protected memory).
- **Role of TEEs in decentralization:** TEEs could allow a form of consent-based data sharing. A user might allow the network to process their personal data only inside TEEs such that nobody can misuse it. It provides a compliance story for things like processing sensitive personal data (health, finance) on a blockchain by containing it in enclaves and only outputting minimal info. In the interim before ZK and FHE are fully mature, TEEs are a more immediately available technology.
- **Concerns and joint controllership:** One issue is that if many validators use the same TEE provider, there is a centralization and trust vector. Also, TEEs require validators to run special software, which does not fit with the inclusion principles associated with development of permissionless networks. While it could be a valid proposition for permissioned setups or subsets of operators. From a GDPR view, using TEEs doesn't remove obligations, but it can help with demonstrating compliance and It might reduce the likelihood of data breaches.
- **Interim solution:** The community has suggested using TEEs for things like sealing block content to mitigate MEV (the concept of encrypted mempools uses threshold encryption and potentially TEEs to manage the key shares). In practice, there might be hybrid approaches such as a block builder using a TEE to conceal the transaction ordering process, preventing even themselves from biasing it. This doesn't directly solve GDPR issues, but it does show how blockchain operators might incorporate confidential computing in parts of the pipeline to protect data. TEEs could be that pragmatic bridge until pure cryptographic methods (which don't require hardware trust) take over.

In summary, TEEs offer a “today” solution pathway to ensure confidentiality and process sensitive data on-chain today with a higher degree of privacy than vanilla execution. They align with GDPR’s “appropriate technical measures” requirement (Art. 32) by adding strong confidentiality safeguards. However, because they introduce trust in hardware vendors and have known vulnerabilities, they are considered an interim step. They do not fundamentally change the data controller picture (the node operator is still running the enclave), but they can reduce risk and exposure of personal data drastically.

Multi-Party Computation (MPC) for Key Management Randomization and Secure Processing

Multi-Party Computation refers to cryptographic protocols where multiple parties jointly compute a function over their inputs without revealing those inputs to each other. In the blockchain context, MPC techniques are being used for things like distributed key signing, threshold encryption, and collectively managing secrets. Two relevant applications are notable for GDPR compliance: 1) distributed validator key signing; 2) threshold cryptography for transaction privacy:

- **Distributed Validator (DVT) & Key Rotation:** Ethereum validators must hold private keys for signing blocks and attestations. MPC can split this key among multiple machines or operators (think of 4-of-7 threshold signing). This has implications for data protection. If a validator’s key is spread across parties, no single operator has the full “identity” of the validator. It can enhance security as there is less chance of single-point key compromise, less chance of unauthorized data manipulation. Also, it enables easier key rotation, in fact the group can generate new key shares periodically. For privacy, frequent key rotation means an external observer can’t easily link that the same validator is behind a set of actions over a long period, which reduces the personal data linkability. While this is more about network integrity, it indirectly supports privacy by making roles more ephemeral and pseudonymous.
- **Threshold Encryption for Mempool (MPC for privacy):** There are proposals where transactions are encrypted with a key that is shared among validators such that no single validator can decrypt the pending transactions, but once enough of them have seen it (or once a block is proposed), they collaboratively decrypt it. An example is [Shutter Network’s](#) idea: users encrypt transactions with a public key, validators have key shares, and only after a transaction is included do they combine shares to reveal it. This prevents front-running and means pending transactions (which may contain personal data) are not visible to everyone, only an authorized subset can decrypt, and only at the right time. MPC is at play to distribute trust. Under GDPR, this can be viewed as a technical measure limiting access to personal data to the minimum necessary and for the minimum time.

Overall, MPC reinforces privacy by ensuring no single point sees or controls all personal data and complements other data obfuscation techniques. Where ZK and FHE remove the need to trust others by using math, MPC removes single trust points by splitting tasks among several parties. In Ethereum’s context, MPC is already used in consensus randomness (distributed key

generation for randomness beacons) and is being considered in validator clusters. For compliance, it provides resilience, and it could be cited as an organizational measure (e.g., “we ensure by design that no single rogue node can expose user data, as decryption requires collaboration of multiple independent parties, making unauthorized disclosure extremely unlikely”).

Enshrined Proposer-Builder Separation (ePBS)

Moving to consensus-based protocol rules, [Enshrined PBS \(EIP-7732\)](#) represents an evolution of the currently implemented consensus logic which frames a proposal to build Proposer-Builder Separation directly into Ethereum’s core protocol. In the current post-Merge setup, an out-of-protocol form of PBS operates via MEV-Boost relays: block builders aggregate transactions including user activity and MEV strategies while proposers (validators) select from these built blocks to propose to the chain. Today this is done through a third-party relay network. Enshrining PBS means Ethereum itself would handle the role split without external relays. The design decouples block content creation from block confirmation. Builders create execution payloads (i.e., full blocks of transactions with a suggested ordering), and proposers (the validators assigned to the slot) simply select one of these payloads and commit it to the beacon chain. Importantly, in ePBS the proposer does not need to see the full transaction data at selection time, the protocol can use a blind bidding mechanism where builders provide a cryptographic commitment to a block and an offer (payment) for inclusion. The proposer picks the bid (usually the highest fee) and later the full block is revealed and attached.

This has a couple of privacy and compliance benefits:

- **Role isolation:** The block builder is the only party that decides which transactions (and thus which personal data) enter the block. The builder could be a specialized entity, potentially easier to subject to GDPR obligations. The validator (proposer) in ePBS does not influence individual transaction inclusion beyond choosing a builder’s bundle; and if the proposer is blind to the bundle’s content, they cannot discriminate or misuse personal data. This separation suggests the builder may be considered a data controller, as they decide to include transactions for profit, while the proposer functions more as a facilitator, potentially even a processor, without jointly determining the specific content.
- **Minimized data exposure:** In ePBS, because the proposer only sees a commitment (and perhaps some non-user bound metadata like total gas and fees, when choosing a block, transaction data is initially only in the hands of the builder. Other validators (attesters) will see the plaintext transactions slightly later, when the block is revealed for execution. But the critical point is that fewer parties see the unconfirmed transactions in real-time. Today, all validators and many observers can see every pending transaction in the public mempool, including ones with personal data. Enshrined PBS replaces parts of this with a private builder bidding process, reducing exposure. If a transaction contains personal data, only the builder may see it before it’s finalized in a block, lowering the risk of unlawful processing.

- **Accountability for builders:** Because ePBS is at protocol level, Ethereum could require builders to register a public key and perhaps even stake some ETH as collateral (the current draft requires builder signatures and slashing for malicious behavior). This has the side effect of identifying builders as distinct actors. From a GDPR perspective, this is useful: one could imagine large builders being legal entities that can be approached by regulators or data subjects. In contrast, today's monolithic miners/validators are numerous and often pseudonymous. Enshrined PBS reduces the set of actors who decide what data goes on-chain to a smaller, possibly more professionalized group (the block builders). This taxonomy aligns well with GDPR's concept of controllership. It becomes easier to say "builders determine the means of processing user transaction data." Meanwhile, validators become more like a passive consensus mechanism, not deeply inspecting or curating transactions.

From a compliance perspective, enshrined PBS could limit the scope of potential joint controllership. The proposer and builder have distinct functions and interact through a protocol-defined interface. Ideally, this could be coupled with arrangements: e.g., builders agreeing to certain terms (not to include illegal personal data, etc.) and proposers simply acting on whatever builders provide as long as it's valid. Thereby, a proposer might defend that they are not jointly determining the inclusion of any given personal data and simply run the blockchain protocol. Section 5 discusses how to handle the remaining controller relationships under PBS in more detail.

Attester-Proposer Separation (APS)

ePBS relies on a shift from architecture perspective which is determined from the [Attester-Proposer Separation \(APS\)](#) where even the consensus role of attesting/validating blocks is separated from the role of proposing block content. In Ethereum's current PoS, every validator does both: occasionally propose blocks and frequently attest to others' blocks. APS imagines splitting these such that some participants only attest, and a subset only propose. APS is intended to mitigate validator centralization caused by multi-block MEV. In practice, APS might involve mechanisms like Execution Auctions or Execution Tickets (research prototypes) where the right to propose a block's transactions is determined separately from the right to vote on the block.

- Attesters (Witnesses) under APS would only validate and attest that proposed blocks are correct. They would not themselves decide which transactions go into blocks. This means an attester's involvement with personal data is limited to verifying cryptographic validity and ensuring consensus rules. Because they don't choose the content, one could argue attesters do not determine the purposes of processing personal data; they serve the purpose of consensus integrity. This could reduce their liability as data controllers, positioning them more as process auditors than initiators.
- Proposers in an APS world become a more distinct class, potentially even a smaller set of actors who specialize in block content. If Ethereum, for example, auctioned off the right

to propose blocks in future slots (as in the execution auction concept), professional block proposers who plan out blocks ahead might emerge. These proposers would be the ones interfacing with transaction pools and thus deciding on including personal data. So, proposers would clearly be data controllers for the inclusion of transactions, like builders in PBS. Attesters, by contrast, might just endorse or reject the block.

The benefit of APS for GDPR alignment is that it could isolate data-heavy tasks to specific actors and keep most validators relatively agnostic about personal data content. If ~90% of validators only ever see proofs or summaries (in a hypothetical future where attesters rely on zero-knowledge proofs of execution), then those 90% might not be handling personal data in a meaningful way. The proposers who do handle plaintext transactions could be more tightly regulated or could implement privacy measures (like encryption or mixing) without involving all validators.

There is, however, an open issue. Even attesters in today's design do receive full block data to verify state transitions. APS by itself doesn't remove that necessity. It mainly tackles how block creators are chosen to prevent any single validator from exploiting consecutive proposal opportunities. But paired with other tech (like zk-SNARK execution), APS could lead to a scenario where attesters validate proofs rather than raw data.

In summary, APS aligns with a clearer role division: Proposers concentrate the "content determination" function (hence, likely controllers for content), attesters handle the agreement function perhaps more akin to a neutral network processing role. This sharp separation could simplify GDPR role allocation. One could imagine an APS future where there are registered block proposers (controllers) and a broader, more anonymous set of attesters who validate blocks without deep inspection, minimizing their involvement with personal data. That narrative would please regulators as it fences off who to hold accountable for data inclusion.

Maximum Extractable Value (MEV)

Maximal Extractable Value (MEV) refers to the ability of certain actors, primarily block builders and searchers, to extract economic value by arbitrarily ordering, including, or excluding transactions within a block. Originally studied as an economic externality of Ethereum's transparent transaction model, MEV has since evolved into a structural force within the execution layer. With the advent of modular networks transactions are sequenced, prioritized, and ultimately confirmed on-chain in the most efficient way possible.

From a data protection perspective, MEV poses several challenges to GDPR compliance in permissionless blockchain networks. The actors involved in MEV extraction, particularly those who gain privileged early access to transaction flows (such as RPC endpoints, mempool observers, and block builders), may process behavioral and transactional metadata that, when combined with wallet identifiers, rise to the level of personal data. This processing may result in user profiling, front-running behaviors, or economic discrimination, raising risks under GDPR principles such as fairness, transparency, and purpose limitation.

Efforts to modularize and decentralize this process such as the adoption of Proposer-Builder Separation (PBS) and its enshrined version (ePBS) represent a significant step forward. As recalled under PBS, block proposers no longer see transaction contents and instead select block payloads based on bids, effectively blinding them to any personal data. While this limits the data exposure of proposers and consensus participants, it does not resolve the problem at the level of builders and searchers, who remain exposed to raw transaction flow especially when submitted through public mempools.

To mitigate these risks and enhance both decentralization and GDPR alignment, several technical and governance mechanisms can be applied:

First, users and application providers can shift toward anonymized transaction submission, using tools like [Flashbots Protect](#), [Shutterized mempools](#), or [Dandelion++](#) style transaction broadcast mechanisms. These systems encrypt or obfuscate the transaction until it is included in a block, thereby denying searchers the ability to front-run or profile the sender based on mempool analysis.

Second, new models such as [MEV-Share](#) allow users to consent to the sharing of transaction flows with builders in exchange for a portion of the extracted value. When accompanied by appropriate disclosures, these schemes move MEV extraction from implicit, opaque surveillance to explicit, opt-in data processing, a critical shift from a GDPR standpoint.

Third, research into zkMEV seeks to enable block builders to commit to an optimal ordering strategy without revealing transaction content. Under this paradigm, builders submit zero-knowledge proofs that their ordering is valid and maximally valuable, while the actual transaction content remains private or encrypted. This removes the need for raw data access entirely, enabling execution ordering to proceed without personal data processing.

Lastly, it is worth remembering that Ethereum's roadmap strongly encourages decentralization of builder infrastructure, preventing the concentration of control over transaction flow within a few large actors.

Peer Data Availability Sampling (PeerDAS)

[PeerDAS](#) is Ethereum's approach to distributed data availability sampling, vital for scaling data throughput via sharding (part of the forthcoming "Danksharding" upgrade). The idea is that not every node needs to store every byte of every transaction; instead, data (especially large blobs of rollup transactions from L2s) can be split into pieces and dispersed. Each Ethereum node will randomly sample small pieces to get confidence that the whole data is available without downloading it all. In practice, Ethereum plans to have each full node responsible for archiving each block's data, and nodes gossip and request chunks from each other to verify availability. This improvement proposal will likely be introduced in the next upgrades called "Fusaka".

Privacy and role implications of PeerDAS:

- **Partial storage means partial knowledge:** With PeerDAS, any given node stores only a subset of the block data. Crucially, these pieces are not complete user records; they are fragments from an erasure-coded matrix, typically looking like random data unless combined with many other pieces. This means for a lot of nodes, the data they hold might not be personally identifiable at all. If one piece by itself cannot be used to identify someone (which is usually true for erasure-coded shards), then from that node's perspective they are not processing personal data, just random bits. Only by assembling enough pieces could the original transaction be reconstructed. This architecture could help argue that many nodes are processing anonymous data. Under Recital 26, if data is rendered anonymous in such a manner that the data subject is not or no longer identifiable, data protection principles don't apply. PeerDAS moves towards that for individual nodes (though someone could still aggregate data from multiple nodes to rebuild a block).
- **Data minimization and storage limitation:** from a compliance perspective, PeerDAS is an embodiment of data minimization; each node only holds what is necessary for security, not the whole dataset. It also aids storage limitation, since shards (blobs) are expected to be dropped after a certain period (e.g., after a few weeks or months once they are no longer needed for rollup verification). This ephemeral storage means personal data is not kept indefinitely by every participant, which is closer to GDPR's requirement that data not be kept longer than necessary. The EDPB draft guidelines themselves encourage designs where if long-term retention is not required, personal data should not be written permanently to the chain. PeerDAS aligns with that by making the availability of data time-bound and distributed.
- **Role of data availability nodes:** If Ethereum eventually separates data availability into its own layer or even separate "DA nodes", those nodes might be considered as a distinct category of service providers. They aren't deciding which data to put on-chain (they just receive blobs from block proposers and ensure availability), so their role could be closer to a processor or a neutral intermediary. However, since they are not under contract to a specific controller (in permissionless context), they might still be seen as independent controllers for the storage operation. It's ambiguous. But because the data is in unintelligible fragments, one could argue that DA nodes are not processing personal data in a meaningful sense (similar to how an email service that stores only encrypted emails without the key might argue it doesn't process the content of the personal data). Regulators might accept that argument if convinced that no realistic re-identification can occur at the node level.

Peer DaaS in the context of blob transactions are arranged into an "extended matrix," KZG-committed per cell, and nodes subscribe to random row/column topics, custodying only $\approx 6.25\%$ of each blob's fragments. Through probabilistic sampling, they gain high confidence in global availability without ever seeing full blobs.

Under GDPR, these archive nodes clearly act as data controllers/processors and must uphold data-subject rights (access, erasure) for any personal data they hold, whereas sampling nodes enjoy a reduced compliance burden thanks to their minimal, ephemeral custody. Together, blob transactions and Peer DAS fundamentally reduce the volume, lifespan, and exposure of on-chain personal data aligning Ethereum’s consensus layer more closely with GDPR’s mandates. Yet true compliance still hinges on encrypting or off-chaining sensitive metadata and operating a Personal Data Store that can honor erasure or consent withdrawal by cryptographically “shredding” decryption keys, leaving only irrevocable commitments on-chain. This layered approach of ephemeral blobs, sparse sampling, off-chain metadata erasure offers a practical path toward a GDPR-friendly permissionless networks

In essence, PeerDAS moves Ethereum toward a model where no single actor has the whole “personal data puzzle.” It’s a technological push toward anonymization. Even if the original data (for example an L2 transaction) contained personal information, once it’s split and distributed, each holder has just an anonymous slice. Only the combination, which is an emergent property of the network, is useful. This complicates the very notion of controllership – who is the controller of the combined data? Perhaps the L2 sequencer or the L1 proposer who originally posted it? The individual DA nodes might escape being labeled controllers of personal data because they never hold it in identifiable form.

In summary, the technical innovations in permissionless networks, from ePBS and APS restructuring roles, to PeerDAS minimizing data per node, to zk-SNARKs, FHE, TEEs, and MPC enhancing privacy are creating a future blockchain that is much more privacy-aligned and modular in responsibility. **These changes support a vision where the execution layer can be privacy-preserving and compartmentalized, while the consensus layer remains decentralized and secure without directly handling raw personal data.**

The next section will build on this by proposing a compliance framework that leverages these innovations to assign roles and responsibilities in a harmonized way.

4. Harmonized Data Protection Practices for Ethereum’s Modular Model

To operationalize GDPR compliance in a permissionless system, this framework sets out common practices that assign responsibilities across layers and prioritize anonymization at the execution layer, while preserving decentralized consensus. This framework builds on the GDPR compliance requirements discussed, integrating its future protocol development (taking Ethereum as a reference for EVM-compatible networks).

The goal is to ensure application-layer actors maintain clear controllership of personal data, while infrastructure (execution and consensus clients) handles only pseudonymized or anonymized data wherever possible. A key strategy here is using metadata erasure from apps connected to the network and cryptographic techniques so that data entering the

blockchain is either anonymized from the start or can be made anonymous by deleting off-chain linkages.

The table below presents a functional taxonomy for data management, illustrating how roles and responsibilities may shift with the implementation of privacy-enhancing technologies (PETs) that prevent reidentification and ensure effective data anonymization. “Before” assumes personal data is processed in the clear on-chain; “After” shows the scenario when only anonymized/pseudonymized data reaches the chain and off-chain metadata can be erased to break any link.

Table 1: Role Assignment and Data Anonymization

Participant	Without PET Implementation	With PET Implementation
Application Client (e.g., User Wallet)	Controller The wallet (or dApp front-end) determines what personal data is sent on-chain (e.g., address, memo field), and may log user metadata (IP, device). By choosing to publish data it “determines purposes and means.”	Data Subject All personal data is pseudonymized or encrypted <i>before</i> leaving the wallet. The client never publishes clear PII on-chain nor logs identifying metadata. It effectively holds the only decryption keys, and so other actors see only meaningless ciphertext. The wallet remains the data subject, but its own processing is purely local.
DApp Provider (Exchange)	Controller Designs smart contracts that collect or store personal data; decides which fields to record on-chain; holds off-chain databases linking addresses to real identities; must answer subject rights.	Controller Continues to decide “why and how” data is used but never puts raw PII on-chain. Implements off-chain storage, metadata erasure, and key rotation . On-chain it only submits hashes, commitments, or encrypted blobs . As the only party with the linking material or decryption keys, it retains controllership for compliance (DPIAs, consent records), while the network handles data only in an unreadable form.
RPC Endpoint	Controller or Processor If logging IPs or user agent data, it determines purposes (fraud detection, analytics), making it a controller; if purely forwarding without logs, it may be a processor.	Processor Configured as a non-logging, privacy-preserving relay . It merely forwards encrypted or pseudonymous transactions under the application client’s instructions, without independent use or storage of metadata. As such, it processes data <i>on behalf</i> of the dApp/user (processor role) and cannot repurpose or identify users.

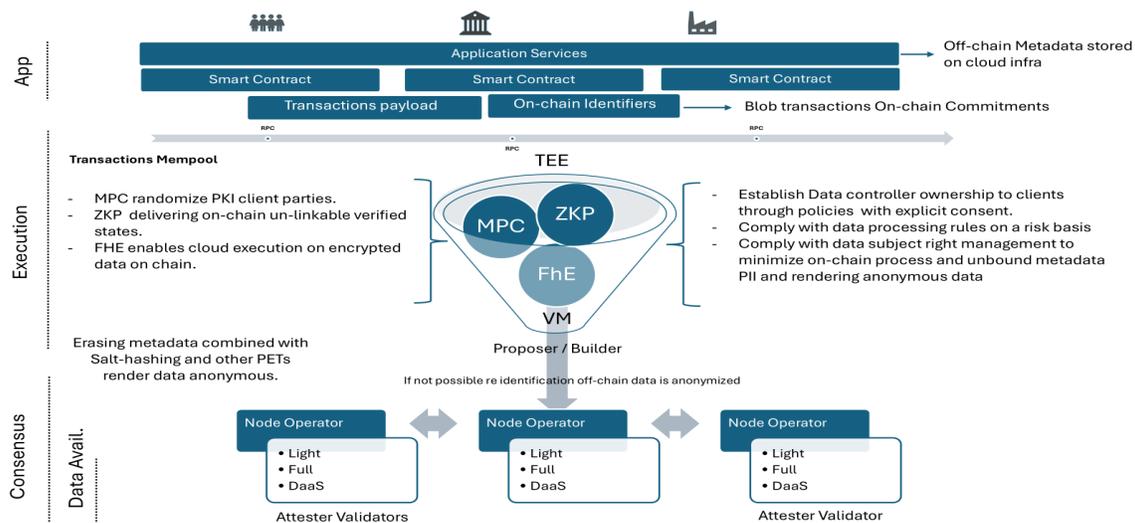
Execution Client (Mempool)	Controller Stores and gossips full transaction payloads (possibly containing PII). By choosing which peers to share with and how long to retain, it exerts influence on means of processing.	Processor Mempool holds only hashed, encrypted, or zk-proof representations. It cannot read or derive PII. Its sole function is to propagate valid payloads under protocol rules. Because it follows deterministic consensus logic without further discretion over content, it acts as a processor (or even pure conduit) under the dApp’s instruction, and no longer “determines purposes.”
Block Builder / Sequencer	Controller Selects, orders, and includes raw transactions (and thus PII) in a block. This discretionary ordering (for MEV extraction or fee maximization) is a clear “purpose and means” decision.	Processor Operates on pseudonymized/encrypted transactions or blinded bundles . Builders no longer see any clear PII and cannot re-identify users. Their role is reduced to packaging and forwarding payloads as instructed by the application layer, making them processors handling anonymized data without access to linking information.
Consensus Validator Proposer	Joint Controller Finalizes and replicates every block containing PII across all nodes. By storing and distributing data, validators effectively determine its continued processing and retention (storage limitation issue).	Neither Controller nor Processor under ePBS/zk-execution , validators only verify proofs (e.g. zk-SNARKs, commitments) and never access raw transaction contents. They operate on anonymous attestations. As they neither decide purposes nor see PII, they fall outside GDPR’s scope for on-chain data; their role aligns more with a neutral verifier .
Data Availability Nodes	Controller (possibly Joint) Stores full or re-assemblable transaction blobs. Even if shards are distributed, they are controllers for any PII they hold that can be reconstructed, and for retention decisions.	Processor Holds only erasure-coded fragments that, in isolation, cannot be reassembled or linked to an individual. These fragments are anonymous data under Recital 26. DA nodes simply ensure availability per protocol rules and prune shards after a defined window. They act as passive processors of non-personal data, and no longer bear obligations like erasure or access for data subjects.

The key theme is anonymizing data at the execution layer while preserving decentralized consensus. The goal is for transaction execution (where data is processed) to use encrypted or pseudonymous inputs, so the consensus layer only sees unintelligible data or abstract proofs, while still performing its role of agreeing on the next block.

The practices above leverage ZKPs and encryption, which allow consensus to validate *that* something is correct without seeing the underlying sensitive data. This preserves trustless, decentralized security, ensuring that all data remains verifiable by network participants without reliance on a centralized database, while maintaining privacy

By anonymizing data at the earliest point on the client side and throughout execution, the data reaching the consensus layer is effectively anonymous. This allows the consensus process to remain fully decentralized, enabling participation by thousands of validators without risking large-scale dissemination of personal information.

Figure 3 describes a neutral, multi-layered system architecture framing an application environment, a smart contract layer, and a modular, permissionless blockchain network adapting PET (Privacy Enhancing Technologies) in the execution, consensus and data availability layers to reduce data management frictions to comply with GDPR.



Technical and Operational Best Practices for Data Risk Management

To support the approach outlined above, this section sets out practical measures that actors at each layer should adopt to ensure compliance within this model:

- Application Layer and Privacy by Design:** Developers of smart contracts and dApps should architect their systems such that personal data never appears in plaintext on-chain. Concretely, this means using off-chain storage for any readable personal details (names, emails, documents) and only storing a hash or reference on-chain. If on-chain functionality is needed (e.g., verifying a credential or executing a logic on personal data), use privacy-preserving techniques: zero-knowledge proofs to prove a statement about the data without revealing it, or encrypt the data and use contracts that operate on ciphertext (via TEEs or zk-SNARK-friendly circuits). Pseudonymize user identities – e.g., have users use one-time wallet addresses per interaction, or implement Stealth addresses (where a user’s public address is not reused and cannot be linked easily). Ensure that any off-chain

personal data repository is under the control of a known entity (the DApp provider or the user themselves) who can fulfill data subject rights. For instance, if a user wants to delete their account, the provider can delete the off-chain records and publish a transaction that nullifies the on-chain link (perhaps by overwriting the on-chain hash with a random value or updating a state to indicate erasure). This would implement the recommendation “remove off-chain data so on-chain data is no longer personal” in practice. Additionally, application providers should maintain clear privacy notices informing users that their data will be stored off-chain and only hashed on-chain and explaining any residual risks (like the impossibility of removing the hash but noting it’s useless without the data).

- **Off-chain Governance & Legal:** Because public blockchains lack formal contracts between participants, alternative governance tools are needed. For Ethereum, one practice could be developing a code of conduct or community charter that large infrastructure providers (mining pools, staking pools, layer-2 operators) voluntarily sign, committing to GDPR principles. For example, builders and validators could pledge not to deliberately include certain categories of personal data (like clear-text personal identifiers) and to cooperate in network-wide erasure efforts if needed (like blacklisting certain data if a court orders removal). Technically data remains on-chain, they could cease processing it in UIs and future transactions. While not legally binding in the traditional sense, such a code could later be material for regulators to show an industry standard. Protocol can publish transparency reports, handle communications for data subjects. This addresses the EDPB’s note that a central contact point is needed even in distributed systems.
- **Data Storage:** Client developers (e.g., Prysm, Geth, Nethermind, etc. in the Ethereum community) may encourage practices of development for features that support GDPR compliance. One key feature is history pruning, clients should, by default, prune personal data after it’s no longer needed for consensus. E.g., [EIP-4444](#) already encourages pruning old receipt and state data beyond different periods of time. This aligns with storage limitations and can be justified under Art. 17(3) if the data is needed for a period for the service, after that period it can be erased. Light node operators should participate without storing full historical data, while confining storage of data to full node operators or DaaS providers.
- **Metadata Erasure Processes:** A formalized practice should be established for metadata erasure. This means whenever a user invokes their right to erasure via the DApp, the following happens: (1) the (DApp) deletes the personal data from their off-chain storage. (2) They broadcast a transaction that either deletes a pointer on-chain (if the smart contract allows, e.g., by nullifying the storage slot or burning an NFT representing the data) or, if deletion on-chain isn’t possible, publishes a proof that the data was deleted off-chain and perhaps a random noise to replace the original hash (making it computationally infeasible to recover the original value even from the hash). (3) Other nodes, upon seeing this, can log that the data was erased and should not be processed further. Because full deletion on-chain isn’t possible, making the on-chain data irreversibly unlinkable is the target. For example, if an on-chain record is $\text{hash}(\text{name}, \text{secret}) = 0xABC$, and the user

wants erasure, the controller can delete name off-chain and publish a new secret that turns the on-chain value into a random number not tied to the name anymore (eg. rotating a key). This may act as functional erasure. The EDPB explicitly approves the approach of off-chain deletion such that on-chain data loses personal reference, so formalizing this in operational playbooks is crucial. DApp providers should also include in their privacy policies that because of blockchain's nature, some minimal hashed data will remain on-chain, but it is no longer personal once we remove the link.

- **Encryption & Key Management:** A robust practice is to use encryption for any personal data that must be recorded in some form on-chain. For instance, if storing a medical record hash, maybe encrypt that hash with a key that only authorized parties have. That way even the hash isn't usable by everyone (though security of such a scheme depends on key distribution). Another example is to use zk-SNARKs to prove relationships instead of publishing data. If a DApp needs to show Alice is over 18, it should not record her birthdate; instead, Alice provides a ZK-proof of age >18 and the contract just logs "proof verified" with no personal data. If later Alice wants that removed, the log "proof verified" is not personal to begin with (it could refer to anyone). In essence, design transactions to carry claims or proofs, not raw personal attributes.
- **Off-chain Data Repositories Records:** Organizations executing parts of this system should maintain a data inventory that clearly delineates what data is on-chain (and in what form: hashed, encrypted, pseudonymous) versus off-chain. For off-chain personal data, normal GDPR procedures apply. For on-chain data, they should document the process for data pseudonymization and justify, per Recital 26, that identifiability is mitigated to the extent possible. This documentation (perhaps as part of their Article 30 record of processing) will serve them well if a regulator inquires so that they can show for instance "User account info is stored in our database for 30 days, and only a userID hash goes on-chain; if user deletes account, database entry is wiped and on-chain hash can no longer be tied to anyone."
- **Network-level Collaboration:** Since permissionless communities are decentralized, achieving data compliance is a shared responsibility to end users. Participants should collaborate on standards. For example, a standard for tagging data that is subject to erasure, maybe a smart contract interface that, when implemented, allows data controllers to signal erasure of associated off-chain data. Or for instance as incident response: if some personal data (like someone's private info) gets on-chain improperly, the community could coordinate to at least stop its propagation at the application layer (e.g., block explorers censoring it, wallets not displaying it). While the data technically persists, this limits further processing, which is part of compliance. (GDPR doesn't require impossible things like deleting from history, but it does require not continuing to actively process and disseminate unlawfully posted data).

- **Consent and Legitimate Interests:** Application layers should carefully choose a legal basis for any on-chain processing of personal data. Often, they will use consent (e.g., a user agrees to publish a piece of data via the blockchain for a certain purpose) or legitimate interests (the decentralized architecture is necessary for the service, and data minimization is applied). If consent is used, one must allow withdrawal. That ties in with erasure. If a user withdraws consent, the controller should remove their data off-chain and take steps to nullify on-chain references. It might also include informing the user that while we cannot scrub history, we ensure your data is no longer accessible in a functional way in the system going forward. If legitimate interest is the basis, the controller should have done a LIA (Legitimate Interests Assessment) weighing the necessity of using a blockchain vs alternatives (e.g., cyber resilience), and showing how they minimize impact. Generally, perspective we tend to agree that a consent basis approach covers most of the market uses.

As technology evolves, the practices can be updated, perhaps moving from “hashed off-chain data” to “encrypted on-chain data with threshold decryption” as a new standard. The framework should be flexible, always aiming to maximize the proportion of data that is anonymous from the network’s viewpoint. Over time, the ideal end-state is where the execution layer is almost entirely a privacy-protecting black box with user-granular access control, and the consensus layer is just securing it without ever seeing personal info. At that point, compliance is much easier. Most of the GDPR’s obligations concentrate on the small set of entities who hold decryption keys (application providers or users themselves), and the network’s role becomes closer to that of an information society service transmitting encrypted data.

This framework is intended to be considered as a pathway of reconciliation with data management regulation to keep personal data at the edges (user and application controller) and off the ledger, while making the ledger hold only what is strictly necessary and in a form that reveals no identity.

Doing so can ensure that the application layer assumes data controllership and can execute GDPR obligations, while the execution and consensus layers serve as data processors or neutral infrastructure handling anonymized data. In effect, preserving technical decentralization but centralizing accountability to the appropriate parties who hold market and user protection requirements.

This harmonized approach aims to uphold the spirit of GDPR, protecting individuals without compromising the utility of permissionless blockchains.

5. Remaining Open Topics & Questions

Even if a future pathway is foreseeable, a few open questions remain to be determined and require further clarification at the intersection of permissionless network design and GDPR compliance. The following outlines key open questions and provisional recommendations for

addressing them in the interim, acknowledging that best practices and potentially formal guidance may evolve over time.

Evolution of executing and validating blocks: toward network specialization avoiding or limiting Joint Controllership:

How can data management specialization be ensured in permissionless networks?

Following future blind and validation of block rules, proposer-builder and attester-proposer separation, who is the controller for data in a block, the builder, the proposer, or both? Formally segregate the roles via contract and policy. For example, if a block builder service is used by validators, there should be an open-source terms-of-service for network developers to accept specifying that the builder is responsible for the content of blocks (thus the data controller for that processing), and the proposer is simply transmitting it.

In ePBS, the protocol could be designed so that proposers are blind to transaction content until after inclusion, reinforcing that they do not determine the purposes of that data processing. This could help argue that proposers are not joint controllers but rather rely on the builder's representations. In practice, until such clarity is recognized by regulators, builders and proposers should perhaps be considered joint controllers by default and implement agreements accordingly (per Art. 26 GDPR). For instance, a large staking pool using a specific builder could sign an agreement on how to handle any personal data in blocks, with the builder taking on data handling duties (like filtering certain data if needed). As the ecosystem matures, a more template approach can be taken; builders as controllers, proposers as technical intermediaries.

Attester-Proposer Separation (APS) further limits any single validator's influence. To avoid joint control among a swarm of attesters, the protocol could ensure attesters only validate predefined rules (not injecting any new purpose), making them more like an audit function. Nonetheless, given the uncertainty, all parties should document their processing and the rationale for their role interpretation (referring to EDPB guidance) to defend against mischaracterization.

i. Data Retention and Deletion for End-Clients:

How can end-users or clients manage proper data deletion in a system that never forgets? We should empower end-clients with deletion tools for the data they control.

For users, this means wallets should allow exporting and deleting one's keys and locally cached data. If a user wants to leave no trace, the wallet can help by removing their account data and perhaps sending a transaction to revoke consent for any ongoing data processing (for instance, calling a smart contract function that signals the user no longer consents to their data usage, maybe relevant in contexts like a social dApp where their content remains visible, a revoke signal could trigger front-ends to hide it).

For node operators, networks could implement pruning modes. A node could run in GDPR-friendly mode where it does not store data beyond what's necessary for the current state. Providers might

have to implement data retention limits. If an end-user posted something and then deleted it off-chain, indexers should update to reflect that (maybe through flagging systems).

Off-chain repositories (like user profile databases, KYC records linked to addresses) must follow classical retention periods (e.g., delete KYC after 5 years as AML laws allow, etc.).

End-users should be educated that while their on-chain transactions can't be erased, they can take steps like rotating addresses and avoiding exposing new personal data to minimize long-term exposure.

In this context, front-end services will play a big role as they determine what data is accessible to a broad public. If they honor deletion requests by not displaying certain on-chain content, that effectively “deletes” it from practical view (even if still on-chain for those digging). So, a provisional measure may be to get major blockchain explorers and DApp UIs to commit to a “erase on request” policy for personal data when legally valid requests come (through some verification), they hide or obfuscate that data in their interface. The data remains on-chain for integrity but is no longer actively processed in mainstream services.

ii. **Metadata Erasure as Anonymization:**

Can deleting off-chain mappings truly satisfy GDPR's anonymization criteria, and how to ensure that?

In summary, yes, metadata erasure may function as GDPR-compliant anonymization if done thoroughly, but providers must consider all likely data sources in the system network's outreach. When using metadata erasure, inform the user that the likelihood for the data to be used to re-identify are eliminated. For example, if the link between an Ethereum address and a user's identity was held only by a certain DApp, and that DApp deletes it, then demonstrate that no other party can easily re-link (perhaps the address was unique to that DApp context and user). Controllers should implement cryptographic features described so that even if someone got hold of the original data, recomputing the hash isn't straightforward without that secret and if the secret is destroyed, re-identification becomes infeasible.

iii. **Off-chain records for public interest or judicial needs:**

Many blockchain systems require keeping identity info off-chain for regulatory compliance (AML, CFT). Can these be kept separate and not violate GDPR, and can Article 23 GDPR (which allows restriction of rights for legal obligations) be invoked?

Yes, separate them and clearly label their purpose. For instance, an exchange might maintain a private database linking wallet addresses to real identities for AML/travel rule purposes. This database should be logically and physically separated from the blockchain transaction processing system. It should only be used when required for public interest (e.g., responding to law enforcement or doing fraud checks), and not integrated into normal on-chain operations. By doing

so, the on-chain network can be treated as pseudonymous for most actors, with only regulated entities having the mapping. Those entities can rely on legal obligations (AMLD laws, etc.) under Article 23 GDPR to refuse certain rights like erasure – e.g., they can legally say “we cannot erase your KYC info after 1 year because anti-money-laundering law mandates 5-year retention.” Article 23 allows derogations for purposes like prevention/detection of crime, so maintaining an off-chain identity repository for financial compliance is lawful, but it must be proportionate and secure. The blockchain itself remains uninvolved in personal data processing beyond pseudonyms.¹

It is also worth considering privacy-enhancing compliance such as zero-knowledge proofs for AML checks to prove the user isn't on a sanctions list without revealing who they are. This way even the off-chain repository can store minimal data. Many of these issues such as joint controllership, handling of erasure, lawful retention will benefit from continued dialogue between the industry, the EDPB, and legal experts. It is recommended the EDPB and national DPAs further engage in an open forum on this topic, monitoring the evolution of such technology and industry practices. Proactively addressing these open questions can help create an environment where permissionless blockchain innovation and strong data protection not only coexist but mutually reinforce one another, ensuring that decentralization supports user autonomy and privacy principles shape the evolution of blockchain architecture.

6. Policy Recommendations: Toward a Modular, Privacy-Preserving and Legally Coherent Blockchain Framework

To bridge the evolving technical architecture of permissionless blockchains with the legal obligations enshrined in the GDPR, we propose the following policy recommendations. These are designed to maximize legal clarity, enforceability, and technical feasibility, particularly in reference to permissionless blockchain networks which embody a modular set up. Each recommendation addresses a specific intersection between technical functionality and data protection law, with the intent of supporting the EDPB's future guidance.

These provisional recommendations offer a starting point for aligning industry and legislators on this sensitive issue, by assigning responsibility appropriately, applying technical measures to mitigate risk, and leveraging legal tools to balance privacy with other obligations.

¹ Practically, to achieve this separation, companies should implement strict access controls and data silos. They should also be transparent: include in user agreements that certain off-chain identifying information is kept as required by law and is not part of the decentralized protocol. Data subjects should know that their rights over that dataset might be limited (e.g., they can't demand deletion before legal retention is over). The recommendation is to use Article 23 narrowly – only for specific law enforcement/AML contexts, not as a blanket to deny blockchain users their rights.

1. Support role attribution across execution, consensus, data availability operators of blockchain networks in EDPB Guidance

The EDPB should explicitly acknowledge the layered nature of modern blockchain networks and adopt a functional role attribution model that reflects this modularity. Roles should be analyzed not in monolithic terms (e.g., "node operator") but as context-dependent based on:

- Whether an actor determines *what* data is processed (execution),
- *How* it is validated or retained (consensus and availability), and
- Whether the actor can *access or re-identify* personal data (e.g., builders vs. attestors).

The model should adopt a “purpose + control over means” matrix for each layer, allowing for roles such as controller, joint controller, or technical intermediary (a non-controller role) depending on the actor's functional influence.

2. The EDBP should proactively engage with industry to further investigate metadata erasure as a functional path to on-chain data anonymization

Concretely, work in this regard may focus on Off-chain data deletion rendering on-chain data anonymized and meaningless for re-identification purposes. This approach aligns with Recital 26 and offers a privacy-by-design compliance route in systems where technical immutability makes physical erasure impossible.

3. Clarify Role Differentiation and Conditional Controllorship Across Blockchain Layers

As permissionless blockchains evolve toward layered modularity, separating execution, consensus, and data availability into discrete technical domains, data protection law must reflect this functional separation with clear role attribution under GDPR. Specifically, it is essential to distinguish between actors that determine the purposes and means of processing personal data (data controllers) and those who merely process data without access or discretionary influence (processors or neutral intermediaries).

We recommend that:

- Where such distinctions cannot be strictly maintained, for example, when validators influence inclusion policies, perform MEV strategies, or jointly govern protocol upgrades, joint controllership arrangements may need to be considered. However, these arrangements must reflect *real-world influence*, not theoretical protocol permissions.
- Data minimization, metadata erasure, and pseudonymization practices should be formally integrated into this analysis. If personal data becomes irreversibly anonymized from the perspective of a given actor, that actor should fall outside the GDPR's scope for that processing activity.

4. Encourage the Use of Zero-Knowledge Proofs, Multi Party Computation (MPC) and FHE for On-Chain Minimization

The EDPB should encourage adoption of provable privacy and zero trust design architectures that minimize exposure to personal data. Technical safeguards such as Trusted Execution Environments (TEEs), Fully Homomorphic Encryption (FHE), or Zero-Knowledge Proofs (ZKPs) should further inform role attribution. Where such technologies demonstrably prevent data visibility or discretionary access, actors executing data logic should not be presumed controllers, and liability should shift to those entities (e.g., dApp developers, client-side agents) who configure and hold keys to decrypt or link data.

This aligns with the data minimization principle and reduces the scope of controllership to only those actors who access decrypted data.

5. Clarify Personal, Pseudonymous, and Anonymous Data per Network Layer

Further guidance should explicitly distinguish between:

- Personal data (e.g., plaintext names, identifiers),
- Pseudonymous data (e.g., public addresses linked to off-chain KYC),
- Anonymous data (e.g., erasure-coded fragments or encrypted blobs with no access to reassembly).

Further investigation into new types of transaction payload such as “blob transactions” should be required, especially highlighting their compliance with on-chain data minimization and their functional purpose regarding the process of anonymized data within the consensus layer. Following this logic, nodes holding only anonymous shards (e.g., in PeerDAS) should be exempt from GDPR obligations related to data subject rights. This avoids overburdening technically passive participants and reflects Recital 26’s threshold of identifiability.²

6. Support a Data Governance Layer Through Community Standards and Registries

The EDPB and standardization bodies should proactively engage in open forums with industry and Community based initiatives operating permissionless networks to support the drafting of harmonized codes of conduct to establish a standardized way to control and process data across blockchain. These instruments would foster a “soft compliance perimeter” around high-capability actors while avoiding over-regulation of pseudonymous participants.

² The key message is that permissionless networks with a modular design enable specialization among participants. Building on this, a strategy to ensure anonymous data on-chain can be implemented at the execution layer through metadata erasure, while still allowing relevant data to be stored off-chain in a separate database for public interest purposes. At the consensus layer, sharding can further support this goal by distributing data processing in a way that functionally erases it across validators. Once data is published on-chain using this architecture, it becomes effectively unusable for re-identification. This approach supports compliance with GDPR provisions related to user data rights and data minimization.