applied sciences

MDPI

Review

Blockchain Technologies for eIDAS Trust Service Providers: A Review of Architectures, Use Cases, and Emerging Trust Frameworks

Andrei Brînzea [1,2,†], Emil Bureacă [1,*,†], Răzvan-Andrei Leancă [1], Ștefan Arseni [3,4] and Florin Pop [2]

[1] Faculty of Information Systems and Cyber Security, Military Technical Academy "Ferdinand I", 050141 Bucharest, Romania; andrei.brinzea@mta.ro (A.B.); andrei.leanca@mta.ro (R.-A.L.)
[2] Faculty of Automatic Control and Computers, National University of Science and Technology Politehnica, 060042 Bucharest, Romania; florin.pop@cs.pub.ro
[3] Faculty of Electronics, Telecommunications and Information Technology, National University of Science and Technology Politehnica, 061071 Bucharest, Romania; stefan.arseni@certsign.ro
[4] Cybersecurity Solutions B.U., CertSIGN S.A., 011356 Bucharest, Romania
* Correspondence: emil.bureaca@mta.ro
† These authors contributed equally to this work.

Abstract

This paper presents a comprehensive review of existing research on the integration of blockchain technologies with the trust service ecosystem governed by the Electronic Identification, Authentication and Trust Services (eIDAS) Regulation of the European Union (EU). While Public Key Infrastructure (PKI) and electronic signature (ES) systems deployed under eIDAS provide strong cryptographic guarantees, standardized protocols, and cross-border legal recognition, their operational model remains largely centralized, concentrating trust in supervised authorities and service providers. This centralization raises concerns related to transparency, auditability, and resilience that blockchain, with its decentralized consensus and immutable distributed ledgers, has been increasingly explored to address. This review covers the most relevant application domains in which blockchain has been proposed as a complementary layer for Trust Service Providers (TSPs): certificate lifecycle management, remote signature services, signature preservation, signature validation, timestamping, content provenance and authenticity, and the European digital identity (EUDI) Wallet ecosystem. For each domain, this paper analyzes how blockchain can strengthen auditability and distributed trust while preserving the interoperability, legal assurance, and standards compliance required by eIDAS and ETSI (European Telecommunications Standards Institute) frameworks. A quantitative comparison of latency, throughput, and operational costs between blockchain-augmented and traditional architectures is provided, together with a technology maturity classification for each application domain. Finally, the paper identifies current limitations, including scalability, regulatory alignment, privacy constraints, and the absence of production-scale pilot data, and outlines open research challenges for the adoption of blockchain in regulated digital trust services.

Keywords: blockchain; public key infrastructure; electronic signatures; trust service providers; eIDAS; digital trust; certificate management; distributed trust

Check for updates

Academic Editor: Luis Javier Garcia Villalba

Received: 12 March 2026
Revised: 9 April 2026
Accepted: 9 April 2026
Published: 15 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The Electronic Identification, Authentication and trust Services (eIDAS) Regulation (Regulation (EU) No 910/2014) [1] establishes the legal and technical framework for electronic signatures, trust services, and cross-border identification within the EU. Under this framework, Qualified Trust Service Providers (QTSPs) deliver services with full legal recognition, including qualified electronic signatures (QESs), seals, time stamps, and registered delivery, subject to conformity assessment, supervision, and inclusion in national Trusted Lists [2]. The operational model underlying these services relies on PKI, in which Certification Authorities (CAs) issue, manage, and revoke digital certificates, and TSPs deliver cryptographic assurance through standardized protocols and formats defined by ETSI [3,4]. This model is mature, interoperable, and legally established, but it is inherently centralized: trust is anchored in designated authorities, audit evidence is held by individual providers, and the transparency of operational events depends on the internal practices of each organization [5,6].

Blockchain and distributed ledger technologies offer an alternative trust paradigm based on decentralized consensus, append-only data structures, and cryptographic integrity guarantees that do not depend on a single trusted entity [5]. In the context of trust services, these properties have been explored to address specific limitations of centralized PKI: the opacity of certificate lifecycle events, the lack of independent auditability for signing operations, the dependence on single providers for revocation information, and the difficulty of verifying the integrity of long-term preservation evidence [7–9]. A growing body of academic work has proposed blockchain-augmented architectures for certificate management [9–11], remote signing audit trails [12], signature preservation [13–15], signature validation [16], and timestamping [17–19]. In parallel, the European Blockchain Services Infrastructure (EBSI) [20] has introduced a permissioned ledger operated by EU Member States to support trust registries, decentralized identifiers (DIDs), and verifiable credential (VC) revocation within the emerging EUDI Wallet ecosystem mandated by eIDAS 2.0 (Regulation (EU) 2024/1183) [21].

Despite this progress, the available literature remains mostly organized around individual trust service domains. Existing studies tend to focus on a single area, such as certificate transparency, timestamping, or identity management, rather than examining together the cross-cutting patterns, regulatory constraints, and practical limitations that arise across the broader ecosystem of eIDAS-governed trust services.

This paper addresses this gap through the following contributions:

- A systematic review of blockchain integration across seven eIDAS trust service domains: certificate management, signature services, signature preservation, signature validation, timestamping, content provenance and authenticity, and the EUDI Wallet ecosystem;
- An analysis of the legal standing, certification requirements, and regulatory audit obligations for blockchain-based evidence within the eIDAS framework;
- A quantitative comparison of latency, throughput, and operational costs between blockchain-augmented and traditional trust service architectures, grounded in published benchmarks, platform documentation, and empirical measurements;
- A technology maturity classification for each application domain, ranging from conceptual proposals to production-level deployments.

The remainder of the paper is organized as follows. Section 2 presents the technological, legal, and standardization context. Sections 3 through 6 review blockchain integration in certificate management, signature services, signature preservation, signature validation, and timestamping, respectively. Section 7 covers content provenance and C2PA alignment. Section 8 addresses the EUDI Wallet and EBSI ecosystem. Section 9 provides the quan-

titative performance and cost analysis. Section 11 summarizes the state of the art, and Section 12 presents conclusions and future work.

2. Technological Context

Any assessment of blockchain in EU trust services must be grounded in the regulatory and technical architecture that governs electronic signatures and trust services in the EU. The eIDAS Regulation (Regulation (EU) No 910/2014) establishes both a legal framework for electronic signatures and an institutional trust model in which supervised TSPs. In particular, QTSPs are accountable for delivering services that carry legal effects [1]. In practice, the cryptographic primitive (a digital signature) is only one component of an end-to-end evidentiary process that also includes identity and certificate lifecycle management, reliable time, revocation state, governance and audit controls, and interoperability constraints across Member States.

2.1. eIDAS Trust Services and the Operational Baseline for TSPs

eIDAS distinguishes between simple electronic signatures, advanced electronic signatures (AdESs), and qualified electronic signatures (QESs). A QES is an advanced signature that is based on a qualified certificate and created by a Qualified Signature Creation Device (QSCD), and is performed under the sole control of the user over the private key. QES benefits from the strongest legal presumption within the EU, including mutual recognition in cross-border settings [1]. These legal properties impose engineering constraints: TSP systems must deliver not only cryptographic correctness, but also demonstrable compliance with organizational and technical controls (security management, incident handling, personnel and physical security, audit logging, and continuity). Part of these requirements are captured by ETSI's general policy requirements for TSPs [2]. Specific trust services refine these general requirements. For example, EU qualified certificates are governed by dedicated ETSI policy and security requirements for qualified certificate issuance [3]. Likewise, interoperable creation and validation of advanced signatures rely on ETSI procedures for creating and validating AdES signatures [4].

The revised framework under eIDAS 2.0—Regulation (EU) 2024/1183—updates eIDAS to establish a European digital identity framework and to enable new high-assurance architectures (including wallet-centric flows) while maintaining the supervised trust model [21]. Of particular relevance for this review, the standardization ecosystem for remote signing has evolved in parallel: ETSI remote-signing requirements explicitly reference the updated regulation and clarify how remote QSCD and server signing services fit into an eIDAS-compliant environment [22].

2.2. Remote QES and Sole Control in Service-Based Signing

Remote signing replaces the traditional assumption that signature creation data is held in a device under the signer's physical control with a service model in which keys are held and protected by a remote Signature Creation Device (SCDev)/QSCD operated by a QTSP. In this setting, the primary technical challenge is to preserve the signer's sole control and intent while the private key is used inside a server-side protected environment. ETSI TS 119 431-1 addresses this by specifying policy and security requirements for TSPs operating a remote signature creation device, including requirements for authorization, auditing, and protection of signature creation data [22]. The standard aligns with the reference architecture for Trustworthy Systems Supporting Server Signing (TW4S), which decomposes server signing into a Server Signing Application (SSA) using an SCDev, and introduces the Signature Activation Module (SAM) as the logical component that enforces authorized use of keys under sole control [23].

In TW4S-aligned designs, authorization is typically expressed through Signature Activation Data (SAD): a signed or authenticated authorization object that binds user authentication and transaction context to a narrowly scoped signing operation. This binding is commonly discussed using sole control assurance level concepts (e.g., stronger assurance when the authorization is cryptographically bound to the specific document hash to be signed) [23]. Remote qualified signing, therefore, becomes a controlled workflow rather than a single cryptographic operation: enrollment and identity verification, certificate issuance, secure key generation and storage, signing authorization, signature creation, and evidence generation must be engineered and audited as a coherent chain of controls [1,2].

Recent academic work illustrates how these requirements translate into practical architectures. Leitold and collaborators discuss remote qualified signatures and certification considerations, highlighting that remote signing is enabled under eIDAS provided that adequate controls and assessments exist, and that certification practices for remote QSCDs are a central practical bottleneck [24]. A device-centric alternative is explored by Scurtu and Patriciu, who emphasize how trust and usability tradeoffs can be addressed by carefully placing authentication and user interaction close to the signer while leveraging service components for scalable key protection and policy enforcement [25].

2.3. Interoperability: AdES Formats, Remote-Signature Protocols, and Application Embedding

Electronic signing must remain interoperable with existing verification tooling and eIDAS-recognized signature formats. ETSI EN 319 102-1 provides procedures for the creation and validation of AdES signatures and frames how common formats (CAdES, XAdES, PAdES and ASiC) are created and assessed in a standards-consistent way [4]. ETSI TS 119 432 specifies protocols that support these integrations in remote signing workflows, including qualified signatures and seals [26].

In parallel, the Cloud Signature Consortium (CSC) specifications have become a de facto interoperability layer for remote signatures, defining a standard API model for remote signature service providers and signature applications [27]. The practical relevance of this API layer is reflected in implementation-focused research. Aciobăniţei et al. discuss lessons learned when implementing an Android client for CSC-based services, highlighting integration and security engineering challenges in mobile deployments [28]. Their SaaS-oriented work demonstrates how qualified signatures can be embedded into productivity platforms (e.g., signing workflows for cloud documents), illustrating the trend toward application-embedded trust services [29]. Ruica et al. survey available cryptographic libraries for cloud qualified signatures, documenting the engineering reality that standards compliance must be matched by careful selection and composition of cryptographic and document-format tooling [30].

More recent proposals extend this engineering line by addressing privacy, modularity, and user authorization. CISRES proposes a comprehensive and privacy-aware remote QES approach, framing remote signing as a multi-component architecture that must meet both compliance and data-minimization constraints [31]. Brînzea and Aciobăniţei propose a helper module intended to streamline remote qualified signature integration across applications, reflecting a broader move toward reusable building blocks for remote signing clients [32]. At the infrastructure layer, the concept of cryptography API proposes cloud key storage and cryptographic operations as a service, which can be viewed as an enabling substrate for server signing systems when combined with policy-enforced authorization [33]. Finally, lightweight out-of-band authorization mechanisms (e.g., QR-code-based authentication) can be integrated as user-consent channels or second factors in authorization flows, supporting the sole control objective under remote operation [34].

2.4. Positioning Blockchain Within the Trust Service Technology Stack

Blockchain and distributed ledger technologies can be viewed as replicated, append-only data structures maintained by distributed consensus. In permissioned deployments, a governance layer and controlled validator membership replace open participation while preserving tamper-evident sequencing and auditability [5,6]. For trust services, this is most relevant as an architectural option for strengthening integrity, transparency, and post-hoc auditability of operational events (e.g., certificate lifecycle events, signature transactions, preservation renewals) rather than as a replacement for eIDAS-defined qualified services. A useful precedent is certificate transparency, which uses append-only logs and Merkle-tree proofs to improve ecosystem auditability for certificate issuance; TSP-oriented blockchain proposals often pursue an analogous goal for trust service evidence [7].

This complementary positioning has important implications for legal validity, certification, and regulatory oversight. Under the eIDAS Regulation, elevated legal presumptions, such as the equivalence of QES to handwritten signatures (Article 25(2)), the presumed accuracy of qualified time stamps (Article 41(2)), or the integrity guarantees of qualified registered delivery (Article 43(2)), are reserved exclusively for services delivered by QTSPs that have undergone conformity assessment (Article 20) and are listed in a national Trusted List [1]. Blockchain records do not satisfy these conditions on their own: a hash anchored on a public or permissioned ledger is not a qualified electronic time stamp, because it is not issued by a supervised QTSP operating under ETSI EN 319 421 [35] with a conformity assessment against EN 319 401 [2]. Nevertheless, blockchain evidence remains admissible as supporting evidence under Article 46, which establishes that electronic documents shall not be denied legal effect solely on the grounds of their electronic form.

For a QTSP that integrates blockchain into its service architecture, the blockchain component falls within the scope of the conformity assessment required by Article 20, which mandates audits at least every 24 months by an accredited conformity assessment body. The audit scope must extend to the full service architecture, including consensus mechanism integrity, validator node key management, smart contract governance, and data retention procedures, all of which map to the organizational and technical controls defined in ETSI EN 319 401 [2]. Furthermore, any on-chain evidence storage must comply with the General Data Protection Regulation (GDPR) Regulation (EU) 2016/679, particularly the tension between blockchain immutability and the right to erasure (Article 17). Compliant designs address this by storing only cryptographic hashes or pseudonymous references on-chain, keeping personal data off-chain. Guggenmos et al. formalize this design constraint by developing a GDPR-compliant blockchain solution for cross-organizational workflows, identifying design patterns, such as off-chain personal data storage with on-chain hash references, that are directly applicable to blockchain-augmented trust services operating under both eIDAS and GDPR [36]. Notably, the revised eIDAS 2.0 framework (Regulation (EU) 2024/1183) [21] introduces electronic ledgers as a recognized trust service category (Article 45h), providing for the first time a regulatory pathway for blockchain-based evidence to be offered as a qualified service, although the implementing acts and technical standards remain under development.

Table 1 consolidates the technological baseline established in this section by mapping each layer of the eIDAS trust service stack to its governing standards, the centralization concern that motivates blockchain exploration, and the blockchain role examined in the subsequent sections.

Table 1. eIDAS trust service technology stack: standards, centralization concerns, and blockchain integration points.

Trust Service Layer	Governing Standards	Centralization Concern	Blockchain Role
Certificate lifecycle (issuance, revocation)	ETSI EN 319 411 [3]; X.509; CRL/OCSP	Single CA as trust anchor; revocation availability depends on CA infrastructure	Transparent, tamper-evident log of issuance and revocation events (Section 3)
Remote qualified signing	ETSI TS 119 431 [22]; CEN EN 419 241-1 [23]; CSC API [27]	Private key held by TSP; sole control assurance depends on provider's internal controls	Independent audit trail for key-access and signing events (Section 4)
Signature preservation	ETSI TS 119 511/512 [37,38]; AdES formats [4]	Preservation evidence held by a single provider; long-term integrity depends on provider continuity	Integrity anchoring of preserved objects and re-augmentation evidence (Section 5)
Signature validation	ETSI EN 319 102-1 [4]; ETSI TS 119 172-4 [39]	Validation reports and audit trails held internally by the validating party	Shared, immutable log of validation operations across consortium (Section 5.4)
Timestamping	RFC 3161 [40]; ETSI EN 319 421/422 [35,41]; EN 319 401 [2]	Timestamp authority as single trusted time source; token integrity depends on TSA availability	Supplementary proof-of-existence via blockchain anchoring (Section 6)
Content provenance	C2PA specification [42]; PKI-based Content Credentials	Trust anchored in issuing CA; no decentralized verification of provenance chain	Decentralized attestation and anchoring of provenance records (Section 7)
Digital identity (EUDI Wallet)	eIDAS 2.0 [21]; ARF v2.8.0 [43]; ETSI TS 119 612 [44]; OpenID4VCI/VP [45,46]	Trusted Lists as centralized registries; credential revocation depends on issuer infrastructure	Trust registries, DID resolution, and privacy-preserving revocation via EBSI (Section 8)
Legal and audit framework	eIDAS Articles 20, 25, 35, 41, 42, 46 [1]; EN 319 401 [2]; GDPR Art. 17	Blockchain evidence lacks qualified legal presumption; GDPR tension with immutability	Supplementary evidence (Art. 46); on-chain hashes only; eIDAS 2.0 electronic ledger category

This review evaluates blockchain-based proposals along complementary dimensions that include their alignment with the eIDAS institutional trust model and ETSI/CEN compliance constraints, such as qualified services, standardized signature and validation procedures, together with their practical operational profile, assessed through latency, throughput, storage cost, and audit efficiency, as discussed in Section 9.

3. Blockchain in Certificate Management

Certificate handling is a central function of trust services, covering the issuance, revocation, and verification of digital certificates across their lifecycle. In traditional PKIs, these functions are performed by trusted centralized actors, most notably Certification Authorities (CAs) and related validation services. Although this model is mature and widely deployed, it also concentrates trust and operational responsibility in a limited number of entities. Consequently, events such as CA compromise, certificate mis-issuance, delayed revocation dissemination, or weak auditability may have broader consequences for both relying parties and service providers.

Blockchain technology has been explored as a complementary mechanism for improving certificate management, mainly because it provides a distributed and tamper-evident record of events. Rather than replacing cryptographic certificates or the legal role of Trust Service Providers, blockchain can support stronger traceability of issuance and revocation events and reduce dependence on a single technical point of failure [5].

The relevance of blockchain here stems largely from three well-known properties: decentralization, immutability, and transparency. Instead of assigning the management of certificate events to a single entity, a blockchain-based design distributes the evidentiary record across multiple participants. Once an event has been recorded, later modification becomes difficult, which reinforces the integrity of certificate history. In addition, several parties can inspect the same ledger, making lifecycle events easier to trace and verify [8]. Still, these properties should be viewed as supporting elements rather than complete solutions, since they do not by themselves resolve all PKI-related issues.

3.1. From Traditional PKI to Blockchain-Assisted Certificate Management

Traditional PKI is based on hierarchical trust. A root CA anchors trust, subordinate CAs may issue end-entity certificates, and relying parties validate certificate chains against trusted roots and status information. This model is mature and interoperable, but it also depends on the correct behavior of Certification Authorities and on the timely availability of revocation and validation information.

From an operational point of view, blockchain use in this area generally follows two directions. One direction treats the ledger as a transparent and verifiable log for issuance and revocation events. The other explores broader decentralized PKI models in which certificates, identities, and trust relationships are handled through smart contracts or distributed registries. The first option fits more naturally with existing PKI practice, while the second is more ambitious and more difficult to align with compliance-driven trust service environments.

To formalize the distinction between these two directions, we describe the certificate verification process under traditional PKI and its blockchain-augmented extension. In standard PKI, a relying party verifies an end-entity certificate $Cert_{EE}$ by constructing and validating a certificate chain $\langle Cert_{EE}, Cert_{CA_1}, \ldots, Cert_{Root} \rangle$. For each certificate $Cert_i$ in the chain, the relying party performs the following checks:

1. Signature verification: Verify that $Cert_i$ was signed by the issuer CA_{i+1}, i.e., $\text{Verify}(pk_{CA_{i+1}}, Cert_i.signature, Cert_i.tbsCertificate) = true$, where $pk_{CA_{i+1}}$ is the public key of CA_{i+1}, used to verify the signature in $Cert_i$.
2. Validity period: Confirm that the current time t satisfies $Cert_i.notBefore \leq t \leq Cert_i.notAfter$.
3. Revocation status: Query the CA's revocation service, either by downloading the Certificate Revocation List (CRL) and checking that $Cert_i.serialNumber \notin CRL$, or by sending an Online Certificate Status Protocol (OCSP) request and receiving a signed response confirming $status = good$.
4. Trust anchor: Verify that $Cert_{Root}$ (the self-signed root CA certificate at the top of the chain) is present in the relying party's trusted root store $\mathcal{T}$, i.e., $Cert_{Root} \in \mathcal{T}$.

The overall verification succeeds if and only if all four conditions hold for every certificate in the chain. This process is well-defined by RFC 5280 and implemented by standard validation libraries, but it depends entirely on the CA infrastructure: the CA must have correctly issued each certificate, the revocation service must be available and up to date, and the root store must be accurately maintained.

In the blockchain-augmented model, the traditional chain validation described above is preserved unchanged. The relying party still verifies signatures, validity periods, revocation status, and trust anchors using standard PKI mechanisms. The blockchain layer adds a supplementary integrity check by providing an independent, tamper-evident record of certificate lifecycle events. Specifically, for each certificate $Cert_i$ in the chain, the relying party may additionally verify the following:

5. Ledger consistency: Confirm that the issuance of $Cert_i$ was recorded on the blockchain, i.e., $\exists tx \in \mathcal{L}$ such that $tx.certHash = H(Cert_i)$ and $tx.event = issued$, where $\mathcal{L}$ denotes the set of committed transactions on the ledger, H is a cryptographic hash function, and tx is a ledger transaction in $\mathcal{L}$.
6. Revocation cross-check: If a revocation event for $Cert_i$ exists on the ledger, verify that $\exists tx' \in \mathcal{L}$ such that $tx'.certHash = H(Cert_i)$ and $tx'.event = revoked$, and that this is consistent with the CRL/OCSP status obtained in step 3.
7. Merkle proof verification: For each ledger record, validate the Merkle inclusion proof π against the block's root hash r_b: $\text{VerifyProof}(\pi, H(tx), r_b) = true$, confirming that the record has not been altered since it was committed.

Steps 1–4 retain the full legal and technical validity of standard PKI verification. Steps 5–7 provide supplementary assurance: they allow the relying party to detect discrepancies between the CA's current state and the historical record on the ledger, for example, a certificate that was silently revoked and later reinstated, or an issuance event that was backdated. Crucially, the blockchain does not replace the CA's authority to issue or revoke certificates; it provides an independent witness that can be consulted to increase confidence in the integrity of the certificate lifecycle.

A representative example is the conceptual framework proposed by Yakubov et al. [9], which explores how blockchain can support X.509 certificate handling while reducing the risks associated with CA compromise. In that model, lifecycle events are registered through blockchain-based structures, creating a verifiable record of operations such as issuance and revocation. The framework combines characteristics of log-based PKI with broader distributed trust ideas in an effort to improve transparency and resilience. At the same time, it also illustrates several practical difficulties, including ledger growth, cost variability on public chains, and technical constraints imposed by the smart contract environment.

3.2. Blockchain as a Transparency Layer for Certificate Operations

One of the most relevant uses of blockchain in this area is the maintenance of a transparent and append-only record for lifecycle events. Under this model, operations such as issuance, renewal, and revocation updates are reflected as blockchain transactions. This allows external parties to confirm both that an event occurred and when it occurred, while also increasing confidence that the corresponding record has not been silently removed or altered.

Several research proposals and prototype-oriented studies have explored the use of blockchain to improve certificate transparency and revocation transparency in PKI-related environments [10–12]. A shared characteristic of these proposals is that issuance and revocation actions are represented through smart-contract-based mechanisms deployed on a ledger. Rather than attempting to redesign the whole PKI ecosystem, these works usually target narrower operational concerns, especially transparency and traceability of certificate-related actions. At the same time, they do not cover the full range of PKI challenges, and some remain affected by issues such as trust-model weaknesses, split-world attacks, or man-in-the-middle scenarios.

Revocation is another area in which blockchain has been explored as a potential enhancement to classical certificate management mechanisms. In traditional PKI, certificate revocation depends on CRLs or OCSP responses. These mechanisms are effective but can introduce operational overhead, availability dependencies, and trust centralization. Recording revocation events on blockchain may improve consistency and auditability, especially in distributed environments where several parties need a shared and verifiable view of certificate status. However, this advantage must be balanced against latency, scalability, and the cost of recording revocation information to the ledger.

In public blockchains, an important advantage lies in the fact that revocation-related records can become broadly visible and independently verifiable by a wide range of participants, without relying on a single trusted entity. This can strengthen transparency and make it more difficult for certificate status changes to be concealed or selectively presented. Nevertheless, these benefits come with trade-offs, including higher latency, transaction costs, and, in some cases, confidentiality concerns arising from the visibility of metadata on a publicly accessible ledger. In permissioned blockchains, by contrast, revocation recording can support stronger coordination among authorized participants, controlled access to status information, and more predictable performance. However, these advantages are achieved under a more restricted trust and transparency model, since visibility and participation depend on consortium governance rather than on open public verification. In either case, the expected benefits must be balanced against latency, scalability, governance design, and the operational cost of recording revocation-related information on the ledger.

3.3. Smart Contracts for Certificate Issuance and Revocation

Some of the more advanced blockchain-based designs use smart contracts to support certificate-related operations directly on-chain. In these models, a CA or another trusted participant interacts with a contract that records certificate hashes, status indicators, issuer references, or revocation data. The contract may expose functions related to issuance, revocation, validation support, and certificate chain handling.

A representative example is the blockchain-based PKI management framework proposed in [9], in which each CA is associated with a dedicated smart contract, and certificate-related operations are exposed through a Representational State Transfer (REST) interface. Instead of reproducing the entire PKI logic in a traditional centralized repository, the framework stores certificate hashes in blockchain-managed structures and allows issuance and revocation events to be registered in a verifiable way. These architectures reveal the practical limitations of blockchain adoption in certificate management. Public smart-contract platforms introduce transaction fees, performance constraints, and ledger growth over time. In addition, implementation choices may be affected by the limitations of smart contract languages and virtual machine environments, including restricted native handling of strings, cryptographic primitives, and data structures. Patsonakis et al. provide empirical evidence for these limitations by evaluating an Ethereum-based smart contract PKI, reporting gas costs per certificate operation and end-to-end latency measurements that quantify the practical overhead of on-chain certificate management [47].

Another important issue is the boundary between on-chain and off-chain information. Full certificates, identity evidence, policy documents, and validation artifacts are often either too large or too sensitive to place directly on a public ledger. For that reason, most realistic solutions store only concise proofs or references on-chain, while preserving detailed data in external repositories. This hybrid arrangement is often necessary, but it also means that blockchain does not remove dependence on outside systems; rather, it redistributes where evidence and trust are anchored.

3.4. Benefits of Blockchain in Certificate Management

In this domain, the main contribution of blockchain is not the replacement of certificates, but the improvement of how lifecycle events are recorded, shared, and checked. This leads to several practical advantages.

One expected benefit is the reduced dependence on a single technical point of failure. Even where a CA remains the legal issuer, certificate-related events can be recorded on a distributed ledger that multiple participants can verify independently. At least at the

level of architecture and system design, this can improve resilience and reduce operational dependence on a single repository or service [5].

In addition, blockchain can strengthen transparency and auditability. If issuance and revocation records are publicly visible, unauthorized certificate actions become harder to conceal. This is particularly valuable in environments where multiple organizations, supervisory actors, or relying parties need a common evidentiary view of the certificate lifecycle [10,11].

Blockchain can also support a degree of automation through smart contracts. Functions such as checking policy conditions, registering issuers, or publishing lifecycle events can be expressed as code and executed in a predictable way. This does not remove the need for governance, but it can make processes more transparent and more consistent across different actors.

3.5. Limitations and Open Challenges

Despite these advantages, several important limitations remain. Interoperability is still a major concern, since trust service ecosystems are built around established X.509 standards, validation procedures, and widely deployed software components. Blockchain-based proposals, therefore, need to integrate with existing systems rather than assume that those systems can simply be replaced.

Cost and scalability also remain challenging. Public blockchain networks introduce transaction fees, performance constraints, and continuous ledger growth, which may make large-scale certificate event registration difficult to sustain in practice. Privacy is another concern, as certificate-related data or metadata may reveal sensitive information when recorded on a transparent ledger. Governance must also be carefully defined, since decentralization does not remove the need to specify who can issue, revoke, or update certificate records.

In eIDAS-aligned settings, blockchain can be useful as a supporting infrastructure for operations such as issuance, status recording, revocation, and event traceability. Its main contribution is not to replace current trust service structures, but to reinforce selected lifecycle functions through transparent logging, improved auditability, and more consistent handling of operational events. Although important challenges remain, controlled and compliant integration can still strengthen the robustness and verifiability of certificate-related processes within the broader eIDAS trust services framework.

4. Blockchain-Based Auditing for Remote Signing Services

Compared with certificate-related use cases, blockchain has been examined to a lesser extent in the area of signature services. One reason is that electronic signature systems, especially those built around remote and qualified signing models, already operate within mature technical and regulatory frameworks. As a consequence, blockchain is usually not considered a substitute for existing signing infrastructures, but rather an additional means of improving auditability, transparency, and user confidence in the signature creation process.

Its relevance becomes more apparent in remote signing environments, where the signing key is generated, stored, and used within the infrastructure of a TSP. Even where these systems comply with applicable technical and security requirements, they still raise a persistent trust concern: the user no longer holds direct physical control over the private key. In that setting, the central question is not only whether the resulting signature is technically valid, but also whether access to the private key can be shown to have occurred under the sole control of the legitimate signer.

4.1. Blockchain as an Audit Layer for Remote Signing

In digital signature services, blockchain can function as an external and tamper-evident audit layer for signing-related events. Rather than changing the cryptographic structure of the signature itself, this type of approach records evidence about how the signing process took place, including the user's intention to sign and the actual access to the private key. The aim is to create an additional trust source outside the centralized signing environment, while still keeping it connected to the signature lifecycle.

This direction is particularly relevant for cloud-based QES, where sole control is a central requirement. In conventional remote signing architectures, compliance is achieved through authentication, authorization, and secure key management. However, even in compliant systems, the user must still rely heavily on the TSP as the operator of the signing environment. A blockchain-based audit mechanism can strengthen this model by providing an immutable and independently verifiable history of key-access-related events, without requiring a redesign of the underlying PKI and trust service architecture.

4.2. Blockchain-Based Auditing of Remote Signing Operations

A relevant example in this area is SABRES (Secure Auditing using Blockchain for Remote Electronic Signatures) [48], presented as a proof-of-concept solution for cloud-qualified QES. The cited work shows how blockchain-based audit evidence can be integrated into remote signing workflows to strengthen trust in the way the signing key is accessed and used within the TSP environment. More specifically, the solution addresses a central concern of cloud signing services: even when the process is compliant with applicable standards, the private key remains stored and used within the provider's infrastructure, which means that the user still depends on the provider's trustworthiness. At the level reflected in the paper, the proposal provides proof-of-concept support for this direction and for improving the auditability and evidentiary traceability of remote signing operations. In this sense, its relevance lies in showing how blockchain-based auditing can reinforce the interpretation of sole control in cloud signing workflows, even if broader deployment-level validation is not discussed in the cited work.

The approach combines the maturity, interoperability, and legal recognition of PKI-based remote signature services with the distributed trust and tamper-evident characteristics of blockchain. In this model, the blockchain acts as an external audit mechanism that records key events related to the signing process, creating an additional source of evidence about the operation itself. The purpose is to increase confidence that the use of the private key can be associated with an explicit user action, rather than being treated solely as an internal server-side event.

From an architectural perspective, the solution [48] preserves the traditional remote signing workflow and remains aligned with SCAL2 [23] and CSC-based service interaction [27]. Instead of redefining the TSP infrastructure, it introduces additional components around it, including a client-side signing application, a reverse-proxy module deployed in front of the signing service, and a blockchain layer used for recording audit events. This design choice is relevant because it allows blockchain-based methods to be added without fundamentally changing the existing signature service logic.

Operationally, the mechanism relies on two complementary blockchain records. The first is a signature intent, generated on the client side to express the user's intention to create a signature. The second is a signature event, generated on the server side when the signing key is effectively accessed after a successful signing request. These two records are linked through shared identifiers, allowing the system to associate a particular request with a particular key-access event. In addition to document-related hash values and credential references, the design also includes elements meant to distinguish repeated

signing operations and to provide evidence that cannot be reconstructed by the service provider alone.

To formalize this audit mechanism, we describe the blockchain-based evidence model for remote signing operations as implemented in the SABRES architecture [48]. The process involves three parties: the user U, the signing service S (operated by a QTSP), and the blockchain ledger $\mathcal{L}$.

- Phase 1—Signature intent recording: Before the signing operation is executed, the client-side application constructs a signature intent record and submits it to the blockchain:
$$I = \langle uid, H(d), credU, nonce, t_{req} \rangle \quad (1)$$
where uid is a unique operation identifier, $H(d)$ is the cryptographic hash of the document d to be signed, $credU$ is a reference to the user's authentication credential, $nonce$ is a freshness value to prevent replay, and t_{req} is the timestamp of the request. The intent is committed to the ledger as a transaction: $tx_I \leftarrow \text{Commit}(\mathcal{L}, I)$;
- Phase 2—Signing operation: The signing service S authenticates the user, verifies the authorization (Signature Activation Data per SCAL2 [9]), and performs the cryptographic signing operation using the private key sk held in the remote QSCD:
$$\sigma = \text{Sign}(sk, H(d)) \quad (2)$$
where sk is the private signing key. This step follows the standard CSC/TW4S workflow and is unchanged by the blockchain layer.
- Phase 3—Signature event recording: After the signing operation completes, the server-side component constructs a signature event record and commits it to the blockchain:
$$E = \langle uid, H(d), H(\sigma), certRef, t_{sign} \rangle \quad (3)$$
where $H(\sigma)$ is the hash of the produced signature, $certRef$ is a reference to the signing certificate, and t_{sign} is the timestamp of the key-access event. The event is committed: $tx_E \leftarrow \text{Commit}(\mathcal{L}, E)$.

An auditor (or an extended signature validation process) can verify the integrity and consistency of a signing operation by performing the following checks against the ledger:

1. Intent existence: Verify that $\exists tx_I \in \mathcal{L}$ with matching uid and $H(d)$, confirming that the user expressed an intention to sign the specific document.
2. Event existence: Verify that $\exists tx_E \in \mathcal{L}$ with matching uid and $H(d)$, confirming that the signing key was accessed.
3. Intent-event linkage: Verify that $tx_I.uid = tx_E.uid$ and $tx_I.H(d) = tx_E.H(d)$, confirming that the key access corresponds to the user's documented request.
4. Temporal ordering: Verify that $tx_I.t_{req} \leq tx_E.t_{sign}$, confirming that the intent preceded the signing event.
5. Merkle proof: For each record, validate the inclusion proof π against the block root hash r_b: $\text{VerifyProof}(\pi, H(tx), r_b) = true$.

If all five checks pass, the auditor obtains cryptographic assurance that the signing operation was preceded by a documented user intent, that the two records are mutually consistent, and that neither has been tampered with since commitment to the ledger. If any check fails, for example, a signature event exists without a corresponding intent, or the document hashes do not match, this constitutes detectable evidence of an irregularity in the signing process that would not be visible from the signature artifact alone.

The key property of this model is that it produces evidence that cannot be reconstructed solely by the service provider. The signature intent originates from the client side, while the signature event originates from the server side. An attacker who compromises only the server cannot forge a valid intent record, and an attacker who compromises only the client cannot forge a valid event record. The combination of both records on an immutable ledger creates a two-sided evidentiary chain that strengthens the sole control assurance beyond what the internal logs of a single TSP can provide.

An important characteristic of the proposed solution [48] is that the blockchain does not replace the signature itself, but supplements it with auditable evidence. Signature validation can therefore be extended beyond conventional checks on certificate status, cryptographic correctness, and signature format. In the SABRES model, validation also involves checking whether the expected blockchain evidence exists and whether the relationship between the recorded signing intent and the recorded signature event is consistent. If this evidentiary link is missing, the signature may still be technically well formed, but it no longer benefits from the additional assurance provided by the blockchain-based audit mechanism.

This makes the proposal particularly relevant in the context of cloud-qualified electronic signatures. Its contribution lies not in altering the cryptographic standards of remote signing, but in introducing a distributed and verifiable method for auditing accesses to the private signing key. In this way, the approach strengthens traceability and supports a more transparent interpretation of sole control in environments where the key remains under the operational custody of the Trust Service Provider.

4.3. Relevance and Practical Challenges

The role of blockchain in this area differs from its role in certificate-related processes. In certificate-focused use cases, the ledger is used mainly to increase transparency of issuance and revocation events. In signature services, by contrast, its value is more closely connected to the verifiability of the signing act itself, especially in cloud scenarios where trust in server-side key usage is essential.

At the same time, this approach also raises practical challenges. Recording signature-related events on blockchain may introduce additional costs and performance overhead, particularly in high-volume environments. It may also complicate deployment if blockchain verification is expected to remain visible to end users rather than being integrated transparently into validation services or middleware. Moreover, blockchain does not replace the broader trust service framework, since remote signing still depends on the TSP for identity management, authentication, secure key storage, and signature generation.

Within eIDAS-oriented trust services, blockchain is therefore best understood as a complementary mechanism that can improve traceability and strengthen the evidentiary value of remote signing operations. Its main benefit lies in reinforcing selected controls around the signing process, rather than replacing the legal, technical, and organizational foundations of qualified signature services.

5. Blockchain in Signature Preservation

In many eIDAS use cases (public procurement, notarization workflows, financial contracts, employment records, medical and administrative archives), signed documents must remain verifiable for years or decades. Over such horizons, the technical elements that support signature validity at signing time (cryptographic algorithms, certificate chains, revocation mechanisms, validation services, and policy rules) inevitably change. Under eIDAS and the ETSI AdES and preservation standards, signature preservation is best understood as the continuous maintenance of the signed object and its signature container, the validation material required to revalidate it later (e.g., certificate chains and revocation

status information), plus time and integrity proofs that bind these elements to an evidence timeline. This typically includes qualified electronic timestamps and/or evidence records that can be periodically renewed to counter algorithm deprecation and other long-term risks. Preservation services offered by QTSPs are governed by dedicated policy, security, and protocol specifications, reflecting the higher assurance requirements and the operational reality of recurrent renewal cycles rather than a one-time verification event [37,38].

Blockchain enters this landscape as a complementary mechanism for strengthening the integrity and auditability of preservation operations. The core preservation challenge is not only cryptographic longevity (e.g., re-timestamping before algorithms weaken), but also evidential robustness: demonstrating to independent third parties, and potentially in court, that a preservation service followed prescribed procedures, maintained correct custody, and did not alter or omit material events. A distributed ledger can provide an additional tamper-evident log for preservation events (e.g., submission, renewal, migration, policy updates) and a public or consortium-verifiable anchoring point for preserved objects via hash commitments. However, the legal presumption and technical semantics of eIDAS qualified electronic timestamps and ETSI evidence records remain central. Blockchain-based time indications are not automatically equivalent to qualified timestamps and should be treated as supporting evidence rather than a substitute. More generally, throughout eIDAS-governed trust services, blockchain-based augmentation may strengthen evidentiary robustness, transparency, or auditability, but it does not by itself create legal equivalence with qualified trust service evidence.

5.1. The Signature Preservation Process and Its Integrity Requirements

Under ETSI TS 119 511 [38] and the associated technical specification TS 119 512 [37], the preservation of a digitally signed object involves a defined sequence of operations that maintain the verifiability of the signature over time. We formalize this process to clarify the points at which blockchain can contribute.

Let $S = (d, \sigma, C)$ denote a signed object, where d is the document, σ is the electronic signature, and $C = \langle Cert_{EE}, \ldots, Cert_{Root} \rangle$ is the certificate chain. At signing time t_0, the signature is valid if

$$\text{Verify}(pk_{EE}, \sigma, H(d)) = true \quad \wedge \quad \text{ChainValid}(C, t_0) = true \tag{4}$$

where ChainValid encompasses the certificate chain validation described in Section 3.1 (steps 1–4). Over time, the elements that support this verification may degrade: the signing algorithm may become cryptographically weak, certificates may expire, revocation information may become unavailable, and the CA infrastructure may change. Preservation addresses this through a recurring augmentation cycle.

1. Initial augmentation: At time $t_1 > t_0$ (while σ is still verifiable), the preservation service validates the signature, collects the validation material V_1 (certificate chain, OCSP responses or CRL entries, and trust anchor references at time t_1), and obtains a qualified timestamp TS_1 that binds the signed object S and its validation material to a verifiable point in time:

$$P_1 = \langle S, V_1, TS_1 \rangle \quad \text{where} \quad TS_1 = \text{TSA.Stamp}(H(S|V_1), t_1) \tag{5}$$

   The timestamp TS_1 is a qualified electronic timestamp according to RFC 3161 [40] and ETSI EN 319 422 [41], carrying the legal presumption of Article 41(2) of the eIDAS Regulation;
2. Re-augmentation: Before the cryptographic algorithm underlying TS_1 is at risk of being compromised (or before TS_1's certificate chain expires), the preservation service

performs a re-augmentation at time $t_2 > t_1$. It collects fresh validation material V_2 for TS_1 and obtains a new timestamp using a current (stronger) algorithm:

$$P_2 = \langle P_1, V_2, TS_2 \rangle \quad \text{where} \quad TS_2 = \text{TSA.Stamp}(H(P_1|V_2), t_2) \tag{6}$$

This creates a nested chain of evidence: $P_n = \langle P_{n-1}, V_n, TS_n \rangle$. Each layer extends the verifiability window by binding the previous state to a new timestamp under a current algorithm. The process can be repeated indefinitely, providing cryptographic longevity.

3. Blockchain augmentation: In the blockchain-augmented model, each preservation event is additionally anchored on a distributed ledger. At each augmentation step t_k, the preservation service commits an anchoring record:

$$A_k = \langle pid, H(P_k), cid_k, t_k \rangle \quad \text{with} \quad tx_k \leftarrow \text{Commit}(\mathcal{L}, A_k) \tag{7}$$

where pid is the preservation object identifier, $H(P_k)$ is the hash of the full preservation state at step k, and cid_k is an optional content identifier (e.g., an IPFS CID as in [49]) pointing to the stored object. The blockchain record does not replace the qualified timestamp TS_k, which retains its legal presumption under eIDAS, but provides supplementary evidence that offers the following:

- Existence: The preserved object in state P_k existed at the time the block containing tx_k was confirmed.
- Integrity: $H(P_k)$ can be recomputed from the stored object and compared against the on-chain record at any future time.
- Non-repudiation of process: The sequence $\langle A_1, A_2, \ldots, A_n \rangle$ on the ledger provides a tamper-evident history of all preservation actions, independently verifiable without relying on the preservation provider's internal logs.

This model is implemented progressively in the works reviewed below. TrustChain [14] focuses on point (1), using blockchain as a durable existence proof for signed objects that can outlive individual PKI actors. LTPS [13] implements the full augmentation cycle with Ethereum anchoring, explicitly distinguishing hash anchoring from validity preservation. The IPFS (InterPlanetary File System) extension [49] adds distributed content-addressed storage, anchoring cid_k values on-chain so that both the integrity proof and the storage location are publicly verifiable.

5.2. Blockchain-Augmented Preservation: From Evidence Anchoring to Distributed Storage

Bralić et al. introduce TrustChain as a first blockchain-oriented model for sustaining the verifiability of digitally signed documents over long periods, motivated by the observation that conventional PKI-based validation degrades when supporting infrastructure changes or disappears (e.g., certificate authorities cease operations, revocation information is no longer obtainable, or certificate chains become difficult to reconstruct) [14]. Their core idea is to use a blockchain as a durable, append-only evidentiary substrate that can outlive individual PKI actors and repositories: rather than relying solely on external availability of validation artifacts, TrustChain records integrity-critical references on-chain so that a verifier can later confirm that a particular signed object (or its signature-related representation) existed in a specific state and has not been altered since anchoring.

Blockchain-enabled signature preservation architectures tend to converge on a common principle: store the preserved content and ETSI-compliant preservation evidence off-chain, and use the blockchain to add an extra layer of integrity and auditability. In a paper from 2022, Arseni at al. position long-term preservation as a core capability rather than a "nice-to-have" feature [15]. A valuable part of this work is its synthesis of the

ETSI landscape and practical market behavior. Their comparative review of preservation solutions highlights that real deployments are strongly shaped by interoperability and user expectations, especially broad support for common AdES formats (notably PAdES), cloud-oriented delivery models, and service interfaces that align with standard preservation workflows. This reinforces a key constraint for any blockchain-based proposal: in practice, blockchain must complement ETSI-compliant evidence management rather than replace signature formats or validation behaviors that mainstream verifiers depend on.

In the LTPS paper [13], their proposal implements an ETSI-aligned preservation service that performs signature augmentation, interacting with a validation component, a timestamp service, and that orchestrates monitoring along with re-augmentation over time. The architectural decomposition (protocol endpoint, policy configuration, augmentation and validation clients, and storage connectors) highlights preservation as a multi-service workflow where the preservation provider acts as an evidence curator operating under formal policies. Blockchain enters this architecture as an additional integrity and traceability layer. LTPS anchors relevant preservation information by registering a digest (and related identifiers) to the public blockchain Ethereum. This yields a distributed record that can function as an independent existence and integrity witness alongside the conventional qualified timestamping mechanisms. The important conceptual move is that the blockchain does not replace TSA (Timestamping Authority). Qualified timestamps remain the primary legally aligned time evidence, while the blockchain record strengthens auditability and tamper detection through replicated immutability.

Crucially, LTPS differentiates hash anchoring from validity preservation. The authors explicitly contrast approaches where the service only stores data on-chain to preserve status with the practical reality that users expect preserved artifacts to remain verifiable with standard tools.

A follow-up paper moves the discussion from evidence anchoring toward the design of the preservation storage layer. Although ETSI TS 119 511 explicitly allows several storage models, including with storage, without storage, and temporary storage, the authors focus on the with storage (WST) scenario, in which the provider stores preserved objects for the entire preservation period and therefore behaves in practice like an archive component within the preservation service boundary. They argue that, in WST, the provider must meet requirements related to traceability, transparency, integrity, and confidentiality, which makes the storage architecture a central design issue rather than a secondary deployment detail [49].

Their proposed implementation integrates IPFS as content-addressed distributed storage and Ethereum as a distributed trust anchor. Their operational flow is: validate and augment the signature, obtain timestamp evidence, store the resulting augmented object in IPFS (receiving a CID), persist metadata internally (including a preservation object identifier), and then write the CID and key metadata/identifiers to the blockchain. Thus, clients can later verify the preservation history and referenced content. The paper also makes an explicit usability point: public chain explorers (e.g., Etherscan) provide a practical third-party verification surface for checking preservation-related transactions.

A distinctive contribution of that work is the empirical comparison between IPFS and several classical storage backends in terms of upload and download time. More specifically, the cited paper measures upload and download duration for 1 MB, 10 MB, and 100 MB files across IPFS, MongoDB, SQL Server, PostgreSQL, MySQL, and FTP, based on repeated on-premise tests that exclude preservation-specific operations such as timestamping and signature validation. The comparison therefore evaluates raw storage access performance rather than the overall suitability of each backend for preservation services. The results show that IPFS is not the fastest option if raw latency is considered alone. However, the

comparison is not entirely direct, because most of the alternative backends evaluated are centralized and do not offer the built-in distribution, content addressing, and redundancy provided by IPFS. For this reason, the results should not be read as proving general storage superiority or inferiority. The authors therefore argue that the observed performance remains reasonable given that IPFS delivers distributed robustness by design, whereas similar redundancy in conventional storage would require additional replication mechanisms implemented separately.

Taken together, these three lines of work, TrustChain [14], LTPS [13], and the IPFS+Ethereum extension [49], represent a progression from existence proofs (anchoring a hash on-chain) to full-cycle preservation with distributed storage, each building on the formal augmentation and anchoring model described in Section 5.1.

5.3. Relevance and Practical Challenges

The blockchain-augmented preservation model formalized in Section 5.1 and implemented by the works reviewed in Section 5.2 offers clear benefits in terms of independent integrity verification and tamper-evident process logging. However, its practical adoption within eIDAS-governed preservation services faces several challenges that must be considered:

- Legal status of blockchain anchors—The qualified timestamps TS_k in the preservation chain carry the legal presumption of accuracy under eIDAS Article 41(2) and are issued by supervised QTSPs in compliance with ETSI EN 319 421 [35]. The blockchain anchoring records A_k, by contrast, do not carry this presumption. A blockchain confirmation proves that a hash was committed before a certain block was produced, but it does not constitute a qualified electronic time stamp: it lacks the accountability of a supervised TSA, the trusted time-source guarantees required by the standard, and the standardized token format that existing validation tools expect [50]. For this reason, blockchain anchors in preservation architectures must be understood as supplementary evidence that strengthens auditability, not as a replacement for the qualified timestamps that form the legal backbone of the preservation chain.
- Interoperability with existing validation tools—A key constraint identified by the authors in [15] is that real-world preservation deployments are shaped by interoperability requirements: relying parties expect to verify preserved signatures using standard AdES validation procedures (ETSI EN 319 102-1 [4]) and common formats (PAdES, CAdES, XAdES). Blockchain-anchored evidence must therefore be structured so that it complements rather than interferes with standard verification. The LTPS approach [13] addresses this by keeping the augmented signature container fully ETSI-compliant and treating the blockchain record as a parallel integrity layer that is consulted during audits but not required by standard validators. The IPFS extension [49] similarly stores the preservation object in its original format and only anchors a CID and metadata on-chain.
- Storage cost and longevity—Preservation services operate over horizons of years to decades. The cost of maintaining blockchain anchors over such periods depends on the chosen platform. On public chains such as Ethereum, the per-anchor gas cost is low when amortized through batching (a single Merkle root covering thousands of preserved objects), but the long-term availability of specific chain state is not guaranteed by protocol, state pruning, chain reorganizations after hard forks, and the evolving economics of node operation introduce risks that do not exist with traditional qualified timestamps stored in standardized containers. On permissioned chains, the cost is absorbed into infrastructure provisioning, but the long-term governance and continuity of the consortium become the critical dependency. The empirical compari-

son in [49] shows that IPFS storage costs are comparable to centralized alternatives ($20/TB/month vs. $23/TB/month for S3), but IPFS retrieval latency (600 ms DHT discovery) is higher than centralized object stores (100–200 ms), a tradeoff that may be acceptable for archival workloads where retrieval frequency is low.

- Integrity of the re-augmentation chain—The nested structure $P_n = \langle P_{n-1}, V_n, TS_n \rangle$ means that the integrity of the entire preservation history depends on the availability and correctness of every intermediate layer. If a blockchain anchor A_k is lost (e.g., due to chain pruning) or if the content behind a CID cid_k becomes unavailable (e.g., IPFS content is unpinned), the supplementary evidence for that augmentation step is degraded. The preservation object itself and its qualified timestamps remain valid, since they are stored off-chain in standard formats, but the additional auditability provided by the blockchain layer is weakened. Preservation service operators must therefore implement retention policies for on-chain records and pinning commitments for distributed storage that match or exceed the intended preservation horizon.

Despite these challenges, the combination of ETSI-compliant preservation workflows with blockchain-based integrity anchoring represents a pragmatic path toward stronger auditability in long-term signature preservation. The qualified timestamp chain provides the legal and cryptographic backbone, while the blockchain layer adds an independently verifiable process history that does not depend on the preservation provider's internal records.

5.4. Blockchain-Augmented Signature Validation

Signature preservation and signature validation are complementary operations within the eIDAS trust service ecosystem: preservation maintains the conditions under which a signature can be validated over time, while validation is the act of determining whether a signature meets the applicable requirements at a given moment. In the eIDAS ecosystem, the validation process is not limited to verifying a cryptographic signature over a document. A conformant validator must apply procedures and policies that combine format and profile-specific AdES checks (PAdES/XAdES/CAdES/ASiC), certificate path construction and validation (including revocation state), and time and qualification reasoning that can depend on Trusted Lists and their historical status at signing time. ETSI EN 319 102-1 formalizes this as a validation model driven by a signature validation policy and specifies the baseline procedures a validation application should follow [4]. In parallel, ETSI TS 119 172-4 defines rules to determine whether a signature or seal meets the requirements of EU QES/seals by taking into account the EU Member States' Trusted Lists and related constraints [39].

We formalize the validation process and its blockchain augmentation as follows. Given a signed object $S = (d, \sigma, C)$ and a validation policy $\mathcal{P}$, the standard validation process produces a validation result:

$$R = \text{Validate}(S, \mathcal{P}, t_v) \rightarrow \textit{TOTAL-PASSED}, \textit{TOTAL-FAILED}, \textit{INDETERMINATE} \quad (8)$$

where t_v is the validation time and the three possible outcomes correspond to the determination states defined in ETSI EN 319 102-1 [4]. The validation process internally performs:

1. Format identification—determine the AdES profile (PAdES, CAdES, XAdES, ASiC) and applicable validation constraints from policy $\mathcal{P}$;
2. Cryptographic verification—verify $\text{Verify}(pk_{EE}, \sigma, H(d)) = true$;
3. Certificate path validation—construct and validate the chain $C = \langle Cert_{EE}, \ldots, Cert_{Root} \rangle$ at time t_v (or at the claimed signing time, depending on the validation model), including revocation checks as described in Section 3.1 (steps 1–4);

4. Time and qualification reasoning—determine the signing time from timestamps or other time evidence, consult Trusted Lists to establish the qualification status of the signing certificate, and apply the policy constraints from $\mathcal{P}$.

In traditional architectures, the validation result R and its supporting evidence (validation report, consulted revocation data, timestamp verification results) are logged internally by the validating party. This evidence is held by a single organization and is not independently verifiable.

Blockchain-augmented validation logging. In the blockchain-augmented model proposed by TALChain [16], the validation process itself is unchanged, and the validator continues to apply ETSI EN 319 102-1 procedures using standard tools. The blockchain layer adds an immutable audit trail by recording a validation event after each validation operation:

$$W = \langle vid, H(d), H(\sigma), R, validatorId, \mathcal{P}.id, t_v \rangle \quad \text{with} \quad tx_W \leftarrow \text{Commit}(\mathcal{L}, W) \tag{9}$$

where vid is a unique validation operation identifier, R is the validation result, $validatorId$ identifies the TSP that performed the validation, $\mathcal{P}.id$ references the applied validation policy, t_v is the validation timestamp, and tx_W is the ledger transaction produced by committing validation record W. TALChain is designed as a permissioned consortium ledger where only accredited TSP nodes can append records, while clients retain the ability to observe the action history, aligning with eIDAS's institutional trust model while leveraging blockchain's immutability and replicated logging [16].

This design positions blockchain as an audit-grade evidentiary layer around signature validation: it records who validated what, when, and under which operational context, reducing disputes and increasing accountability without breaking interoperability with established standards and tools. One key contribution of TALChain is that it does not attempt to reimplement AdES verification rules on-chain, it logs the outcome of standard validation operations, preserving full compatibility with ETSI EN 319 102-1.

The validation audit record W and the preservation anchoring record A_k (defined in Section 5.1) are complementary: A_k proves that a signed object was preserved in a specific state at a given time, while W proves that the object was validated with a specific outcome at a given time. Together, they form a complete evidentiary chain covering both the preservation process and its verification. An auditor examining a preserved signature can cross-reference the preservation anchors $\langle A_1, \ldots, A_n \rangle$ against the validation records $\langle W_1, \ldots, W_m \rangle$ on the shared ledger, obtaining a unified, tamper-evident history of how the signed object was maintained and verified over its lifetime, without depending on the internal logs of any single provider.

Finally, TALChain clarifies a critical tradeoff for any eIDAS-aligned blockchain validation design: moving from public chains to a permissioned PoA consortium avoids crypto-economic volatility and yields predictable performance, but shifts trust to governance and signer management.

6. Blockchain in Timestamping

Digital signature services are a core element of PKIs, since they provide authenticity, integrity, and non-repudiation for electronic data. In regulatory settings such as eIDAS [1], it must be possible to show that a signature was created at a moment when the associated certificate was still valid and had not been revoked. For this reason, timestamping is used as a complementary cryptographic service that binds data to a specific point in time by relying on a trusted time source. In standard implementations, this role is fulfilled by a Timestamping Authority (TSA), which issues a signed timestamp token, typically

in accordance with RFC 3161 [40], to attest that particular data existed at or before a given time.

Blockchain has been explored in this context as a way to strengthen timestamp evidence through distributed and tamper-evident recording mechanisms. Rather than depending solely on a centralized TSA, blockchain-based approaches use a distributed ledger to show that a given hash or timestamp-related proof was anchored before a certain verifiable point in time. This makes blockchain particularly relevant for timestamping scenarios that require stronger auditability, integrity, and protection against undetected modification.

To clarify the distinction between traditional, blockchain-only, and hybrid timestamping approaches reviewed in this section, we formalize the three models.

Model A is represented by traditional RFC 3161 timestamping. A client wishes to obtain a proof-of-existence for a document d at time t. The process follows RFC 3161 [40]:

1. The client computes the digest of the document $h = H(d)$ and sends a timestamp request $Req = \langle h, nonce, policyId \rangle$ to a TSA, where $policyId$ represents an identifier for the TSA's timestamping policy under which the token should be issued;
2. The TSA constructs a TimeStampToken containing the hash h, the serial number sn, and the current time t from its trusted clock, and signs it with its private key sk_{TSA}:

$$TST = \text{Sign}(sk_{TSA}, \langle h, sn, t, policyId \rangle) \tag{10}$$

3. The client receives TST and can later prove that document d existed at time t by presenting TST to any verifier, who checks $\text{Verify}(pk_{TSA}, TST) = true$ and confirms $H(d) = TST.h$, which is the hash field extracted from the token.

The trust assumption is that the TSA operates a synchronized clock, protects sk_{TSA} in an HSM, and is supervised under eIDAS as a QTSP. The resulting token carries the legal presumption of Article 41(2).

Model B, defined as blockchain-only timestamping, and described in decentralized approaches such as OriginStamp [17], specifies that the client anchors the document hash directly on a public blockchain without involving a TSA:

1. The client computes the document digest $h = H(d)$;
2. A timestamping service collects h (possibly batched with other requests into a Merkle tree with root r) and submits a transaction: $tx \leftarrow \text{Commit}(\mathcal{L}, r)$;
3. The proof-of-existence is the tuple $\langle h, \pi, r, tx, b \rangle$, where π is the Merkle inclusion proof and b is the block containing tx.

Verification consists of $\text{VerifyProof}(\pi, h, r) = \text{true}$ and confirming that tx containing r is included in block b at time t_b. The temporal precision is limited to the block granularity (∼12 s on Ethereum, ∼10 min on Bitcoin). No TSA is involved, and no RFC 3161 token is produced. Batching (as in OriginStamp) amortizes cost but increases delay. Chronos [51] and Chronos+ [18] improve temporal accuracy by using block ordering rather than block timestamps.

Model C has its architecture is proposed in [19], and combines both models:

1. Standard issuance: The client sends Req to the TSA, which produces TST exactly as in Model A. The token is returned immediately to the client (latency $\approx$ 93 ms).
2. Asynchronous anchoring: After token issuance, a server-side component computes the hash of the timestamp token $h_{tst} = H(TST)$ and derives an anchor identifier sn_{anchor} from the serial number sn. It then commits an anchoring record to the blockchain:

$$B = \langle sn_{anchor}, h_{tst}, t \rangle \quad \text{with} \quad tx_B \leftarrow \text{Commit}(\mathcal{L}, B) \tag{11}$$

The serial number is partitioned: one part indicates the blockchain environment, the remaining part serves as a lookup identifier during verification [19]. This anchoring is asynchronous, thus it does not delay the token returned to the client.

3. Hybrid verification: A verifier performs standard RFC 3161 verification on the token, $\text{Verify}(pk_{TSA}, TST) = \text{true}$, and may additionally check the blockchain anchor:

$$\text{VerifyProof}(\pi_B, H(tx_B), r_b) = \text{true} \quad \wedge \quad tx_B.h_{tst} = H(TST) \tag{12}$$

If both checks pass, the verifier has two independent sources of evidence: the qualified timestamp (legal presumption under eIDAS) and the blockchain anchor (tamper-evident, publicly verifiable). If only the TSA check passes, the timestamp remains fully valid. If only the blockchain check passes, the evidence is weaker (no qualified status) but still demonstrates existence. The three models are summarized in Table 2.

Table 2. Comparison of timestamping approaches: traditional, blockchain-only, and hybrid.

Property	Model A	Model B	Model C
Standards compliance	RFC 3161 [40], ETSI EN 319 421/422 [35,41]	None	RFC 3161 [40], ETSI EN 319 421/422 [35,41]
Legal status (eIDAS)	Qualified electronic time stamp (Art. 41(2))	No qualified status; admissible under Art. 46	Qualified token + supplementary blockchain evidence
Trust assumption	Supervised TSA with synchronized clock	Public blockchain consensus	Supervised TSA + blockchain consensus
Temporal precision	Sub-second (TSA clock)	Block granularity (12 s Ethereum; 10 min Bitcoin)	Sub-second (TSA clock)
User-perceived latency	93 ms	12 s (Ethereum) to 10 min (Bitcoin); batching adds delay	93 ms (anchor is async)
Independent verifiability	Verify TSA signature + certificate chain	Verify against public ledger	Both: TSA verification + Merkle proof
Single point of failure	TSA	None (decentralized)	TSA for qualified status; blockchain for supplementary proof
Interoperability	Full (standard token format)	None (custom proof format)	Full (token unchanged; blockchain proof is additive)
Representative works	RFC 3161 [40], ETSI EN 319 421 [35]	OriginStamp [17], Chronos [51], Chronos+ [18], Estevam et al. [52]	Brinzea et al. [19]

6.1. Blockchain-Based Timestamping Approaches

Existing work in this area generally follows two main directions. The first uses the ledger as the primary timestamping mechanism, usually by anchoring hashes directly on a public chain and treating that recording as proof of existence. The second combines conventional timestamping services with blockchain anchoring, using the ledger as an extra verifiable layer rather than as a complete substitute for the TSA. The first direction emphasizes decentralization, while the second aims to preserve compatibility with established trust service infrastructures.

Among the earliest and most influential examples is OriginStamp [17], which uses the Bitcoin blockchain [53] to publish aggregated hashes of submitted data. The main idea is to collect multiple requests, combine them efficiently, and anchor the resulting proof on

a public blockchain. This reduces the number of on-chain transactions and demonstrates that blockchain can be used as a durable proof-of-existence mechanism without depending on a centralized TSA. The approach is relevant because it showed that public blockchains can support decentralized timestamping in practice. However, it also reflects a typical trade-off: grouping multiple requests together lowers costs, but reduces the immediacy and temporal precision of each submission. If a real-time recording is needed, individual anchoring remains possible, but at an increased cost.

Later work moved toward more practical concerns, especially timestamp accuracy, confirmation delay, and overall efficiency. For example, Chronos [51] does not rely directly on the timestamp written in the block, but instead uses the ordering of blocks to estimate time more reliably. This matters because block timestamps may be affected by network conditions and are not always sufficiently precise on their own. Chronos+ [18] extends that idea further through batch processing and performance evaluation, with the aim of reducing the interval between the user request and final blockchain confirmation. Taken together, these proposals show that blockchain-based timestamping is not only about placing data on-chain, but also about how temporal evidence is derived and how efficiently the system can operate. They also show that the practical value of blockchain timestamping depends not only on immutability but also on how accurately and quickly verifiable time evidence can be produced. However, these improvements do not eliminate the inherent limits of ledger-based timestamping, since final confirmation still depends on the blockchain itself and the resulting evidence does not automatically have the same status as a traditional TSA-issued timestamp.

Other proposals have explored the use of smart contracts and decentralized reporting mechanisms on Ethereum-based infrastructures [52]. In these models, the blockchain is used not only as a passive storage medium, but also as an execution environment for timestamp-related logic. These designs use multiple observers or additional time sources to improve timestamp accuracy and reduce reliance on a single source. This reflects a broader direction in blockchain timestamping, namely the attempt to combine decentralization with more reliable time evidence.

Although these approaches demonstrate the technical potential of blockchain in this area, they do not fully reproduce the role of a traditional TSA. A public ledger can provide strong proof that data was anchored before a certain moment, but it does not inherently offer the same guarantees regarding trusted time sources, standardized token structures, or legal recognition. This distinction becomes especially important in regulated environments. Qualified timestamping services operate under strict technical and organizational requirements and issue standardized tokens that remain interoperable with existing PKI processes. Blockchain-only schemes do not automatically satisfy those conditions.

For this reason, hybrid models have become increasingly relevant. In such approaches, a traditional TSA continues to issue a conventional token, while blockchain is used to anchor additional evidence related to that token or to its issuance. This preserves compatibility with existing validation mechanisms and at the same time adds a distributed and independently verifiable audit layer. From a trust service perspective, this hybrid direction is more practical than a full replacement model because it combines the legal and operational maturity of conventional timestamping with the integrity and transparency benefits of blockchain.

Overall, current work in this area points to two main categories: decentralized proof-of-existence models and hybrid timestamping architectures. The first shows how blockchain can provide durable and tamper-evident evidence without a central authority, while the second is more appropriate where interoperability, trusted time information, and compatibility with established trust service models remain essential. This is why hybrid approaches ap-

pear especially relevant in regulated settings where technical assurance must be combined with compliance and operational usability.

6.2. A Hybrid Blockchain-Assisted Timestamping Architecture

A hybrid blockchain-assisted timestamping architecture is presented in [19]. The proposed model combines a traditional TSA with blockchain anchoring while maintaining the standard issuance flow. Rather than replacing the TSA with a fully decentralized mechanism, the architecture continues to rely on the TSA for token generation, while using blockchain as an additional integrity and audit layer. This is particularly relevant because many blockchain-only timestamping proposals provide tamper-evident proof that data existed before a certain moment, yet do not remain fully interoperable with established PKI-based timestamping and signing processes. By contrast, the architecture described in [19] preserves the conventional token format and introduces blockchain only as a supplementary verification mechanism.

Another important characteristic of the solution is that it is not restricted to a single blockchain model. The architecture supports both Ethereum [54], used as a public blockchain environment, and Hyperledger Fabric [55], used as a permissioned blockchain infrastructure. This makes the solution adaptable to both open verification scenarios and enterprise settings where controlled participation, governance, and privacy requirements are more important. From a technical perspective, the solution presented in [19] preserves the standard timestamp issuance workflow. The client computes or provides the document hash and sends it to a server-side component that builds an RFC 3161-compliant request. That request is forwarded to the TSA, which processes it in the usual way by generating a timestamp token, assigning a serial number, and signing the token with its private key. As a result, the issued token remains compatible with conventional PKI-based services.

The blockchain-related extension is introduced after the token has been generated. Instead of saving the full token on-chain, the architecture computes a hash of the complete timestamp response and anchors that hash together with an identifier derived from the serial number. This reduces the amount of on-chain data, avoids exposing the token itself, and still provides a verifiable integrity proof linked to the original response.

A technically useful element of the architecture is the way the timestamp token serial number is structured so as to connect the standard TSA workflow with the blockchain layer. In the implementation described in [19], the serial number is 20 bytes long: the first 4 bytes encode the target blockchain network (e.g., ETH: for Ethereum), while the remaining 16 bytes serve as a unique identifier for lookup. When the token is generated, the TSA-side blockchain module computes a SHA-256 hash of the full timestamp response and stores on-chain a key-value pair of the form $SN(16bytes) : Hash(timestamptoken)$. During verification, the verifier extracts the serial number from the generated timestamp token, determines the blockchain environment from its first 4 bytes, and uses the remaining 16 bytes to retrieve the anchored hash. In this way, the verifier can locate the corresponding blockchain evidence without changing the timestamp token structure or introducing a non-standard validation format.

Another practical feature is asynchronous anchoring. The timestamp token can be returned to the client immediately after it is signed by the TSA, while the blockchain write operation takes place in parallel. This avoids adding significant latency to issuance and keeps the client-facing behavior close to that of a conventional timestamping service.

The model described in [19] also brings several practical advantages. First, it preserves interoperability with existing PKI-based timestamping and signing services by leaving the token format unchanged. Second, it adds an external and tamper-evident integrity layer beyond the internal records of the TSA. Third, it improves auditability,

since timestamp-related evidence can be checked independently on the blockchain. Finally, because only token-derived hashes and identifiers are stored on-chain, the architecture improves traceability without exposing the underlying document or the full token contents.

6.3. Standards Compliance and Trust Service Relevance

A relevant characteristic of the architecture presented in [19] is that it extends timestamping with blockchain anchoring while preserving compatibility with existing standards. while preserving compatibility with existing standards. Since the TSA continues to issue RFC 3161-compliant timestamp tokens, the solution remains compatible with established validation workflows and with traditional PKI-based trust service infrastructures. In practice, timestamp generation and validation rely on RFC 3161 [40] and RFC 5816 [56], while qualified services must also align with ETSI requirements such as EN 319 421 [35], EN 319 422 [41], and EN 319 401 [2]. Because the TSA in this architecture continues to issue standard-compliant timestamp tokens, the solution remains compatible with established PKI validation workflows and trust service infrastructures.

In this model, blockchain is not intended to replace the TSA or the existing standards-based trust framework, but to complement them with an additional integrity and audit layer. The legal and technical basis of the timestamp still relies on the trust service itself and on compliance with the relevant RFC, ETSI, and eIDAS requirements, while blockchain adds an extra layer of traceability and independent verification. For this reason, the hybrid architecture described in [19] is more suitable for practical trust service adoption than blockchain-only timestamping schemes that do not preserve standard interoperability.

7. Blockchain in Content Provenance and Authenticity

The integrity of digital media has become one of the defining challenges of the current technological era. The convergence of increasingly accessible generative AI tools, social media amplification, and the erosion of traditional editorial gatekeeping has created an environment in which misinformation, disinformation, and synthetic content spread at an unprecedented scale. Studies have established that misleading information poses systemic risks not only to individual decision-making but also to democratic institutions, financial markets, and supply chain resilience [57,58]. In the EU, this challenge has been recognized at the highest policy level, with the European Commission explicitly promoting the development of technical mechanisms for authenticating information flows and tracing content to verified sources.

Against this backdrop, the field of Content Provenance and Authenticity (CPA) has emerged as a multidisciplinary endeavor at the intersection of cryptography, distributed systems, artificial intelligence, and regulatory policy. CPA aims to establish verifiable, tamper-evident chains of custody for digital assets, enabling any party to determine who created a piece of content, with which tools, at what time, and what modifications have been applied since its origin. The premise is straightforward: if the provenance of a digital asset can be credibly traced to a trusted source, its perceived authenticity increases significantly. Achieving this at scale, however, requires a combination of standardized metadata formats, cryptographic integrity guarantees, and a trust infrastructure that does not depend on any single authority, requirements for which blockchain technology offers a compelling set of capabilities.

Content provenance shares foundational mechanisms with the trust service domains reviewed in the preceding sections: hash anchoring (Section 5), timestamping (Section 6), and certificate-based trust chains (Section 3). However, CPA introduces distinct requirements that do not arise in traditional trust services: the need to handle diverse media formats (image, video, text), the challenge of detecting near-duplicate or derivative content

(where cryptographic hashes are insufficient), the integration of AI-based detection alongside cryptographic verification, and the need to operate at internet scale with content that is frequently copied, compressed, and re-encoded. Table 3 positions blockchain-based CPA within the landscape of trust service primitives established in earlier sections, highlighting both the shared mechanisms and the CPA-specific extensions.

Table 3. Content provenance approaches in the context of eIDAS trust service primitives.

Aspect	Trust Service Domains	C2PA Standard (PKI-Based)	Blockchain-Based CPA	QTSP-Integrated CPA
Primary trust anchor	CA hierarchy + Trusted Lists	CA-issued certificate (web PKI)	Blockchain consensus (public or permissioned)	Qualified electronic seal (QTSP on Trusted List)
Integrity mechanism	Certificate chain, CRL/OCSP, qualified timestamps	C2PA Manifest: assertions + claim + claim signature	On-chain hash or perceptual hash anchoring + Merkle proofs	Qualified seal bound to on-chain smart contract [59]
Content types	Certificates, signatures, timestamps, preservation evidence	Images, video, audio, documents (C2PA-supported formats)	Images, video, text (multi-format) [60]; fintech documents [61]	Same as C2PA, with eIDAS legal binding
Near-duplicate detection	Not applicable (exact-match hashes)	Not native (relies on exact Manifest integrity)	Perceptual hashing for similarity detection [60,62]	Not yet addressed
AI/deepfake detection	Not applicable	Not native	Hybrid architectures combining BC anchoring + AI models [63]	Not yet addressed
Legal status (eIDAS)	Qualified services carry legal presumption (Art. 25, 35, 41)	No eIDAS status; voluntary industry standard	No eIDAS status; proof-of-existence only	Qualified electronic seal (Art. 35(2)); legally presumptive across 27 Member States
Single point of failure	CA/TSA (mitigated by Trusted List supervision)	Issuing CA + signer key material	None (decentralized anchoring)	QTSP (supervised, audited)
Verification model	Standard PKI validation (ETSI EN 319 102-1)	C2PA Manifest verification against signer certificate	On-chain proof verification + optional off-chain AI scoring	Standard qualified seal verification + on-chain trust chain from LOTL [59]
Scalability for internet content	Designed for trust service operations (millions/day)	Designed for content platforms (billions of assets)	Variable; limited by blockchain TPS	Inherits QTSP infrastructure + blockchain anchoring
Maturity	Production (decades of deployment)	Pilot/early production (Adobe, Microsoft, BBC adopting)	Conceptual/Prototype [60,61,63]	Conceptual (architecture proposed in [59])

The comparison in Table 3 reveals that blockchain-based CPA extends the same foundational primitives used across the trust service domains reviewed in this paper, hash anchoring, timestamping, and certificate-based trust chains, but adapts them to the specific requirements of digital media at internet scale. The key differentiators are the need for perceptual hashing to handle content transformation (a requirement absent from traditional trust services), the integration of AI-based detection models as a complement to cryptographic verification, and the opportunity to bind provenance records to eIDAS-qualified electronic seals through smart contracts anchored to the European Commission's List of Trusted Lists [59]. This last approach is particularly significant because it would elevate content authenticity attestation from a voluntary technical measure to a component of Eu-

rope's regulated trust infrastructure, providing legally presumptive provenance evidence across all 27 Member States.

7.1. The C2PA Standard and the Provenance System

The most significant standardization effort in the CPA domain is the Coalition for Content Provenance and Authenticity (C2PA), a Joint Development Foundation project formed through an alliance of Adobe, Arm, Intel, Microsoft, and Truepic [64]. The C2PA specification, on track for adoption as an ISO standard, defines a technical architecture for attaching cryptographically signed metadata, termed Content Credentials, to digital assets [42]. Each Content Credential consists of a set of assertions (statements about actions performed on the asset), a claim that references those assertions, and a claim signature generated using the actor's private key. These components form a C2PA Manifest, which is embedded directly within the asset file or stored externally with a binding reference.

Despite its growing maturity, the C2PA framework relies on centralized PKI infrastructure and does not natively incorporate blockchain or distributed ledger technology. This is a deliberate design choice that prioritizes compatibility with established web PKI ecosystems. However, as several researchers have observed, the lack of a decentralized anchoring mechanism introduces a single-point-of-trust dependency: the integrity of Content Credentials ultimately rests on the trustworthiness of the issuing Certificate Authority (CA) and the security of the signer's key material. Blockchain-based extensions to C2PA have been proposed precisely to address this limitation by providing an immutable, distributed record of provenance events that can be independently verified without reliance on any single authority.

7.2. Blockchain-Based Approaches to C2PA

The application of blockchain to content authenticity verification has been explored through a growing body of research. Early proposals focused on the conceptually straightforward approach of storing cryptographic hashes of digital content on a distributed ledger, thereby creating a timestamped, immutable proof that a specific version of an asset existed at a given moment. While elegant in its simplicity, this approach suffers from a fundamental limitation: because hash functions are deterministic and collision-resistant, even a single-pixel modification to an image produces an entirely different hash, rendering exact-match verification useless for detecting near-duplicate or derivative content [60]. Their architecture is notable for being the first to simultaneously combine fake content prevention, distributed proof of origin, content-format versatility across text, image, and video, and C2PA standard compliance within a single system. The framework's design as an attestation service, rather than a standalone platform, is particularly relevant to its potential integration within the eIDAS trust services ecosystem.

Subsequent work has introduced more sophisticated techniques to overcome this limitation. Perceptual hashing algorithms, which produce compact binary representations of visual or textual content that remain similar for perceptually similar inputs, have been integrated into blockchain-based systems to enable similarity detection alongside exact-match verification [60,62].

A more recent development is the emergence of decentralized content verification frameworks that combine blockchain anchoring with attestation services. The VeriNet framework, for instance, leverages the Ethereum Attestation Service to integrate on-chain and off-chain attestations for content verification, with proof-of-concept implementations in both deepfake detection and fintech credit scoring [61]. Similarly, hybrid architectures that combine provenance signals anchored via C2PA Content Credentials with blockchain

hash anchoring and AI-based detection models have shown measurable improvements in deepfake identification accuracy [63].

7.3. Alignment and Integration with the eIDAS Trust Ecosystem

Content provenance metadata anchored on a qualified electronic ledger operated by a QTSP would carry legally recognized evidentiary value across all 27 EU Member States. A content consumer verifying an asset's provenance would not merely be checking a technical claim against a distributed database. They would be relying on a legally presumptive record operated under the supervisory framework that governs all eIDAS qualified trust services. This elevates content authenticity attestation from a voluntary technical measure to a component of Europe's regulated trust infrastructure.

The practical realization of blockchain-based CPA within the eIDAS framework depends on integration with several complementary elements of the European trust ecosystem. Trust Service Providers that already offer remote QES could extend their service portfolios to include content provenance attestation with minimal additional infrastructure, since the core operations, key management, digital signing, timestamp generation, and certificate lifecycle management, are already part of their supervised workflows. This integration pathway is particularly significant because TSPs operate across multiple economic sectors and maintain established client relationships, offering a natural distribution channel for CPA capabilities.

Recent architectural research has demonstrated how qualified electronic seals issued by eIDAS QTSPs can be cryptographically bound to smart contracts on public blockchains, establishing machine-verifiable trust chains from the European Commission's List of Trusted Lists (LOTL) down to individual on-chain addresses [59]. This mechanism could be adapted to create verifiable links between CPA provenance records and the legal identities of content creators.

8. Blockchain in EUDI Wallet

Blockchain and distributed ledger technologies (DLTs) have emerged as compelling enablers for several critical functions within the EUDI Wallet ecosystem, particularly trust registry management, credential revocation, and decentralized Public Key Infrastructure. This chapter focuses specifically on the role of DLTs within the EUDI architecture, analyzing how blockchain-based infrastructure can satisfy the tamper-evidence and decentralization requirements that SSI principles demand while operating within the strict regulatory boundaries imposed by eIDAS 2.0 and GDPR. The analysis proceeds from the architectural alignment between SSI principles and the EUDI trust model, through the EBSI as the EU's concrete blockchain identity layer, to credential revocation mechanisms, privacy-preserving unlinkable revocation, and the open technical challenges at this intersection.

8.1. SSI Foundations and the EUDI Trust Model

EUDI Wallet, mandated under the revised eIDAS Regulation (EU 2024/1183), represents the most ambitious deployment of VC infrastructure to date, aiming to provide every EU citizen and resident with a state-backed digital identity wallet by 2026. The Architecture and Reference Framework (ARF) [43] defines a multi-stakeholder ecosystem comprising Wallet Providers, Person Identification Data (PID) Providers, Qualified Electronic Attestation of Attributes (QEAA) Providers, Relying Parties, and Trusted List Providers—all interconnected through standardized issuance (OpenID4VCI) and presentation (OpenID4VP) protocols. While the ARF itself remains technology-neutral regarding the underlying trust infrastructure, blockchain and distributed ledger technologies (DLTs) have emerged as compelling enablers for several critical functions within this architecture,

particularly trust registry management, credential revocation, and decentralized public key infrastructure.

The conceptual underpinnings of the EUDI Wallet align substantially with Self-Sovereign Identity (SSI) principles as formalized by Preukschat and Reed [65], who identify blockchain as one of the three foundational pillars of SSI alongside VCs and DIDs. In the canonical SSI architecture, blockchain serves as the Verifiable Data Registry (VDR)—a decentralized, tamper-evident layer for anchoring DID documents, credential schemas, revocation registries, and issuer accreditations without reliance on centralized authorities. The ten SSI principles—existence, persistence, protection, control, access, transparency, consent, minimization, portability, and interoperability—map directly onto EUDI design requirements: user sole control over cryptographic material stored in Wallet Secure Cryptographic Devices (WSCDs), selective disclosure via SD-JWT and mdoc formats, explicit user consent for every presentation, and cross-border interoperability through standardized protocols. Sedlmeir et al. [66] further contextualizes this paradigm shift by distinguishing VC-based wallets from centralized identity silos, noting that DIDs enable end-to-end encrypted communication and verifiable trust establishment without intermediaries, properties essential for a pan-European identity infrastructure.

However, the current EUDI architecture diverges from pure SSI in significant ways. The ARF anchors trust in a hierarchical PKI model centered on Trusted Lists conforming to ETSI TS 119 612 [44], where Member State-appointed Trusted List Providers publish signed, machine-readable registries of authorized Wallet Providers, PID Providers, and QTSPs [43]. This approach, while providing strong legal certainty and regulatory compliance, introduces centralized trust anchors rather than the fully decentralized registries envisioned in SSI literature. The tension between regulatory assurance and decentralization is a defining characteristic of the EUDI ecosystem and motivates the exploration of blockchain-based alternatives that can satisfy both imperatives.

8.2. EBSI as the European Blockchain Identity Layer

The European Blockchain Services Infrastructure (EBSI) [20] represents the EU's primary effort to bridge this gap. Established in 2018 by the European Blockchain Partnership and now governed by the EUROPEUM-EDIC, EBSI operates as a public, permissioned blockchain network built on Hyperledger Besu—an enterprise-grade, EVM-compatible Ethereum client employing the Quorum Byzantine Fault Tolerant (QBFT) consensus mechanism. The network currently comprises approximately 40 nodes distributed across EU Member States, with an 8 s block time and energy consumption averaging approximately 100 Wh per node. EBSI exposes a suite of core service APIs: the DID Registry API (v5) for registering and resolving did:ebsi identifiers for legal entities, the Trusted Issuers Registry (TIR) API (v5) for managing issuer accreditations as on-chain Attribute Envelopes via smart contracts, the Trusted Schemas Registry (TSR) for credential schema validation, and the Timestamp API for on-chain notarization.

EBSI implements a hierarchical trust chain comprising Root Trusted Accreditation Organisations (Root TAOs), Trusted Accreditation Organisations (TAOs), and Trusted Issuers (TIs). credentials—VerifiableAccreditationToAccredit for TAOs and VerifiableAccreditationToAttest for TIs—are stored in the TIR smart contract, and verifiers traverse this chain by recursively dereferencing termsOfUse references until reaching a VerifiableAuthorisationForTrustChain anchored to the Root TAO. This mechanism provides real-time, tamper-evident governance of issuer permissions—a substantial improvement over periodic XML-based Trusted List updates under ETSI TS 119 612. For natural persons, EBSI employs did:key, a self-resolving DID method that encodes the public key directly in the identifier, requiring no on-chain registration and ensuring GDPR compliance by keeping

personal data entirely off-chain. This hierarchical trust model is illustrated in Figure 1, which presents an example from the education domain.

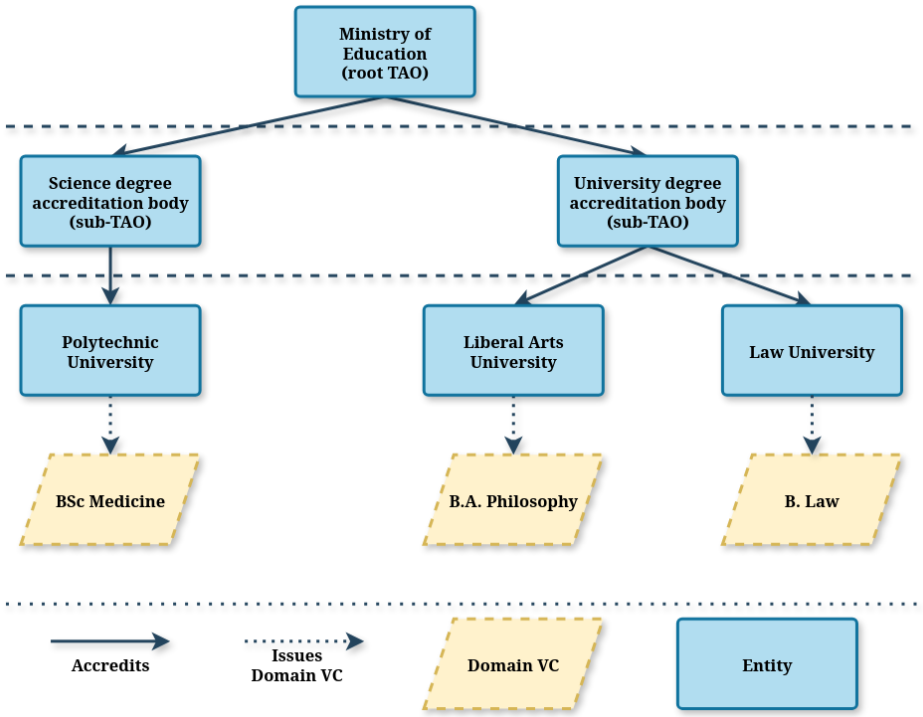


Figure 1. Trust chain example for the education domain.

The alignment between EBSI and EUDI Wallet protocols is deliberate: both mandate OpenID4VCI [45] for credential issuance and OpenID4VP [46] for verifiable presentations, and the EBSI-VECTOR project is actively developing a migration path from EBSI-conformant wallets to fully EUDI-certified implementations. Ou et al. [67] demonstrate a complementary SSI blockchain framework for financial institutions, where smart contracts enforce fine-grained access control over customer credit data—illustrating how blockchain-based identity governance extends beyond public administration into regulated private-sector applications. The applicability of multi-chain architectures to EU-scale identity infrastructure also merits consideration. Wood's Polkadot framework [68] introduces the relay chain paradigm for heterogeneous multi-chain interoperability, where independent parachains—each with sovereign state-transition functions—share security through a coordinating relay chain. This model offers an architectural analogy for EU identity infrastructure: a shared European trust layer (analogous to the relay chain) coordinating sovereign national identity systems (parachains) while enabling cross-border credential verification through standardized cross-consensus messaging. Although EBSI currently operates as a single-chain network, the design principles of shared security without centralization and heterogeneous chain interoperability address scalability and sovereignty challenges that will intensify as EUDI adoption scales across 27 Member States.

8.3. Blockchain-Enabled Credential Revocation Mechanisms

Credential revocation represents perhaps the most technically consequential application of blockchain within the EUDI ecosystem. The fundamental challenge is that revocation data must be:

- Tamper-evident to prevent unauthorized status manipulation;
- Universally accessible to enable offline or near-real-time verification;

- Privacy-preserving to prevent correlation and tracking of credential holders across verification events.

Traditional revocation mechanisms—CRLs and the OCSP—suffer from the well-documented need to "phone home" problem, where each status check requires contacting the issuer's server, thereby enabling the issuer to track when, where, and by whom credentials are verified.

Huynh et al. [69] provide a systematic survey of blockchain-based revocation solutions, identifying the fundamental tension between blockchain immutability and the operational necessity of credential revocation. The literature reveals a spectrum of approaches with distinct cryptographic foundations and privacy properties. At the simplest end, the World Wide Web Consortium (W3C) Bitstring Status List (formerly StatusList2021) assigns each credential a fixed index position within a compressed bitstring published as a signed VC; revocation flips the corresponding bit. EBSI supports this mechanism alongside several alternatives including CRLBloomFilter2023, DynamicSLBloomFilter2023, and one-time status credentials European Commission. While computationally efficient, bitstring approaches provide only "herd privacy"—the large bitstring (minimum 131,072 bits) obscures which specific credential is being checked, but the static index position constitutes a persistent correlation vector across presentations.

Adja et al. [70] propose a more robust architecture in which each Certificate Authority deploys a dedicated smart contract managing whitelists (issued certificates) and blacklists (revoked certificates). Revocation distribution points are represented as Bloom filters stored on a public blockchain, with integrity guaranteed through Merkle tree verification: the hash of each Bloom filter serves as a leaf in a Merkle tree, and the CA signs the Merkle root using ECDSA. Clients retrieve the appropriate Bloom filter, verify its integrity against the signed Merkle root, and perform a membership test to determine revocation status. This decentralized approach eliminates single points of failure inherent in centralized CRL distribution while providing cryptographic tamper-evidence through the combination of Merkle trees and blockchain immutability.

Xu et al. [71] advanced the state of the art by proposing the first revocation mechanism that simultaneously achieves privacy preservation and tamper-evidence for W3C VCs. Their construction combines BBS+ signatures for credential issuance with selective disclosure, pairing-based cryptographic accumulators for representing the non-revoked credential set as a constant-size value and a role-based blockchain (Hyperledger Iroha) for tamper-evident storage of accumulator updates. The holder generates a zero-knowledge proof of non-membership in the revocation set using their witness and the on-chain accumulator value, which the verifier checks against the blockchain state. This approach eliminates join-revoke linkability—a vulnerability in Sovrin's CKS accumulator where observers can correlate credential additions and removals—while adding only approximately 42.86 ms to presentation overhead and 31.36 ms to verification overhead.

Fang et al. [72] extend this direction with a distributed identity management scheme employing Pointcheval–Sanders (PS) signatures, which support re-randomizable signatures and efficient zero-knowledge proofs of knowledge. Their framework introduces hierarchical revocation via dynamic accumulators managed through smart contracts, combined with threshold traceability using Verifiable Secret Sharing (VSS)—enabling $\tau + 1$ regulatory authorities to recover holder identity in exceptional circumstances while maintaining full anonymity under normal operations. In threshold cryptography, $\tau + 1$ refers to the minimum number of parties required to reconstruct a secret or perform a cryptographic operation. For example, if the system uses a $(\tau, n) = (2, 5)$ threshold scheme, then 5 is the total number of parties and τ is the threshold; any 3 out of 5 authorities must come together to recover a holder's identity. This conditional privacy model addresses a crucial

regulatory requirement for the EUDI ecosystem, where law enforcement access must be balanced against citizen privacy rights under GDPR.

8.4. Privacy-Preserving Unlinkable Revocation

The most directly relevant contribution to the EUDI Wallet revocation challenge comes from Bureacă et al. [73], who propose an unlinkable revocation framework specifically designed for Qualified Electronic Attestations of Attributes within the eIDAS 2.0 regulatory context. Their approach publishes daily randomized revocation lists as cascaded Bloom filters—a technique derived from Larisch et al.'s CRLite [74] construction that eliminates the false-positive problem inherent in standard Bloom filters by layering multiple filters in cascade. Critically, the revocation indexes are randomized daily, ensuring that the same credential produces different revocation tokens across verification sessions and thereby preventing temporal correlation by verifiers or colluding parties. This mechanism extends EBSI's open-source infrastructure and addresses a specific gap identified in the existing EBSI Bloom filter approach, where verifiers cannot independently validate holder-provided tokens, potentially enabling malicious holders to forge arbitrary status assertions. Table 4 compares blockchain-based credential revocation approaches for the EUDI Wallet ecosystem.

Table 4. Comparison of blockchain-based credential revocation approaches for the EUDI Wallet ecosystem.

Method	Blockchain Layer	Cryptographic Basis	Privacy Model	Verifier Unlinkability	Issuer-Verifier Unlinkability	EUDI/EBSI Alignment
W3C Bitstring Status List	Off-chain VC	Compressed bitstring; signed VC	Herd privacy only	Partial	No	Native EBSI support
Bloom Filter, Merkle Tree [70]	Public blockchain (CA smart contract)	Bloom filter; ECDSA; Merkle proof	Probabilistic membership	Partial	No	CRL-BloomFilter-2023 (EBSI)
Cascaded Bloom Filter [73]	Ethereum + IPFS (ledger-agnostic design)	Cascaded Bloom filters; BBS+ signatures; ZKP; daily re-randomized indices	Temporal unlinkability (verifier-side)	✓	No	ARF-compatible; extends EUDI open-source reference implementation
ZKP Accumulator + BBS+ [71]	Hyperledger Iroha	BBS+ signatures; pairing-based accumulator; ZKP non-membership proof	Full unlinkability	✓	✓	Candidate for TS13/TS14; not yet standardised
PS Signatures + Accumulator + VSS [72]	Smart contract (generic)	Pointcheval–Sanders signatures; VSS threshold traceability	Full anonymity + conditional traceability (τ+1)	✓	✓	Regulatory-aligned; not yet adopted
EBSI TIR (on-chain accreditation) [20]	Hyperledger Besu (EBSI)	Smart contract; VerifiableAccreditation VCs; QBFT consensus	No revocation privacy	No	No	Native EBSI

Note: ✓ indicates that the corresponding property is supported.

Nevertheless, these mechanisms achieve only verifier unlinkability; they do not prevent issuer-verifier collusion, since the issuer knows all credential instances it generated. Full unlinkability—where no party, including PID Providers or Attestation Providers colluding with Relying Parties, can determine whether selectively disclosed attributes describe the same user—requires zero-knowledge proof-based signature schemes such as BBS+ signatures, which enable a holder to derive an unlinkable proof from a signed credential that reveals only selected attributes without exposing the underlying signature. ARF treats this limitation not merely as a noted gap but as an active area of standardization work: dedicated technical specifications (TS4, TS13, and TS14) now address ZKP (zero-knowledge proof) integration across two broad approaches—arithmetic-circuit-based schemes such

as zk-SNARKs, and multi-message signature schemes that natively support unlinkable selective disclosure, including BBS+, PS multi-signatures, and CL-signatures—informed by the analysis in ETSI TR 119 476 [75]. However, significant barriers to deployment remain. BBS+ signatures require pairing-friendly curves (BLS12-381) and impose higher computational costs on mobile devices. The IETF standardization of BBS+ remains incomplete, and none of the candidate zk-SNARK schemes have yet achieved formal standardization. SOG-IS approval of the underlying cryptographic primitives—required for Level of Assurance High compliance under EU CIR 2015/1502—has not been obtained for pairing-based curves. Additionally, recent technical scrutiny of TS14 proposals has raised concerns about the absence of formal security proofs for some proposed BBS-based constructions, and the challenge of curve bridging for holder binding has not been fully resolved. These obstacles create an ongoing tension between the strong privacy guarantees that ZKP-based credentials would provide and the practical and regulatory constraints on their deployment within the EUDI Wallet ecosystem.

8.5. Technical Challenges at the Intersection of Blockchain and EUDI

Several technical challenges remain at the intersection of blockchain technology and EUDI Wallet deployment. First, scalability constraints of EBSI's single-chain Hyperledger Besu architecture, with its 8 s block time and approximately 40 validator nodes, are adequate for current identity and credential operations but may prove insufficient as the system scales to serve hundreds of millions of EU citizens. Sharding or multi-chain architectures, drawing on principles from Polkadot's parachain model, could address throughput limitations while preserving the sovereignty of national identity systems. Second, the DID method fragmentation between did:ebsi (legal entities), did:key (natural persons), and alternative methods such as did:web creates interoperability challenges that universal resolver infrastructure must address. Third, GDPR constraints fundamentally prohibit storing personal data on-chain, requiring all personally identifiable information to remain in user-controlled wallets while only pseudonymous references, accumulator values, or hashed revocation artifacts are anchored to the blockchain. Fourth, the evolving specification landscape—with rapid API version changes and breaking updates across EBSI services—imposes a significant integration burden on wallet developers and credential issuers.

The convergence of blockchain infrastructure with the EUDI Wallet ecosystem ultimately reflects a pragmatic synthesis: blockchain provides the tamper-evident, decentralized trust layer that SSI principles demand, while the regulatory framework of eIDAS 2.0 supplies the legal certainty and governance structure that pure decentralization cannot. The ongoing research into privacy-preserving revocation mechanisms, zero-knowledge credential presentations, and cross-chain interoperability is progressively closing the gap between the theoretical ideals of Self-Sovereign Identity and the operational realities of pan-European digital identity infrastructure.

9. Quantitative Performance and Cost Analysis

The preceding sections have examined how blockchain can complement traditional trust service architectures across six operational domains: certificate management, signature services, preservation, validation, timestamping, and credential revocation within the EUDI Wallet ecosystem. While each proposal offers qualitative arguments for improved transparency, auditability, and distributed trust, the practical viability of these approaches depends on measurable operational characteristics. This section consolidates quantitative evidence drawn from the reviewed literature, published platform benchmarks, and publicly available infrastructure pricing to provide a cross-domain performance comparison

between blockchain-augmented and traditional trust service architectures. Four metrics are compared:

- Latency: time from request submission to confirmed response;
- Throughput: operations processed per unit time under load;
- Storage cost: cost per operation or per GB for persisting trust evidence;
- Audit efficiency: time or number of verification steps required to confirm the integrity of a historical event after the fact.

Where blockchain proposals follow a hybrid architecture, preserving the traditional service for the user-facing operation and anchoring evidence asynchronously, we report both the user-perceived latency and the end-to-end anchoring latency separately.

9.1. Performance Comparison by TSP Domain

Table 5 summarizes the latency overhead introduced by blockchain across the six TSP domains reviewed in this paper. For each domain, the table identifies the traditional mechanism that serves as the operational baseline (e.g., OCSP for certificate revocation, RFC 3161 for timestamping, relational databases for audit logging) and the blockchain-augmented alternative proposed in the reviewed literature.

The traditional baseline latencies are drawn from published measurements. OCSP response times reflect the median of 20 ms reported by Zhu et al. in a large-scale measurement study that found 94% of OCSP queries are now served by CDNs [76]. Prior to CDN adoption, OCSP latencies ranged from 186 to 776 ms across major CAs [77]. The TSA's issuance latency is around 93 milliseconds [19]. Database write and read latencies (0.1–0.5 ms for INSERT, 1–3 ms for indexed SELECT) represent local-connection performance documented in PostgreSQL benchmarks [78]. Object store latencies (100–200 ms first-byte) correspond to the performance documented by AWS for S3 Standard [79].

Table 5. Performance overhead of blockchain-augmented trust services relative to traditional baselines.

Domain	Traditional Baseline	Traditional Latency	Blockchain Alternative	BC Latency	Overhead Factor	Primary BC Benefit	Hybrid Viable
Certificate revocation	OCSP responder (CDN)	20 ms	On-chain status registry	4–12 s	200–600×	Tamper-evident publication; prevents silent CRL suppression	Yes (local filters + BC anchor)
Signing audit trail	Relational DB (PostgreSQL)	0.1–0.5 ms	Permissioned ledger [48]	1–12 s	2000–24,000×	Independent verifiability; evidence survives provider shutdown	Yes (async write)
Preservation anchoring	Object store (S3)	100–200 ms	IPFS + Ethereum [13,49]	12 s	60–120×	Content-addressed integrity; public verifiability	Yes (async anchor)
Validation audit trail	Relational DB	1–3 ms	Consortium ledger [16]	50–200 ms	20–100×	Cross-provider portability; dispute-grade evidence	Yes (async logging)
Timestamping	RFC 3161	93 ms	Hybrid TSA + blockchain [19]	93 ms (user) + 8–12 s (anchor)	1× (user)/16–60× (anchor)	Decentralized proof-of-existence; removes single-TSA dependency	Yes (async anchor)
Credential revocation (EUDI)	OCSP	20 ms	Accumulator + blockchain [71]	74 ms	3.7×	Privacy-preserving unlinkable verification + tamper evidence	Yes (local check + BC registry)

Blockchain latencies are derived from protocol-level parameters of the platforms referenced in the reviewed papers. Ethereum L1 operates with a fixed 12 s slot time under Proof-of-Stake and achieves economic finality after two epochs. Each epoch comprises 32 slots

of 12 s each (384 s), yielding an economic finality time of 2 × 384 = 768 s = 12.8 min [80]. EBSI, built on Hyperledger Besu with QBFT consensus, uses an 8 s block time and provides deterministic single-block finality [20,81]. Hyperledger Fabric achieves deterministic finality in 1–5 s depending on endorsement policy and block-cutting configuration [55,82]. The accumulator-based revocation overhead of 42.86 ms (presentation) and 31.36 ms (verification) is taken directly from the experimental evaluation of Xu et al. [71].

Several patterns emerge from the comparison. The latency overhead of blockchain write operations ranges from 3.7× (credential revocation with local accumulator checks) to 24,000× (audit trail writes to a public chain versus a local database). This overhead is intrinsic to distributed consensus and will not be eliminated through optimization alone. However, five of the six reviewed proposals adopt a hybrid architecture in which the user-facing operation proceeds at traditional speed and the blockchain anchor is written asynchronously. In the hybrid timestamping architecture of [19], for instance, the RFC 3161 token is returned to the client in 93 ms, identical to a conventional TSA, while the Ethereum or Hyperledger Fabric anchor is committed in the background. This way, the user never waits for blockchain confirmation.

The one domain where blockchain overhead is closest to traditional performance is credential revocation in the EUDI Wallet ecosystem. The accumulator-based approach of [71] adds approximately 74 ms to a verification flow, a 3.7× overhead relative to CDN-served OCSP. This overhead is modest and delivers a property that traditional OCSP fundamentally cannot: unlinkable verification that prevents the issuer from tracking which credentials are being checked and by whom. The cascaded Bloom filter approach of [73] reduces the check latency further to below 10 ms (comparable to CRLite [74]) while randomizing filter indexes daily to prevent cross-session correlation.

9.2. Cost Comparison

Table 6 compares the operational costs of traditional and blockchain-based trust services, normalized to a workload of 10,000 operations per day. Three deployment models are considered: the traditional (non-blockchain) baseline, public blockchain (Ethereum L1), and permissioned blockchain (Hyperledger Fabric or Besu/EBSI).

Table 6. Operational cost comparison for 10,000 daily trust service operations.

TSP Domain	Traditional Cost/Day	Public BC (Ethereum L1) Cost/Day	Permissioned BC Cost/Day
Timestamping	EUR 300–500	$0.01–$0.04	$0
Signing audit trail	Infrastructure only	$130–$430 (individual tx) or $0.01–$0.04 (batched)	$0
Preservation anchoring	$0.02 (S3 storage for 10 K × 10 KB objects)	$0.01–$0.04 (single batched Merkle root)	$0
Credential revocation	Infrastructure only	$0.13–$0.43 per status list update tx	$0

Traditional timestamping costs are based on published price lists from eIDAS-qualified TSPs: Datasure (EUR 0.03–0.04 per token at volumes above 10,000) [83] and Disig (EUR 0.05 per token for 10,000-unit packages) [84]. At lower volumes, per-token prices rise to EUR 0.10–0.15 (Datasure) or EUR 0.117 (Disig for 100-unit packages). DigiStamp, a US-based provider, offers rates from $0.33/token (30 tokens) down to $0.021/token (2.5 million/year) [85].

Ethereum L1 gas costs are computed from protocol-level parameters. Storing a 32-byte hash via the SSTORE operation costs 22,100 gas (post-EIP-2929) [86]. A full smart contract

call that stores a hash and emits an event uses approximately 45,000–65,000 gas. At the gas price observed in March 2026 (0.3 gwei) and an ETH price of approximately $2000, this yields $0.013–$0.043 per transaction [87]. However, gas prices are highly volatile: during congestion events, prices have historically exceeded 100 gwei, which would increase these costs by a factor of 300 or more. For this reason, batching, anchoring a single Merkle root that covers an entire batch of operations, is essential for any production deployment on a public chain. A single batched anchor reduces the per-operation blockchain cost to sub-cent levels regardless of batch size.

On permissioned blockchains (Hyperledger Fabric, Hyperledger Besu/EBSI), there is no per-transaction gas fee. The cost is entirely in infrastructure provisioning: operating validator nodes, maintaining network connectivity, and ensuring availability. For an organization that already participates in a consortium such as EBSI, the marginal cost of additional transactions is effectively zero. Fabric achieves 1000–3500 TPS [82] and EBSI's effective throughput is conservatively estimated at 50 TPS given its pan-European distribution of 40 validator nodes [20,81], both far exceeding typical TSP workloads.

The cost comparison reveals a structural asymmetry. The dominant cost of blockchain-augmented trust services on public chains is the per-transaction gas fee, which is eliminated entirely on permissioned chains. The dominant cost of traditional trust services is either per-operation licensing (timestamping) or fixed infrastructure hosting (databases, OCSP responders). In a hybrid model, where the traditional service handles the user-facing operation and blockchain provides supplementary anchoring, the blockchain cost is additive but marginal: a single daily Merkle root anchor on Ethereum L1 costs $0.01–$0.04, regardless of how many operations it covers.

9.3. Limitations

Several methodological limitations should be acknowledged:

- The performance figures compiled in this section originate from different papers, measured on different hardware, network configurations, and software versions. Direct cross-paper comparisons are indicative rather than definitive. Where possible, we have preferred published benchmarks from the papers reviewed in the current article [13,16,19,48,49,71,73].
- Ethereum gas prices fluctuate by orders of magnitude. The cost figures cited here reflect low-congestion conditions (March 2026, 0.3 gwei). For production eIDAS services, this volatility must be mitigated through batching, Layer 2 usage, or permissioned alternatives. Layer 2 solutions such as Arbitrum, Base, and Optimism offer sub-cent transaction costs ($0.001–$0.01) and sub-second soft confirmation [88], but their reliance on centralized sequencers raises questions about their suitability for regulated trust services that require supervisory accountability.
- Throughput and latency figures for Fabric and Besu assume configurations representative of the cited literature (4–40 validators, Raft/QBFT consensus). Performance varies significantly with network size, endorsement policies, and geographic distribution.
- Fourth, with the exception of the IPFS storage comparison in [49] and the revocation overhead measurements in [71], most reviewed proposals do not include production-scale benchmarks. The comparison in this section combines published micro-benchmarks with platform reference data rather than end-to-end system measurements.

9.4. Discussion

The quantitative evidence compiled in this section supports three conclusions.

First, blockchain introduces measurable latency overhead for write operations, but hybrid architectures hide it from end users. Across all six domains, writing to a blockchain is 3.7× to 24,000× slower than writing to a local database or querying a centralized service. This overhead is intrinsic to distributed consensus and will not be eliminated through optimization alone. However, five of the six reviewed proposals adopt a hybrid architecture in which the user-facing operation proceeds at traditional speed and the blockchain anchor is written asynchronously. In the hybrid timestamping architecture of [19], for instance, the RFC 3161 token is returned to the client in 93 ms, identical to a conventional TSA, while the Ethereum or Hyperledger Fabric anchor is committed in the background. The user never waits for blockchain confirmation. This pattern, present in SABRES [48] for signing audit trails, LTPS [13] and its IPFS extension [49] for preservation, TALChain [16] for validation logging, and the hybrid TSA [19] for timestamping, represents the most practical integration path for blockchain in eIDAS-compliant trust services.

Second, blockchain is cost-competitive when batching is employed or when permissioned platforms are used. On public Ethereum L1, individual transaction costs are acceptable at current gas prices ($0.013–$0.043) but subject to extreme volatility. During congestion events, costs can spike by a factor of 300 or more. Batching, anchoring a single Merkle root per batch, reduces the per-operation blockchain cost to sub-cent levels, making it negligible relative to existing TSP operational costs (EUR 0.03–0.05 per qualified timestamp, for comparison). At a workload of 10,000 operations per day, the total blockchain anchoring cost with batching is $0.01–$0.04 regardless of volume. On permissioned chains (Hyperledger Fabric, Hyperledger Besu/EBSI), per-transaction costs are effectively zero beyond fixed infrastructure provisioning, and both platforms far exceed typical TSP throughput requirements.

Third, the primary quantitative advantage of blockchain is audit efficiency, not operational speed. Traditional provider-controlled logs are faster by orders of magnitude for both writes and reads, but they require trusting the log operator. A Merkle proof against a blockchain block hash provides cryptographic assurance of record integrity in a single verification step, without requiring trust in the record-keeping entity. This property, independent, post-hoc verifiability, is the quantitative basis for the qualitative claims about transparency and auditability made throughout Sections 3–8. It is this audit efficiency, rather than latency or throughput, that constitutes the measurable value proposition of blockchain for eIDAS trust services.

10. Future Directions: Privacy, Interoperability, and Cryptographic Resilience

The integration of blockchain into trust services is still evolving, and several emerging technologies are likely to influence its future development. Among the most relevant directions are privacy-preserving verification through zero-knowledge proofs (ZKPs), interoperability across heterogeneous blockchain environments, and the transition toward post-quantum cryptography. These directions are particularly important in eIDAS-oriented and standards-based settings, where trust services must combine legal reliability, technical interoperability, long-term security, and strong protection of user data.

10.1. Zero-Knowledge Proofs and Privacy-Preserving Verification

A major future direction for blockchain-enabled trust services is the integration of ZKPs into validation, revocation, and attribute verification processes. In trust service and digital identity environments, verifiers often need confirmation that a certain condition is achieved without learning the full underlying data. This is especially relevant in selective disclosure scenarios, where a user may need to prove possession of a valid attribute,

credential, or status without exposing the entire credential set or creating a presentation pattern that can be linked across multiple interactions.

This direction is particularly important in the context of VCs, qualified electronic attestations, and privacy-oriented wallet infrastructures. Current developments in European digital identity architectures, including the EUDI Wallet ecosystem and related privacy-preserving revocation models, already point toward selective disclosure, unlinkability, and privacy-aware validation as core design objectives [43,73,75]. In this setting, ZKPs can support a more advanced form of verification, where a relying party learns only the result of the required check while the holder preserves control over the disclosed information. This can reduce unnecessary data exposure and strengthen compliance with privacy-by-design principles.

The relevance of this direction also extends to revocation and status checking. Traditional revocation methods may reveal more metadata than necessary or make repeated checks linkable over time. Privacy-oriented designs suggest that future trust service infrastructures may benefit from combining blockchain anchoring with cryptographic proofs that demonstrate non-revocation or validity without disclosing stable identifiers [71,73]. In this sense, ZKPs represent not only a privacy enhancement but also a way to make blockchain-based trust services more suitable for regulated environments where both verifiability and confidentiality matter.

10.2. Cross-Chain Interoperability

A second important direction concerns interoperability across multiple blockchain environments. Many current solutions assume that trust service evidence, revocation information, or credential-related records remain limited to a single ledger. However, future deployments are likely to span public networks, permissioned ledgers, and sector-specific ecosystems. This creates the need for mechanisms that allow evidence or status information to be verified across ledger boundaries without duplicating infrastructure or weakening trust assumptions.

This issue is particularly relevant in decentralized identity and credential ecosystems, where different organizations may adopt distinct blockchain technologies while still needing to validate the same category of trust evidence. In such settings, cross-chain approaches may support the transfer, synchronization, or verification of revocation data, credential status, or audit records across otherwise isolated networks. This could improve portability and reduce fragmentation, especially in scenarios where several institutions must cooperate while preserving independent infrastructures.

Recent bridge-based architectures suggest that such interoperability is technically achievable, at least in controlled environments where governance and verification logic are clearly defined. In this perspective, cross-chain mechanisms may become relevant not only for VCs, but also for broader trust service functions such as status checking, preservation evidence, or timestamp-related anchoring [89]. At the same time, interoperability layers, relays, or bridges may themselves become critical trust dependencies, which means that security and governance requirements remain essential.

10.3. Post-Quantum Cryptography and Long-Term Security

A third major direction is the transition toward post-quantum cryptography. This is especially relevant for trust services because many associated security guarantees must remain valid over long periods of time. Signature evidence, preservation records, timestamp tokens, and blockchain anchors may all need to remain verifiable beyond the lifetime of the cryptographic algorithms currently used to create them. For this reason, long-term cryptographic resilience is becoming an increasingly important design objective.

In blockchain-enabled trust services, this issue appears at two levels. The first concerns trust service artifacts themselves, such as certificates, signatures, timestamps, and preservation evidence. The second concerns the blockchain layer used to anchor or support those components. In both cases, future migration must be possible without invalidating previously created evidence. This makes cryptographic agility essential. Architectures that rely on re-anchoring, re-timestamping, stronger hash functions, or algorithm migration paths are therefore likely to become increasingly important, especially in timestamping and long-term preservation scenarios [19].

From a broader perspective, post-quantum cryptography is highly relevant for systems that rely on long-term trust. In future blockchain-assisted trust services, post-quantum mechanisms may be needed not only for classical PKI components, but also for blockchain-linked evidence models intended to remain valid for many years. This shows that blockchain does not remove the need for cryptographic evolution, but rather increases the importance of long-term migration planning.

Overall, the combination of zero-knowledge proofs, cross-chain interoperability, and post-quantum cryptography suggests that future blockchain-enabled trust services will need to evolve along three complementary directions: stronger privacy, broader interoperability, and long-term cryptographic resilience. Their integration into eIDAS-aligned and standards-based trust service models remains an open but highly relevant direction for future research and implementation.

11. Technical State-of-the-Art Summary and Discussion

This section summarizes the main technological domains discussed throughout the paper and highlights the current state of the art regarding the integration of blockchain technologies with PKI and electronic signature ecosystems.

PKI-based Trust Service Infrastructures—The current trust service ecosystem relies primarily on PKI-based infrastructures operated by Trust Service Providers (TSPs) under regulatory frameworks such as eIDAS. These infrastructures implement standardized mechanisms for certificate issuance, validation, and revocation using well-established protocols and data formats such as X.509 certificates and ETSI electronic signature formats. Remote electronic signature architectures have evolved significantly in recent years, enabling cloud-based signature creation and scalable trust service deployments. Several implementations and architectural proposals have been documented in the literature, including SaaS-based signature services, cloud cryptographic infrastructures, and modular remote signature components [29–33]. These approaches demonstrate the maturity and interoperability of PKI-based systems, while also highlighting the centralized trust model inherent to certificate authorities and trust service operators.

Blockchain for Trust Service Transparency and Certificate Lifecycle Management—Blockchain technologies have been widely explored as mechanisms for improving transparency and auditability in trust service infrastructures. In many proposals, distributed ledgers are used to record certificate lifecycle events [9], such as issuance, revocation, or status updates, in order to provide tamper-evident audit trails. Such approaches aim to complement traditional PKI systems by adding a transparency layer that allows independent verification of certificate-related operations [11]. Several studies have analyzed the potential of blockchain infrastructures to improve traceability and accountability in digital trust ecosystems while maintaining compatibility with existing PKI frameworks [31]. However, most proposals emphasize hybrid architectures where blockchain does not replace PKI, but rather augments its transparency and verification capabilities [48].

Signature Validation and Long-Term Integrity Mechanisms—Signature validation and long-term preservation remain critical aspects of electronic signature infrastructures.

Existing solutions rely on standardized formats such as PAdES, XAdES, and CAdES, combined with trusted timestamping and certificate validation services. While blockchain has been proposed as an additional mechanism for storing integrity proofs or anchoring cryptographic hashes, its use in signature validation workflows remains limited in practice. Current research primarily explores blockchain as a complementary integrity verification layer rather than as a replacement for traditional validation infrastructures [19,48].

EUDI Wallet and VC Ecosystems—The emergence of the EUDI Wallet introduces new architectural paradigms for digital identity and trust services. In this ecosystem, VCs are used to represent digitally signed claims that can be presented and verified across different services. Infrastructure initiatives such as the European Blockchain Services Infrastructure (EBSI) explore the use of blockchain technologies to support credential registries, trust anchors, and revocation mechanisms for VC [73,89]. These architectures aim to enable decentralized identity models while maintaining compliance with European regulatory frameworks. Across the reviewed domains, a recurring pattern is that blockchain-based architectures do not simply add decentralization, but also reveal a set of broader cross-domain challenges and trade-offs. Greater transparency and independent verifiability can improve auditability, but these gains must be balanced against privacy constraints, interoperability requirements, governance design, scalability limits, and the legal role of supervised Trust Service Providers. This is particularly visible in the EUDI Wallet ecosystem, although the same pattern also appears in certificate management, remote signing, preservation, validation, and timestamping. For this reason, the most viable approaches identified in this review are generally hybrid rather than purely decentralized, combining blockchain-based anchoring or auditability with standardized procedures, qualified services, and regulatory supervision.

Content Authenticity and Emerging Provenance Frameworks. Beyond traditional trust services, recent initiatives have explored mechanisms for ensuring the authenticity and provenance of digital content. Frameworks such as the C2PA specification define mechanisms for attaching cryptographically verifiable provenance information to digital assets. Although these frameworks are not directly part of traditional PKI-based trust service infrastructures, they rely on similar cryptographic primitives and trust anchors. As a result, future research may explore how blockchain-based transparency mechanisms and electronic signature infrastructures can support content authenticity ecosystems and digital provenance verification [60].

As presented in Table 7, the current state of the art indicates that blockchain technologies are primarily positioned as complementary mechanisms that enhance transparency, traceability, and distributed trust in PKI-based ecosystems. Rather than replacing existing infrastructures, most proposed architectures rely on hybrid models in which blockchain provides a verifiable evidentiary and audit layer, while traditional trust services continue to deliver identity binding, cryptographic assurance, and regulatory compliance. Accordingly, blockchain is most appropriately understood as a supplementary anchoring layer for existing trust service evidence, while the qualified services and standardized mechanisms required for legal recognition under eIDAS remain central.

To complement the conceptual comparison, Table 8 classifies the technology maturity of blockchain integration for each application domain reviewed in this paper. The maturity levels follow a simplified scale:

- Conceptual (theoretical proposals without implementation);
- Prototype (proof-of-concept implementations evaluated in controlled settings);
- Pilot (deployments tested in realistic or multi-stakeholder environments);
- Production (operational services available to end users).

Table 7. Conceptual comparison between PKI-based and blockchain-based trust mechanisms.

Concept	PKI	Blockchain	Key References
Identity binding	CA-issued X.509 certificate	Verifiable credential/DID	[1,3,20,89]
Governance	Trusted authorities (national supervisory bodies)	Consensus network (validator governance)	[1,2,20]
Transparency	Conformity assessment audits (Art. 20)	Public/consortium ledger	[2,5,16]
Integrity	Digital signatures (RSA, ECDSA)	Cryptographic hashing + Merkle proofs	[4,9,11]
Certificate/credential status	CRL/OCSP	Ledger-based status registry; accumulators; Bloom filters	[9,11,71,73]
Signature validation	Standardized (ETSI EN 319 102-1)	Standard validation + supplementary ledger log checks	[4,16,48]
Audit evidence	Provider-internal logs (mutable)	Immutable on-chain records (Merkle-provable)	[16,19,48]
Timestamping	Qualified TSA (RFC 3161, ETSI EN 319 421)	Blockchain anchoring (proof-of-existence); hybrid TSA + BC	[17–19]
Trust model	Centralized trust (EUTL/national Trusted Lists)	Distributed trust (consortium consensus or public PoS/PoW)	[1,5,20]
Architecture	Hierarchical PKI (root CA → sub-CA → EE)	Decentralized registry (smart contracts, DID registries)	[7,9,20,89]
Preservation	ETSI-compliant augmentation chain	Blockchain integrity anchoring of preservation evidence	[13,14,49]
Content provenance	PKI-signed C2PA Manifests	Blockchain-anchored attestation + perceptual hashing	[59–61]
Legal standing	Qualified services carry legal presumption (Art. 25, 35, 41)	Admissible as supplementary evidence (Art. 46); eIDAS 2.0 electronic ledger (Art. 45h)	[1,21]

Table 8. Technology maturity of blockchain integration by TSP application domain.

Application Domain	Maturity Level	Justification
Certificate management	Conceptual/Prototype	Smart contract frameworks proposed and evaluated on testnets; no integration with production CAs reported.
Signature services	Prototype	Proof-of-concept with reverse-proxy architecture tested against a CSC-compliant signing service; not deployed in a production TSP environment.
Signature preservation	Prototype/Pilot	LTPS implements end-to-end preservation with Ethereum anchoring; IPFS extension includes empirical storage benchmarks; no QTSP deployment.
Signature validation	Prototype	Permissioned PoA ledger with explicit validation workflow instrumentation; evaluated architecturally but not deployed across multiple TSPs.
Timestamping	Pilot/Production	OriginStamp operates as a public service on Bitcoin; the hybrid TSA supports both Ethereum and Hyperledger Fabric with standard-compliant token issuance.
Content provenance	Conceptual/Prototype	Blockchain-based C2PA extensions proposed; proof-of-concept attestation services demonstrated; no integration with production C2PA workflows.
EUDI Wallet/EBSI	Pilot	EBSI operational with 40 nodes across EU Member States; credential issuance and verification piloted in education and identity domains; pre-production for eIDAS 2.0 wallet ecosystem.

12. Conclusions

This paper presented a comprehensive review of blockchain integration across seven eIDAS trust service domains: certificate management, signature services, signature preservation, signature validation, timestamping, content provenance, and the EUDI Wallet ecosystem. The review yields several key findings. First, across all domains, blockchain is consistently positioned as a complementary layer that strengthens auditability and distributed trust without replacing the qualified services that carry legal presumption under eIDAS (Articles 25, 35, 41). The hybrid architecture pattern, in which the user-facing qualified service operates at traditional speed while blockchain anchoring occurs asynchronously, emerges as the dominant design approach, present in SABRES [48] for signing audit trails, LTPS [13] and its IPFS extension [49] for preservation, TALChain [16] for validation logging, and the hybrid TSA [19] for timestamping. The quantitative analysis in Section 9 confirms that this pattern is practical: blockchain write latency (1–12 s depending on platform) is 16× to 24,000× slower than traditional alternatives, but asynchronous anchoring hides this overhead from end users entirely. Cost analysis shows that batching a single Merkle root per day reduces public chain anchoring costs to $0.01–$0.04 regardless of volume, while permissioned chains (Hyperledger Fabric, EBSI) incur no per-transaction fee. The technology maturity classification (Table 8) reveals that timestamping has reached Pilot/Production

status, and the EUDI Wallet/EBSI ecosystem is in active Pilot across EU Member States, while certificate management and content provenance remain at the Conceptual/Prototype stage. Blockchain evidence does not carry the legal presumption reserved for qualified trust services, but is admissible as supplementary evidence under Article 46, and the eIDAS 2.0 framework (Regulation 2024/1183) [21] introduces electronic ledgers as a recognized trust service category (Article 45h), providing a future regulatory pathway.

Several limitations of this review should guide future research. First, the quantitative comparison in Section 9 aggregates data from heterogeneous sources, different papers measured on different hardware and software configurations, and most reviewed proposals lack production-scale benchmarks. Future work should prioritize controlled end-to-end evaluations under realistic eIDAS workloads. Second, the legal and certification pathway for blockchain components within qualified trust services remains undeveloped: no blockchain-native service has achieved qualified status, and the implementing acts for eIDAS 2.0 electronic ledgers are still under development. Third, the tension between blockchain immutability and GDPR erasure rights (Article 17) requires further investigation, particularly for credential revocation mechanisms in the EUDI Wallet ecosystem where privacy-preserving approaches such as zero-knowledge proofs and cascaded Bloom filters [73] are promising but not yet standardized. Fourth, cross-chain interoperability, post-quantum cryptographic resilience, and the scalability of EBSI's single-chain Hyperledger Besu architecture for hundreds of millions of EU citizens remain open challenges. Future research should focus on real-world pilot deployments that combine blockchain with established eIDAS trust services, the development of ETSI conformity assessment criteria for blockchain-augmented QTSPs, and the standardization of privacy-preserving verification mechanisms for the European digital identity infrastructure.

Author Contributions: Conceptualization, E.B. and F.P.; methodology, A.B.; validation, Ș.A., F.P.; formal analysis, A.B.; investigation, R.-A.L., E.B. and A.B.; writing—original draft preparation, R.-A.L., A.B., E.B. and F.P.; writing—review and editing, E.B., Ș.A. and A.B.; visualization, A.B.; supervision, F.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Executive Unit for Financing Higher Education, Research, Development and Innovation (UEFISCDI), Romania, grant number 51PTE/2025.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: Author Ștefan Arseni was employed by the company CertSIGN S.A. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PKI	Public Key Infrastructure
ES	Electronic Signature
TSP	Trust Service Provider
CA	Certificate Authority
EUDI	European Digital Identity
EU	European Union
QTSP	Qualified Trust Service Provider

QES	Qualified Electronic Signature
QSCD	Qualified Signature Creation Device
ETSI	European Telecommunications Standards Institute
AdES	Advanced Electronic Signature
SCDev	Signature Creation Device
TW4S	Trustworthy Systems Supporting Server Signing
SSA	Server Signing Application
SAM	Signature Activation Module
SAD	Signature Activation Data
CAdES	CMS Advanced Electronic Signature
PAdES	PDF Advanced Electronic Signature
ASiC	Associated Signature Container
CSC	Cloud Signature Consortium
API	Application Programming Interface
SaaS	Software as a Service
CRL	Certificate Revocation List
OCSP	Online Certificate Status Protocol
REST	Representational State Transfer
IPFS	InterPlanetary File System
TSA	Timestamping Authority
ARF	Architecture and Reference Framework
SSI	Self-Sovereign Identity
VC	Verifiable Credential
QBFT	Quorum Byzantine Fault Tolerant
DIDs	Decentralized Identifiers
TIR	Trusted Issuers Registry
EBSI	European Blockchain Services Infrastructure
TAO	Trusted Accreditation Organisation
TI	Trusted Issuer
GDPR	General Data Protection Regulation
W3C	World Wide Web Consortium
ZKP	Zero-Knowledge Proof

References

1. European Parliament and the Council of the European Union. Regulation (EU) No 910/2014 of 23 July 2014 on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market and Repealing Directive 1999/93/EC (eIDAS). EUR-Lex. Available online: https://eur-lex.europa.eu/eli/reg/2014/910/oj/eng (accessed on 10 February 2026).
2. *ETSI EN 319 401 V3.2.1 (2026-01)*; Electronic Signatures and Trust Infrastructures (ESI); General Policy Requirements for Trust Service Providers. ETSI: Sophia Antipolis, France, 2026. Available online: https://www.etsi.org/deliver/etsi_en/319400_319499/319401/03.02.01_60/en_319401v030201p.pdf (accessed on 10 February 2026).
3. *ETSI EN 319 411-2 V2.6.0 (2024-12)*; Electronic Signatures and Trust Infrastructures (ESI); Policy and Security Requirements for Trust Service Providers Issuing Certificates; Part 2: Requirements for Trust Service Providers Issuing EU Qualified Certificates. ETSI: Sophia Antipolis, France, 2024. Available online: https://www.etsi.org/deliver/etsi_en/319400_319499/31941102/02.06.00_20/en_31941102v020600a.pdf (accessed on 10 February 2026).
4. *ETSI EN 319 102-1 V1.4.1 (2024-06)*; Electronic Signatures and Trust Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 1: Creation and Validation. ETSI: Sophia Antipolis, France, 2024. Available online: https://www.etsi.org/deliver/etsi_en/319100_319199/31910201/01.04.01_60/en_31910201v010401p.pdf (accessed on 10 February 2026).
5. Yaga, D.; Mell, P.; Roby, N.; Scarfone, K. Blockchain Technology Overview. NIST Internal Report 8202. 2018. Available online: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf (accessed on 17 February 2026).
6. Zheng, Z.; Xie, S.; Dai, H.-N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
7. Laurie, B.; Langley, A.; Kasper, E. RFC 6962: Certificate Transparency. RFC Editor. 2013. Available online: https://www.rfc-editor.org/rfc/rfc6962.html (accessed on 17 February 2026).

8. Lin, I.-C.; Liao, T.-C. A survey of blockchain security issues and challenges. *Int. J. Netw. Secur.* **2017**, *19*, 653–659.
9. Yakubov, A.; Shbair, W.M.; Wallbom, A.; Sanda, D.; State, R. A blockchain-based PKI management framework. In Proceedings of the First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block), Colocated with IEEE/IFIP NOMS 2018, Taipei, Taiwan, 23–27 April 2018; pp. 1–6. [CrossRef]
10. Kubilay, M.Y.; Kiraz, M.S.; Mantar, H.A. Certledger: A new PKI model with certificate transparency based on blockchain. *Comput. Secur.* **2019**, *85*, 333–352. [CrossRef]
11. Wang, Z.; Lin, J.; Cai, Q.; Wang, Q.; Jing, J. Blockchain-based certificate transparency and revocation transparency. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 163–177.
12. Yang, K.; Sunny, J.J.; Wang, L. Blockchain-based decentralized public key management for named data networking. In Proceedings of the International Conference on Computer Communications and Networks (ICCCN 2018), Hangzhou, China, 30 July–2 August 2018; pp. 1–9.
13. Arseni, Ş.-C.; Togan, M.; Aciobăniţei, I.; Bureacă, E.; Coca, M. LTPS—Service for long-term preservation of digital signatures. In Proceedings of the 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Ploiesti, Romania, 30 June–1 July 2022; pp. 1–6. [CrossRef]
14. Bralić, V.; Kuleš, M.; Stančić, H. A Model for Long-term Preservation of Digital Signature Validity: TrustChain. In Proceedings of the INFuture2017: Integrating ICT in Society, Zagreb, Croatia, 8–10 November 2017. [CrossRef]
15. Arseni, Ș.-C.; Bureacă, E.; Togan, M. Long-term preservation of digital signatures: A need-to-have or a nice-to-have? *J. Mil. Technol.* **2022**, *5*, 41–48. [CrossRef]
16. Bureacă, E.; Patriciu, V.-V. TALChain—A permissioned-ledger architecture to improve security, auditability, and reputation of trust service providers. *J. Mil. Technol.* **2025**, *8*, 67–72. [CrossRef]
17. Gipp, B.; Meuschke, N.; Gernandt, A. Decentralized Trusted Timestamping using the Crypto Currency Bitcoin. *arXiv* **2015**, arXiv:1502.04015. [CrossRef]
18. Zhang, Y.; Xu, C.; Cheng, N.; Li, H.; Yang, H.; Shen, X. Chronos+: An Accurate Blockchain-Based Time-Stamping Scheme for Cloud Storage. *IEEE Trans. Serv. Comput.* **2020**, *13*, 216–229. [CrossRef]
19. Brînzea, A.; Leancă, R.-A.; Aciobăniței, I.; Pop, F. Standard-compliant blockchain anchoring for timestamp tokens. *Appl. Sci.* **2025**, *15*, 12722. [CrossRef]
20. European Commission; European Blockchain Services Infrastructure (EBSI). Digital Strategy. Available online: https://digital-strategy.ec.europa.eu/en/policies/european-blockchain-services-infrastructure (accessed on 17 February 2026).
21. European Parliament and the Council of the European Union. Regulation (EU) 2024/1183 of 11 April 2024 Amending Regulation (EU) No 910/2014 as Regards Establishing the European Digital Identity Framework (eIDAS 2.0). EUR-Lex. Available online: https://eur-lex.europa.eu/eli/reg/2024/1183/oj/eng (accessed on 18 February 2026).
22. *ETSI TS 119 431-1 V1.3.1 (2024-12)*; Electronic Signatures and Trust Infrastructures (ESI); Policy and Security Requirements for Trust Service Providers Operating a Remote Signature Creation Device; Part 1: General Requirements. ETSI: Sophia Antipolis, France, 2014. Available online: https://www.etsi.org/deliver/etsi_ts/119400_119499/11943101/01.03.01_60/ts_11943101v010301p.pdf (accessed on 18 February 2026).
23. *EN 419 241-1:2018*; Trustworthy Systems Supporting Server Signing—Part 1: General System Security Requirements. European Committee for Standardization: Brussels, Belgium, 2018.
24. Leitold, H.; Konrad, R.; Posch, R. Qualified Remote Signatures—Solutions, its Certification, and Use. 2019. Available online: https://tugraz.elsevierpure.com/ws/portalfiles/portal/22262371/WS2019_RemoteSignatures_Leitold.pdf (accessed on 18 February 2026).
25. Scurtu, P.; Patriciu, V.-V. Device Centric Cloud Signature Solution under eIDAS Regulation. *J. Mil. Technol.* **2020**, *3*, 49–54. [CrossRef]
26. *ETSI TS 119 432 V1.1.1 (2019-03)*; Electronic Signatures and Infrastructures (ESI). Protocols for Remote Digital Signature Creation. ETSI: Sophia Antipolis, France, 2019. Available online: https://www.etsi.org/deliver/etsi_ts/119400_119499/119432/01.01.01_60/ts_119432v010101p.pdf (accessed on 18 February 2026).
27. *CSC API V1.0.4.0*; Standard—Remote Signature Protocols and API (2019-06). CSC. Available online: https://cloudsignatureconsortium.org/wp-content/uploads/2020/01/CSC_API_V1_1.0.4.0.pdf (accessed on 18 February 2026).
28. Aciobanitei, I.; Urian, P.D.; Pura, M.L. Learned Lessons from Implementing an Android Client for the Cloud Signature Consortium API. In *Innovative Security Solutions for Information Technology and Communications. SecITC 2019*; Simion, E., Géraud-Stewart, R., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12001. [CrossRef]
29. Aciobanitei, I.; Luculescu, I.; Pura, M.-L. Qualified Electronic Signature SaaS Solution for Google Docs & Google Sheets Documents. In Proceedings of the 2020 13th International Conference on Communications (COMM), Bucharest, Romania, 18–20 June 2020.
30. Ruica, E.-C.; Pura, M.-L.; Aciobanitei, I. Implementing Cloud Qualified Electronic Signatures for Documents using Available Cryptographic Libraries: A Survey. In Proceedings of the 2020 13th International Conference on Communications (COMM), Bucharest, Romania, 18–20 June 2020; pp. 113–118. [CrossRef]

31. Aciobăniței, I.; Arseni, Ș.-C.; Bureacă, E.; Togan, M. A Comprehensive and Privacy-Aware Approach for Remote Qualified Electronic Signatures. *Electronics* **2024**, *13*, 757. [CrossRef]
32. Brînzea, A.; Aciobănitei, I. A Versatile Helper Module for Remote Qualified Electronic Signatures. In Proceedings of the 2024 Advanced Topics on Measurement and Simulation (ATOMS), Constanta, Romania, 28–30 August 2024; pp. 244–247. [CrossRef]
33. Aciobanitei, I.; Urian, P.D.; Pura, M. A Cryptography API: Next Generation Key Storage Provider for Cryptography in the Cloud. In Proceedings of the 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Iasi, Romania, 28–30 June 2018; pp. 1–4. [CrossRef]
34. Aciobanitei, I.; Buhus, I.C.; Pura, M.-L. Using Cryptography in the Cloud for Lightweight Authentication Protocols Based on QR Codes. In Proceedings of the2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 17–19 May 2018; pp. 000539–000542. [CrossRef]
35. *ETSI EN 319 421 V1.2.1*; Electronic Signatures and Infrastructures (ESI); Policy and Security Requirements for Trust Service Providers Issuing Time-Stamps. ETSI: Sophia Antipolis, France, 2023. Available online: https://www.etsi.org/deliver/etsi_en/319400_319499/319421/01.02.01_60/en_319421v010201p.pdf (accessed on 27 February 2026).
36. Guggenmos, F.; Rieger, A.; Wenninger, A.; Fridgen, G.; Lockl, J. How to Develop a GDPR-Compliant Blockchain Solution for Cross-Organizational Workflow Management: Evidence from the German Asylum Procedure. In Proceedings of the Proceedings of the 53rd Hawaii International Conference on System Sciences, Maui, HI, USA, 7–10 January 2020. [CrossRef]
37. *ETSI TS 119 512 V1.2.1 (2023-05)*; Electronic Signatures and Infrastructures (ESI); Protocols for Trust Service Providers Providing Long-Term Data Preservation Services. ETSI: Sophia Antipolis, France, 2023. Available online: https://www.etsi.org/deliver/etsi_ts/119500_119599/119512/01.02.01_60/ts_119512v010201p.pdf (accessed on 3 March 2026).
38. *ETSI TS 119 511 V1.2.1 (2025-10)*; Electronic Signatures and Trust Infrastructures (ESI); Policy and Security Requirements for Trust Service Providers Providing Long-Term Preservation of Digital Signatures or General Data Using ETSI: Sophia Antipolis, France, 2025. Available online: https://www.etsi.org/deliver/etsi_ts/119500_119599/119511/01.02.01_60/ts_119511v010201p.pdf (accessed on 27 February 2026).
39. *ETSI 119 172-4 V1.1.1 (2021-05)*; Electronic Signatures and Infrastructures (ESI); Signature Policies; Part 4: Signature Applicability Rules (Validation Policy) for European Qualified Electronic Signatures/Seals Using Trusted Lists. ETSI: Sophia Antipolis, France, 2021. Available online: https://www.etsi.org/deliver/etsi_ts/119100_119199/11917204/01.01.01_60/ts_11917204v010101p.pdf (accessed on 27 February 2026).
40. Adams, C.; Cain, P.; Pinkas, D.; Zuccherato, R. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). In *RFC 3161*; ACM: Nuew York, NY, USA, 2001. [CrossRef]
41. *ETSI EN 319 422 V1.1.1*; Electronic Signatures and Infrastructures (ESI); Time-Stamping Protocol and Time-Stamp Token Profiles. ETSI: Sophia Antipolis, France, 2016. Available online: https://www.etsi.org/deliver/etsi_en/319400_319499/319422/01.01.01_60/en_319422v010101p.pdf (accessed on 27 February 2026).
42. Content Credentials: C2PA Technical Specification. C2pa.org. Available online: https://spec.c2pa.org/specifications/specifications/2.3/specs/C2PA_Specification.html (accessed on 3 March 2026).
43. Architecture and Reference Framework v. 2.8.0 (ARF). Available online: https://eudi.dev/2.8.0/architecture-and-reference-framework-main/ (accessed on 27 February 2026).
44. *ETSI TS 119 612 V2.4.1 (2025-08)*; Electronic Signatures and Trust Infrastructures (ESI); Trusted Lists. ETSI: Sophia Antipolis, France, 2025. Available online: https://www.etsi.org/deliver/etsi_ts/119600_119699/119612/02.04.01_60/ts_119612v020401p.pdf (accessed on 3 March 2026).
45. OpenID for Verifiable Credential Issuance 1.0 (OpenID4VCI). Available online: https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html (accessed on 3 March 2026).
46. OpenID for Verifiable Presentations 1.0 (OpenID4VP). Available online: https://openid.net/specs/openid-4-verifiable-presentations-1_0.html (accessed on 4 March 2026).
47. Patsonakis, C.; Samari, K.; Kiayias, A.; Roussopoulos, M. On the Practicality of Smart Contract PKI. *arXiv* **2019**. [CrossRef]
48. Aciobăniței, I.; Dediță, V.; Pura, M.-L.; Patriciu, V.-V. SABRES—A proof of concept for enhanced cloud qualified electronic signatures. In 2020 13th International Conference on Communications (COMM), Bucharest, Romania, 18–20 June 2020; pp. 103–108. [CrossRef]
49. Arseni, Ş.-C.; Bureacă, E.; Aciobăniţei, I. Digital Signatures Long-Term Preservation Services, IPFS and Blockchain are a Good Match. In Proceedings of the 2024 15th International Conference on Communications (COMM), Bucharest, Romania, 3–4 October 2024; pp. 1–6. [CrossRef]
50. Sorge, C.; Leicht, M. Blockchain-based electronic time stamps and the eIDAS regulation: The best of both worlds. *Script-Ed* **2022**, *19*, 61–87. [CrossRef]
51. Zhang, Y.; Xu, C.; Li, H.; Yang, H.; Shen, X. Chronos: Secure and Accurate Time-Stamping Scheme for Digital Files via Blockchain. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6. [CrossRef]

52. Estevam, G.; Palma, L.M.; Silva, L.R.; Martina, J.E.; Vigil, M. Accurate and decentralized timestamping using smart contracts on the Ethereum blockchain. *Inf. Process. Manag.* **2021**, *58*, 102471. [CrossRef]
53. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. White Paper. 2008. Available online: https://bitcoin.org/bitcoin (accessed on 4 March 2026).
54. Buterin, V. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. Available online: https://ethereum.org/en/whitepaper/ (accessed on 4 March 2026).
55. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger Fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1–15. [CrossRef]
56. Santesson, S.; Pope, N. ESSCertIDv2 Update for RFC 3161. In *RFC 5816*; ACM: New York, NY, USA, 2010. [CrossRef]
57. Olan, F.; Jayawickrama, U.; Arakpogun, E.O.; Suklan, J.; Liu, S. Fake news on Social Media: The Impact on Society. *Inf. Syst. Front. J. Res. Innov.* **2022**, *26*, 443–458. [CrossRef] [PubMed]
58. Petratos, P.N.; Faccia, A. Fake news, misinformation, disinformation and supply chain risks and disruptions: Risk management and resilience using blockchain. *Ann. Oper. Res.* **2023**, *327*, 735–762. [CrossRef] [PubMed]
59. Vaziry, A.; Wronka, C.; Garzon, S.R.; Küpper, A. Know Your Contract: Extending eIDAS trust into public blockchains. *arXiv* **2026**. [CrossRef]
60. Bureacă, E.; Aciobăniței, I. A blockchain-based framework for content provenance and authenticity. 2024 16th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Iasi, Romania, 27–28 June 2024; pp. 1–5. [CrossRef]
61. Coppolino, L.; Cristiano, G.M.; D'Antonio, S.; Giglio, J.; Mazzeo, G.; Romano, L. A blockchain solution for decentralized content verification and its application to deepfake detection and fintech credit scoring. *Blockchain Res. Appl.* **2025**, *in press*. [CrossRef]
62. Buțincu, C.N.; Alexandrescu, A. Blockchain-based platform to fight disinformation using crowd wisdom and artificial intelligence. *Appl. Sci.* **2023**, *13*, 6088. [CrossRef]
63. Prasad, P.M. Blockchain + AI in Combating Deepfake Content Circulation. *Sci. J. Artif. Intell. Blockchain Technol.* **2024**, *1*, 10–17. [CrossRef]
64. Coalition for Content Provenance and Authenticity (C2PA). Overview. Available online: https://c2pa.org/ (accessed on 5 March 2026).
65. Preukschat, A.; Reed, D. *Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials*; Manning Publications: Shelter Island, NY, USA, 2021.
66. Sedlmeir, J.; Smethurst, R.; Rieger, A.; Fridgen, G. Digital identities and verifiable credentials. *Bus Inf. Syst. Eng.* **2021**, *63*, 603–613. [CrossRef]
67. Ou, H.-H.; Chen, G.-Y.; Lin, I.-C. A Self-Sovereign Identity Blockchain Framework for Access Control and Transparency in Financial Institutions. *Cryptography* **2025**, *9*, 9. [CrossRef]
68. Wood, G. Polkadot: Vision for a heterogeneous multi-chain framework. *White Pap.* **2016**, *21*, 4662. Available online: https://polkadot.com/papers/Polkadot-whitepaper.pdf (accessed on 5 March 2026).
69. Huynh, P.; Pham, K.; Tan-Vo, K.; Nguyen, T.; Nguyen-Hoang, T.A.; Nguyen, T.; Dinh, N.T. Beyond Immutable: The Landscape of Blockchain Credential Revocation Solutions. In *International Conference on Intelligent Systems Design and Applications*; Springer Nature: Cham, Switzerland, 2023; pp. 321–330. [CrossRef]
70. Adja, Y.C.E.; Hammi, B.; Serhrouchni, A.; Zeadally, S. A blockchain-based certificate revocation management and status verification system. *Comput. Secur.* **2021**, *104*, 102209. [CrossRef]
71. Xu, L.; Li, T.; Erkin, Z. Verifiable Credentials with Privacy-Preserving Tamper-Evident Revocation Mechanism. In Proceedings of the 2023 Fifth International Conference on Blockchain Computing and Applications (BCCA), Kuwait, Kuwait, 24–26 October 2023. [CrossRef]
72. Fang, J.; Feng, T.; Guo, X.; Wang, X. Privacy-enhanced distributed revocable identity management scheme based self-sovereign identity. *J. Cloud. Comp.* **2024**, *13*, 154. [CrossRef]
73. Bureacă, E.; Leancă, R.-A.; Ciobanu, I.; Brînzea, A.; Aciobăniței, I. Unlinkable Revocation Lists for Qualified Electronic Attestations: A Blockchain-Based Framework. *Electronics* **2025**, *14*, 2795. [CrossRef]
74. Larisch, J.; Choffnes, D.; Levin, D.; Maggs, B.M.; Mislove, A.; Wilson, C. CRLite: A Scalable System for Pushing All TLS Revocations to All Browsers. In Proceedings of the 2017 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 22–24 May 2017. Available online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7958597 (accessed on 5 March 2026).
75. *ETSI TR 119 476 V1.2.1 (2024-07)*; Electronic Signatures and Trust Infrastructures (ESI); Analysis of Selective Disclosure and Zero-Knowledge Proofs Applied to Electronic Attestation of Attributes ETSI: Sophia Antipolis, France, 2024. Available online: https://www.etsi.org/deliver/etsi_tr/119400_119499/119476/01.02.01_60/tr_119476v010201p.pdf (accessed on 5 March 2026).
76. Zhu, L.; Amann, J.; Heidemann, J. Measuring the latency and pervasiveness of TLS certificate revocation. In *Passive and Active Measurement*; Springer International Publishing: Cham, Switzerland, 2016; pp. 16–29. . [CrossRef]

77. OCSP Responder Performance Needs Improvement. Unmitigatedrisk.com. Available online: https://unmitigatedrisk.com/?p=33 (accessed on 6 March 2026).
78. Sharma, S. Fermi Estimates on Postgres Performance. Citus Data. Available online: https://www.citusdata.com/blog/2017/09/29/what-performance-can-you-expect-from-postgres/ (accessed on 6 March 2026).
79. Amazon Web Services. Amazon S3 Performance Optimization. Available online: https://docs.aws.amazon.com/AmazonS3/latest/userguide/optimizing-performance.html (accessed on 6 March 2026).
80. Ethereum Foundation. Proof-of-Stake: Gasper. Available online: https://ethereum.org/developers/docs/consensus-mechanisms/pos/gasper/ (accessed on 6 March 2026).
81. European Blockchain Services Infrastructure (EBSI). Blockchain Technical Documentation. Available online: https://hub.ebsi.eu/blockchain (accessed on 6 March 2026).
82. Thakkar, P.; Nathan, S.; Viswanathan, B. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform. In Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Milwaukee, WI, USA, 25–28 September 2018. [CrossRef]
83. Datasure. Qualified Electronic Timestamp Rates. Available online: https://www.datasure.net/en/our-services/eidas-qualified-electronic-timestamp/prices-qualified-electronic-timestamp/ (accessed on 9 March 2026).
84. Disig. Price List for Qualified Electronic Timestamps. Available online: https://eidas.disig.sk/en/qualified-electronic-time-stamps/pricelist/ (accessed on 9 March 2026).
85. DigiStamp. Price of Digital Timestamps from a Trusted Timestamp Authority. Available online: https://www.digistamp.com/subpage/price (accessed on 9 March 2026).
86. Swende, V.B.M. EIP-2929: Gas Cost Increases for State Access Opcodes. Ethereum Improvement Proposals. 2020. Available online: https://eips.ethereum.org/EIPS/eip-2929 (accessed on 9 March 2026).
87. Etherscan. Ethereum Gas Tracker. Available online: https://etherscan.io/gastracker (accessed on 9 March 2026).
88. L2BEAT. Layer 2 Activity Dashboard. Available online: https://l2beat.com/scaling/activity (accessed on 9 March 2026).
89. Sofronie, M.; Brînzea, A.; Bratu, A.; Aciobăniței, I.; Pop, F. A Bidirectional Bridge for Cross-Chain Revocation of Verifiable Credentials in Segregated Blockchains. *Algorithms* **2025**, *18*, 734. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.