BLOCKSTAND

Standardizing On-chain IP Rights Management

Abstract

This work outlines a blueprint for a European on-chain IP rights management for creator and stakeholder identification, and an NFT-based licenses and royalty contracts to encode IP rights and revenue sharing on-chain. This standardized approach could greatly enhance the creator economy by democratizing copyright protection and monetization.

It builds on emerging best practices from industry and aligns with initiatives by European institutions (such as the EU Blockchain Services Infrastructure for trusted and WIPO's efforts toward decentralized identifiers for IP In the following sections, we delve into the system design, a step-by-step user/data flow, a use case analysis of NFT licensing and royalty contracts, and finally the regulatory considerations that guide the design.

As referenced in the previous document a Web3 Passport refers to an on-chain identity credential that users carry across decentralized applications, enhanced with privacy-preserving attestations. It serves as a digital identity in Web3, allowing individuals to prove facts about themselves (e.g. authorship, age, citizenship) without exposing personal data. In essence, a Web3 Passport is an identity based on a DID (Decentralized Identifier) anchored on-chain, to which verifiable credentials can be attached and later verified via ZK proofs. This approach empowers users with control over their identity while enabling trust-less verification by third parties.

The significance of such an identity system in the context of intellectual property (IP) is profound: creators can link their on-chain identity to their works through verifiable claims of authorship, and licensees or platforms can verify those claims without requiring the creator to divulge sensitive data. This paper proposes a standardized framework to protect and monetize digital IP rights (for example copyright) in the creator economy by leveraging these Web3 identity attestations in tandem with NFTs and traditional ERC20 smart contracts.

The core idea is to use non-fungible tokens (NFTs) as license tokens representing usage rights to creative works, and to use smart contracts (potentially with fungible tokens) to automate royalty payments. By registering creative works in an on-chain copyright registry and linking creator identities via ZK attestations, we can establish a tamper-proof record of ownership and licensing.

This framework aims to standardize how creators mint NFT licenses for their content, how licensees obtain and use those rights, and how royalties flow back to rights holders, all while complying with European regulatory requirements. In the Web2 world, enforcing copyright and tracking royalties is cumbersome and opaque; in contrast, our Web3 approach uses blockchain for transparency and smart contracts for automation, providing a more efficient and creator-friendly system.

Design & Standardized User & Data Flow

A proposed architecture comprises four key components working in concert: (1) an on-chain registry for copyright and authorship, (2) Trusted Execution Environments (TEE) for privacy-preserving attestation generation, (3) an NFT-based licensing model (ERC-721 tokens representing licenses), and (4) royalty payment mechanisms using fungible tokens (ERC-20, including MiCA-compliant electronic money tokens). Below is an outline of each component and how data flows between them:

1. **On-Chain Registry**

   A smart contract on a blockchain that serves as a reference point for on chain data ownership attestation issued and verified through reliable off chain data sources (Trust Registry). When a creator produces a new work, they register a hash or fingerprint of the work on this ledger, creating an immutable timestamped record of authorship. The registry entry links the content's identifier (e.g. IPFS hash or a unique content ID) with the creator's decentralized identity (DID). Crucially, this registry is pan-European, meaning it's recognized across EU member states – for instance, an implementation could leverage the EU's EBSI network (European Blockchain Services Infrastructure) to ensure the record is trusted EU-wide. By using a DID method compliant with W3C and the EU standards, the registry can interoperate with European Digital Identity systems. The on-chain registry doesn't store the work itself or personal data, only proofs and references, aligning with the principle of data minimization. It acts as the single source of truth for verifying copyright claims: anyone can query the registry to confirm if a given content hash is registered and who the declared owner is. (In practice, national IP offices or the EUIPO could act as trusted issuers to vouch for these records, bridging traditional copyright registration with the blockchain ledger.

2. **Trusted Execution Environments (TEE) for Attestations**

   Privacy is paramount since creators must not be forced to put personal or sensitive information on-chain. Here the concept of TEE was introduced as a complement to ZK proofs. A TEE is a secure enclave that can perform computations on sensitive data and produce an attestation, a cryptographic proof or signature, that certain verifications were done correctly, all while the raw data remains protected[1]. In this design, a TEE could be used by an authorized party, a trust registry holder (e.g. a copyright office or a credential issuer) to validate off-chain information and issue a credential or attestation. For example, a TEE service might verify a creator's real identity and authorship evidence (such as a digital watermark or the original file) and then issue a verifiable credential asserting "Person X is the author of Work Y (content hash)". This credential is signed and can later be proven on-chain via ZK proofs.

   The role of the TEE is to ensure that even the verification process keeps data confidential – the TEE's remote attestation function allows others to trust that the code ran as intended without leaking data.

   By using TEEs in combination with ZK proofs, It is possible to get a robust attestation service: the TEE can handle complex checks or interact with off-chain databases (e.g. scanning a fingerprint of the work against a database to ensure originality) and output a succinct claim that can be checked on-chain. This hybrid approach leverages the strengths of both technologies – TEEs provide efficient private computing for heavy tasks, while ZK proofs provide trustless verification of facts on-chain. All attestations about identity or content that are generated in TEEs may be anchored to the Web3 Passport (the user's DID). They might be stored off-chain (e.g. in a credential wallet or IPFS) but are referenced on-chain via a hash.

[1] 4 Ways to Compare Trusted Execution Environments and Zero-Knowledge Proofs

In essence, the TEE + ZK mechanism allows a licensee or any verifier to ask "Does this user possess a valid credential proving they are the copyright owner of content X?" and to get a yes/no proof on-chain, without either party seeing any private data. This preserves data privacy as by keeping personal data off the public ledger, only publishing non-sensitive proofs.

3. **NFT-Based Licensing Model (ERC-721,5281)**

Once a work is registered and the author's identity attested, the creator can issue NFT licenses as standardized ERC-721 tokens[2]. Each NFT represents a license to use the work under certain terms. For convenience it is adopt a model where the original NFT (token ID 0, for example) corresponds to the root ownership of the work's copyright, held by the creator, and additional NFTs can be minted as license tokens (sublicenses) linked to that root.

This structure is inspired by the proposed ERC-5218 standard for NFT licenses, which defines a tree of licenses emanating from an original on chain proof[3]. The NFT license contract's metadata includes fields that specify the licensing terms and any compliance constraints.

For instance, metadata might include: a reference to the work (content hash or URI), the identity of the licensor (e.g. DID of the creator, proving they own the rights), the scope of the license (e.g. exclusive vs non-exclusive, territory, duration), usage rights granted (e.g. display, reproduction, derivative works), and a link to human-readable license terms or a smart contract address that enforces them. Compliance flags could also be included, such as an indicator that the license respects certain copyright exceptions or has age restrictions (useful for downstream platforms to automatically enforce content rules).

When a creator mints a new license NFT, the smart contract can enforce that only the copyright owner's address (the holder of the root token) is allowed to do so, thereby preventing unauthorized licenses. The NFT license is then transferable or revocable according to preset conditions. By default, a license NFT might be transferable to allow secondary markets for licenses (e.g. a photographer could sell a license NFT to a publisher, who later transfers it to another publisher if allowed). A standardized approach would allow the creator to specify whether transfer or sub-licensing is permitted, and the smart contract would enforce those rules. Notably, ERC-5218 suggests that if an NFT (root) is transferred, its active licenses should either transfer or be revoked accordingly. A standardized design would incorporate similar logic to maintain consistency of ownership and licensing. he enforcement of licensing terms, particularly in secondary market transactions, remains a key challenge due to the decentralized nature of NFT trading.

While licensing conditions are encoded in smart contracts, the execution of policy rules—such as sublicensing restrictions, regional compliance, or mandatory royalty payments—relies on marketplace adherence or self-regulation by license holders. This creates a vulnerability where licensees may bypass contractual obligations, leading to unauthorized transfers, untracked sublicensing, and revenue losses for the original rights holders.

[2] ERC-721 Non Fungible Token Standard

[3] ERC-5218: NFT Rights Management

To address these challenges, merklized credentials combined with zero-knowledge attestations provide a cryptographic enforcement layer that strengthens on-chain verification of policy conditions before allowing NFT transfers.

By utilizing Merkle trees to structure policy rules and ZK-SNARKs/STARKs for privacy-preserving validation, we introduce a decentralized and automated mechanism that ensures NFT licensing compliance without relying on intermediaries.

A Merkle tree is an efficient data structure that allows multiple policy conditions to be hashed and committed on-chain in a way that verifiers can selectively prove compliance without revealing the entire set of rules. For NFT-based licensing under ERC-5281, it is possible to encode critical sublicensing and transfer constraints as individual leaves within a Merkle tree.

These constraints may include:

- Transfer Restrictions: Defining whether NFT resale is permitted or restricting transfers based on regional or KYC requirements.
- Sublicensing Limits: Setting predefined rules on the maximum number of sublicenses that can be issued.
- Royalty Enforcement Conditions: Ensuring that royalty payments must be settled before transfers can occur.
- License Expiration or Revocation: Encoding a time-based validity period or a mechanism for automatic revocation.

Each of these policy constraints is hashed and stored in a Merkle tree, with the Merkle root stored on-chain in the ERC-5281 smart contract. This structure enables an efficient ZK-proof mechanism to verify whether a proposed NFT transfer complies with the committed policy constraints, without exposing sensitive licensing details.

Once the policy constraints are structured as a Merkle commitment, NFT transfers must be subjected to ZK-proof verification before execution. A licensee initiating a secondary market transaction would need to prove compliance with licensing terms by generating a ZK attestation. This attestation is derived from:

- The licensee's Web3 identity (DID-based credentials)
- The NFT license metadata (linked to ERC-5281 hierarchy)
- The relevant policy conditions stored in the Merkle tree

A zero-knowledge prover then submits a cryptographic proof that the transaction adheres to licensing constraints without revealing all policy rules. The ERC-5281 smart contract includes a verifier function, which checks:

1. Is the current transaction compliant with the encoded licensing terms? (e.g., sublicensing within limits, transfer region compliance)
2. Has the required royalty been paid to the original rights holder?
3. Is the recipient an authorized licensee under predefined terms?

If the ZK proof validates these conditions, the NFT transfer proceeds. Otherwise, the transaction is blocked at the smart contract level, ensuring that policy violations cannot occur even if an external NFT marketplace does not enforce these constraints.

4. **Royalty Payment Mechanisms (ERC-20 tokens issued by EMTs)**

To monetize the licensed IP, the integration with a royalty contract that automates payments to the rights holder(s) whenever the content is used or a license is exercised is essential. This can be realized in a couple of ways. One straightforward approach is using a smart contract that holds and distributes ERC-20 tokens to stakeholders according to predefined split rules. For example, an automated royalty splitter contract could be deployed for each work or each license, which receives usage fees (in tokens) and immediately allocates them: e.g. 70% to the creator, 30% to a publisher. The royalty smart contract can either pull funds from licensees (for instance, a license NFT could be designed such that each time the licensee uses the content, they call a function and pay a micropayment) or push funds to the author (for example, periodic payouts from an aggregated revenue pool). In NFT marketplaces, a common standard for secondary sale royalties is ERC-2981[4] which allows an NFT to declare a royalty percentage and recipient.

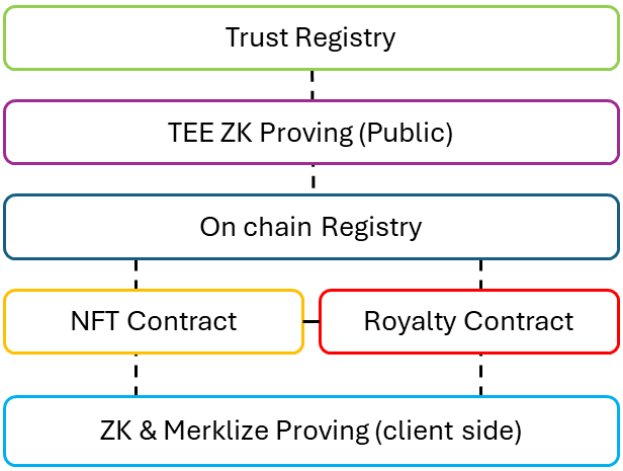
The approach generalizes this: the NFT license terms can include a royalty percentage for certain uses or resales, and a smart contract will enforce it. For instance, if a license NFT is resold to a new holder, the sale transaction can be required by the contract to send X% of the price to the original artist's royalty contract, which then distributes to stakeholders. Beyond resale, consider royalties from end-use: imagine an NFT music license that allows the licensee to stream a song – the streaming platform could integrate with the smart contract so that every play triggers a micro-transaction to the contract, which instantly splits the payment to the songwriter, producer, etc. via on-chain logic.

To process payments, one suggestion may be utilizing ERC-20 tokens[5] for these micro-payments and distributions due to their divisibility and adoption. The system could issue a specific token per project or more simply use existing stablecoins as the currency of payment. The latter is more straightforward and with EMT compliance ensures that using the token is akin to handling electronic money. The royalty contracts must implement proper accounting and compliance checks.

[4] ERC-2981: NFT Royalty Standard

[5] ERC-20 Fungible Token Standard

Because the entire flow is on-chain, transparency is inherently achieved: creators and rights holders can see in real-time how much revenue has been generated and distributed, which is a stark improvement over opaque royalty reporting in traditional media.



User & Data Flow

Hereafter a standardized flow of data and actions from the perspective of the main stakeholders:

1. Content Registration by Creator – On chain Attestation

A creator (author, artist, etc.) first registers their work on the on-chain copyright registry. Using their Web3 identity wallet (which manages their DID and credentials), the creator generates a content credential. This involves computing a unique hash of the creative work (e.g., hashing the image or text file) and sending a transaction to the registry contract that records this hash along with the creator's DID or blockchain address.

The creator's identity is verified during this process via a ZK attestation: for instance, the creator might present a ZK proof that "I possess a verifiable credential for instance from EUIPO or another authority confirming I am the author of this content" . This proof provides an data attestation of ownership empowering the user with an ownership claim and is checked by a smart contract (a ZK verifier contract deployed on-chain) which interacts with the on-chain registry of attestations. If the proof is valid, the content gets an entry in the registry, and the creator's DID is marked as the owner. At this stage, the creator has an on-chain proof of authorship: anyone can see that a DID resolvable to a certain , perhaps pseudonymous real-world identity via EUIPO has registered content with hash XYZ at a certain timestamp. This provides immutable proof of creation date and ownership which is valuable if any dispute arises later.

2. Issuance of License NFT to Licensee

Suppose a licensee (e.g., a buyer who wants to use the content) approaches the creator for rights. They agree on terms, which could be off-chain negotiation for instance via a marketplace app listing. The creator then uses the NFT licensing smart contract to issue a license token (ERC-721) to the licensee's address. This might be done by calling createLicense(tokenId, terms) on the contract, where tokenId is the ID of the original work's NFT, and terms could be a URI pointing to the license agreement text or an identifier of standard terms.

The contract mints a new NFT (say tokenId=1 for the first license) to the licensee. Internally, this action triggers several on-chain checks:

1. The caller must be the owner of the root NFT, ensuring that only the copyright owner has the authority to issue licenses.
2. The license tree data structure is updated, and the newly issued NFT is recorded as a child license of the root NFT.

The metadata of this license NFT 1 includes key information such as:

- The licensor's DID (creator's identity).
- The licensee's DID (either explicitly encoded or inferred as the NFT holder).
- Expiration date, if applicable.
- Rights granted, such as display, reproduction, or sublicensing permissions.
- Merklized Licensing Conditions: A cryptographic commitment to policy rules stored in a Merkle root, ensuring that sublicensing or transfer actions comply with predefined constraints.

Since this license NFT is standardized, any platform, marketplace, or third-party application can parse its metadata to understand permitted rights and constraints. For example:

- If the license is non-transferable, the NFT contract marks it as soulbound (transferable: false in metadata), which marketplaces can detect and automatically prevent resale attempts.
- If sublicensing is allowed, the licensee (holder of license NFT 1) might be able to call createLicense(parentLicenseId=1, ...) to issue a sublicense.

However, before allowing any sublicense issuance, the ERC-5281 contract integrates Zero-Knowledge (ZK) verification, ensuring that:

1. The sublicensing terms comply with the original license policy, verified via a Merkle-proof attestation.
2. The licensee has not exceeded the allowed sublicenses (e.g., a maximum of 3 sublicenses).
3. The sublicensee meets regional or KYC-based constraints, validated through ZK attestations linked to Web3 identity.

This policy enforcement prevents unauthorized sublicensing or resale attempts, ensuring that NFT licenses cannot be exploited in ways not originally permitted by the creator.

3. Verification & Usage

Now that the licensee holds an NFT license, they can exercise their rights. When the licensee goes to use the content (e.g., uploading a song to a streaming service or printing an image in a publication), they must prove that they have the necessary rights. This is where the ZK attestation service verification come into play.

The licensee presents the NFT to a service or platform (by connecting their Web3 wallet to prove ownership) and additionally presents any required licensing conditions. Some licenses might have regional restrictions or require that the user holds a verified attribute, such as:

- Only users in the EU can access the content.

- Only verified adults (18+) can consume the content.
- Non-commercial users must prove their status to use the license.

Such attributes are validated through a Zero-Knowledge proof (ZK proof) derived from Merklized Credentials. The ZK attestation allows the licensee to prove compliance with these conditions without exposing personal information.

The service or platform uses a combination of:

1. On-chain queries to check the NFT contract state (to confirm validity, expiration, and metadata rules).
2. ZK-proof verification against the Merkle root of encoded licensing conditions.

If both conditions are met:

- The NFT license is recognized as valid.
- The licensee is confirmed as entitled to use the content.
- The platform grants access or enables further processing.

Because the licensing terms are encoded in a standardized and programmatic manner, any content platform can automate usage verification.

This results in:

- Automatic enforcement of usage limits (e.g., "License allows 100 uses, block further use after limit is reached").
- Auto-expiration (e.g., "License expires on Dec 2025, restrict access post-expiry").
- Seamless compliance with privacy regulations (GDPR, eIDAS) through privacy-preserving ZK attestations.

This standardized verification process removes the need for manual intervention, making Web3-native content licensing trustless and automated.

4. Royalty Trigger & Distribution

Whenever a licensed content is used or monetized, the royalty distribution mechanism is automatically triggered. For instance, if the licensee generates revenue from streaming or resale, the smart contract ensures automatic payments to the original rights holder.

To prevent royalty evasion in secondary sales, the royalty contract integrates Merklized Credentials to verify compliance:

1. Before an NFT license is resold, the seller must generate a ZK proof confirming that royalty conditions are met.
2. The smart contract checks the Merkle proof to ensure that the correct royalty percentage has been routed to the creator.
3. If verification fails, the transaction is blocked, ensuring that the seller cannot bypass royalty obligations.

This automated enforcement model eliminates reliance on NFT marketplaces for royalty execution. Unlike Web2 platforms that rely on centralized reporting, on-chain verification ensures that creators always receive their royalties in a transparent and auditable manner.

The royalty system:

- Uses ERC-20 tokens (EMTs) for settlements, ensuring compliance with MiCA financial regulations.
- Supports real-time on-chain logging, allowing creators to track their earnings transparently.
- Maintains compliance records for tax and auditing purposes, reducing legal risks for platforms.

5. Updates and Revocation

Throughout the lifecycle of an NFT license, creators may need to update terms or revoke existing sublicenses due to violations or expiration.

The NFT contract includes a revokeLicense(licenseId) function, which:

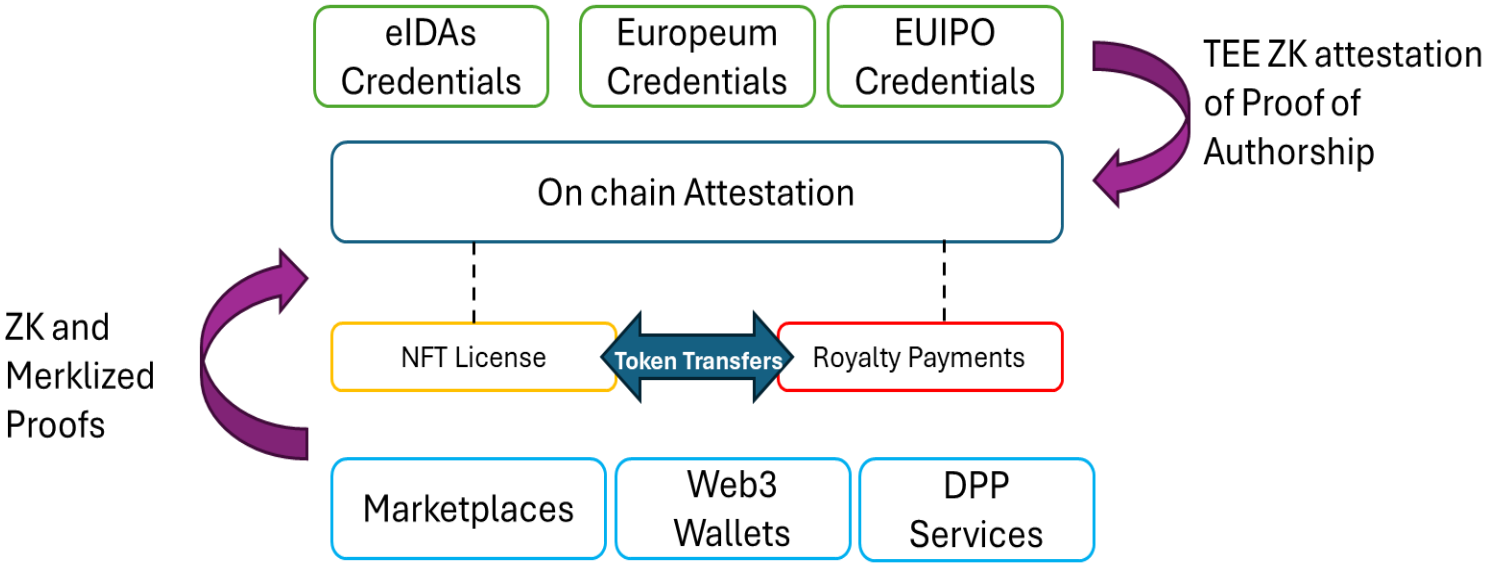
- Marks the license as revoked, preventing further use.
- Prevents unauthorized sublicensing by checking Merkle proofs before issuance.
- Ensures expired licenses are automatically marked as invalid.

In cases where a licensee breaches terms, a ZK proof of non-compliance can be generated and submitted to the ERC-5281 contract, triggering automated revocation. This on-chain policy enforcement prevents malicious actors from misusing sublicenses. Furthermore, if a creator wants to update terms, they can issue a new Merkle root commitment, reflecting modified conditions for new licensees. This enables dynamic licensing updates without requiring full contract redeployment.

To summarize, there two essential metadata type necessary to enable this trust-less verification logic.

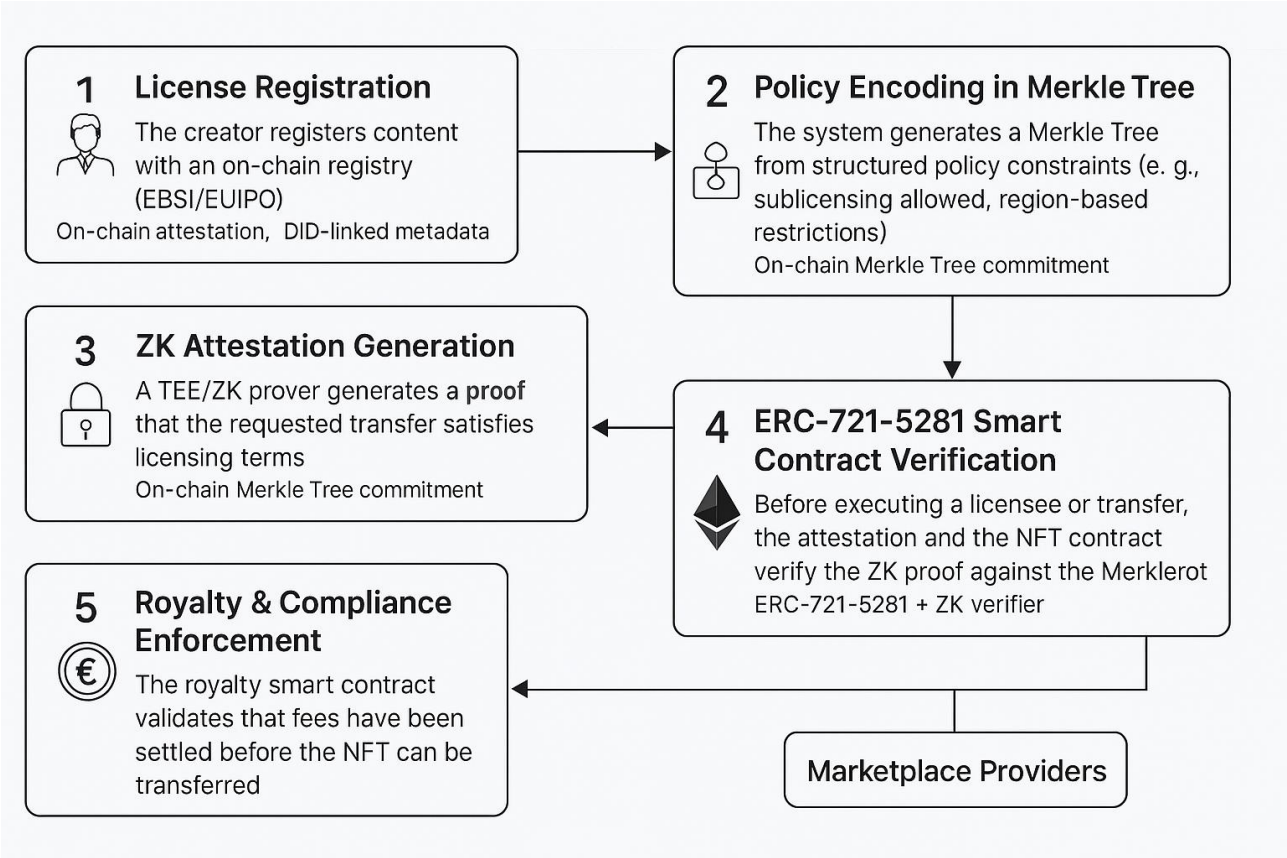
Metadata Type	**Purpose**	**Data stored On-Chain**
On-Chain Attestation Metadata	Verifies authorship and IP ownership (linked to eIDAs/EBSI/EUIPO credentials)	Yes (but only hashed commitments) as ZK Proof generated in TEE.
NFT License Metadata	Defines licensing rights, sublicensing rules, royalty terms, etc.	Only the Merkle root is stored on-chain

The following picture describes the execution of this trust-less verification logic flow



I tried then to recap the main steps in the following table:

Step	Process	Technical Component
License Registration	**The creator registers content with an on-chain registry (EBSI/EUIPO).**	**On chain attestation, DID-linked metadata**
Policy Encoding in Merkle Tree	**The system generates a Merkle Tree from structured policy constraints (e.g., sublicensing allowed, region-based restrictions).**	**Off-chain Merkle Tree commitment**
ZK Attestation Generation	**A TEE/ZK prover generates a proof that the requested transfer satisfies licensing terms.**	**ZK-SNARK or STARK proof**
ERC-721- 5281 Smart Contract Verification	**Before executing a license or transfer, the attestation and or the NFT contract checks the ZK proof against the Merkle root.**	**ERC- 721- 5281 + ZK verifier**
Royalty & Compliance Enforcement	**The royalty smart contract validates that fees have been settled before the NFT can be transferred.**	**ERC-20-based royalty enforcement**



Use Case Analysis: NFT Licensing & Royalty Contracts with Merklized Credential Enforcement

This trustless, enforceable licensing model enables Web3-native IP monetization across various industries:

1. Media & Entertainment: Ensuring fair compensation for digital artists, musicians, and filmmakers by enforcing resale royalties.
2. Enterprise Software Licensing: Automating software sublicensing based on geography and usage restrictions.
3. Publishing & Academia: Enforcing fair use policies for research papers and educational content through cryptographically verified sublicenses.
4. Gaming & Virtual Assets: Allowing players to resell in-game assets with built-in royalty enforcement.

To operationalize the proposed model for NFT-based licensing and automated royalty distribution, this section analyzes the design and implementation of two core smart contract types: a standardized NFT Licensing contract (ERC-721) and a standardized Royalty distribution contract (ERC-20/EMT). These contracts ensure that intellectual property (IP) rights are tokenized, sublicenses are regulated, and royalty payments are automated through Merklized Credentials and Zero-Knowledge (ZK) attestations.

By enforcing licensing constraints at the smart contract level, this model eliminates reliance on centralized platforms for policy enforcement and enables trustless, privacy-preserving copyright protection and monetization.

The NFT Licensing Contract

The NFT Licensing contract is an ERC-721 compliant smart contract designed to tokenize legally binding license rights for copyrighted works. It establishes a structured approach for issuance, transfer, and sublicensing of NFT-based licenses.

Unlike conventional NFT frameworks, which rely on metadata for tracking permissions, this model integrates Merklized Credential verification to ensure that every license issuance, sublicensing, and secondary sale adheres to predefined constraints before execution.

Each NFT license token serves as a verifiable proof of licensing rights, carrying structured metadata to define:

- Root Author Ownership (IP Owner): The original on chain attestation representing copyright ownership, linking to a registrar entity (e.g., EUIPO).
- Child License Tokens (Sublicenses): Issued under ERC-5281 to represent individual licensing agreements, maintaining hierarchical traceability.

Unlike traditional metadata-based constraints, this model stores licensing policies in a Merkle tree, with only the Merkle root stored on-chain. Licensing transactions require a ZK proof validating that the requested action adheres to these cryptographic constraints.

The licensing policies reflected in the merklization process for verification may be listed as the following:

Field	**Description**	**Role in Merklized Validation**
rootId	Links to the original copyrighted asset	Ensures traceability
licensor	DID of the issuer	Verifies issuance legitimacy
licensee	DID of recipient	Requires ZK proof of eligibility before transfer
rights	Enumerates granted rights	Checked cryptographically before execution
duration	Defines start and expiration dates	Enforced via ZK attestation
territory	Restricts geographical use	Verified through ZK proof of compliance
sublicenseable	Indicates sublicensing permissions	Must be validated via Merkle tree proof
royaltyTerms	Defines revenue share	Ensures royalty payment before transfer

By leveraging Merkle tree commitments, licensing policies become immutable yet flexible, allowing updates without requiring full contract redeployment.

The core enhancement of this approach is the Merkle tree-based enforcement of licensing policies. Instead of storing licensing conditions as on-chain metadata, licensing rules are hashed and committed to a Merkle tree, with only the Merkle root stored on-chain. Each licensing transaction requires the submission of a ZK proof, demonstrating that the requested action adheres to the Merkle-committed policy constraints. This ensures that only valid sublicenses can be issued, only compliant transfers can be executed, and policy violations are cryptographically prevented at the smart contract level.

To maintain interoperability and transparency, the NFT licensing contract follows a structured metadata schema, defining key attributes such as root ownership identification, licensor and licensee identities, rights granted, territorial restrictions, sublicensing permissions, royalty terms, and compliance indicators. The contract also may also integrate a hierarchical licensing structure (ERC-5281), wherein sublicenses are issued under a proof of authorship root, forming a verifiable chain of rights.

Each sublicensing request is cryptographically validated using a ZK proof submitted to the ERC-5281 smart contract.

Before approving a sublicense, the smart contract verifies that:

1. The sublicensing action is permitted by the original license terms, as verified against the on-chain Merkle root.
2. The sublicensee meets eligibility conditions, such as geographical restrictions or KYC requirements.
3. All royalty obligations have been settled before sublicensing is allowed.

By integrating Merklized Credential enforcement, the NFT licensing contract eliminates unauthorized sublicensing, prevents arbitrary license transfers, and ensures that on-chain compliance is enforced without exposing sensitive data. This enhanced licensing framework guarantees that NFT-based intellectual property licensing aligns with both commercial and regulatory requirements.

Royalty Contract

The Royalty Contract governs automated revenue distribution for NFT-based IP monetization. The contract ensures that royalty payments are cryptographically validated before NFT transfers occur, preventing revenue evasion in secondary markets. Unlike traditional royalty mechanisms that rely on marketplace adherence, this implementation enforces royalty obligations at the smart contract level, ensuring that NFT sales cannot be completed without a cryptographic proof of royalty settlement.

Under this model, whenever an NFT license is transferred or sublicensed, the seller must submit a ZK proof confirming that the royalty has been settled. If the proof is invalid or missing, the smart contract automatically blocks the transaction, thereby enforcing on-chain policy compliance without relying on external marketplaces.

To ensure financial stability and regulatory alignment with the Markets in Crypto-Assets Regulation (MiCA), the royalty contract integrates Electronic Money Token (EMT) payments, ensuring that royalties

are processed using MiCA-compliant stablecoins. The contract rejects non-compliant assets, protecting rights holders from financial instability and ensuring legally compliant revenue distribution mechanisms.

This automated royalty enforcement system operates as follows:

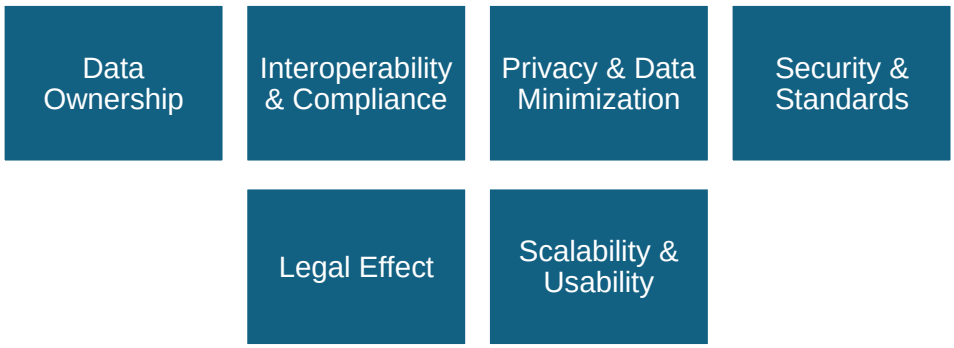
1. A buyer initiates an NFT license transfer on a secondary market.
2. Before executing the transaction, the smart contract checks for a ZK proof, verifying that:
    - The royalty payment has been made to the original rights holder.
    - The buyer meets regional and licensing restrictions, validated against the on-chain Merkle root.
3. If verification fails, the transaction is blocked, ensuring that licensing policies and financial obligations are enforced before execution.

This cryptographically enforced approach eliminates reliance on centralized NFT marketplaces for royalty enforcement and ensures that NFT creators receive fair compensation for secondary market transactions.

Requirements Set

A side of different technical options it is in the author intent try to summarize a set of key requirements and capabilities that the NFT license and royalty contracts must fulfil to be successful.

A few pillars have been identified to guide the standardization of the service:



- **Authenticity & Ownership Assurance:** The system must guarantee that only the legitimate copyright holder can register a work and issue licenses for it. This is achieved through on-chain checks against the identity registry and attestation of ownership Technically, this means the contracts need access to the attestation (perhaps via a boolean function like isOwnerVerified(address user, bytes32 contentHash) implemented as a call to a ZK verifier contract). Business-wise, this assures creators their rights won't be stolen on-chain, and assures buyers that the NFT license they purchase is coming from the true owner (preventing fraud).
- **Interoperability & Standard Compliance:** Use of ERC-721 and ERC-20 standards so that existing wallets, exchanges, and analytics tools can work with these tokens. Align metadata with schemas that can be recognized by intellectual property databases or other blockchain

networks. The contracts should be chain-agnostic enough that they could be deployed on EBSI or a public chain and still operate similarly.

- **Privacy & Data Minimization:** Personal data of users (creators/licensees) should never be published on-chain in plain form. This requirement is fulfilled by using DIDs and ZK proofs for identity, and by keeping any sensitive details (like the actual content file, or addresses of individuals if private) off-chain. GDPR compliance is essential: storing a hash of personal data may raise some concerns for the data by itself, and so it must be compressed and reduced as much as possible leveraging Zero Knowledge cryptography where possible. Our license metadata might contain a DID (which is arguably personal data if it can be resolved to a person). Using pairwise-pseudonymous DIDs or one-time tokens for different works can mitigate correlation. The contracts should not store things like names, emails, etc. – any such need can be handled via off-chain credentials. This also implies that if a user wants to invoke their right to be forgotten, the off-chain data can be erased, and the on-chain hash becomes less sensitive. So from design to implementation, data minimization is a guiding principle.

- **Security & Reliability:** Smart contracts must be audited and secure to prevent hacks that could steal NFTs or tokens. On a business level, stakeholders need confidence that if they rely on this system, funds won't be lost to exploits and licenses can't be tampered with. Techniques include using standardized well-tested libraries for ERC-721 and ERC-20 logic (e.g. OpenZeppelin[6] contracts), and possibly formal verification for critical components like the ZK verifier integration (since an incorrect verification could allow someone to falsely claim authorship). The system should also handle edge cases gracefully (e.g., what if a licensee loses their private key? Perhaps a procedure to reissue a license to a new address given proof of identity handled off-chain by keyless authentication methods (passkeys, OTP).

- **Scalability & Usability:** The user experience should be as smooth as possible. While this is more about apps built on top of these contracts, the contracts should be designed to not hinder UX. For instance, gas costs for minting licenses should be reasonable, possibly achieved by using efficient data structures and avoiding heavy on-chain storage which could be achieve by using zk compressors.

- **Legal Recognizability:** The structure should align with legal frameworks. In Europe, a voluntary registry like these complements existing copyright law as copyright exists even without registration). Ensuring the framework is not contradicting any law is a paramount; for example, EU copyright directives allow creators to license on their terms, the framework allows that. This also aligns with initiatives like EBSI-ELSA on product counterfeiting. In fact, in this design a model could feed into this system. EUIPO has been exploring blockchain for anti-counterfeiting and could to copyrights. It is crucial to incorporate the WIPO recommendations of using DIDs for IP management[7].

In essence, the NFT licensing and royalty contracts form the backbone of the on-chain IP economy. When designed and implemented to meet the above requirements, they enable a self-sustaining ecosystem: creators and licensees connect directly via smart contracts, value is exchanged automatically, and all parties (including oversight entities) have confidence in the system's fairness and legality. The next section turns to regulatory alignment, discussing how this architecture complies with

[6] OpenZeppelin Contract Github repo
[7] WIPO - Blockchain for IP ecosystem WP

and complements current European regulations and standards, ensuring that our technical design is not operating in a legal vacuum but is part of a holistic approach to digital IP rights management.

Regulatory Alignment

Designing an on-chain copyright and licensing system in Europe necessitates careful consideration of the regulatory landscape. The solution must not only comply with laws like GDPR and MiCA, but also integrate with broader EU initiatives on digital identity (EIDAS/EUDI) and intellectual property. In this section, I examine how the proposed framework aligns with key regulations and standards, ensuring that it is legally robust and can gain institutional support.

GDPR & Data Minimization: The General Data Protection Regulation (GDPR) is central to any system dealing with personal data in the EU. A fundamental challenge is that blockchain's immutability conflicts with GDPR's requirements (like the right to erasure). Our design addresses this by minimizing on-chain personal data to near-zero. Personal information (names, etc.) is kept off-chain in secure credential issuers or wallets, and only hashes or proofs end up on-chain. As one guidance notes, one should "maintain any form of personal data at an off-chain level with only a hash leading to the data recorded on the ledger", so that if a deletion is needed, the off-chain data can be erased and the on-chain hash becomes less sensitive[8]. For example, the creator's identity attestation is stored off-chain; on-chain might store just a hash of a certificate. This means if a creator decides to revoke their consent or a licensee wants to exercise privacy rights, the off-chain issuer can delete the credential, and no readable personal data persists on-chain. Moreover, by using zero-knowledge proofs, even the act of verification doesn't reveal personal data – e.g. proving "I have a credential that says I'm the author" does not reveal the author's name or address, only the fact that the proof is valid. This aligns with GDPR's principle of data minimization and privacy by design. This also ensure that any content data (which could potentially relate to personal data if the content itself had personal info) is hashed and/or stored off-chain (like on IPFS or a private database) rather than recorded in full on a public chain. Another aspect is user consent and control: individuals (creators) will explicitly choose to register their works and link their identity; nothing is pulling personal data on-chain without consent. GDPR also refers about pseudonymization use of DIDs is a form of pseudonymization. If the DID method is well designed (e.g. rotating identifiers), it can ensure that correlating activities to a single user is hard for outsiders, preserving privacy. In summary, the architecture is GDPR-compliant by design: it keeps personal data off the ledger, uses encryption/hashing for any references, allows for deletion of off-chain data, and uses privacy-enhancing tech (ZKPs, TEEs) to avoid unnecessary data exposure.

MiCA & Use of EMTs for royalty payments: the Markets in Crypto-Assets Regulation (MiCA) is the EU's regulatory framework governing crypto assets, including stablecoins. Since our system utilizes token-based payments for royalties, it is crucial to ensure compliance with MiCA to avoid regulatory issues. To this document, I align with MiCA's classification by using Electronic Money Tokens (EMTs) blockchain-based equivalents of e-money, akin to regulated stablecoins.

Under MiCA, an EMT is defined as a token that maintains a stable value by referencing a single fiat currency and is issued by an entity authorized under electronic money regulations. In practical terms, this means that if a Euro-pegged stablecoin issued by a licensed e-money institution, which complies with reserve and redemption requirements, it functions much like traditional electronic money. The

[8] Are blockchain and GDPR unintelligible? | activeMind.legal

advantages of this approach are twofold: first, value stability (e.g., if a creator receives a 100 EUR token, its value remains equivalent to 100 EUR in fiat), and second, regulatory clarity, as EMTs are legally recognized in the EU and do not carry the risks associated with unregulated crypto assets. The system merely accepts EMTs as a payment mechanism, which does not introduce additional regulatory burdens. Additionally the model could create an ERC-20 token representing fractional royalty rights, the situation would be more complex. A tokenized royalty right might qualify as an Asset-Referenced Token (ART), triggering requirements such as a prospectus, additional oversight, and financial reporting obligations. To avoid regulatory complexity, we strictly use existing stable-value tokens rather than issuing new royalty-related assets.

EU Copyright and IP Frameworks: Copyright law in the EU and globally via treaties like Berne Convention automatically grants rights to creators when they fix their work in a tangible form. Registration is generally not required, but a voluntary registration may be extremely helpful for proof of authorship and managing rights in the use of digital services. This approach may provide a pan-EU "voluntary registry" that does not replace the law but supplements it. For instance, a license NFT is essentially a contract offering certain rights, which is perfectly allowed as long as the rights holder agrees. We must ensure licenses encoded can reflect all the nuances of copyright exceptions and limitations – e.g., no license can override moral rights where they are unwaivable, etc., but that's in the hands of the licensor when defining terms.

One big regulatory challenge is interoperability of copyright enforcement systems across EU states. A shared IP registry under management of EUIPO could be a solution. If adopted, all member states would refer to the same ledger for registrations, avoiding fragmentation. This echoes the goal of the Europeum-EDIC and EBSI initiatives to harmonize such records. To support this thought, the EBSI ELSA platform[9] which already follow similar approach to protect trademarks and address product counterfeiting could be extended for the management of copyrights. The benefit is that if the EUIPO and national IP offices integrate with EBSI, the identities of creators or the verification of a work can be certified by those authorities, which our contracts can recognize. For instance, the creator could obtain a verifiable credential from EUIPO confirming they've deposited a copy of the work or claiming authorship; then they register on-chain with a ZK proof of that credential. This forms a bridge between traditional IP infrastructure and the blockchain.

On the global stage, WIPO (World Intellectual Property Organization) is looking at blockchain and has highlighted the need for decentralized identifiers in IP management[10]. By using DIDs in the system, we ensure interoperability with WIPO's vision. If WIPO creates a global IP registry or facilitates inter-registry communication, our use of globally resolvable DIDs for creators and content can plug into that. WIPO's focus on a "unique global IP registry"[11] and DID standards suggests that our approach is future-proof: each piece of content could have a DID or content ID that could be recognized across jurisdictions, and each participant has a DID – allowing cross-border licensing deals to be automated if other regions adopt similar approaches.

Worthy to mention is industry led initiatives such as the Coalition for Content Provenance and Authenticity[12] (C2PA). It represents a significant industry effort to standardize the use of digital content

[9] EBSI ELSA Project (EUIPO)

[10] WIPO: blockchain for IP ecosystem WP

[11] WIPO Global Identifier Project

[12] Coalition for Content Provenance and Authenticity Website

credentials, serving as an efficient tool for managing intellectual property (IP) rights. The C2PA aims to address the prevalence of misleading information online by developing technical standards that certify the source and history (provenance) of media content.

European Trust Framework:

The European Commission is rolling out the European Digital Identity Wallet[13] (EUDIW), designed to store digital identifiers and Verifiable Credentials (VCs) in compliance with the eIDAS 2.0 regulation. Given that system relies on zk-based on-chain attestations for NFT licensing and royalty distribution, integrating EUDI-based identity verification may strengthens trust and compliance in the process of issuing the on-chain attestation.

The European Blockchain Services Infrastructure[14] (EBSI) provides a foundational layer for credential issuance and verification. In this context, an on-chain attestation for private NFT licensing contracts could be classified as a Non-Qualified Attestation of Attributes a cryptographic proof recorded on EBSI and derived from a public data sources of national data IP registries by trusted authorities within the Europeum EDIC framework verified through EUIPO EBSI ELSA. This structure ensures that licensing contracts and royalties are accessible only to credentialed participants, while preserving privacy through zero-knowledge proofs (ZKPs).

A creator, for instance, could use their national eIDAS identity or an EUDI Wallet to establish their on-chain identity as Web3 passport. This could be achieved by signing a verifiable attestation with their national eID, linking it to a Decentralized Identifier (DID). Alternatively, EUIPO or another trusted issuer could issue an authorship credential directly into the creator's EUDI Wallet upon registering their work.

Furthermore, the EUDI Wallet Architecture Reference Framework[15] (ARF) emphasizes pseudonymity and selective disclosure, allowing users to prove attributes (e.g., proof of authorship, residency, or professional qualification) without revealing their full identity. This aligns seamlessly with our ZK-proof approach, where a user can demonstrate ownership of a qualified attestation without exposing unnecessary personal details. Additionally, EUDI's strict GDPR compliance aligns with our privacy-preserving architecture, which enables selective disclosure while preventing unauthorized access to sensitive identity data.

By aligning our NFT licensing system with EUDI Wallet standards and EBSI credential model, as Europe we reinforce legal validity and trust in digital rights management. An NFT license tied to an EIDAS-verified identity ensures strong legal standing for rights enforcement—if needed, parties could reveal their identities in legal disputes.

Furthermore, the support for Advance Electronic Signatures (AES), would represent a equitable solution to allow creators to cryptographically sign the on chain attestation to enable simultaneous licensing terms executions.

[13] European Digital Wallet Website
[14] EBSI Website
[15] EUDIW ARF Github Rep.

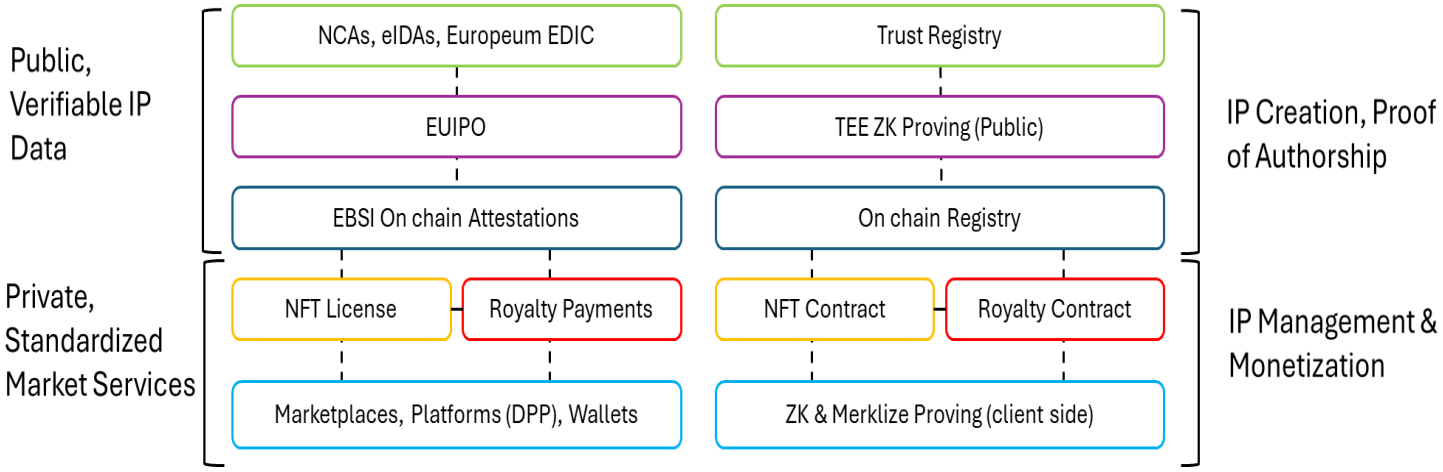
Conclusion

With the following deliverable it is possible to envision a solution to leverage solutions to enable a standardized management and monetization of IP rights (for example within the specific context of copyrights). The foundation of the system is an on-chain registry that formalizes authorship and copyright claims. This registry operates within the EBSI (European Blockchain Services Infrastructure) framework, utilizing Verifiable Credentials (VCs) to record and validate authorship claims in a legally recognized and decentralized manner. The EUIPO (European Union Intellectual Property Office) plays a regulatory role, ensuring that copyright registrations align with existing EU intellectual property laws and facilitating interoperability between national copyright offices. Additionally, the Europeum EDIC, as a European Digital Infrastructure Consortium, provides governance oversight to align the registry with EU-wide standards, enabling seamless integration with member-state copyright registries. Given its expertise in privacy-preserving identity solutions, private market providers may support the issuance of cryptographic proofs linking to trusted authorship records to individual creators while minimizing unnecessary on-chain data exposure.

To preserve privacy while ensuring the verifiability of rights ownership, the system employs Trusted Execution Environments (TEEs) for generating attestations. These TEEs allow encrypted copyright claims and licensing terms to be verified without exposing the underlying sensitive data. The EBSI network, which may include privacy-preserving credential verification mechanisms, serves as the backbone for identity attestations and authorship verification within this module. Market smart contract providers, specializing in confidential computing solutions, can develop modular execution layers that integrate TEEs, ensuring compliance with data minimization principles by issuing zero-knowledge (zk) attestations, ensuring that licensing credentials and copyright claims are verifiable without revealing unnecessary information about the creator or licensee.

The licensing model is tokenized using ERC-721-based NFTs, where each token represents a specific licensing agreement tied to a copyrighted work. These NFT licenses can encode terms such as territorial usage rights, duration, and transferability. The Europeum EDIC ensures that these licensing mechanisms adhere to European regulatory and interoperability standards, allowing cross-border recognition of digital rights. EUIPO, in its oversight role, establishes the legal enforceability of these NFT licenses within the European intellectual property framework. Market smart contract providers implement and maintain smart contracts that define the rules for licensing transactions, ensuring that license transfers and revocations are executed in accordance with predefined conditions. EBSI, in turn, supports this framework by linking NFT licenses to verified authorship claims through Verifiable Credentials, adding a legally recognized layer of authenticity to the on-chain licensing process.

Finally, to ensure that creators receive royalties in a seamless and compliant manner, the system may consider integrating fungible tokens (ERC-20), including stablecoins and MiCA-compliant electronic money tokens. The EUIPO provides guidance on structuring these payments in accordance with EU copyright law, ensuring that smart contract-based royalty settlements are legally enforceable. Market smart contract providers develop automated royalty distribution contracts that execute payments whenever licensed content is used or resold. Additionally, MiCA-compliant electronic money issuers facilitate transactions using regulated digital euros or stablecoins, ensuring that payments remain within the EU's financial compliance framework. EBSI plays a role in verifying these transactions within its infrastructure, ensuring that only authorized parties can trigger royalty distributions while preserving the transparency and auditability of payments.



Together, these components establish a cohesive and interoperable ecosystem for copyright management, balancing decentralization with regulatory compliance. By integrating key European institutions such as EUIPO, EBSI, and Europeum EDIC, alongside privacy-preserving and smart contract innovations from market providers, this framework provides a legally sound and technologically advanced model for digital rights protection and licensing in the EU.

Eugenio Reggianini