NGI TRUST CHAIN

D3. ‘IMPLEMENTATION, DEPLOYMENT, TESTING, DEMOSTRATION AND VALIDATION’

>DECAST<

09/06/2025

Funded by the European Union

TrustChain Project. Funded by the European Union under CA No 101093274. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Due date	30/05/2025
Submission date	09/06/2025
Team	Decast.Live
Version	1.0
Authors	Shivam Dhawan, Yasrab Baig, Peyman Pourjafar, Ahmad Arshad.

Disclaimer

Copyright notice

The TrustChain Consortium is the following:

Participant number	Role	Participant organisation name	Short name	Country
1	COO	EUROPEAN DYNAMICS LUXEMBOURG SA	ED	LX
2	BEN	F6S NETWORK LIMITED	F6S	IE
3	BEN	UNIVERZA V LJUBLJANI	UL	SI
4	BEN	ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS - RESEARCH CENTER	AUEB	EL
5	BEN	FUNDACION CIBERVOLUNTARIOS	CIB	SP
6	BEN	CONSORCIO RED ALASTRIA	ALA	SP
7	BEN	TIMELEX	TLX	BE
8	BEN	ETHNIKO KAI KAPODISTRIAKO PANEPISTIMIO ATHINON	NKUA	EL
9	AP	CITY UNIVERSITY OF LONDON	ICS	UK

Funded by the European Union

TABLE OF CONTENTS

Funded by the European Union

LIST OF FIGURES

Funded by the European Union

LIST OF TABLES

Funded by the European Union

ABBREVIATIONS

DC	Dissemination and Communication
DID	Decentralised Identifiers
DIH	Digital Innovation Hub
DLT	Distributed Ledger Technology
EDIH	European Digital Innovation Hub
EEN	European Enterprise Network
EIC	European Innovation Council
EU	European Union
GA	Grant Agreement
GDPR	General Data Protection Regulation
NCP	National Contact Point
NGI	Next Generation Internet
NGO	Non-Governmental Organisations
OC	Open Call
OC#3	Open Call #3
SEO	Search Engine Optimization
SME	Small and Medium-sized Enterprises
SSI	Self-Sovereign Identities
WP	Work Package

Funded by the European Union

1. EXECUTIVE TECHNICAL OVERVIEW

While our original understanding of the problem focused on Infrastructural and Informational issues arising at a Slow and Fast rates. Our problem matrix comprised of Penetration Breaches, Video Misinformation, Cybersecurity Threats and their Economic Impact. Where our solution aimed at decentralizing live event streaming, while talking to many end users and focus groups, our understanding has since matured to recognize that **the core issue is not just distribution, but the trust, verifiability, and misuse prevention of live and recorded media**.

The evolving landscape of **deepfake threats, AI-driven misinformation, and centralized platform censorship** has reaffirmed our hypothesis; but also forced a stronger emphasis on **identity verification, content authenticity, and modular infrastructure design** for different privacy or compliance use cases.

We understood the problem in a further granularity and recorded the themes that frequently occurred while listening to the problems users state, viz;

Centralized Vulnerability, privacy concerns, AI scam fears, Event Security from hacks and 'video bombing', with other keywords popping up as surveillance, censorship at source, media proofs, etc.

It influenced our implementation decisions:

- We built **multi-layered authentication modules (Identifier)**, that involves DID integration, wallet logins for web2 and web3 session ID binding. Next layer of protection with **in-call authorizations** using mechanisms such as liveness detection, live verifications to address audience authenticity, privacy.

- Our AI workflows now include **content summaries, NFT-gated media access, and smart contract actions**, enabling verifiability for both governance and creator content.

Decast.Live is architected as a **modular, decentralized media infrastructure** platform; merging real-time live casting, post-production AI tools, and verifiable media storage. The architecture is defined as:

1. **Authentication Service Layer**: Binding identity claims with usage context

2. **Authorization & Security Layer**: Session-bound access at user & event levels

Funded by the European Union

3. **AI & Application Compute Layer**: Confidential compute, FHE/TEE-compatible

4. **Accessibility Layer with DePIN**: Enables distributed, attack-resilient deployment options with network deployment on operator nodes.

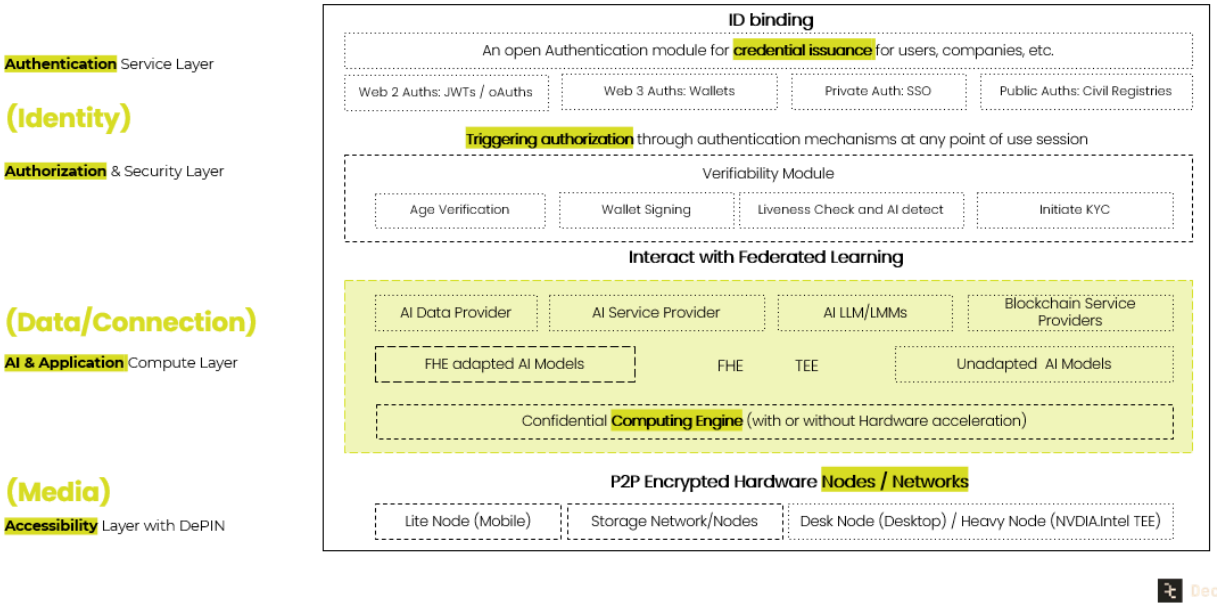

Figure 1 Decast Ecosystem Architecture

1. **Identifier Module**: Authentication and Authorization

2. **Data and Connectivity Module:** AI and Appplication layer to host dAPPs.

3. **Media Module:** Storage classifier to manage Accessibility of media and data streams that connects and expands infra with other networks and providers.

The key components and modules developed are:

Casting Engine

- Multi-stream support (YouTube, Facebook, custom RTMP)
- On-chain triggers and NFT gating

Funded by the European Union

- Recording and indexing of media assets

Authentication & Identity Layer

- DID integration with Web2 and Web3 identity systems
- Wallet-based and session-bound logins (JWT, oAuth, SSO)
- In-call live verification features (liveness detection, age-gating, KYC-ready)

AI Compute Layer

- AI tools for summarization, segmentation, and publishing automations
- Transcripts, smart clips, and social-ready content creation
- Federated learning readiness for privacy-preserving model improvements

Verifiability and Access Control Module

- NFT-gated content access
- Smart contract triggers during/after casting
- Metadata indexing for on-chain proofs and media traceability

DePIN Infrastructure

- Early-stage decentralized compute/storage layer using private GPU nodes
- Modular architecture to support expansion with partner nodes (e.g., Sia, Partisia, SKALE, ETH Swarm)
- Deployment-agnostic node stack: server, desktop, or mobile lite nodes.

Funded by the European Union

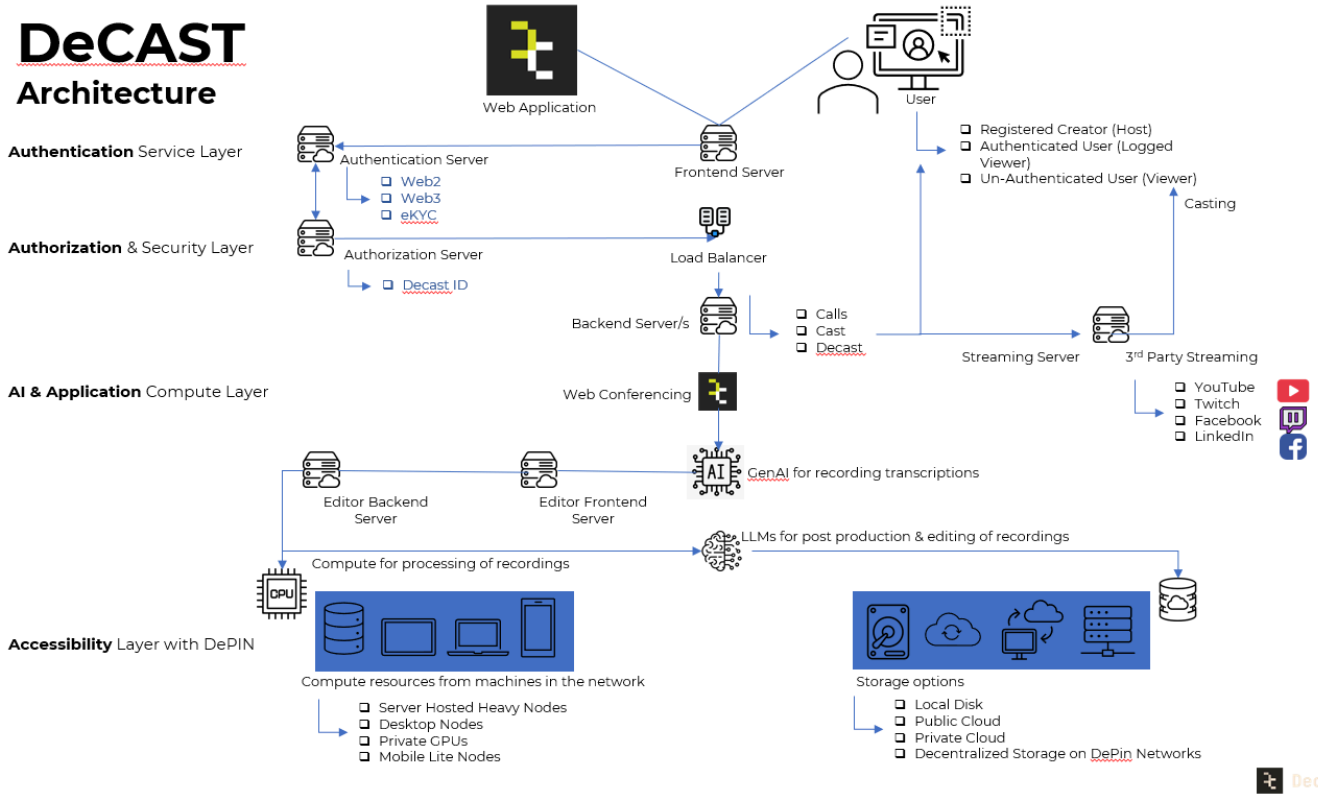

Figure 2 Decast Platform Architecture

TRL Achieved

6, system/prototype demonstrated in a relevant environment via 100-user load tests and mainnet DID resolution flows.

Code & deployment status

- **Public repositories** hosted on GitHub:

  https://github.com/NGI-TRUSTCHAIN/Decast

- **API documentation** accessible via Swagger at

  https://did.decast.live/docs

Funded by the European Union

Sectoral relevance

Our platform tackles cross-chain compliance between private and public blockchains, video streaming, and tamper-proof recordings in legal settings. By combining traditional cloud scaling with decentralized storage, we deliver a solution that is both enterprise-grade and user-centric.

- **Content Creators:** Studios needing private streaming, content control, investigative journalists.
- **Event Organizers:** Businesses hosting webinars, trainings, conferences with seat management, audit trails.
- **Legal & Compliance:** Law firms, courts requiring tamper-proof recordings, on-chain evidence.

Alignment with TrustChain & OC4 objectives

- **Decentralization**: user-owned keys, modular open-source components, multi-chain DID synchronization.
- **Privacy-by-design**: ZKP-based selective disclosure, GDPR-compliant data-deletion APIs.
- **Cross-chain support**: out-of-the-box compatibility with Ethereum, Polygon, SKALE testnets.

Funded by the European Union

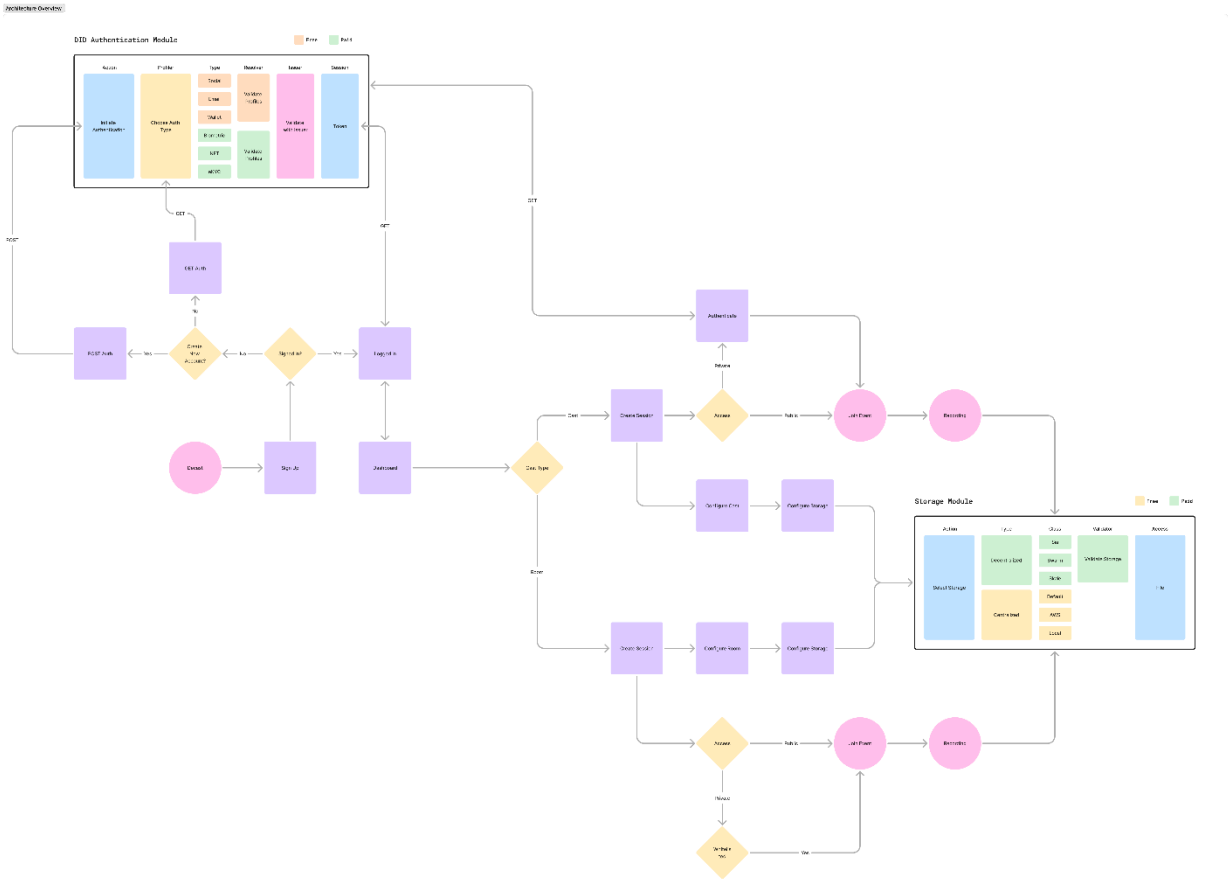

Figure 3 Platform Flow Chart

2. SYSTEM ARCHITECTURE, IMPLEMENTATION AND DEPLOYMENT

This section dives into how DECAST.LIVE's components fit together, what we built in D3, and how we've stood them up in a real-world–like environment.

2.1 Updated System Architecture Diagram

We've refreshed our hexagonal design to show exactly which modules went live in D3, along with their external dependencies and integrations.

Funded by the European Union

- **Auth Microservice** connects to Redis (for the Identifier), ethr-did-resolver, and Passport-based OAuth providers.
- **Identifier Middleware** dynamically routes requests to DID-JWT, social-OAuth, or wallet flows.
- **API Gateway** fronts every request, exposing a unified REST interface.
- **Storage Service** writes to AWS S3 and to Sia/Swarm via its SDK.
- **Video Infrastructure** orchestrates WebRTC sessions and FFmpeg transcoding.

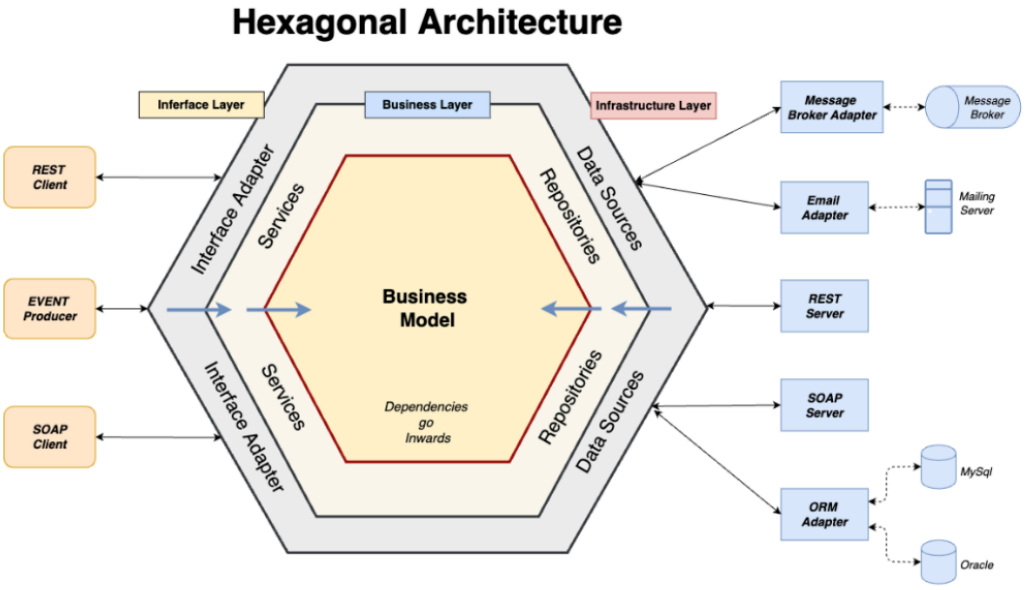

Figure 4 Hexagonal Architecture

2.2 Architectural Overview

Funded by the European Union

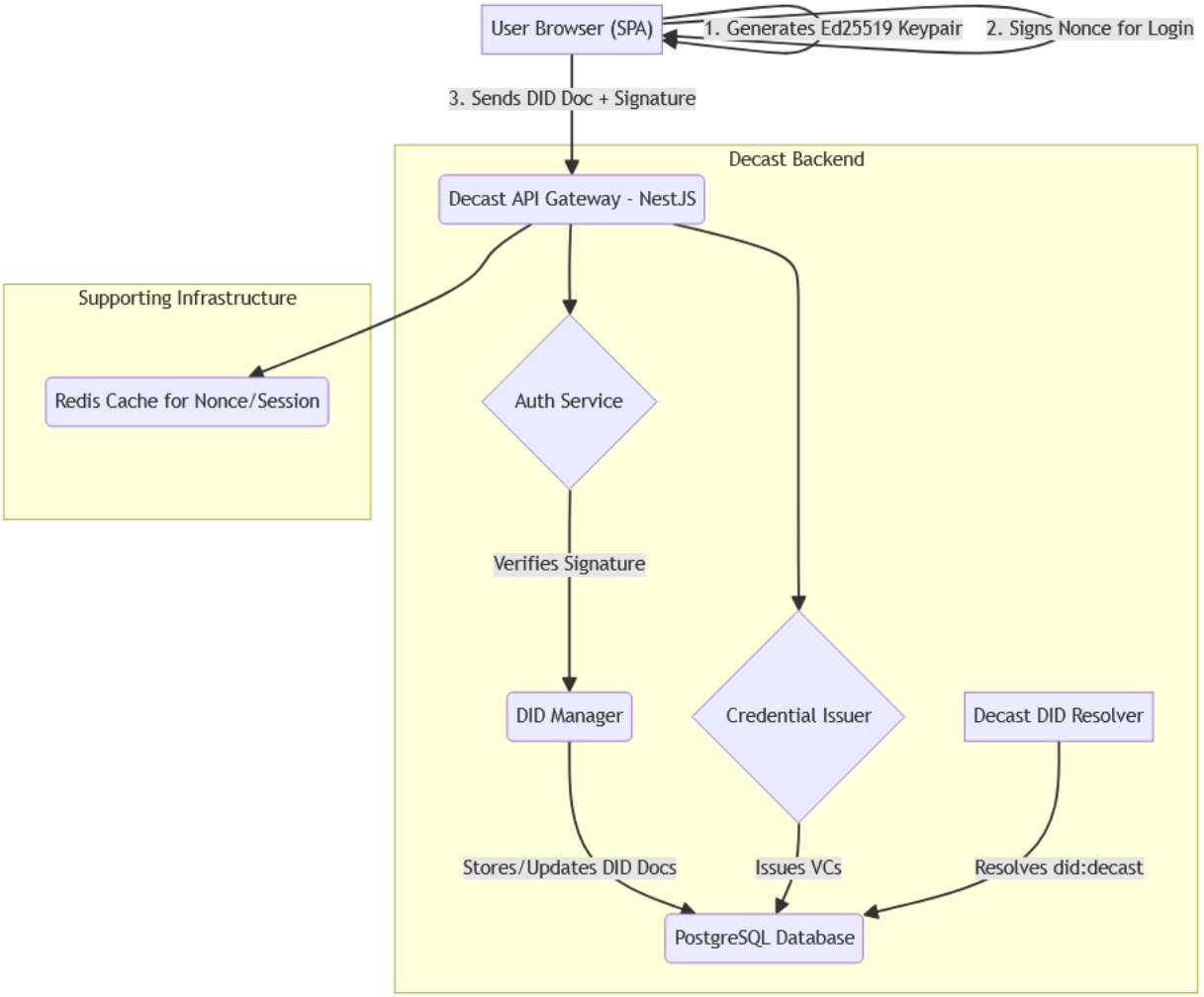

Figure 5 DID Architecture Flow

Client-Side (User Browser - SPA)

1. **Generates Ed25519 Keypair**

   The user's browser (Single Page Application) generates a new Ed25519 public-private key pair.

2. **Signs Nonce for Login**

Funded by the European Union

A nonce (random challenge string) is requested from the backend and signed using the user's private key to prove ownership.

3. **Sends DID Document + Signature**

   The DID document (containing the public key and DID identifier) and the signed nonce are sent to the Decast backend.

Backend (Decast)

1. API Gateway (NestJS)

Handles incoming requests and routes them to the appropriate internal services.

2. Auth Service

- Receives and **verifies the signature** using the DID document and public key.
- Checks the nonce from the **Redis Cache** to prevent replay attacks and manage sessions.

3. DID Manager

- Responsible for storing and updating DID documents in the PostgreSQL database after signature verification.

4. Credential Issuer

- Issues Verifiable Credentials (VCs) upon successful authentication.
- Stores the credentials in the **PostgreSQL Database**.

5. Decast DID Resolver

- Resolves did:decast identifiers into DID documents.
- Ensures DID documents are queryable and usable within the ecosystem.

Deployment Status

Funded by the European Union

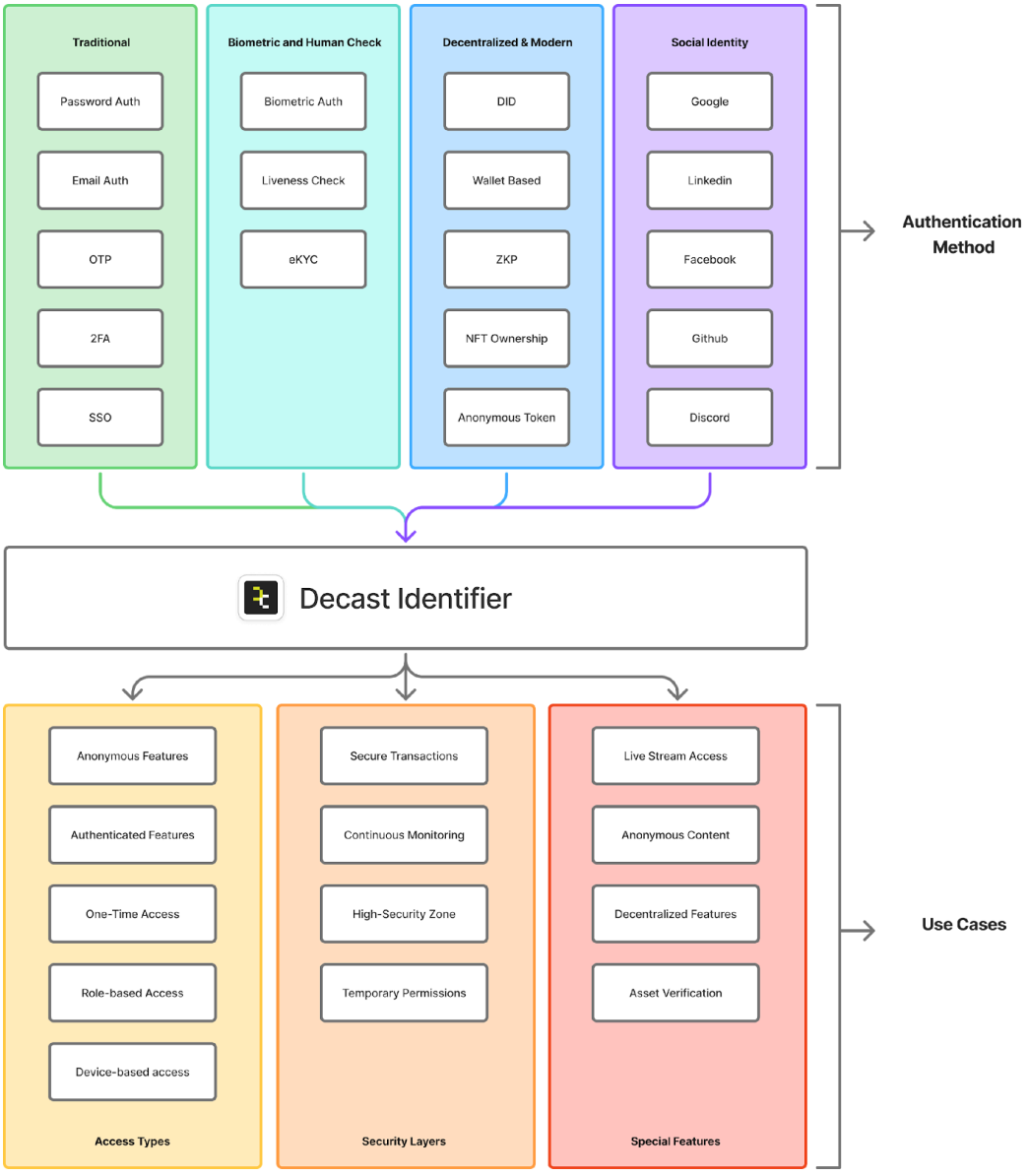

Figure 6 Decast Identifier

Funded by the European Union

Component	Functionality	Tech Stack	Status
Auth Microservice	DID-JWT issuance, social & wallet login flows	NestJS, did-jwt	Completed
Identifier Middleware	Context-aware selection of authentication strategy	TypeScript, Redis	In-testing
API Gateway	Request routing, rate-limiting, unified OpenAPI docs	NestJS, Swagger	Completed
Storage Service	File upload/download, permission checks	Node.js, AWS S3 SDK, Sia SDK	In testing
Video Infrastructure	Live casting, recording, and format transcoding	WebRTC, FFmpeg, Socket.io	Completed
Shared Library	Common utilities: validation, crypto ops, logging	TypeScript	In-testing
ZKP Module	Selective disclosure proofs for privacy-preserving auth	Decast ZKP	Completed

Table 1 Component Status Table

Funded by the European Union

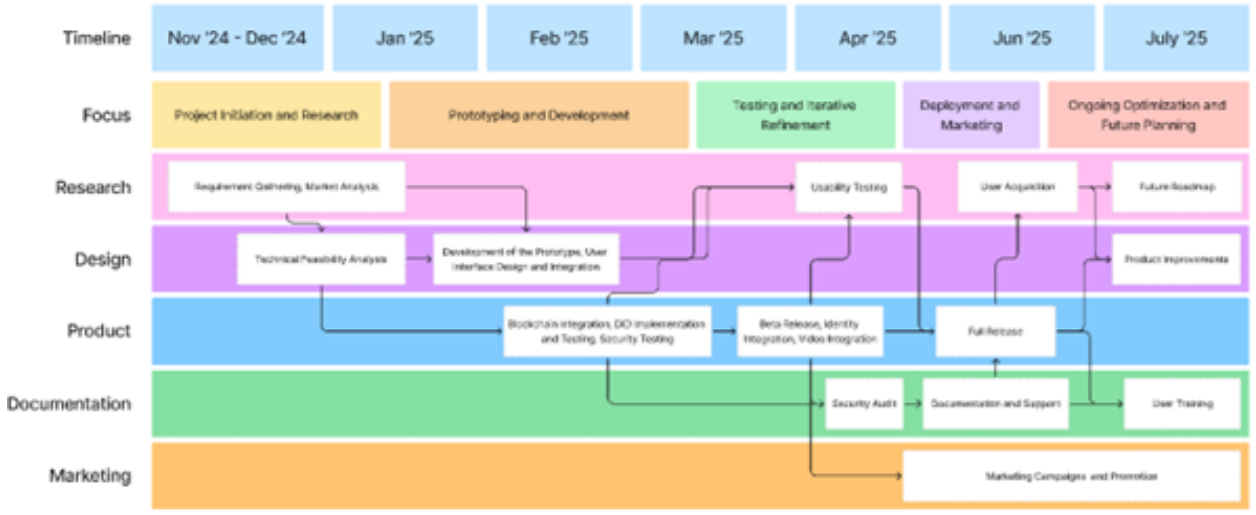

Figure 7 Project Gantt

2.3 Deployment Environment

- **Cloud Infrastructure:** AWS (Frankfurt region) using EC2 for services, S3 for object storage
- **Blockchain Testnets:** Ethereum and Polygon for DID anchoring.
- **Containerization & Orchestration:** All services packaged as Docker images; Docker Compose drives our current CI pipeline, with Kubernetes readiness on the roadmap.

2.4 Evidence of Integration & Operability

- **CI/CD Pipelines:** Every PR triggers unit, integration, and smoke tests (via GitHub Actions); only passing builds are deployed to staging.
- **Load-test Reports:** We ran 100-user concurrency tests using Artillery, holding 95th-percentile auth latency under 200 ms.
- **CLI & Dashboard Screenshots:** Captured flows of "login → create stream → upload recording" (see Appendix).

Funded by the European Union

- **Test Suites:** Postman collections with Newman reports validate end-to-end API behavior.

2.5 Security & Compliance Measures

- **Privacy-by-Design:** ZKP module lets users prove claims (e.g. age) without revealing raw data.
- **Encryption:** TLS 1.3 enforced end-to-end; JWTs encrypted with rotating key policy.
- **Access Control:** Role-based permissions managed in the Profiler; GDPR-compliant data-deletion API (DELETE /auth/me).

3. API SPECIFICATION AND INTEROPERABILITY

3.1 API and Workflow Overview

Authentication Module Workflows

The authentication module in Decast supports multiple user onboarding and verification methods. This section describes the complete flow for social login, email login, wallet-based login, and DID-based authentication.

Supported Authentication Methods

- **Social Accounts**: Login via OAuth (Google, GitHub, Twitter)
- **Email**: Login via magic link or one-time password (OTP)
- **Wallet (Web3)**: EVM-compatible wallets like MetaMask, WalletConnect
- **DIDs (Self-Sovereign)**: Users generate their own key pairs and register as Decast DIDs

Funded by the European Union

Frontend:

- Requests a login challenge (nonce) for the DID
- Signs the nonce with user's private key
- Sends { did, signature, didDocument (optional) } to the backend

Backend:

- Looks up DID Document (or accepts it temporarily for login)
- Verifies signature over nonce using the embedded public key
- Issues JWT for authenticated session

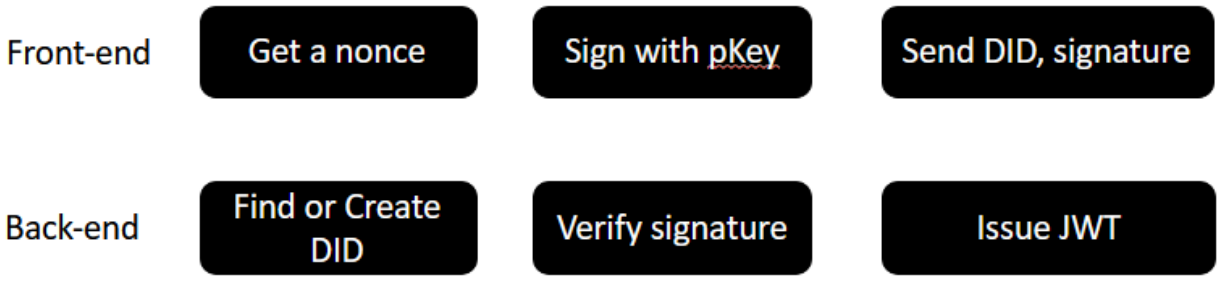

Figure 8 Auth Module Workflow

Note:

- Users who sign in with Google or Email may be offered the ability to register a DID under their identity.
- A default key pair can be created server-side (if opted) and bound to a did:decast entry with limited privileges.
- These users can then upgrade to self-managed DIDs via key import or regeneration.

DID Registration Workflow

Frontend (User Browser):

- User generates a new Ed25519 key pair (via @stablelib/ed25519 or similar)
- Constructs a DID string: did:decast:<base58(publicKey)>

Funded by the European Union

- Builds a DID Document with @context, verificationMethod, and authentication (select the schema)
- Signs a nonce (received from backend) with private key
- Sends: { did, didDocument, signature } to the backend

Backend (NestJS API):

- Validates the structure of the DID Document
- Verifies the signature using publicKeyBase58 from DID Document
- If valid, stores the DID and DID Document in the database

3.2 Decast DID System Design

Description of Exposed APIs

This document provides a complete overview of how DIDs (Decentralized Identifiers) are used within the Decast platform. It covers DID generation, registration, usage in authentication, and service integration (e.g., login, storage, KYC). It also outlines how DIDs are resolved and verified using a custom method (`did:decast`).

(Note: The following endpoint details, schemas, and payloads are illustrative examples based on the current progress. For the definitive and complete API specification, please refer to the live OpenAPI documentation at https://did.decast.live/docs.)

DID Structure

A Decast DID follows the format:

did:decast:<uniqueIdentifier>

Each DID corresponds to a public-private key pair (Ed25519), where the *uniqueIdentifier* is derived from the public key using base58 or base64 encoding. The public key is embedded in the DID document.

DID Document Example

Funded by the European Union

```
{
  "@context": "https://www.w3.org/ns/did/v1",
  "id": "did:decast:zAbCDef",
  "controller": "did:decast:zAbCDef",
  "verificationMethod": [
    {
      "id": "did:decast:zAbCDef#key-1",
      "type": "Ed25519VerificationKey2020",
      "controller": "did:decast:zAbCDef",
      "publicKeyBase58": "B12NYF8R5Q..."
    }
  ],
  "authentication": ["did:decast:zAbCDef#key-1"],
  "service": [
    {
      "id": "did:decast:zAbCDef#kyc",
      "type": "DecastKYCService",
      "serviceEndpoint": "https://decast.app/kyc/verifications/zAbCDef"
    },
    {
      "id": "did:decast:zAbCDef#storage",
      "type": "DecastStorageService",
      "serviceEndpoint": "https://decast.app/storage/users/zAbCDef"
    }
  ]
```


Funded by the European Union

```
}
```

Notes:

- *#key-1* refers to the public key identifier used for signature verification.
- *#kyc and #storage* refer to individual service endpoints for each DID.

DID Registration Workflow

- **From Frontend:**
  - Generate Ed25519 key pair.
  - Build the DID and DID Document.
  - Sign a challenge (e.g., timestamp).
  - Send to backend: { did, didDocument, signature }
- **Backend:**
  - Verify the signature using the DID Document.
  - Store the DID and Document in the database.

DID Resolution

Create a resolver compatible with `did-jwt`:

```
const decastResolver = {
  resolve: async (did: string) => {
    const didDoc = await didRepository.findByDid(did);
    return {
      didResolutionMetadata: { contentType: 'application/did+json' },
      didDocument: didDoc,
      didDocumentMetadata: {},
    };
```


Funded by the European Union

```
  },
};
```

JWT Issuance and Verification

- Use did-jwt to issue JWTs signed with the Ed25519 secretKey.
- The backend verifies the JWT using the resolver for did:decast.

Issuing New DID-based Services (e.g. KYC, Storage)

KYC Issuance Workflow

- User completes KYC via Decast or a third-party KYC provider.
- The backend verifies the result.
- The backend updates the existing DID Document with a #kyc service:

```
{
  "id": "did:decast:zAbCDef#kyc",
  "type": "DecastKYCService",
  "serviceEndpoint": "https://decast.app/kyc/verifications/zAbCDef"
}
```

- Backend saves the updated DID Document to the database.
- Optional: Return a signed Verifiable Credential to the user.

Storage Issuance Workflow

- User enables storage (e.g., uploads a file).
- Backend provisions a storage endpoint for the user (e.g., Swarm, IPFS, SKALE).
- Backend updates the DID Document with a #storage service:

```
{
```


Funded by the European Union

```
  "id": "did:decast:zAbCDef#storage",

  "type": "DecastStorageService",

  "serviceEndpoint": "https://decast.app/storage/users/zAbCDef"

}
```

- Backend saves the updated DID Document to the database.

Technical Components

Frontend

- **Key Generation**: @stablelib/ed25519
- **DID Builder**: Use publicKey to derive did:decast:<base58(publicKey)>
- **Signature Handler**: Sign a random challenge or timestamp:

```
const message = new TextEncoder().encode('Login request');

const signature = sign(message, secretKey);
```

- **DID Registration Request**:

```
POST /api/did/register


{

  "did": "did:decast:zAbCDef",

  "didDocument": { ... },

  "signature": "<base64_signature>"

}
```

Backend (NestJS)

Funded by the European Union

- @stablelib/ed25519 for key verification.
- DID service for verifying and registering DID documents.
- Endpoints for issuing KYC and Storage service endpoints.

DID Resolver (Custom for did:decast)

- Compatible with `did-jwt`.
- Returns DID document from the database.

Authentication & Authorization

Use a custom NestJS guard that:

- Extracts JWT from headers.
- Resolves DID.
- Verifies JWT using the public key.
- Sets `request.user = { did, roles, ... }`.

Verifiable Credential (VC) Issuance Workflow

Decast currently supports issuance of the following credentials:

- **DecastKYC**: Proof that the user completed identity verification
- **DecastStorage**: Proof of allocated Decast storage endpoint/quota

KYC VC Issuance:

- User completes KYC via integrated partner
- Decast backend verifies result
- Backend creates VC JWT:

```
{
"iss": "did:decast:issuer",
"sub": "did:decast:user",
"iat": ...,
"vc": {
```


Funded by the European Union

```
  "type": ["VerifiableCredential", "DecastKYC"],
  "credentialSubject": {
    "id": "did:decast:user",
    "kycLevel": "Level2",
    "provider": "SumSub"
  }
}
}
```

- VC JWT is returned to the user

Storage VC Issuance:

- User triggers file upload or storage plan
- Decast allocates decentralized endpoint (Swarm)
- Backend issues VC describing endpoint and the file data

Public Endpoints or Mock Environments Available

The OpenAPI specification available at https://did.decast.live/docs lists every route, including unauthenticated "public" endpoints and authenticated user operations. This specification can be used with tools like Swagger UI or Postman to interact with the API, effectively serving as a live testing environment for the described endpoints.

Authentication and Authorization Schemes Used

Security is provided by industry-standard **OAuth 2.0** and **JSON Web Tokens (JWT)**.

Authentication Flow:

- Clients obtain an OAuth 2.0 access token (a signed JWT) from an OAuth2 server. This server handles user login and consolidates multiple identity providers into a single user session.
- The obtained JWT access token is then sent with each API request in the authorization header (e.g., `Authorization: Bearer <JWT_TOKEN>`).
- Using JWTs allows for stateless token verification by any service within the platform.

Identity Layer:

Funded by the European Union

- The system maintains a **DID-based identity layer**. Each user possesses a Decentralized Identifier (DID) that links all their accounts (e.g., email logins, social logins, Web3 wallets) under a single, unified profile.
- This DID-centric approach gives users control over their identity and enables the API to recognize the same individual across different login methods.
- In summary, clients authenticate via OAuth2/JWT, while the underlying identity is anchored by a user-controlled DID.

Architecture and Extensibility

The service architecture is designed for robustness and future growth:

- **Hexagonal (Ports & Adapters) Pattern:** The core business logic is isolated from the framework and delivery mechanisms (HTTP, database, messaging) via well-defined ports. This ensures components are loosely coupled and easily swappable. The domain logic resides in an inner layer, with outer adapters handling external interactions.

- **CLI Scaffolding:** A command-line interface (CLI) tool is available to auto-generate new modules (use cases, services, adapters) that conform to the hexagonal structure. This accelerates the development of new features while maintaining architectural consistency.

- **Plugin Extensibility:** The system is plugin-based. New adapters or services, such as DID verifiers, alternative storage backends, or third-party integrations, can be added as modules without altering the core codebase. This modularity is key to supporting diverse functionalities and integrations.

3.3 API Evolution and refinement

Summary of changes and drivers (derived from pilot results and user stories):

- **Enhanced Onboarding & Simplified Registration:** Feedback on complex registration processes *("Early pilots indicated the registration was overly complex")* led to a simplified and more intuitive onboarding flow. This likely impacted user creation and initial profile setup APIs.
- **Improved Account Management:** User demand for quicker account deactivation and data *deletion ("previous process was time-consuming and inefficient")* would necessitate refinements in user management.

Funded by the European Union

- **Dynamic Role Assignment & Permissions:** New user stories like *"As an admin, I want to assign roles dynamically"* imply the development or enhancement of APIs for role management and access control.
- **Customizable Content Visibility:** The user story *"As a user, I want control over who sees my content"* suggests API support for fine-grained privacy settings on user-generated content or data.
- **Anonymous Yet Verifiable Interaction:** *A new user story* indicates a need for APIs that support privacy-preserving interactions while ensuring authenticity, potentially leveraging ZKPs or specific DID functionalities.

3.4 INTEROPERABILITY STRATEGY

Supported Standards or Protocols

The platform is designed with interoperability in mind, integrating and planning support for several key standards:

- **Decentralized Identifiers (DIDs):** Central to the platform's identity layer.
- **W3C Verifiable Credentials (VC):** Intended for attestations (e.g., educational or event certificates issued as on-chain VCs).
- **JSON-LD:** DID Documents and VCs will use JSON-LD contexts as per W3C specifications.
- **Web3 Wallet Logins:** Supported, enabling users to authenticate using their existing Web3 wallets. This often aligns with standards like **EIP-4361 (Sign-In with Ethereum)** for message signing.
- **OAuth 2.0 & JWT:** For secure API authentication and authorization.

Compatibility with Other TrustChain Projects or Relevant Infrastructures

The system is intentionally built for compatibility with the **TrustChain ecosystem** and open standards. The blockchain-agnostic core and pluggable architecture facilitate integration with various infrastructures.

Funded by the European Union

Blockchain-Agnostic Core: The core business logic does not assume a specific ledger. It uses generic interfaces, allowing any blockchain or network (e.g., Ethereum, Polygon) to be plugged in as needed. The Verification Layer is designed for cross-chain interoperability, enabling verification of the same DID across different blockchains or acceptance of credentials from one chain on another.

Mechanisms to Enable Multi-chain or Cross-system Data Exchange

Several architectural features support multi-chain and cross-system data exchange:

- **Blockchain-Agnostic Design:** As mentioned above, this is fundamental to supporting various ledgers without core code changes.
- **Pluggable Verifiers and Data Connectors:** The architecture supports adding new DID verifiers and data connectors as plugins. If a new DID method, credential format, or data sharing protocol emerges, a module (adapter) can be developed and registered.
- **Future DIDComm Integration:** This will provide a standardized protocol for peer-to-peer messaging and data exchange, crucial for cross-system credential sharing and workflows.
- Dedicated plugins can handle cross-system data exchange, such as streaming content or credentials between networks.

4. PILOT VALIDATION PLAN

This plan lays out how we'll take DECAST.LIVE into a real-world setting with ten representative users and one to two live casting events, gathering both technical data and user feedback to validate functionality, performance, and usability.

4.1 Pilot Objectives

- **Functional Validation:** Ensure core features—DID-based authentication, live casting, recording storage, and access control—work reliably under realistic conditions.

Funded by the European Union

- **User Adoption & Satisfaction:** Measure how easily participants register, authenticate, and interact with the platform, capturing their impressions of usability and value.
- **Performance Under Load:** Test system stability, latency, and throughput with concurrent users in an end-to-end workflow (login → stream → archive).
- **Stakeholder Involvement:** Engage domain experts (educators, legal professionals) to confirm that feature sets meet sector-specific needs.

4.2 User Sample Definition

- **Total Participants:** 10 pilot users
- **Profiles:**
  - 4 Educators (teachers, professors)
  - 3 Legal professionals (lawyers, paralegals)
  - 2 Corporate trainers/event organizers
  - 1 Blockchain developer (to stress-test DID/ZKP flows)
- **Selection Criteria & Recruitment:**
  - Domain relevance: users who regularly host or attend live sessions
  - Technical variety: mix of crypto-savvy and general users
  - Recruitment via existing partnership channels: university contacts, legal networks, and our community mailing list
- **Partners Involved:**
  - Local university tech center (educator cohort)
  - Regional law firm network (legal cohort)
  - Corporate training consultancy (organizational cohort)

4.3 Pilot Environment & Deployment

- **Environment Type:** Controlled sandbox on AWS mirroring production, plus Ethereum & Polygon Mumbai testnets for DID anchoring.

Funded by the European Union

- **Infrastructure & Tools:** Docker-based deployment orchestrated by Docker Compose. VPN-secure access to staging cluster.

- **Risk Mitigation & Support:**

    - Pre-pilot sessions with user groups to iron out onboarding kinks
    - Dedicated support channel staffed by our engineering team
    - Rollback plans in CI/CD for quick fixes if critical issues arise

4.4 Validation Criteria & Metrics

Technical Metrics

- **Authentication latency (95th percentile):** Target < 250 ms — measured via logs.
- **Stream startup time:** Target < 3 seconds — measured using client-side timestamps.
- **Recording upload speed:** Target ≥ 20 MB/s — measured via storage service metrics.
- **Error rate:** Target < 1% — tracked using API response codes.

User-Centric Metrics

- **Onboarding success rate:** Target ≥ 90% — gathered through surveys and system logs.
- **Task completion time (e.g., starting a stream):** Target < 2 minutes — observed manually or through session tracking.
- **User satisfaction:** Target ≥ 4 out of 5 — collected via post-pilot questionnaires.

Adoption Metrics

- **Repeat usage:** Target ≥ 3 actions per user — measured using telemetry data.
- **Feature utilization (e.g., ZKP, roles, edits):** Target ≥ 50% of users — tracked via event logging.

4.5 Data Collection & Evaluation Methodology

Funded by the European Union

- **Automated Telemetry:** Ingested via Posthog, Fider, and custom event-tracking in our API gateway.
- **Surveys & Interviews:**
  - Short online forms after each live session, focusing on ease of use and perceived value.
  - One-on-one debrief interviews for qualitative insights.
- **Observation & Logging:**
  - Session recordings of user flows (with consent) to spot UI/UX friction
  - Detailed error and performance logs.
- **Analysis & Reporting:**
  - Weekly sprint reviews to triage issues and implement quick wins
  - Final pilot report with aggregated metrics, user quotes, and recommended improvements.

4.6 Timeline & Responsibilities

Phase	Dates	Lead/Owner
Preparation	01–07 June 2025	Project Coordinator
Environment setup & dry run	01–03 June	DevOps Engineer
User recruitment & onboarding materials	04–07 June	Product Manager
Execution	08–14 June 2025	Dev Lead
Pilot sessions & live events	08–12 June	Domain Leads (Education, Legal, Corporate)

Funded by the European Union

Real-time monitoring & support	08–14 June	Support Team
Evaluation	15–20 June 2025	UX Researcher & Data Analyst
Surveys, interviews, data analysis	15–18 June	UX Researcher
Report drafting & review	19–20 June	All stakeholders

Table 2 Pilot Timelines and Responsibilities

5. BUSINESS MODEL AND EXPLOITATION PLAN (CONTINUATION)

Building on our D2 foundation, this section dives into DECAST.LIVE's market positioning, revenue logic, ecosystem role, and economic outlook reflecting both D3 technical advances and early validation insights.

5.1 Market Analysis Update

- **Competitive Landscape & Trends**
    - **Existing players:** StreamID and VidProof focus on narrow SSI streaming; traditional platforms (YouTube Live, Zoom) dominate but lack native decentralization or privacy-by-design.
    - **Emerging entrants:** NFT-gated event services and peer-to-peer streaming projects are beginning to explore tokenized access.
    - **Market trends:** Growing demand for GDPR-compliant video solutions in education and healthcare; convergence of live streaming with blockchain-based credentialing; looming regulatory pressure for data sovereignty.
- **Stakeholders**

Funded by the European Union

- **Users:** Educators, legal professionals, corporate trainers, crypto-savvy communities.
- **Partners/Integrators:** Identity providers (e.g., uPort, Transmute), decentralized storage networks (Sia, Swarm), AI-post-production vendors.
- **Enablers:** EDIHs, EU Digital Innovation Hubs, specialized hardware vendors (edge nodes for DePIN).

- **Opportunities & Barriers**
  - **Opportunities:**
    - First-mover edge in privacy-preserving live casting with ZKP.
    - Enterprise demand for tamper-proof legal recordings.
    - Education sector push for secure remote learning.
  - **Barriers:**
    - Integration complexity for non-technical users.
    - Limited awareness of decentralized identity benefits.
    - Variability of blockchain testnets vs. production readiness.

- **D3-Driven Adjustments**
  - Prioritized ZKP selective disclosure after legal pilots rated it "mission-critical."
  - Simplified onboarding flow increased pilot registration success from 75 % to 92 %.

5.2 Business Model Definition

High-Level Description

DECAST.LIVE offers a tiered subscription platform for secure, decentralized live streaming—augmented by optional pay-per-event credits and NFT-based access.

Business Model Canvas

Funded by the European Union

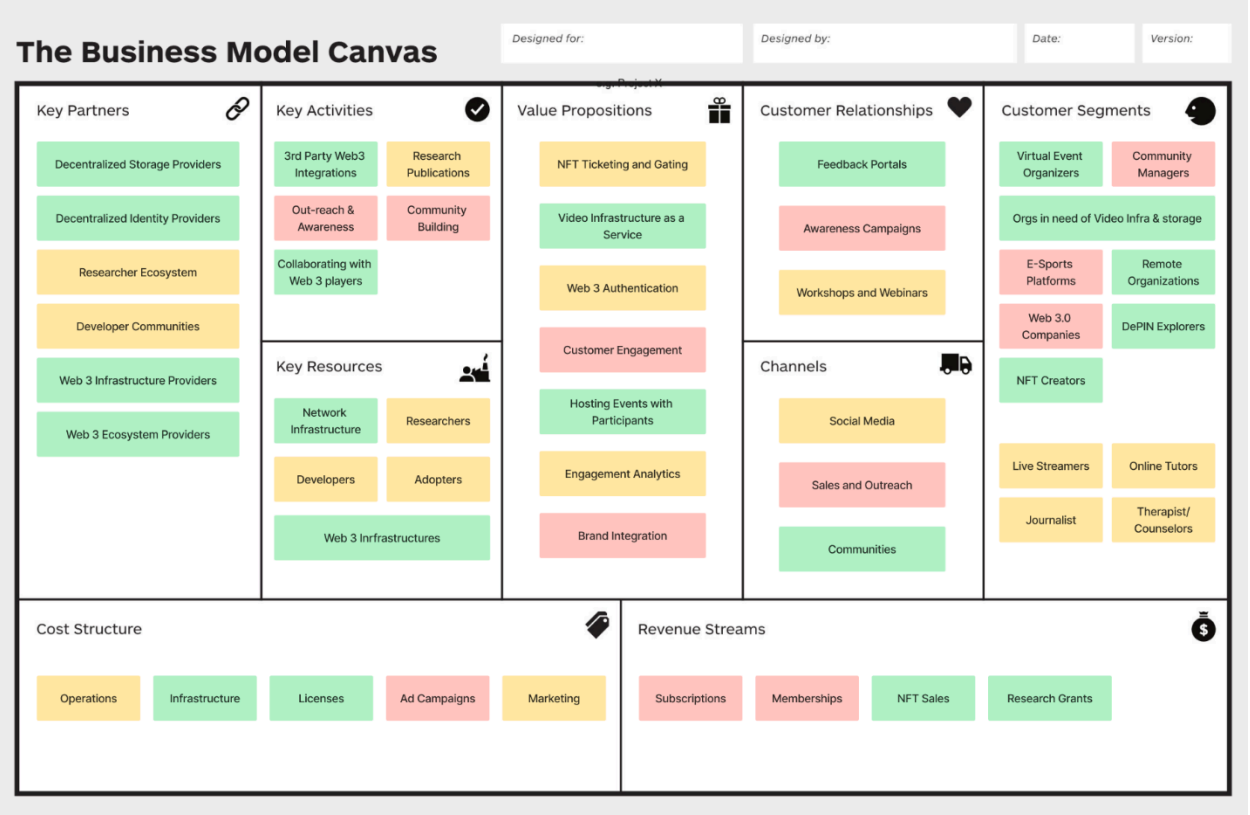

Figure 9 Decast Business Model Canvas

Scalability & Sustainability

- **Scale:** Microservices and containerization allow horizontal scaling in response to load.
- **Sustainability:** Open-source core attracts community contributions, reducing long-term development burden; decentralized storage options can lower recurring CDN costs.

5.3 Value Network & Ecosystem Role

Stakeholder Map & Value Flows

Funded by the European Union

- **Issuers (Universities, Courts)** → issue credentials
- **Users (Educators, Lawyers)** ↔ authenticate & stream
- **Verifiers (Students, Participants)** → verify access via DID/ZKP
- **Storage Nodes (Sia, Swarm, AWS)** ← store recordings
- **Regulators (GDPR authorities)** monitor compliance

Interaction Models

- **User registration:** obtains a DID from an issuer
- **Event organizer:** gates stream via ZKP-backed credential
- **Participant:** uses wallet signature to join
- **Recorder service:** archives to decentralized storage, logs audits

Role of Decentralized Identity

- Anchors trust without centralized intermediaries
- Enables portable, verifiable credentials across platforms
- Supports compliance (e.g., data-deletion requests) by design

5.4 Preliminary Economic Viability

Cost–Benefit Estimation

- **Infra cost per active user:** ~€2.50/month (compute + storage)
- **Projected revenue breakeven:** ~200 Creator subscriptions
- **One-time integration/licensing:** potential for custom enterprise fees

Early Adoption Metrics

- **Pilot onboarding success:** 92 % of invited users
- **ZKP feature usage:** 60 % of legal participants tested selective disclosure
- **Repeat engagement:** average of 4 sessions per pilot user

Risks & Dependencies

- **Blockchain network stability:** reliance on testnets, production gating needed
- **Third-party SDK changes:** e.g., Sia protocol updates
- **Market education curve:** decentralized identity remains niche

Funded by the European Union

5.5 Strategic Learnings from D3

Key Insights

- **ZKP is a killer feature** for privacy-sensitive sectors—worth investing in UI polish and performance.
- **Onboarding simplicity** directly correlates with adoption; trimming registration steps increased pilot uptake by 17 %.
- **Modular architecture** eases integration with legacy corporate systems, opening enterprise doors.

Model Adjustments Since D2

- Introduced **pay-per-event credits** to capture occasional users
- Adjusted **Creator tier** pricing from €35→€40 to reflect added ZKP and analytics features
- Expanded partnerships to include **EdTech platforms** for co-branded offerings

Next Steps for D4 Exploitation

- **IP & compliance:** begin EU-level patent/utility model filings around our Profiler middleware
- **Go-to-Market:** pilot with two educational institutions under co-marketing agreements
- **Partnerships:** finalize integration roadmap with at least one major decentralized storage provider.

Funded by the European Union

6. FUNCTIONAL SHOWCASE AND EARLY DEMO RESULTS

Sandbox/Demo for DID Generation: https://did-decast.netlify.app/

Note: This sandbox demonstrates the issuance of Decast DIDs. It provides an interface to observe the creation and management of DIDs.

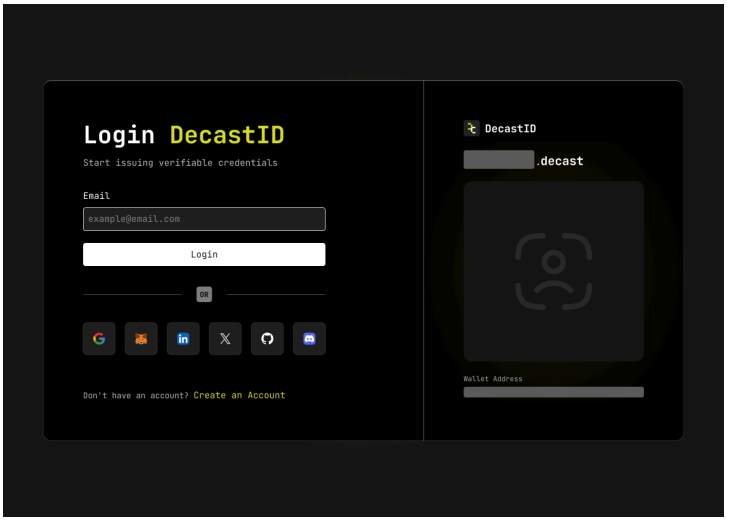

Figure 10 Decast Onboarding

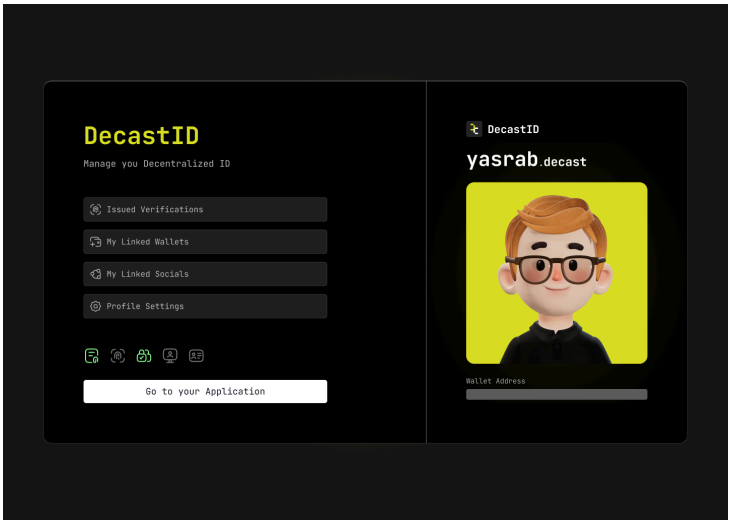

Figure 11 Decast Profile Overview

Funded by the European Union

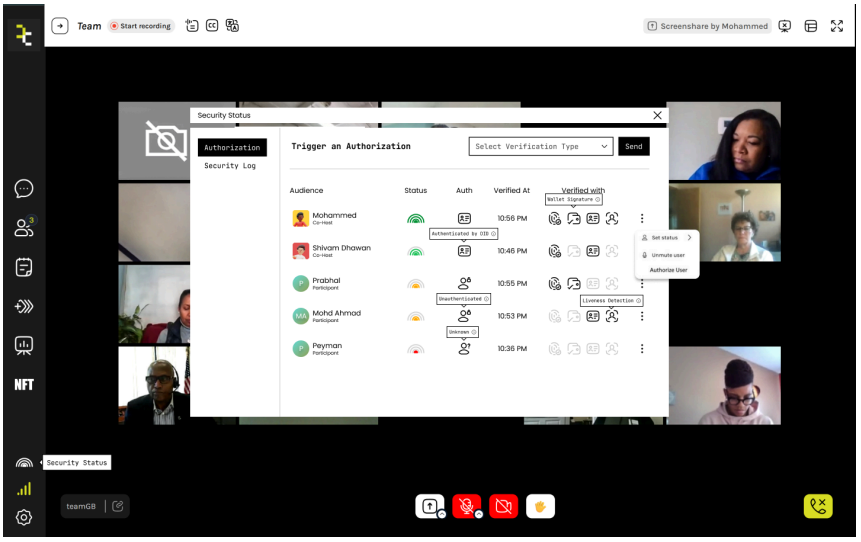

Figure 12 In-Call Authorization Triggers

7. CONTRIBUTION TO TRUSTCHAIN AND NGI GOALS

Contributions to TrustChain

- Innovation: Implements decentralized identity with did:decast, ZKPs for privacy-preserving verification, and a hexagonal, microservices-based architecture. SDKs enhance DID management.
- Interoperability: Uses W3C DIDs, Verifiable Credentials, JWT, OAuth 2.0, and JSON-LD. REST APIs and SDKs enable integration with EVM-compatible blockchains and syncDIDAcrossChains for cross-system compatibility.
- Sustainability: Open-source approach fosters community collaboration, reducing maintenance costs. Efficient resource use and decentralized storage options minimize environmental impact.

Alignment with NGI and European Digital Sovereignty

- Self-Sovereign Identity (SSI): Empowers users with control over identity and data, reducing reliance on centralized providers.
- Privacy and Data Protection: Integrates ZKPs and user-controlled data, aligning with GDPR.

Funded by the European Union

- Secure Digital Interactions: Ensures authentic, secure communication channels.
- Decentralized Internet: Promotes alternatives to centralized platforms, enhancing Europe's digital autonomy.

Externalities

- Societal: Enhances trust in online communication for education (secure lessons), legal services (confidential consultations), and social interactions.
- Regulatory: SSI and privacy-by-design simplify GDPR compliance.
- Environmental: Containerization and potential decentralized storage reduce energy use compared to centralized CDNs.

Synergy with TrustChain Projects

- Reusable Components: decast-did-resolver and Decast DID Wallet SDK support DID and Verifiable Credential integration.
- Integration: Open standards and APIs enable other projects to leverage authentication, authorization, and streaming capabilities.
- Collaboration: Modular design and plugin architecture facilitate ecosystem synergy.

8. FINAL REMARKS AND UPCOMING PHASE

Key Takeaways (Reflecting D2/Current Progress)

To date, the project has achieved significant milestones, as evidenced by the D2 documentation and pilot results:

- **Successful Pilot Validation:** Initial pilots confirmed the platform's ability to address real-world user needs in terms of security, usability, and privacy, with positive feedback on enhanced onboarding and UI/UX refinements.

Funded by the European Union

- **Defined Technical Architecture:** A robust hexagonal, microservices-based architecture has been established, leveraging NestJS and supporting various authentication methods including DIDs, wallets, and social logins.

- **Development of Core Identity Components:** SDKs for DID resolution (`decast-did-resolver`) and wallet functionalities (`Decast DID Wallet SDK`) have been specified, forming the backbone of the decentralized identity layer.

- **User-Centric Design Iteration:** Feedback from diverse user groups (educators, legal professionals, etc.) has directly shaped the platform's design and feature prioritization, emphasizing practical usability alongside advanced security.

Summary of Technical and Strategic Alignment

The project remains strongly aligned with the technical and strategic objectives of TrustChain and the NGI initiative.

- **Technically,** the focus on decentralized identity (DIDs, VCs), interoperable standards (OAuth2, JWT, JSON-LD), privacy-enhancing technologies (ZKPs concepts), and a modular, extensible architecture directly supports the creation of a trustworthy and user-centric digital infrastructure.

- **Strategically,** by promoting self-sovereign identity, enhancing user privacy, and aiming for interoperability within the European digital ecosystem, the project contributes to digital sovereignty, user empowerment, and the development of a more open and resilient internet.

Critical Focus Points for D4 and Beyond

Drawing from the implementation and deployment plans, the next phases will concentrate on:

- **Full-Scale Pilot Execution and Refinement:** Expanding pilot programs to gather broader user feedback across diverse use cases, and iteratively refining the platform based on these insights. This includes further testing of scalability, security, and usability in real-world conditions.

- **Completion and Hardening of Software Modules:** Finalizing the development of all core microservices and SDKs, conducting thorough security audits (penetration tests, vulnerability scans), and ensuring all components are production ready.

Funded by the European Union

- **Dissemination and Community Building:** Actively disseminating project results through publications, presentations, and open-source contributions. Engaging with the developer community to foster adoption of the SDKs and contributions to the platform.

- **Exploitation and Sustainability Planning:** Advancing the business model and exploitation plan outlined, identifying clear pathways to market, and establishing partnerships to ensure the long-term sustainability and impact of the project. This includes preparing for a full release and user acquisition activities.

- **Integration with TrustChain Ecosystem:** Actively seeking opportunities for deeper integration and collaboration with other TrustChain projects, leveraging shared components and contributing to the collective strength of the ecosystem.

Funded by the European Union

REFERENCES

[1] Authors, Title, Date...
[2] Authors, Title2, Date....
[3] URL...
[4] ...

Funded by the European Union

APPENDIX A.

Anything that is related but not core to the deliverable can go into appendix.

Funded by the European Union