Modulr Whitepaper

Welcome to UndChain!

We're thrilled to have you here! UndChain is building the next generation of decentralized cloud services, and we'd love for you to join us in shaping the future.

🛠️ Developer Discussions & Community Support

If you're a developer interested in contributing to UndChain, have questions, or simply want to join the discussion with like-minded enthusiasts, we invite you to connect with our community:

- **Discord**:

We will be releasing a discord so that developers can easily communicate and help with the project. Keep in mind that all contributions are rewarded. i.e. you can earn on network tokens. This will be automated in the future with Code Ledger.

What is UndChain?

UndChain is a PoU (Proof of Utility) blockchain; that can best be thought of as a decentralized cloud service in which anyone can participate. There are three different utilities that can be performed on the network,

that work together to create similar functionality as conventional cloud service providers.

1. **Computation** - The thought behind this function would be items that either are too complex for a ordinary machine to complete in a reasonable amount of time (training AI models, 3D rendering or DNA sequencing) or something that requires modifying data that must be shared with a group (think servers).
2. **Storage** - In order to compute you must have access to data to compute on. UndChain has storage built into it's language so you don't have to use another service. This allows for things such as databases and the ability to store large AI models.
3. **Access** - This is a functionality that will be expanded more in the future, but for now think of this as the ability to have access to specialty equipment in a specific location such as linking two or more users together, surrogates or VPN like services. When fully realized it is meant to replace the IP protocol by enable secure communications that are digital signed from the source.

When mixing these three utility types this will enable a full cloud experience that can do things such as spin up gaming servers, webpages and virtual computers. What makes this different than the way the internet is ran today is that anyone can participate and earn rewards for providing utility to the network. This reduces the need for massive data centers and spreads out computational resources throughout a geographic area. This will **NOT** eliminate centralized services, in fact there will be ways (via access) in which you can still direct to a centralized service. The goal of the network is to allow both the centralized and the decentralized systems to co-exist and allow the consumer to choose.

Co-Chains

Co-Chains is how the network expands it's abilities and services. Anyone can create and own a co-chain, in fact there is a classification of user called **Creators** mentioned later. The goal of a co-chain is to expand the functionality of the network and not to make new coins, as such it is up to the Creator if they wish to create a new token or use the blockchain's native utility token (USP). The Creator can still set their fees even with the native chains token just as they can with their own. The difference is you cannot control the emission rate nor the quantity. The goal in doing this is to reduce the amount of tokens that go in circulation. It also simplifies the usability of the blockchain the less swaps that need to be performed at the action house.

Why Co-Chains?

The purpose of utilizing co-chains rather than having traditional layer 2 blockchains are multifaceted:

1. It allows for better security since at each block completion a co-chain blends it's hash with a co-chain it's linked to, this reduces the risk of a attack on the network.
2. It expands Pseudo (UndChain's Pythonic programing language), by allowing others to import functions from other chains. You can choose to private a function or set fees on it. Remember you own it...
3. It provides a sense of ownership for your work and allows you to monetize on it so you can continue to make it better and help you with your personal goals. You can also transfer ownership of a co-chain to another user on the network.
4. Personally it's better than saying layer two; you're a creator on this network and providing value, you should be recognized for it. Even

the main chain is a co-chain. *Note: There will be some verbiage changes in here that I believe are more accurate and beneficial to the industry.*

Understanding Co-Chains: Building Blocks of UndChain

Co-chains are specialized blockchain applications built on UndChain's decentralized protocol. They enable developers to create targeted solutions, each tailored to specific use cases, while still integrating seamlessly with the broader ecosystem. To illustrate their potential, I've designed several foundational co-chains. These not only add value to the network but also serve as practical examples for developers aiming to build on UndChain.

Why I'm Building These Co-Chains

As the creator of UndChain, my goal is to demonstrate how co-chains can unlock innovative solutions. By developing these foundational co-chains, I aim to:

1. **Showcase Efficient Data Management:**
   Demonstrate how to use storage, computation and access, to interact with the network in a secure and efficient manner.
2. **Promote Token Standardization:**
   Provide examples of token creation while encouraging developers to use existing tokens like USP to reduce market fragmentation and maintain interoperability.
3. **Highlight Co-Chain Collaboration:**
   Illustrate how co-chains can integrate and expand upon one another

to create complex, interdependent systems (e.g., using Live with Smack or Player2).

4. **Inspire Developer Creativity:**
   Present co-chains that provide tangible value to users, showcasing how developers can design systems that drive engagement, adoption, and utility.
5. **Encourage Best Practices:**
   Serve as a blueprint for building secure, scalable, and sustainable blockchain applications that align with user needs and network goals.

A Platform for Innovation

These co-chains are not just standalone solutions; they are stepping stones for developers. By integrating and iterating on the concepts presented, developers can reduce duplication, enhance functionality, and focus on creating new value for the ecosystem. Whether it's a decentralized streaming platform, a social media protocol, or a robust marketplace, the possibilities are endless when co-chains work together.

Main Chain

The Main Chain serves as the foundation of UndChain, coordinating critical protocols, utilities, and governance across the network. Its responsibilities include:

1. **Messaging and Communication**:
   - Establishes base peer to peer messaging protocols to facilitate interaction between user types (validators, partners, and clients).
   - Introduces an **emergency messaging system** that allows Creators to broadcast critical updates and alert messages to all users on the network.

2. **Token Management**:
    - Creates two native tokens:
        - **UGP (UndChain Gold Piece)**: A connection token enabling interactions with validators.
        - **USP (UndChain Silver Piece)**: A utility token for partners to perform tasks across co-chains.
            - This is the token that can be overridden with a require collateral which will be paid out to users in the event that liquidity is removed from the token in a short time frame (UndChain's anti-rug protocol). *When we create AdCoin I will show how this process will work.*
3. **Alias System (UnaS)**:
    - Provides users with customizable usernames (e.g., `@username`) for simplified transactions and easier identification.
    - This system is case insensitive so users won't have to worry if any potion is capitalized or not.
4. **Chain Rules Framework**:
    - Sets the format for defining co-chain-specific rules, including:
        - Developer compensation.
        - Tokenomics structure.
        - Validator preferences.
5. **Interoperability and Subnet Management**:
    - Stores contact information for co-chains, ensuring seamless redirection to updated versions.
    - Maintains cross-network links for interoperability across multiple networks and subnets.
6. **Digital Asset Management**:
    - Enables **digital asset assignment**, allowing one or more users to co-own assets and share fees (e.g., between a songwriter and

a musician).
    - Supports user groups to collectively manage assets.
    - Introduces **Open Transactions**, keeping asset transactions active for ongoing exchanges (e.g., tips or recurring contributions).
    - Implements **Standard Digital Asset Contracts**, a predefined system for defining asset rights and restrictions.
    - Introduces the **Digital Asset Protection (DAP) Protocol**, allowing assets to be protected and claims to be enforced both on and off-chain.
7. **Security Protocols**:
    - **Will Protocol**: Allows users to predefine where their digital assets should go if their account becomes inactive for a user-defined period.
    - **Freeze Protocol**: Locks an account to prevent any transactions if a breach is suspected, activating the Will Protocol if necessary.
    - **Limiter Protocol**: Sets transaction limits or delay timers for assets, minimizing losses during potential account compromise.
8. **Licenses and Contracts**:
    - Provides a framework for creating:
        - Digital Asset Contracts (e.g., usage rights and restrictions).
        - Terms of Use for individual co-chains.
9. **Subscription Model**:
    - Implements a subscription system, enabling users to pay recurring fees for co-chain services with predefined expiration dates.
10. **Redemption System**:
    - Allows users to generate unique codes that others can redeem for assets, similar to prepaid or gift cards.

11. **Reliability Score**:
    - Tracks and evaluates user behavior to identify trustworthy participants and mitigate bad actors.
12. **Activity Metrics and Ratings**:
    - Maintains standardized metrics for co-chain activity and performance.
    - Implements a rating system for users to assess co-chains' reliability.
13. **Convergence Protocol**:
    - Prevents blockchain size from becoming unmanageable by consolidating historical data into a new genesis block while preserving data integrity.
14. **Block Time and Synchronization**:
    - Establishes and maintains consistent block time across the network to ensure synchronization.
15. **Address book other co-chains:**
    - Stores the validator addresses (contact information) for other co-chains so that if a user doesn't have that information stored in their system, they can request it.

Mimic

Mimic is UndChain's dedicated AI co-chain, designed to revolutionize how artificial intelligence is deployed and used across decentralized networks. The Mimic co-chain focuses on creating **specialized AI models** tailored for specific tasks, ensuring efficiency, modularity, and adaptability. These models are connected to a **hypervisor system**,

which intelligently routes incoming requests to the appropriate AI for seamless performance.

Key Features:

1. **Specialized AI Models**
   - Mimic avoids the pitfalls of monolithic AI systems by focusing on **smaller, task-specific models**. Each model excels at a particular function, making them easier to maintain, scale, and adapt to hardware limitations.
   - This modular approach also enables partners to support specific models based on their available resources, ensuring that Mimic remains inclusive and scalable.
2. **Hypervisor for Smart Routing**
   - A hypervisor serves as the backbone of Mimic, dynamically directing user requests to the AI model best suited for the task. Whether it's language translation, image recognition, or data analysis, Mimic ensures every request is handled optimally.
3. **M3L and GSS Integration**
   - Mimic seamlessly integrates with M3L and GSS, enabling dynamic and user-friendly AI-driven experiences across the network. For example:
     - Applications built with M3L/GSS can include **context-aware assistants** that help users navigate features and troubleshoot issues.
     - Custom AIs can be pre-loaded with documentation, FAQs, and specific app functionality, making onboarding and support intuitive and accessible.
4. **User-Customizable AIs**

- Users can train and deploy their own models on Mimic, transforming them into digital assets. These models can be monetized, shared, or enhanced by others in the community, enabling a decentralized marketplace for AI innovation.

5. **Hardware Adaptability**
   - Mimic's design ensures that models can run on a wide range of hardware, from high-performance GPUs, NPUs, FPGAs and to more modest devices. This makes AI capabilities accessible to a broader audience and avoids centralization on specialized infrastructure.
6. **Privacy-Focused AI**
   - Mimic prioritizes user privacy by ensuring that all data used for training or inference remains secure and decentralized. Users retain control over their inputs, and models operate within strict privacy constraints.
7. **Dynamic Upgradability**
   - By focusing on specialized models, Mimic enables faster updates and improvements. Developers can modify or replace individual models without impacting the entire system, ensuring agility and resilience.

Use Cases and Potential:

- **Personal Assistants for Apps**
  Mimic AIs embedded in M3L/GSS applications can guide users through features, troubleshoot issues, and provide a tailored user experience. For example, a gaming co-chain could include an AI that helps players strategize, while a business app might offer an AI-driven project planner.

- **Decentralized AI Marketplace**
  Mimic enables developers to monetize their models, creating an ecosystem where innovation thrives. Users can purchase or license custom AIs, opening new opportunities for collaboration and revenue generation.
- **AI-Powered Content Creation**
  From generating blog posts to designing graphics, Mimic models can assist creators across the network, integrating seamlessly with applications like Pages and Smack.
- **Accessible AI for Everyone**
  Mimic's modular, hardware-agnostic approach ensures that AI is not limited to those with cutting-edge systems. Partners with modest hardware can still contribute, making AI-driven services truly decentralized.

In Summary

Mimic redefines AI for decentralized networks, combining modularity, scalability, and user empowerment. It's not just an AI service—it's a foundation for innovation, creativity, and community-driven progress on UndChain.

Pages (Decentralized Web Platform)

Pages is UndChain's decentralized Web4 system, offering a modern alternative to traditional web technologies. Replacing HTML and CSS with **M3L** (Multi-Media Markup Language) and **GSS** (Global Style Sheets), Pages provides a scalable, efficient, and privacy-centric experience for both creators and users.

Legacy Domain Protection

UndChain respects legacy (HTTP) domain names ending in **.com**, **.org**, **.gov**, **.edu**, and **.net**. These domains are reserved exclusively for their rightful owners as registered on the traditional internet to prevent impersonation and fraud. This ensures that trusted brands and institutions retain their identity in the transition to decentralized web technologies.

All other domain types (e.g., **.io**, **.xyz**) are not automatically recognized or reserved within the UndChain ecosystem. This boundary is established to maintain clarity and fairness while minimizing complexity.

Key Features:

1. **Efficient Content Delivery**
   - **M3L** and **GSS** revolutionize how content is served by allowing updates to individual components rather than entire documents, significantly reducing network load.
   - Dynamic scaling ensures that popular pages are hosted across multiple partners based on demand and region, maintaining uptime and low latency even during high traffic.
   - To learn more about M3L and GSS: M3L and GSS Documentation
2. **Advanced Content Types**
   - Unlike HTML, **M3L** supports more advanced content types, such as **video**, **chat**, and **canvas**, enabling developers to build richer, more interactive applications directly on the Pages co-chain.
   - M3L is designed for modularity and efficiency, referencing content locations dynamically rather than embedding them, reducing redundancy and improving scalability.

3. **Tailored User Experience**
    - Pages includes a **featured content section** personalized to each user's interests. This section is fully editable, giving users control over the recommendations they see.
    - Content is also **classified and rated** to ensure users know what to expect from a site. Ratings are especially useful for identifying mature content or categorizing pages by their intended audience.
4. **Page Rating System**
    - Users can **rate pages**, helping to maintain honesty and accountability. If creators falsely claim their pages are safer than they are, they receive a penalty to their Reliability score. Similarly, users who submit false negative claims about a page's rating will also experience a decrease in their Reliability score.
    - Only ratings from users with **reputable Reliability scores** are taken seriously, ensuring the system rewards fairness and transparency.
5. **Privacy by Design**
    - By eliminating cookies and intrusive tracking methods, Pages protects user privacy while still providing a seamless browsing experience.
    - The platform leverages decentralized systems to ensure user data remains secure and under their control.
6. **AdCoin: The Native Token**
    - AdCoin is the native utility token for Pages, enabling a frictionless user experience. Users can choose to earn AdCoin by interacting with ads, assisting in **Mimic AI model training** (e.g., teaching AI how to draw or solve logical problems), or swap with other in network tokens to support an ad-free browsing experience.

- AdCoin integration across the network ensures seamless monetization options for creators and users alike.

7. **Age Verification and Content Control**
   - The **Adult_Swim Protocol** introduces a decentralized age verification system. Instead of storing user data, it uses markers to verify whether an account holder meets age requirements.
   - Verified entities can grant age status, but any organization providing false records loses its ability to verify accounts, and all associated accounts lose their status. The UndChain organization can act as the default age verifier but primarily delegates this function to trusted entities.
8. **Seamless Integration with Web4**
   - Pages works seamlessly with other UndChain co-chains, leveraging **M3L** and **GSS** to enable powerful tools like Mimic-powered assistants. These assistants can guide users through the network, troubleshoot issues, and provide real-time help for specific features in the apps they support.
9. **Content Reaction System**
   - Pages includes built-in tools for users to react to and interact with content. This feedback helps site owners gauge audience sentiment and improve their offerings. For example, news articles and videos can be rated positively or negatively, providing valuable insights.
10. **Integrated A / B Testing**
    - Pages provides users the ability to do A / B testing, allowing developers the ability to run two different sites to see which gets the most interaction.

LINK (Localized Interface for Network Kinetics)

LINK (Localized Interface for Networked Kinetics) is UndChain's **standardized protocol for remote robotics**, designed to make piloting physical systems as seamless as accessing an app. Whether operating a humanoid surrogate, flying a drone, or steering an autonomous mower, LINK provides the ultra-responsive foundation for decentralized embodiment.

By creating a shared communication and control standard, LINK ensures secure, modular, and low-latency access to machines within a user's region, while enabling global coordination and oversight through AI-assisted supervision across UndChain subdomains.

Key Features:

1. **Unified Piloting Protocol**
   - LINK serves as a **hardware-agnostic communication layer** that can be implemented by any robot, drone, or control station. It standardizes how movement, vision, and feedback are encoded, ensuring interoperability across devices and vendors.
   - Devices that support LINK can be driven manually, supervised by AI, or operated collaboratively—**enabling both real-time embodiment and semi-autonomous delegation.**
2. **Ultra-Low Latency Pipeline**
   - LINK is engineered to support **sub-12ms motion-to-action latency**, making it suitable for everything from tactile robotics to competitive drone piloting.
   - Its architecture includes:

    - 360° visual systems for head-tracked VR environments
    - UDP-based data channels for rapid control & feedback
    - Timestamped motion frames for consistent state prediction
3. **Augmented Reality (AR) Guidance**
    - In **AR Mode**, LINK overlays spatial highlights to visualize interactable objects and predicted AI behavior.
    - This is especially powerful in **Supervisor Mode**, where a user can observe the robot's intended actions and intervene before errors occur—turning the user into a proactive guide rather than a passive viewer.
4. **Phantom Mode (Simulated Embodiment Layer)**
    - For environments with limited bandwidth or unpredictable input, LINK supports a **simulated piloting mode** where the world is mirrored into a lightweight, game-like 3D space.
    - Instead of streaming high-bandwidth video, the user interacts with **preloaded 3D assets and sensor inputs**—dramatically reducing network load while enhancing control speed.
    - This mode also enables **gamified task execution**, useful for training, coordination, or user-friendly job interfaces.
5. **Access by Design**
    - LINK was built to **embody UndChain's "Access" layer**: the ability to control, touch, and affect the real world through secure, decentralized systems.
    - From robots in homes and farms to machines in factories and cities, LINK ensures users can connect, control, and coordinate without centralized gateways or proprietary lock-in.
6. **Mimic Synergy**
    - LINK integrates tightly with the Mimic co-chain for **supervised automation and AI-directed operation**.

- Mimic will:
    - Handle low-level motion planning
    - Auto-suggest next actions
    - Monitor operator behavior for safety and efficiency
    - Swap users between “Full Pilot” and “Supervisor” roles on the fly

7. **Future-Proof Modularity**
    - LINK is designed for easy extension into:
        - **Fleet Management** for enterprise-scale deployments
        - **AI-Driven Swarms** for coordinated task execution
        - **Permissioned Control Leasing**, where users rent LINK-compatible bots across networks
        - **On-chain action verification** via zk-motion logs and validator-signed events

---

Use Cases and Potential:

- **Remote Farming & Equipment Control**
  Use LINK-enabled rigs to remotely operate tractors, pickers, and autonomous harvesters—with AR assistance and fallback Phantom Mode for unreliable network zones.
- **Surrogate Robotics for Home & Care**
  Allow users to step into LINK-compatible home bots to check on loved ones, assist with tasks, or provide companionship—controlled securely and respectfully.
- **Logistics & Infrastructure Maintenance**
  LINK offers a standardized framework for inspecting, repairing, or

managing distributed systems such as pipelines, bridges, or transport hubs.

- **Security & Surveillance**
  Coordinate robotic patrols or autonomous drones with override-ready Supervisor Mode, reducing risk while maintaining accountability.
- **Immersive Control & Gamified Workflows**
  Train operators through Phantom Mode. Gamify real-world tasks. Reduce fatigue by turning repetitive operations into engaging digital interactions.

In Summary

LINK makes it possible to **pilot and coordinate physical machines** through a unified, decentralized control layer. Within a user's local region, LINK offers **real-time, low-latency embodiment**—while globally, users can guide AI-driven systems through **Supervisor Mode** or interact via a lightweight simulation in **Phantom Mode**.

It's more than a protocol—it's how **Access becomes physical** on UndChain.

Auction House

The Auction House co-chain is UndChain's decentralized marketplace for minting, trading, and protecting digital assets on the network. It uses **USP** (UndChain Silver Piece) as the default transaction token. Users can seamlessly swap other registered tokens like AdCoin for USP at checkout.

Key Features

1. **Digital Asset Contracts**:
    - Assets are minted using UndChain's **standard Digital Asset Contracts** as defined by the main chain. Those assets are as follows:
        - **Gray (Unregistered)**: Default for unprotected assets.
        - **Green (Rental)**: Allows limited-use rentals with royalties sent to the creator.
        - **Blue (Single Use + Sharing)**: Enables personal use and sharing, with a kickback for creators when shared content generates value.
        - **Purple (Ownership)**: Allows reselling and distribution but restricts modifications.
        - **Gold (Full Use)**: Complete ownership, including rights to resell, modify, and monetize.
    - Certain categories of assets, like **music, videos, or games**, automatically integrate **royalty payments** for creators.
    - Asset metadata includes ratings, version history, and licensing terms to help buyers make informed decisions.
2. **Fraud Detection and Claims**:
    - The Auction House connects users to **Digital Asset Protection Providers (DAPPs)**, who handle protection, disputes, and enforcement actions.
    - Filing a claim requires a **fee**, with the amount determined by the **Reliability score** of the claimant. Users with lower scores pay higher fees to discourage abuse.
    - Winning claims result in asset ownership being transferred to the rightful owner, along with the claim fee. False claims reduce the claimant's Reliability score and reward the accused with the fee.

- **RWA Protection**: Digital assets proven to infringe on real-world copyrights (RWAs) are transferred to rightful owners or destroyed.

3. **Trust and Transparency**:
   - Sellers are classified visually on the Auction House based on their **Reliability score**:
     - High-trust sellers are marked with a **verified badge** or highlighted visually for buyers.
     - Low-trust sellers have clear **warnings or indicators** near their listings.
   - Listings display whether the seller is the **original creator** or a **reseller** of the asset, helping buyers identify primary sources.
   - All transactions, bids, and claims are stored **on-chain** for full transparency.
4. **Resale Royalties and Sharing Incentives**:
   - **Green and Blue Contracts** integrate royalty systems:
     - **Green Contracts**: Suitable for rentals (e.g., short-term music, video, or asset licensing). Creators receive royalties on repeated rentals.
     - **Blue Contracts**: Allow creators to benefit from content sharing (e.g., reaction videos or embedded content), receiving a share of generated revenue.
   - Creators can automate royalty payments through smart enforcement using DAPPs.
5. **Fraud Prevention and AI Support**:
   - The Auction House leverages **Mimic AI** to scan for duplicates, unauthorized content, or stolen RWAs. This is used as a warning system to the creator to let them know they *MAY* have created a

copyright work. The creator will then respond to that as to why it's not which can be used if the asset is challenged in the future.
        - *Note: This does not apply to gray assets as they do not hold any protections on the network*
    - Buyers receive warnings if flagged assets are listed for sale, ensuring informed purchases.
6. **Buyer Protections**:
    - Buyers are encouraged to research sellers' **Reliability scores** and **transaction histories**.
    - If a purchased asset is later proven to infringe an RWA, buyers can **file claims** for compensation against sellers through the DAPP system.
7. **Subscription Features**:
    - Sellers can opt for premium subscriptions to gain tools such as featured listings, detailed analytics, and automated pricing suggestions.
8. **Monetization Features**
    - The Auction house introduces referral links as well as a coupon code system which gives sellers the ability to provide discounts or incentivize influencers the ability to receive a commission from a sale.

How It Works

The Auction House offers a fair, transparent, and decentralized experience for creators, buyers, and sellers:

1. **Creators** mint assets using standard contracts, protecting ownership and enabling monetization.

2. **Buyers** purchase or bid on verified assets, with tools to check seller trust levels and asset legitimacy.
3. **Resellers** can list assets while ensuring creators receive royalties if applicable.
4. **DAPPs** ensure claims, enforcement, and disputes are handled without UndChain involvement, preserving decentralization.

Visual Seller Classification

To improve trust and clarity, sellers will be visually classified:

- **Verified Sellers**: Creators or trusted resellers with high Reliability scores receive a **verified badge** and enhanced listing visibility.
- **Warning Sellers**: Sellers with low Reliability scores or flagged histories have **warnings** displayed near their listings.

Reliability scores are dynamic and affected by seller behavior, claims, and buyer feedback.

Final Notes

The Auction House ensures that digital assets remain protected, accessible, and monetizable through a robust and decentralized system. It empowers creators with tools to enforce ownership, buyers with transparency, and the network with fraud prevention mechanisms.

Live - Decentralized Real Time Streaming

The Live co-chain is UndChain's dedicated decentralized audio/video streaming co-chain, built to deliver **low-latency**, **interactive streaming**,

and enhanced accessibility while empowering creators to retain ownership and monetization of their content. Its modular design integrates directly into other co-chains like **Pages** and **Smack**, with advanced tools for user interaction, adaptive scaling, and intelligent AI integration through **Mimic**.

Key Features

1. **Decentralized Streaming with Low Latency**
    - Live targets **AV1 encoding**, a royalty-free, bandwidth-efficient codec, enabling smooth, low-latency streaming with minimal resource consumption.
    - Designed for interactive experiences, Live powers **real-time events**, including live chat, reactions, and audience engagement (reactions or polling).
2. **Dynamic Content Distribution**
    - Live leverages a **spread protocol** that dynamically scales streams based on **popularity** and **regional demand**, ensuring high-traffic content is delivered efficiently.
    - Content distribution happens organically as partners prioritize high-demand streams, enhancing availability and reducing latency for end users.
3. **Interactive Streaming Applications**
    - Live supports **advanced user interactivity**, allowing content to respond to viewer actions within the stream. Examples include:
        - **Clickable Video Interactions**: Users can click specific areas of the video to trigger events (e.g., playing different videos or navigating a decision tree). This system allows developers to create interactive video-based applications

like **choose-your-path games** or **interactive learning tools**—serving as a precursor to **Player2**.
        - **Dynamic Polls**: Streamers can integrate real-time polls and feedback mechanisms to gauge audience sentiment and preferences.
    - **Audience-Driven Content**: Viewers' actions and interactions can influence the flow or content of the stream, creating a uniquely immersive experience.
4. **Intelligent AI Integration with Mimic**
    - Live incorporates **Mimic**, UndChain's AI co-chain, to enhance content accessibility and usability:
        - **Screen Reading**: When users pause a stream, Mimic analyzes the screen to enable text recognition. Users can copy and paste relevant text, such as captions, notes, or displayed information.
        - **Closed Captioning**: Mimic automatically generates real-time captions, making streams accessible to a broader audience, including those with hearing impairments.
        - **Intelligent Insights**: Mimic can assist streamers by analyzing viewer behaviors and content interactions, helping optimize future broadcasts.
5. **Screen Sharing with M3L Efficiency**
    - Live introduces a **screen share system** designed for bandwidth optimization and seamless interactivity:
        - Instead of streaming raw video, Live transmits the **M3L (Multi-Media Markup Language)** structure of the shared screen to recipients. This significantly reduces bandwidth usage while ensuring clarity.
        - **Audio Support**: While M3L handles visual elements, audio streams separately to maintain synchronization.

        - **Content Sharing**: If the shared screen plays a video, Live ensures all users pull the video from the same source, eliminating redundancy and maintaining efficiency.
    - This feature is ideal for **tutorials**, **collaborative workflows**, and **interactive presentations**, where precise, low-bandwidth screen sharing is essential.
6. **Full Ownership and Monetization**
    - Streams and recorded broadcasts are tokenized as **digital assets**, enabling creators to maintain ownership, set pricing models, and monetize their content.
        - This will be done with the assistance of the Auction House co-chain.
    - Creators can define **royalty structures** for specific content types like music, video, or interactive streams, ensuring they earn revenue whenever their assets are reused or resold.
    - Revenue-sharing mechanisms allow creators and collaborators to split earnings transparently.
7. **Integration with Pages and Smack**
    - **Pages**: Streamers can embed Live broadcasts directly into M3L-based web pages for seamless Web4 integration.
    - **Smack**: Live supports video uploads and sharing within decentralized social media applications.
8. **Regional and Adaptive Scaling**
    - Popular content is redistributed to local partners dynamically, enhancing performance and reducing latency.
    - Partners are incentivized to prioritize streams with high demand, aligning with UndChain’s decentralized resource-sharing model.
9. **Monetization with USP and AdCoin**
    - Live content can be monetized using:

    - **USP**: The native utility token for Live transactions.
    - **AdCoin**: Can be swapped for USP or earned through network 'ad hubs'.
  - Users benefit from flexible payment options, while streamers enjoy consistent monetization streams.
10. **Content Ratings and User Safety**
  - Live adheres to UndChain's **content rating system**, ensuring streams are appropriately classified.
  - Stream creators can implement **age restrictions** for sensitive content using the Adult_Swim protocol, which verifies users' eligibility without exposing personal data.

Use Cases

- **Interactive Streaming Platforms**: Live alternatives to Twitch, YouTube Live broadcasts, or webinar services with enhanced interactivity.
- **Choose-Your-Path Applications**: Interactive games, training tools, or learning modules where viewers' choices guide the video content.
- **Tutorial and Collaborative Workflows**: Efficient, low-bandwidth screen sharing for educational and professional use.
- **Content Accessibility**: Closed captioning, screen text recognition, and content optimization with Mimic integration.
- **Content Monetization**: Pay-per-view, subscription models, and royalty-based earnings for creators.

Why Live Stands Out

Live combines decentralized streaming, low latency, interactive features, and AI-driven enhancements to deliver a streaming platform unlike any

other. It empowers creators, enriches viewer experiences, and integrates seamlessly into UndChain's broader Web4 ecosystem. Whether embedded in Pages, used on Smack, or powering standalone applications, Live provides the tools to build the next generation of interactive, decentralized streaming solutions.

Developer Focus: Building with Live

Live demonstrates how UndChain's co-chains work together to create powerful, interconnected systems. By integrating components like **Mimic's AI tools**, **Pages' content distribution**, and **Smack's social sharing features**, Live showcases the flexibility and modularity developers can leverage when building their own co-chains.

With UndChain, developers are encouraged to think beyond isolated systems—creating dynamic, interoperable applications that scale effortlessly and provide unmatched user experiences.

Code Ledger

Code ledger is the heart of software development on UndChain, specifically designed for Pseudo. It combines decentralized collaboration, code verification, interactive education, and monetization into one seamless experience for developers. Code Ledger isn't just a repository—it's a **living ecosystem** that rewards innovation, encourages learning, and ensures secure development.

Key Features

1. **Date-Based Versioning**

- Instead of traditional VCS (e.g., Git), Code Ledger adopts **UndChain's date-based versioning** (e.g., 2024.12.17.0) as the default. This ensures clarity and consistency across all projects.
- The **current version** is displayed prominently in the project view, ensuring developers and users always see the latest state of the code.

2. **Module Marketplace**
   - A **dedicated marketplace** where developers can share, buy, or sell standalone modules as **digital assets** (protected under UndChain's Digital Asset Contracts).
   - Modules can be imported from other co-chains, offering tremendous flexibility for developers. For instance:
     - Use **Live** modules for streaming and interactive tutorials.
     - Use **Mimic** modules for AI-powered assistance.
   - The marketplace features:
     - **Hot Modules**: Trending and highly-rated modules.
     - **Categories**: Searchable categories like UI components, AI libraries, utilities, or educational tools.
     - **Mimic Assistance**: Mimic recommends modules based on the developer's project goals, streamlining development.
   - Developers can monetize their modules through a **one-time purchase** or **royalty-based licensing**.
3. **Decentralized Code Audits and Verification**
   - Code Ledger introduces a **peer verification system** where trusted developers (high Reliability Scores) audit code for security, compliance, and performance.
   - Audited projects receive a **Trusted Badge**, helping other developers and users confidently adopt or contribute to the project.

- Auditors are incentivized through rewards for their contributions, with a focus on accuracy and reputation building.

4. **Interactive Developer Education**
    - Code Ledger integrates **interactive tutorials** to guide new developers in learning Pseudo and building decentralized applications.
    - **Mimic-Powered Learning**: Mimic serves as a personal coding tutor, providing explanations, debugging help, and contextual suggestions.
    - Early tutorials will leverage **Live** for video-based courses created by other developers. These video tutorials can be monetized, rewarding experienced developers for helping newcomers.
5. **Integrated Compilation and Testing**
    - Code Ledger features an **on-chain compiler** for Pseudo, ensuring code compliance and eliminating the risk of malicious machine code.
    - Developers can **test** their code directly on the **test net** before deploying it to the main network. Test results include build reports, error diagnostics, and optimization suggestions.
    - Mimic further enhances this by assisting with debugging and optimization during testing.
6. **Blockchain-Based Licensing**
    - Code Ledger supports **standard software licenses** like MIT, GPL3, or Apache. Developers can choose these licenses during project creation.
    - All licenses are protected under **Digital Asset Contracts**, ensuring clarity and enforceability within UndChain’s ecosystem.
    - Custom licensing options are also supported, empowering developers to define unique usage rights and monetization

terms.

7. **Milestone-Based Funding**
    - Code Ledger introduces a **milestone-based funding system** to help projects gain traction and community sentiment.
    - Developers can create funding proposals tied to specific project milestones. Contributions can come from:
        - The UndChain grant system.
        - Community-based funding.
        - Bounty submissions for specific features.
    - Milestone payouts are transparent, immutable, and tied to clear deliverables—building trust between developers and funders.
8. **Forking with Digital Asset Compliance**
    - Developers can **fork** projects safely, preserving full code history while respecting existing **Digital Asset Contracts**.
    - Forks inherit all licensing and usage terms, ensuring continued compliance with the original project.
    - Forked projects are tagged, linking back to the parent project for transparency.
9. **Live-Enabled Collaboration**
    - Code Ledger incorporates **Live's screen share capabilities** for:
        - Pair programming.
        - Collaborative code reviews.
        - Real-time development sessions.
    - By streaming the **M3L content** of a developer's screen instead of video, Live ensures minimal bandwidth usage and a lag-free experience.
10. **Reliability Score for Contributors**

```
- Developers’ contributions are reflected in their **Reliability Score** based on:
    - Successful code merges and accepted contributions.
    - Positive peer audits or reviews from trusted developers.
    - Community adoption of their modules or projects.
- A higher Reliability Score unlocks more opportunities for developers, such as:
    - Eligibility for high-impact project bounties.
    - Faster approvals for code merges.
    - Trusted status when conducting audits or reviews.
- This system rewards contributors who consistently deliver quality code and act with integrity, further strengthening trust within the Code Ledger ecosystem.
```

Why Code Ledger Matters

Code Ledger sets a new standard for decentralized software development by:

- **Empowering Developers**: Tools for collaboration, learning, and monetization make it easier for anyone to build impactful projects.
- **Ensuring Security**: Integrated compilation, peer audits, and reputation systems maintain a high standard for code safety and quality.
- **Driving Innovation**: Features like milestone funding, a module marketplace, and interactive tutorials encourage creativity and collaboration.
- **Integrating the Ecosystem**: By importing modules from co-chains like **Live** and **Mimic**, developers can focus on building unique

solutions without reinventing the wheel.

Final Thoughts

With Code Ledger, UndChain doesn't just provide a decentralized alternative to platforms like GitHub—it builds an **evolutionary system** that rewards developers, fosters innovation, and ensures secure, scalable development for the future. It's the ultimate toolkit for building the decentralized applications of tomorrow.

SQeeL (Decentralized SQL Database System)

SQeeL is UndChain's decentralized SQL database system, designed to create and manage on-chain databases with seamless scalability, robust user features, and enhanced developer tools. Built to balance power and usability, SQeeL transforms decentralized database management into a streamlined, interactive experience.

Key Features

1. **Intuitive Spreadsheet-Like Interface**
    - SQeeL offers a familiar, spreadsheet-inspired interface for visual data interaction, editing, and querying. This reduces the learning curve for non-technical users while still supporting advanced SQL queries for developers.
    - Users can write raw SQL queries or interact with tables visually, similar to Excel or Google Sheets.
    - The UI includes dynamic graphing tools (e.g., bar charts, line graphs, heatmaps), allowing users to visualize data instantly.
2. **Dynamic AI Assistance (Powered by Mimic)**

- **Query Optimization**: Mimic suggests optimizations, flags inefficient queries, and recommends schema changes for better performance.
- **Graph Insights**: Mimic analyzes graph data to highlight trends, anomalies, and usage patterns.
- **Interactive Tutorials**: Mimic provides interactive tutorials on features, query writing, and best practices, helping both beginners and experienced developers.

3. **Cross-Chain Data Linking**
   - SQeeL enables seamless integration with other UndChain co-chains, enriching datasets without duplication.
   - Example: Pull user profiles from Smack, transaction data from the Auction House, or content metadata from Pages.
4. **Real-Time Collaboration**
   - Teams can collaborate on databases in real-time with permission-based access control.
   - Integration with Live allows screen sharing for pair programming, code reviews, or troubleshooting.
5. **Snapshot System and Audit Logs**
   - **Snapshots**: Create immutable “snapshots” of databases for backup, versioning, or archival purposes. These snapshots are stored as digital assets and can be restored on demand.
   - **Audit Logs**: Every change (e.g., row insertion, update, or deletion) is immutably logged on-chain for full transparency and compliance.
6. **Decentralized Data Storage**
   - SQeeL integrates natively with UndChain’s decentralized storage providers for scalable, secure, and redundant storage of large datasets.

7. **Data Monetization Marketplace**
    - Public or private datasets can be monetized as digital assets through UndChain’s marketplace.
    - Subscription models or pay-per-query access unlock valuable datasets for researchers, businesses, and developers.
8. **Module Marketplace**
    - A marketplace for reusable database modules lets developers integrate pre-built systems. Modules are categorized, monetized as digital assets, and can be recommended via Mimic.
    - Example: Developers building a video library can import modules for content tagging, search functions, or user interaction systems.
9. **Forking and Licensing**
    - SQeeL respects UndChain’s digital asset contracts to ensure proper licensing and monetization.
    - Databases or modules can be forked for customization while adhering to the creator’s licensing terms.
10. **Clone Protocol for External Data Integration**
    - The Clone Protocol allows users to ingest external datasets from traditional Web2 or Web3 systems onto the blockchain.
    - Users can point the protocol to existing datasets (e.g., CSVs, APIs, or public datasets), enabling seamless integration and utilization.

Why SQeeL?

SQeeL revolutionizes decentralized database management by combining powerful SQL capabilities with a visual, user-friendly interface and AI-driven assistance. With tools for collaboration, monetization, and integration across co-chains, SQeeL provides a seamless experience for developers, creators, and enterprises alike.

Whether building public datasets, private repositories, or interactive applications, SQeeL empowers you to store, visualize, and manage data effectively on UndChain.

SMACK (Social Media Application for Curators and Keepers)

Smack redefines decentralized messaging and social interaction, offering a robust protocol for communication, community building, and content ownership. Smack is not a social media system but rather a **protocol** that enables platforms and creators to build decentralized applications and networks tailored to their needs.

Key Features

1. Decentralized Messaging and Social Interaction

- Supports direct messaging, group creation, and forum-style discussions.
- Enables scalable social networks, offering both private and public content-sharing options.
- **Interactive Features**:
  - Users can interact with posts using reactions (e.g., likes, dislikes, and emojis).
  - Content can be shared with other users or groups, encouraging collaboration and dissemination.
  - Comment systems allow for threaded discussions and dynamic interactions.

2. Profile Portability and Ownership

- Profiles and content are tied to the user, not the platform. If removed from one curator's system (e.g., Facebook or X), users retain their content, connections, and digital presence across other systems.

3. Curators and Keepers

- **Curators**: Platforms that publish user content, acting as hubs for interaction and visibility.
- **Keepers**: Content creators who maintain full ownership of their works, enabling them to control distribution and monetize their content.
    - Keepers can allow curators to post their content or enter exclusive agreements for monetization.

4. Content Propagation and Promotion

- **Vanilla Smack** includes a basic system for propagating and promoting content using factors like:
    - **Content Type**: Prioritizing text, video, or images based on user preferences.
    - **Current Events**: Highlighting trending topics or local news.
    - **Creator Popularity**: Amplifying content from creators with a strong reputation.
    - **Engagement Metrics**: Factoring in likes, reactions, and shares.
    - **Time Spent Viewing**: Boosting content that retains user attention.
- Users can customize their feed preferences, enabling or disabling specific content types and controlling their viewing experience.

- Transparency: Users can discover why specific content appears in their feed.
- **Curator Override**: Platforms can implement their own curation systems to replace or augment Smack's default propagation behavior, offering tailored content promotion strategies.

5. Monetization Opportunities

- Users can monetize content directly using UndChain's native tokens (e.g., USP or AdCoin).
- Supports subscription-based models, tipping systems, and ad revenue sharing to reward creators.

6. Integration with Pages and M3L/GSS

- Leverages the Pages co-chain for hosting user-generated content like blogs, forums, and community hubs.
- Uses M3L and GSS for adaptive layouts, ensuring accessibility across devices and input methods.

7. Mimic Integration for Personalized Experience

- Mimic AI enhances the platform's experience by curating personalized feeds, suggesting content, and providing real-time translation for multilingual interactions.
- **Important Note**: Mimic can be implemented by creators or platforms (curators) to moderate content or provide insights. **Smack itself does not moderate content**, as it is strictly a protocol, not a centralized system.

8. Privacy and Security

- Messages **can** be encrypted end-to-end, depending on the service (curator) the user interacts with.
    - Example: Direct messages or private group chats can be encrypted, while public posts (e.g., akin to X tweets) are not encrypted since they are intended to be publicly visible.
- Implements zero-knowledge proofs for user verification without exposing personal information.

9. Page's Age Verification System

- Smack integrates with the **Page's age verification system**, enabling platforms to enforce age restrictions. This feature supports compliance with local regulations restricting underage access to social media while ensuring privacy. Platforms can choose whether to allow underage users to participate, fostering safer digital spaces.

10. Data Portability and Compliance

- Users can export their content and connections in a standardized format, ensuring full data portability.
- Aligns with global privacy regulations to offer compliance-friendly solutions for enterprises and individuals.

11. Dynamic Content Scaling

- Uses UndChain's dynamic spread protocol to scale content delivery based on demand and region, ensuring smooth interaction even during high traffic.

Why Smack?

Smack empowers creators (keepers) and platforms (curators) to build decentralized networks and social ecosystems while respecting user ownership and privacy. Its flexibility allows for seamless integration into applications, enabling unique monetization and interaction models.

With tools for personalized content discovery, transparent propagation, and adaptive moderation, Smack creates a fair, user-centric ecosystem. It is the foundation for enabling the next generation of decentralized social applications where users and creators truly own their digital presence.

Player2

Player2 is UndChain's innovative co-chain designed to revolutionize gaming through decentralized technology, immersive storytelling, and cutting-edge tools for developers and players. From multiplayer experiences to single-player narratives, Player2 empowers creators to build the next generation of games while offering players unparalleled ownership and interactivity.

At the core of Player2 is the **Echoes Trilogy**, a showcase of Player2's capabilities that spans generations and genres. This episodic RPG series invites players to explore rich worlds, confront legendary adversaries, and shape the fate of interconnected stories across time and space.

Key Features of Player2

1. **Decentralized Gaming Infrastructure**
   Player2 enables developers to host live game servers on

decentralized networks, ensuring global scalability, low latency, and resilience against downtime or interference.

2. **World Engine: Dynamic Simulated Worlds**
   Player2's **World Engine** empowers developers to create living game worlds where NPC actions and player decisions ripple across the environment, creating a deeply immersive experience.
3. **AI-Powered Game Development (Mimic Integration)**
   - **2D and 3D Content Creation:** Mimic assists developers in designing detailed sprites and 3D models, streamlining asset creation for games of any scale.
   - **Advanced NPC Behavior:** By giving NPCs a background narrative and goals, developers can create dynamic characters that interact with the game world and players in realistic, evolving ways.
   - **Gameplay Analytics:** Mimic provides developers with data-driven insights to optimize game mechanics and player experiences.
4. **Storyboard Environment**
   Player2 incorporates a **Storyboard Environment** where developers can collaboratively design their game's narrative structure.
   - Includes tools for writing text, adding reference images, and mapping out branching storylines.
   - Leverages **Code Ledger** for version control and team collaboration, ensuring a seamless and organized development process.
5. **Meta Space Integration**
   Digital assets acquired in Player2 games are portable, enabling players to use their characters, items, and creations across games and applications within UndChain's **Meta Space**.

6. **Episodic Release Framework**
   Player2 supports episodic game releases, allowing developers to launch shorter, more focused stories and build anticipation for future chapters.
7. **Player-Centric Asset Ownership**
   Using UndChain's **Digital Asset Contracts**, Player2 ensures players have full control over their in-game assets, which can be monetized, transferred, or utilized across compatible games.
8. **Cutting-Edge Hardware Integration**
   Player2 supports a decentralized gaming hardware ecosystem, enabling users to share idle gaming PCs and earn rewards while powering the network.

   Players can also leverage low-powered devices that sync with high-performance PCs on the network, delivering a premium gaming experience without requiring personal investment in high-end hardware.

The Echoes Trilogy

The **Echoes Trilogy** demonstrates Player2's potential, combining compelling storytelling, cooperative gameplay, and groundbreaking technology.

Game 1: Echoes of the Past

In a medieval-inspired world, *Echoes of the Past* follows Eldin as he confronts a neighboring kingdom consumed by corruption.

- **Story Highlights:**
  - Explore a world filled with legendary monsters like witches, hydras, and krakens while unlocking elemental abilities to aid in

combat.
    - Episodic chapters introduce immediate challenges while slowly unraveling a deeper, darker mystery tied to the magical power sword.
    - Forge alliances with unique companions and battle to restore balance to a world on the brink of darkness.

Game 2: Echoes of Fate

Set in a Western-themed frontier, *Echoes of Fate* follows three brothers as they navigate a world filled with moral dilemmas and an unfolding legacy.

- **Story Highlights:**
    - Explore a rugged world shaped by player choices, with evolving relationships and alliances.
    - Encounter unique mechanics such as horseback combat and large-scale battles.
    - A mysterious antagonist pulls the brothers deeper into the secrets of their ancestors.

Game 3: Echoes of Destiny

In a modern world with some futuristic aspects, *Echoes of Destiny* ties the trilogy together in an epic finale.

- **Story Highlights:**
    - Navigate a world where ancient magic meets cutting-edge technology, introducing players to robotic adversaries and futuristic landscapes.
    - Prepare for the ultimate confrontation as players face off

against a malevolent force that threatens the fabric of reality.
- Meaningful choices shape the outcome of the trilogy, offering a personalized conclusion to this generational story.

Why Choose Player2?

1. **For Developers:**
   Player2 provides robust tools to create multiplayer and single-player games with decentralized infrastructure, ensuring resilience, scalability, and seamless integration with UndChain's ecosystem.
2. **For Players:**
   From owning and trading assets to immersive gameplay and connected experiences, Player2 puts players at the heart of a decentralized gaming revolution.
3. **For the Future of Gaming:**
   Player2 sets the foundation for a decentralized gaming ecosystem that prioritizes fairness, creativity, and innovation.

Player2 is more than a co-chain; it's the cornerstone of a decentralized gaming future. With tools for developers, rewards for contributors, and unforgettable experiences for players, Player2 invites you to shape the next era of gaming.

Summary on Co-Chains

As you can see, UndChain offers an expansive framework for developing diverse co-chains, each contributing unique functionality to the ecosystem. While I may not be able to personally develop every co-chain idea presented here, my goal is to create a strong foundation that

inspires innovation and ensures the protocol can adapt to the varied needs of these co-chains.

Through firsthand experience in building these systems, I've learned that actively using a product uncovers insights that might otherwise be missed. My hope is that developers approach their creations with a focus on delivering tangible value—not just another asset class—to enhance the ecosystem as a whole.

Potential Co-Chain Ideas

To spark inspiration for future development, here are a few ideas for co-chains that I believe could thrive on UndChain:

1. **Decentralized Rideshare and Delivery Services**
   Imagine a decentralized version of Uber or DoorDash, enabling peer-to-peer ridesharing and delivery services without centralized control. Such a system could bring fairer pricing, more equitable profit distribution, and enhanced user control.
2. **Decentralized Marketplace**
   A decentralized Amazon-like co-chain could link small, independent shops into a unified network. This would empower local businesses to compete with larger corporations while offering consumers diverse options from around the world.
3. **Decentralized Science (DeSci)**
   A co-chain designed for scientific collaboration could connect users to a network of scientific equipment. From weather stations and geomagnetic sensors to RF analyzers and telescopes, this system could create a global monitoring network for advancing research and discovery.

4. **Decentralized Learning Platforms**
   A co-chain dedicated to education, offering decentralized courses, certifications, and knowledge sharing. Instructors can host live classes, sell pre-recorded lessons, or even provide one-on-one tutoring sessions. This co-chain could integrate Mimic to act as a personalized tutor, guiding students through material and answering their questions.
5. **Crowdfunding and DAO Co-Chains**
   A co-chain designed for fundraising via decentralized autonomous organizations (DAOs). Projects can raise funds through transparent smart contracts, while contributors have voting power to decide the project's direction. This system could include milestone-based funding, ensuring accountability for developers.
6. **Digital Healthcare Services**
   A decentralized Tela-Health co-chain where patients can securely connect with healthcare providers for consultations, prescriptions, and medical advice. Health data could be stored privately on the network, and Mimic could assist with diagnostics or routine health inquiries.
7. **Decentralized Talent Marketplaces**
   A co-chain where freelancers and businesses connect without intermediaries. Users can showcase their skills, bid on projects, and receive payments through UndChain's token systems. Built-in reputation systems ensure fair and trustworthy transactions.
8. **Event Ticketing and Management**
   A co-chain for creating, selling, and managing event tickets. Tickets would be issued as digital assets, ensuring authenticity and preventing fraud. Event organizers could even create virtual venues using Live and Player2 for decentralized concerts or esports events.

9. **Decentralized Energy Sharing**
   A co-chain allowing users to buy, sell, or share surplus energy locally. Homeowners with solar panels, for instance, could sell excess electricity to their neighbors in real time, creating a decentralized energy market.
10. **Legal and Compliance Services**
    A co-chain offering tools for creating, sharing, and enforcing digital contracts and agreements. It could feature a library of pre-built templates for licenses, NDAs, and other legal documents, with built-in dispute resolution protocols.
11. **Decentralized Travel Booking**
    A co-chain that connects users with travel providers, allowing them to book flights, hotels, and tours directly. This system would remove intermediaries, reducing costs and providing a more transparent booking experience.
12. **Supply Chain Management**
    A CRM (Customer Resource Management) system that allows users to track and monitor manufactured goods over an entire organization. This system could provide manufacturing instructions, estimate material usage and plan accordingly and provide real time tracking that could be provided to customers.
13. **Decentralized Music and Video Streaming Services**
    This co-chain enables creators and platforms to host music and video content in a decentralized ecosystem. By integrating features from **Live**, it facilitates seamless content streaming with robust scalability and low latency. The use of **rental digital asset contracts** allows content providers to define specific usage rights, such as viewing-only or limited-time access, ensuring creators maintain control over their work while providing users with a seamless experience.

This co-chain paves the way for decentralized streaming platforms—think of it as a blockchain-powered alternative to services like Netflix or Spotify—where content ownership, control, and monetization are shifted back to the creators and their audiences. Additionally, creators can use the built-in functionality of Pages to host their libraries and leverage AdCoin for monetization options.

Open vs. Closed Source

While I firmly believe in the principles of open-source development, UndChain also provides the flexibility to run closed-source code. This capability is achieved through **Preferred Partners**, systems that execute Creator code directly without requiring compilation by the Code Ledger co-chain. However, implementing closed-source systems introduces challenges, particularly regarding security and transparency, as public scrutiny of the code is limited.

This approach may seem to contrast with the ideals of decentralization, but UndChain is designed to go beyond rigid definitions. By blending the strengths of both decentralized and centralized systems, UndChain ensures developers have the freedom to use the tools that best meet the unique requirements of their applications. Whether it's full transparency through open-source projects or the tailored control of closed-source solutions, UndChain is built for the future—empowering developers to innovate without compromise.

User Types in UndChain

UndChain features four distinct user types, each designed to balance power and ensure no single entity can compromise the integrity of the

chain. Here's how they function:

1. **Creators**
   Creators are the architects of a co-chain. They define its functionality, tokenomics (fees), and governance rules. Upon launching a chain, owners receive a digital asset certifying their ownership, which can be transferred or shared with up to two additional partners. Owners also select **preferred validators** (often themselves) and determine the number of validators on their network, with a minimum of three. *Concerns about centralization? Don't worry; safeguards are in place.*
2. **Validators**
   Validators act as the backbone of the network, akin to routers. Their responsibilities include:
   - Connecting clients with partners to fulfill requests.
   - Validating and recording transactions, ensuring data integrity.
   - Maintaining the **Users File**, a secure directory mapping usernames to public keys.
     Validators play a crucial role in keeping the network functional, secure, and efficient.
3. **Partners**
   Partners are UndChain’s equivalent of miners, but instead of solving cryptographic puzzles, they perform meaningful tasks in three utility categories: computation, storage, and access.
   - Partners collaborate with validators by fulfilling jobs listed in the **Jobs File**.
   - After completing a task, they sign a receipt with the client, ensuring transparency and accountability.
     Partners are essential for driving the decentralized cloud services that UndChain provides.

4. **Clients**
   Clients are the end-users of UndChain’s resources, engaging with services for entertainment, productivity, or business. They’re not merely consumers but an integral part of the chain's security.
   - Clients sign receipts upon completing transactions, ensuring trust and verifying job fulfillment.
   - By participating in this process, clients help secure the network while benefiting from its decentralized services.

Block Generation

Blocks on UndChain are generated using a fixed block time rather than size to ensure consistent and predictable operation. This design choice mitigates the risk of rapid chain expansion by offloading completed items such as job files, work files, and payout lists to partners for storage. Additionally, a process called **The Convergence** consolidates the chain to optimize storage and performance; this is explained further in the encryption standards section. Each block is generated in three distinct stages, with each stage consisting of three steps.

The timing of block generation will initially be experimental. Blocks need to be long enough to accommodate a sufficient number of job requests while maintaining responsiveness for users. The current target is **44 seconds per block**. While blocks finalize at this interval, transaction propagation is near-instantaneous, providing users with a fast and seamless experience.

Transaction Throughput and Responsiveness

UndChain decouples transaction placement from block finalization. Transactions are processed and placed in the **job file** almost immediately after being initiated, allowing clients and partners to proceed without waiting for block finalization.

This approach ensures:

1. **High Transaction Throughput (TPS):** The network can process a significant number of transactions per second by leveraging real-time job file updates.
2. **Responsive User Experience:** Transactions are swiftly propagated to validators, enabling rapid initiation of utility tasks.
3. **Scalability:** Validators and partners dynamically distribute workloads to maintain system responsiveness even under high demand.

Stage 1 - Initialize

1. **Request:** A client submits a service request to the validators. Validators record this request in a job file, enabling partners to identify and fulfill the required utility.
2. **Link:** Partners signal their availability to the validators, indicating they can perform the requested utility. This step helps maintain a steady flow of tasks to eligible partners.
3. **Sync Jobs:** Validators synchronize the job file for the current block with other validators to ensure consistency across the chain.

Stage 2 - Link

1. **Search:** Validators actively monitor for available partners to fulfill job requests or directly contact known partners for specific tasks.
2. **Partner(s) Found:** Upon receiving sufficient responses, validators establish a connection between the client and selected partners. Proxies may be employed to safeguard against potential DDoS attacks.
3. **Connection Established:** The connection details and corresponding block number are recorded in the **work file**. Validators also listen for relevant hashes from interconnected co-chains.

Stage 3 - Utility Complete

Utility tasks are not required to conclude within a single block and may span multiple blocks. Completion occurs either when the client or partner terminates the transmission or at predefined payout intervals (e.g., every 4 hours on the main chain).

1. **Receipt:** Both the client and partner must submit a receipt of the utility. This receipt, signed by both parties, serves as verification of the transaction. Submissions from both parties are recommended to ensure redundancy.
2. **Utility Complete:** Validators mark the utility task as complete, recording the completion block and appending the transaction receipt. *Note: Receipts must remain under 128 characters to optimize storage, though they may link to larger off-chain files.*
3. **Payout:** At the Creator's defined intervals, payouts are processed. Transactions not included in the current payout are appended to the **pay sheet** for inclusion in the next interval. Validators hash the payout data and append it to the block, sharing the hash with interconnected co-chains.

4. **Subscription Model**: UGP will introduce the subscription style model system. If users choose they can subscribe to UndChain's services for a flat rate which allows near infinite requests. *There is an anti-spam system*

Note on Adjustments

Block timing and related parameters may be adjusted based on real-world data to ensure optimal transaction throughput and storage efficiency. This iterative approach ensures UndChain remains both responsive and sustainable as the network grows.

Tokenomics

UndChain employs two primary utility tokens on its main chain—**UndChain Gold Piece (UGP)** and **UndChain Silver Piece (USP)**—to incentivize validators and partners. Additionally, **AdCoin**, a token specific to the Pages co-chain, demonstrates how developers can implement custom tokens to override USP for specific functionalities.

Note: The blockchain experienced a soft launch on April 24, 2024. All token emissions from this date until the network officially launches are effectively burned and will not enter circulation.

UndChain Gold Piece (UGP)

UGP serves as the primary reward token for validators, following a deflationary model inspired by Bitcoin. It introduces periodic reductions in daily emissions, referred to as **floorings**, to promote long-term sustainability and maintain validator interest without inflating fees.

Key Features

1. **Initial Supply:** 4,444,444 UGP minted at launch for development, advertising, and airdrops.
2. **Daily Emission:** Begins at **4,444 tokens per day** and reduces every four years through floorings until it stabilizes at **1 token per day** indefinitely.
3. **Subscription Model:** UGP introduces a subscription-based system, enabling users to pay a flat rate for nearly unlimited requests while implementing anti-spam measures to protect the network.
4. **Infinite Supply Model:** The diminishing emissions ensure gradual token scarcity while maintaining validator incentives.

Flooring Schedule

Flooring	Year	Tokens Per Day
1	1	4,444
2	4	2,222
3	8	1,111
4	12	555
5	16	277
6	20	138
7	24	69
8	28	34

Flooring	Year	Tokens Per Day
9	32	17
10	36	8
11	40	4
12	44	2
13+	48+	1

UndChain Silver Piece (USP)

USP rewards partners for providing computational, storage, and access services. Unlike UGP, USP's emission schedule is dynamically adjusted through voting by active partners.

Key Features

1. **Initial Supply:** 44,444,444 USP minted at launch.
2. **Daily Emission:** Begins at **4,444 tokens per day** with periodic adjustments through partner voting.
3. **Voting Mechanism:** Partners vote every four years to either halve or double the emission rate.
4. **Eligibility for Voting:** Partners must complete at least one transaction per week and maintain a Reliability score above **444** to participate in the vote.

This dynamic voting mechanism enables decentralized governance and can serve as a testing ground for broader governance features across UndChain.

AdCoin

AdCoin is the native token of the Pages co-chain and integrates into other systems, such as Mimic, to facilitate user rewards and microtransactions.

Key Features

1. **Initial Supply:** 4 trillion AdCoin minted at launch to ensure accessibility and affordability.
2. **Utility:** Designed for micro-transactions, rewarding users for interacting with ads or classifying content, and granting access to various co-chain functionalities.
3. **Onboarding Support:** Distributed through airdrops and tutorials to educate new users on earning and spending AdCoin.
4. **Burn Mechanism:** During halving events, users can vote to implement a burn mechanism to reduce AdCoin supply.

Token Guidelines

- **Decimal Precision:** All tokens are divisible up to **12 decimal places** for precision, using metric prefixes (e.g., milli, micro, nano). For instance, **100µ UGP** represents **0.0001 UGP**.
- **Cross-Chain Token Sharing:** Tokens like UGP and USP are distributed across co-chains based on transaction volume, ensuring equitable rewards for contributors.
- **Burn Address:** A dedicated burn address, reserved as `@Burn`, is built into the protocol to enable irreversible token destruction.

Fees

UndChain’s fee system reflects its core principles of digital ownership and fairness:

1. **Partner-Set Fees:** Partners can independently set fees for services, with a portion allocated to the Creator. Validators cannot set fees.
2. **Network Suggested Fee:** Partners can opt for a network-suggested fee based on what the UndChain protocol recommends.
3. **Auto-Fee System:** Dynamically adjusts fees based on token value and network activity.
4. **Fee Reduction Voting:** Users and partners can vote to reduce fees temporarily if token values become overinflated.
5. **No-Fee Model:** Partners and co-chains can opt to charge no fees, though this forfeits emission-based rewards. This supports non-profit initiatives.

Transaction and Scalability Considerations

1. **Encouraging Collaboration:** Developers are incentivized to use established tokens like USP to minimize market fragmentation.
2. **Scalable Design:** Token-sharing mechanisms ensure rewards align with chain activity, promoting a sustainable ecosystem.

Digital Assets on UndChain

In the era of decentralized networks, **digital assets** are the backbone of user engagement, creativity, and commerce. Unlike traditional NFTs, which merely record ownership, UndChain's assets bring enforceable rights, clear usage terms, and choices for creators and users alike. This section provides an overview of the contract types, enforcement mechanisms, and protections that make UndChain's digital assets uniquely valuable.

Predefined Digital Asset Contracts

Each digital asset on UndChain is governed by a **predefined contract** that determines how it can be used, shared, and monetized. These contracts are color-coded for easy recognition, ensuring transparency for creators and users at all levels. Here's a summary of the five contract types:

1. ⚪ **Unregistered (Gray)**: Basic NFT-like ownership recorded on-chain without enforceable rights or protections.
    - **Details**: Unregistered assets simply establish ownership on the network but lack any claims or IP protections. If the owner chooses, they may upgrade an unregistered asset to a registered type with full protections later; however, the protections only apply from the reclassification date onward.
    - **Example**: An artwork uploaded without protections can later be upgraded to ensure royalties on resale, but initial transactions remain unaffected.
2. 🟢 **Rental (Green)**: Limited, single-use access for temporary asset consumption.
    - **Example**: Renting an e-book, video, or course with limited access rights.

- **Key Benefits**: Provides short-term access without resale rights, allowing creators to retain full control over the asset's use.

3. 🔵 **Single Use + Content Sharing (Blue)**: Unlimited personal use with limited sharing rights for reactions, commentary, and resale with royalties back to the creator.
   - **Example**: Purchasing a video to enjoy personally, resell with royalties to the creator, or share in a reaction or commentary format (e.g., YouTube reactions).
   - **Key Benefits**: Supports creative engagement through limited sharing while maintaining control and royalties for the creator.
4. 🟣 **Ownership (Purple)**: Full ownership, granting users the right to share or resell the asset as-is without modifications.
   - **Example**: Buying a collectible or media asset to keep, share, or resell without altering.
   - **Key Benefits**: Grants users greater control over the asset while preserving the creator's original work and intention.
5. 🟡 **Full Use (Gold)**: Complete ownership with the freedom to modify, remix, resell, and use without restriction.
   - **Example**: Licensing music or an artwork to incorporate into new projects, remix, or resell with modifications.
   - **Key Benefits**: Highest level of creative freedom, ideal for assets meant for open use, allowing users to fully repurpose the asset as if they created it.

These predefined contracts address the limitations of traditional digital assets, giving both creators and users well-defined rights and responsibilities.

Enforcement Through Digital Asset Guardians (DAGs)

UndChain’s **Digital Asset Guardians (DAGs)** are independent entities responsible for enforcing the network’s Digital Asset Program. By bridging decentralized principles with real-world compliance, DAGs ensure that asset contracts are meaningful and actionable.

Key Roles of DAGs:

1. **Asset Validation:** DAGs verify that digital assets comply with UndChain's standards and terms. This includes monitoring for unauthorized use or duplication of existing works.
2. **Dispute Resolution:** DAGs mediate disputes between creators and claimants, providing a fair and balanced resolution mechanism.
3. **Revenue Distribution:** For assets with multiple contributors or shared ownership, DAGs oversee revenue splits to ensure fair compensation.

Mimic’s Role in Enforcement:

- **Duplicate Detection:** Mimic analyzes digital assets for potential duplicates, flagging works that may infringe on existing copyrights.
- **Consumer Warnings:** If a duplicate is detected, Mimic provides a warning visible to potential buyers, along with the creator's response.
- **Neutral Role:** Mimic does not block minting but empowers consumers with transparent information, enabling informed decisions.

Consumer Protections:

- **Transparency:** Buyers can see warnings and creator responses directly on the asset's listing.

- **Accountability:** Creators minting flagged works must provide an explanation, fostering trust and reducing misuse.

Why DAGs Are Essential

Without an enforcement mechanism, digital asset contracts are meaningless. DAGs bring legitimacy to UndChain's asset system by offering active protections and reliable dispute resolution. By combining DAG oversight with Mimic's AI-driven insights, UndChain ensures a balanced ecosystem that empowers creators and protects consumers.

Claim Filing and Dispute Resolution

To ensure transparency and fairness, UndChain incorporates a decentralized process for filing and resolving claims regarding digital assets. This system operates independently through DAGs and is supported by the protocol's foundational tools.

Filing a Claim:

1. **Initiating a Claim:**
   Creators or rights holders can file a claim via a DAG. To initiate, they must:
   - Provide evidence supporting their ownership or IP rights (e.g., copyright documentation, timestamps, or original work).
   - Pay a **claim fee** (determined by the DAG and dependent on their Reliability score) to discourage frivolous claims.
2. **Sponsor Involvement:**
   Claimants select a **sponsor**, an independent DAG member or organization, to vouch for their claim's validity. Sponsors validate the provided evidence and enhance the credibility of the claim.

3. **Notifying the Asset Owner:**
   Upon filing, the current asset owner is notified and given a set period (e.g., 7-14 days) to respond. The asset in question may be temporarily frozen to prevent transfers or modifications during the dispute.

Responding to a Claim:

1. **Providing Evidence:**
   The accused owner can contest the claim by presenting their own evidence, such as proofs of prior creation or purchase.
2. **Counter-Claim Filing:**
   If the accused believes the claim is false or malicious, they can file a **counter-claim**, which also requires a fee based on their Reliability score.
3. **Resolution through DAGs:**
   DAGs mediate the process by reviewing evidence, consulting sponsors, and reaching a verdict. All findings and outcomes are transparently logged on-chain.

Resolution and Outcomes:

1. **Valid Claim:**
   If the claim is deemed valid:
   - The asset is transferred to the rightful owner.
   - The claim fee is returned to the claimant.
   - The original owner may face a **Reliability score penalty** for knowingly or unknowingly minting an infringing asset.
2. **Invalid Claim:**
   If the claim is rejected:

- The claim fee is awarded to the accused owner as compensation.
- The claimant's Reliability score is penalized to discourage baseless claims.

3. **Legal Escalation:**
   In cases involving significant legal implications, DAGs may require both parties to identify themselves and comply with jurisdictional laws.

Mimic's Role in Disputes:

Mimic aids both parties during the claim process:

- **Claimant Support:** Mimic assists in gathering evidence and preparing documentation for the claim.
- **Owner Support:** Mimic helps owners compile counter-evidence and verify timestamps or metadata.
- **Transparency Tools:** Mimic provides clear, accessible overviews of claim details and outcomes for all stakeholders.

Protocol Forks for Ownership and Representation

UndChain's protocol remains neutral, avoiding mandatory enforcement to reduce network burden. Instead, **protocol forks** enable flexibility and creator-defined enforcement:

- **Ownership Fork:** Sets the asset's primary owner, governs revenue distribution, and can initiate actions like asset freezes in case of disputes.

- **Representation Fork:** Allows creators to assign an **enforcement mechanism** (such as DAGs) for handling claims, enforcing rights, and managing disputes if needed.

These forks create the foundation for a decentralized enforcement system, giving creators full control over their level of protection without altering the core protocol.

Built-in User Protections and Decentralized Dispute Resolution

UndChain's digital asset model includes comprehensive user protections to create a balanced environment:

- **Transparency with Color-Coded Contracts:** Clear asset terms prevent misunderstandings and allow users to make informed decisions.
- **Accountability in Dispute Resolution:** Both claimants and users are accountable in disputes, with sponsors providing support and verification to prevent misuse.
- **Modular, Decentralized Enforcement:** Optional DAGs and protocol forks allow enforcement and monitoring to evolve without impacting the core protocol, promoting innovation within the UndChain ecosystem.

Advantages of Enforceable Digital Assets on UndChain

The UndChain digital asset model brings unique advantages over traditional NFTs and decentralized assets:

- **Clear Rights and Responsibilities:** Predefined contracts outline specific usage rights, empowering creators and users to understand and control asset interactions.
- **Optional, Effective Enforcement:** By choosing DAGs, creators can access a robust IP protection system that operates independently of the main protocol.
- **Upgrade Flexibility for Unregistered Assets:** Unregistered (Gray) assets offer a straightforward entry point, with the flexibility to upgrade protections as desired.
- **Future-Proof Flexibility:** The enforcement system, combined with protocol forks, allows the ecosystem to grow, adapt, and incorporate new enforcement mechanisms without requiring protocol changes.

Through enforceable digital assets, UndChain enables a **new standard in digital ownership** for Web4, balancing creator rights with user freedom while remaining scalable and neutral at its core.

Network Fundamentals

Accounts

Anyone can generate an account on UndChain just as they can with any other blockchain, by generating a public/private key pair. Once you have generated your keys, simply send in a new user request message. Usernames, which begin with an @ symbol (e.g., @Undline), provide an easy means of transferring digital assets using a name instead of a long public key. These names are resolved on the validators' end using UnaS (UndChain Naming System), which acts like DNS.

While usernames are not a requirement, public keys are necessary for adding you to the public ledger and allowing other users to reach you. This ensures that even new users without tokens can still participate and interact on the network. Users can receive tokens through drops or earn them by actively participating in the network.

If you choose to register a username, we implement a staking system to ensure they are genuinely used and to prevent malicious actors from squatting on common names. Users are required to stake a set amount, equivalent to **44 USP**. If the user remains active on the network (using chains, sending digital assets, or creating content consumed by other users), they will receive the stake back. The account will be reviewed 30 days after generation to determine if it qualifies for the refund. If the user fails to meet the activity requirements, the period will be extended by another 30 days. If after 90 days, the user still doesn't meet the requirement the stake is lost.

Although this approach necessitates a stake for securing a unique name, it effectively prevents gaming the system while maintaining user anonymity within the base protocol. Note that account names are non-transferable due to security concerns, ensuring that each username is tied to its original account.

Accounts are refundable after demonstrating active participation in the network, and usernames should be considered a critical part of the security infrastructure, especially when we need to switch encryption protocols in the future. If you cannot come up with a name, one can be autogenerated for you. Users should choose a username for enhanced security, as it allows us to modulate our encryption schemes more effectively.

Key Points:

1. **User Request:** Generate a public/private key pair and send a new user request message to create an account.
2. **Public Key Requirement:** Public keys are necessary for adding you to the public ledger and for other users to reach you.
3. **Username Registration (Optional):** Register a username with a refundable stake, reviewed after 30 days of network activity.
4. **Non-transferable Usernames:** Usernames cannot be transferred to ensure security.
5. **Autogenerated Names:** If needed, usernames can be autogenerated.
6. **Enhanced Security:** Users should choose a username for better security and encryption modulation.
7. **Token Acquisition:** New users can receive tokens through drops or earn them by participating in the network.

Organizations

An organization is a group of user accounts on UndChain that have access/privileges to shared resources. There are different roles within an organization that can control various utilities (e.g., digital assets, data, software, and points of contact). There are predefined account types such as president, developer, support, board member, manager, editor, contributor, auditor, moderator, advisor, marketing, treasurer, analyst, and operations. Each organization will begin with a % symbol followed by the organization name (e.g., %UndChain for the UndChain system).

Unlike user accounts, organizations do cost tokens to mint. However, after minting, you can add as many members as you wish with however many roles. Members can also choose to transfer their role in the same way that any digital asset is transferred on chain, with the exception that the president must agree to the transfer as well as the transferee.

Organizations are not based on a staking period and are not refunded. The cost to mint an organization is equivalent to $44 in 1986 USD, which must be paid in USP.

An organization is automatically created when you purchase a new co-chain, providing an immediate structure for managing the co-chain's operations and resources. Note: The cost to purchase a co-chain is 444$ in 1986 USD

The president can also appoint managers to handle member additions and deletions on their behalf, allowing for delegation of responsibilities.

Enhanced Features:

- **Customizable Roles:** Create custom roles in addition to predefined ones, tailored to the organization’s specific needs.
- **Role-Based Access Control (RBAC):** Assign specific permissions to each role for more granular control and security.
- **Activity Logs:** Maintain detailed logs for tracking changes and auditing actions.
- **Multi-Signature Accounts:** Require multiple signatures for critical actions to enhance security.
- **Smart Contracts:** Manage smart contracts directly within the organization.
- **Voting Mechanisms:** Built-in voting systems for governance and decision-making.
- **Notification System:** Alerts for important actions, changes, or events.
- **Membership Tiers:** Different access levels and benefits for various membership tiers.

- **Collaboration Tools:** Shared documents, project management, and communication features.
- **External Integration:** Connect with external services and APIs for enhanced functionality.
- **Reputation System:** Recognize and reward valuable members based on their contributions.
- **Escrow Services:** Secure transactions with built-in escrow services.

Role Descriptions:

- **President:** The leader of the organization with the highest level of control, including the ability to approve role transfers and appoint managers.
- **Developer:** Responsible for managing and developing digital assets, data, and software within the organization.
- **Support:** Provides assistance and support to other members and users interacting with the organization.
- **Board Member:** Participates in decision-making and governance of the organization.
- **Manager:** Handles day-to-day operations, including adding and deleting members, and can be delegated tasks by the president.
- **Editor:** Manages content creation and editing within the organization.
- **Contributor:** Provides content or resources with limited decision-making power.
- **Auditor:** Reviews and audits the organization's activities and transactions.
- **Moderator:** Manages community interactions and ensures adherence to rules and guidelines.
- **Advisor:** Provides strategic advice and guidance to the organization.

- **Marketing:** Manages marketing strategies and campaigns for the organization.
- **Treasurer:** Manages the financial assets and transactions of the organization.
- **Analyst:** Conducts analysis and provides insights on various aspects of the organization's operations.
- **Operations:** Handles day-to-day operations and logistics.

Consensus and Rotation

UndChain operates on a 2/3 majority consensus mechanism, the reason behind this is that since a Creator is able to set up to 44% of the available slots with **known validators** (which guarantees them a spot), they have to get buy in from another 23% of unknown validators. This consensus system also works in regulating unknown validators since they must have buy in from at least 11% of the known validators. There is a potential problem with this however, the Creator could simply create more validators that are not listed and have them run on the network.

Due to this potential vulnerability, a rotation schedule for validators will be implemented that forces **active validators** to swap out with **passive validators** which should reduce the risk.

Between the rotation schedule and the 2/3 consensus mechanism I believe this will enforce the network to move forward. This also allows up to 32% of the validators to act in a deceptive way. There is a mechanism for validators and partners who operate in a deceptive manner which is called the Reliability score. If malice is detected your Reliability score is dropped.

Reliability Score

The idea behind the Reliability score is so that a validator or client can judge the merits of a user without having to do blockchain analysis on that user. This system **MUST** be closely monitored and can **NEVER** be used as a means of preventing transactions from occurring. This isn't a social credit score, but more of a reliability rating of that particular user. All users can experience a reduction in their Reliability score, but validators and partners are the most affected since it can prevent them from providing service on the network. Much like the ledger, the score is maintained by the validators (meaning they request the score to be updated), but is stored by the partners.

Reliability Score - Loss

Validators can loose their Reliability score if

1. They provide false records of transactional events
2. They do not maintain up to date user / job / work / payout files
3. They fail to perform a transaction on a particular user(s)
4. Drop connection without notice
5. Being blocked from a large number of clients

Partners can lower their Reliability score if

1. Failure to maintain saved files
2. Falsify a transaction with a client
3. Are too slow for the task they promised they could do
4. Drop connection without notice
5. Blocked from several clients - *Not sure about this one*

Reliability Score - Gain

It's important to stress that this is **NOT** a social credit score, but like a credit score you can rebuild it. Since I am sure there will be instances where this happens by accident (especially with disconnects).

Validators can gain Reliability score by

1. Being a passive validator
2. Participating on the test net
3. Recovering from a outage 7x (that means the resource has been used 7 times with no problems)
4. Having a higher number of connections than your peers

Partners can gain Reliability tokens by

1. Participating in the test net
2. Getting un-blocked
3. Recovering from a outage 7x (that means the resource has been used 7 times with no problems)

I am concerned about using the block feature as a means of determining a Reliability score as I am sure it will be used in a poor manner however, clients only hurt themselves when they block other users since that prevents them from using their services. Also, we should consider placing a mechanism that disregards the block if it's noted that client has a abnormally large blocked list.

In order to make building the Reliability score more transparent we are introducing the XP system. With this system you will be able to see how much XP you need to get to the next level on your Reliability score. Along with XP, we will have seasonal events where if a participant meets the goals of the network they can earn XP boosts.

Scaling and Security

UndChain should move very quickly since we use so few validators in order to issue work on the chain, however there is a problem with network delay due to physical size so the use of parallel networks is to be used. The idea is that a parallel network operates in the same manner as each other with the exception being their user base. The user declares which network they wish to be apart of and with that they are provided a list of the validators that are responsible in that region to interact with. This doesn't mean that you are stuck on that network permanently though. You can initiate a transfer at anytime and your assets will travel with you. Again the reason for the parallel networks is to increase QoS by making content more local to that specific space, this includes the ledger. Much like how tokens are dispersed across co-chains they are also dispersed the same way with networks. Token allotment is decided by network 1 and is based on activity.

How do you create a new network? You don't :)

I still haven't decided just how I am going to do this, but I believe it will most likely be a mixture of algorithmic as well as Creator responsibility. Perhaps I could leave it to a vote?

Convergence

I believe that this blockchain is going to get large fast even with some of the space constraints that I am already planning on implementing. So what I am planning on implementing is a by product of the fact that we have decentralized storage on chain. What I am planning is that at least every flooring we implement The Convergence which is a process where we store all of the blockchains information on file and effectively restart the blockchain using the hash of the old chain as a genesis block. This

would effectively lower the barrier in regards to how much RAM a validator needs since they can offload it onto the partners to store. The convergence will also occur when we have a cryptography swap (example switching to Kyber).

Security

There is a concern with quantum computing on the rise that traditional cryptographic algorithms may become obsolete. If history tells us anything is that it's a matter of when a cryptographic algorithm will be cracked rather than if. Because of that potential danger, UndChain can modulate and / or change it's cryptographic algorithm. *This is why we have usernames rather than just use public keys* Initially we will use standard AES encryption since that is industry standard and seems like it should work for quite a while longer. When we are close to that algorithm being compromised we will initiate a crypto swap. This will notify users to generate a new public / private key pair using a more secure cryptographic algorithm. The user then responds with the public key to the new protocol and signs it with the current private key so we know it comes from them. The Validators then update the users file to reflect the new public key and on the day of go live we use those signatures instead. If a user fails to update their keys then during the next transaction we initiate a forced key migration (which sounds worse than what it is, basically we send a emergency message back saying to update the key). If that fails, then we allow the account to continue, but notify them that the account is high risk and can be compromised at anytime. UndChain should NEVER force a user to update their keys, but should warn them of the danger of not doing so. *Important to note, once we reach zero day (meaning the attack is live) we do not allow the user to change the key since we don't know if it's them. Instead they can use*

the freeze feature and hopefully it will go to a updated account. Or they can loose their digital assets if and when someone cracks their account.

- Interesting thought would be to allow users to update their key at anytime, but I am not sure why anyone would do this. The validator has to mark the active public key anyway so it wouldn't matter if they did update their keys.
    - Could lead to an attack if users are constantly updating their keys. Maybe charge for out of sequence key update?

Pseudo Language

What is Pseudo?

Pseudo is a Pythonic-style language designed for securely creating real-world programs that run directly on chain within the UndChain protocol. It's main goal is to allow developers (Creators) a way to execute decentralized applications while maintaining a high level of security, fixability and accuracy. It provides a sandboxed environment that ensures no partner nor validator can exploit the system, protecting personal files and resources during the execution of these scripts. Pseudo is a heavily typed language so you will need to declare a variables type prior to using it such as: `my_var: i32 = 0`.

Sandbox Environment

Pseudo runs all of it's scripts within a sandboxed environment that is designed to limit access to the hosts (partner or clients) machine, so that malicious code should not be able to access data or resources that are outside the scope that the program can run within. This is done by the partner dedicating portions of their system for UndChain specific tasks; for example when setting up a new worker the user is asked how much

of their storage space they want to dedicate to the blockchain (has to be within 4GB segments). At that point the 'working folder' cannot go outside this area. Any attempts to do so will result in a drop in Reliability score from the Creator.

On Chain Compilation

Other than having the language create a sandbox environment, another security feature is that Pseudo code is compiled on chain. This means that if there is an attempt to access data outside of the range / scope of this program it will return as an error. The one exception to this rule is when UndChain is being ran on a local network, although technically it's still remote because the partner user type would need to compile the code.

Co-Chain Extensibility

Developers can import functions from other co-chains, extending the language functionality using the `from` and `import` keywords (in the same way that python does). *I am debating on 'installing' a co-chain as a means of preventing issues with naming conflicts, but I think this will work as is*

Example

```
from pages import ratings, client

if ratings(undchain.info) >= client.age:
    show()
```

In this example we are importing the ratings section of pages as well as the client profile and from there we are checking to see if this client

profile is allowed to see this page.

Custom Types

Pseudo introduces new custom types that make coding easier are much easier to understand as well as accurate. One of those types is the decimal type, which is meant to resolve the issue with float accuracy. This is done by creating two of *x* size, with one being the whole number and the other being the decimal number.

Example

```
# Create a decimal number of 32 + 32 bits in size
token_balance: d32 = 123.456
big_numba: i64 = 1,234,456
```

Types are declared in the same way they are in Python (yes you can declare types in Python, but it only helps the intellisense) using the colon (:) symbol. I also showed that you can use the comma to separate large numbers to make them more readable.

New Keywords

When

The **when** keyword is meant to be used for asynchronous process (which are common for UndChain since everything will happen over the network). This provides a structure that makes this easier to read and write when you are waiting on a specific event. Plus it eliminates the need to poll a resource all the time to wait on a specific value.

Example

```
from pages import partner

READY: bool = 1
NICE: i16 = 69
partner_status: listener = partner.ready

when partner_status == READY:
    some_method(NICE)
```

This is not the best example since you would normally say `when partner.ready == READY`, but I also wanted to show off the listener type. The when keyword operates much like an if statement with the exception being it only runs when that event has happened. I will expand this so that you can use this for parallelization in the future rather than just an asynchronous tasks.

Case Insensitive

At this time it is our intent to make the language case insensitive, meaning that the keyword `when` and `WHEN` are the same. This is also true for variable names like `apple` and `Apple`. I don't believe this will cause issue other than coding style debates which will always be a problem.

Decentralization

Ensuring that the network stays decentralized and ensuring that there is no concentration of resources going to one area is a key factor with UndChain. With that in mind the chain will envelop protocols that are designed to keep with those philosophies. One such method is the separation of powers between the different user types. With that in mind

this section will discuss how these separation of powers come into play regarding the network.

Creators

While Creators do create the rules for the chain, they are bound by the fact that they cannot enforce them as this is a role that is split between Clients, Validators and Partners. It is very possible that a Creator could release an update and if the Validators and partners choose not to run observe it, the new version would simply be skipped. This was done to stop potentially bad code from running on the network and to keep Creators honest in their updates. This doesn't mean they cannot spin up a new chain with this new code, but it does ensure that the community agrees to the terms outlined in those changes.

Validators

Validators are only as authoritative as clients allow them to be, if a group of validators are acting within self interest clients can simply ignore them and drop their Reliability scores and since the network is only interested in staying online it will switch over to validators that have that higher Reliability score. Validators can also be blocked by a Creator through various means (primarily through setting the minimum threshold on the Reliability score), but a Creator could block a particular Creator from participating. Lastly validators are graded based on the connections they make and if partners are not contacting them this lowers their Reliability score (as it does with clients) and they will be taken offline. There is a small exception to this with the use of known validators, but there is a threshold to which even a known validator will lose their spot if the Reliability score drops.

Partners

What use is a partner that is never used? If a partner is not being utilized then they are just burning electricity for no tokens and just like validators partners can loose Reliability score if they are providing no utility. Unlike partners, a Creator cannot remove them from the chain from providing utility (I don't think its necessary). Partners can drop in Reliability if they claim to perform utility and either cannot perform the utility as promised or doesn't do it at all. The validator has the choice to not provide a partner with work *only if* they have a proven track of being unreliable.

Clients

It is illegal on UndChain to not process a transaction regardless of how poorly a user acts on chain. If you make a request and you meet the fee then the transaction is performed. So, what UndChain does in cases where a client who has a low Reliability score, is in cases where you are submitting a claim against another user it reduces the likelihood you will win that claim without substantial proof. How do clients have low Reliability scores? Mainly through false claims, but also when they act in either of the other user types. Think that you are a partner who has earned tokens and then you want to use network resources with those tokens, but you provided poor service. As always, you can redeem yourself by acting as a partner or validator.

Latency / Cost

While a balance of power is key that doesn't completely ensure decentralization, so another method we use is latency. Along with affecting your Reliability score, having a low response time (or cost) makes you more preferred on the network and in the case of partners, gives higher rewards. This encourages the network to be more spread out since if you are closer to the client (assuming the same hardware and networking) then it will guarantee you win the latency game.

Remember that latency is how fast you can respond to a clients request. You could be right next door to your client using LoFi, but if it takes you 10ms to process a request that another partner a state away can process in 1ms then you will loose.

Pool-less Protocol

While having a separation of power between the validators who maintain the blockchain and the partners who hold the ledger and process the payments. There is another reason the validators exist, and that is to keep mining pools away from UndChain. There is no need for them; the Validators already perform token distributions and traffic control so there is no benefit for another entity to perform that function. This also gives the partners full rewards since validators don't earn SP.

Wallets

Had a thought while doing some development work in regards to having hot and cold wallets, and since UndChain needs the ability to sign transactions (fairly regularly) I think it may be wise to implement the idea of hot and cold wallets so that users already have that as a additional means of security. What this would mean is that instead of just one public address in a user accounts name you would have two. I need to think more on this and how it should be implemented. I will say for sure that fees between those two wallets will be zero.

Code Bounties

In this section I am going to list out the active *bounties* for the UndChain project. The Bounty program is meant to not only list what is needed for chain development, but also provide reward structures for each *contract* that is listed.

One of the ideas I have for UndChain is to make it so that open source software doesn't mean free, which means for everyone not just the project. This not only refers to the organization which can receive tokens, nor the Creator(s) who can collect fees, but also freelance developers who are looking to assist in a project.

UndChain itself has a startup developer fund of 1 million GP as well as 10 million SP. UndChain will also use this fund to support new co-chains since new blockchains coming into UndChain help support the eco system and make UndChain better.

Currently there are no active bounties at this time as the system is still under development, but I am going to make some examples of what a warrant should look like. *Note: this is currently being posted in GitHub, but will eventually only exist on UndChain once the Code Ledger side chain goes live*

Example of Bounties

Example of a Co-Chain Contract

Wanted: Need to develop a co-chain that allows users to lease high end hardware with low latency to a users for use in gaming environments. The name of this chain is **Player2** and the goal is to allow thin clients the ability to stream content and extremely low latency rates (>10ms) via the access protocol.

Status - Open

Payout: 10,000GP

Participants: @Undline(0)

Total Contribution: 0 - Never

Breadcrumb - Need to outline the technical specifications as well as how this can work.

Timeline: Technical outline due out by August 2025

Explanation

While UndChain is the only chain that should be requesting new chains (although anyone can do that). I believe this shows a good example of what a contract should looks like.

1. **Wanted** - This defines what is being sought after and should be treated as an overview, meaning what the final product should be. *I should probably limit this field to a specific amount of characters to enforce the idea that this should be a brief description.*
2. **Status** - The current status of the contract. Open means you are looking for candidates, active means that the module is being worked on and is not open for additional candidates at this time (this should only be reserved for large projects that may have too many participants) and closed which means the contract was completed.
3. **Payout** - As the name suggests, it's the total payout from the contract. This payout will be subdivided across all participants on the team with an emphasis on usage. Meaning how often a contribution is used or referenced. *payout is going to be the trickiest portion of the algorithm to implement since people will always try to game the system to get more.*
4. **Participant** - This is the account(s) / organization that is currently registered to work on this contact. There is also a listing of how much of the project (in percent) that has been contributed by each individual.

5. **Total Contribution** - Refers to how many commits have been made to the project along with the last date. This gives any prospective developer an idea on how active a project is.
6. **Breadcrumb** - This section gives the developer an idea on what is the primary focus at this time. I think this would only be needed for building Co-Chains, but it's possible this may be helpful in smaller projects as well.
7. **Timeline** - This states where the project is at in it's development stage as well as target dates. ideally displayed in something like a Gantt chart.

I believe there will need to be more things that need to be added so this needs to be open for extension...

Example of a Improvement Contract

Wanted - Add documentation to the readme file that describes what a warrant / contract looks like.

Status: Active

Payout - 4SP

Participant - @Undline(100)

Total Contribution - @Undline(20) - 2024-06-23

Breadcrumb - Describing how the Warrant system works for improvements

Timeline - 2024-04-04

Explanation

The definitions are the same from before, but this shows two things. First, what it should look like when you have some contribution going on and second, that it doesn't have to be *code* that is being worked on. Ideally anything that can be represented digitally could be used here. In this case we are requesting that better documentation exist.

Example of a module update

Wanted - Need to implement a LRC caching system for faster response times when a client requesting pages that are in high demand.

Status - Open

Payout - 2GP

Participant - None

Total Contribution - None

Breadcrumb - Start

Location - data_strcutures.py

Timeline - 2024-07-09

Explanation

This is a quick example of what it would look like to have a contract open on a specific module. In this example I added a new category called location. I am not sure if the other contacts would need it so I am going to leave it as `none` in those files.

Example of an Improvement Contract

Wanted - Need to add functionality that allows multiple clients to stream updates to one another over a singular document. The largest challenge will be managing conflicts. Mutex certain portions of the document?

Status - Open

Payout - 100GP

Participants - None

Total Contributions - None

Breadcrumb - Define project objectives

Location - New

Timeline - None

This example is meant to show how you could create a contract that adds functionality to an existing project.

Example of a Hardware Contract

Wanted - Need to create a wireless device that can use multiple radio frequencies that broadcast UndChain routing protocols that enable fast local communication.

Status - Open

Payout - 100,000GP, 40,000SP

Participants - None

Total Contributions - None

Breadcrumb - Define operating parameters to stay compliant with local laws. Use LoRA and WiFi as a starting point?

Location - Altera

Timeline - None

Not sure if this will work out well, but I was thinking we could also make real world projects using this same open system. In this case the location could be to use a program (I used Altera as an example since that's where you would make PCBs).

Conclusions

This should all be housed inside programs that have a form of revision control so that you can maintain and control projects. Ideally you would have an *organization* (in UndChain) responsible for creating these projects that way you can define roles and positions that you need. *Tip: never skip out on Documentation or Quality Control as it is critical for any successful project...*

Technical Documentation / Roadmap

Code Guidelines

Any submitted code should have type definitions as well as doc strings to accompany any new methods.

Roadmap

Follow this link to see the internal roadmap for UndChain to get more in depth knowledge on how the technology works Internal UndChain Roadmap

We are currently in Phase 1