

# Transcript of "LENS & LEGO of WBE3 APPLICATIONS"

Slides: <https://magickbase.com/presentations/lens-and-lego-of-web3-applications/slides.pdf>

---

Hello everyone! I am CHEN YU from the Magickbase team, and the topic I prepared today is Lens and Lego of Web3 Applications, how should we shine the off-chain tools?

As new things upcoming in the ecosystem, the issues that off-chain tools need to handle are becoming more numerous and important.

Today, I'll start with some challenges of the blockchain ecosystem and explore the core values it brings to Web2.

Finally, I'll present a guideline for how to design tools focused on privacy and automation.

## **Currently, the ecosystem faces 3 challenges and conflicts**

The first one is about "**decentralization and fragmentation**"

As we all know decentralization is one of the principles of blockchain.

It aims to eliminate reliance on central services while ensuring data transparency and security.

However, as blockchain evolves, information and data are scattered across multiple chains and platforms, making them "insulated islands". This fragmentation leads to "local centralization" with individuals who cannot migrate themselves across chains easily.

There are 3 reasons for this: first, cross-chain technology is not ready for broad usage, it's not secure enough, it's not stable enough. second, projects tend to develop independently, each creating its own ecosystem and hard to work with each other; third, we had pursued decentralization too far away, that we almost placed everything on the chain, and it's hard to migrate them anymore.

The second problem is the **conflict between transparency and privacy**

Blockchain's design allows every transaction to be publicly verified, ensuring data authenticity and security, however, users' privacy is also being exposed. Specifically, details like value and address are accessible on-chain, allowing anyone to easily track a given address.

It makes blockchain hard to adopt in privacy-sensitive areas like healthcare and finance.

The third one, **user sovereignty, and the learning curve**

There is a saying that "not your keys, not your coins", highlighting users' absolute control over the data. No one can access them without the private key. But, how to keep the key secure becomes a huge challenge to general users.

To be honest, blockchain is not inherently user-friendly for newcomers. They are all facing a steep learning curve, they have to understand the importance of private keys, and how to store them securely. It's much more complex than web2 applications. Mistakes are usually irreversible, preventing users from embracing the blockchain's freedom.

So, how can we handle these?

There's also a saying that "All problems in Computer Science can be solved by another level of indirection"

When designing web3 products, we can introduce layers to simplify the usage and **decouple data, values, and actions**.

**Let's rethink the values of blockchain that empower a web2 app to be a web3 dapp**

1. The first one is **Storage of value**: Blockchain makes data secure and immutable by consensus and decentralization. It offers users a choice of storing value that is free from the control of central services.
2. It can also be considered as the **Medium of exchange**: Through the blockchain, users can exchange their data and value without relying on central services.

3. and, Blockchain entitles users to a **unified identity across various applications** without centralized authentication.

**Correspondingly, we can sort them into 3 abstract roles**

### **Value-centric abstraction**

When we talk about value-centric we are talking about the intrinsic value that data delivers, rather than focusing on data itself.

It shifts the emphasis **from “what” the data is to “why” it’s valuable**, aiming to highlight the core utility or worth that the user gains from the data.

For example, when using a blockchain-based voting system, we are expected to immediately enjoy the benefits of trust and transparency instead of the binary code revealed to the public.

We should focus on the **insight** data can deliver, and why these data are grouped and adopted, not how these data are organized or executed. The practical insight matters, not the process behind it.

Besides, intent-centric interactions should be encouraged. Usually, users don't know what to do and how to do it, and sometimes they even do it in the wrong way. we should encourage them to express what they want, like. the potential returns they want in DEFI, not which algorithm or parameters will lead to the returns. It would clarify the value users receive and reduce the learning curve.

A step further, we can get **Intent-centric abstraction**

intent-centric abstraction is a design paradigm that prioritizes users' desired outcomes (intentions) over the underlying technical processes.

It shifts the focus from **how an action is performed to the result it brings**. They don't need to know how blockchain works, they simply specify their desired **outcome or intent**, while the system manages the steps to it. For instance, if a user wants to transfer tokens, they don't need to understand the fees, and transaction paths. The system abstracts away all details, handling them in the background, so the user simply enjoys the outcome.

And we should guide users rather than educate users. Users are not professionals, it's unnecessary for them to understand the underlying technology, parameters, and configurations. Instead of teaching users how to use the tool, ask them for what they want.

Of course, we should double-check if the results align with their expectation. That's the **intent verification**.

### **The last abstraction is the Identity-centric abstraction**

It centers on the user's identity, as well as the unique traits and permissions that define it.

Unique traits are the core of identity to recognize each user's **distinctiveness**. And permissions are also important because some of the automation, or to say, interoperability, should be enabled by authorization.

No matter how identity is managed, privacy without compromising security is crucial. User information is sensitive and we should try to avoid revealing it when verifying identity.

Very brilliant works have been done by our partners, and we can more or less find the pattern of these 3 abstractions here

1. The RGB++ protocol, separates the value from the concrete, so the values can be migrated between different chains, regardless of architectures and specifications, that's the value-centric abstraction
2. Fiber Network, it compresses the intents by the channel and delivers the final outcome onto the chain. Obviously the intent-centric abstraction
3. JOY ID is a very typical account abstraction, chain-agnostic, private-key-less, and recoverable in multiple ways, nice identity-centric abstraction

Let's take a step back and look at the commonalities among them.

We can find A very important point. all of them **placed the value and user sovereignty on the chain while shifting data and computation off-chain**.

**This is the new perspective, moving from on-chain solutions to off-chain evolution.**

We will apply these insights to upgrade one of our major products, the CKB Explorer, into a comprehensive Web3 platform.

Compared to the current CKB explorer, the new platform will support asset management across multiple chains, with each asset being represented as a model named meta asset.

The **meta-asset** is defined in 3 aspects, identity, measurement, and interface

1. each meta-asset includes an **identity** that helps us locate it regardless of its chain or structure;
2. for the **measurement**. the diversity of asset type and structure often leads to inconsistency, with wide variation in data formats and measurement methods. To simplify this, we will define a standard measurement that unifies the characteristics and values of different types of assets. The measurement serves as a "universal descriptor", interpreting the complexities into qualitative and quantitative value, providing a common foundation for comparison and analysis.
3. **interface**: also, a standardized interface for the meta asset is a must-have, only with the universal interface, assets can be combined or pipelined in a general way.

The meta asset will be modeled in these 3 dimensions, and work for the next feature, local AI agent pool.

Nowadays, using AI agents to build automated tasks has become a common option. Especially when we are handling intents, the AI deduction would help a lot.

We will build this function like this:

1. **Local Agent**: we will make the agent local-first, so that it can work directly with the node, so users' privacy will be properly protected and the reliance will be guaranteed because no remote services are required.
2. Then we will add the **deduction module**, each local agent with the deduction can transform users' instruction into meta asset. This approach allows users to focus on the goal without worrying about the underlying details.
3. **Interface** again, only with the universal interface, the agent can be pipelined together to work out complex tasks.

When we have plenty of agents, we will make a space for them to connect and distribute. That is the **distributable workflows**.

I'm not sure if you have tried the shortcut app of Apple. The shortcuts, designed by the official or individuals, can be downloaded anywhere and chained locally to perform a series of tasks. Each shortcut only works for one task but with the logical condition connections, it will be a streamline.

We are going to do the same thing but for activities on the blockchain.

1. as mentioned, the agents conform to the universal interfaces so they can be connected as a workflow
2. and the platform will provide a **space for workflow distribution**, so they can be shared or subscribed to easily. It's much like the app store of iOS and Android, fostering the community to grow.
3. also, we will provide a **space for workflow delegation**, it would be the most powerful feature of the platform. Users can delegate the platform to run workflows, enabling real-time data monitoring and automated response. This functionality is especially suited for scenarios that require frequent updates or decision-making based on data changes.

When I introduced the intent-centric abstraction, I said "Guiding users rather than educating users" Users may manipulate the tools incorrectly, but they can always tell if the result is right.

We will try it in a heuristic way.

On the platform, a very **intuitive entry will be highlighted as a chatbox**, everybody is good at using the chatbox, right? No click here, click there, just say what you want. With the help of natural language processing, the requirements will be extracted from the instructions and suggestions will be respond. Users can select a suggestion and the system will continue to guide users until they reach the ultimate goal. Once users find the result that matches, the proper workflow can be recommended.

Let's see a case, the task list for airdrop

traditionally, a KOL would first gather and organize a series of tasks for the airdrop, then publish the task list on social media. Followers, once seeing the list, would have to go through each task step by step, often leaving questions when they have difficulties. This leads to high communication costs for the KOL, who has to respond to numerous questions.

With the platform, how will it be? this process can be automated. Using the natural language entry, the KOL can quickly transform these tasks into a workflow, and share it on the platform, followers can directly copy the workflow and run it with a click. This eliminates the ambiguity of language. And The entire process becomes simple and intuitive.

Here is the Recap

Today, I started with the three challenges in the blockchain ecosystem, analyzed the values and roles of blockchain in Web3 dapps, and proposed a design methodology. I also introduced how we use the methodology to upgrade our CKB Explorer.