Universal Decentralized Asset Protocol(UDAP)

A White Paper

Version 0.7.1, Draft

UDAP Foundation， 3/7/2018

Executive Summary

UDAP is a blockchain-based asset protocol that provides Restful APIs and a Universal Asset Wallet for third party application developers to create powerful blockchain based application efficiently to deal with crypto-token encoded real-world assets, without the steep learning curve and concerns of scalability, privacy and security.

UDAP uses a Multi-Chain architecture that supports both "Virtual Private Chains" and physical application zones, where each applications runs on its own private secure blockchain and storage, with user configured privacy level and blockchain nodes. UDAP uses both vertical sharding and horizontal sharding to achieve potentially millions of transactions per second. Massive parallelism is realized in the nodes where all the cores can process all the transactions in parallel. Multiple nodes form processing zones and zones form zone clusters to create enormous transaction processing power.

Multiple levels of encryptions and obfuscations are used to ensure of the privacy of the assets registered on the chains.

A Universal Asset Wallet is provided for application developers to write exciting applications directly in the wallet as plug-ins.

Applications can immediately register token-based assets on chain and offer the built-in marketplace to their customers. Applications can even create their own in-app currencies backed by blockchains and use it to build the in-app economy. The app-specific currencies can eventually be traded in the UDAP platform to offer unlimited liquidity.

UDAP is a standalone public chain cloud that offers itself as an advanced streamlined BAAS for the booming real world token-economy beyond crypto-currencies.

Contents

1. Background

In the beginning, there was Bitcoin.

Although touted by its inventor as "digital cash", Bitcoin has been pushed to play pivotal roles beyond cryptocurrency. Many projects have created protocols on top of Bitcoin to facilitate the exchange of general assets. Omni Layer(formly Mastercoin)[1]Counterparty[2]Prism[3]

But because Bitcoin has been first and foremost designed to work as a fungible currency, using it beyond as such has proved to be clunky and limited.

Ethereum (the E, for short reference) has emerged from the crowed to carry its inventor's destiny, initially as "programmable money", then as a general blockchain based "world computer" usable for financial and non-financial applications. A general-purposed smart contract architecture lends its well to building a variety of on-chain applications to fulfill blockchain's promise of being the backbone of the "Internet of Values".

But the most challenging thing posed to Ethereum is how to represent the "values". For the moment, like almost all the other open ledger platforms, Ethereum is exclusively handling digital currency transfers, as Buterin put in one of his podcast in early 2018:

"The very first vision was basically a general purpose platform for financial contracts. If X happens then send $5 to account Y, if Z happens send $5 to account B. That was basically what I thought Ethereum would be for"[4]

Ethereuem's model of "value" is primitive. In the basic model, for any kind of value/asset the only attribute associated with this asset is the balance of it, a number. This simplified model probably will fit financial transactions well, but it's very lacking dealing with variety of asset types in the operation of the world economy.

The three-year-old smart contract implementation in Ethereum, although has attracted thousands of development teams to deploy applications, mostly in Solidity, on the platform, is far from being mature and productive towards becoming the world computer:

- Limited in features and functions. For instances:
  - Limited parameter types.
  - Lack advanced language features that JavaScript, Python, Java can offer.
  - Primitive collection processing and manipulations.
- Slow performance. The EVM is not really a modern virtual machine like JVM or V8 JavaScript engine. It's an interpreter that parse the smart contract opcode and run the underlying supporting libraries. The EVM is at least an order slower than regular system programming languages. The total throughput of the blockchain is directly impacted by the slow performance of the virtual machine. There are many reasons that faster and mature virtual machines cannot be used in the current version of Ethereum. And there are quite a few efforts in development to enhance the performance of the virtual machine.
- The general purpose computing power in the current version of EVM, on the other hand, has rendered itself vulnerable to many security breaches, as documented by Making Smart Contract Smarter[5]. It's very hard for relatively inexperienced programmers to get the Smart contract right. Considering many of the smart contract are handling many millions of dollars of assets, customers take a huge risk in moving forward with a smart contract strategy.
- Smart contract makes

If we can draw some analogies between web application development and blockchain application development, smart contract is to blockchain development what CGI is to web app development. It's rather young.

The limitations of the current mainstream Smart contract platforms have become a serious impediment to peoples ever-growing interest in tokenization of everything. Tokenization has become the symbol of The new economy, short named as token economy. In this new economy enabled by block team technologies, every bit of value is represented by a crypto-graphical token. With the security feature enabled by blockchain technologies, tokens have basically transformed the asset we deal with every day into capital, characterized by its high liquidity. Tokens make the people who own assets to realize the economic returns that those assets are capable of, in the meantime, tokens will make those people who are in need of capital to get the capital in a most feasible way. Tokens are basically the gasoline of the new economy. If there is one thing that will happen in the next few year for the Internet industry, this "one thing" is to "tokenize everything".

- Crypto tokens represent (or be pegged to) shares of right, access to services, voting power, real world financial assets, etc.
- Tokens serve as accounting units in bookkeeping and payments.
- Tokens eliminate the requirement of intermediaries in many trading scenarios thus simplify and expedite the trading process at very low cost. Fungible tokens are very easy to trade en mass, while none-fungible tokens can help to track the asset flow in an economy thus to help people understand the dynamics of the economy.
- In short, crypto-tokens are the private money for applications.

Tokens have been used in crowdfunding a new wave of innovative applications based on blockchain and related technologies. This has disrupted the startup model based on VC funding. On one side startups can receive funding much earlier in their product development. On the other side, everyone can invest in projects they believe in and gain the potential for investment return which used to the privilege of so called "accredited investors".

Tokens are the tickets to the next Internet banquet.

UDAP has been designed to support tokenizing everything movement natively, in the low-level storage model and communications. We are shifting away from the lump sum description of values into individually identifying each every and every items in the physical world in virtual world alike. On top of the identifications, we are going to beauty brand-new user experience to deal with their personal properties intuitively, to sell them, to trade them, to pledge them for capital, to lead them to friends, or to give them away. In the back, UDAP is a protocol for any third parties to port their existing applications easily and quickly to open ledger technologies and a decentralized storage Technologies. From the back end to the user experience end, UDAP has been designed to as a enabling technology to fulfill the promise of the Internet of values.

Blockchain technology is nicknamed the Internet of the moment However, when we look closely at the carrier of value transfer currently carried by the blockchain, 99% of the applications are actually limited to the best liquidity and the most easily solved type of asset, which is cryptocurrency. So the current blockchain is more accurately "The Internet of Money"[6]

But our neighborhood faces massive, palpable and non-palpable assets. Are these good blockchain, or related decentralized, bookkeeping techniques, good value, discovery, value recognition, and value transfer? There is no doubt that the answer is yes. There have been quite a few blockchain-based technologies that are attempting to address real-world non-monetary assets and how to decentralize their networks in recognition of the exchange of recognition. However, we find these protocol offerings and platforms based on the existing blockchain, the second-level structure based on currency transfers. Around 2014, when Bitcoin technology was recognized by the majority, many projects proposed the notion of a colored currency, one that is grafted onto the currency of permutable value - Unique identification technology that allows us to use currency symbols to represent real-life accessible and non-accessible assets.

2. The Goals

Our ultimate goal is to build the Internet of Assets. Its value proposition has a clear boundary, and includes the following concrete deliverables:

1. Develop an asset protocol through understanding the "assets" in our world from a blockchain perspective, and create abstraction of the general behaviors of assets; Define convenient APIs for traditional vertical applications to integrate with blockchains, which do not require application developers to have a deep understanding of blockchain and decentralized technologies.

2. Provide a reference implementation of UDAP protocol, and deploy a permissionless asset blockchain so that app vendors can catch the express train of decentralized computing and decentralized storage without changing the existing application architecture. We abstract the most central part of decentralized computing, decentralized storage technology and decentralized messaging mechanism into a concise and effective API. Considering that the decentralized computing and storage technologies are still in an immature phase and are in a period of rapid growth, we believe such technology platforms will bring tremendous value to third-party application developers.

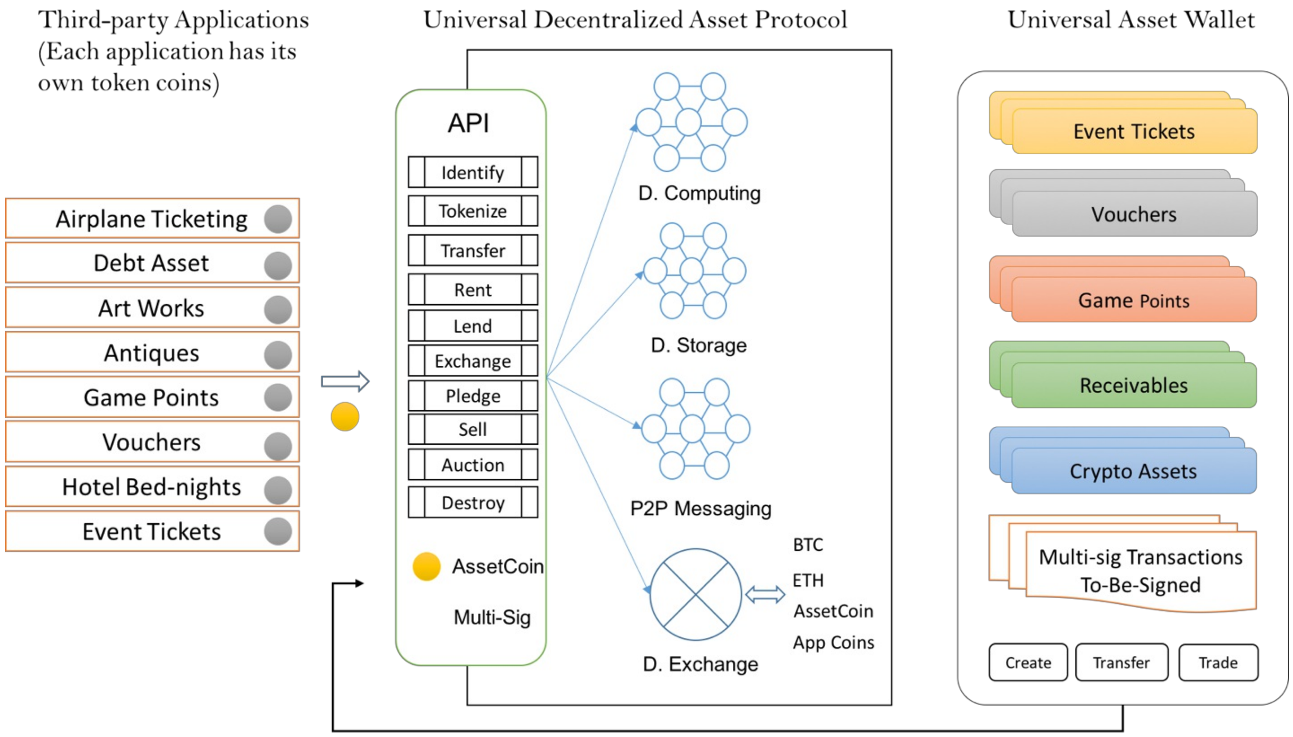
3. Implement a Universal Asset Wallet(UAW) for end-users. The UAW will capture the most useful interaction patterns with assets that allows users to store, copy, transfer, sell/buy, and trade a variety of assets deposit from third party applications in one place. Such a design for third-party application development, in fact, is a very favorable news, because the third-party platforms can focus on the current core business logic without having to figure out how to build their own assets blockchain and user Interaction.

4. Engage cloud service partners to offer Asset Blockchain as a Service (ABaaS) so that organizations can easily create their own private or consortium asset chains that can connect to the UDAP public chain for value exchange.

5. Build a global C2C marketplace to enable asset trading and exchange without intermediaries.

6. Support business to transform to token economy as a technology enabler through

- a highly scalable architecture that allows for linear scalability and supports thousands of applications and near one-second response with finality, with a total throughput of 10k~100k TPS.
- an end-to-end privacy protection mechanism that may handle highly sensitive asset information for their customers.
- a unique security model to reduce attack surface.
- a multi-chain architecture with inter-blockchain communication support that bridges permissioned blockchains and public blockchains.


Third-party Applications
(Each application has its own token coins)
Universal Decentralized Asset Protocol
Universal Asset Wallet
Airplane Ticketing
Debt Asset
Art Works
Antiques
Game Points
Vouchers
Hotel Bed-nights
Event Tickets
API
Identify
Tokenize
Transfer
Rent
Lend
Exchange
Pledge
Sell
Auction
Destroy
AssetCoin
Multi-Sig
D. Computing
D. Storage
P2P Messaging
D. Exchange
BTC
ETH
AssetCoin
App Coins
Event Tickets
Vouchers
Game Points
Receivables
Crypto Assets
Multi-sig Transactions To-Be-Signed
Create
Transfer
Trade

3. Design Principles

The basic idea of UDAP is to build a "thick protocol layer," supporting "Thick Protocol + Thin Application." pattern generally formed from blockchain application development practices.

The main concern at the application level is the interaction with the particular vertical customer base. Many of the common asset disposal aspects are handled by the universal wallet. One of the visions of UDAP is that application developers need only have enough knowledge and understanding of their vertical industry, then based on our thick protocol layer they can quickly develop the features asked by their customers, thus they would take the shortest time to the market.

In comparison to Ethereum, which is our starting point, the biggest difference between UDAP design philosophy and Ethereum’s is about "Features." Vitalik Buterin believes that one of Ethereum's design ideas is that Ethereum is a "Feature-less" computing platform.

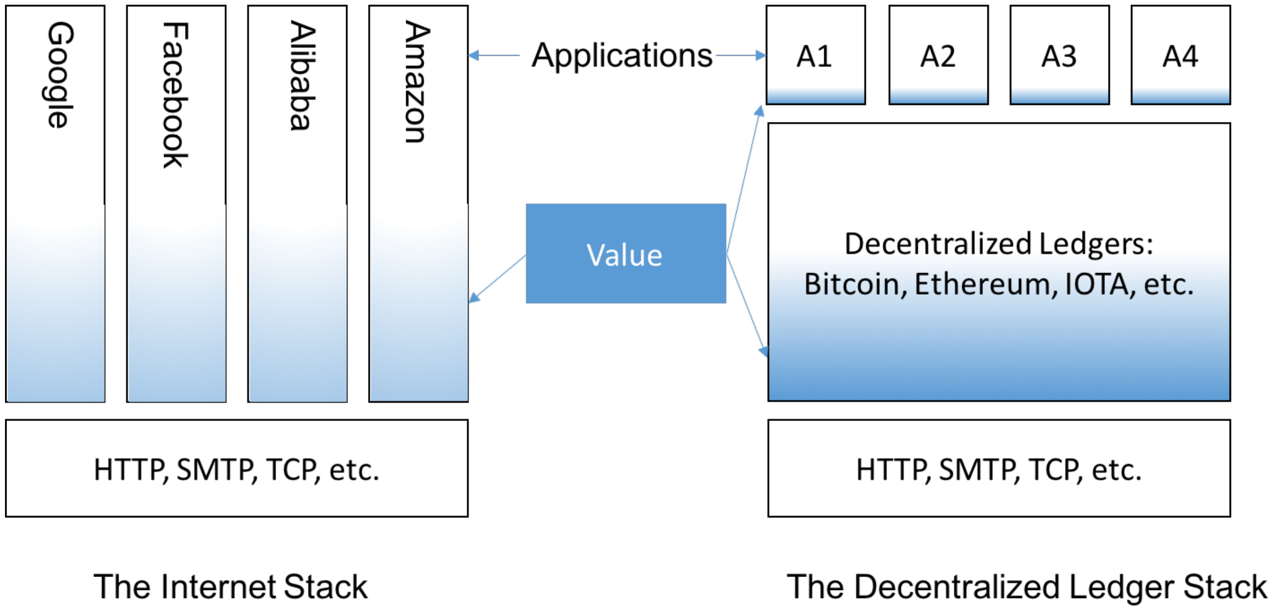
We Have No Features: as a corollary to generalization, we often refuse to build in even very common high-level use cases as intrinsic parts of the protocol, with the understanding that if people really want to do it they can always create a sub-protocol (e.g., ether-backed subcurrency, bitcoin/litecoin/dogecoin sidechain, etc) inside of a contract. [7]

Ethereum clearly has the main goal of building a blockchain platform for general financial computing. Any high-level abstraction is pushed up to user-defined smart contracts. The rational and restrained approach is agreeable, but it is not the "Asset Internet" protocol we have conceived and the direction we sought to pursue. Smart contracts offer dApp developers relatively great flexibility because of their "Turing Complete" computing power, but they also bear obvious technical obstacles and real risks. Although Smart Contract "Turing Complete", but provide programming ability is still very limited, for ordinary application developers have a lot of "holes" they need to patch. In the meantime, because of all the code being exposed, user-defined smart contracts have exposed a large attack surface, as evidenced by several incidents that have caused serious economic losses in the past two years in Ethereum based dApps. Bitcoin networks have very limited smart contract capabilities. However, many loyal fans of Bitcoin networks see the limited design as a very good way to prevent bitcoin networks from being subjected to attacks. A very important indicator of the bitcoin network robustness is that bitcoin networks have endured countless attacks for the last eight yeas, yet essentially no major failure has happened in the protocol level.

"Asset Internet" is not a universal computing platform by design, but a platform that handles general assets encoded and protected by blockchain technology. It is a "Featured Chain." From a protocol perspective, it is a thicker and more specialized protocol that natively supports the tokenization of assets and general asset behavior such as storage, transfer, trading, sales, etc. We are delivering a robust, secure, and high-performing asset management and trading platform for everyone. We will not be supporting custom smart contracts (or at least not exposing the capabilities of smart contracts to end users), for safety and performance reasons.

We think this "Thick Protocol + Thin Application" model will be the core architecture model for blockchain applications and will greatly boost the new applications of blockchain and Internet in general. This model is technically innovative and value-conscious, allowing the value of the Internet to sink from the application layer to the protocol layer, turning the protocol layer into a cross-application big data repository. The result is that the protocol layer becomes more strategic and investment value.

The Value Distribution in the “Classic” Internet and “Blockchain” Internet



UDAP offers a selected set of features that simplify blockchain application development, without sacrificing performance and security. Particular UDAP offers API in the following area:

- Identity Management: multi-persona identities are supported.
- Multiple Signature are used across all the major functions, so proper authorization can be set up to authenticate users to interact with asset in a secure way. Multi-Sig can be applied to identity management, asset registration and asset transfers.
- UDAP offers a set of rich abstractions of asset operations, including transfers, sell/buy, and trading.
- The Universal Asset Wallet is considered a significant part of UDAP package. It must offer generalized asset operations across applications such that the third party apps do not have to implement the common user interactions. In the meantime, it must offer a pluggable architecture such that asset interactions can be customized and enhanced if the apps choose to do so.
- UDAP offers asset market and exchange API and implementation in the wallet. Once registered, assets can acquire liquidity immediately without relying on other space to provide liquidity.
- UDAP does not provide built-in people and assets identifications that are detached from applications. All logics are defined by applications. Even those that look like built-in features are actually applications that are built on the UDAP API.
- Sovereignty for each application.

4. Value Proposition

1. Unbreakable database capabilities brought by blockchain. This kind of database capability is what the traditional database system was dreaming of in the past and could be very expensive to set up and manage, because it involves very complex database synchronization, backup, disaster recovery, and other key technologies. In the past, these abilities required the attention of senior DBAs and network engineers to get it right.

2. Easier integration solutions for business partners. The second type of value is captured when the application of the blockchain is not limited to the users within its walled domain. Blockchain technology provides a secure, open ledger system that naturally applies to the upstream and downstream integration in business partners in an industry. The integration of upstream and downstream data in a vertical supply chain is a continuing technical challenge in the traditional applications. Integration solutions in the upstream and downstream process, due to the lack of trust between each other, or a more complex commercial agreement and a third party that both trust, or a leadership, are often outdated and have a high cost. Now the blockchain technology offers the ability to act as the trusted party.

3. Support of "Token Economy". Tokenization has never been easier and formalized by smart contract mechanism. Tokens can be used in in-app marketing, incentives, payment, and new models of distribution of benefits. However, using Ethereum's standard means of setting up your own economic currency requires the ability to program smarter contracts or hire a consulting team to do the job. UDAP enables users to distribute tokens that can be used in their economy simply by configuring a few standardized parameters. This values represented by the tokens immediately show up in the universal asset wallet we offer. Not only that, the UAW themselves provide the ability as a Decentralized Exchange that enables the trading of custom-issued assets. The immediate liquidity offered by UDAP and the Universal Asset Wallet greatly empower the applications.

4. Buffering and performance improvements. The Ethereum network is now capable of handling millions of applications a day, but this ability is still far behind that of traditional centralized applications such as VISA and SWIFT. Performance problems will be gradually resolved in the next 2-3 years[8]. For third-party applications that require high-throughput today, UDAP will provide this ability.

5. Identity management. Blockchain technology is based on the modern cryptography and, to a large extent, provides new security standards for third-party applications. It requires that all users behave in a secure manner. Each user, after their authentication, signs their own behavior with their own keys and assumes the corresponding responsibilities. Managing digital identity is itself a relatively complex process that requires a set of programming methodologies. Our asset management agreement brings a simple but effective package of identity management tools. Third-party applications use this third-party application based on guaranteed security to fully enjoy the convenience provided by blockchain encryption. Identity management, the identity of third-party in-app users, is often only relevant in the context of their own applications in the past world. After entering the blockchain event, the asset behavior of all users must be bound to one or more accounts in the blockchain.

6. Privacy. Asset Protocol will provide encryption for all types of assets managed by third party users. Users have complete control over the storage of their assets in the network and the information they control over their assets is open to specific populations or applications. Users can control their own asset transfer path, leaving it open, or completely private. Users and applications within each application are completely isolated from each other to ensure the privacy of users and their assets. Our universal asset wallet can import multiple third-party applications, each configured with a separated account.

7. Workflow support. UDAP has built-in support of multi-sig signing flow. Key asset actions, such as the transfer of assets, the exchange of assets and the sale of assets, often require the protection of multiple signatures. Applications can be very easy to set the key action signature protection, strategy and rules. Once a protected action is discovered or invoked, the user's generic asset wallet automatically prompts the user for the signature. In my own wallet, I can sign, or refuse, the processing of assets by multiple signatures of other people in real time. This type of workflow can be done not just between individual users holding Universal Asset Wallets, but also between the third-party application's interface and the Advocate Universal Wallets. This type of multipoint communication and communication is supported by the underlying P2P-based messaging protocol on the asset network.

8. Immediate liquidity. The protocol and implementation will provide the ability to trade similar assets and cross-border assets, especially to provide and mature cryptocurrency trading.

5. Asset Protocol

A protocol is a specification of and a normative guide to the exchange and communication of information between and within systems. UDAP's asset protocol regulates how asset are presented, stored, communicated and interacted on the blockchain, how authenticity of assets is verified, and how consensus is reached.

5.1. On-Chain Asset Model

Traditional ERP systems have established their asset models based on centralized storage and computation. The asset model is an abstraction of the assets created by their issuers, which governs how the systems manage the attributes, operations, and security of assets. Unlike traditional ERPs, UDAP's asset model focuses on an on-chain standardization of the description, interaction, security, privacy, and authenticity of assets.

> Why assets need to be tracked and managed on the blockchain? In general, Crypto assets have the following advantages over digital assets (in this case, traditional ERP-managed assets):

1. Clear ownership: The rights and interests of asset issuers, asset owners and transaction signers can all be clearly defined and cryptographically protected. Asset owners can easily provide irrefutable proof of their rights and interests; without the consent of the transaction signers, the assets can not be exchanged and traded on the blockchain; meanwhile, asset issuers have the rights to determine some of the basic attributes of the assets, for example, an asset issuer can restrict the transfer and trading of assets.

2. Information Permanence: Digital assets requires a permanent storage to manage their lifecycle. The advent of blockchain and decentralized storage finally gives us confidence that we can save information for a long time.

3. Anti-counterfeiting and anti-tampering: Once the assets are on the chain, the relevant data and transaction records can be effectively protected. Anti-counterfeiting and anti-tampering goals can be achieved, and thus moral hazard and financial risks can be reduced.

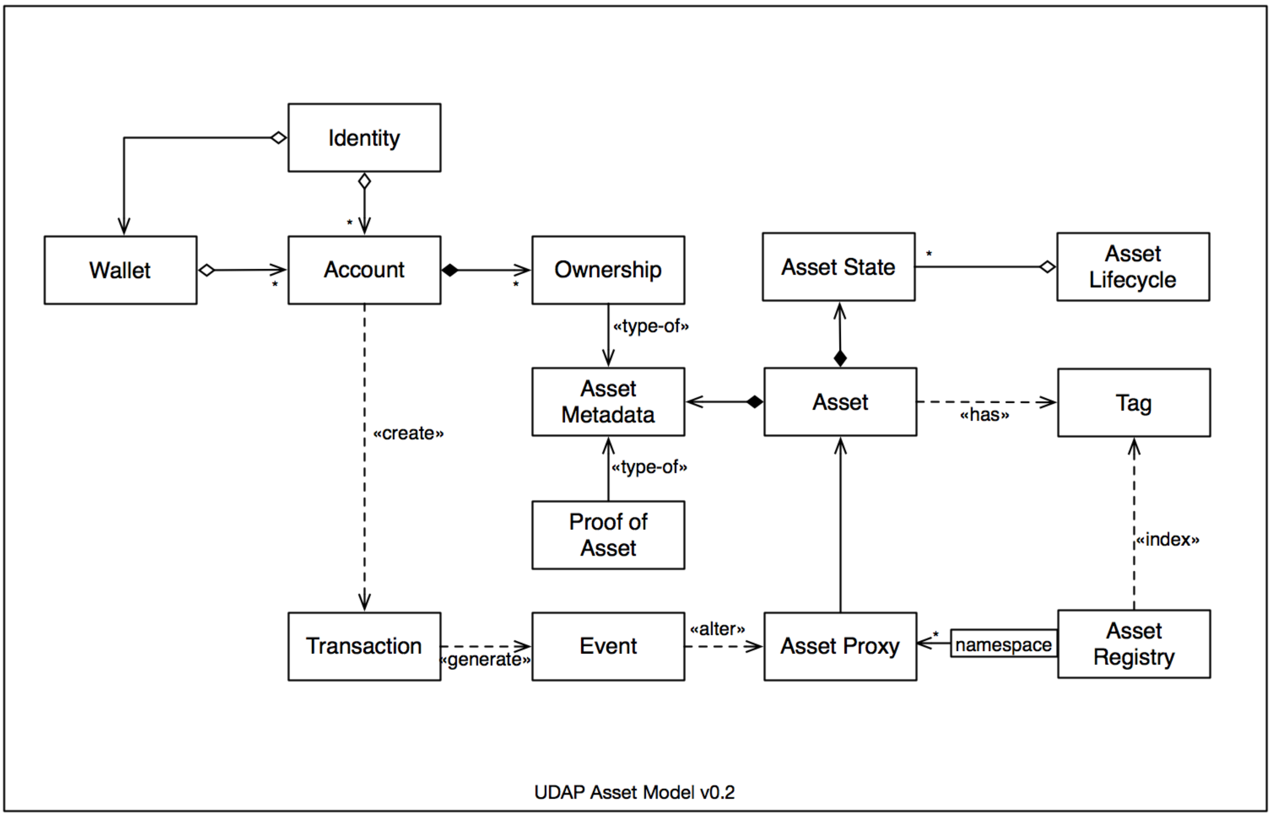
4. Liquidity Demand: The relationship between value and liquidity is inextricably correlated. Liquidity is the term used to describe how easy it is to convert assets to cash. The more liquid the assets are, the easier their values are to be recognized. So liquidity has a very important influence on asset's value. Money, as medium of exchange, has a very high liquidity. In facet, it is the most liquid asset compared to everything else. Assets with fair liquidity includes cash equivalents such as stocks, bonds and options. And assets such as houses, cars, or farms, factory equipment, etc., have relatively low liquidity and are difficult to value. Therefore, their market values may differ significantly. Traditionally, liquidity and value realization are implemented through intermediaries like eBay and Taobao, however, through tokenization blockchain has more potential for liquidity optimization.

In addition to the above requirements, privacy is also a mandatory need that crypto assets must meet.

Most of the so-called crypto assets in current blockchain world are aimed at a special kind of fungible assets, that is, crypto-currency. Blockchain-based application protocols or platforms are mainly to facilitate the creation, distribution and exchange of crypto assets. Protocols that govern crypto-currencies include ERC20 and some of its simple extensions. For example, one famous third-party trading protocol for fungible assets is 0x protocol. This protocol assumes that digital currencies have been fully distributed among different owners, and that the problem this protocol addresses is to become a decentralized digital asset trading venue, especially for transactions between fungible assets.

Various efforts have been put on standardization and specifications of non-fungible assets, such as ERC721 protocol, which was implemented in the popular CryptoKitties game and its various clones.

UDAP defines a conceptual model based on the analysis and abstraction of various real-world assets in combination with efforts and achievements by MediaChain[9], Digix[10], BankEx[11] and other blockchain projects[12][13]:



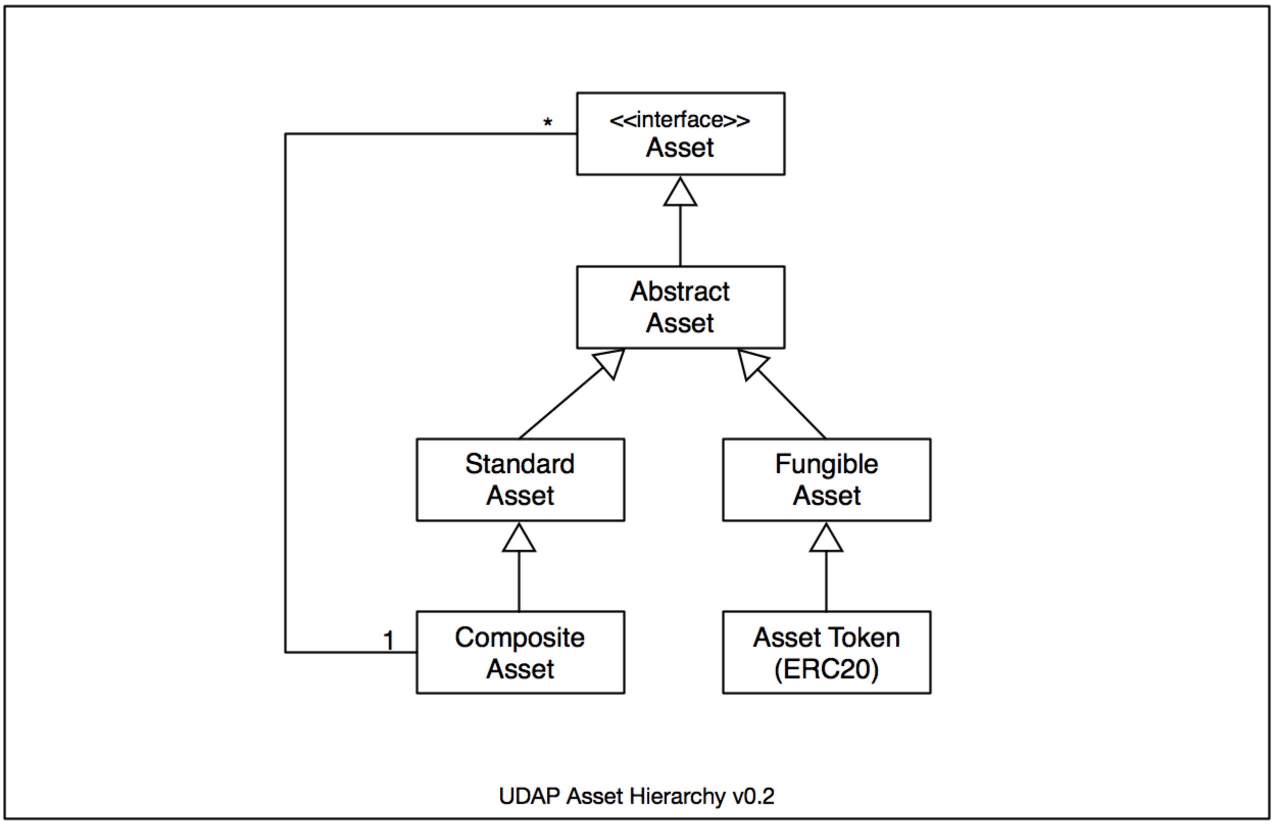
UDAP Asset Model v0.2

This on-chain asset model defines assets and related objects, as well as the relationships between these objects. This asset model is compatible with ERC721 but offers a richer set of attributes and operations as described below.

5.1.1 Asset

Anything that is capable of being owned or controlled to produce value, is considered asset. For example, goods, services, trademarks, securities, warehouse receipts, purchase agreements, licenses, copyrights, music, videos, games, loyalty program points, game equipment, event tickets, collectibles and other physical assets and digital assets. Currency (including cryptocurrencies) is also an asset. When you see "asset" in the white paper of a blockchain project, it most likely refers to cryptocurrency.

Assets have attributes. Some common attributes, for example, asset identifier, namespace, issuer, fungibility, transferability, etc., are determined by asset issuers and can not be modified after assets are issued. Other attributes, such as name, description, owner, and states, can be modified during the life cycle of assets. Based on the fungibility of assets, UDAP presents a hierarchical model as described in the following diagram, which defines a standard interface, an abstract type that provides basic attributes and operations, and multiple derived asset types.



UDAP Asset Hierarchy v0.2

The UDAP asset protocol itself does not specify any implementation details, however, to better describe the model Solidity is used to illustrate the components and their relationships. We may use a more implementation-neutral interface description language to describe the component model in a later version of the protocol.

```
contract Asset is ERC721 {
  event AssetCreated(address indexed _asset, uint indexed _id);
  event AssetTransferred(address indexed _to, uint indexed _id);
  event AssetDestroyed(uint indexed _assetId);

  function id() public view returns (uint);
  function issuer() public view returns (address);
  function owner() public view returns (address);
  function namespace() public view returns (bytes);
  function transferrable() public view returns (bool);
  function fungible() public view returns (bool);
  function metadataHash() public view returns (bytes);

  function transfer(address _to) public;
  function destroy() public;
}
```

The asset component described in the component model have the following attributes:

- Asset Issuer

  An asset issuer in UDAP refers to the address of a UDAP account that issues and registers assets on asset blockchains. This is a immutable attribute.

- Asset Owner

  An asset owner refers to the address of a UDAP account that owns crypto assets registered on the asset blockchains.

- Fungibility

Fungibility refers to interchangeability of assets with the same amount and of the same type. Based on asset fungibility, assets are usually classified into two categories: that can be replaced and that can not be replaced. In the asset model, the two types correspond to FungibleAsset and StandardAsset, respectively. An non-interchangeable asset means that although both individual assets have similar attributes and external behaviors, their possession is not replaceable for a particular owner because they have different identities. This is analogous to the fact that although both have iPhones, there is no substitute between an iPhone and another iPhone because each iPhone has its own unique attributes that make it disappear if the iPhone is switched. Each iPhone has its own unique phone number and a unique purchase time, so between the two iPhone they are not interchangeable with each other, at least in most cases.

In the physical world, there are a large number of assets that can not be easily interchanged. For example, most of the real estate properties can not be replaced during the transaction. Other examples include financial assets such as stocks and debts, which are non-fungible assets in many business scenarios.

The most common of what we call fungible assets is money or digital currency (AssetToken in UDAP term, which is also an ERC20 token). In most cases, one hundred dollar bill is completely equal to and replaceable with another one hundred dollar bill because the main purpose of using one hundred dollar bill is to use one of its most prominent attribute, medium of exchange. Although each bill has some special attributes, such as its unique serial number printed on the the paper bill, it has no specific significance and influence in most cases. Therefore two hundred dollar bills are completely replaceable. Another example of fungible assets is commercial goods, such as apples in a warehouse. Although apples have some special attributes, such as origin, variety, grade, size, color, etc, however, when we say that we have 600 tons of Yantai-produced first class Red Fuji apples with a size of 85mm, these 600-tonne apples are traded as fungible assets because the 600-tonne apples are classified as one group according to industry specific standard. There is no difference to the buyer in the trade.

One special type of non-fungible assets, which is referred as CompositeAsset in the asset hierarchical model, is composite asset or asset portfolio. This type of assets typically contains a number of other assets, for example, asset portfolio in the financial sector, warehouse receipts in the supply chain, and many asset bundles that require the packaging of different assets as a whole for trading. Below is the definition of the interface.

```
contract CompositeAsset is StandardAsset {
  function getAmount() public returns (uint);
  function getAsset(uint idx) public returns (address);
}
```

- Transferability

  The transferability of assets determines whether an asset can be transferred to other institutions or individuals after being issued. Transfer of asset can be done either by direct sale or through exchange. If an asset is not transferrable, then its ownership cannot be changed.

- Resellability

  Resellability is a special attribute that asset issuers can use to restrict the resale of assets. Although in most scenarios assets are re-sellable, in certain special cases, such as shopping vouchers, issuers can restrict the resale of vouchers. In this way, the vouchers can not be transferred or sold after the vouchers have been issued to them, thus limiting the circulation of such assets in the secondary market and ensuring that such assets are available only to recipients designated by the issuer.

- Namespace

  An asset namespace refers to the naming rules for asset classification and grouping so as to facilitate the distinction between different assets. Namespaces are commonly structured as hierarchies to allow reuse of names in different contexts, for example, a warehouse receipt can be identified via udap://xinong/wr/WR-12345678, where "xinong" is the registered app name (or chain name), "/wr/WR-12345678" is the namespace id given by the app (or chain) to identify a warehouse receipt coded as WR-12345678.

- Asset Identifier（AID）

  Asset identifiers are unique identification codes set by the UDAP platform for assets and are automatically created for each asset during asset registration process by a standard algorithm as follows:

  ```
  AID = uint(keccak256(issuer_address, namespace, metadata_multihash))
  ```

  This algorithm generates a unique AID for an asset with its issuer's account address, its namespace, and the hash value of the asset metadata description file in IPLD[14] or JSON-LD[15] format. Once AID is generated for an asset, it cannot be modified anymore. Assets issued by different asset issuers, or assets issued by the same asset issuer but with different namespace id, or different asset metadata descriptions, will always have different AIDs.

- Metadata MultiHash

  It refers to the content-addressable MultiHash value that is generated from asset metadata to uniquely address the asset's metadata. See 5.1.2 for details about asset metadata.

- Other Attributes

  In addition to the above attributes that usually can not be changed once identified, assets also have mutable attributes that can be updated after issuance, such as the amount of fungible assets (e.g., 1 kg of gold, or 500 tones of apples), the state of the asset (leased, unused, listed, etc.), ownership, proof of assets, description, etc.

5.1.2 Asset MetaData

Metadata is commonly referred to as "data about data." In UDAP asset metadata refers to the descriptive information applied to assets and is defined by asset issuers. The structure and meanings of these metadata are known to asset issuers and the associated applications. In UDAP, asset metadata is presented as JSON data in conformity with JSON-LD specification, and is stored off-chain, while on-chain crypto assets must hold Multihash values of their off-chain metadata. Smart contracts can obtain the multihash value through the metadataHash() method to address and retrieve the related asset metadata.

```
  function metadataHash() public view returns (bytes);
```

JSON-LD, or JavaScript Object Notation for Linked Data, describes how linked data is represented in JSON as a directed graph, and how to represent interlinked and non-interlinked data in a single document. For example, metadata about a recipe asset can be recorded in JSON-LD format (shown below) and published to IPFS or other P2P storage. Typically, asset metadata can be encrypted before posted to P2P storage to enforce data privacy.

```
{
   "name": "Mojito",
   "ingredient": [
     "12 fresh mint leaves",
     "1/2 lime, juiced with pulp",
     "1 tablespoons white sugar",
     "1 cup ice cubes",
     "2 fluid ounces white rum",
     "1/2 cup club soda"
    ],
```

```
   "yield": "1 cocktail",
   "instructions": [
      {
        "step": 1,
        "description": "Crush lime juice, mint and sugar together in glass."
      },
      {
        "step": 2,
        "description": "Fill glass to top with ice cubes."
      },
      {
        "step": 3,
        "description": "Pour white rum over ice."
      },
      {
        "step": 4,
        "description": "Fill the rest of glass with club soda, stir."
      },
      {
        "step": 5,
        "description": "Garnish with a lime wedge."
      }
   ]
}
```

The above metadata can be recorded as a merkle-link on the blockchain so that applications can address through merkle-link to obtain the relevant off-chain asset metadata.

```
{"md",{"/","QmdnuRNwdmZzHfHVUMVHZFXKXAe6DjvBvPdKy27HpJUN9H"}}
```

UDAP adopts a simple method to record only the multihash value on the blockchain which points to the off-chain metadata. The specification defines how to obtain content-addressed objects through the hash value.

```
{"metadataHash","QmdnuRNwdmZzHfHVUMVHZFXKXAe6DjvBvPdKy27HpJUN9H"}
```

Usually applications need to address each item of metadata to obtain the resolution of the related data. Therefore, the content-addressable network data model is used to address the asset metadata through the merkle-path. For example, IPFS DAG's javascript interface (ipfs.dag.put) can be used to upload metadata of the recipe asset to the IPFS so that each metadata item is available by invoking "ipfs.dag.get".

Metadata on IPFS as DAG node:

```
  ipfs.dag.put(metadata, { format: 'dag-cbor', hashAlg: 'sha3-512' },
    (err, cid) => {
      console.log(cid.toBaseEncodedString())
     // zdpuAz4HbUHTKQbdpnn42Zo4GUsU7yrBpvb2W9BF2NwvBaLn6
  });
```

DAG node through merkle-path:

```
 ipfs.dag.get('zdpuAz4HbUHTKQbdpnn42Zo4GUsU7yrBpvb2W9BF2NwvBaLn6/name',
  (err,result)=>{
    if (err) {
      console.error('error:'+ err);
     }
    else {
      console.log(result.value);
     }
 });
```

5.1.3 Ownership

The ownership of an asset is a type of asset metadata that tracks who owns an asset. An asset can have multiple owners. The ownership of assets can be changed over their lifecycle, for example, when an asset is transferred to another person from current owner, the ownership of this asset is changed. In UDAP asset protocol, while asset ownership belongs to a type of metadata, UDAP manages asset ownership as an independent attribute that can be tracked by a smart contract defined as follows:

```
contract Ownership {
  function ownerOf(address asset) public view returns (address);
  function ownerOf(address asset, uint asOf) public view returns (address);
}
```

The above interface gives the caller the capability of obtaining current owner account of an asset as well as the owner account as of a given time in the past.

5.1.4 Asset State and Lifecycle

Asset states can be used to accurately track assets at a detailed level. In addition to a few pre-defined asset states such as CREATED, TRANSFERRED, PLEDGED, LEASED, applications can define and name their own states according to business needs and record asset states on the blockchain via UDAP.

The life cycle of an asset refers to asset state at different points in time. One of the main functions of UDAP is to provide asset lifecycle management APIs on the blockchain that allows applications to track the entire lifecycle of assets from issuance to destruction to meet various business needs.

Asset lifecycle can be represented with a smart contract as follows:

```
contract Lifecycle {
  // returns current state of a given asset
  function stateOf(address asset) public view returns (bytes32);
  // returns the state of asset at given asOf time
  function stateOf(address asset, uint asOf) public view returns (bytes32);
}
```

5.1.5 Proof of Asset

Proof of Asset (PoA) is an important concept of the UDAP asset model. It is one of the key elements in determining the authenticity of assets. It is also a type of asset metadata in the UDAP asset model, represented as an array of JSON objects in JSON-LD or IPLD format. Each of the JSON objects defines a proof including name, description, and a content-addressable linkHash value of the "proof". The "linkHash" represents a MultiHash value that can pinpoint this proof, which may be a digitally signed PDF file or a scanned shopping receipt. Proofs are stored off-chain and can be obtained through the metadata's merkle-path.

```
"proofs":[
  {
    "name":"Storage Contract",
    "description":"Storage contract for warehouse receipt #123456",
    "linkHash":"QmWwr4ZfeLJfbWNAuCQfefwo1aHtxC5yjyU8C5WG4DYrYe"
  },
  {
    "name":"Purchase Receipt",
    "description": "Purchase receipt for warehouse receipt #123456",
    "linkHash" :"QmXF4LR4QkuRVh3WQbB56seTX2aPm3Tz7b4Y8heoLAiTkk"
  }
]
```

Proof of Asset is an optional but important attribute of crypto assets. Usually asset buyers will require some sort of proofs, however, without a proof an asset can still be traded or exchanged on the market. More discussions about Proof of Asset can be found in 5.3.

Proof of Asset may have different forms in different use cases. For example, in the supply chain warehouse environment, a warehouse receipt is a proof to demonstrate the authenticity of the assets. Other relevant proofs include purchase agreement, storage contract, third-party certificates, etc. Who owns the warehouse receipt owns the rights of goods stored in the warehouse; in the manufacturing sector, manufacturers can use RFID tags or two-dimensional bar codes to uniquely identify their products. In this case, RFID of a product is a proof that demonstrates the authenticity of this product. As such, proofs are data defined and provided by crypto asset issuers to prove the authenticity of assets that can be either numbers or files or images, all in JSON-LD format.

5.1.6 Tags

Asset tags are keywords or labels that are attached to assets to facilitate identification, classification, retrieval, and inventory control of assets. Multiple tags can be given to an asset by its issuer or owner. Asset tags enable applications in a variety of industries to track and monitor valuable assets.

5.1.7 Asset Registry

An asset registry maintains a bi-directional binding between crypto assets and real-world assets on the blockchain. All assets posted to the UDAP blockchains need to be recorded in the asset registry by its issuer. At the same time, the asset registry also maintains a number of different data structures and indexes to simplify search and retrieval of assets. Different institutions or applications have their own proprietary registry of assets. Asset issuers can broadcast their assets across the entire network or just to some designated addresses, so that a proprietary registry has access to assets registered in other registries via listeners. For example, landlords can post rental information on multiple rental sites simultaneously, and sellers can initiate auctions on both Site A and Site E. When an asset on Chain A is transferred to an address on Chain B, UDAP must ensure the removal of this asset from the registry of Chain A, and the addition of this asset to the registry of Chain B.

5.1.8 Account, Wallet, and Identity

Accounts are users' address on the blockchain. A user may have multiple accounts, while an account may have multiple assets. An asset may be associated with multiple accounts. For example, an asset can have multiple owners, an issuer, and multiple signatories. An account can also play different roles in different trading scenarios, e.g., asset issuer, asset owner, or transaction signer.

> Asset signatory refers to the account that signature of transactions is required before trading the asset on the UDAP blockchains.

An asset wallet, similar to a safe or a deposit box, is a universal wallet provided by UDAP to allows tracking and management of assets associated to a user's account. An asset wallet can manage multiple accounts, each associated with multiple assets on the blockchain.

Identity refers to the user's personal or social information, for example, ID card, Facebook account, e-mail address, phone number and other information that can represent the user's identity. Blockchain accounts are anonymous, but in some scenarios users have to provide proof of identity to complete the regulatory requirements for KYC and AML. UDAP, in conjunction with a self-governed third-party identity management system such as uPort, provides the application with a mapping of identities to UDAP accounts and asset wallets.

5.1.9 Transactions and Events

Transactions refers to any operation on the assets on the UDAP blockchain. For example, issuance, exchange, ownership change, minting, recasting, pledge, approval and so on are all transactions. When a user initiates a transaction on the UDAP blockchain, the UDAP generates related events and broadcasts to related listeners, which are responsible for processing the transaction.

5.2. Asset Services

On top of the asset model, UDAP has defined a service model for asset operations and management. Common services are exposed to application developers through micro-services (REST APIs and

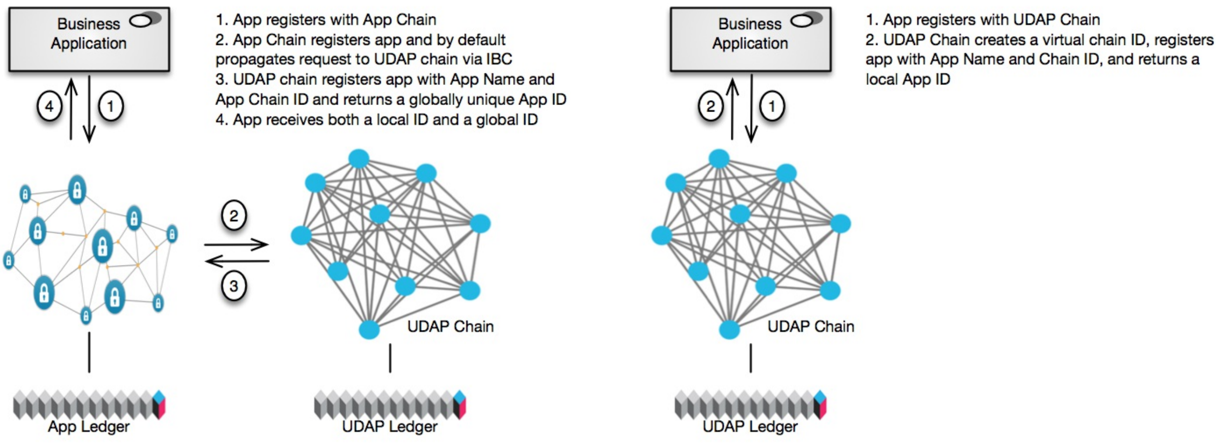
WebSockets). Unlike the traditional centralized cloud service model, UDAP provides a decentralized service architecture. In this decentralized service architecture, the API gateway and service host are a special type of miners that provide the host container for running UDAP asset services. The gateway is the entry point for third-party applications to connect to the UDAP blockchain, and is responsible for automatic routing application requests, and provides service metering capability as a basis for service charges. At the same time, the nodes that provide the service gateway and the service container are also rewarded by the network. Hosts providing asset services need to deposit a small amount of locked-in tokens and need to broadcast their identities to the UDAP blockchain. UDAP chooses the node serving the service based on the proof of the node.

5.2.1 User Registration

User registration establishes the mapping between the user space in the application domain and the user space of UDAP blockchain. User accounts between different applications are completely isolated. Applications are responsible for registering their user accounts with UDAP blockchain to create a mapping. See 5.4 for more description.

5.2.2 Application Registration

A UDAP-based application has either an independent asset chain (deployed and owned by application vendor) or a virtual chain sitting on top of UDAP main chain. These two deployment settings support both private/consortium and public blockchain configuration. In either case, applications need to connect to UDAP main chain and register themselves with the Application Registry on the UDAP main chain. When applications are registered, each application is given a unique App Id and a unique App Name. App Name is used as the level 0 namespace id of the managed assets. In the registration process, each application also receives an App Key and an App Secret that are used to securely connect to the UDAP main chain.



1. App registers with App Chain
2. App Chain registers app and by default propagates request to UDAP chain via IBC
3. UDAP chain registers app with App Name and App Chain ID and returns a globally unique App ID
4. App receives both a local ID and a global ID

1. App registers with UDAP Chain
2. UDAP Chain creates a virtual chain ID, registers app with App Name and Chain ID, and returns a local App ID

a) application registration with private setting

b) application registration with public setting

When application prefers a private or consortium configuration for asset lifecycle management, it can leverage UDAP's Asset Blockchain as a Service (ABaaS) to deploy a private or consortium asset chain. This UDAP-enabled permissioned blockchain is specific to this application and is by default automatically registered with the UDAP main chain. Transactions on the app chain are stored locally in a private ledger on the ABaaS managed nodes. This configuration gives the app chain the capability of broadcasting asset information to or communicating with other UDAP-enabled chains through an Inter Blockchain Communication protocol. If an application doesn't want to have an independent network, it can choose a virtual private chain configuration, where application's ledger is stored and managed on the validator nodes of UDAP main chain.

5.2.3 Asset Registration

Assets need to be registered on the UDAP blockchain for applications to query and manage their states on the blockchain. In the meantime, applications receive asset registration information broadcast by other applications, enabling cross-application and cross-chain asset transfer and trading. Assets registration is a two-way binding process between real world assets and crypto assets. Asset metadata is identified and uploaded to off-chain P2P storage network at this stage, and a hash value of the off-chain metadata is stored and associated to the crypto assets. In this process, asset issuers need to make detailed configuration of asset attributes, for example:

- Transferability： an asset can be either transferable or non-transferable. If an asset is not transferable, the asset is usually a warrant asset. The only meaningful operation is "delivery", which means that the warrant owner delivers to the original issuer the promised product or service.

- Sellability： an asset can be either sellable or non-sellable, which defines the ability to be sold. If the asset is configured to be non-transferrable, the asset is essentially non-sellable.

- Multi-signature requirements: A multiple signature (or multi-sig for short) requirement represents that a transaction requires multiple approvals from different participants. Multi-signature addresses and transactions broaden this model by creating identities on the chain which are managed collectively by multiple parties. UDAP uses “m-of-n” bitcoin-style multi-signatures, in which a multi-sig address A is defined as: Given n regular addresses, at least m of the private keys corresponding to those addresses must sign a transaction to perform an action for A.

5.2.4 Tokenization

The purpose of asset tokenization is to make asset transfer and trading more easier. This is a fairly frequent operation that the asset issuer completes mapping from the real world to the crypto world after registering a real asset attached to the application context to UDAP. Token issuance allows assets to be traded in part rather than as a whole. For example, a painting can be tokenized as a certain amount of tokens through what is sometimes called "tokenization" (sometimes referred to as "minting,") so that the painting can be sold to multiple owners, where each owns a portion of the rights and interests of the painting.

As seen from the UDAP asset model, tokens are fungible assets. Fungible assets are usually tokenized at the time of registration, whereas non-fungible assets are traded as a whole in most scenarios and therefore do not require the issuance of tokens. It is only necessary for issuers to mint tokens when they want to trade their assets partially.

5.2.5 Asset Recast

Recasting refers to the process of burning tokens for the rights to redeem goods or services, which creates new proof of asset for the token owner. This process is usually valid for physical assets in specific scenarios. This is because tokens themselves do not necessarily have the associated attributes of physical assets and the tokens are issued by the asset owner to enhance liquidity. When the physical asset is in the custody of a third party, the tokens issued by the asset owner may not always be accepted directly by the custodian of physical assets. Therefore, in many cases, tokens can not be used to directly redeem physical assets and new crypto assets need to be generated through the process of "recasting". For example, after Alice registers her 500-tonne apple on the blockchain to form a crypto asset (crypto warehouse receipt), Alice can issue a token per tonne. Alice then transfers 100 tokens (corresponding to 100 tonnes of apples) to Bob. After Bob receives 100 tokens, he can submit the tokens to the UDAP recast contract to generate a certificate to redeem asset (e.g., bill of lading) and destroy the corresponding tokens. Then Bob can redeem goods with this bill of lading. After redemption, the related crypto assets (warehouse receipts) are automatically destroyed on the blockchain to prevent double spending.

5.2.6 Multisig

Multisig is an additional security protection mechanism in the process of asset trading. It refers to the process that multiple accounts digitally sign the same transaction before it is executed. Only when required signatures are collected will the transaction be broadcast to the chain. In many scenarios multiple signatures are required to complete a specific asset operations, for example:

- Registration: For expensive assets, the application may require signatures of the designated accounts be provided at the time of asset registration in order to prove the authenticity of the asset. Auditors and witnesses are possible co-signers in this process.
- State Change: Some key state changes may need to be confirmed by multiple parties.
- Trading: Co-ownership of assets requires the signatures of multiple owners when assets are transferred or sold to a third party.
- Pledging: Pledges usually require multiple approvals to ensure authenticity of assets, accuracy of price, and security of transaction.
- Asset freeze (冻结): The asset freeze may require the signature of the court and the parties involved.
- Asset write-off (核销): This operation usually requires the approval from multiple supervisors. For example, reimbursement process requires the signatures of direct managers and CFO to complete expense reimbursement.

5.2.7 Base Coin Issuance

App chains may need to issue their own tokens as utility tokens for users to use their services or as base coins for pricing managed assets. For example, if a person wants to use event ticketing application to sell an event ticket, he may need to price the ticket with the base coin and pay the service fee with the base coin. UDAP supports application to issue base coins just like Ethereum supports Dapps to issue ERC20 tokens. As base coins are also assets, they can be traded against UDAP token (UP) or other application-specific tokens through an exchange.

5.2.8 Other Asset Services

In addition to the basic services described above, UDAP provides the following services:

- transfer
- rent
- buy and sell
- C2C trade
- pledge
- auction
- escrow
- redemption and destruction

5.3. Counterparty Risk and Proof of Asset

All the tokens on the UDAP chain are tied to assets. Tokens and assets are the counterparties of the bonding. Since we are dealing with real world situations, anything can happen to the assets without being noticed by the token system. This is the counterparty risk.

UDAP assumes a few basic principles about the authenticity of assets, the counterparty of the token system:

- Authenticity is not provided by the UDAP protocol.

- Authenticity is only valuable in its application context.
- The authority and authenticity of assets can be confirmed through a mechanism that is considered to be reliable and adequate by specific applications.

UDAP is a distributed system. It does not have a single operating entity to verify the authenticity of assets. While asset verification may be done in a distributed and decentralized manner, UDAP currently does not design such mechanism.

However, if a person claims a crypto asset that he owns on the UDAP network is backed by a real-world asset, how could he prove this claim? UDAP proposes following schemes and all the primitives are supported by UDAP:

1） Proofs of Asset

Under normal circumstances anyone can issue assets on the blockchain via an app. At the time of asset issuance, the issuer has to provide a detailed description about the asset in the form of texts, images and other media. The issuer may also present some real-world confirmation of the asset, for example, a certificate of property, an impartial letter, a warehouse receipt, a purchase receipt, and so on. All of these supporting documents do not necessarily guarantee the authenticity of the assets, but these additional attributes to a certain extent increase the authenticity of the assets. The specific scheme is application specific.

2） Guarantee or Insurance

Asset issuers or owners can guarantee the authenticity of the assets by providing some form of guarantees or insurance.

3） Multi-Sig Protection

Asset issuers can leverage the multi-sig mechanism in the registration process to enhance the authenticity of assets. In this process, multiple proofs from related parties are usually required.

The first step in enhancing the credibility of assets is when assets are mapped from the real world to the crypto world, which is what we mean by registration. With the support of UDAP, a multi-party signature must be provided when an issuer declares an asset on blockchain. The signature comes from the current owner of the asset. Other signatures may be obtained from current custodian of the asset, the notary and the third party auditor who verify and confirm the ownership of the asset.

If a third-party application considers that the registration of a user's asset is important, it can leverage the multi-signature mechanism provided by UDAP to allow the relevant guarantor (including asset custodian, notary and auditor) of the asset during asset registration to submit proofs of asset and sign the transactions. For example, if a person claims that he owns a gold bar himself, the application requires that the user must, at the same time as the declaration, submit a gold asset certificate issued and signed by a custodian that certifies such ownership, as well as the digital signatures and associated certificates from other third party auditors. In the absence of any of these digital signatures will result in the denial of asset registration.

4） Escrow

The ultimate value of assets is achieved through circulation. Escrow is a common method of trading assets between untrustworthy individuals. The owner of an online store may claim to own any goods. The platform does not recognize the credibility of such claims. Instead it offers a custodian mechanism that usually holds temporarily assets that are easily deposited by both parties in the transaction, for example, purchases of ordinary merchandise from an e-Mall, the money paid by the buyer does not go directly to the owner of the goods, but goes into the escrow contract first, and then the buyer must confirm the authenticity of the goods after the buyer receives the goods. The traditional e-commerce platform basically adopts this kind of mechanism. For example, on the localbitcoin.com[16] platform, the escrowed objects are the commodities to be traded, that is, bitcoin, while the money to buy bitcoin is paid offline.

5. Staking

An application may choose to require the users to put in a stake in the system before registering the assets. The stake is in the form of in-app currencies or UDAP tokens. The stake is used as the collateral in case of malicious behavior.

For those products that are very expensive, the applications may choose to deploy multiple protection layers, for example, a combination of above mechanisms, which prevents possible fraudulent activity during asset trading. UDAP provides API for third-party applications to establish a flexible mechanism to ensure the authenticity of assets and to ensure the reliability of the transfer process.

5.4. Identity Management

The main purpose of UDAP is to manage real-life assets with blockchain technologies, and to use these technologies to increase their liquidity and thus enhance its permanent durability. At the same time, one important goal of UDAP is to allow these real world assets in the form of crypto-assets to be easily transferred between owners.

The first and foremost problem is the identity of the participants in the asset related transaction. For real-world asset transfers, it may not be good enough to know only the account numbers. Because the conversion of these assets may involve the requirements of the real identity, for example, a contractual relationship established by the two sides through the asset network, if traceability is important, then the identity of the parties have very clear requirements. The two parties to the contract need to know each other very clearly, that is, the individual that they want to make the transaction. Once the contract is in trouble during execution, there is always another way for the contract party to track and confirm the problem in the real world. Considering again that if third parties' law enforcement agencies are able to participate in this process, the identification of the identities of both parties to the contract is a prerequisite for their participation.

Each application will determine how much they need to be aware of the identity of the user in the real world. In more stringent applications, they may need to be fully aware of the ID of the user of the application in the real world, his address, and his contact details. For example, a cryptocurrency exchange may require users to submit their own proof of identity in real life, such as ID cards, passport photos, and to verify their phone number and EMAIL address. Moreover, proof of their places of residence may be required and fully verified. In fact, this is the basic verification process that most of the major crypto exchanges are conducting. Other applications may not have such KYL requirements. Maybe they just need to know the registered user's EMAIL address, or maybe they are not interested in knowing any of these information.

There are several types of identity management systems on the Internet:

- Fully centralized identity management system, such as citizen ID card system.
- Fully siloed authentication system, for example, each application has its own authentication system.
- Federation Identity System: Internet applications that have taken shape to a large scale provide authentication for other applications through OpenID, OAuth, or similar mechanisms. The provided process can add user's confirmation of identity and additional property exposure.
- Self-Sovereign Identity Management.

Given that UDAP provides an open ledger service to third-party applications, we decide to adopt an evolutionary path for UDAP to transition from a siloed identity system to a Self-Sovereign identity system:

- A person's identity is provided by a third-party application. There is no abstract identity outside of the application. That is, each application corresponds to a persona. UDAP does not pursue as a single identity provider. The independence of application identity can provide privacy protection and account security.
- Applications are responsible for user account setup. But the process to create a public/private pair in creating an account on the chain is executed by UDAP and the private keys are not supposed to be touched by the applications. Private keys are never stored in applications. They must be kept in the mobile phone in an encrypted form.
- UDAP provides a key ring to securely store the private keys of the accounts in each of the applications that a user choose to register.
- Private keys are mainly used to sign transactions initiated by a user, either from the application plugins in UAW, or from standalone mobile apps.

In the second phase of UDAP Identity Management:

- UDAP provides a decentralized PKI interface (dPKI) to third party apps.
- UDAP integrates a Self-Sovereign identity system that allows users to have multiple identities and have full control over the use of their identities across applications.
- Asset Wallet (UAW) provides Hierarchical Deterministic account creation and management mechanisms.
- UDAP supports account recovery to prevent users from losing their identities if their devices or passwords are stolen or lost.

We are looking closely at the development of some third-party identity management systems and will consider the possibilities of integrating their services into UDAP network. These systems include (but not limited to) uPort, ShoCard, Civic, Mooti, and others.

6. Universal Asset Wallet

As a very important part of the platform, wallet plays a key role in the entire agreement and interaction between system and user. The usual crypto-wallets are a single-function wallet that shows the amount of particular cryptocurrencies.

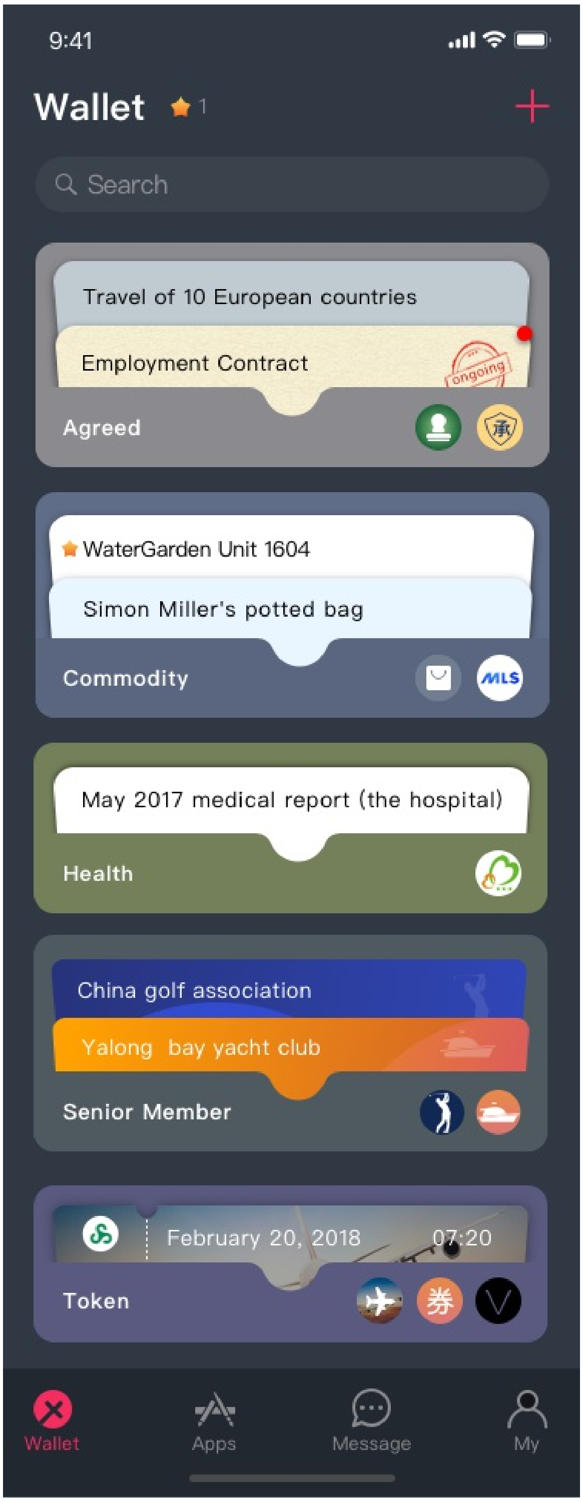
As a general-purpose assets wallet, UAW's role has been significantly expanded. Think of it as a personal safe on one's cell phone. It can contain any stuff you would put in a safe, such as certificates, important documents, diamond rings, antiques, securities, IOUs, etc. In fact the asset wallet stores the unique tokens that cryptographically represent the assets. These asset-backed tokens are created by various applications running on the UDAP chain are organized by applications.

General Assets Wallets are more than just the storage of assets, but are also very handy for supporting common operations on assets such as asset transfers, sales, trading, auctioning, cancellation and more.

6.1. Assets Precipitated From Third Party Apps

UAW is the user of the assets obtained in a variety of third-party applications. Each application has its own unique user management system that generates user accounts that correspond to a unique account on the blockchain, with the help of UAW. Obviously account will not be shared or reused between applications.

An account has security significance and identity only in the context of third-party applications. The purpose of doing so is to ensure that the user's privacy. In this context, a wallet, in fact, must import the assets from each application. After the account is imported, the assets of the corresponding user in the third-party application are displayed in the general-purpose assets wallet in different groups. Users can interact with the asset in a generic way, such as transferring, submitting to markets, auctioning and sending copies to third parties.



Travel of 10 European countries

Employment Contract

Agreed

★ WaterGarden Unit 1604

Simon Miller's potted bag

Commodity

May 2017 medical report (the hospital)

Health

China golf association

Yalong bay yacht club

Senior Member

February 20, 2018 07:20

Token

6.2. Personal Assets

UAW not only imports assets from third-party applications, it also allows users to create personalized digitally encrypted assets directly in the wallet. In fact this feature is also an application from the architectural point of view.

One scenario is that users can create IOUs. The purpose of IOUs in daily life is to show that a person owes a particular piece of asset to another person and he/she promised to return/pay back at some time in the future. People used to write down IOUs on a piece of paper. Now UDAP client gives users an easier and securer way to write Crypto-IOU, with support of voice, pictures and even videos, tamper-proof and irrefutable, no worry of loss.

In the process of personal loans or IOUs, one can also use the wallet to conduct multi-sig signing. For an example, in the process of creating an IOU, a third-party witness may be required to witness the contract. The borrower can send the original IOU to the witness, who then signs it and sends it to the creditor.

User can even introduce the [guarantor] role. Unlike the [witness], the guarantor has to assume the corresponding obligation of security if the debtor can not execute the contract in the agreed time for debt repayment, then the guarantor must bear the repayment obligations.

In summary, crypto-IOU is an extremely powerful.

a. Saving notarization and costs b. Very convenient to keep safe c. Personal debt can further be traded.

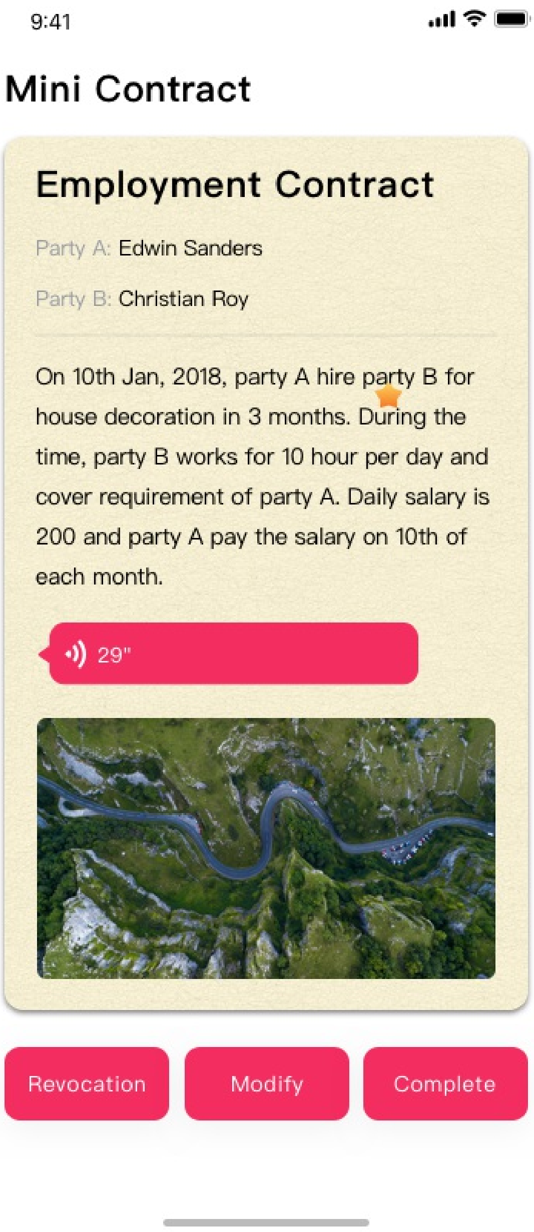
In other scenarios, a user of UAW can register any personal belongs through the description of words or pictures or videos. Once tokenized, the person belongs can enter a market for sale.

6.3. Debts

UAW manages assets in the financial sense while also managing liabilities, or "negative assets."

Often times, what we mean by assets is something that is valuable to ourselves. In many scenarios though, we not only need to know how much assets we owns, we also need to manage liabilities and the reimbursement requirements and conditions for those liabilities.

The most common example of "negative assets" is the various types of "utility bills" that arrive on a monthly basis: water / electricity / gas / telephone bills.



Mini Contract

Employment Contract

Party A: Edwin Sanders

Party B: Christian Roy

On 10th Jan, 2018, party A hire party B for house decoration in 3 months. During the time, party B works for 10 hour per day and cover requirement of party A. Daily salary is 200 and party A pay the salary on 10th of each month.

Of course there needs to be an application that connects the utility companies to UAW. Once we have it, users can use the wallet to pay directly in the UAW, or the user can ask someone else to pay for them.

Universal Asset Wallet is a very powerful tool for everyone to handle their finances. Not only does it handle some of the personal lending activities of everyday life just as it is, but it can also be used by small businesses such as family hotels or family restaurants as a tool for sending discount coupons or vouchers.

6.4. Multi-Sig Support

UAW supports multi-sig transactions.

When a transaction is made and requires multiple signatures, UAW will send a message to the appropriate signer's wallet and prompts the designated signer to review the transaction details and then "agree" or "reject" the transactions.

The UAW message queue presents the list of transactions that need to be signed. The history of the signature will also be preserved.

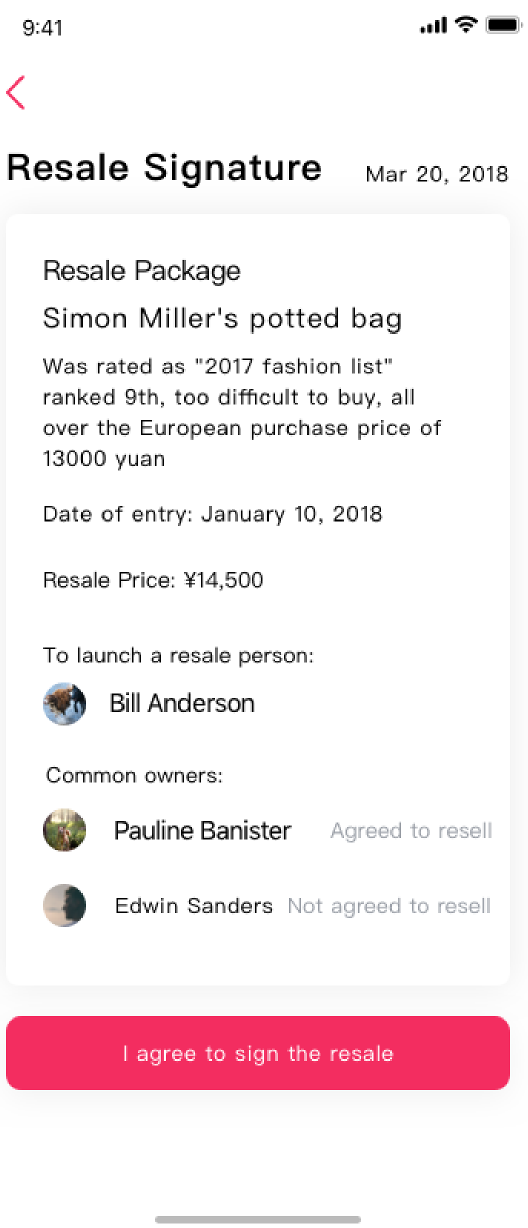
Initiation of the transaction can be from another UAW user, or from a UDAP-enabled application. This is very convenient for third-party APPs, since they do not need to build their own multiple-signature workflow.

6.5. App Store

UAW is a powerful and extensible plug-in architecture, and its main purpose in fact is to support a large number of third-party apps that can generate a wide range of assets. It provides a unified user experience integration with UDAP's ability to provide background integration for third-party apps. The plug-ins for these clients constitute the ecology of the entire Asset Internet. All third-party applications appear in the App Store of the UAW for the users to choose from.

Technically UAW is a hierarchical architecture, and a considerable part of the user experience has been built in the UAW. Applications from third-parties range from ticketing, memberships, financial assets, health records, academic records, IOUs and crypto-contracting.

Not all third-party applications provide a UAW plug-in. An App can have its own native app, or a web application.



Resale Signature Mar 20, 2018

Resale Package

Simon Miller's potted bag

Was rated as "2017 fashion list" ranked 9th, too difficult to buy, all over the European purchase price of 13000 yuan

Date of entry: January 10, 2018

Resale Price: ¥14,500

To launch a resale person:

Bill Anderson

Common owners:

Pauline Banister Agreed to resell

Edwin Sanders Not agreed to resell

UAW is an eco-system. It provides a programmable UI for third-party applications to develop a complete user experience. The UAW itself provides a series of built-in features that save the time it takes for a large number of third-party applications to reach their own user base.

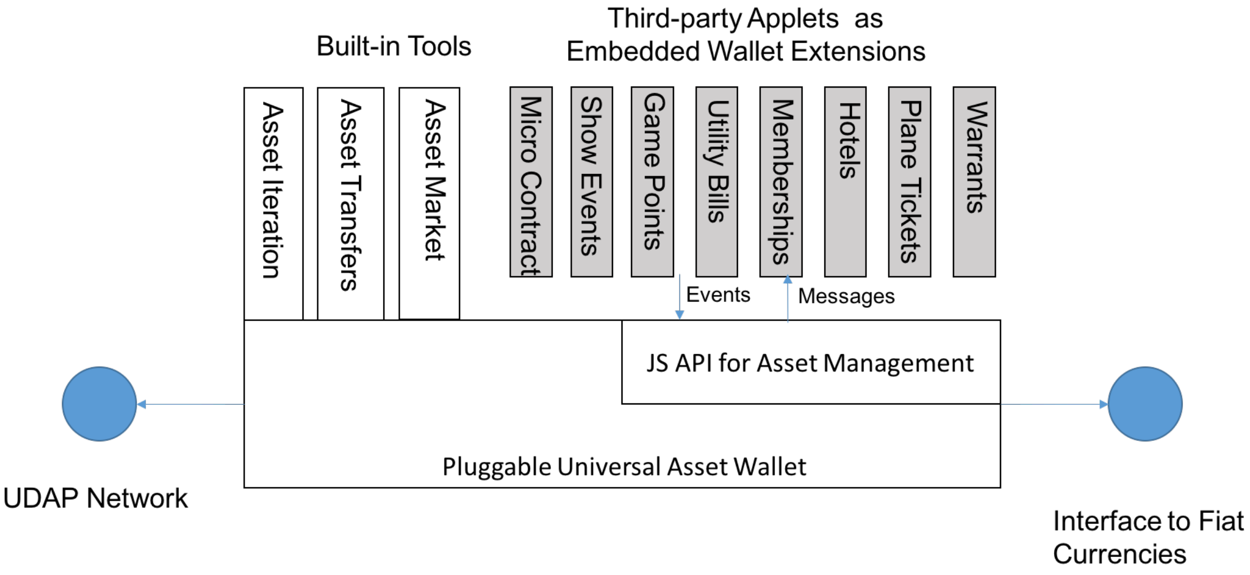
1. Simple Asset Transfers: Users are like sending emails, or as usual Crypto Currency. Encrypted assets are sent from one account to another.

2. Obtain permission to transfer: In order to prevent the rubbish assets from flooding wallets, users can set to require other parties to obtain permissions to send any transfers.

3. Ticketing: This is the *deliver* tool that comes with the UAW. Event tickets are in fact the promise of services provided by asset issuers who will deliver the final product or service within a certain period of time. The redemption process is actually transferring the tokens back to issuer.

4. Market: UDAP comes with market tools that asset owners to sell their assets.

5. Multiple Signatures support.



UAW offers JavaScript APIs for third-party application developers. Some of the functions are as follows:

1. getCurrentUser: Get the current user's identity information, including the chain address.
2. getAssetCollection: Get the current user's list of assets.
3. initTransfer： Transfer specific assets. Activate UAW's multi-signature mechanism if asset transfer requires multiple signatures.
4. offerToSellInMarket： Offer to sell an asset in the market.

More API will be provided in the future, including various auction methods, escrowed P2P transactions and decentralized exchanges.

7. Implementation

7.1 Protocol Implementation

As mentioned before, UDAP at the highest level is a conceptual model of real-world assets. The following section provides the descriptions of the ongoing implementation of the model. It's subjected to change as the development work moves along.

UDAP addresses are divided into account address, asset address, and assetProto address. Where account represents the actual account of the user, organization, service provider, etc. Asset represents an asset, AssetProto is a prototype of a type of asset, and an asset must be associated with an asset prototype in a one-to-one correspondence.

These three types of address have a versionHash and lastCommitTimestamp property. VersionHash represents the changing status of this address. Each change of an address will generate a new random and never duplicated versionHash value. The lastCommitTimestamp will record the last point in time when an address was received after the transaction was processed.

UDAP transactions are always

A transaction usually includes the following data

- The originator of the transaction
- Recipient of the transaction
- The method of trading
- Trading method parameters
- The digital signatures required for the action
- versionHash, optional transaction acceptor versionHash.

The originator of the transaction must be an account address, and the recipient of the transaction can be an asset address or an assetProto address.

When the UDAP accepts the transaction, the rule matcher verifies whether the signature is of the transaction request, whether the signature is out of date, whether the signature time is later than the lastCommitTimestamp of the transaction acceptor, and if the versionHash of the transaction acceptor is given if the versionHash is given Given conditions and all legitimate signers would meet the signature rule. If so, UDAP API then accepts the transaction request to complete the operation and generate a new versionHash.

The signature includes the time of signing, the signature's validity period, and the versionHash that may contain the current asset. When the app collects enough required signatures, it initiates the transaction to UDAP, which verifies that the provided list of signatures satisfies the rule by interpreting the method parameters. If so, it changes the state of the asset, generate a new versionHash and record the latest lastCommitTimestamp in all signatures as the signature is for the transaction method parameters. UDAP creates an optional versionHash and lastCommitTimestamp Time guarantee to prevent replay attack. VersionHash transaction request and digital signature are optional. It is designed to ensure that the asset data will not be phantom read, similar to what optimistic locking does. Whether you need to verify that versionHash is in the hands of the originator of the transaction.

7.2. The Architecture of Virtual Private Chain (VPC)

A flat network of blockchain nodes is not scalable. A flat topology requires enormous amount traffic to reach consensus and synchronization and in effect in the throughput of the entire network is constrained by the computing power and IO performance of a single node.

The transitions from POW to various POS designs and other BFT consensus variations are inevitable if the blockchains are to place the role of world computers.

In the meantime, homogeneous sharding mechanisms have being proposed and being implemented in various projects, such as Ethereum Sharding, ELASTICO.

Plasma and Polkadot are trying t build blockchain of blockchain to integrate heterogeneous blockchains to achieve great scale.

Cosmos uses interconnected zones to realize linear scalability. It's a sharding scheme both at the consensus level and state level.

UDAP architecture is largely based Cosmos architecture and loosely based on Plasma. It will eventually phase in the Plasma hierarchy, for best security and scalability.

At the root is the UDAP root chain, which stores the meta information of UDAP applications. Each application is assigned an application chain which is by default a private chain that's only visible to application nodes. An application can choose to deploy to the shared infrastructure of UDAP or to specific private nodes.

A number of nodes are grouped to form a "zone" under the UDAP root. The number of nodes is chosen such that the POS (Tendermint to be specific) works the best in terms of the balance of security and TPS/# of nodes.

Application chains are logical chains that can span multiple "zones" to achieve near-linear scalability. Each zone becomes a shard of a VPC.

Multiple applications are supported concurrently on UDAP network, just like multiple VPN sessions run on the same TCP network.

Each node can run every application too, supporting the transactions of each application and store the transaction history, the global state trie and receipt trie. This does not exclude the possibility that some applications choose to localize to some subset of the nodes to form a pure exclusive private chain, in which case, the "virtual" part of the VPC is transformed to "real".

Some applications may also localized to particular zones, either because the other zone are already fully loaded and not accepting new application deployment or they choose to do so for geological proximity.

In each node, the application transactions are processed in parallel. This is possible because there is no dependencies between applications. All the cores of any modern server will be able to participate in processing transactions at the same time, in contrast to the serialized transaction processing which can use one core only no matter how many cores the computer has.

There are support for parallel processing of single application transactions too. This is made possible by the fact that all the UDAP API exposed to the applications are deterministic in terms of what account are affected by each API call. The smart dispatcher in each node will able to group transactions in multiple queues that are not dependent on or otherwise interfere with each. Thus the queues can work in parallel to take advantage the multiple cores available.

7.3. Performance Implementation

There are four major factors that are accountable for the abysmal throughput typical of Bitcoin network and Ethereum:

1. The same transaction must be replayed on all the nodes, no matter how expensive they might be. A great deal of computing power is wasted in competing repetitive calculations. This has been proved to be one important way to make sure all the nodes behave by the rule and great security

is maintained. However, this first-gen technique is very expensive in nature and is the primary reason that blockchains' throughput are incredibly poor, considering how much computing power is behind each network.
2. On each node, all the transactions are serialized in building into the blocks and there is no way to build the blocks in parallel because there is no partitioning of the transactions which is required to avoid race conditions in reading and writing account information.
3. Smart contracts as implemented in Ethereum are slow. They're interpreted at runtime instead of running in native speed or in highly optimized VMs, such as JVM.
4. API does not give finality. Transaction initiators basically send transactions and wait for confirmations, which varies from a few minutes to hours.

The first issue is being tackled by a few solutions such as POS based consensus and some other protocols which do not use blockchain at all. UDAP RI specifically uses Cosmos[17] as the general network architecture. Cosmos offers the following features that UDAP can immediately leverage:

1. A POS consensus implementation (Tendermint) that provides 1000+ TPS performance.
2. An API that provides *finality*, which is a must for UDAP to become an easy to use API.
3. Cosmos can scale linearly with adding more "zones" to the blockchain hub.
4. It gives 1-2 seconds response delay for API invocations.

For the second limiting factor, the lacking parallelism in transaction processing is caused by lacking transaction demarcation, which is in turn because there is no sufficient information about account dependencies in general blockchains. One transaction may be depositing money to an account while another is taking money out of it. Working on the same account needs synchronization, which basically serialize the access to accounts.

Fortunately, UDAP defines boundaries for applications natively:

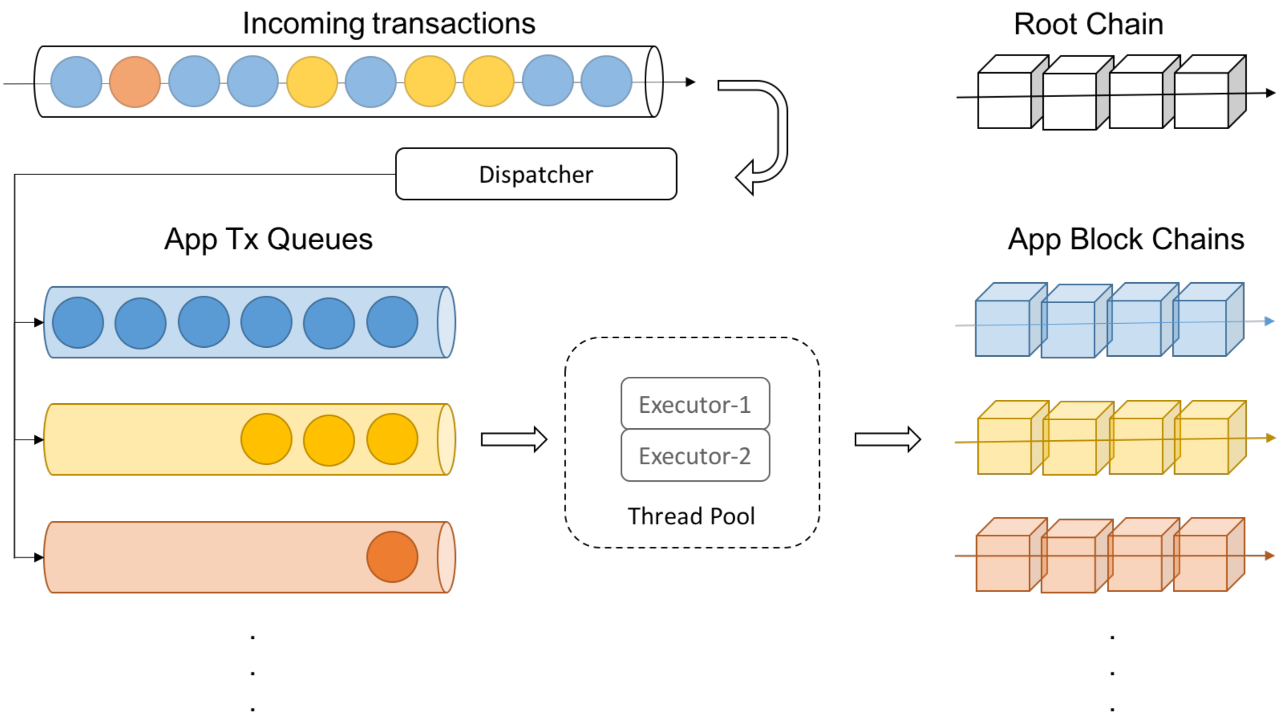
1. User accounts for each application are unique. In fact, a user must register for each application and receive an address. There is no chance that the same account is used in more than one application.
2. Transactions always executed between the same type of account.
3. Security models and transactions are application specific. There is no shared information among applications.

In comparison, Ethereum ERC20 tokens are not the naturally wall between applications. Some transactions may involve multiple tokens. An user's account (external account) is not bound to a single type of token. User accounts are generic accounts that are not bound to any specific dApp, therefore there is no natural demarcation boundary for transaction verification and mining to operate concurrently on multiple cores of a machine.

In UDAP, each application is assigned an application ID and owns an independent chain, meaning there will be a thousand blockchains in UDAP network if there are a thousand application registered. Each chain has its own branch of state trie, storage trie and transaction receipts. State transitions in each application takes place in a separate thread.

The following picture shows how the transactions received by a node are properly dispatched to separate transaction queues for each application. All the CPU cores are assigned to process the transactions in parallel and the transactions are Merkleized in an application specific blockchain.

A Node of Multiple App Chains



One chain per application is a major design choice we have made that is very flexible in optimizing the performance and security, which are the primary two requirements for any applications that claim to deal with any assets. Each chain is an overlay chain on top of the generic UDAP transaction streams. We call the mechanism Virtual Private Chain(VPC), as analogous to VPN over TCP/IP.

As described previously, the root chain is there for

- application registration and configurations.
- Generic user account registration, which provides custodian service for the account of the third-party applications running in UAW. Users can optionally choose to register them with UDAP to take advantage of the identity attestation service.
- It also periodically takes snapshots of the application chains states in case that fraudulent or faulty behaviors are reported from the app chains and state enforcement is required. Eventually the root chain will probably become a Plasma chain which is the parent chain of all the applications chains.

Although we have improved the parallelism at the application level, there is still chance of blocking in a single application level. Transactions for a particular application still have to be processed serially. If one application becomes so popular that it consumes most of the bandwidth in a period of time, all the transactions coming to a node might just be from it, and they will be processed one of another. The workload cannot be spread over to the multiple cores on the same machine.

The above issue can be solved by a smart thread dispatcher in a UDAP node, presuming that UDAP APIs mostly affect one account at a time. The dispatcher carefully inspects the incoming transactions and separates them by affected accounts. With transaction partitioning still feasible, the transaction verifications and block constructions can still be done in parallel.

With no competitive and repetitive computing among all nodes, and with the optimized parallelism in the transaction processing in the node software, UDAP would provide significant better scalability both horizontally and vertically. We expect UDAP network offers TPS in the 10K~100K range with 600 nodes. We also note that we need to make sure the disk IO in each node must be optimized to take advantage of parallel IO, by using NVMe interface, rather than simple SATA interface.

In the meantime, we're closely monitoring the progress of the Plasma Project led by Joseph Poon and Vitalik Buterin, which is a hierarchical multi-chain architecture that utilizes recursive Map/Reduce computing with faulty behavior correction mechanism to achieve unlimited scalability. The Plasma development will be one of the core effort from a team of the best talents in the industry in 2018. We plan to leverage the work in the future to solve the scalability issue for once and all. In the meantime, other on-chain scaling solution such as the Ethereum Sharding proposed by Vlad Samfir is regarded by some as the "true secure scaling solution for Ethereum", and "will provide for all of the scalability requirements of the blockchain without sacrificing on security or the trust model." It's our goal that any change or improve deployed underneath won't affect the users of UDAP.

7.4 Privacy Enforcement

Privacy on blockchains is counter-intuitive for many people, because blockchains usually promote openness and publicity, at least for public chains.

Bitcoin and Ethereum are not really privacy centered blockchains. In fact, most public chains can be trusted with correctness but not confidentiality. Bitcoin, for example, is one of the most traceable currencies that offer little privacy protection. Additionally some form of KYC policies are required of many applications and exchanges, therefore users of most of the on-chain applications are heavily exposed to privacy breaches.

There are indeed multiple levels of solutions today that cover the privacy issues in part. Coin mixers (such as Mimblewimble and TumbleBit) improves privacies by mixing transactions so it's harder to trace the origin and destination of transactions. Monero provide partial transaction confidentiality. ZCash, with Zero-knowledge Succinct Non-Interactive Arguments of Knowledge provides strong confidentiality, at a much higher cost of computation power and engineering complexity.

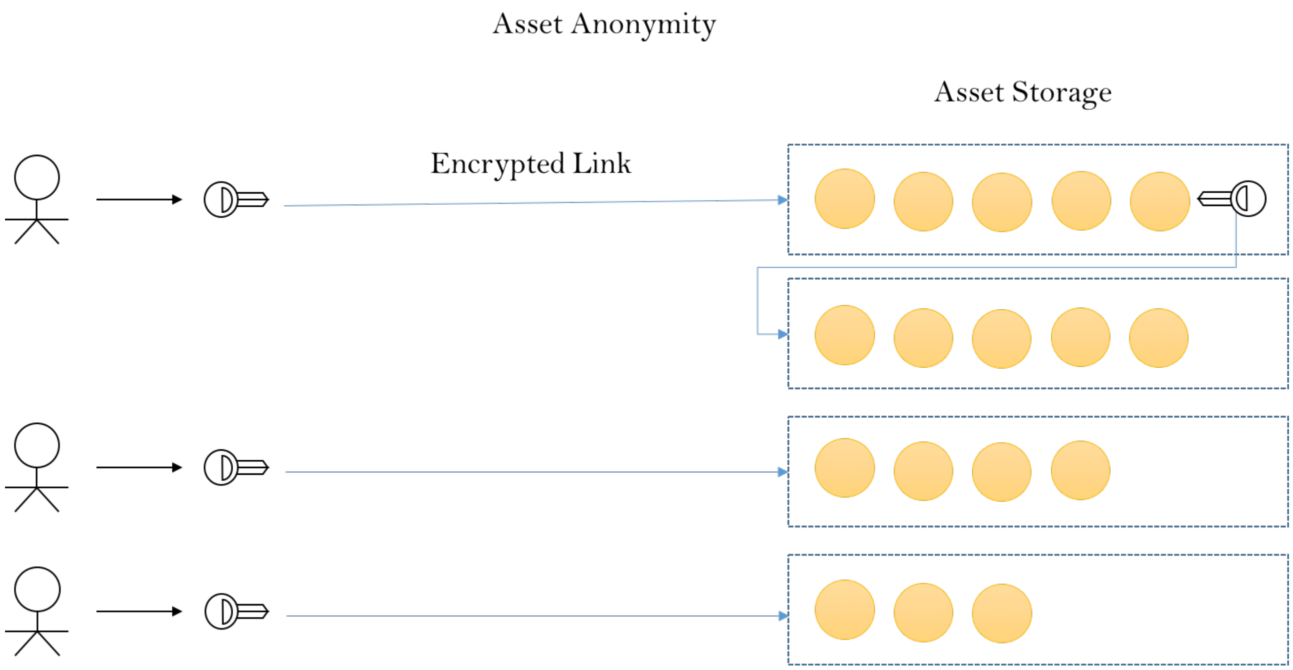
Data obfuscation techniques try to hide the identity of data and cut the traceability of the data flows, so that the connections between people and their assets are decoupled. Coin mixers, ring signatures are of this type. Data anonymity is achieved with an extra layer of processing either locally or as a network service. Performance takes a hit necessarily.

Using cryptography to encrypt data is the other major mechanism to preserve privacy. It ranges from simple symmetric/asymmetric encryption to the sophisticated zero-knowledge scheme based mechanisms.

In UDAP, VPC (Virtual Private Chain) is the first line of defense of privacy. UDAP respects that and by default it allows app developers to choose the private deployment model whereby only trusted nodes are allowed to process the transactions and store the state. Private deployment mode is the default mode of application chain. Specifically, VPC features:

1. Streamlined chain access only from the exposed high level API.
2. No block browsing for participants;
3. Auditing as requested by authenticated application operators and by authorized personal customers.

Although we believe most of the third-party apps will be deployed on permissioned UDAP app chains, thus having the basic firewall to fend off privacy attacks, we also believe The second defense line is with data obfuscation and encryption, as shown in the following diagram.



User account has an encrypted pointer pointing to one of the storage slots, which are the vaults for asset tokens. Think of the design like the custodian vaults in banks where customers use their own key to open the storage of assets. No one knows who owns which vault. The vaults are anonymous; thus privacy is protected.

The connections between the wallet and the chains are protected in three layers of encryption, slightly similar to what the TOR protocol does:

1. Secure Socket.
2. Data is encrypted with a user's key and the data packets are signed by the user.
3. Data stream goes through the application's endpoint which in turn encrypts and signs the already encrypted data with the application key.
4. The UDAP nodes must make sure all the traffic coming from an application's client carry the proper signature of the application and and the check cannot be circumvented. The use of the layered encryption and signing is not so much about concealing the IP address of the function invoker, rather it's for the purpose of letting the application to verify the validity of the API call of the user in the context of a specific application. Theoretically some functions calls made by users can be directly targeted to the UDAP node which does the chain dispatching. But we feel most applications need a mechanism to intercept function calls from their users and may decide to activate more business logic upon such invocations, such as more involving authorization. UDAP nodes which receives such transactions would definitely know that the transactions have been authorized by the corresponding applications, such that a higher level of security is ensured.

In the future we are considering:

1. either implementing the zkSNARKs in the protocol level to meet the rigorous privacy requirement of data sensitive rules;
2. or other emerging technologies such as Solidus for more streamlined privacy architecture;
3. quantum computing resistance will be placed in our road map in the next few years to keep our platform up to date and future proof.

As required by many states, KYC is integrated with most blockchain exchanges and some blockchain applications. KYC means that people's identifiable information is stored somewhere central. UDAP does not store

7.5 Key Rings and Identity

Today's cryptocurrency wallets put too much burden on end users in managing their accounts/passwords. People feel so much pressure in keeping the account credentials in safe place and in the meantime still feeling convenient to use them.

We believe a key ring technology similar to Apple's iCloud key ring is required to make a first class secure experience with the asset wallet. UDAP Key Ring should:

- Have a single lock key to protect all the accounts/passwords.
- All the private keys must not be saved to the network.
- Two factor authorization must be tuned on. UAW will provide TFA service to all the applications registered with UDAP.
- In case of password loss, a combination of email and cell phone is required to recover the parent account with UDAP.

User accounts are application specific. Any account is associated with an app. Different apps don't share accounts. But account registering needs the help from the Universal Asset Wallet, for absolute security.

Private keys are created in the UAW. They never should be exposed to applications. Once the public/private key pair is created, the public key is presented to the application's account creating process which may require more information from the users. The private key is encrypted and stored in UAW key ring.

Each application must repeat the above process to acquire new customers. A UAW user will have as many identities as the number of applications he/she uses.

7.6 Data Storage Strategy

Every transaction on the blockchain incurs a fee. This is partly due to the fact that the public blockchain is a public support resource, which requires some incentive mechanism to encourage the participation and voluntarily provide public blockchain computing and storage infrastructure. On the other hand, transaction fees can greatly limit any malicious attacks on the blockchain network, because such attacks are economically unrealistic. So while we believe the transaction costs on the blockchain will decrease dramatically when the performance and scalability issues of the blockchain are resolved, however, as a decentralized asset chain even though the entire lifecycle of assets is required to be managed on chain, it is impossible for us to store all the data related to managed assets on the blockchain. Therefore, an important architectural decision is what kind of data needs to be stored on chain and what kind of data needs to be stored off-chain. Such an architecture decision needs to be considered in many aspects such as context, processes, costs, performance, and realizability.

From the perspective of business, costs and performance, blockchain is not suitable for storing frequently updated data and large volume of data. Any changes to the data may trigger commitment of transactions and data replication on each node. Through the analysis of the on-chain asset model, we believe that the basic attributes and the metadata of assets can be separately stored. The basic attributes and transaction data of asset management are stored either on the blockchain while the metadata is stored off-chain. For example, for a Multiple Listing Service (MLS) in the real estate industry, metadata about a property such as description and pictures are stored on cloud storage or P2P storage, however, owner information and transaction history are stored on the blockchain. Therefore, a UDAP full node has two logically isolated "nodes" (a blockchain node that stores transactions and assets, and a storage node for asset metadata storage). Assets have access to their related metadata via merkle-link. Application developers don't have to understand where the data are stored.

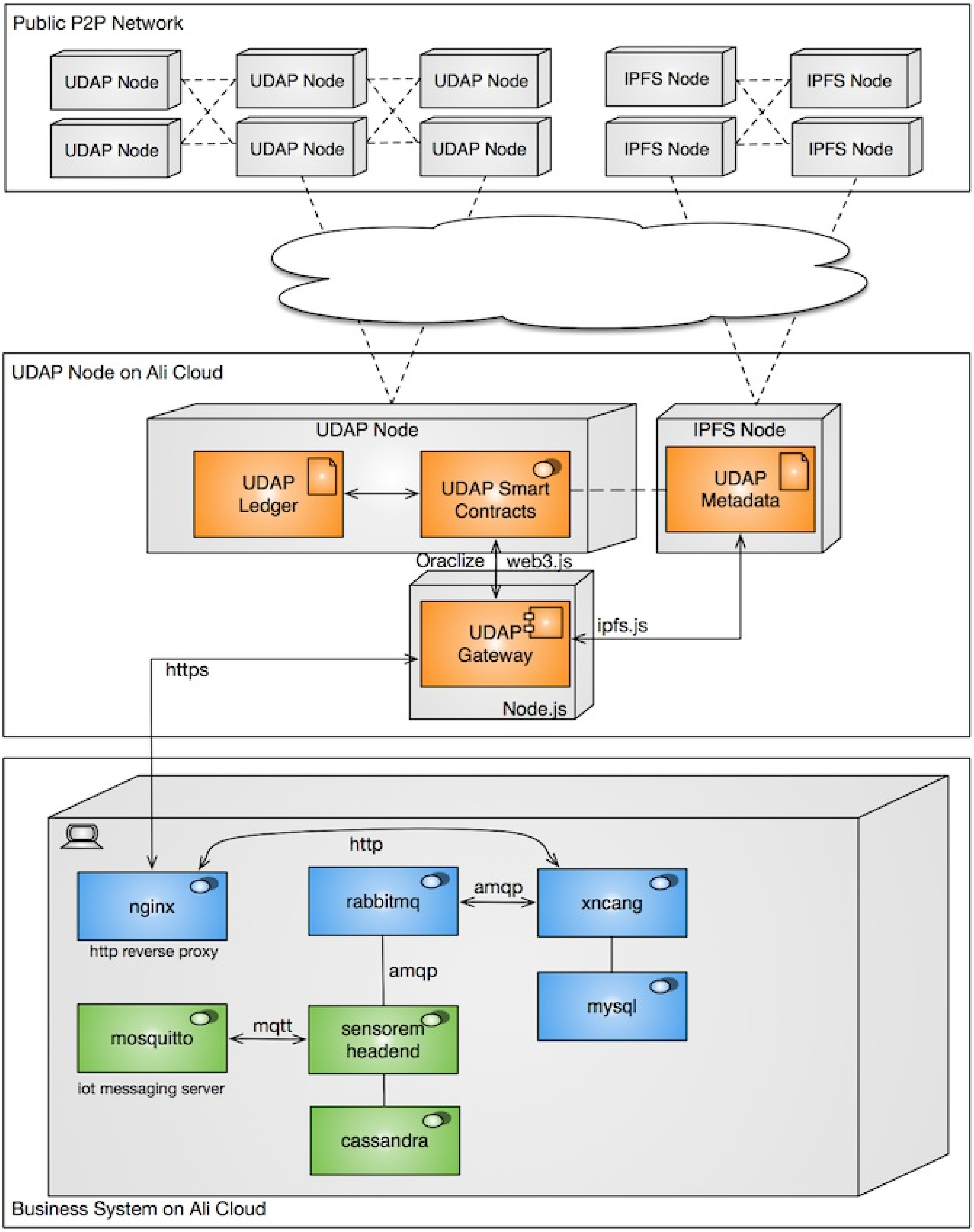
In general, there are two options for metadata storage: a centralized cloud storage such as Amazon S3 and a decentralized P2P storage like Swarm, IPFS, StorJ, or MaidSafe. Both options have their own advantages and disadvantages, but for an asset of great economic value, it may be more reassuring to have a decentralized storage that does not rely on any centrally managed storage services. Although

UDAP prefers a decentralized storage to offer a comprehensive decentralized service with a decentralized computing infrastructure, this protocol does not directly define a physical P2P storage technology. It only requires the data is presented as merkle-link and merkle-dag, so that the application can address and query any relevant data via merkle-path. This merkle-link represents a link between two objects, which maps the cryptographically hashed values of the target object's content to the source object, and therefore allows us to get the target object through this link. This approach has at least four advantages.

- data can be easily presented in JSON-LD format
- data encryption and integrity check are supported
- data is immutable
- data is addressable through merkle-path

These advantages are particularly well suited for the management of asset metadata, as most metadata items are static and descriptive data with few changes over the life of an asset. And in many scenes, once metadata is uploaded to storage it is not allowed for modification, for example, university diploma, license agreement, and contracts and so on. Even a slight change on the metadata will lead to an obcious change of the hash value of the metadata. Therefore either the hash value on the blockchain needs to be updated, or a naming service such as IPNS is required as a tag of the metadata. The former needs to submit a transaction on the chain, which incurs a small amount of costs. The later needs to introduce a new level of data abstraction. In either case it is transparent to applications. Services provided by UDAP allow application developers to completely ignore the underlying storage logic and operate directly on assets.

In addition to asset metadata, applications often have large volume of business data that are usually stored in their own databases. While decentralized storage may not be a good place for storing large volume of frequently changed business data, from the application point of view, many business systems need to adopt a hybrid storage model through careful data analysis. Business systems need to put some of their data on the chain (including decentralized storage). Meanwhile, business systems also serve as off-chain Oracles that provide data services to smart contracts. For example, in the case of warehouse receipts as collaterals in supply chain finance, after warehouse receipts are registered as crypto assets, their market values are calculated by smart contracts with real-time price data obtained from business systems or third-party Oracles via Oraclize[20] service. The price data is then signed and recorded on the decentralized storage as a basis for future value verification. The following is an example of an infrastructure deployment view from one of our demos, where xncang is a business system that manages warehouses and inventories, which connects to a PoC blockchain via an API gateway.


Public P2P Network
UDAP Node
UDAP Node
UDAP Node
UDAP Node
UDAP Node
UDAP Node
IPFS Node
IPFS Node
IPFS Node
IPFS Node
UDAP Node on Ali Cloud
UDAP Node
UDAP Ledger
UDAP Smart Contracts
IPFS Node
UDAP Metadata
Oraclize
web3.js
ipfs.js
https
UDAP Gateway
Node.js
http
nginx
http reverse proxy
rabbitmq
amqp
xncang
amqp
mysql
mosquitto
mqtt
sensorem headend
iot messaging server
cassandra
Business System on Ali Cloud

8. Related Work

The year of 2018 is an important year for non-fungible crypto assets. The world is in a transition from cryptocurrencies to crypto assets. A lot of efforts have been done to make this transition into reality. We have reviewed related blockchain projects and summarized our key findings as follows:

BankEx is a blockchain project that targets financial asset management and offers "Bank as a Service" cloud service. This project builds permissioned blockchains on Ethereum and creates smart contract based static asset model. New asset types are manually registered through a centralized approval mechanism.

Digix is a blockchain project specialized in tokenization and trading of gold. It uses gold as collateral to create crypto gold. It creates "recast" concept adopted by UDAP to handle redemption of physical goods or services.

AChain [19] is a platform that offers token issuance, smart contracts, and Dapps development. It creates a multi-chain architecture through a forking mechanism.

Bytom [20] is an exchange protocol for diversified byte assets that uses POW as a consensus approach, supports limited asset types and mainly focuses on the financial aspect of the assets (in other words, tokens without asset metadata).

0x Project [21]is a decentralized exchange for cryptocurrencies. It has a very interesting exchange model that UDAP may adopt to create a C2C exchange for assets.

Bitshares [22]is an exchange for trading cryptocurrencies and assets. It is a permissioned blockchain with a single chain architecture.

WAX is a marketplace for virtual game assets exchange and trading.

MediaChain is a singular data fabric for open-first media applications. It is a decentralized blockchain for applications and users to publish, discover, and collaborate on media metadata. It is built on Ethereum and IPFS.

We have also reviewed and researched a few key blockchain projects that focus on multi-chain architecture with inter-blockchain communication. These projects include Plasma, Polkadot, Aion Network, Wanchain, and Cosmos.

9. Use Cases

9.1 Event Ticketing

This use case includes performances, live shows, sporting events, ticket management for various gatherings.

A third-party event ticketing platform that focuses on ticketing issuance, distribution and marketplace services. It is often the case that event ticket holders may not be able to attend a event for whatever reason, and that they need to be able to transfer the tickets to others in the best way possible. In the opposite direction, some people may have missed the ticket sale event and do not have a reasonable and convenient way to get tickets for the event, that is, this type of assets lacks a convenient secondary market. The performance market is still a relatively good market, because in private, scalpers play a liquidity role. Though not through a formal channel, they indeed improve the liquidity, help balance the supply and demand and thus get incentivized.

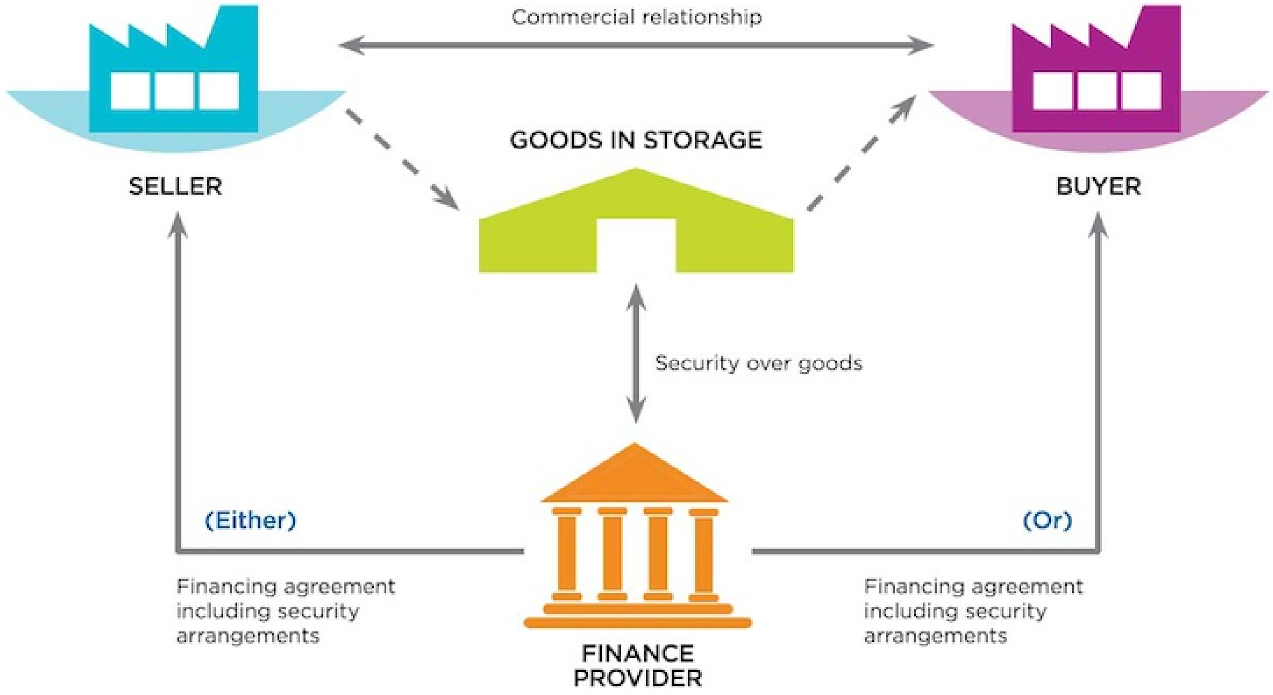
By tokenizing event tickets, tickets can be easily sent to or transferred between friends, or put on a secondary market through UAW for resale. In the process of trading, the original issuer can put control over the trading frequency, liquidity, and price range. They can also set restrictions on whether an asset is allowed to be resold or transferred. This provides an extremely handy feature for ticket management apps with unparalleled security and liquidity. Our universal asset wallet (UAW) can be used directly to execute operations such as ticket transfer, trading or on-site check-in, which are common to concerts, movies and other types of events.

9.2 Brand Valuation

In the crypto world, people are gradually realizing that tokens are a brand new economic model. The essence of tokens is actually to digitize all kinds of accessible and inaccessible assets in the real world and to manage them with cryptographic and blockchain technologies. Tokens corresponding to these assets can make full use of the high liquidity brought by blockchains and cryptocurrency exchanges to form a brand new token economy. One of these very innovative ideas is the emergence of a new possibility for everyone to issue tokens that represent their reputation and brand. Imagine that in the future, everyone will be able to issue their own crypto currency which is backed by their personal image and reputation, credibility, and promises. This type of tokens represents some kind of credit and commitment individuals make to the world and other people. Because of the liquidity of tokens, values of individuals can be more accurately represented. With the support of our platform, individuals can easily build their fan clubs and control issuance and distribution of membership cards. Those who can provide services could tokenize their services and put tokens on the market. When demands for their services change, the value of their tokens changes accordingly. This scenario offers an unprecedented possibility of full employment and customer demand satisfaction.

9.3 Supply Chain Finance

Nearly 82% of businesses fail because of cash-flow problems, however, the complexity and scale of existing supply chain finance (SCF) solutions has posed major challenges in ensuring adequate funding and efficient operations. Finance instruments in SCF include factoring, reverse factoring, payables financing, inventory finance, and dynamic discounting. The following figure illustrates the transaction flow of a typical inventory finance solution (source from Global SCF Forum)


Commercial relationship
SELLER
GOODS IN STORAGE
BUYER
Security over goods
(Either)
(Or)
Financing agreement including security arrangements
FINANCE PROVIDER
Financing agreement including security arrangements

UDAP will essentially enable all parties in SCF solutions to act on a shared ledger, where suppliers and manufacturers, along with every other participant, will solely update their parts of the transaction, enabling efficiency and an “unprecedented” level of trust and transparency on a ledger record that is immutable.

9.4 Marketing and Promotions

In the retail industry, cash vouchers, discount coupons, membership vouchers and promotional certificates can all be tokenized for monetization to create a more convenient and more liquidity asset. With the support of UDAP, these assets from different issuers can be easily exchanged, traded, transferred, and redeemed. Like event tickets issuers can also set resale rules on the crypto assets to protect or increase their interests.

9.5 Sharing Economy

UDAP enables businesses to quickly build decentralized marketplaces on the blockchain for a sharing economy. Buyers and sellers of decentralized "airbnb" or "uber" like car-sharing or home-sharing could transact on a decentralized and open platform without traditional intermediaries. All facility sharing rules are transparent to customers. A decentralized arbitration mechanism resolves disputes fairly and grow a network of mediators through incentives. In some scenarios, reservations can become assets. When buyers have to cancel the bookings for some reasons they may face penalties at present, however, with UDAP it is possible for buyers to resell their reservations on the marketplace to reduce loss.

9.6 Game Assets

Gray markets exist for exchange and trade of digital assets (e.g. equipment, resources, accounts, points) in all kinds of e-sports games. Game developers may be reluctant to allow the players to freely trade the game equipment and resources, so that players have to obtain new equipment and resources through in-game purchase. However, there are quite a few games that realize that providing an open marketplace for game props is a way to enhance the user experience, attract more users, and increase revenues via the resell of game resources. UDAP offers APIs that enable game developers to register certain types of crypto assets they issue and manage the trading rules for those types in trading and exchange. At the same time, the ecology of asset trading conforms to the dynamic model expected by game designers.

9.7 Arts and Collectibles

Spot trading of collectibles is a very promising application of our asset management network. Our platform not only provides basic computing functions, but also provides the file storage and multimedia storage capabilities required for the preservation of art collection information. Therefore, all kinds of digitization, encryption and tokenization required in the circulation of artwork can be used to directly manage the trading. There are two main types of transactions that existed in the past: antique and art shops with direct acquisitions from individuals, which were then offered to consumers for purchase. In addition, A trading model takes place in the private, free-market model where art owners and potential buyers make deals directly; a common selling model for art is the auction model because artwork is usually an asset with insufficient liquidity that have huge disagreements about the pricing of artwork, and have huge gray spaces, sometimes used for money laundering and improper business activities. Blockchain technology will help eliminate frauds and provide traceability and authenticity guarantee with a flexible transaction model.

9.8 ICO

UDAP supports applications to issue their own tokens as utility tokens for exchange of services or application specific base coins for asset pricing. The application tokens can be exchanged with other tokens. This will help transform business into a token economy.

10. Conclusion

Capital market is an engine for economic growth, both for business entities and individuals. Monetization is the main way to profit from the economy.

In the next few years, world economy will be reshaped greatly by token-based businesses. The nature of the "universal assets" that we are advocating here is the store and the realization of value.

Blockchain supports permanent retention of asset information through its tamper-proof feature. In a sense, it is the permanent existence of assets.

At the same time, the blockchain-derived token economy and the liquidity as the core of the token economy provide the key channel for the value recognition of assets.

The relationship between liquidity and the health of the entire industry is like a human blood circulation system and human health. Much of what is studied in economics as a whole is actually about how to improve the liquidity of a local system. Although trading liquidity is frequently over-estimated, which is referred as "liquidity illusion", a closer look at various industries around us reveals that lack of liquidity is almost always a continuing challenge for all industries. Even if we are already satisfied with current liquidity provided by a system, at a higher level and in the future, this liquidity may become inadequate again. So in current reality, increasing liquidity is always of importance.

Liquidity is not a panacea. In fact, liquidity itself may also bring system instability and even harm. Just like our blood circulation system, smooth blood circulation does not mean that there is no control of blood circulation. Therefore, controlling and optimizing liquidity throughout the market is actually the second challenge for asset owners.

Many industries face a big challenge that asset issuers lack control over liquidity of assets. In the past there was no good solution to this problem, but with the advent of distributed ledger technologies we are able to overcome this challenge for the first time.

This white paper is about an ongoing project referred to "Internet of Assets", which we are trying to make our unique contribution in three main technology areas:

- Blockchain-based Asset Lifecycle Management
- Decentralized C2C Exchange for Assets
- Virtual Private Chain Technology that solves sore issues with respect to privacy, security, scalability.

We believe our highly targeted networks offer an unprecedented new technology platform for a large number of applications that can precipitate a wide range of assets and support businesses to transform into a token-based economy.

References

[1]: http://www.omnilayer.org/

[2]: https://counterparty.io/

[3]: https://prism.exchange

[4]: http://unchainedpodcast.co/vitalik-buterin-creator-of-ethereum-on-the-big-guy-vs-the-little-guy

[5]: https://www.comp.nus.edu.sg/~loiluu/papers/oyente.pdf

[6]: https://theinternetofmoney.info

[7]: https://github.com/ethereum/wiki/wiki/Design-Rationale

[8]: https://blog.ethereum.org/2018/01/02/ethereum-scalability-research-development-subsidy-programs

[9]: http://docs.mediachain.io

[10]: https://digix.global

[11]: https://bankex.com/en/whitepaper

[12]: https://www.cmswire.com/cms/digital-asset-management/the-building-blocks-of-digital-asset-management-interoperability-021996.php

[13]: https://www.ibm.com/developerworks/cloud/library/cl-adopting-blockchain-for-enterprise-asset-management-eam/index.html

[14]: https://ipld.io

[15]: https://json-ld.org

[16]: http://www.localbitcoins.com

[17]: http://cosmos.network

[18]: http://www.oraclize.it

[19]: https://www.achain.com

[20]: http://bytom.io

[21]: https://0xproject.com

[22]: https://bitshares.org