SOCIAL COIN WHITE PAPER

TOKENIZE YOUR SOCIAL ASSETS

VERSION 1.0
BY SoCo TEAM

ABSTRACT

Today, all social networking platforms are centralized systems. Users submit their personal data to the platform. The platform takes advantage of the data, casts target advertisements, and reaps massive profits. However, users do not get any return on profits, while suffering the cost of leaking their personal privacy. Under this business model, the current social platform is actually selling users as commodities to advertisers. This is not what users originally expected. We initialized this project in order to solve this problem.

The SoCo Project involves building a decentralized social network ecosystem, powered by *Social Chain* and called Social Network Protocol, by leveraging Blockchain 3.0 technology. The SNP is a novel business model that strives to return the ownership of social assets to individuals, as well as transfer their social assets from centralized platforms to a decentralized ecosystem. Additionally, the project aims to resolve a few major problems and limitations with existing blockchain technology, by developing Social Network Protocol (SNP) stack so that it can be backed up by a number of consensus algorithms and protocols.

SNP is a stack of open and extensible protocols across all layers, which authorizes, appreciates and tokenizes personal social assets through blockchain technology. SNP enables all APPs to have social capabilities, so that users can use Social Coin (SoCo) in the aforementioned APPs and receive the full monetary value of their social assets.

The social chain is a Blockchain OS consisting of 4 layers, the SNP API layer, the service layer for distributed storage and social chatting, the third layer for smart contracts, account blockchain, contract blockchain, auditing proof of service record (PoSR) and transaction records, and the fourth layer for SoCo PoRS and transaction consensus which are hash-saved into a vector of SoCo chains.

SNP plugs security into basic building blocks. It introduces an extra level of Account class and Contract class. Any such new class will be subject to harsh verification process before being saved in the Account chain or Contract chain. The first level of verification is compiler checking, the second is tool scanning (e.g. Coverity, valgrind and etc) and the third is complexity testing. If complexity testing

fails, then the class is subject to BFT consensus from a number of selected wallets (software engineers). Individual AccountBot or ContractBot instantiated from an on-chain class is allowed to program rules but rules will be subject to additional verification consensus.

The social chain introduces a novel network architecture, with a powerful consensus overlay in the core and service overlays on the serving edge: service nodes for regular services and consensus nodes for auditing, contract execution and consensus; a service node can be promoted to a consensus node after an election based on reputation, and a consensus node can be demoted to a service node; the reputation score of a node is determined by the SoCo holding percentage, consensus service term, average service response time, available bandwidth, storage and CPU, and so on. Again, election is a mechanism where rules apply after a broad range of consensus from service to consensus nodes.

A service node can spawn an account node, a storage node and a social chat node, and a consensus node can spawn an account node, a contract node and a SoCo node. Each type of SNP node is best customized for SNP protocol operation. SNP horizontally enforces contract execution, and vertically enforces contract API authentication and contract auditing and consensus until adding into the relevant blockchains. All SNP nodes are ultra-lightweight containers.

In the consensus overlay, all nodes can be hashed on SoCoID into two groups: higher hash values go to the contract overlay and lower hash values go to the SoCo overlay. Nodes with the same hash value form a candidate ring and they take turns to service the contract overlay or the SoCo overlay. On top of this autonomous mechanism, rules after consensus can be programmed to the contract overlay and the SoCo overlay, to ensure that there is fairness between all consensus nodes, working nodes are behaving properly, longer and better nodes are rewarded with better chance and gains, and so on.

The social chain scales up in two novel approaches. First, instead of a single blockchain, it introduces a vector of SoCo blockchains. By hashing wallet addresses into a pair of indices, each transaction record or PoSR can be hashed into a pair of Time Division Network Channels for consensus, and afterwards the record is saved in a pair of SoCo blockchains. These SoCo chain records are then broadcasted to the entire consensus overlay through all candidate rings. Second, from the consensus overlay to the service overlay, nodes are vertically hashed (based on owner wallet addresses) and partitioned into user channel overlays. Each individual SoCo chain record needs only to broadcast to its matching user channel overlay, thereby greatly improving service node storage scalability.

The contract nodes execute ContractBots, audits PoSR and resulting transaction records, and submits for SoCo nodes to execute BFT-TDPOS algorithm towards consensus. An execution of a ContractBot is a session of ordered dialogues, producing transactions in such an order that if any earlier transaction fails auditing because of insufficient funds, it will abort the entire session.

The social chain builds up PoSR (Proof of Service Record) from a variety of algorithms: proof of authorization, proof of accounting, proof of access, proof of store, proof of presence, and proof of relay.

The SNP and social chain are the infrastructure of the next generation's social networks. All SNP-enabled APPs establish a win-win decentralized social networking ecosystem. SoCo is the official token used in this ecosystem to promote and sustain ecological operations.

The SNP and social chain will create a brand new business model that adds value to individual social assets, and will bring a tremendous expansion capacity to ecological organizations based on social assets.

Table of Contents

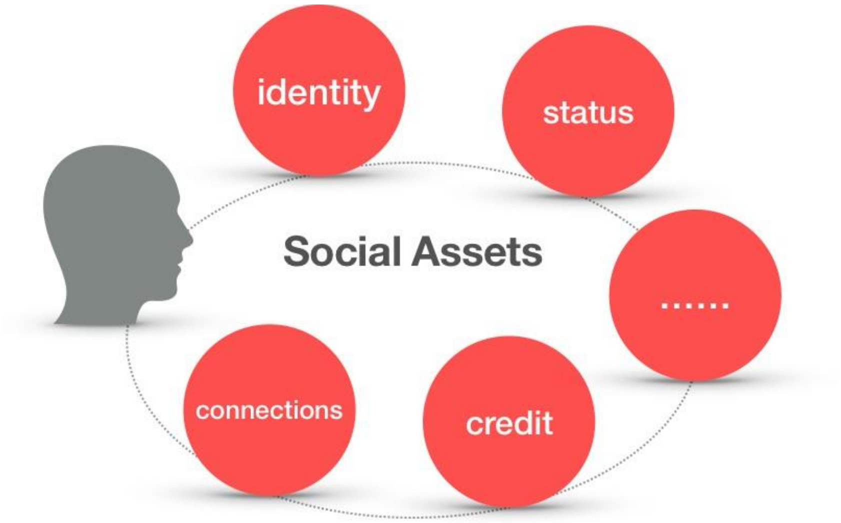
1. Background

“People are social people; humans are social animals.”

All human interactions (i.e. discussing idease) and value exchanges (i.e. purchasing goods and services) are accompanied by social activities. The Internet has made our social interactions richer and more efficient, as social spaces are no longer limited by space or time. Since its inception, social networks have become the center of the Internet: they are a gathering place for user traffic and user behavior, and a place for user asset and information exchanges. More and more applications are relying on social network development. The major explosion of Internet socialization has created extremely rich and invaluable social data, which is the reason for social assets.

1.1 What are Social Assets?

Social assets are intangible assets people generate in their social activities and social relationships. Social assets include one's identity, status, connections, credit, etc. One's social assets depend on one's own abilities and social networks.



Social networking is the main stage where people complete their asset transactions. Therefore, social assets can be transformed into personal wealth and all kinds of tangible assets. They are important resources that everyone maintains and develops for life.

We can refer to social assets as the third category of assets, relative to tangible assets (the first category) and IP intangible assets (the second category).

- The first category of assets: Tangible assets, including factories, equipment, fixed assets, etc., are the most traditional assets. Around 1970, more than 70% of North American companies listed their assets as tangible assets.

- The second category of assets: intangible assets, centered on the relationship between people, creativity and knowledge, include trademarks (brands), patents (technology), copyright (creation), and business secrets. Around 2015, more than 70% of North American companies listed their assets as intangible assets, most of which being intangible IP assets.
- The third category of assets: Social assets, intangible assets that surround a person's social relationships in society, including identity, status, connections, credit, and so on. This is one of the oldest human asset categories in the world; however, due to social and technological reasons, it has been hidden, difficult to estimate and unable to be confirmed.

Common digital social assets generated by the Internet include:

1. Various attributes of one’s personal account profile, such as gender, age, region, hobbies, tags, status, etc.
2. One’s social relationships, contacts, and groups, including friends, colleagues, interest groups, fans, etc.
3. One’s social behavior and generated data, including chat history, published photos, videos, and text, live broadcasts, shared content, comments, praises, etc.

1.2 The current state of the Internet’s Social Networks and Social Assets

For thousands of years, people have been consciously and unconsciously using their own social assets, but due to technical limitations, these behaviors could only be carried out offline.

However, in recent years, with the advancement of science and technology and the popularization of the mobile Internet, people's social behaviors have shifted more and more towards online spaces, gradually but effectively moving their social assets online as well. This movement has made social interaction more efficient and has promoted the trading of other assets and expanded the definition of social assets themselves. For example, applications involving e-commerce, group purchases, news distribution, etc. have become more convenient to use with less transaction issues.

Here is what’s happening on today’s social networking platforms:

Each user submits his or her social assets to the Internet platform as individual nodes. Some examples of such social assets are personal data, friendships, social media content, social interactions, and storage on social platforms.

With each new user interaction, the social networking platform becomes larger, yet the social networking platform reaps benefits independently of its contributors, the users.

Users between social networking platforms are not interconnected, and thus users are separated into their own islands.

Other applications rely heavily on social networking platforms to gain traffic and improve user interaction experience.

The scenario above results in two consequences:

The monopoly of traffic: it is difficult for all new applications, not only social applications, to grow organically.

Users' social data value monopoly: the income generated by users' social assets becomes the platform's profit.

Social Assets going Online
but Users CAN'T Control

Social networking platforms are occupying users' social assets, and using their powerful traffic and profit monopolies to maintain their own status and interests, rather than serve developers and individual users. The power gap between users/ developers and platforms is becoming increasingly wide, and the application and user ecosystems are deteriorating.

We believe that the data within social assets have important value and significance to individuals. Like other private assets, social assets should be deemed personal and sacred. However, individuals cannot currently control and protect their own digital social assets, and thus their social assets cannot be effectively evaluated.

Now, for the first time ever, the development of blockchain technology enables us to confirm and trade social assets online.

Now is the time to change everything!

2. Vision

Our vision is to build the value ecology of human social assets with Social Coin (SoCo) through Social Network Protocol (SNP) stack, over a new blockchain 3.0-powered Social Chain, to discover, appreciate, protect, authorize, and trade everyone’s social assets.

For the first time in human history, the social networking community is exposed to this great opportunity to promote the rising value of social assets, which are comparable to and even exceed the prevalence of traditional tangible assets and intellectual properties.

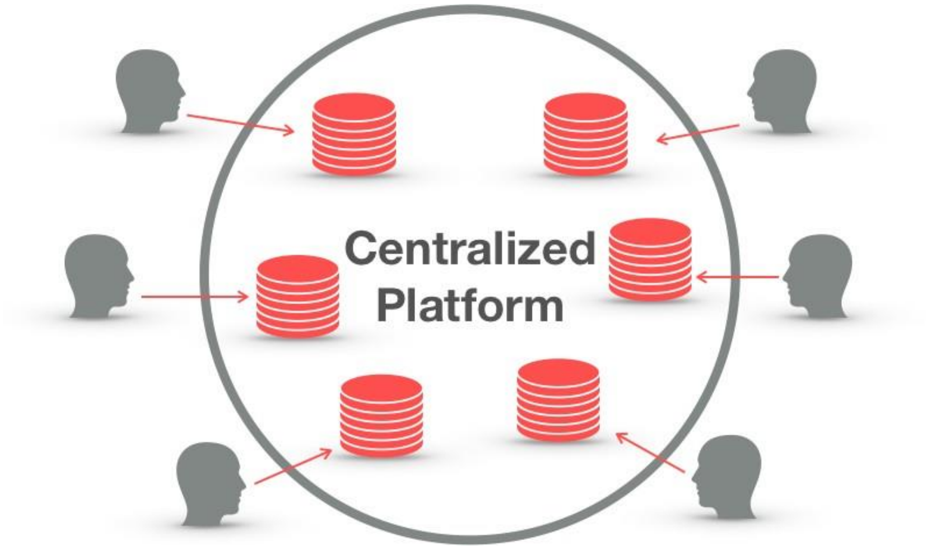
In 2015, the total amount of real estate in the world was about 200 trillion U.S. dollars, while the total amount of social assets and their corresponding executed transactions added up to multiple trillions of U.S. dollars.

The SNP will help people use and cash out (realize) their social assets online in the following ways:

- **Discovery**: The SNP will promote tools and mechanisms to discover people’s potential social assets.
- **Availability**: Social assets are freely available in a variety of scenarios in a user-controllable fashion.
- **Authorization**: Users own their social assets, which are exclusively for their own use, and any third party usage will require authorization from the user.
- **Appreciation**: Through SNP, users can have their own social assets appreciated and obtain financial benefits in return.

3. Existing Issues

Current popular social networking platforms are all products of centralized thinking. Each user, as an individual node, hands over their own social assets to the centralized platform, resulting in the following three issues:



- Misplacement of ownership of social assets
- Traffic Monopolization of Centralized platforms
- Isolated islands formed by users' Social assets

These three issues have brought inconvenience and loss to both individual users and application developers:

3.1 Misplacement of ownership of social assets

Social assets should be treated as private property, sacred and inviolable, but are now possessed by centralized platforms.

Although users' social assets are, to a certain degree, protected by law, the reality is users can only pray that platforms will comply with laws and hope that the platforms will "do no evil." However, it is not uncommon for centralized platforms to act out of their own interests and generate huge profits by analyzing the social assets and behaviors submitted by a large number of users, creating a scenario with asymmetric information. The value and influence produced by the social assets of these tens of millions of users are, however, not available to these individual users. What's more, in these value discovery and generation processes, the centralized platforms sacrifice user privacy and ownership of social assets without the users' knowledge. Individual users cannot confirm their own social assets and realize the profits they deserve.

3.2 Monopoly of Centralized platforms

The more users available to contribute their social assets to the centralized platforms, the more power and influence the platforms gain to gather more users, eventually resulting in social networking monopolies.

Among the existing isolated islands of social assets, monopolistic centralized platforms have gained their great monopoly power because they possess the social assets belonging to their users, and they exclusively use these capabilities to maintain their own status and interests (rather than foster developers' and users' interests).

APP developers can only rely on these centralized platforms, and must therefore comply with the rules of the platforms' interests.

Furthermore, the users' needs are not well met, since they cannot choose their favorite applications, and are instead limited to the platforms' selection of applications. Additionally, new APPs are very difficult to grow, due to the platforms' traffic monopoly.

3.3 Isolated islands formed by social assets

Due to the lack of universal technical protocol support, social assets are separated from each other and have become isolated islands, and therefore cannot realize their own due values.

Social assets such as identity, status, connections and credit are intrinsically interconnected, but the current system artificially divides them into different application areas that are controlled by different agencies.

The lack of information communication among agencies and APP developers is creating isolated islands of social assets and allowing for a lot of duplication of efforts (overlapping work and repeated information gathering, which is very inefficient).

Moreover, a large number of core social assets have been omitted. The total collection of social assets from all institutions is much smaller than the total amount of human social assets. The records of all transactions brought about by social exchange, which are an integral part of the expansion of social assets, are also scattered throughout various trading platforms, E-commerce, media, recruitment, leasing and other applications.

We cannot help but ask: should social assets really be controlled by a few big companies? Should social assets and their resulting transaction records remain fragmented and scattered?

We believe that since social assets are, at their core, a form of sacred private property, it is essential to give the ownership of social assets back to individuals. That is why we created a new social currency. The goal of social currency is to "create the next generation's decentralized value ecology of social assets," whose core philosophy is "to bring ownership of social assets back into everyone's hands, to prosper deals and business models, and to fully realize the complete potential of human beings' social assets."

4. Our Solution

We believe that each user is the root of his or her social network. The user is the link between social relationships, the input source of digital content, and the real source of digital traffic. With our solution, new applications can use rewards to incentivize users into using their social assets. This will carry out valuable social activities that will decentralize current social networks, compelling friends to join new applications, forming multi-application centers, and achieving interconnected social networks.

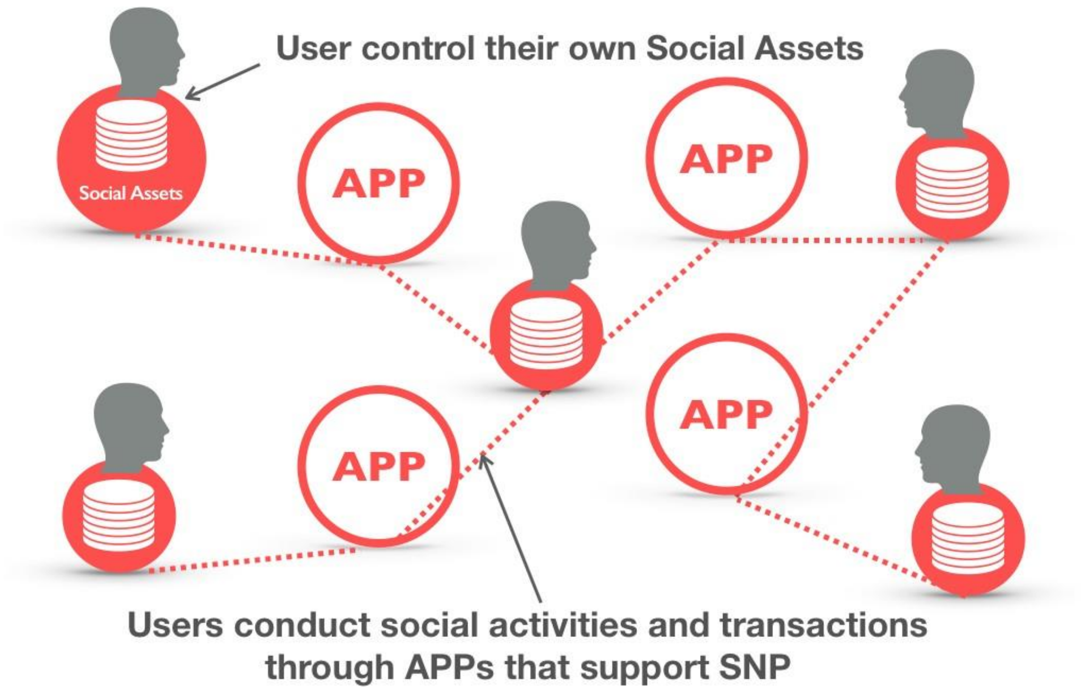
Without the support of blockchain technology, there are already a lot of things APPs are offering: features such as inviting friends, news viewing, giving cash rewards, etc. With the current social network monopolies, the high-traffic centralized social network is introduced to the user, and the user is benefited with the aforementioned features. However, in the absence of blockchain technology, the user's assets and interests are difficult to protect and it is easy to form a new monopoly. The mutual trust between applications is also difficult to achieve.

As its foundation is in blockchain technology, SoCo and SNP provide a set of social networking protocols that enable APPs to have social capabilities, allowing users to communicate through smart contracts and helping APPs spread their own social relationships and gain SoCo incentives. This will improve user trust and trust between APPs, ultimately transforming the current system into a healthier, interlocking social network application and user ecosystem with each APP as a node.

4.1 Social Network Protocol

Social Network Protocol (SNP) is an open and extensible application layer protocol. It realizes the confirmation of personal social assets through blockchain technology, and allows all APPs to have social capabilities so that users can exchange Social Coin (SoCo) to use and achieve realization of their own social assets.

The SNP uses a series of blockchain-based distributed protocols to create a wallet of social assets for any SNP-based APP user. Only users can manage their own social assets through private keys. Any other parties, including the SNP itself, cannot access or operate users' social assets, such as user information, friend relations, and social digital content. Any SNP-based APP is able to allow its users access to its social functions. While it is unable (and does not need) to take ownership of its users’ social assets without their consent, it can be facilitated


User control their own Social Assets
Social Assets
APP
APP
APP
APP
Users conduct social activities and transactions
through APPs that support SNP

when the users are using their own social assets in doing business with the social APP (through APP smart contract trading and APP's business logic), therefore allowing users to monetize their social assets.

Simply put, "autonomous and controllable monetization" of personal social assets is the core tenet that SNPs and Social Coin (SoCo) bring to APP users and developers.

4.2 Social Asset Wallet

We will develop a personal social asset wallet based on SNPs to help users acquire, manage and monetize social assets, as well as manage Social Coin (SoCo), our social asset token.

4.3 SNP APP Ecology

SNP allows any 3rd party to design their own APPs so that their users are able to enjoy the capabilities supported by the SNP social networking protocols such as finding good friends, friendships, distributed messaging and other agreements. SNP allows such APPs to have decentralized social functions while using users' own social assets. This will greatly enhance the efficiency of user transactions and interactions with the APPs, and help users monetize their social assets in the APPs through Social Coin (SoCo) issued by SNPs. Thus, instead of cashing in on the APPs themselves, the APPs are focused on monetizing users' social assets.

The following APPs could enable their users to have social functions via SNP, making it easier to trade between internal users:

- Social Entertainment: social games, strangers making friends, live, hobby communities, knowledge sharing, etc.
- Electrical businesses: second-hand resale, leasing, micro-business, micro-store, group buy, p2p, etc.
- Finance: loans, red envelopes, insurance, and more
- Sharing economy: such as bed and breakfast rental, carpooling, flash delivery, and so on
- Other types of APPs that require social interactions.

All of these APPs cannot centralize or possess the social assets belonging to users, as they are all passive nodes that cannot become the center of users, and cannot affect other APPs. As a result, the entire ecosystem is platform-symbiotic, more open, and more advanced than the current centralized social network. These APPs will become user-centric, continually helping users enrich their social assets, providing a more valuable social interaction foundation for users on different APPs, facilitating transactions and allowing users to monetize their social assets.

For example, if a group-buy APP requires the functions of friend recommendations, friend comments and friend chat, it is entirely possible to use the SNP to enable users to obtain these social functions and use their own social networks without allowing the APP access to the users' social assets (such as a contact list). As you can see, the difference between SNPs and today's centralized social networks is that users have complete jurisdiction over their digital social assets and maintain their right to cash out of an APP as well. All of these APPs' access to users' personal digital assets are user-aware and controllable. Users can access their social assets through an SNP within an APP, interact with or trade with their friends or others through the business logic of the APP, and the APP has no knowledge of personal data (such as social relationships, users' own social profiles, etc.) belonging to any users, while still providing the users with complete social features to improve the transaction efficiency of the APP itself. SNP-based APPs can also interact with each other through SNPs to form a social networking application ecosystem. In this ecosystem, each user is his or her own center, and thus social assets do not belong to any one APP and all APPs are equal. In such a decentralized environment, every user can monetize his or her own social assets.

Furthermore, SNP Ecology will provide SoCo holders with crowdfunding systems for APP developers, encouraging developers to submit their prototypes and business plans early. SoCo holders will be able to provide their own SoCo to developers as support, to ensure that users' needs and wishes for the service are reflected, and therefore enable the users themselves to decide the direction of ecological development.

4.4 Social Coin

The SNP will issue SoCo (Social Coin), a social token that uses social tokens' economic systems to promote the formation, operation and monetization of social asset value ecology. The parties using SoCo include users, APP developers in SNP Ecology, and more.

4.5 Market Pattern Analysis

Currently, other distributed social projects on the market are aimed at replacing existing social networks, just with a few distributed technologies used at the bottom. From the application ecology point of view, the platform and the application are still centralized.

The biggest difference between this project and other projects is that the goal ofSNP SoCo itself is not to create another social network, but to provide social functions for other APPs based on the premise that “social assets are private”. SNP provides an open social protocol that helps all APPs build their own social scenes and make social networking ubiquitous. In the SNP ecosystem, social assets belong to users, so no APP is the center, rightfully placing the user as the center of their social network instead. They make full use of and extend themselves through the functions and businesses implemented by users in the entire ecosystem of social assets, eventually forming the ecological value of social assets.

At the same time, other preexisting social products in the market, whether centralized or decentralized (FACEBOOK, TWITTER, WEIBO, WECHAT, WHATSAPP, etc.), are storage areas for users' existing social assets. For these products, users can submit their own social assets (personal data, friend relationships, address book and digital content, etc.) through SNP's protocols and ecological rules into their social capital wallets and make them controllable and monetized private property. Moreover, users can invite their friends to participate in the SNP and its APP ecosystem through these other social products to obtain SoCo awards. Plus, any other social products and their eco-APPs will be fully supported by the SNP and SoCo tokens, allowing users to fully own the monetization rights of their social assets through their social asset wallets. This will provide users with new opportunities regarding social asset realization and transactions, and such social networking products will also be directly linked to SNP Eco SoCo rewards.

5. Social Coin Economy System

5.1 SoCo’s Usage

• Scene of users earning and consuming SoCo

Users earn SoCo in APPs by contributing their own social assets, such as helping advertisers spread content to their friends, helping business and commerce owners by recommending jobs or merchandise to their friends, or posting their own professional content to gain rewards.

At the same time, users consume SoCo to acquire various types of APP services. Users consume SoCo to trade with other users within the APP and purchase the services provided by the APP. For example, users can consume SoCo to create a chat group, give SoCo to join a special chat group, consume SoCo to read a friend's original content, consume SoCo to obtain a former employer's work experience endorsement in a job search APP, consume SoCo to establish friendships with the people they are looking for, consume SoCo to find professional answers to their own professional issues, and more.

• Scene of APPs earning and consuming SoCo

APPs provide services to earn SoCo. For example, when users reach a blind date in a dating and marriage APP through social assets, the APP will earn SoCo; when a carpool completes an order in a friends circle car service APP, the APP will earn SoCo; when a user applies to the APP for transaction evidence of a blockchain record, such as a loan certificate between friends, a rental certificate, etc., to prove the existence of the transaction to others, the APP can charge SoCo.

At the same time APPs also consume SoCo; for example, an app can reward users for using their social assets to invite more friends to the APP. When an APP hopes to directly obtain the user's social assets (such as gender, age, contact list, etc.) to complete its commercialization, it can pay the user SoCo. Inter-APP communications can also be set to consume SoCo; for example, if a dating APP wants to get e-commerce features and make e-commerce sales in its dating services, it needs to pay an e-commerce APP SoCo.

• Scene of SNP ecology using SoCo

SNPs can use SoCo to reward users, APP developers, developers with many users, innovative development teams, and more. In addition, SNP Ecology will provide SoCo holders with crowdfunding systems for APP developers, encouraging developers to submit their prototypes and business plans early. SoCo holders will be able to provide their own SoCo to developers as support, to ensure that the users' needs and wishes for the service are reflected, and therefore let the users themselves decide the direction of ecological development.

5.2 How to get SoCo

Project Coins: SoCo (Social Coin) is the official Token for SNP SOCO. Before the official launch of Social Chain network, SoCo is an Ethereum token based on the ERC20 standard and Ethereum Smart Contract. The ERC20 SoCo will be issued as total of 1 billion pieces of Social Coins.

The ERC20 SoCo will be transformed to Social Chain native token equally when the formal Social Chain network officially launches. The ERC20 SoCo should be burned.

SoCo fund will reserve 10%(100,000,000) SoCos as a bounty pool to reward the consensus nodes' work. For every T seconds the SoCo Chain vector concurrently produces a new block, a certain amount of SoCos from the bounty pool will be transferred to miner's address as a bounty bonus.

During the first year (0~2102400 blocks), every new block rewards 12 SoCos as bounty. The second year (the 2102401~4204800 blocks), every new block rewards 10 SoCos. The bonus will decrease annually (every 2102400 blocks) by 2 SoCos, until the seventh year, when the bounty of the new block is 1 SoCo. From the seventh year forward, the bounty of new block will consistently remain 1 SoCo per block.

To promote the Social Chain network growth, SoCo fund will reserve 10%(100,000,000) SoCos as a pool to incentivize the contribution of worker nodes that are providing the SNP services. Any worker node will obtain a computing & resource sharing bonus with their authorized proof of work. SoCo maintains some public algorithms to calculate the service fee. During the early stage, SoCo fund guarantees the profit of honest nodes by applying the incentive plan, and then the users and app providers will be free to use SNP services.

Distribution plan is as follows:

- 40% for private investors
- 15% for user registration airdrop promotion and marketing
- 15% for team
- 10% for advisory team and special community contributors
- 10% bonus pool for consensus nodes bounty
- 10% bonus pool for incentive worker nodes bounty

ETHs raised from private investor round is to be used as follows:

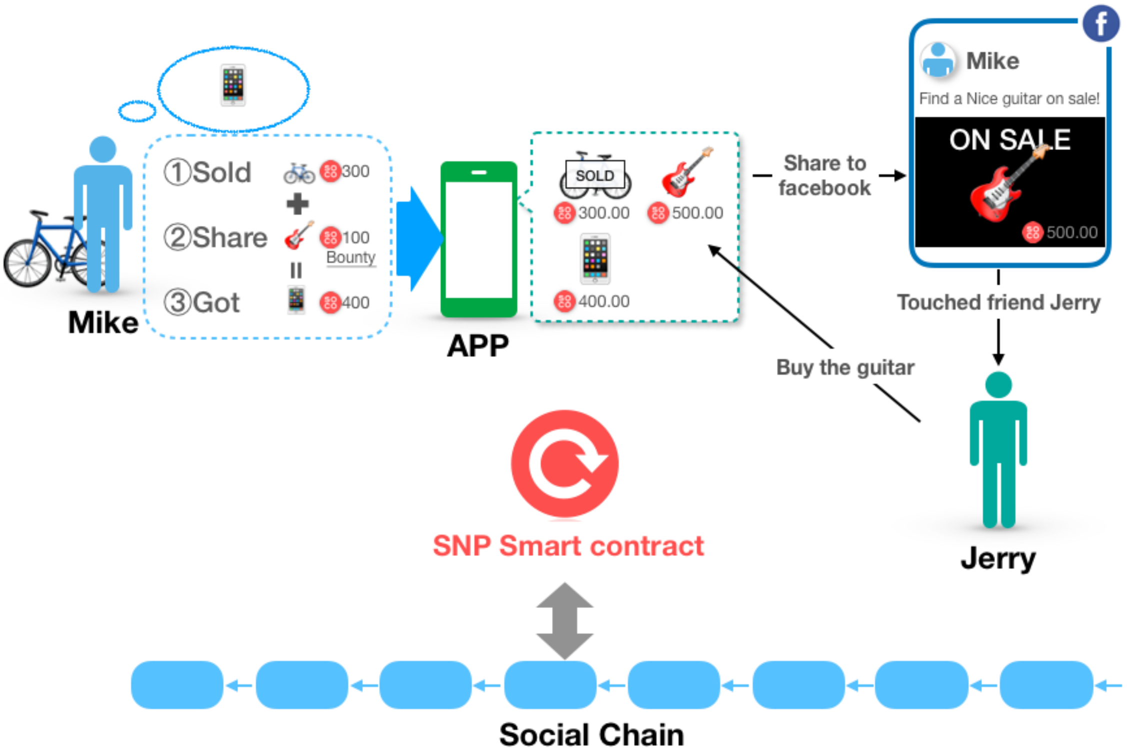
- 40% for core development of SNP SoCo Protocols, including costs of technical research and recruitment of talent
- 30% for SNP operations, including various operating costs such as hardware and software
- 20% for marketing and collaborative ecosystem, enabling users and APPs to understand and use SNPs and SoCo tokens
- 10% for legal, accounting and other advisory service

6. Use Cases

6.1 Secondhand Trading

Mike had an unused bicycle, and wanted to trade it for a secondhand iPod he liked.

- Mike opened an SNP-based APP dedicated to selling secondhand items, and used his fingerprint login authentication to post he was selling his bicycle at 300 SoCo.
- Next, Mike found an iPod that he liked, which costed 400 SoCo.
- Mike sold his bike for 300 SoCo, but still needed 100 SoCo to buy his desired iPod.
- Mike discovered a referral reward feature in the APP. Thus, he found a nice secondhand guitar and shared it on his Facebook page. His friend Jerry saw it via Mike's Facebook post, and decided to buy the guitar through the APP. Mike earned 100 SoCo through his referral, bringing his SoCo total to 400.
- Mike can finally afford his secondhand iPod, and buys it with the SoCo he's accumulated.



6.2 Social job search

MM Company urgently needed to hire an accounting director, and posted the recruitment info through a job search APP. Mike saw the post and found his friend Peggy was shown as looking for a job. Moreover, she was a good fit for the job posted by MM Company. Thus, he recommended Peggy to MM company. Through the job search APP, MM Company saw Peggy's true past experiences and felt that Peggy basically met the requirements. Immediately, MM company initiated an invitation to Peggy, and the two sides agreed to an initial online interview. (The job application charged 300 SoCo from MM and sent 10 SoCo to Mike as introductory rewards). After the interview, both parties had the intention of recruitment. MM again saw the remarks of Natalia, a former colleague from Peggy's resume, and asked him about Peggy's actual work through the APP. Natalia answered MM Company's five questions (and MM Company paid Natalia 10 SoCo as reward). In conclusion, MM Company found the right employee through the SNP-enabled APP, with both Mike and Natalia realizing their social assets.

6.3 Social lending

Mike posted a $10,000 request in a social lending APP, and his friend Tom helped him forward it to his friend Linda. Linda then agreed to lend Mike $10,000 for two months. The two parties signed an agreement on the loan APP. If one month later, Mike repays all the borrowed money at the appointed time, the repayment information order would be generated at the Standard-Citi Bank, and the loan relationship between both parties would end. If Mike does not return the entire loan at the agreed time, a public and permanent credit record would be generated in Mike's credit history. In the future, Mike's SoCo will be automatically reimbursed to Linda's account until the equivalent payment is returned. At the same time, Tom helped Mike achieve his borrowing needs by providing his own social assets, so he earned a SoCo reward from Mike and the APP.

6.4 Social matchmaker

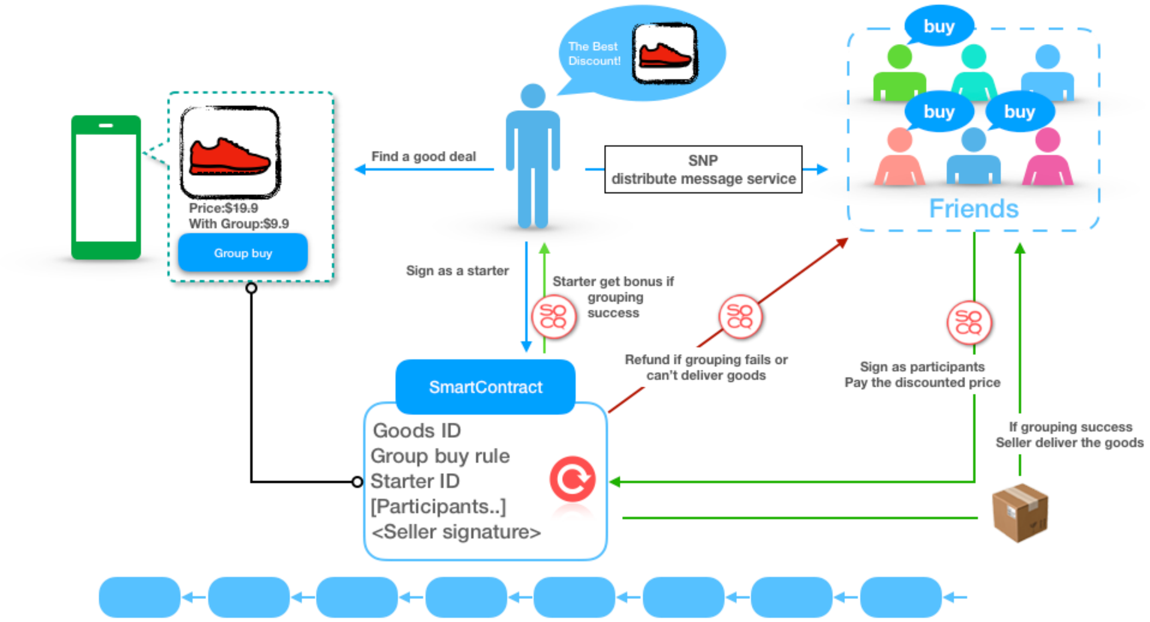
James wanted to find a girlfriend via a blind date APP. Because all information is based on the blockchain for real-name system uniqueness verification, all the member information is true and read-only, thus greatly increasing the blind date success rate. James posted the request of his ideal girlfriend; his friend Louis saw the request and recommended Ada. Ada and James were satisfied with each other.After the first date, both parties felt they had found a good match, so they evaluated each other in the blind date APP. As a result, Louis earned ten SoCo as a match-ups bonus and realized his social assets.

6.5 Knowledge realization

This is to integrate knowledge sharing platforms like zhihu, Google answer, Yedda and other similar APPs. Users can recommend experts they know to the users who are asking, the experts with accepted answers will earn SoCo rewards, and the referrers will get SoCo rewards, too. This will enable all the economic benefits of professional experience to be realized via social assets. Ultimately, the weight of knowledge in society will be triggered out in an explosive manner.

6.6 Social Group Shopping

Social recommendations have become more and more popular for E-Commerce nowadays. Group shopping has been attractive to both stores and shoppers. In the past, there has been no decentralized application that can group the same interests of shoppers.

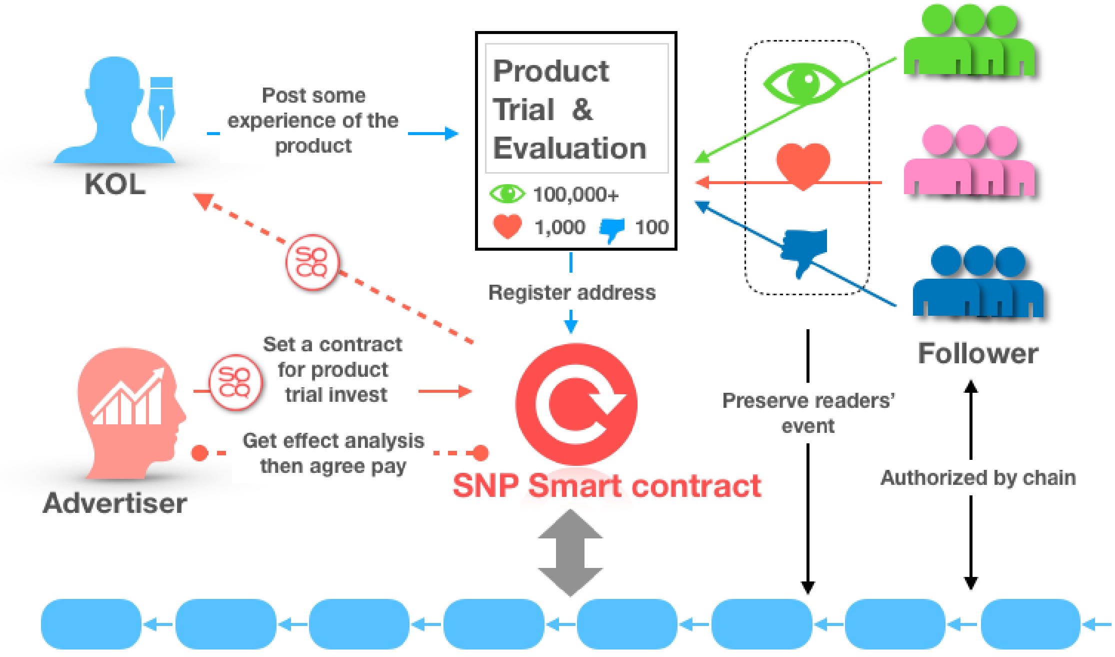


The social chain now effortlessly enables group shopping applications. E-Commerce app developers can set up a smart contract for a shopper who invites his or her friends: for example, they could receive a huge discount for a large volume purchase for some merchandise. The shopper initiates sending this merchandise through this application in a message to friends. The contract indicates how much SoCo bonus the shopper would earn and the discounts based on the purchase order numbers before the deadline.

When the deadline approaches, friends complete the group shopping order, and the initiating shopper will get the bonus according to the developer's SNP-based smart contract.

6.7 Social Advertisement

The social chain will be an effective way to spread advertisements, especially for small businesses.


KOL
Post some experience of the product
Product Trial & Evaluation
100,000+
1,000
100
Register address
Follower
Set a contract for product trial invest
Get effect analysis then agree pay
Advertiser
SNP Smart contract
Preserve readers' event
Authorized by chain

A small business as an advertiser can set up a smart contract to specify the content of an advertisement, keeping these elements in mind: identification of a relay miner, a decision on how much SoCo an audience can earn if relaying to his friend or to his friend circle, and a statement on how much he'll earn if commenting on an advertisement. Any comment will be posted real time to the relay miner. The advertiser can update the advertisement at any time.

If choosing to relay, each audience sends a relay request to the relay miner. The relay miner sends an authorization request to a relevant circle miner, along with all key words for the advertisement. If the key words have little to do with this particular circle, this relevant circle miner sends an authorization denial back to the relay miner. Upon a successful authorization from the relevant circle miner, the relay miner forwards the live advertisement to a particular friend or all friends in the circle.

On a regular basis, the advertiser will observe the small contract, make SoCo payments to each relay audience, the audience who makes comments, the relay miner, and the circle miners in accordance with the advertisement contract.

These are common application scenarios where users will be provided very powerful and rich features by a variety of APPs (and their mutual interoperability) for a variety of needs. Users' assets in each APP may increase the value and utility of other APPs, and vice versa, to create an ecosystem with networking effects. For example, positive feedback, credit records, employment history in recruiting APPs, and positive reviews from colleagues would all positively impact the success rate of users on a bed and breakfast APP.

6.8 Social Leasing/Renting

John is a homeowner with several properties in Boston. He listed his three-bedroom apartment on an SNP-supported home rental website, indicating its cost to rent and sharing rewards. His friend Mike felt the house was a good deal and helped share it to his Facebook page. Fortunately, Mike's friend Jerry had recently decided to move, but did not need a three-bedroom apartment. Thus, he sent out a sharing request on his Facebook to gather friends that would share the apartment with him. Three days later, his friends Larry and Maggie saw the offer and felt happy with the living space, so they rented the apartment with Jerry. Ultimately, because of Mike's social sharing behavior (his online recommendation), John's apartment was quickly rented out; therefore, Mike received a 100SoCo reward.

7. Technology

7.1 Introduction

7.1.1 SoCo SNP

SoCo stands for Social Coin, a cryptocurrency realized by Blockchain Technology. **SNP** stands for Social Network Protocol, which regulates how the social chain building blocks work together to reach consensus and then to save SoCo transaction and supporting data into SoCo blockchains. The **social chain** is a social network of SNP speakers, each associated with a miner to execute SoCo rules and regulations; the social chain collectively maintains a secure, healthy consensus network and makes sure blockchains irreversible.

A **SoCo Wallet** stores the public and private keys which can be used to receive or spend SoCo. The SoCo is stored decentrally and maintained in an array of publicly available ledger blockchains based on hashing payer and payee's public keys. Every piece of SoCo has a private key. With the private key, it is possible to write in the public ledger, effectively spending the associated SoCo.

7.1.2 Social Asset and SNP Proof of Service

The SoCo SNP open source project is to build up a social asset friendly blockchain OS. A **social asset** demonstrates its value (SoCo) through service. A SNP contract class specifies a mechanism how one type of service unit is turned into SoCo unit. However, how much SoCo (its market value) a type of service can translate to is dynamically appreciated by SoCo client accounts. So rules can be programmed into an individual ContractBot to lower the unit SoCo price and trigger more successful sessions, or increase the unit SoCo price and trigger more expensive sessions with the ContractBot.

Proof of Service Record (PoSR) is the key to evaluate social assets. Each type of social asset has a distinct data representation. Each type of PoSR is accomplished with a proof algorithm (e.g. proof of authorization, proof of access, proof of store, proof of presence, proof of relay and so on). SNP requires all Account and Contract classes define PoSR data structure and implement how PoSR records be built into Merkle trees, from a AccountBot server to a client. In this manner, PoSR along with resulting SoCo transaction can be saved into the same SoCo blockchains.

7.1.3 SoCo Account

SoCo **Account Class** is a user account agent class registered on the SoCo Account Chain, which implements the semantics of a user “account”. Each Account class will be subject to a series of checking in consensus and is then stored on the account chain. The Account Class defines the interface (handlers, cookies, data structures) and the code that implements the interface. The code is compiled into a canonical bytecode format that nodes can retrieve and execute. Each Account class must provide PoSR implementation.

An instance of an SoCo Account Class is called an **AccountBot** object or simply **AccountBot**. An AccountBot is deployed and executed on an SNP Node.

An AccountBot consists of a client thread called **BotClient**, and a server thread called **BotServer**. The BotClient is activated manually by SoCo wallet. The BotServer is a state machine activated by external events and governed by pre-programmed rules, thereby accomplishing service robot functionality.

A wallet can deploy and execute an AccountBot in an SNP node it owns, or lease an AccountBot that gets executed by an SNP node owned by another wallet.

A SoCo wallet can delegate to its own or leased AccountBot a subset of responsibilities that needs a digital signature. In order for this delegation, the AccountBot needs to store the wallet’s public and private keys, and any third party requesting for such a signature will need to present a privilege token acquired previously from the owner’s wallet and an authorization server.

The AccountBot will store other information such as a SoCoID, wallet address, friends list, and so on. In order to retrieve such information, one needs to present a token previously acquired from the owner’s wallet and an authorization server. Each token is used to retrieve one particular piece of information.

7.1.4 SoCoID

In traditional social networks, user accounts are identified and authenticated by a centralized authentication service. In different social products, users need to register separately in the name of these accounts to use the services and exchange information. Although there are cross-service account verification protocols such as OAuth SSO, these accounts are essentially independent and mutually immune. Therefore, in a social network, our social identity is actually fragmented because of these polycentric ties.

SoCoID is a unique URI across the Social Chain, e.g. mike@soco.social, 13701871521@soco.social, etc.

A **SoCoID Service ContractBot** is deployed in the SNP contract chain. When a wallet requests for a SoCoID with a proposal, the service ContractBot will verify the uniqueness. If not unique, the ContractBot will respond with a few alternatives. Then, the wallet requests again. This process is repeated until a unique SoCoID is accomplished.

Next, the SoCoID is subject to consensus over the contract Overlay. After consensus, the SoCoID is saved into the SoCo Account Chain.

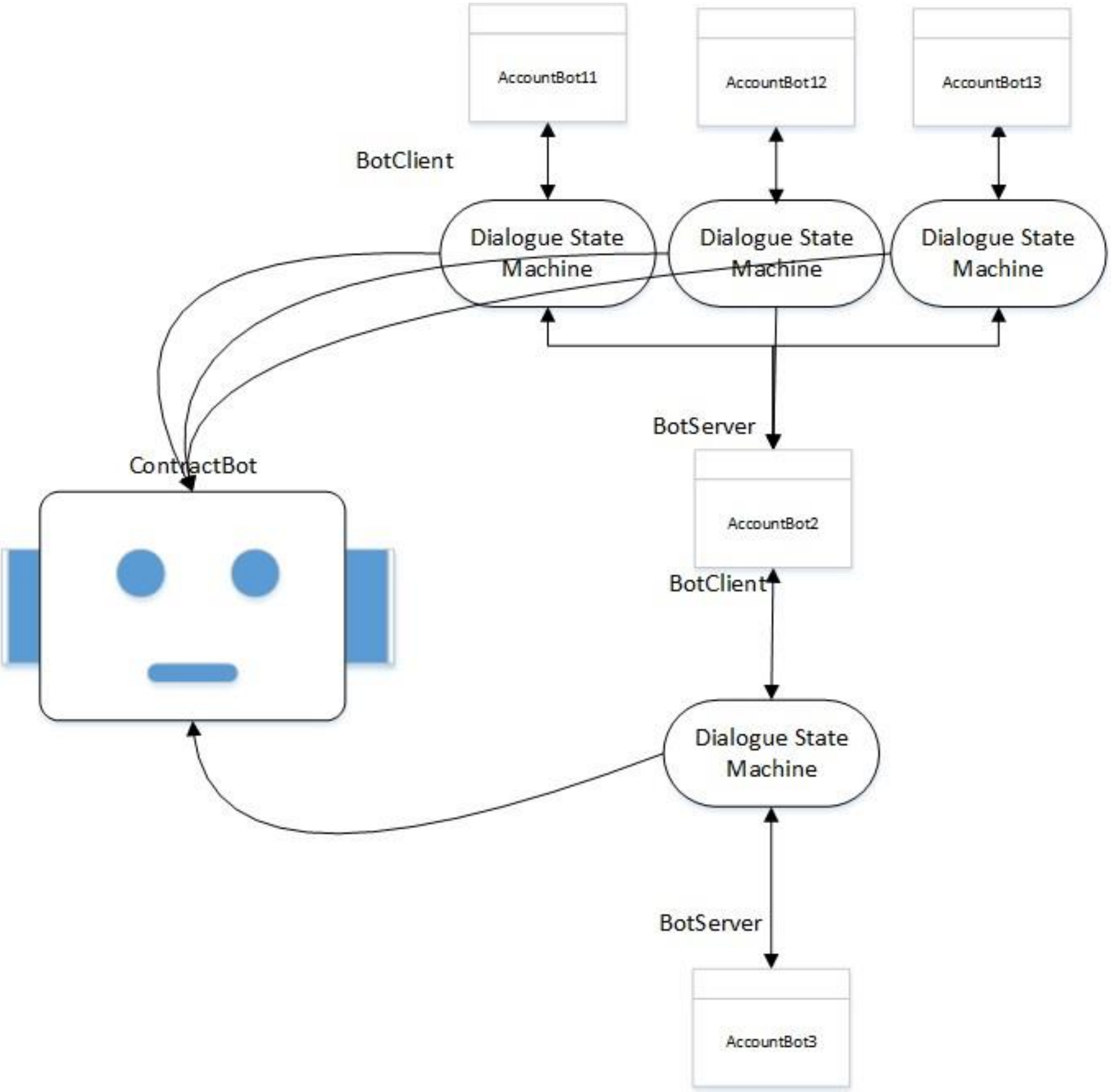
The SoCoID will be used for Social Chat over the social chain.

SoCoID token will be irreplaceable and able to be tracked and identified for verification. A SoCoID token holder can only apply (own) or destroy a SoCoID token, but cannot transfer or trade it.

Based on Each SoCoID token, users can build up basic social identity with his or her own social account with the basic profile he or she is willing to show the public to facilitate the social discovery possibilities. Social APPs in SNP ecology use SoCoID token as the basic credential for user account authentication. Any APP or business process other than the owner of SoCoID token cannot obtain access to the account, and users who are separate from the APP provider will be the true owners of social identities.

7.1.5 SNP Contract

SNP **Contract Class** is an account session class registered on the SoCoBot Chain, which implements the semantics of a “contract”. Each Contract class will be subject to a series of checking in consensus and then is stored on the contract chain. The Contract Class defines the interface (handlers, cookies, data structures) and the code that implements the interface. The code is compiled into a canonical bytecode format that nodes can retrieve and execute. Each Contract class must provide PoSR implementation.



An instance of a SNP Contract Class is called **ContractBot** object or simply **ContractBot**. A ContractBot will be executed on a contract node and its ledger of request handlers as part of the session class are being stored on the blockchain.

A ContractBot consists of one or more **dialogues** in order. Each dialogue is between a BotClient and a BotServer and a state machine is associated. There may be one or more BotClients (thereby multiple dialogues) at each level (e.g. group shopping). The ContractBot manages between each two levels of dialogues, and makes sure at least one dialogue succeeds and there is sufficient income produced from a lower level of dialogues, and then the session can proceed to next level of dialogues. A **session** is a full set of dialogues in succession from the first level to the last level.

A ContractBot allows APP to program rules between each two levels of dialogues. The ContractBot executes these rules and then determines whether to proceed next level of dialogues or the session is aborted.

Upon a successful session SNP Node and AccountBot Server will produce **Proof of Service Record** (**PoSR**) and a set of transactions in succession. The transactions are audited in the same order as the dialogues of the session. The transactions, along with the PoSR, are committed to the SoCo Chain Vector.

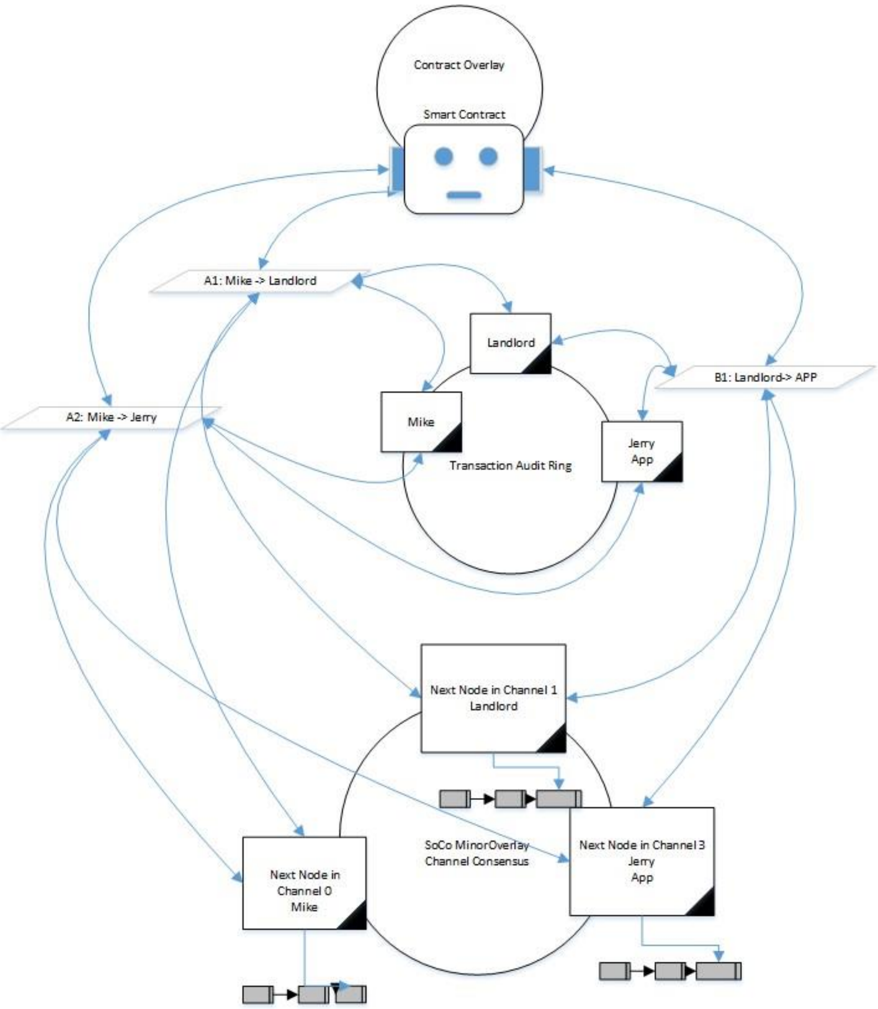
To commit, the ContractBot delivers the transactions and the PoSR to a pair of channels of SoCo Miner Overlay for consensus; the pair of channel indices are derived from hash values on payer and payee's wallet addresses. After consensus they are saved in the corresponding pair of SoCo Chains.

7.1.6 SNP Security Control

Over the last few years there were various security issues and concerns with cryptocurrencies. One reason was contracts were too powerful. To resolve this line of security issues, SNP regulates security rules in the social chain building blocks, Account class and Contract class.

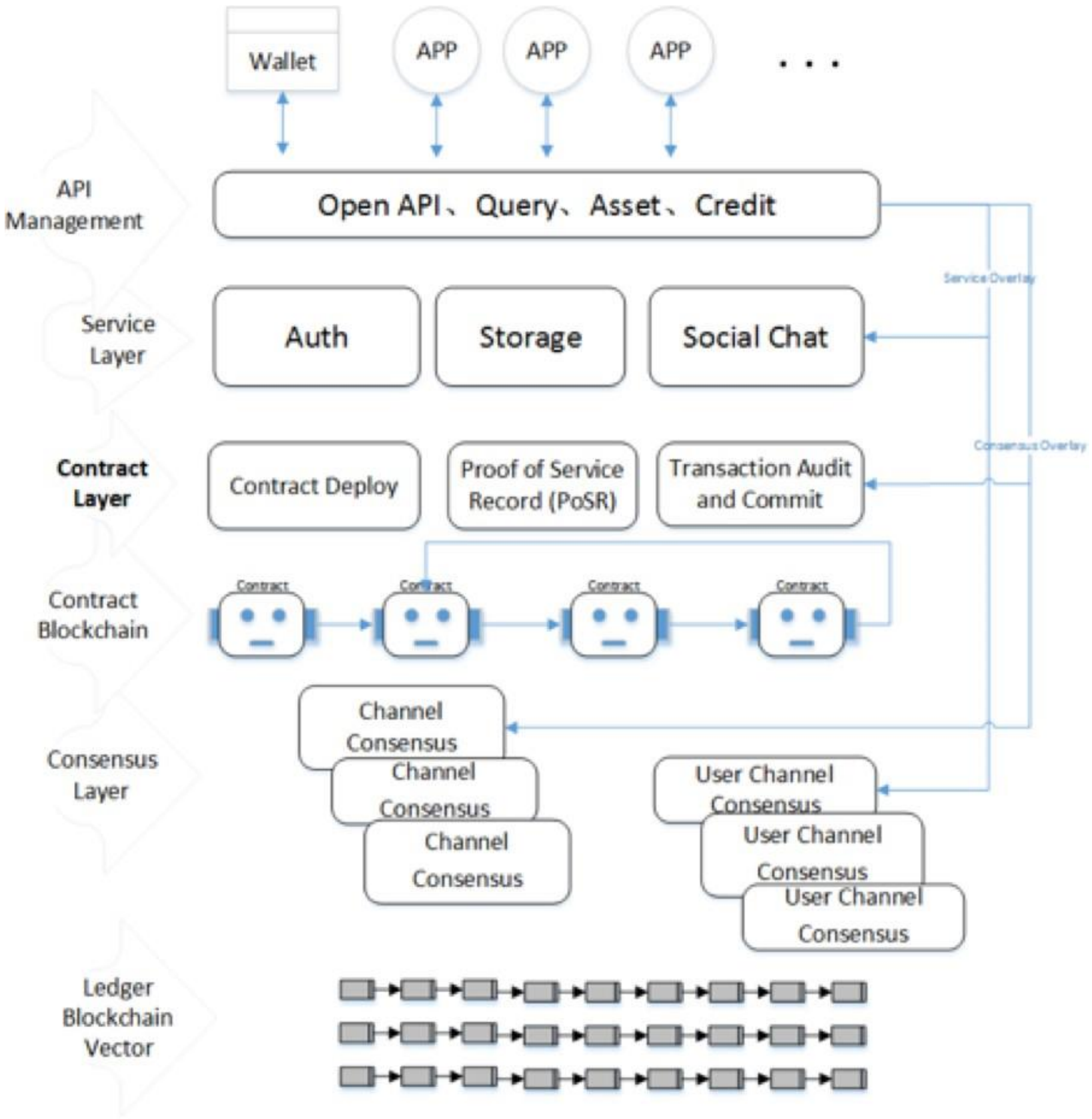
A new class as a mechanism will be subject to harsh verification process before being saved in the Account chain or Contract chain (we use 2 different chains for lookup performance). The first level of verification is compiler checking, the second is tool scanning (e.g. Coverity, valgrind and etc), and the third is complexity testing. If complexity testing fails, then the class is subject to BFT consensus from a number of voted wallets (software engineers). Individual AccountBot or ContractBot instantiated from an on-chain class is allowed to program rules but rules will be subject to additional verification consensus.

A second security control applies to ContractBot dialogues in order to avoid dialogues interfere with each other. A ContractBot can involves multiple parties and each session consists of one or more dialogues in payment order. Each dialogue is between a client AccountBot and a server AccountBot and a state machine is associated. The ContractBot manages between each two levels of dialogues, and makes sure there is sufficient income produced from a lower level of dialogue(s), and then the session can proceed to next level of dialogue(s). A successful session is a full set of dialogues in succession from the first level to the last level. Each dialogue produces a SoCo transaction.



A third security control applies to SoCo transactions in order to avoid multiple spendings. Before committing to extensive SoCo consensus, all resulting SoCo transactions must pass auditing in the same order as the ordered dialogues to avoid any attacking. Auditing a transaction is to make sure payer has sufficient fund. In order to reflect account balance in real time in a particular SoCo node, each transaction is sent to a pair of SoCo nodes, one sent to payer address hashed node (and its candidate ring) and the other to payee address hashed node and its candidate ring. As a result, each account balance can be reflected on the same node (and its candidate ring).

7.1.7 SoCo SNP Blockchain OS



The social chain consists of 4 layers, the SNP API layer, the service layer for distributed storage and social chat, the contract layer for SoCo account blockchain, SNP contract blockchain, auditing Proof of Service Record (PoSR) and transaction records, and the fourth layer for SoCo PoRS and transaction consensus, each hash-saved into a pair of SoCo chains.

To scale up and keep up performance, there will be multiple SNP blockchains.

A **SoCo Account Chain** is to store all AccountBots, which include SoCoID, public key, wallet address, and so on. It is used for SoCoID lookup and wallet address lookup. On the SoCo Account Chain, there are **Account Class Record** for account class, and **AccountBot Record** for an instance of an account class.

A **SNP Contract Chain** is to store all ContractBots, which include contract address, ordered dialogues, dialogue state machine and handlers, and so on. On the SNP Contract Chain, there are **Contract Class Record** for contract class, and **ContractBot Record** for an instance of a contract class.

A vector of SoCo Chains, called **SoCo Chain Vector**, store **SoCo transaction records** and corresponding **Proof of Service Record (PoSR).**

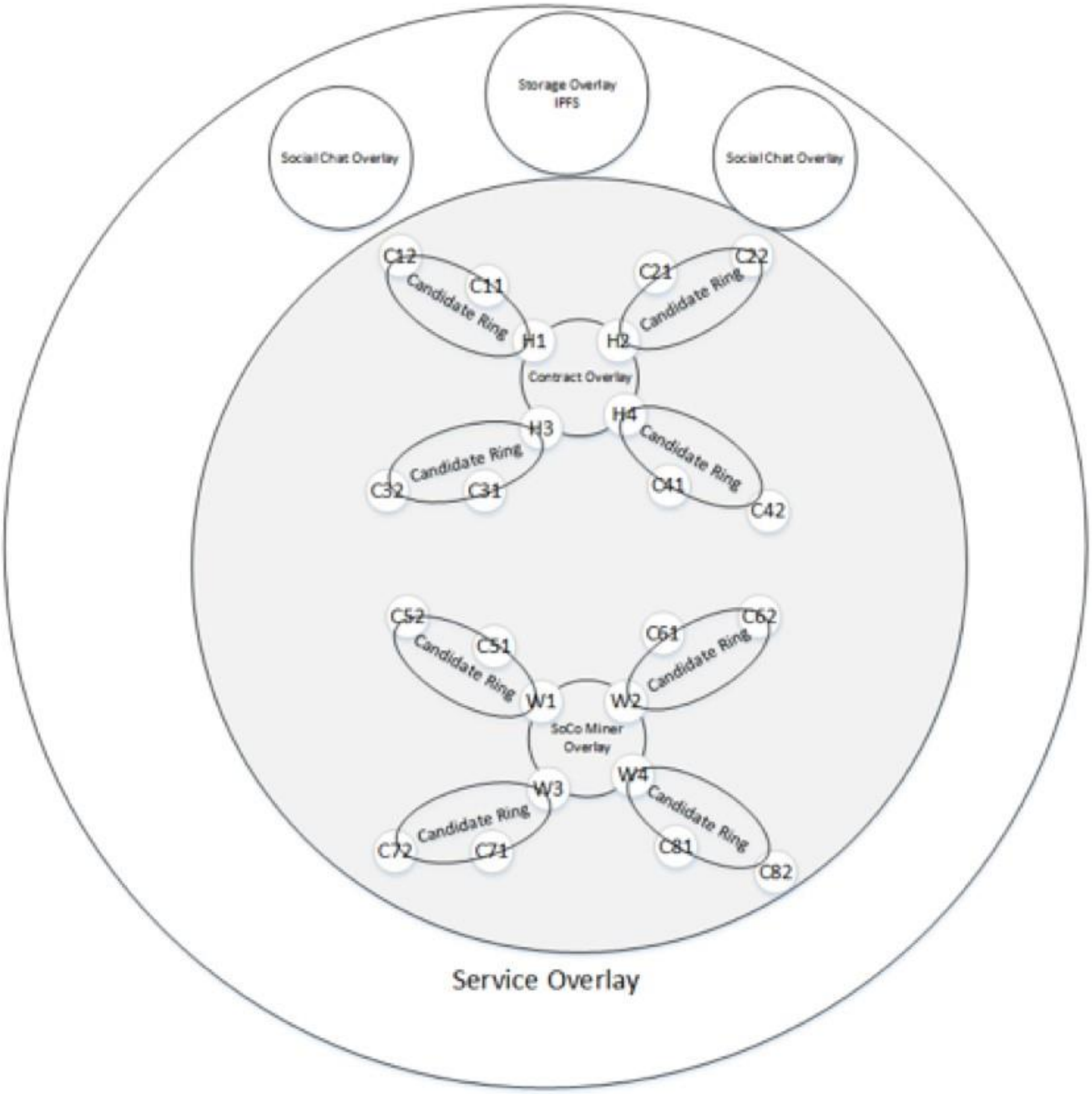
Each transaction and PoSR references a SoCo service and payment from wallet A to wallet B. Such record will produce two hashes, one on wallet A's address and the other on wallet B's address. Each resulting hash value maps to an index. After consensus, the transaction and PoSR are saved into a pair of SoCo Chains referenced by these two indices.

7.1.8 Social Chain Overlays

Over the last several years, a simple one overlay turns out unstable. Lean client nodes come and go, producing so much churns to the overlay network. Quality and capacity also differs from node to node: some have so old slow CPU, old slow memory, old slow disk and so on, while others are equipped with latest CPU, memory and storage technology.

On the other hand, there are various tasks available over the P2P network. Contract execution and consensus require fast CPU and fast preemption in order to quickly respond to external events. Producing new blocks requires clock synchronization and thus CPU frequency must be fast enough. Consensus requires a lot of CPU and network bandwidth. Some tasks are service oriented and can be executed with best of effort.

As such, there is a need to partition the overlay network in order to efficiently utilize nodes and provide differentiated service.



The social chain chooses two layers of overlays architecture, with a powerful consensus overlay in the core and service overlays on the serving edge; service nodes for regular services and consensus nodes for auditing, contract execution and consensus; election can promote a service node to a consensus node and demote a consensus node to a service node. Election is a mechanism where rules apply after a broad range of consensus from service to consensus nodes.

In the consensus overlay, all nodes can be hashed on SoCoID into two groups: higher hash values go to the contract overlay and lower hash values go to the SoCo overlay. Nodes with the same hash value form a candidate ring and they take turns to service the contract overlay or the SoCo overlay. The SoCo overlay is time sliced and all slices (channels) are exclusively used for SoCo chain consensus. The contract overlay is exclusively for contracts.

To best utilize network resource, a service node can spawn an account node, a storage node and a social chat node, and a consensus node can spawn an account node, a contract node and a SoCo node. Each type of SNP node is best customized for SNP protocol operation. All SNP nodes are ultra-lightweight containers.

SNP horizontally enforces contract execution, and vertically enforces contract API authentication and contract auditing and consensus until adding into the relevant blockchains.

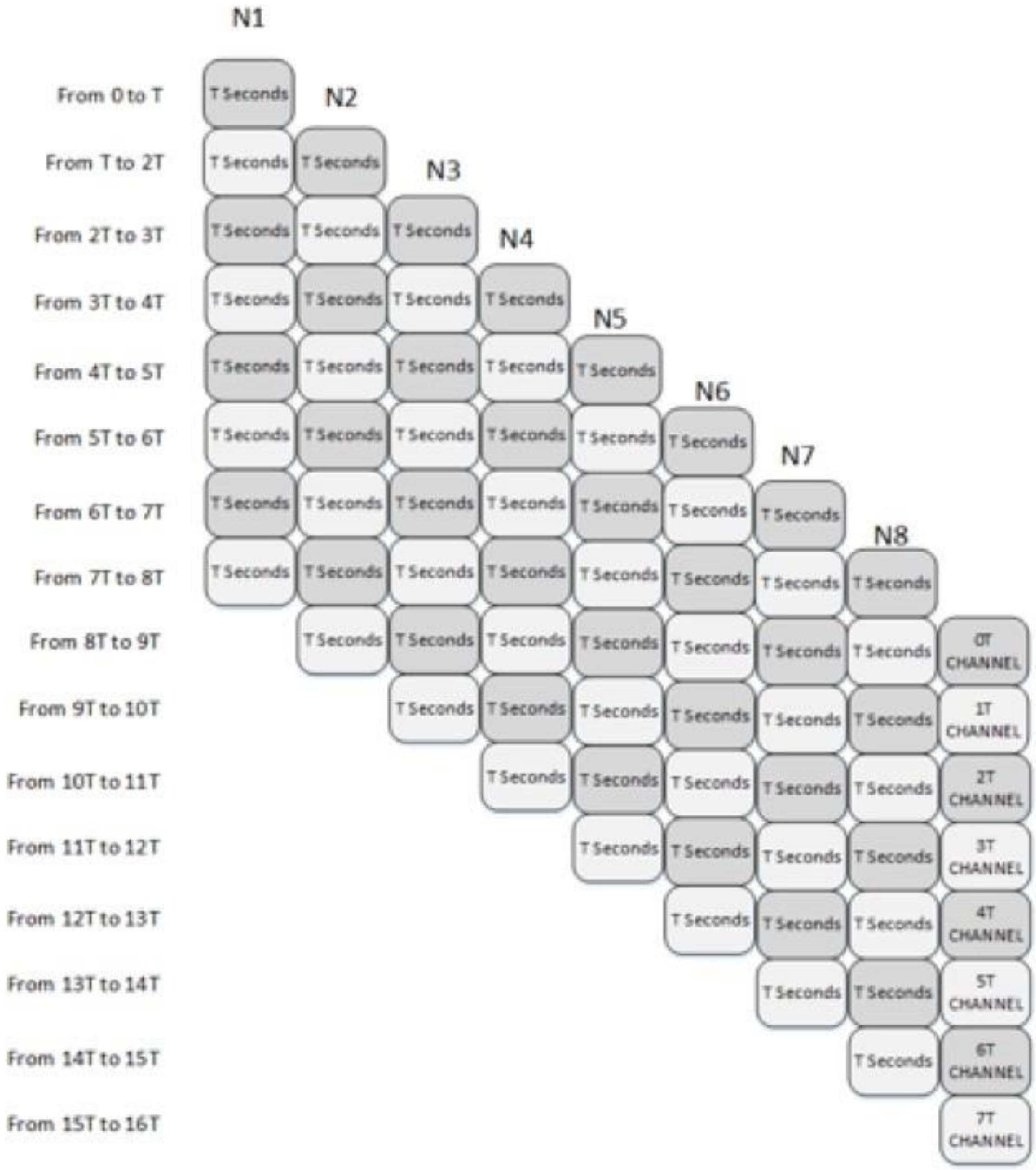
7.1.9 SoCo Chain Scalability

Bitcoin, Ethereum and other cryptocurrencies are known with single blockchain scalability. Today Bitcoin chain now costs over 200G storage which is way too much for a lean laptop. One solution on the market is to use database sharding technology; it distributes a single blockchain across multiple nodes.

The social chain chooses to use hashing algorithms to distribute transactions into an array of SoCo chains; a user lean laptop only need to store one of these

SoCo chains. The main idea is that, if I have data on a chain, then I store this chain; otherwise do not store the chain (and don’t even give to me).

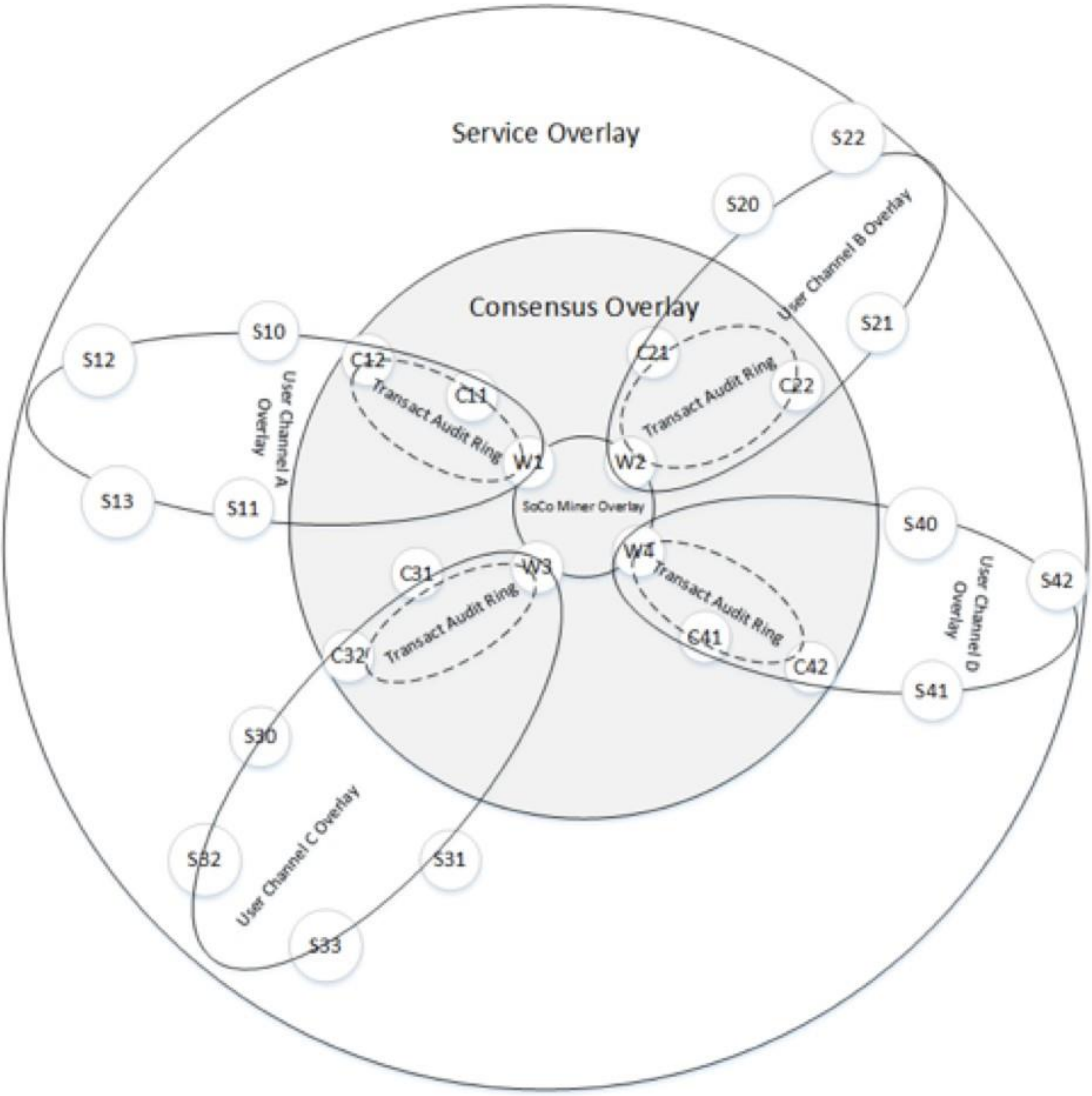
In order to accomplish this hash based SoCo chain array, SNP introduces network time division mechanism and network consensus channel as illustrated below.



In the above example, there are 8 SoCo nodes in the SoCo overlay. 8 nodes take turn to use T seconds for producing new blocks; this network serving path is one channel. If each node only services one channel, then 7/8 CPU time is wasted.

The idea is these 8 nodes produce 8 time division channels across network, each spanning T seconds. 8 channels runs concurrently, each servicing a separate SoCo chain in the vector. For each channel, SoCo nodes take turn to own T seconds. SoCo node can only produce, sign a new block and then commit to the relevant SoCo chain, or otherwise will be denied by consensus.

For each PoSR and SoCo transaction, there is a pair of wallet addresses (payer and payee). By hashing wallet addresses into a pair of indices, each transaction record or PoSR can be hashed into a pair of network channels for consensus, and afterwards the record is saved in a pair of SoCo blockchains using the same hash value.



After a new block has accomplished consensus in all SoCo nodes, it will be broadcasted to the entire consensus overlay through all candidate rings.

In order to continue to populate the new block to a broader range of lean clients (e.g., laptops) in a lean way, every node hashes its wallet address and nodes with the same hash value form a user channel overlay. A user channel overlay will have consensus nodes as its overlay member.

If a consensus node receives a new block, and it determines one of its pair hashes match a user channel overlay, it will broadcast this new block to the user channel overlay for consensus. As a result, lean client nodes in the service overlay benefit a lot and greatly reduced storage and network usage.

7.2 SNP Service Architecture

7.2.1 SNP

SNP is a stack of open and extensible application layer protocols, which authorizes and tokenizes personal social assets through blockchain technology. SNP enables all APPs to have social capabilities so that users can use Social Coin (SoCo) in APPs to have social assets appreciated.

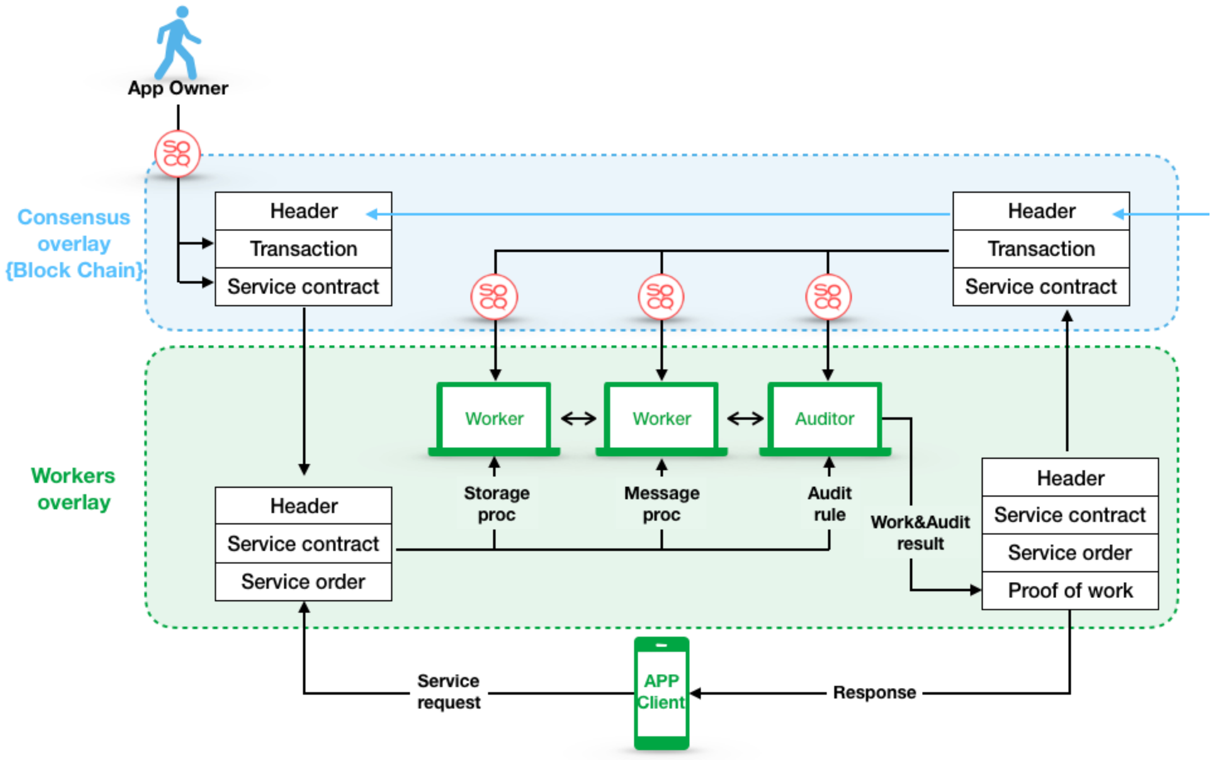
In order to accomplish decentralized social ecology, SNP integrates de-centralization technologies (Blockchain 3.0, smart contracts, distributed data storage and the like) in the underlying architecture, to create account system, distributed address book, data storage, data security, user social relations and other core services and interface protocols.

Based on the core protocol layer, SNP provides a stack of API protocols, SDKs and application frameworks to further support the implementation of various upper-layer APPs. Users use a variety of decentralized social services through applications built into the SNP technology framework.

7.2.2 SNP Service Flow

Through SNP API layer, a couple worker overlays are identified to compete. Each worker overlay provides APP with SNP server, or SNP miner, usually with a P2P DHT lookup. API execution in each worker overlay will be subject to auditing in a separate worker overlay. The auditors in this other work overlay come to consensus using BFT algorithm. Then auditor overlay then submits the contract, the work done, and the auditing consensus, to the consensus overlay. The consensus overlay then triggers SoCo implicit payment transactions from the SNP wallet who requested for the service with the contract, to the worker overlay, and to the auditing overlay in accordance with the contract. These implicit transactions are dispatched for SoCo overlay to build up a full SoCo block which is in turned submitted to the consensus overlay to reach up to a consensus and add to the SoCo chain.

Here is a high level view of the SNP service flow:

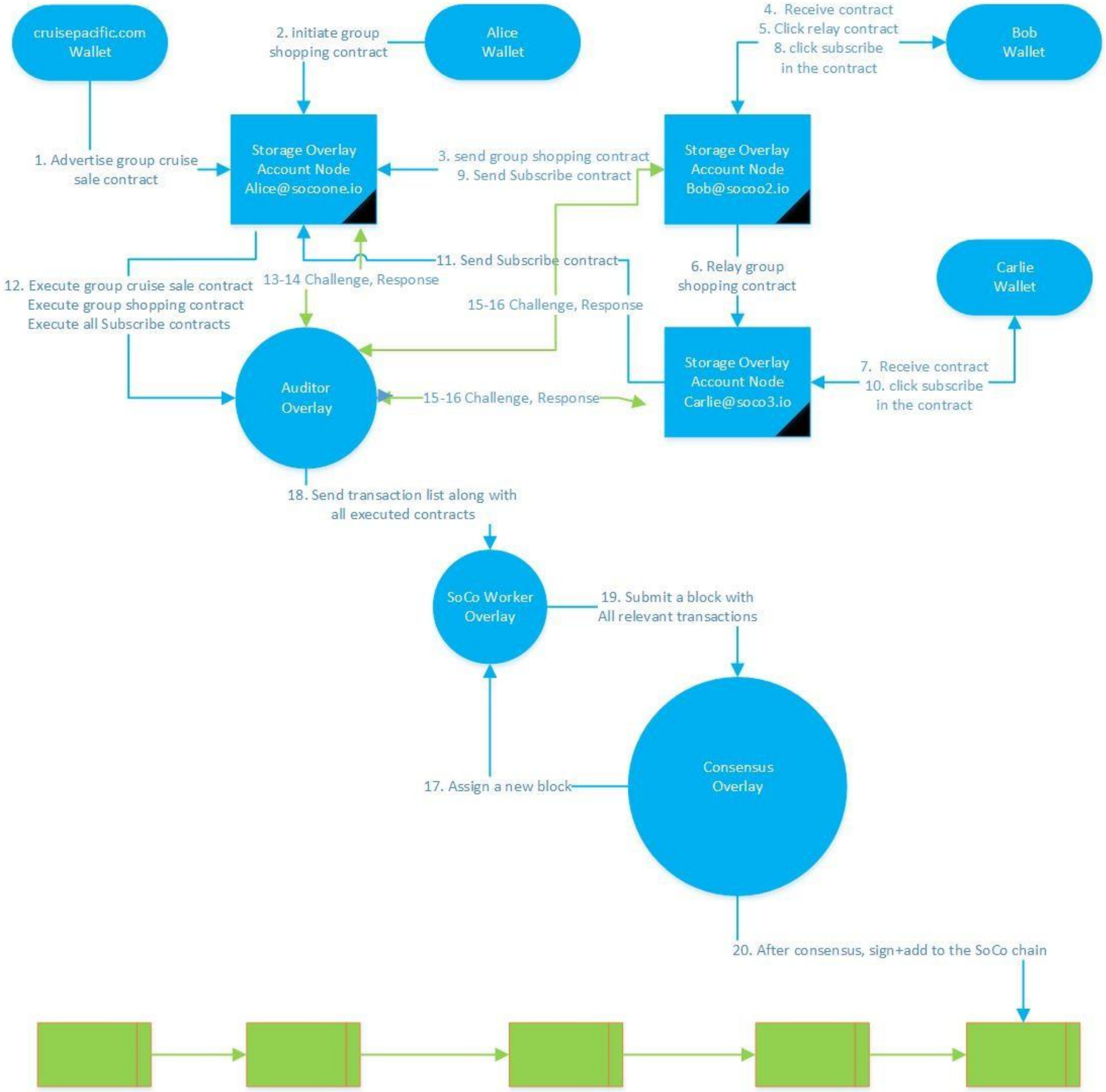


To illustrate further, take an example of group shopping cruise tickets:

cruisepacific.com sends advertisement contract (structured email) to alice@socoone.io in Alice account node in a storage overlay. Alice receives and then initiates group shopping contract and sends it to bob@soco2.io in Bob's account node in a storage overlay. Bob receives it, relays this contract to Carlie@soco3.io in order to earn the SoCo incentive for this relay, and subscribes for this group shopping which in turn triggers a Subscribe contract (which depends on the group shopping contract) to alice@socoone.io. Carlie receives it, and subscribes for this group shopping which in turn triggers a Subscribe contract (which depends on the group shopping contract) to alice@socoone.io.

Alice receives sufficient subscriptions, and requests the auditor overlay to execute cruise sale contract, the group shopping contract, and all subscribe contracts based on the group shopping contract.

The auditor overlay challenges account nodes for alice@socoone.io, bob@soco2.io, and carlie@soco3.io. These account nodes respond to the auditor


cruisepacific.com Wallet
2. initiate group shopping contract
Alice Wallet
4. Receive contract
5. Click relay contract
8. click subscribe in the contract
Bob Wallet
1. Advertise group cruise sale contract
Storage Overlay Account Node Alice@socoone.io
3. send group shopping contract
9. Send Subscribe contract
Storage Overlay Account Node Bob@socoo2.io
11. Send Subscribe contract
12. Execute group cruise sale contract
Execute group shopping contract
Execute all Subscribe contracts
13-14 Challenge, Response
15-16 Challenge, Response
6. Relay group shopping contract
Carlie Wallet
Auditor Overlay
15-16 Challenge, Response
Storage Overlay Account Node Carlie@soco3.io
7. Receive contract
10. click subscribe in the contract
18. Send transaction list along with all executed contracts
SoCo Worker Overlay
19. Submit a block with All relevant transactions
17. Assign a new block
Consensus Overlay
20. After consensus, sign+add to the SoCo chain

overlay to prove all the work they did. The auditor overlay then compiles all payment transactions from the relevant contracts, looks up SoCo Miner Overlay DHT for payer's wallet address, finds a SoCo block worker, and submits all these transactions along with the contracts to the SoCo block worker.

The consensus overlay assigns a new block to every SoCo block worker. The SoCo worker validates the transactions based on all the available contracts, builds up Merkle tree and the block with these transactions, and then submits to the consensus overlay.

The consensus overlay executes consensus algorithm, proves the block, and then signs and adds the block to the SoCo chain.

SNP will provide API for client to produce programable SNP contract. A worker node fulfills client's request in accordance with the programable SNP contract. The follow-up auditor verifies the work based the SNP contract.

7.2.3 SNP Contract Service Flow

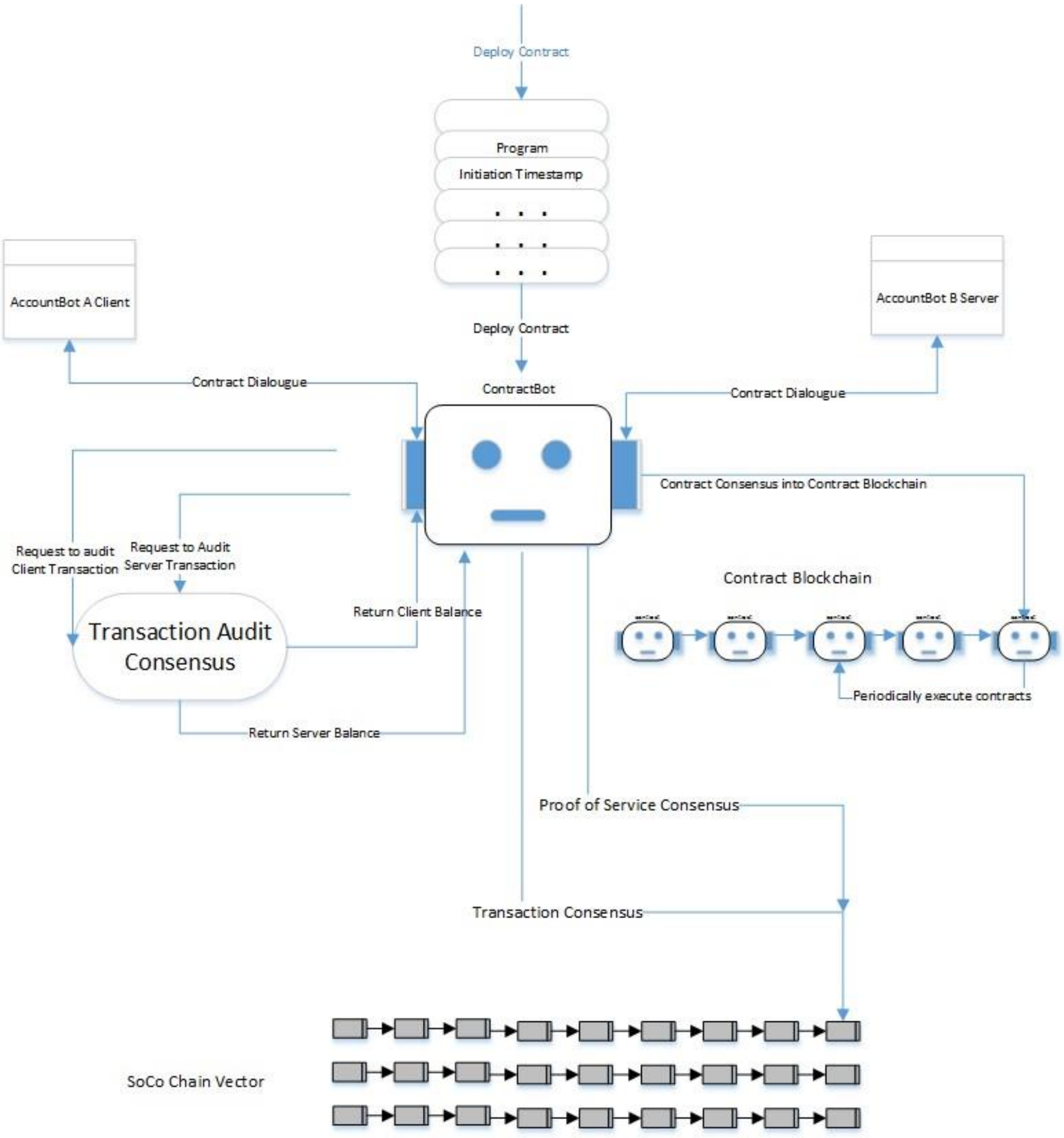
When APP deploys a new contract class, through DHT lookup this class will be delivered to a contract node and the contract node will run harsh verification process against this new class. The first level of verification is compiler checking, the second is tool scanning (e.g. Coverity, valgrind and etc), and the third is complexity testing. If complexity testing fails, then the class is subject to BFT consensus from a number of voted wallets (software engineers).

If the new contract class passes the number of harsh verifications, it will be sent to a consensus channel of the contract overlay for consensus. After consensus, every contract node saves this new class to the contract chain. The contract overlay continues to populate the class to the entire consensus overlay through all candidate rings, and all consensus nodes run consensus on the new class and save it to the contract chain.

Later when APP attempts to instantiate from an on-chain contract class, along with a set of rules. This particular ContractBot along with the rules will be sent to a consensus channel of the contract overlay for additional verification consensus. After consensus, every contract node saves ContractBot along with the rules to the contract chain. The contract overlay continues to populate the ContractBot along with the rules to the entire consensus overlay through all candidate rings, and all consensus nodes run consensus on the ContractBot along with the rules and save them to the contract chain.

A client AccounBot responds to and wakes up the ContractBot and a dialogue is between the client AccountBot and the server AccountBot. Upon a successful session is accomplished, each dialogue in the session produces a SoCo transaction.

Before committing to extensive SoCo consensus, all resulting SoCo transactions must pass auditing in the same order as the ordered dialogues to avoid any attacking. Auditing a transaction is to make sure payer has sufficient fund.



In order to reflect account balance in real time in a particular SoCo node, each transaction is sent to a pair of SoCo nodes, one sent to payer address hashed node (and its candidate ring) and the other to payee address hashed node and its candidate ring.

After all resulting transactions are audited, the transactions each in order is sent to a pair of consensus channels in SoCo node for consensus. After consensus, each transaction on a consensus channel is saved to a SoCo chain corresponding to the consensus channel.

7.2.4 SNP Service Incentive

The social chain employs a set of proof of work to evaluate the amount of work that a worker node has accomplished, which is then used to determine how much incentive would be credited to the worker node.

The ratio of SoCO to gold is changing constantly while the worker's cost is relatively stable. As such, the social chain regulates a variable coefficient, m, to offset the volatile price.

- Set w = SoCo/m as the unit of worker's incentive
- Set initial value of m to 10^3
- m is set initially by the social chain foundation when SoCo is released
- m is adjusted regularly by the social chain foundation
- With the advance of the social chain community, more rules and regulations will be developed to migrate to market driven SoCo pricing

Service proof is priced by w.

7.2.5 APP Development Framework

The SNP offers a variety of social scenes API (application programming interface), SDK (Software Development Kit), and development frameworks for development of upper-level applications, so that developers can quickly build social applications based on SNP protocol. Including the basic registration certification, authorized login, user information, address book, Social Graphic, messaging, message pushing, and more. Any APP developer can call these open interfaces to develop decentralized applications that have social capabilities and join the SNP ecosystem.

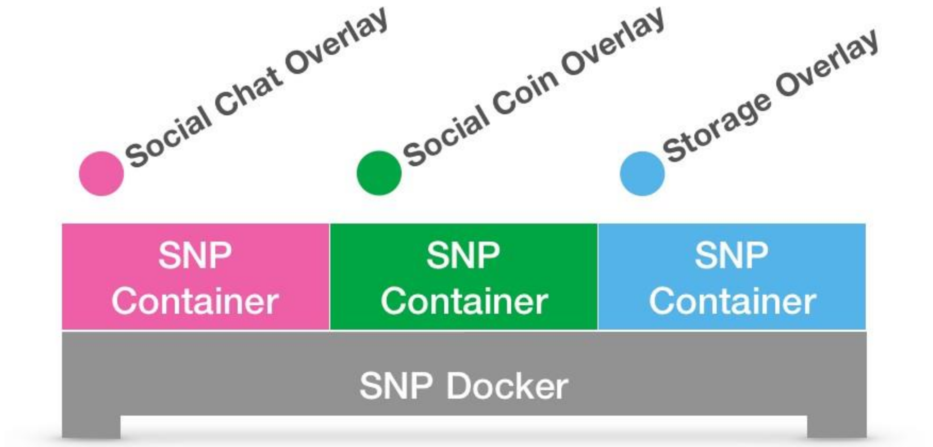
7.3 SNP Network Architecture

7.3.1 SNP Container Linux

SNP Container Linux is an ultra-lightweight container Linux and is purposely built for the social chain. Each SNP service can be distributed over the social chain as a SNP container. As such, each physical node in the social chain can be distributed and installed multiple SNP containers, e.g. AccountBot Container, SoCo Miner Container, Storage Container, Social Chat Container, and so on.

SNP Container Linux starts a special system service container called **SNP Docker**. A Docker daemon runs in this SNP container. For each SNP service, a SNP container is distributed and installed inside this SNP Docker.

SNP Container is a Linux container, a virtual instance of Linux OS with support of Linux namespaces of network, file system, process ID, user ID, and so on. SNP container is SNP friendly, with built-in SNP API support and built-in P2P protocol support. **SNP Node** is a SNP container that runs in any **SNP overlay**.



Account Node is SNP node that runs one or more AccountBots. **Auth Node** is SNP node that executes Open Auth 2.0, authentication, authorization, and provides privileged token.

7.3.2 Consensus Nodes and Consensus Overlays

Consensus Container is the SNP Docker as a container. It is capable of producing new blocks and verifying blockchain transactions. The consensus container must have AccountBot co-located in order to sign blocks. The consensus container should have sufficient CPU, memory and storage space.

Consensus Node is a consensus container that supports blockchain 3.0. Each Consensus Node keeps the full SoCo chain vector, the contract chain, the account chain, and so on. All consensus nodes are synchronous over the SNP network. A service node can be promoted or elected as a consensus node if it has a good score of reputation. The reputation score of a node is determined by the SoCo holding percentage, continuous service term, average service response time, available bandwidth, storage and CPU, and so on.

Contract Node is spawned by a consensus node whose SoCoID is hashed in higher range into an autonomous overlay, called **Contract Overlay**. Contract nodes with the same higher hash value are called **Contract Candidates** and they form as a **Contract Candidate Ring**. Contract candidates take turn to service on the contract overlay. Contract nodes execute ContractBots and contract consensus, AccountBots and account consensus, save contracts into the contract chain, and save accounts into the account chain. Contract nodes audit PoSR (Proof of Service Record) and resulting transactions, and then commit to the relevant channels of the SoCo overlay.

SoCo Miner Node (or SoCo Node) is spawned by a consensus node whose SoCoID is hashed in lower range into an autonomous overlay, called **SoCo Miner Overlay (or SoCo Overlay)**. SoCo nodes with the same lower hash value are called **SoCo Miner Candidates** and they form as a **SoCo Miner Candidate Ring**. SoCo candidates take turn to service on the SoCo overlay. SoCo nodes runs BFT-TDPOS consensus algorithm on PoSR (Proof of Service Record) and corresponding transactions, and save them into the SoCo chain vector.

Each consensus node shares the same resource as its spawned node, contract node and SoCo node:

- they use the same system clock, thereby all derived nodes clock synchronized;
- they share the same network ports if possible in order to reduce P2P messaging;
- they share the same filesystems, thereby sharing the same set of blockchains;

7.3.3 Service Nodes and Service Overlays

A **Service Node** is the SNP Docker as a container. All service nodes form a **Service Overlay**.

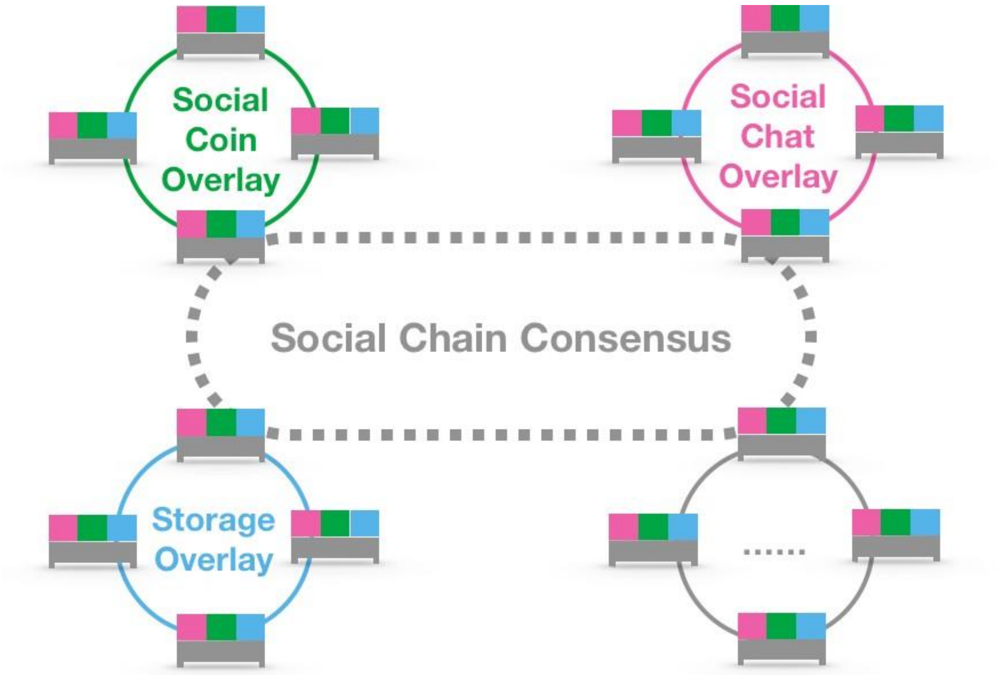
Storage Node is a SNP container spawned by a service node, which supports file storage service, proof of access and proof of store. The storage node should have sufficient storage space. All storage nodes form a **Storage Overlay**.

Social Chat Node is SNP container spawned by a service node, which supports SIP registration, SIP presence, SIP proxy, proof of presence, proof of relay and so on. The social chat node should have sufficient CPU and memory. All social chat nodes form a **Social Chat Overlay**.

As in figure below, containers in each color represents a class of service nodes for a particular SNP service. This class of nodes are grouped into numerous overlays using structured topology control mechanisms, regional strategy, BreakMaxDegree and BreakMinDegree.The social chain is built up collectively from two layers: consensus overlay layer and service overlay layer.

DHT lookup applies to each service overlay. Storage overlay uses IPFS and the lookup key is IPFS key. For Social Chat worker overlay, the lookup key is SIP URI. DHT lookup is deterministic and every time it will locate the same service node.

Each service node performs easy task but should respond real time; this way small computing devices can make little profit in regular basis.



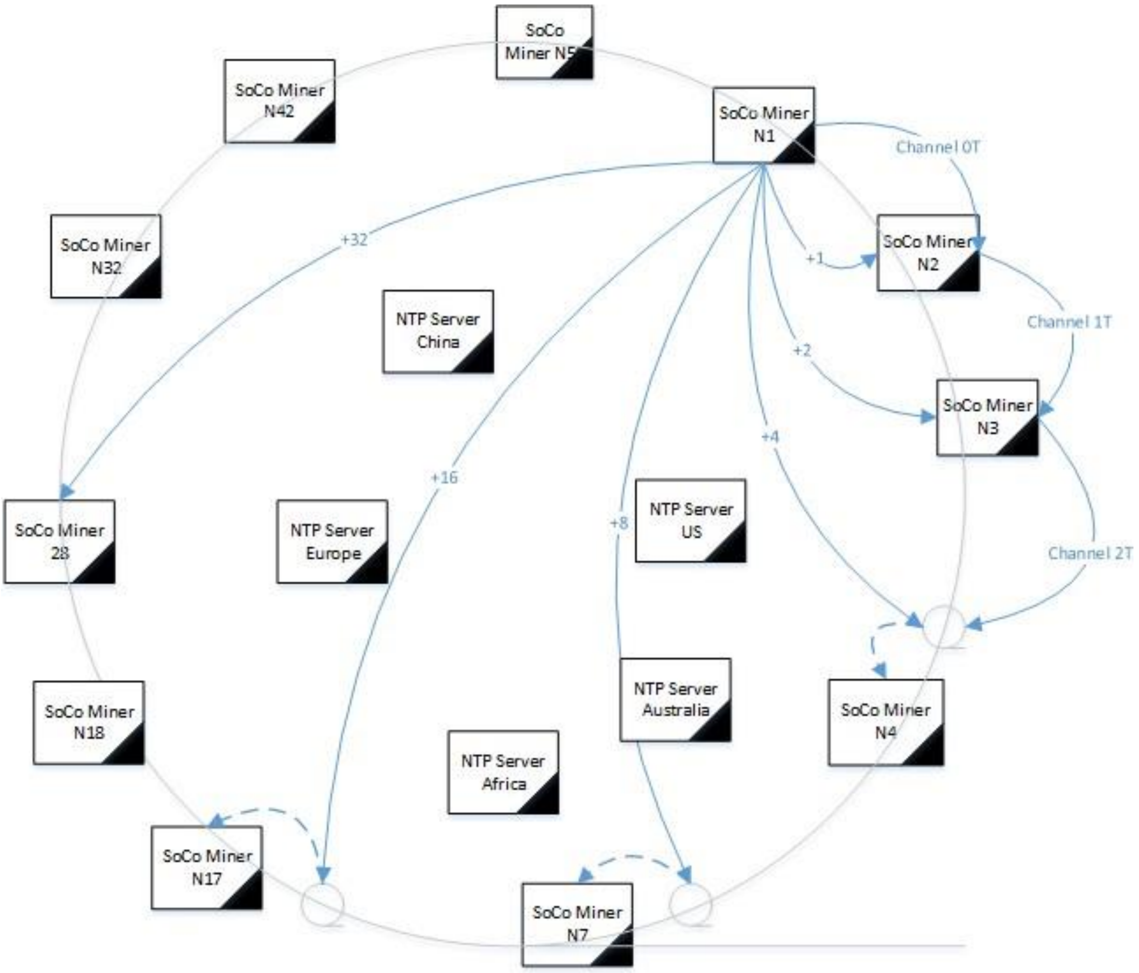
Service nodes will provide the following services:

DHT Lookup	API Execution	OAuth	Storage
Auditing	Block of Transactions	Account Agent	Broadcast
Registration	Notify online/offline	Lookup Firewall Ports	Relay

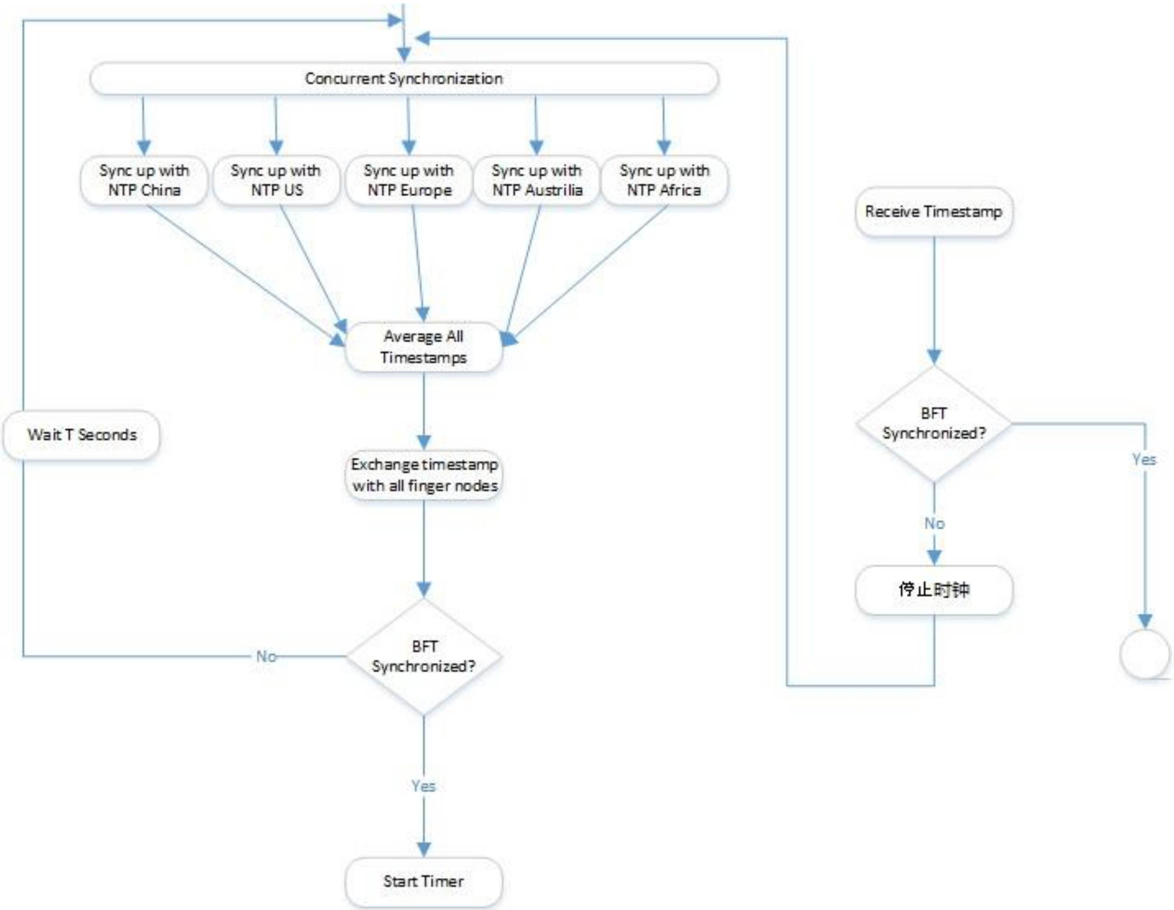
7.3.4 Network Consensus Synchronization

The key for BFT-TDPOS is SNP network clock synchronization, in order to produce unique block hash. As such, all nodes in the consensus overlay periodically asserts they are synchronized with the rest of the overlay.

The diagram below is a structured P2P network based on CHORD algorithm. All consensus nodes use 5 NTP servers each in a distinct network geographic area;



The algorithm of synchronizing system clock over the consensus overlay is illustrated as below:



After synchronizing with all NTP servers from distinct network geographic areas, each consensus node exchanges its synchronized timestamp with its finger peers. If its peer has a similar timestamp (difference less than D milli-seconds), then the consensus node has clock consensus with this peer.

If a consensus node accomplishes P% of clock consensus with all its peers, then this consensus node is considered clock synchronized with the whole overlay.

7.4 SNP Blockchains and Consensus

7.4.1 Acquiring Privileged Token

Each Contract Node or SoCo Miner Node is owned by a wallet. The relevant wallet will have its AccountBot executed in the Contract Node or SoCo Miner Node and delegate the responsibility of signing blocks to the AccountBot.

The Contract Node or SoCo Miner Node will regularly acquire a privileged token thru OAuth 2.0.

When a block is ready, the Contract Node or SoCo Miner Node will produce a hash based on the latest privileged token and the entire block and send the block along with the hash to the AccounBot. The AccountBot receives the block, verifies the hash using last issued privileged token, signs the block, and sends back to the Contract Node or SoCo Miner Node.

7.4.2 Delivering Records to SoCo Channel

When a ContractBot gets executed, it verifies PoSR and then produces transactions. Transactions are subject to auditing in an order. After successful auditing, the PoSR and resulting transactions are sent to SoCo nodes on the relevant channels.

For each PoSR, it will engage a pair of wallet addresses: one is server wallet and the other is client wallet. For each transaction, it will engage a payment from client wallet address to server wallet address.

Client wallet address is hashed into index i, which references SoCo channel i and the i-th SoCo chain in the vector; server wallet address is hashed into index j, which references SoCo channel j and the i-th SoCo chain in the vector. PoSR or transaction with such client wallet and server wallet will be delivered to next SoCo node on channel i with label i on the record, and to next SoCo node on channel j with label j on the record.

When a SoCo node receives such PoSR or transaction, if record comes with label i, the SoCo node places it in **consensus queue** Q[i]; if record comes with label j, the SoCo node places it in consensus queue Q[j];

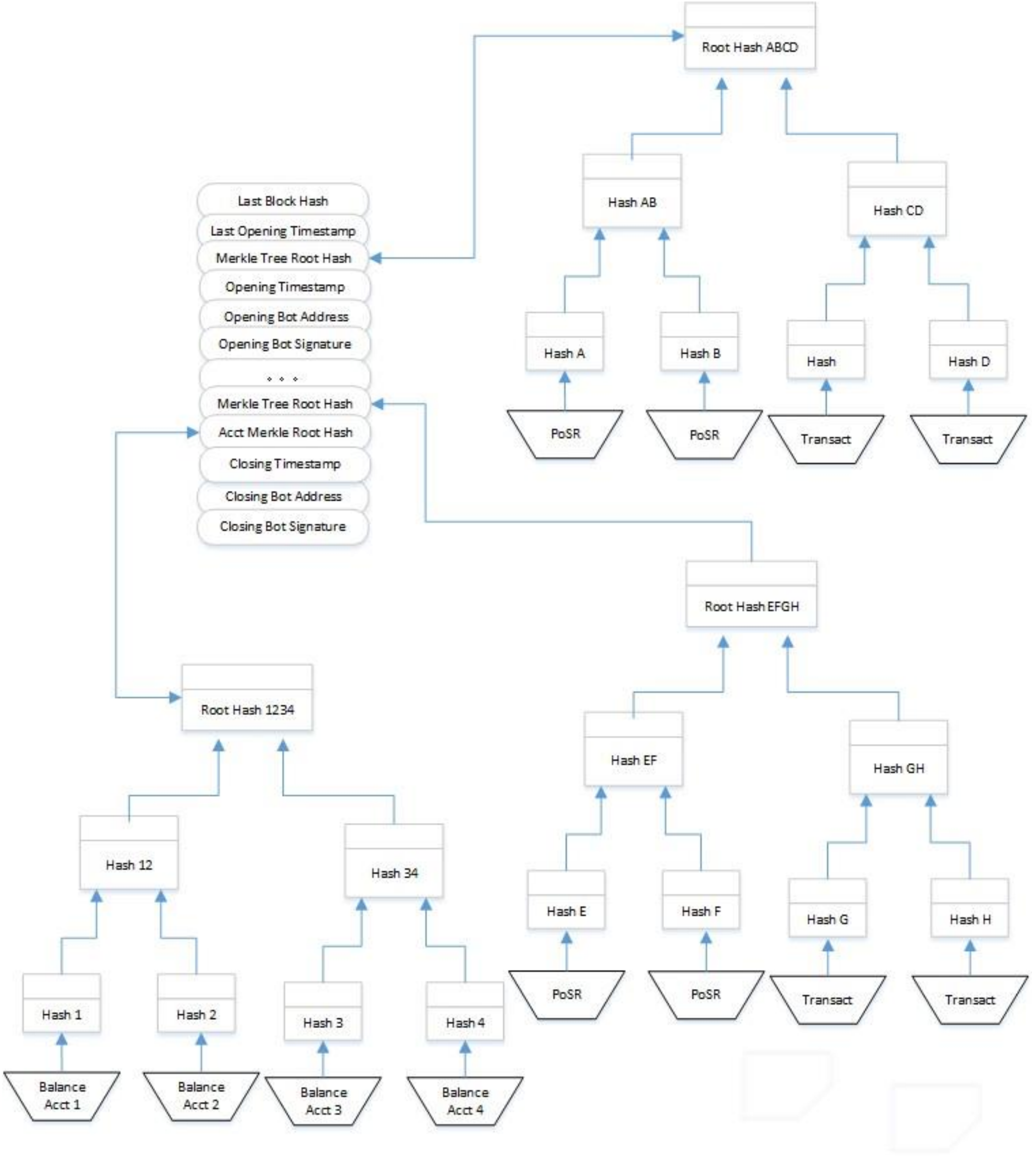
7.4.3 Producing SoCo Blocks

The SoCo node produces new SoCo blocks based on the following algorithm:

7.4.4 SoCo Block, Merkle Tree and Channel Consensus

A SoCo block consists of multiple Merkle trees and is labeled with channel index i.

After a new block is produced and signed, the SoCo node starts populating the new block to the SoCo overlay for consensus. It exchanges the new block with its finger peers. Each receiving SoCo node will perform a list of verifications:



- For each PoSR record, verify signature;

- For each transaction, verify it is consistent with the sourcing PoSR record, the sourcing ContractBot and calculation result ;
- Verify Each signature was done by the SoCo node who owns the channel at the time of signing the block (timestamp), and the signature is valid;
- Verify the payer account balance is sufficient;

If the SoCo node passes all verifications, then the SoCo node has consensus on the block with this peer.

If a SoCo node accomplishes P% of block consensus with all its peers, then this SoCo node is considered to have consensus with the whole overlay.

The SoCo node then saves the block into the relevant SoCo chain for channel i.

7.4.5 Consensus Extension to Entire Overlay

After accomplishing consensus with the SoCo overlay and saving a block into a SoCo chain for channel i, each SoCo node starts populating the block to all other consensus nodes in its candidate ring.

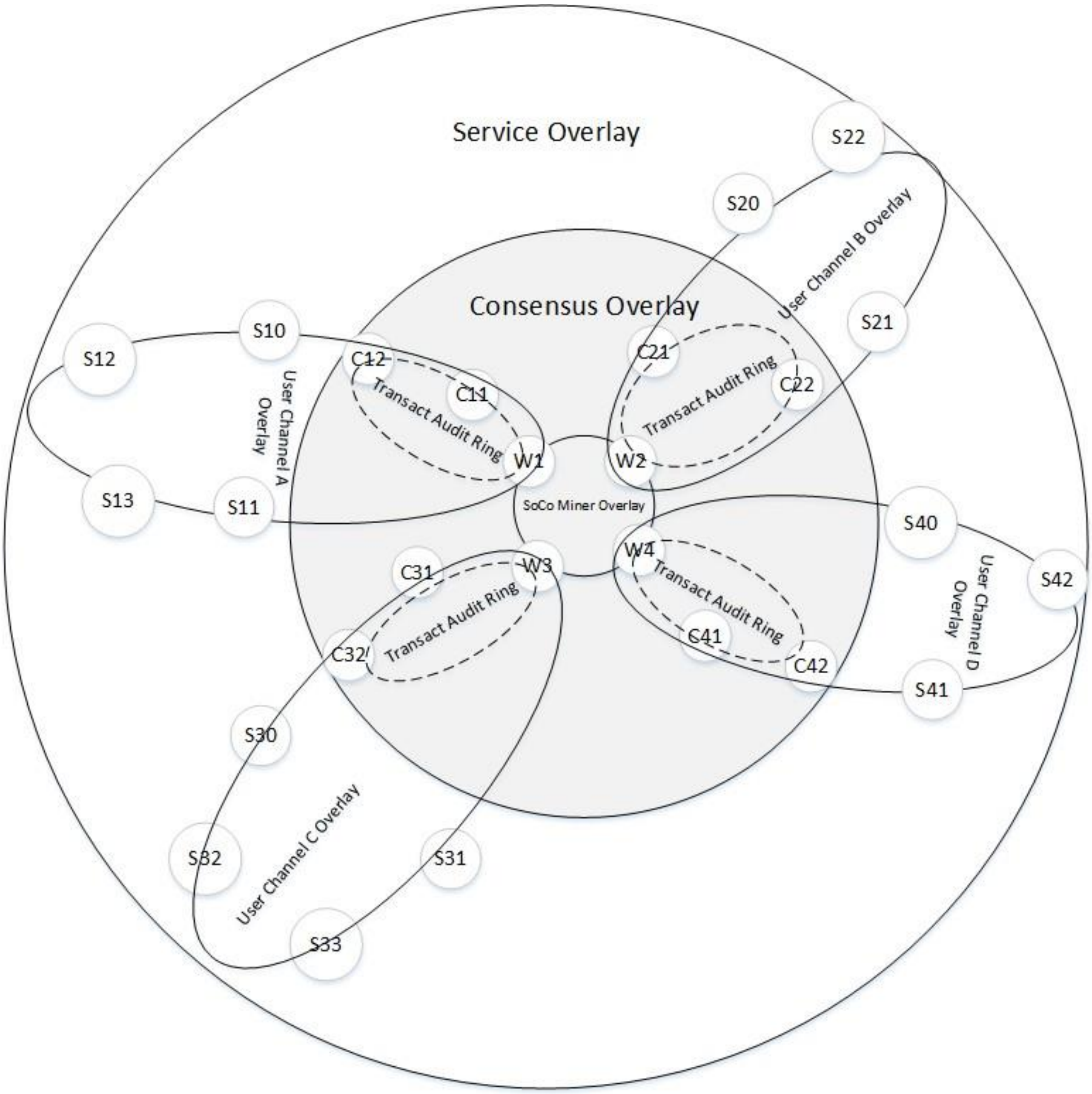
Each receiving candidate SoCo node will perform a list of verifications as in 7.4.5. If the SoCo node passes all verifications, then the candidate SoCo node has consensus on the block with its target SoCo doe. The candidate SoCo node then saves the block into the relevant SoCo chain for channel i.

7.4.6 Consensus to Service Overlay

If a consensus node in a candidate ring determines block label (channel index i) matches its own node ID in the Chord overlay, it will send this block to all its neighbor nodes in the user channel overlay for channel i.

If the receiving node is a candidate SoCo node, it does nothing as it already accomplished consensus on the block.

Otherwise, the receiving node is a service node. It performs a list of verifications :

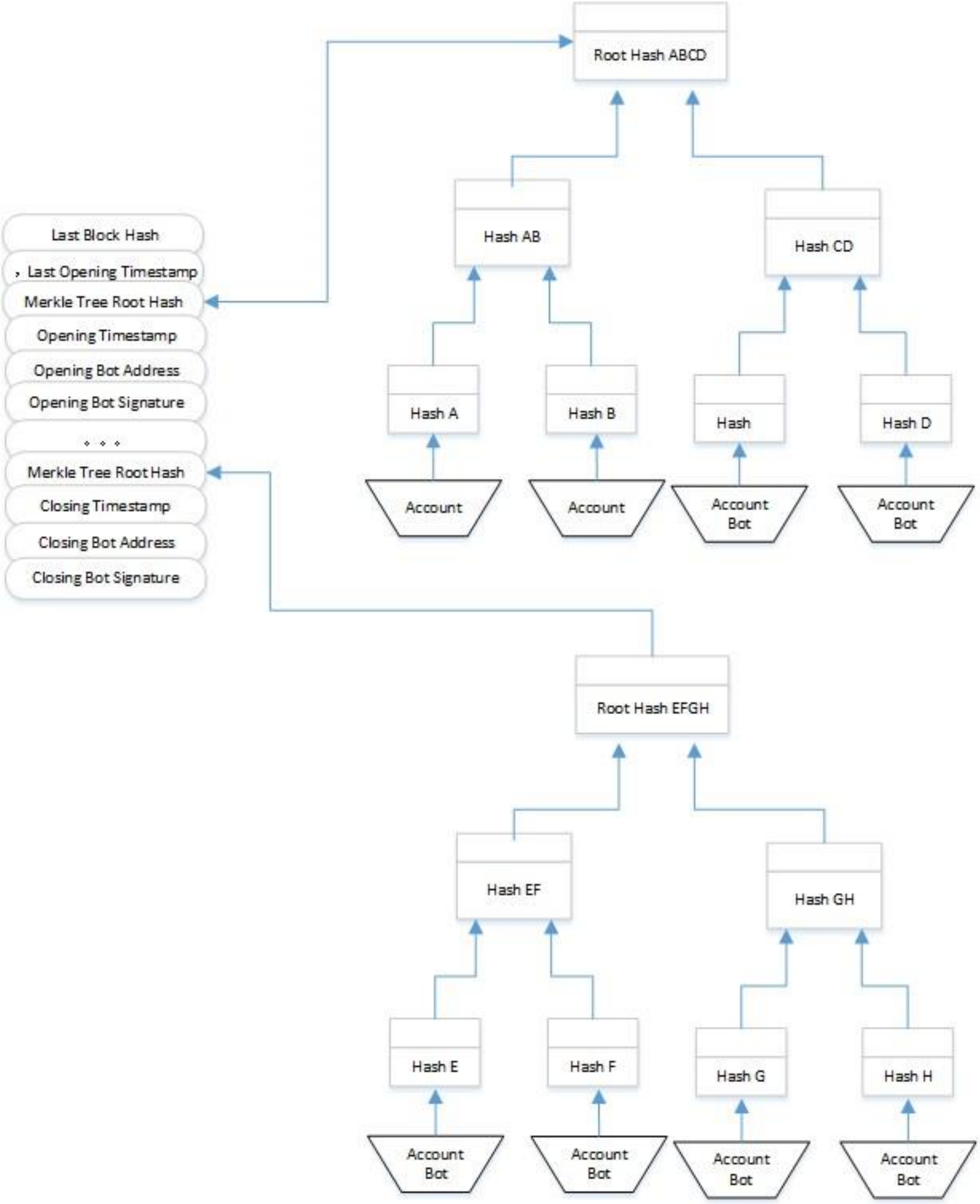


- For each transaction, verify it is consistent with the sourcing PoSR record, the sourcing ContractBot and calculation result ;
- Verify Each signature is valid;
- Verify the payer account balance is sufficient;

If the receiving node passes all verifications, then it has consensus on the block with this user channel overlay. The receiving node then saves the block into the SoCo chain for channel i.

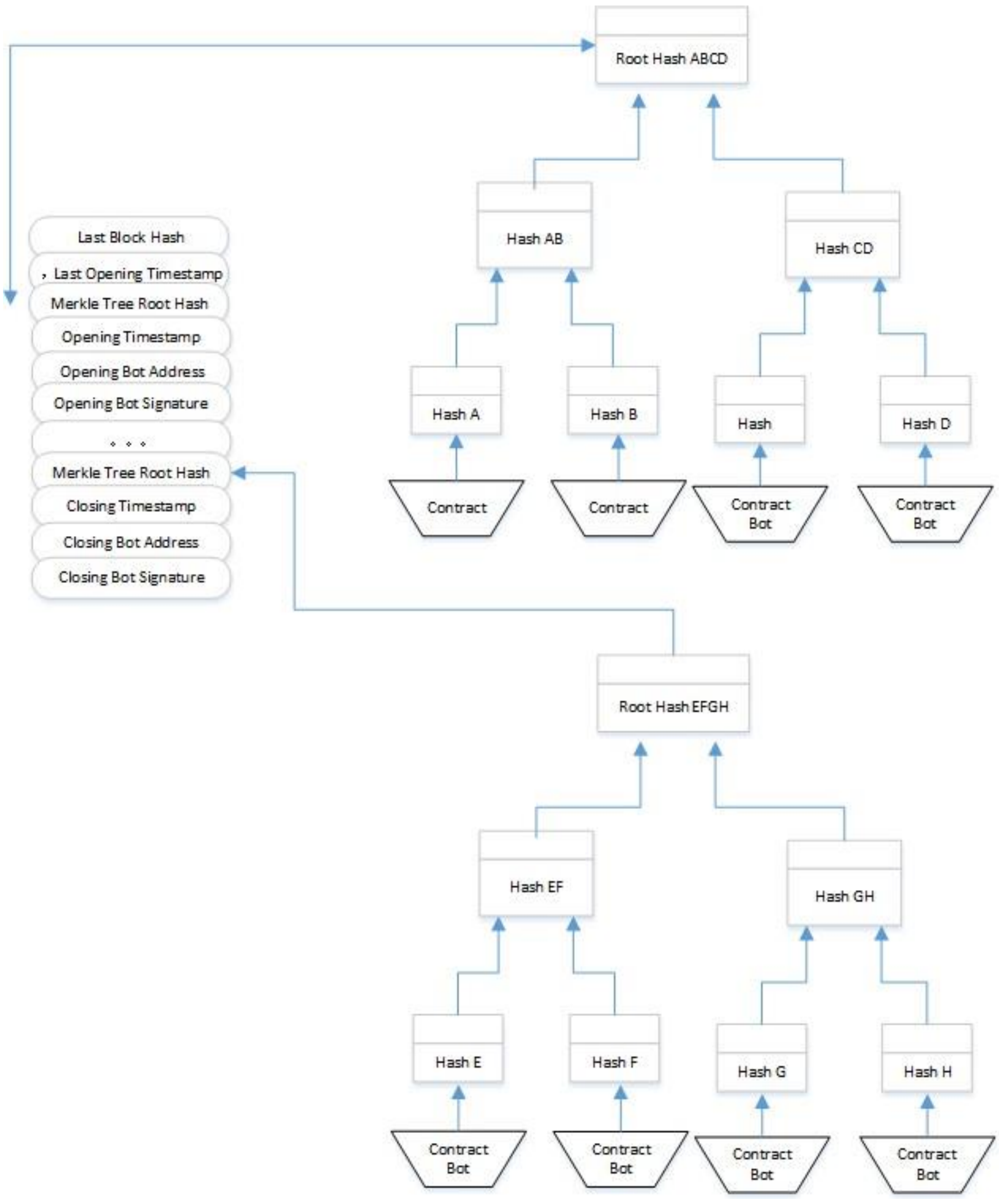
7.4.7 Data Structure of Account Block

The SoCo Account Chain is a blockchain that stores Account classes and AccountBots.



7.4.8 Data Structure of Contract Block

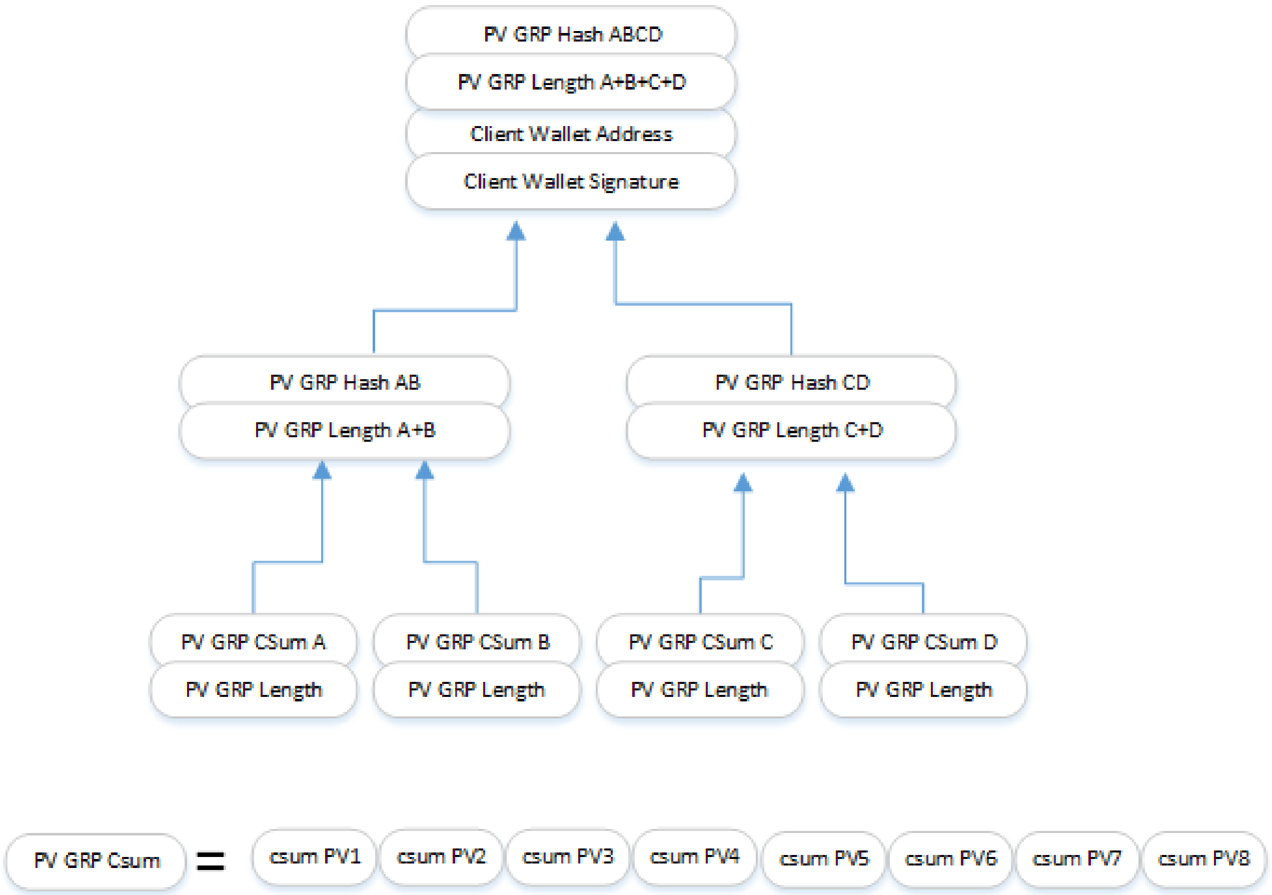
The SNP Contract Chain is a blockchain that stores Contract classes and ContractBots.



7.5 SNP Proof of Services

7.5.1 Proof of Page View

Page view data is the key to measure how good a web page design and companies who lease the page for advertisements can make SoCO payments based on PageView (PV) data. Here is the data structure of Proof of PV Record:



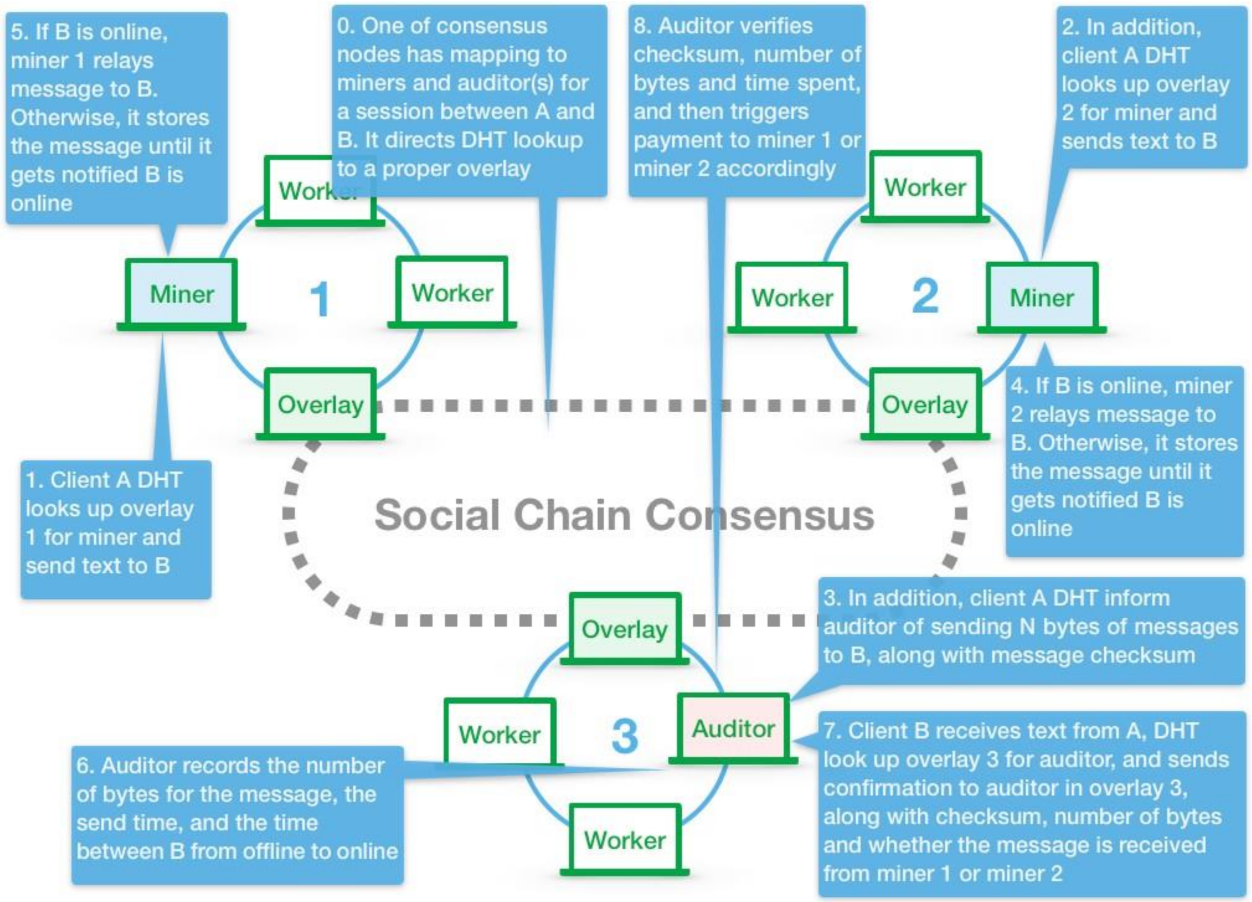
A client wallet should lease a service node to run web server, and the web server should store Proof of PV records. APP who provides web server software should redirect Proof of PV record to his own node.

Upon APP's request, the web server produces a set of Proof of PV records in Merkle tree and sends to APP's service node. APP service node verifies the Proof of PV records and sends to a relevant ContractBot. Thereafter the process follows ContractBot service flow and as a result APP gets paid by the client wallet.

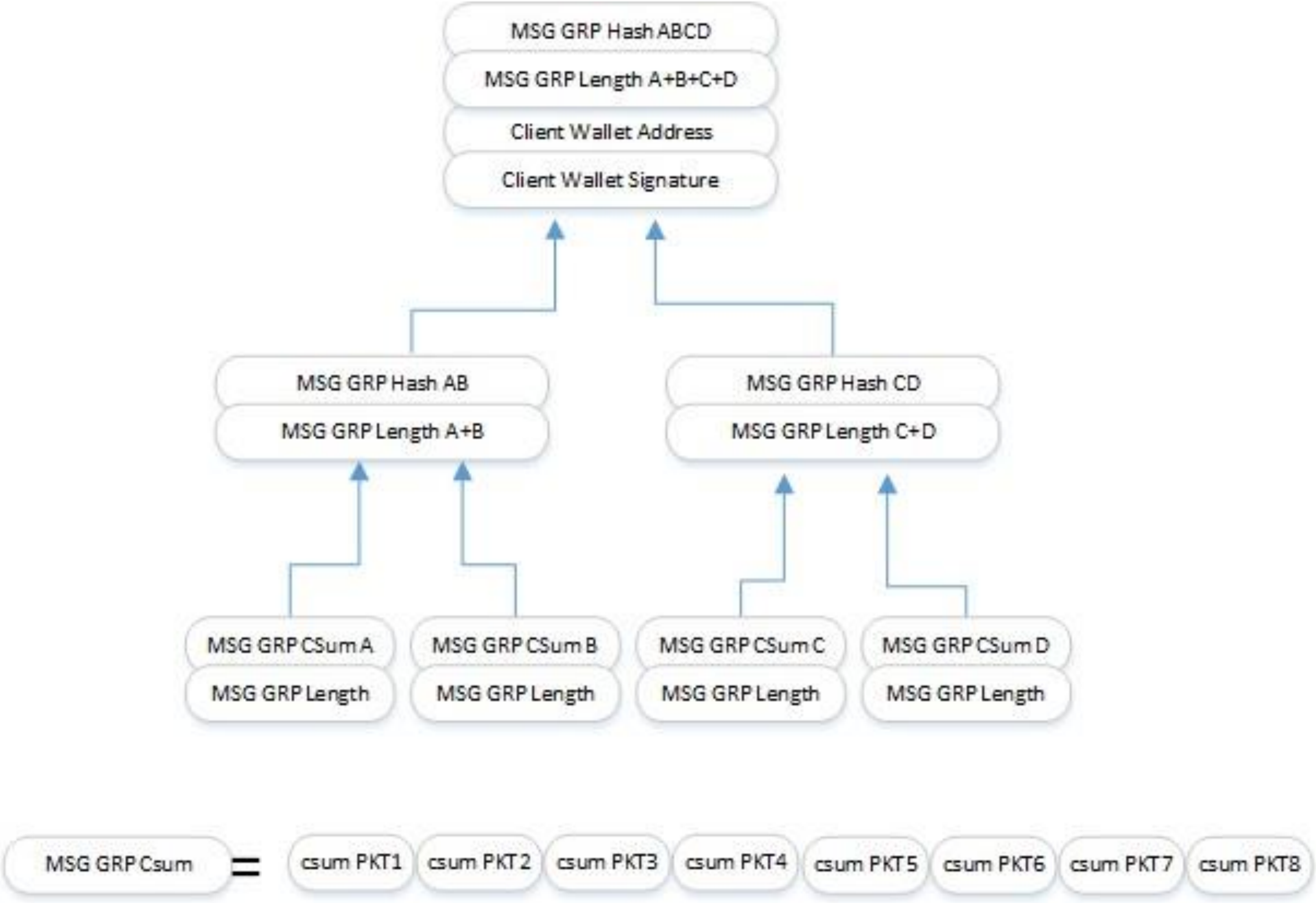
7.5.2 Proof of Relay

The SNP will provide a distributed messaging system that supports instant messaging. For point-to-point chat and message groups, the SNP will differentiate between business scenarios and provide corresponding data interfaces and distributed services. Since message is the most important application scenario in the social ecosystem, SNP will also support the push of many types of messages. With the support of APP, users can complete various activities and data transactions in the chatting scenario. For example, cryptocurrency transfers, payments, red envelopes, APP discoveries, virtual gifts, virtual pet games, and more.

Here is a high level diagram to prove a worker server has provided message relay service in timely fashion:


5. If B is online, miner 1 relays message to B. Otherwise, it stores the message until it gets notified B is online
0. One of consensus nodes has mapping to miners and auditor(s) for a session between A and B. It directs DHT lookup to a proper overlay
8. Auditor verifies checksum, number of bytes and time spent, and then triggers payment to miner 1 or miner 2 accordingly
2. In addition, client A DHT looks up overlay 2 for miner and sends text to B
Worker
Miner
1
Worker
Overlay
Worker
Worker
2
Miner
Overlay
4. If B is online, miner 2 relays message to B. Otherwise, it stores the message until it gets notified B is online
1. Client A DHT looks up overlay 1 for miner and send text to B
Social Chain Consensus
3. In addition, client A DHT inform auditor of sending N bytes of messages to B, along with message checksum
Overlay
Worker
3
Auditor
Worker
6. Auditor records the number of bytes for the message, the send time, and the time between B from offline to online
7. Client B receives text from A, DHT look up overlay 3 for auditor, and sends confirmation to auditor in overlay 3, along with checksum, number of bytes and whether the message is received from miner 1 or miner 2

Message relay data is the key to measure how much service a relay may provide and message receiver can make SoCo payments based on Message relay data. Here is the data structure of Proof of Relay Record:

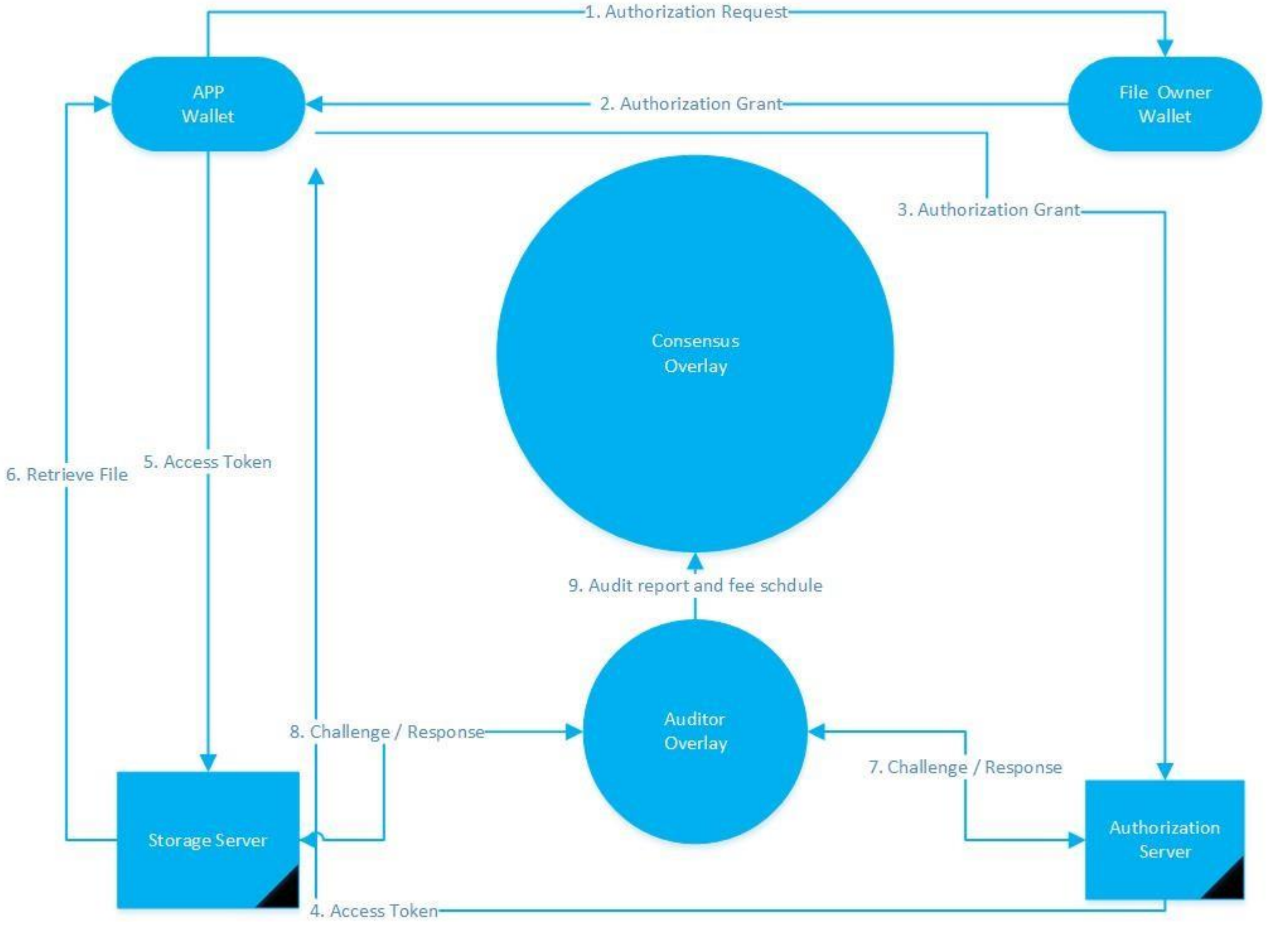


Both the message receiver and the relay node should store Proof of Relay records.

Upon request from the relay's wallet, the receiver produces a set of Proof of Relay records in Merkle tree and sends to the relay node. The relay node verifies the Proof of Relay records and sends to a relevant ContractBot. Thereafter the process follows ContractBot service flow and as a result the relay's wallet gets paid by the receiver.

7.5.3 Proof of Access

Proof of access is for client to retrieve files thru SNP API. This procedure will follow standard Open Auth 2.0 protocol in peer-to-peer fashion.



7.5.4 Proof of Store

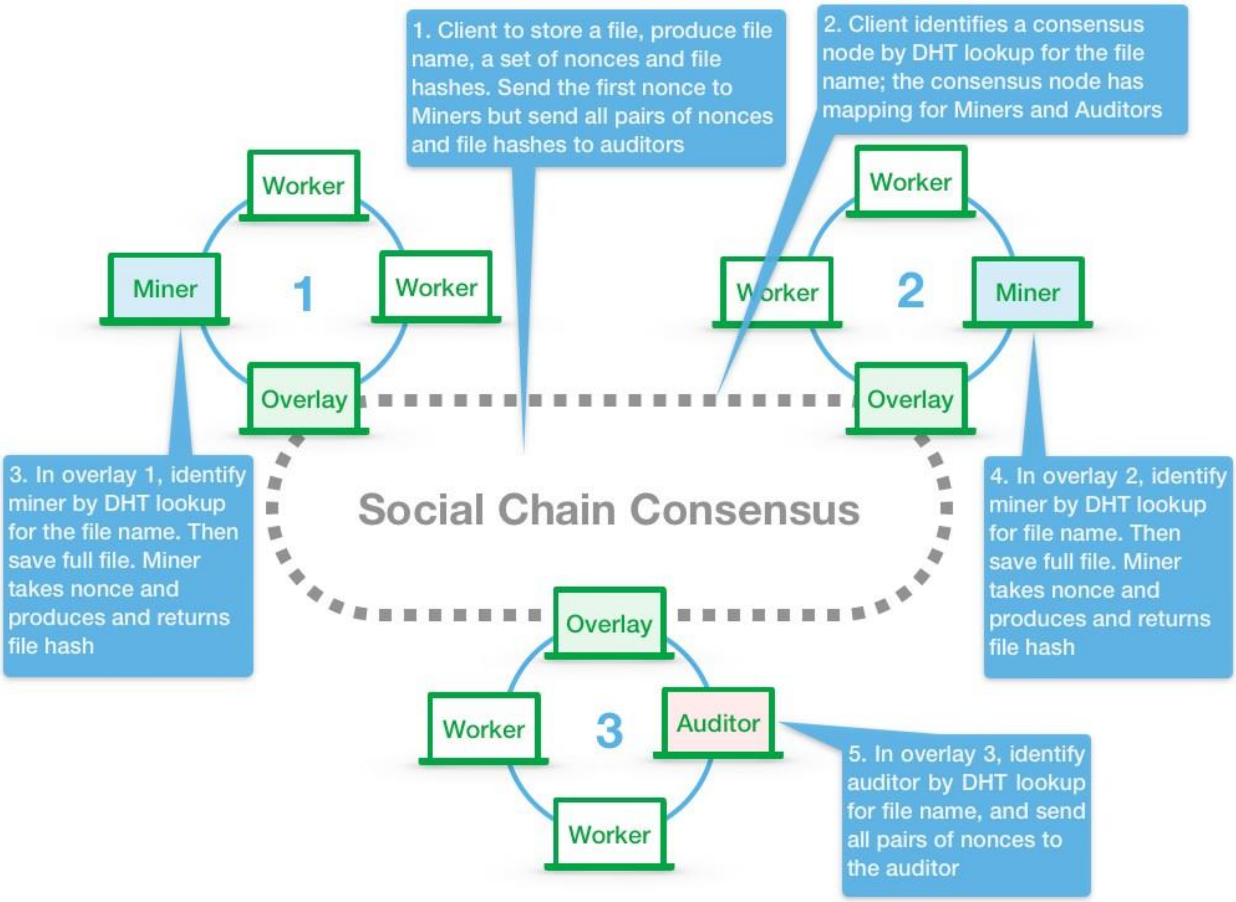
The SNP offers a family of underlying features and protocols for distributed storage and data sharing of consumer digital assets.

Distributed data storage layer: the SNP will be compatible with a variety of distributed data storage protocols, combined with SNP account authentication and authorization mechanisms to support users of social data storage and distributed networks.

Distributed data sharing: the SNP provides the basic data sharing protocol for SNP accounts. Depending on the usage scenario, users can grant access permission of their data contents to all accounts or some accounts.

In data authorization mechanisms, data storage and sharing system, the data privacy and data security are key attributes. Therefore, corresponding authorization mechanism is designed in the SNP, and any transaction involving any data needs to notify the data owner for authorization.

Here is a high level diagram to prove a worker server has provided storage service in reliable fashion:



1. Client to store a file, produce file name, a set of nonces and file hashes. Send the first nonce to Miners but send all pairs of nonces and file hashes to auditors

2. Client identifies a consensus node by DHT lookup for the file name; the consensus node has mapping for Miners and Auditors

3. In overlay 1, identify miner by DHT lookup for the file name. Then save full file. Miner takes nonce and produces and returns file hash

4. In overlay 2, identify miner by DHT lookup for file name. Then save full file. Miner takes nonce and produces and returns file hash

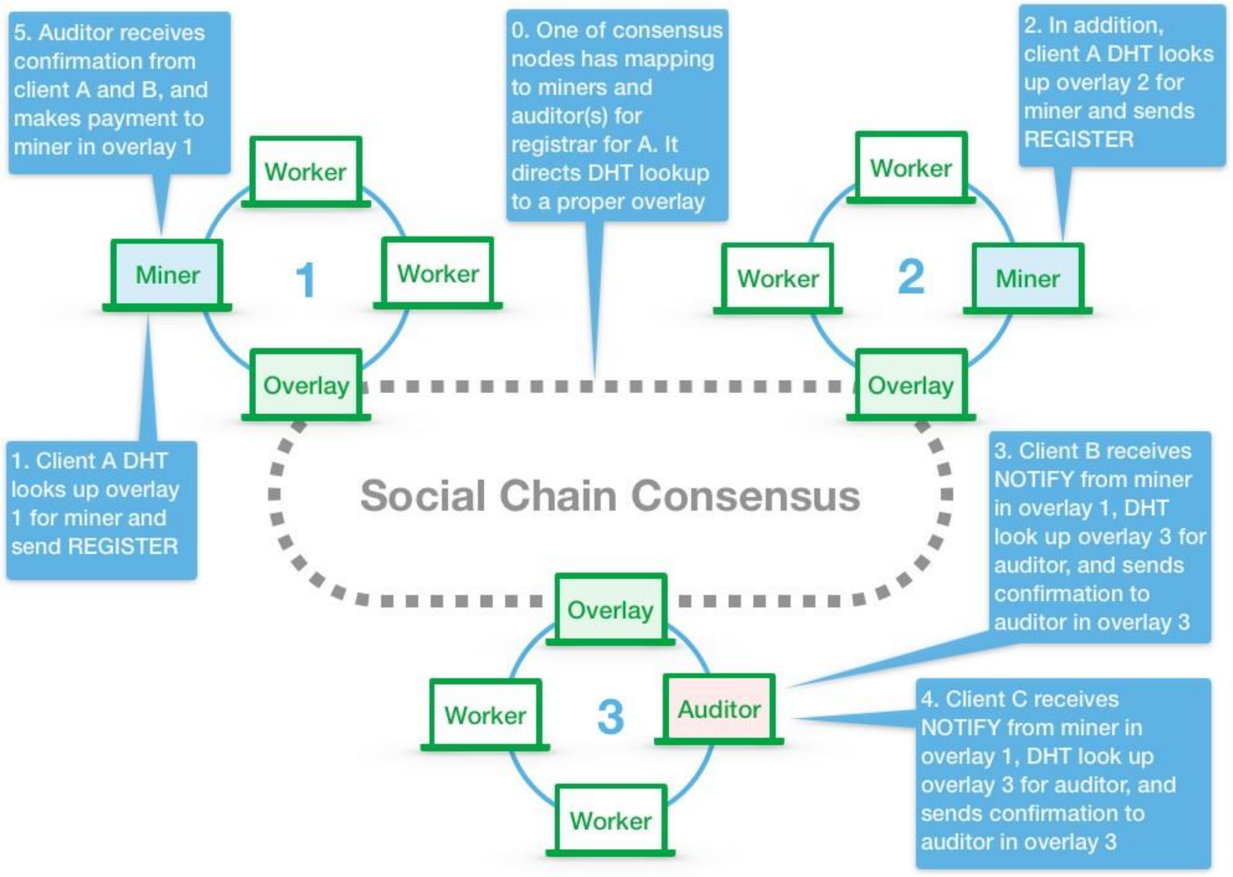
5. In overlay 3, identify auditor by DHT lookup for file name, and send all pairs of nonces to the auditor

7.5.5 Proof of Presence

Proof of Presence is for client to register and get notified online/offline thru SNP API. Client registers with a worker registrar every so often (e.g. every 60 seconds). When the client registers, its state is online. If the client fails to register in certain time frame (e.g. 3x60 =180 seconds), then its state becomes offline.

The user's friends or a relay worker may subscribe for its online status. The worker registrar should inform all subscribers of this client's online status whenever the state change from online to offline, or from offline to online.

Here is a high level diagram to prove a worker registrar has provided presence service in timely fashion:



5. Auditor receives confirmation from client A and B, and makes payment to miner in overlay 1

0. One of consensus nodes has mapping to miners and auditor(s) for registrar for A. It directs DHT lookup to a proper overlay

2. In addition, client A DHT looks up overlay 2 for miner and sends REGISTER

1. Client A DHT looks up overlay 1 for miner and send REGISTER

3. Client B receives NOTIFY from miner in overlay 1, DHT look up overlay 3 for auditor, and sends confirmation to auditor in overlay 3

4. Client C receives NOTIFY from miner in overlay 1, DHT look up overlay 3 for auditor, and sends confirmation to auditor in overlay 3

7.6 Distributed Relation

The formation of social networks is based on the association or relationship between people. The relationship in reality is generated independently in the social scenes and gradually spreads to other social scenes, forming a relationship chain with a single person as a node. The abstraction of the relationship in real life to the ecology of the social chain, is the relationship chain that links user accounts in various APPs.

In the social chain, the user relationship is bound with the social scene provided by the APP, and spread by the user account as a node. Through the social chain trust system, a user will possess ownership of the relationship between his account and other accounts. The user can also authorize the sharing of his relationship chains among multiple APPs. The APP will read the data of the user-shared relationship chains limitedly according to the user's authorization, to support some cross-scene social services and applications. However, the APPs cannot actively change or transfer users' data.

7.7 Distributed Trust

SNP will establish a distributed multi-source trust system. A user account or APP body within the social chain can sign a statement of trust on its own account or on another account. Compared with a single-source trust system, users and specific APPs in a SNP trust system can be the source of other entities' claims. Personal relationships, personal social behaviors and transactions can form a chain of trust from users, producing claims with data records and passing them on in the trust system. These claims will be widely recognized in the repeated use of the declaration to further expand the chain of trust. APPs and individual users can identify users based on the trust system of SNP and complete the interactive activities such as data assets transaction and payment.

SNP will leverage Open Auth 2.0 and RFC 7070 “Architecture for Reputation Reporting” in the distributed trust social chain.

The social chain will be self-policing. A set of shared ethics of the user population will be defined and enforced by the consensus overlay.

The social chain will maintain anonymity. Any social chain wallet’s reputation will be associated with SOCOID token, an opaque identifier.

The social chain will not assign any profit to newcomers. Any social chain wallet’s reputation will be obtained by consistent good behavior through several transactions, and it will not be advantageous for malevolent peers with poor reputations to continuously change their identifiers to obtain newcomers status.

The social chain will have minimal overhead in terms of computation, infrastructure, storage, and message complexity.

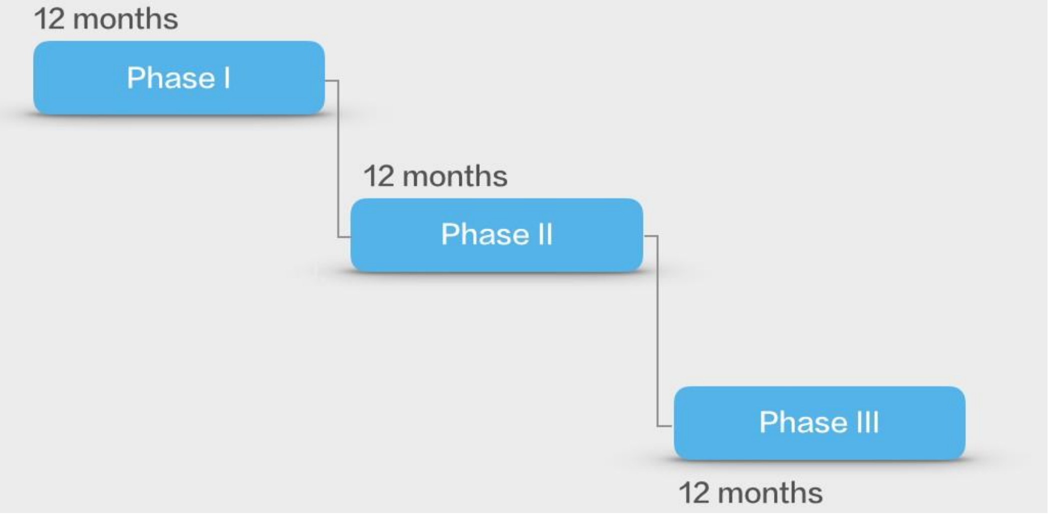
The social chain will be robust to malicious collectives of wallets who know one another and attempt to collectively subvert the social chain.

8. Road Map

Tasks Accomplished:

- 2017.12 Invention of SNP, SoCo, and Social Chain
- 2018.01 Market feasibility study
- 2018.02 Technical feasibility study & prototype architecture design
- 2018.03 White paper version 0.1
- 2018.04 White paper with technology details version 0.2

Development phase schedule:



Phase I (To be completed in 12 months)

- To develop core protocol layer implementation based on existing public chains, such as Ethereum, Quantum Chain, and NEO.
- To implement a social wallet APP
- To create sample APP implementations to verify SNP capabilities

Phase II (To be started within 12 months from ending of Phase I)

- To build the SNP Ecology APP Developer Community step-by-step
- To provide strong support to SNP-based APP development

Phase III (To be started within 12 months from beginning of Phase II)

- To gradually develop and improve support for various public chains, and develop SNP's own public chain as the infrastructure of the next generation's social networking ecosystem
- To migrate existing Social Coin and APPs to the new underlying public chain

9. Team introduction

9.1 Team Profile

The founding team of Social Coin and SNP is composed of current entrepreneurs, venture capitalists, and senior experts in distributed technology from Singapore and the United States. As entrepreneurs they have raised over 100 million U.S. dollars and successfully founded several publicly-listed companies. As venture capitalists they manage a number of funds that total more than 1 billion U.S. dollars. As technical experts they have more than 20 years of research and development experience in distributed technologies.

9.2 Core Founding Team

Mingliang Jiang (Strategy & Operation)

https://www.linkedin.com/in/jiangmingliang/

Mr. Mingliang Jiang is responsible for guiding SNP and SoCo's strategy, leading the overall development of the project. He is a current entrepreneur and a member of the Management Committee of Suzhou Research Institute at the National University of Singapore. He is the inventor of the CBRP Distributed Network Protocol(see https://scholar.google.com.sg/citations?user=wg-8VnAAAAAJ&hl=en). He has participated in several publicly-listed companies, including Men Dui Men, Jiangxi FengShang shopping, and others. He studied at the China Europe International Business School, National University of Singapore, and Shanghai Jiao Tong University. In 1994, he received the Singapore government full scholarship.

- In 2006, he founded Jiyi Media, whose major business is television shopping. In 2010, Jiyi sold its shopping business to the ChangBai Group, which is listed in Chinese A-Shares (600856).
- From 2005 to 2006, he worked in Livedoor Japan as COO of DuoLaiMi Chinese Website, in charge of its two core sectors: company portal and wireless value-add services
- He joined TeckWah Group (SESDAQ: TECKWAH) Singapore in 2003 as the executive vice president of its Internet business subsidiary, TeckWah Online.
- He was involved in founding of a number of Internet companies with a wide range of network connections. In 2008, he was awarded the title of "Shanghai Top 10 IT" by the Shanghai Government.
- He graduated from the Department of Computer Science at the National University of Singapore in 1999 and received an EMBA degree from the China Europe International Business School (CEIBS).

Roderick Chia (Marketing & Finance)

https://www.linkedin.com/in/roderickchia/

Mr. Chia Yeow Kheng (Roderick Chia) is responsible for the promotion of SNP and SoCo's eco-operations. He is the founder of rodVENTURE and the managing director of IDM Venture Capital Ventures. He has also been a successful inventor, entrepreneur, educator and investor for his longstanding achievements and affirmation in many fields.

- As an entrepreneur, he co-founded Postkid in 1999, allowing the company to grow quickly and efficiently within three years, and successfully back behind the scenes with extraordinary results. The Singapore government awarded him the Phoenix Finalist Award in recognition of his entrepreneurial determination and outstanding leadership.
- As a successful continuous entrepreneur and inventor, he has won the Tan Kah Kee Young Inventors' Awards (Tan Kah Kee Foundation) five times and has served as Chief Knowledge Officer, Chief Technology Officer, Chief Strategy Officer and Chief Executive Officer at various companies.
- As an educator, he graduated from Nanyang Technological University (NTU, Singapore) - Department of Electrical and Electronic Engineering, and he obtained his postgraduate diploma and master's degree in science, technology and entrepreneurship from NTU and Washington University through continuous education. Additionally, he also received the Trainer's Diploma from Kauffman Foundation and a Diploma in Executive Development Program co-organized by NTU and Tsinghua SEM. Currently, he is employed by NTU as a visiting lecturer and mentor, but he is also working as a permanent resident of science and technology entrepreneurs in the Department of Electrical and Electronic Engineering.

Mr. Chia Yeow Kheng has not only achieved outstanding results in many fields, but also devoted himself to social services. He has since long been the judge of Tan Kah Kee Young Inventors' Award, the judge of design and technology awards, the steering committee of the Ministry of Education Innovation Course and the vice chairman of South View (Nanshan) Primary School Advisory Committee. In 2005 and 2009, he was awarded with NTU Alumni Service Award and NTU Outstanding Young Alumni Award, respectively.

John Li (Technology)

https://www.linkedin.com/in/yunzhouli/

Mr. Li is responsible for leading the social chain technology development. He has over 20 years of experience in the networking industry as a proven technology leader focused on technologies such as distributed systems, IP multicast routing and security, wireless and mobile, P2P video casting, cloud technology, VoIP, network routing, multicast security, VPN, etc., and he obtained a total of 21 patents in related fields.

Mr Li is currently a network architect at ViaSat, a US satellite communications company. He is involved in the research and development of a new generation of satellite Internet communications.

- 1988 Master of Science in Mathematics, Tsinghua University
- 1996 Master's degree in Computer Science, National University of Singapore
- He Worked at Software Institute, Chinese Academy of Sciences, and successfully completed China's first National level 7th- and 8th-FIVE ISDN project.
- In 1995, he led a team at the National University of Singapore to successfully develop a Linux Kernel -based mobile IP system. IT Asia praised this achievement as cutting-edge technology innovation in Asia.
- He Founded Lorent Networks in 2009 and developed P2P video premium-casting technology.
- He was awarded with the title of Top 10 Inventors of the company during the Nortel service.

9.3 Advisory Team

Professor Ooi Beng Chin（OOI Beng Chin）
https://www.linkedin.com/in/beng-chin-ooi-34b0634/

Mr. Ooi Beng Chin is an Outstanding Professor at the National University of Singapore (NUS) and a Yangtze River Scholars Distinguished Professor in Zhejiang University, a member of the Singapore Academy of Sciences, ACM Fellow and IEEE Fellow. His main research interests include database performance issues, indexing techniques, memory data management, cloud computing and parallel system research and advanced applications, and he is committed to the innovation and industrialization of the ABC field (AI, Big Data, Cloud Computing).

Professor Ooi has published more than 200 papers in top international conferences such as ACM SIGMOD, VLDB, IEEE ICDE and top academic journals such as ACM TODS, IEEE TKDE, VLDB, etc., cited more than 17,000 times and H-index reached 71. He has chaired several program committees for a number of CCF Class A international conferences, including the Vice Chair of the Program Committee at IEEE ICDE'00, 04, 06, and the Chair of ACM SIGMOD'07, VLDB'08 (core DB track), and IEEE ICDE'12.

He was also the Editor-In-Chief of the Journal of Big Data Research (JDBR), Chairman of the VLDB Endowment Committee, and board member of the VLDB Endowment Council (2014-2017). He is currently chief editor of IEEE Transactions on Knowledge and Data Engineering (TKDE), and a member of the ACM SIGMOD Advisory Committee (2015-present).

Professor Ooi will provide blockchain technical support for the development of SNP. His strong university laboratory will also provide various technical services and incubation for SNP.

Professor Zhi Da

https://www.linkedin.com/in/zhi-da-8b11673

Dr. Da Zhi holds a position as professor of finance at the prestigious University of Notre Dame, which was named the top business school by BusinessWeek. He also published many financial-related articles and essays, and has a remarkable academic contribution to the field. He was invited to give lectures by famous institutions, including Lehman Brothers, Barclays Bank, New York Stock Exchange, Shanghai Institute of Advanced Finance, etc., and well-known business schools like Indiana University, Arizona State University, University of Edinburgh, Chinese University of Hong Kong, University of Michigan, University of Southern California, University of Houston, Peking University, London Metropolitan University, and more.

He has an important academic status in finance and economics. Dr. Da Zhi will provide consulting and support for SNP and ecology in financial compliance, financial investment and financing operations, etc.

- 2006 PhD in Finance, Northwestern University
- 2001 Master's Degree in Financial Engineering, National University of Singapore
- Awarded the DBS Gold Award and the ABN AMRO Scholarship in 2001 with the best performance
- Finalist 2005 Lehman Brothers Financial Excellence Research Fellowship
- 2006 WFA New York Stock Exchange Grant
- 1st Prize winner of the Welch Memorial Award essay competition (2010)
- 2016 China Financial Research Conference Outstanding Paper Award
- Best Paper of 2016 China XY Investment Award
- Research Analyst for Financial Derivatives at DBS Singapore
- Currently serves as Visiting Professor at the University of Notre Dame in Finance and Education and Peking University
- Chartered Financial Analyst (CFA) since 2001
- Global Risk Professionals Association (GARP) Certified Financial Risk Manager
- American Financial Association and Western Financial Association

David Pan

https://www.linkedin.com/in/davidspan/

David Pan is CEO of SDChain and a General Partner of AY9 Capital. Pan joined SDChain with 20 years of industry experience serving North American and Asian technology and investment companies in IoT, semiconductor, software, electronics manufacturing and telecom sectors. Prior to SDChain, David served as the Director of IoT Asia Marketing at Arm, growing and supporting Arm's IoT developer ecosystem and partnerships in Asia. Prior to Arm, David served as the President of Ayaris 9 with sell-side mandates in IoT, clean tech and e-commerce sectors, as well as managing buy-side venture capital specialized in pre-IPO projects. Pan graduated from Harvard with a Masters in Finance, Golden Gate with a Masters in Enterprise Systems, and UC Berkeley with a BA in Architecture.

Adam Cheng

https://www.linkedin.com/in/adam-cheng-49018415b/

Adam Cheng is a partner at Hauzen LLP in Hong Kong (in association with Anjie Law Firm in mainland China). His practice areas cover cross border M&A, PE investment, corporate financing and IPO for both Chinese and international companies. Recently, he has been actively involved in the cryptocurrency sector including ICO and cryptocurrency fund formation. Prior to joining Hauzen in 2018, Cheng had 20 years of practice experience in leading UK and US firms including Sidley Austin, Herbert Smith and Skadden Arps. He is qualified in New York State, Hong Kong and mainland China.

9.4 Partners

AY9 Capital

AY9 Capital

AY9 Capital invests in US pre-IPO unicorn companies and promising ICO projects with rigorous due diligence on blockchain applications and team experience.

TRENDBLOCKS

Trendblocks

Trendblocks is a cryptocurrency accelerator focused on providing pre-ICO due diligence and post-ICO community operation services for invested projects.

10. Disclaimer

This document is only for conveying information and does not constitute an opinion on the transaction of project shares or securities. Any proposal or request for offer to such effect will be made under credible terms in accordance with the permission of applicable security laws and other related laws. The users are strongly urged to review the Notice to Residents ([link]) and, if deemed necessary, seek independent legal advice. The information and analysis above does not constitute any investment decision or concrete advice.

This document does not constitute any investment proposal, investment intent or investment solicitation on securities. This document does not constitute and shall not be construed as a transaction offer or an invitation to transact any form of securities, neither is it a contract or promise in any form.

SoCo clearly states that users with relevant intent shall have clear knowledge of risks on the SoCo platform. The users are strongly urged to review the Risk Disclaimers ([link]). By participating in this platform, each contributor confirms its knowledge and acceptance of all the project risks, and is willing to personally take responsibility for all corresponding results or consequences. SoCo clearly states that it will not take responsibility for any direct or indirect losses arising from the participation in the Soco project, including: (i) reliability of all information provided in this document; (ii) any resulting mistake, negligence, or information inaccuracy; (iii) or any subsequent behavior.

SoCo is a digital token used, besides other scenarios, in the SoCo ecosystem. SoCo is not an investment target and we cannot guarantee its value which value may decrease under certain circumstances. Due to unpredictable factors, milestones listed in this white paper may be subject to change. While the SoCo team will make the best efforts to implement all milestones stated in this white paper, all individuals and groups purchasing SoCo shall bear the risks on their own. SoCo does not represent the rights of ownership or control. Owning SoCo does not provide ownership of SoCo or other associated applications. SoCo does not confer any rights on any individual to exercise participation in, control over or decision making of SoCo or other SoCo based applications.

SoCo hereby reiterates that it adopts the highest standard of regulatory compliance and business ethics, including without limitation to:

- Our Know Your Client and Anti-money Laundering Policy ([link]);
- Our Privacy Policy (as posted on the Foundation's website ([link]);
- Terms of Use of the Foundation ([link]);
- Our Risk Disclaimers ([link]); and
- Our Notice to Residents ([link]).

These policies and disclaimers are updated by the Foundation from time to time. We strongly urge you to read carefully and understand fully these policies and disclaimers as they form as an integrated part of this White Paper. Should you have any questions, please feel free to contact us at soco@soco.social.