Let Every Contribution Be Seen: Rural Co-construction Digital Identity (Semi)

Nantang DAO · Technical Detailed Version

Website Summary

Rural Co-construction Digital Identity (Semi) is an open-source tool built together by the partners of Nantang DAO. It gives every rural construction participant — whether a returning youth, an intern or a digital nomad — a digital identity account of their own, accessible with a phone number or email, no technical knowledge required. Cooking a meal, pulling weeds, painting a wall, writing code… every effort made for the community can be recorded in the form of "community points", stating what was done, openly and verifiably, and never quietly altered. It grew out of Nantang, and we welcome more villages and communities to use it too.

Technically, Semi consists of a Nuxt/Vue front end and a Rails/PostgreSQL back end. Behind each account is a smart account based on the Safe and ERC-4337 standards, with on-chain gas fees sponsored by the platform; contribution remarks are written to the blockchain through a smart contract; and it also provides OAuth 2.0 single sign-on, so that community applications such as the task system "Bai" share the same identity.

I. Starting from a Village

Nantang Village in Fuyang, Anhui, is a village with a tradition of "working things out together". Around 1998, villagers organized themselves spontaneously to tackle problems they all faced; in 2003, an elders' association and a women's arts troupe were founded one after another; and in 2004, on that foundation, they established the Nantang Xingnong Cooperative. From then until now, the people here have been practicing rural collaboration for more than twenty years.

Nantang also has a well-known set of "Radish & Cabbage Rules". They come from Robert's Rules of Order, a thick American book devoted to "how to hold a meeting properly". In 2008, Yuan Tianpeng, a promoter of rules of order, came to Nantang and, together with the cooperative, distilled it into the "Nantang Thirteen Rules" that villagers could remember. The village elders could not pronounce "Robert" properly, so they called it "Radish & Cabbage" with a laugh, and the name stuck.

These thirteen rules are all in plain language: the chair only manages the process, offers no opinions and does not "sum things up" on everyone's behalf; what is discussed must be a specific, actionable proposal — "a motion is a motion, that is, a proposal for action"; you raise your hand before speaking, each person speaks for no more than two minutes at a time and no more than twice on the same matter; no going off topic, no personal attacks, only the matter at hand; a proposal passes only when more people support it than oppose it, and a tie means it fails. With these rules, the villagers learned something that looks simple but is actually hard: how to hold a good meeting. The big and small affairs of the cooperative have been

deliberated one by one in this way. Long before digital technology arrived here, the seeds of democracy and collaboration were already buried in this soil.

In 2011, Liu Bing, a long-time volunteer, was drawn to Nantang's democratic practice and came here to volunteer, beginning a lasting connection with the village. Later, through his introductions, Nantang came to know friends from online communities such as SeeDAO and LXDAO. In 2024, a group of young people who cared about public affairs and were familiar with blockchain came to the village to live and work alongside the villagers. In August of that year, Nantang DAO was formally established. "DAO" can be simply understood as: an organization in which a group of people, rather than taking orders from a boss, discuss and act together according to open rules. Since then, returning youth, interns, digital nomads and remote participants have joined one after another, becoming the main force doing the work in Nantang today.

These two threads could come together because they are very much alike: on one side, a cooperative tradition of collaboration rooted in the countryside for more than twenty years; on the other, a new way of open collaboration and co-building in online communities. Neither believes that a few people should decide everything from the top down; both want more people to be able to take part. Nantang DAO also keeps reminding itself: the village first, the DAO second.

II. The Problems We Ran Into

After working together for a long time, we found several problems we could not get around.

First, much of the work goes unremembered. A partner spent a whole morning cooking a big communal meal; someone weeded the fields under the sun all afternoon; an intern organized a village event; a distant partner stayed up late organizing documents and led a session of online co-learning. These things are warm and important, yet they have no place in a traditional ledger and can only be remembered through personal favors. Favors wear thin over time, and those willing to give gradually grow tired.

Second, records are easily lost, and they are hard to trust. Records written in a notebook or stored on someone's computer may be gone when the person in charge changes or the system changes; and when one person keeps the books, others inevitably have doubts.

Third, people come and go, and there is no shared "ruler". In the past, in a society of acquaintances, everyone knew who had helped and who had done the work. But the people working in Nantang today include some who stay for a year and some who come for only two weeks; some live in the village, some are a thousand miles away. With different backgrounds, constant comings and goings, and unfamiliarity with one another, "everyone knows in their heart" is no longer enough. In the past it was "I trust you because I know you"; now what we need is "we can all feel at ease because the rules are open and the records can be checked".

Blockchain happens to help here. Put simply, it is a public ledger that everyone keeps together: once a record is written, everyone can see it, and no one can secretly change or delete it. Its defining features — openness, transparency and immutability — make it a trustworthy way of recording.

But then the problem: blockchain is too hard to use. Among the young people who come to Nantang, many are not familiar with blockchain; they would have to memorize a long string of English words (a recovery phrase), understand professional concepts such as "private key" and "gas fee", and might lose their account with one wrong move. At first we could only have an administrator operate on everyone's behalf in the back end, which brought us back to "one person decides".

Rural Co-construction Digital Identity (Semi) was made precisely to solve this problem of "a good thing that people cannot use".

III. What It Is

Rural Co-construction Digital Identity (Semi) is a personal digital identity account for rural construction participants.

You can think of it as a "community passport" plus a "contribution record book":

- A passport, proving "you are a member of this community";
- A record book, noting "what you have done for the community".

Behind it is the blockchain, but the people using it need know nothing about blockchain — just as we use electricity without understanding how it is generated. For young people living in the village, it is a little notebook for jotting down their own contributions; for distant partners, it is a door that keeps them connected to this land.

Although it grew out of Nantang, it does not belong only to Nantang. Any village, cooperative or community can use it to record and recognize the contributions of its members.

IV. What It Can Do Today

1. Your own identity with just a phone number or email. Enter a phone number or email and log in with the verification code you receive. On first use, set a 6-digit "payment code", and the system will automatically create an account of your own. No complicated password to remember, no long string of English recovery words to copy down.
2. Community points recorded, with every entry "having something to say". This is what Semi most wants people to see. When community points are issued and passed on between members, a remark can be attached, for example "February 13, finished painting the old wall of the cooperative" or "Helped in the kitchen on Saturday, cooked lunch for twenty people". This remark is written into the blockchain together with the record, becoming a permanent, public proof of contribution.

So a record is not just a number; it tells a specific story: who, for the community, did what. Looking back many years later, these records are still there, and they will not disappear because the person in charge changed or the system changed.

3. Community badges. For taking part in a co-creation camp or completing a long-term task, the community can issue you a digital badge as a memento and a mark of recognition. Once you confirm that you accept it, it stays in your personal account.

4. Multi-party confirmation for public affairs. A community can set up a public account managed jointly by several members. Operations involving public affairs take effect only after the number of confirmations agreed in advance is reached, so that no single person decides alone. This is exactly the spirit of the Radish & Cabbage Rules carried on in the digital world.

5. Communities can create their own community points. Different villages and different communities can, according to their own rules, create their own community points to record the contributions of local members.

6. One identity, many community apps. Semi is the "identity gateway". For example, in our community collaboration assistant "Bai", people publish and claim community tasks, and they log in with their Semi identity. When a task is completed in Bai, the corresponding community points and remark are recorded in the Semi account, so that tasks and contribution records match one to one. Applications built by other communities and developers can also connect to Semi, so that people do not have to register new accounts everywhere.

V. How It Works (Technical Section)

This part is for friends interested in technology. Readers who are not familiar with technology can skip it and just remember one sentence: the complex things are left to the system, and the user only needs a phone number, a verification code and a 6-digit payment code. Everything mentioned below can be found in the code and documentation published on GitHub.

1. Overall Structure: Three Layers

Semi consists of three parts:

- Front end semi-app: a single-page web application written with Nuxt 4 and Vue 3 (open the page and use it, no app to download), with a UI built on Nuxt UI and Tailwind CSS, state management with Pinia, and support for Chinese and English. All operations that interact with the blockchain are done with the viem library. The server built into Nuxt (Nitro) handles only the few things that must run on the server, such as writing badges to the chain.
- Back end semi-backend: a pure API service written with Ruby on Rails 8, with data stored in a PostgreSQL database. It handles login, account profiles, transaction records, multi-party proposals and signature collection, and single sign-on (OAuth). User and account identifiers are time-sortable TSIDs.
- Blockchain core semi-core: an independent module extracted specifically from the front-end code, handling the "one wrong calculation and things break" logic of smart accounts, keys, fee sponsorship and multi-party signatures. It does not depend on the Semi back end; the web client and the server share the same code, and it has unit tests as well as tests that run against a replicated environment of real on-chain state (fork tests), verifying that account address derivation and transaction execution match the real contracts.

2. Login: Just a Phone Number or Email

There are only two ways to log in: a phone SMS verification code and an email verification code; there is no password login. SMS messages are sent through Alibaba Cloud and emails through Resend.

The verification code is a random 6-digit number valid for 15 minutes; at any time there is only one valid code per phone number or email, and each code can be tried at most 5 times; sending frequency is also limited, to prevent SMS flooding or repeated code guessing. Verification codes are never written to logs. After a successful login, the back end issues the browser a login credential, which is sent with every subsequent request.

3. How the Account Is "Created Automatically"

Many people assume that a "blockchain account" means they must look after a recovery phrase themselves. Semi's approach is to hide that behind the scenes:

1. After first login, set a 6-digit payment code. The browser generates a set of keys locally (technically a standard 12-word recovery phrase, from which a signing private key is derived according to the BIP-39/BIP-44 specification); the user does not need to see it or copy it down.
2. Lock the keys with the payment code. The browser encrypts the keys with the payment code (PBKDF2 with 100,000 iterations derives the encryption key, then AES-GCM 256-bit encryption), producing an encrypted package (keystore) that is saved on the back end. The payment code itself is used only in your browser and is never sent to the server, so what the server stores is merely a "locked box" that it cannot open itself.
3. You get a smart account. Your account on chain is not a bare private key but a Safe smart account. Safe is a widely used open-source account contract on the blockchain; Semi uses version 1.4.1, together with the ERC-4337 standard (an "account abstraction" standard that lets accounts define their rules in code, for example having someone else pay the gas fees). The account address is computed in advance using CREATE2: you have a definite address as soon as you register, so you can receive points and badges right away, while the account contract is deployed to the chain only when you first actually do something.
4. Prove "this key is yours". When binding the account, the browser signs a fixed sentence with this private key; the back end verifies the signature and recomputes that the Safe address is indeed derived from this key. An account can be bound only once, and one key and one address can belong to only one account, so no one else can impersonate you.

From then on, every time you want to do something on chain, such as issuing community points, you only need to enter the payment code; the browser unlocks the key locally, signs, and submits. If needed, users can also export their private key in the settings and keep it themselves.

4. Gas Fees: Invisible to Users

Writing records to the blockchain normally requires paying a "gas fee", which is one of the biggest barriers for ordinary people. Semi solves this with two roles from ERC-4337:

- Bundler: submits the operations signed by users (called UserOperation) to the chain;
- Paymaster: pays the gas fees on the user's behalf.

Both services are connected through ZeroDev, and the on-chain node and query services use Alchemy. On the Optimism and Arbitrum chains, fees are paid from a sponsorship balance that Nantang has deposited in advance, and each account has a certain number of fee-free operations. Throughout the process, users do not need to know what a gas fee is, let alone prepare for it.

5. On-Chain Remarks: How One Sentence Gets Written into the Blockchain

This is where Semi and "Bai" work most closely together.

The remark contract. On chain there is a smart contract dedicated to storing remarks (RemarkLogicV1, deployed on Optimism, wrapped in an upgradeable proxy contract so that bugs can be fixed later without changing the address). It has only one write function, saveRemark, which takes four things: a task pool ID, a task ID, a public remark, and a note written by the person who completed the task. The contract stores these two sentences keyed by "task pool + task", records the time of writing, and emits a RemarkSaved event, so that they can be looked up by task later. It is responsible only for "attesting", nothing else. A remark is at most 32 characters.

Written together with community points, in one transaction. When issuing community points, Semi puts the two actions — "issuing points" and "writing the remark" — into the same on-chain operation (the same UserOperation) and submits them together: either both succeed or neither happens, so the points and the sentence attached to them are always bound together.

Matching up with the task in Bai. The flow works like this:

1. A partner claims a task in Bai and submits proof of completion, and can write a short note along the way;
2. After the person who published the task approves it in Bai and clicks to issue community points, Bai redirects to Semi's issuance page and passes along the task pool ID, the task ID, a default remark (for example "Task: <task title> + time", which can be edited) and that note;
3. Semi converts the task ID from Bai into an on-chain numeric ID by way of a keccak256 hash; after the issuer enters the payment code to confirm, the points and the remark go on chain together;
4. Afterwards, Bai's back end can look up the corresponding RemarkSaved event on chain by task ID, retrieve this issuance and its remark, and match them to the task.

Ordinary issuances that do not come from Bai can also include a remark; Semi generates a random ID for them.

Records can be verified. After an issuance is completed, the front end syncs the record to the back end so that it can be shown in both parties' histories. The back end does not simply accept it: it reads the receipt of this operation on chain, confirms that the sender really is this person's smart account and that the recipient really appears in this operation, and takes on-chain data as the authority for the gas fee and for whether it succeeded.

6. Community Badges and Collectible Display

Badges are built on a set of badge contracts: the community first creates a "profile registry" (the Profile Registry contract), then creates a badge type, and then issues it to members. After issuance, a badge is in a "pending acceptance" state; it is minted on chain only after the member confirms acceptance, and it can also be declined.

Writing a badge on chain requires a signature from an administrative account. This administrative private key is kept only on the server and never reaches the web page; when a user initiates an issuance or an acceptance, they sign with their own key to prove their identity, and the server only verifies the signature — it never has access to the user's encrypted package or payment code. To prevent repeated minting caused by network interruptions, the server checks on chain whether the badge already exists before minting.

The personal page also displays the digital collectibles in the account in one place, including badges and souvenirs. The display data is read through Alchemy's indexing service; images and descriptions can come from distributed storage such as IPFS and Arweave, and music-related souvenirs can even open a play link directly.

7. Multi-Party Confirmation: Writing the Radish & Cabbage Rules into Code

A community's public account is itself a Safe smart account, except that it has several members and a "threshold": for example, at least 2 of 3 members must agree for an operation to take effect.

- Proposing: any member can initiate an operation, and the back end places it in a queue awaiting confirmation.
- Individual confirmation: each member uses their own payment code to sign the proposal in their own browser (a structured signature under the EIP-712 standard, so that what is signed is clear at a glance). The back end collects the signatures and verifies one by one that each really comes from the corresponding member.
- Execution once the threshold is reached: when there are enough signatures, any member can submit it for execution on chain. The gas fee can likewise be sponsored by the platform.
- What you see is what you sign: to prevent "the page shows one thing but something else is actually signed", the front end regenerates the operation from the content displayed on the page before signing and compares it byte by byte with what is to be signed, refusing to sign if they differ; the back end also checks each time it receives a proposal and a signature. Moreover, each proposal may contain only one operation, with nothing else slipped in.
- Changes to membership also need everyone's consent: adding or removing a member, or adjusting the threshold, is itself a proposal requiring multi-party confirmation.

This is precisely the spirit of the Radish & Cabbage Rules — "the minority submits to the majority, and no single person has the final say" — carried out in a different technical form.

8. Communities' Own Community Points

Semi has a "contract factory": a community can fill in the name, abbreviation and icon of its points on the page, along with which account is responsible for issuing them, and deploy a points contract belonging to that community in one click. This contract uses the ERC-20 standard, which is common in the Ethereum ecosystem — a standard technical format for recording "who holds how many points" — and any tool that supports this standard can read it. This feature is currently deployed on Optimism. Newly created points are by default visible only to the creator; they appear in the public list that everyone can see only after an administrator reviews them, which prevents anyone from trading on another community's name.

9. One Identity, Many Apps: Single Sign-On

The Semi back end includes an OAuth 2.0 authorization service. This is the same standard behind the familiar "log in with WeChat" and "log in with GitHub", and it provides OpenID Connect-style configuration discovery and public key endpoints, so that other applications can integrate automatically.

- Developers can register their application on Semi's developer page, setting the callback address and the permission scopes they are allowed to request;
- Login uses the "authorization code + PKCE (S256)" flow, so that even if the authorization code is intercepted in transit, it cannot be exchanged for a login credential;
- Permissions are granted separately: basic profile (profile), on-chain account address (wallet), points viewing (token:read), and so on; users can see them clearly on the authorization page, and can later review and revoke granted applications in Semi at any time;
- The authorization page cannot be embedded in other web pages, which prevents "clickjacking".

This is how "Bai" connects: people click to log in with Semi in Bai, and after authorization, Bai finds or creates the corresponding user by Semi's user ID, so that the same person is the same identity in both systems, and points are never issued to the wrong person.

10. Which Blockchains Are Supported

Semi currently supports four chains, and you can switch between them at any time on the page:

- Optimism: the main chain for everyday use; fee sponsorship, the remark contract, the badge contracts and the community points contract factory all live here;
- Arbitrum One: supports smart accounts and multi-party confirmation, with fees likewise sponsored;
- Ethereum mainnet: supports smart accounts and multi-party confirmation, but fees are not sponsored and must be borne by the account itself, so it is generally not used day to day;
- Sepolia testnet: for developers to test with.

Safe's official contracts have the same set of addresses on these chains, so the smart account address computed for the same person is also the same across chains.

11. How Other Villages Can Deploy It Themselves

Both the front end and the back end of Semi provide Docker image build files, and can be deployed on your own servers:

- The back end needs a PostgreSQL database and its own SMS and email sending accounts;
- The front end needs the keys for the blockchain node service (Alchemy) and the bundler and paymaster services (ZeroDev), as well as the back-end address. All chain-related configuration is read from a single place, and a misconfiguration causes a clear error at startup rather than a problem only when a user acts;
- To connect a local collaboration tool, you only need to register an OAuth application in Semi.

"Bai" is open source as well: the front end is Nuxt/Vue, the back end is Node.js/Express with Supabase, and smart contracts such as the remark contract are written in Solidity and deployed with Hardhat, also on GitHub.

VI. Why We Think About "Points" This Way

In Nantang, community points are not a bonus and not a commodity; they are more like a shared ledger of contributions: a pot of food someone cooked for everyone, a line of code a distant partner wrote, can both be seen fairly in this ledger. What it seeks to do is turn contribution from the memory of a few into a shared record; in Nantang, this record is also an important basis when people take part in discussions of community affairs and make decisions together.

This idea draws on international thinking about "contribution accounting" and community mutual aid. Michel Bauwens, a Belgian scholar, is the founder of the P2P Foundation and has long studied sharing and collaboration. He and his colleagues proposed that a good way of recording should capture all kinds of contributions from each person, not only the labor that the market is willing to pay for; care, companionship and mutual aid — the "invisible work" — also deserve to be recorded. At the same time, we remind ourselves: recording is meant to improve relationships between people, not to turn people into cold strings of numbers.

On the question of "identity", we were also inspired by Yamakoshi Village in Japan. This mountain village, with only a few hundred residents, invites people around the world who care about it to become "digital villagers" and build a long-term connection with the village. We likewise hope that anyone who has been to Nantang, or who cares about the countryside from afar, can have an identity that "belongs here".

Rural construction is not just building roads and houses; it is a group of people creating a way of life together. The real difficulty has never been in technology, but in the shared life between people. Every contribution being seen is where that shared life begins.

VII. Open Source, Come Join Us

Rural Co-construction Digital Identity (Semi) is an open-source project written by the community together, with the code public on GitHub (github.com/nantang-dao), including the front end semi-app, the back end

semi-backend and the task system Bai; anyone can view, use and help improve it. Whether you are a returning youth, a digital nomad, a rural construction team, or a friend willing to write code or do design, you are welcome to help make it easier to use.

- Semi website: semi.ntdao.xyz
- Community collaboration assistant "Bai": bai.ntdao.xyz
- Code: github.com/nantang-dao
- WeChat Official Account: Nantang DAO

If your village or community also has a lot of "contributions that were never recorded", you are welcome to come and talk with us.