EASE Protocol Whitepaper

A Framework for Building Value and Data Platforms for Everyone

Douglas Horn, GoodBlock Holdings, LLC

August 24, 2024

AI Disclosure:

This original paper was written entirely and solely by the stated author, Douglas Horn, without the use of any large language model or other artificial intelligence tools whatsoever.

Douglas Horn

Digitally signed by Douglas Horn
Date: 2024.09.12 23:32:47 -07'00'

NOTICE

This paper provides a technical description of the intended features and benefits of the EASE Protocol. Computer development naturally iterates and changes over time due to perceived needs, market conditions, competitors, technological developments, legal and financial considerations. Any and all features described in this document are subject to change without notice for any reason the EASE Protocol project team deems necessary or beneficial for the success of the project.

THIS DOCUMENT IS NOT A SOLICITATION TO PURCHASE ANY TOKENS, EQUITY IN THE EASE PROTOCOL, AYETU BLOCKCHAIN, OR ANY OTHER ASSET.

Contents

EASE Protocol Whitepaper

A Framework for Building Value and Data Platforms for Everyone

The Need For EASE

The unfettered opportunity of blockchain technology, which once promised a revolution of personal financial power, ownership, and democratization of data for everyone, has been utterly squandered over the past decade by project founders seeking little more than to offer ever faster demonstrations of the Greater Fool Theory. Two factors are responsible: inexhaustible human greed, of course, but more pointedly, a failure by blockchain developers to target their projects towards the seven billion end users whose lives could immediately and immensely benefit from the technology instead of the tiny subset of users who, like the developers themselves, are extremely impassioned with computer technology, economics, and a fascination with addressing problems of the future that are yet to affect *anyone* today. In the developers' eagerness to attract other users uncannily similar to themselves, they have ignored the simple needs of the masses who have broadly rejected the clumsy interfaces and risky security concessions that adopting the technology has demanded. What is needed to reverse this trend is an entirely new approach to the target audiences, development priorities and even security decisions surrounding blockchain with an eye towards creating tools that practically *everyone* finds simple, accessible, and needed. With the right tools, even human greed can be made to serve a useful purpose.

The EASE Protocol (Easy Adoption, Sequestered Encryption) is a new paradigm in blockchain usage aimed at addressing the friction points that have limited mass adoption over the past fifteen years of blockchain systems despite clear areas of superiority compared to traditional monetary, financial and data recording systems.

As a new generation of blockchain technology, the EASE Protocol aims to better address the real-world risks to blockchain users such as hacking, private key loss, and exchange or bridge failures, rather than the esoteric risks like government seizure that concern a relatively small cohort of digital currency users. EASE places a higher value on the privacy of individual users compared to current blockchain systems that rely on "security through obscurity" of pseudonymous blockchain addresses that reveal all their secrets once pseudonymity is (often quite easily) broken. Beyond these, EASE blockchains provide numerous features to improve quality of user convenience, security, speed, capacity and practical features such as integrated messaging, contacts lists, identity and reputation features and more.

Networks built on the EASE Protocol will provide communities with powerful digital features, for users with low levels of digital literacy, and will do so while limiting or mitigating risks currently presented by blockchain technology. It will open the ability for any user to leverage smart contracts and blockchain immutability from simple interfaces without needing to deploy code, by offering highly configurable pre-compiled and audited contract executables. Communities will benefit from access to the world's most advanced digital governance platform, that can be configured by any community for their specific needs and rules. Numerous tools will be incorporated into EASE blockchains with the goal of simplifying the use of many digital tools and with the hope of empowering users to become entrepreneurs improving their own lives through the ability to build value for themselves and others. All of this will occur in a landscape of government acceptance resulting from a system that allows governments to investigate and

enforce their laws with their citizens and within their jurisdictions. In fact, governments can be expected to be among the beneficiaries of this technology which has the potential to greatly improve the efficiency and reduce the cost of operating revenue authorities, elections commissions, social services authorities, license and certification authorities, and even central banks and treasury departments.

The Approach of this Whitepaper

This document assumes the reader has intermediate or above digital literacy and at least a passing knowledge of blockchain technology. There should be no need to directly reference the *Bitcoin Whitepaper*[1], *Ethereum Whitepaper*[2], *Ethereum Yellow Paper*[3] or *EOS.IO Whitepaper*[4] as respective examples of first-, second-, and third-generation blockchain technologies, although the contributions of their authors and many others working to develop these technologies is gratefully and humbly acknowledged. The aim of this paper is to take advantage of the tremendous leaps forward that each of these new generations offered the world and to further contribute to the mission of bringing the democratizing and self-empowering advantages of blockchain to a broader set of users than ever before.

Document Structure

This whitepaper will first examine the shortcomings of current blockchain technology that necessitates a paradigm shift and the solutions presented by each of the core elements: Easy Adoption and Sequestered Encryption. Next, it will describe the new approach and feature set made possible by this new approach. Finally, the paper will provide a detailed discussion of the technical structure of this blockchain technology.

[1] Satoshi Nakamoto (2008), *Bitcoin: A Peer-to-Peer Electronic Cash System*, (https://bitcoin.org/bitcoin.pdf)

[2] Vitalik Buterin (2014), *Ethereum Whitepaper: A Next-Generation Smart Contract and Decentralized Application Platform*, (https://ethereum.org/en/whitepaper)

[3] Gavin Woods, *et al.* (1994), *Smart Contracts*, (https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html)

[4] Daniel Larimer (2017), *EOS.IO Technical Whitepaper*, (https://cdn.bitturk.com/whitepaper/eos.pdf)

Easy Adoption

Why is there a need for easy adoption features?

How and why have existing features failed to capture a broad user base?

What is needed to fix this?

Blockchains for Everyone

In 2018, GoodBlock adopted our friendly logo/mascot, "Blocky" and the motto, "Blockchain is for everyone" based on the realization that blockchain technology inherently drives towards democratization and self-empowerment. This ethos was imbued in the creation of the Telos Blockchain[5] in the same year. However, six years later, it has become clear that if the benefits of blockchain technology are ever to benefit the masses of humanity, they will need significant improvements in usability and forms of security that are aligned to the needs and real world challenges of everyone, not just the technical elite.

A new generation of blockchain tools based on ease of access and better security must arise for these tools to truly be accessible to everyone. Specifically, significant changes in the systems operations need to develop in five categories: Ease of Use, Loss Prevention, Privacy, Acceptance, and Connectivity.

Ease of Use

Current blockchain technology is kludgy and cumbersome, requiring management of keys, separate "wallet" apps for signing transactions, resource management and/or "gas" fees, "nonces", "gas limits", spending approvals and other impediments to easy, confident use. Most users follow the extra steps in the process under the belief that they somehow make their experience more secure, but without really knowing how.

Interfaces and systems of next-generation blockchains will need to be as easy to use as typical internet (a.k.a. "Web2") applications and use similar or identical tools. Signing blockchain actions needs to occur within the primary interfaces, without launching third-party "signing wallets" such as MetaMask, which then require additional decisions and confirmations. Functions should make intuitive sense to people familiar with typical computer applications and not require an additional tier of specialized knowledge.

Loss Prevention

Current blockchain systems are rife with loss for users. While the private key encryption system itself is immune to cracking when used correctly, practically everything else in the ecosystem presents a near constant risk. The use of private keys forces users to navigate a schizophrenic protection mechanism where they must be careful to record their keys in a way that they will never lose, while simultaneously keeping them secret from anyone else lest the keys be stolen and their accounts emptied. Third-party bridges and exchanges that exist to simplify and extend use across multiple chains are, in fact, common arenas of user losses due to hacks. Phishing attacks, where attackers leverage the complexity of the systems to trick inexperienced users into giving up their assets, are often a product of the hard-to-use interfaces.

[5] Douglas Horn (2018), *Telos Whitepaper: A Sustainably Decentralized EOS.IO Network*, (https://www.allcryptowhitepapers.com/wp-content/uploads/2019/10/telos_white_paper_english.pdf)

The next generation of blockchain systems must prevent or mitigate the major causes of loss: lost private keys, third-party tools, and social engineering (phishing) attacks. These risks are the byproduct of traditional blockchain security decisions.

Privacy

Existing generations of blockchain systems fail to protect user privacy. Though users commonly believe that these systems provide a high level of privacy because the addresses are pseudonymous, in fact, blockchain addresses and accounts are all easily viewable in their entirety to anyone using a public block explorer. As a result, once a person's identity *is* linked to a Bitcoin address, whether by sending someone a payment, or by being analyzed by the sophisticated chain analysis tools that have existed for governments and corporations for nearly ten years, their entire balance and transaction history is free for the viewing. This presents a real risk to users who can become targets for theft or violence and businesses whose transactions are all exposed to their business rivals. Most blockchain users remain unaware of this risk.

The next generation of blockchain technology must protect user privacy. Account data should only be viewable by their owners. Transactions will need to be immutable and provable, while remaining private.

Acceptance

A blockchain system that is denounced as illegal by governments is of limited value to general users who must live their lives within the bounds of laws, regulations, and enforcement bodies. The shaky status of existing generations of blockchain technologies in the eyes of most governments introduce significant challenges to use of these digital currencies compared to fiat-based digital payments. Merchants cannot feel comfortable accepting digital currencies as methods of payment unless and until they receive some form of acceptance from their government and financial infrastructure.

The greatest challenge for new blockchain systems will be to build broad acceptance through a near ubiquitous network of businesses that accept these digital currencies and can use them to pay their own debts. This will require government acceptance and a basis in law that can never emerge while blockchain systems have an adversarial relationship with governments based on features that make regulation and compliance impossible[6]. The next generation of blockchain must forge a better relationship with governments so as to provide users more security.

Connectivity

Current blockchain systems must store private keys on the user's device and use them to encrypt all transaction instructions and key signatures, and then send these to the blockchain. This requires both a higher CPU demand of the user's device and a larger amount of network bandwidth to transmit the signed transactions. For users with powerful devices and access to high-speed internet, this poses very little problem, but in areas where internet connectivity is low and/or costly, and where users must make do with low-powered devices, the increased demands pose real challenges not faced by Web2 apps with lower requirements, which perform far better in these situations as a result. And yet these are the conditions of the majority of the world's people who live in developing economies or rural regions–even of the world's richest nations. The next generation of blockchain must become more accessible to these users by requiring lower CPU power and reduced network bandwidth, if it is truly to be for everyone.

[6] This illusory security against government intrusion does not actually exist for 99.999% of blockchain users.

Missed Opportunities Due to Difficult Adoption

Despite their unique, powerful and deeply needed advantages, Bitcoin and other cryptocurrencies have failed to achieve broad adoption in the decade-and-a-half since they emerged. While adoption of radical new technologies takes time, these currencies have impediments to adoption that are inherent to their core operations, meaning that difficult adoption will remain throughout the lives of these technologies. Bitcoin cannot change its private key cryptography and remain Bitcoin, meaning that all the difficulties associated with private keys, such as the steep learning curve, risk of loss and expensive transaction fees, are baked into the protocol and will always make adoption difficult for the vast majority of users.

Bitcoin is and will continue to be a revolutionary invention and unmitigated success, but it will not achieve the mass adoption that a secure, yet easy-to-use system could. Of course, some of Bitcoin's unique protections would not apply to such easy adoption technologies, but these protocols can solve other problems to empower users.

New usability improvements to Bitcoin *et al* that have been offered in recent years are largely in the form of additional layers of interfaces and third-party intermediaries trying to create more convenient, adoption-friendly infrastructures around Bitcoin, such as wallets, bridges and exchanges, but these conveniences are at the very expense of Bitcoin's native security protections.

The EASE Protocol takes a different approach by giving up some of the protection of strict private key cryptography self-custody in favor of **sequestered encryption**, where private keys are stored in an enclaved system that no one can access and used to sign blockchain transactions without ever leaking private keys. This allows such conveniences as **single sign-on**, lost key replacement, compliance with government regulations, secure bridges and exchange functions without relying on intermediaries, among others.

For the vast majority of potential users, this simplicity, convenience and enhanced security around real-world risks they face more than offsets the protections native to Bitcoin. Neither is inherently better, but one may be more useful for specific users or uses. Naturally, some will claim that EASE is not as secure as Bitcoin, which may be true, but then neither is Bitcoin, itself, in the way it is actually used by the vast majority of users, who rely on exchanges, bridges or custodial wallets to make its use convenient and understandable.

Sequestered Encryption

What security assumptions are built into private key cryptography systems and do they really apply to the majority of users?

How does requiring users to manage their own private keys reduce overall security for most users?

What is sequestered encryption and how does it improve functional security and usability?

When More Protection is Less Secure

The EASE Protocol is the product of many years of work by the GoodBlock team in assessing the shortcomings of current blockchain technology in terms of adoption and functional security. Existing blockchain systems address security through private key cryptography intended to be entirely self-managed by each user. This requires all users to adopt more stringent and demanding security protocols than they are familiar or comfortable with and with every transaction they must perform additional checks and confirmations they often don't understand. When crypto users are asked to approve smart contracts to transfer their funds, they rarely understand the risks involved, or it's unlikely they would routinely approve the spending of more than trillions of tokens, which most DeFi users do when approving the default transfer amounts. These transactions are more cumbersome and confusing than the credit card or payment systems they're used to and therefore, feel outdated and clunky.

Attempts to simplify use of private key systems introduce a variety of third-party risks, which account for the vast majority of losses from hacks. Further, the loss of one's self-custodied private keys results in the loss of all funds. These scenarios challenge adoption and introduce risks of catastrophic loss. Paradoxically, the high level of protection is, in practice, making most users **less** secure. This is directly at odds with the prevailing trends of general computing which has moved to simpler interfaces and single sign-on security. As general computing becomes easier, the difficulties of private key management become ever greater impediments to adoption.

Sequestered Encryption reverses this paradigm and allows easy adoption through the types of access tools that general computer users are already familiar with for the majority of modern computer interactions, such as social account sign-in, while allowing key management that is sequestered from other operations on enclaved servers which do not give *anyone*, including the system administrators or validator nodes, the opportunity to use the keys themselves, and also completely eliminate the risks of loss through transmission of the private keys. This section describes the rationale, architecture, implementation and use cases for such a sequestered key management system (SKMS) that securely encrypts and manages private keys without entrusting custody of the accounts to the managers. In this manner, the EASE Protocol brings both ease of use and greatly improved asset security when measured against real-world security challenges.

The Private Keys Problem

The blockchain industry was born from Bitcoin creator Satoshi Nakamoto's desire to give humanity a better form of money that would have a fixed, programmatic inflation, secure transmission, and freedom from government control. Bitcoin users were meant to be free from anyone forcing them to spend their money in the manner that banks and governments can confiscate fiat monies. Bitcoin and some of the digital currencies that followed it provide a powerful monetary tool for many people, but no single system can be expected to solve all problems related to usage, security and adoption, and the Bitcoin Protocol is no exception.

Self-custodied private key security as used by Bitcoin and most digital currency systems necessitates extreme key security measures, unfamiliar interfaces and numerous ecosystem risks that, for the majority of users, have so far proven to limit its utility. The difficulty users experience in trying to understand these interfaces opens vectors for asset loss.

By entrusting total control for assets under a private key that is unrecoverable if lost and has the ability to control all assets if found by someone other than its rightful owner, these systems facilitate frequent losses from theft of eternally frozen assets. By making all asset transactions and holdings visible to anyone who can connect a person's identity to an address, these systems reduce financial and personal privacy and can make targets of individuals. By having unclear legal status with governments, these systems force their users to go through inefficient and risky means of use and exchange that empower intermediaries rather than disintermediating transactions as originally envisioned. In practice, the real-world usage of these systems is often the opposite of the touted protections. In short, the actual risks digital currency users face: theft, phishing, rug pulls, bridge hacks, lost keys, targeting and exchange exploits create losses far outweighing the protections gained by self-custody for a majority of users. As a result, adoption has been slowed and countless assets have been lost without hope of recovery.

Bitcoin is a phenomenal protection against government intrusion for those who need this, but many other people in the world face different needs that must be addressed with digital currencies employing different sets of protections.

Protection Selection

No one could live a useful life while protecting themselves against every imaginable threat, every moment. We must assess risks by how dangerous they are, how they might be mitigated, and how likely they are to affect us.

The hippopotamus, for example, is one of the most powerful and dangerous animals in the world. They are a real threat, killing more humans each year than any other wild mammal, including elephants, big cats, wolves and bears combined. However, for most people, avoiding a hippo attack is very easy: stay away from African rivers and don't enter zoo enclosures. Heart attacks, on the other hand, are a widespread danger affecting far more people and lowering one's risk requires a different set of precautions like a healthy diet, stress management and frequent exercise. Both of these risks are entirely real, however, most people would extend their lives more by protecting against mundane risks like heart attacks rather than exotic ones like hippo attacks.

When it comes to securing digital assets using systems where users directly manage their own private keys, the protections are against government appropriation of private funds, and not loss of keys, phishing, or the other risks associated with the current blockchain security mechanism. Far more digital assets are lost to these risks than have ever been seized by governments. In other words, current

blockchain technologies protect us against risks like government asset seizure that, for most people, are as unlikely as hippo attacks, where the EASE Protocol aims to protect against much more common risks–metaphorical heart attacks–like lost private keys or theft from bridges or exchanges. Again, both types of attacks are very real, but one applies to a much larger group of people[7]. EASE does not aim to supplant the enormously useful Bitcoin. Instead, EASE employs different tools to address different risks.

Even the protections purported by Bitcoin can fail to hold up in the real world. There have now been many examples of threats of violence or imprisonment forcing asset owners to willingly give up their private keys to thieves and/or governments in exchange for their lives or liberty. In reality, there are few people who have the expertise or means to live out their lives entirely outside the reach of determined criminals and/or government agents. Even the computer security genius John McAfee, with all of his assets and expertise was unable to avoid government reach and, as some believe, extrajudicial execution[8].

Given the futility of fully resisting risks of governments overwhelming force for most people in the world, the protections offered by Bitcoin and the like have practical limitations. A system that gives less importance to trying to escape all government risks and instead focuses on more common sources of asset loss can still improve many problems with traditional monetary systems and will be better for mass adoption, and no less secure, when real-world security aspects are considered.

Sequestered Encryption and Programmatic Custody

If users should not be burdened with self-custody of their private keys, who can be trusted to manage their keys for them? The answer is, no one.

The current paradigm of private key (and therefore digital asset) custody is that *someone* is responsible for the keys: either the owner, which is self-custody, or a third party custodian. Sequestered Encryption, as used in the EASE Protocol, adds a third type of key management arrangement, *programmatic custody*, in which no one has control over private keys and instead, they are programmatically controlled by the autonomous system and accessible only to their owner. The private keys themselves exist in an enclaved space that no one can access and are never transmitted from that space.

The only power that the system administrators possess is to allow the keys to be reissued in the event of a verified owner request or valid court order from an account owner's jurisdiction. When such a key reissuance occurs, the private keys are stored in encrypted form within the enclaved space and only the public keys derived from the private keys are ever transmitted from the SKMS back to the owner's account. Reissuing private keys protects against key loss, theft and transmission interception, while leaving a clear and immutable record of the key reissuance to signify the change in status, should there be any questionable activity.

Since no one has control over the private keys, this system is neither custodial nor self-custody, but programmatic custody, at least until a key reissuance initiated by someone other than the owner occurs. In the case of a valid court order, the keys are remanded to the custody of the court or its representative and managed under the laws of that jurisdiction.

---

[7] Consider, also, that the fact that governments do, in fact, still manage to seize cryptocurrency assets, despite their foolproof protections illustrates that the existence of these protections does not provide the iron-clad guarantee that many people hope for.

[8] John McAfee was arrested in Spain for US tax evasion charges and while held in a Spanish prison awaiting extradition died by hanging in his cell not long after publicly claiming that he would never commit suicide. There's no way to know exactly what happened, but the circumstances are suspicious and his widow does not believe he killed himself. Whatever occurred, McAfee's security expertise and self-custody of his digital assets did not provide him the free life he expected. Musumeci, Natalie (6 July 2021). "Antivirus mogul John McAfee's wife says she doesn't believe he died by suicide". *Business Insider*. Archived from the original on 17 July 2021. Retrieved 17 July 2021.

The Origins of Sequestered Encryption

The Sequestered Encryption system arises from how private keys are stored on devices. There are several ways that one could store keys on a device: they could exist in plain text in an unencrypted file, or in an encrypted and/or coded file. In the early days of blockchain, these were the options available to users and key security was weak. As crypto and other encryption usage became more widespread, devices introduced enclaved key storage. Enclaved storage exists on discrete zones of CPUs that only store encryption keys. Rather than enter the keys to the signing program, where they could be exposed, the signing programs now send the unsigned transaction to the enclaved space for signing. Overall, this is a far more secure system than the previously mentioned options.

In the examples above, the keys are under the control of whoever has access to the files, unless they are stored in the enclaved space, in which case, they are entrusted to the device's CPU, itself, for its use in signing and no one can retrieve them.

The EASE Protocol's Sequestered Encryption expands on this concept by using an enclaved server to generate the private keys as well as manage them, meaning that the private keys never exist anywhere except within that enclaved space. They are stored in encrypted form with the associated decryption key stored in the owner's user account, such that it can be retrieved only by the owner when signed-in to the system with a valid login JSON Web Token (JWT).

The system administrators do not have the ability to access the decryption key or the private key. The one vector of control they have is the ability to cause the enclaved server to reissue new private keys for the account, in order to allow for government compliance or the recovery of lost private keys (a truly remote possibility in the context of the system described).

Always Know Where You Put Your Keys

With the user freed from guarding their private keys, in favor of signing by the keys in the enclaved server, blockchain transactions no longer have to be signed by their devices or third-party wallets and transmissions are now in the form of API calls requiring little network bandwidth instead of relatively large signed/encrypted blockchain transactions, and the device CPU did not need the power to compute the signature. The user does not need to perform additional confirmations, nor deal with any error-handling or resource management issues that may arise from the proposed transaction. These can all be checked and addressed by the system as the final transaction proposal is assembled for sending to the enclaved server. In short, the majority of the user experience and technical challenges of previous blockchain generations are solved.

The problems of ease of use and connectivity can now largely be addressed in ways not previously possible. The need for separate wallets like MetaMask are now eliminated. Account owners can sign in and control their accounts from the same authorizer systems used by Web2 apps. Connectivity is also now on par with Web2 apps. Loss Prevention and Privacy can now be addressed by building systems that constrain access to information based on policies around account ownership. An account owner can see their own balances and transactions, but no one else can, except for seeing their own transactions to or from those accounts. Government agencies will have the ability to view account information for transactions occurring or accounts domiciled in their jurisdiction, within the constraints of their pertinent laws.

In providing these two concessions to governments: the ability to take custody of an account under a court order, by issuing it new private keys, or providing auditability of accounts in accordance with local laws, networks built on the EASE Protocol will address the needs for acceptance by governments and, in tandem, merchants and financial institutions.

The remainder of this document will describe the key features of EASE enabled by these modifications and explain their architecture and operations.

EASE Protocol Core Features

There are 30 core features of the EASE Protocol that improve the adoptability and usefulness of the platform. Each addresses a specific adoption impediment of current blockchain interfaces. Taken together, they create a powerful, yet easy to use infrastructure for not just personal finance, but entrepreneurial operations at every level. EASE users who previously did not have access to commercial tools–especially at lower socioeconomic levels–will suddenly acquire the ability to turn almost any productive endeavor into a small business with payment, escrow, messaging, contract notarization, proof-of-creation, no-code smart contract deployment, a built-in point-of-sale system and more. Almost the only thing that EASE blockchains cannot provide is the entrepreneurial spirit.

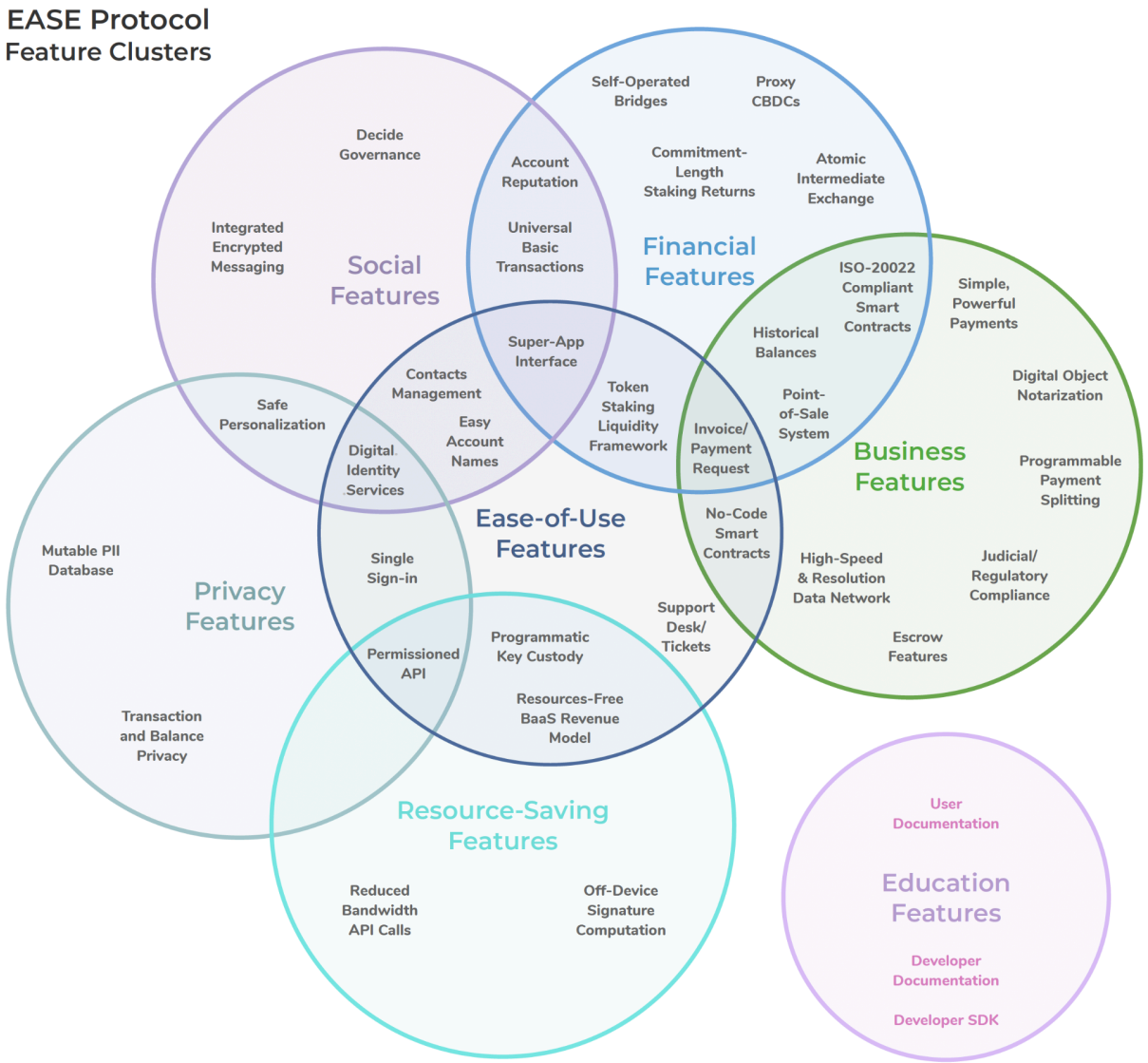


Fig 1. EASE Protocol Feature Clusters

EASE Features List

#	Feature Name	Cluster	Problem Addressed
1	Single Sign-On	E P	The need for additional "wallet" apps and configuration is confusing.
2	Programmatic Key Custody	E R	Private keys can be lost or stolen.
3	Judicial/Regulatory Compliance	B	Inability to comply with governments reduces acceptance.
4	Resource-Free BaaS Revenue Model	E R	Gas fee and resource cost revenue models challenge users.
5	Universal Basic Transactions	S F	Access to economic systems is limited to those with existing assets.
6	Super App Interface	E S F	Multiple interfaces make networks confusing and allow phishing.
7	Simple, Powerful Payments	B	Payment systems can be complex and costly for users.
8	Programmable Payment-Splitting	B	Single-recipient transactions can be inefficient for many uses.
9	Transaction and Balance Privacy	P	Public blockchains reveal all transactions and balances to anyone.
10	Historical Balances	B F	Public blockchains cannot provide past token balances.
11	Proxy CBDCs	F	Stable-coin providers present third-party risks.
12	High-speed & Resolution Data Network	B	Many areas lack affordable, reliable, scalable data recording.
13	Reduced Bandwidth API	R	Self-signing blockchain wallets are challenging for users of low-powered devices and/or low bandwidth networks.
14	Easy Account Names	E S	Complex address names confuse users and can lead to asset losses.
15	Safe Account Personalization	S P	Account personalization may reveal personally-identifying information.
16	Contacts Management	E S	Lack of integrated social connection creates risk of identity-spoofing.
17	Integrated Encrypted Messaging	S	Using separate and insecure messaging puts users at risk of loss.
18	Account Reputation	S F	Lack of account reputation allows known bad actors to persist.
19	Digital Identity Service	E S P	Identity systems reveal unwanted personal details.
20	No-Code Governance System	S	Governance tools are lacking and those that exist are cumbersome.
21	No-Code Smart Contracts	E B	The difficulty of deploying smart contracts limits their usage.
22	Atomic-Intermediate Exchange	F	Existing liquidity systems are inefficient across multiple token-pairs.
23	Commitment-Length Staking Returns	F	Existing staking systems fail to align LP incentives to provide system stability and ample liquidity.
24	ISO 20022 Compliant Smart Contracts	B F	Existing systems do not incorporate the incoming standard for global banking ecosystems at smart contract level.
25	Self-Operated Bridges	F	Third-party cross-chain bridges introduce risk of hacks and asset loss.
26	Digital Object Notarization	B	Many people lack an easy system for proving their priority in creation or ownership of digital objects or notarization of contracts.
27	Mutable PII Database	P	Immutable blockchains are incompatible with "right to be forgotten" laws.
28	Point-of-Sale System	B F	Small entrepreneurs do not have access to financially integrated PoS systems that would benefit their businesses in many areas.
29	Escrow System	B	Buyers and sellers have no easy, dependable, inexpensive way to ensure fair commerce.
30	Programmatic Key Custody	E, R	Private keys can be lost or stolen leading to asset loss.

Fig 2. List of EASE features and the problems each addresses

The Architecture of EASE

Features Overview

The EASE Protocol utilizes the core contracts and structure of the Antelope.IO Leap 5.x software with several modifications.

RPC and API Access

To preserve user privacy, the public RPC node typically used in Antelope.IO systems is replaced by two routes: 1) for users of the Super App interfaces, a privacy-securing collection of API endpoints, and 2) for Antelope.IO legacy apps and users who wish to use Antelope.IO wallets with key self-custody outside the SKMS, a RPC Relay, which replicates the RPC operated outside the system firewall and, in turn, talks to a filtered RPC system inside the firewall, to maintain the privacy filters on all returned data.

Key Management

Private keys are stored on the SKMS servers behind system firewalls that are encrypted and inaccessible even to the server operators. To access these, access tokens are used to decrypt users' private keys within the SKMS to return a signed version of a submitted transaction and broadcast it to the blockchain nodes. This allows both lost key recovery and transaction signing via low-bandwidth API calls that do not require CPU-intensive private key signing operations which can tax some low-cost mobile devices.

Resource Management

System resource management which is often a challenge for Antelope.IO systems is entirely removed from user view, with the system managing resources holistically based on each account's Universal Basic Transactions or subscription tier-based daily usage allotment where users will never experience a transaction failure due to insufficient RAM or resource staking.

PII Management

Personally-identifying information (PII) is secured on an off-chain database to allow for GDPR-compliant "right to be forgotten" abilities that would not be possible with on-chain recording. The PII Servers will index their records to on-chain account names, thereby extending the security policies of the chain itself.

EASE–EVM Bridges

The cross-chain bridge system will manage transactions between EASE and select EVM blockchains such as Ethereum. When AET tokens are sent from one EASE blockchain to an EVM chain, the native AET tokens will be stored in the EASE Bridge Manager smart contract and the same amount of wrapped AET (a.k.a. wAET) tokens are minted by the wAET token smart contract on the EVM chain. The AET balance on the EASE blockchain will remain the same, while the amount of native AET tokens held in the staking contract will always match the amount of wrapped tokens on the EVM. While on the EVM, tokens can be transferred and traded like any other ERC-20 standard token. When any wAET token-holder wishes to send these back to their native EASE blockchain for native tokens, they will be burned using the `bridgeburn()` action (which must include the recipient account name), triggering the release of the native AET tokens. All data about bridge transactions will be recorded in ISO 20022 format.

EASE–EASE Bridges

A similar, but more powerful version of the bridge will be available between EASE blockchain ecosystems. In addition to what EVM bridges can do, Bridges between two EASE blockchain ecosystems will be able to exchange native messages and ISO 20022 messages and actions, a select set of smart contract actions can also be triggered along with a transfer, creating a highly functional network of

cross-chain operations. This will allow different ecosystems to interact with simplicity, clarity, and power. Banks on one EASE system will be able to easily work across multiple EASE ecosystems with access to ISO 20022-empowered smart contracts.

Additional Features

Several new action types have been developed over and above typical Antelope.IO features. These include the liquidity system, API access, batch transfers (allowing transaction splitting), ISO 20022-compliance features, no-code smart contract deployment, and a robust and extensible governance system.

System Design

EASE Protocol blockchains, like any computer system exist as a lattice of computer hardware (such as servers, routers, firewalls), network infrastructure (local, regional, and global interconnected network cabling), protocols (TCP/IP, HTTP, SSL, SSH), software (operating systems, monolithic applications, middleware, micro-services) and people (end users, system administrators, developers), and the creation and continued existence of any such system relies on the presence of each necessary component. EASE blockchains can exist in almost endless variations without altering the core code, and with only modern modifications of the configuration rules and interfaces. All EASE blockchains will be identical or very similar in terms of protocols, software, and generally similar in terms of hardware and network infrastructure, meaning different brands and models of hardware may be used and networks may be broad or constrained, but the functions of these elements does not change within the system. The primary difference between EASE blockchains, then, will be the people: the project architects seeking to offer a solution to a problem and the end users who will use the system if that problem is real and the solution offered is valid.

It's crucial, even at the most technical level of description and documentation, to always bear in mind that the needs of real people must always be at the heart of any design, if the system is to have value. The driving organizational idea behind the technical architecture of the EASE Protocol is to provide a clear framework for any community of potential users to design a system for their own unique needs, without requiring the ongoing involvement of developers as core project managers and owners. Any group can roll out an EASE blockchain community and simply contract the technical deployment, operations, and system administration to professional infrastructure operators.

The Role of Blockchain Developers

The EASE Protocol seeks to address the perennial problems that have limited mass adoption of blockchain technology and the list of EASE features represents that. However, one intractable problem is not included in the list: the ongoing reliance on blockchain developers as system operators and project founders and leaders. Across Layer One blockchains (blockchains that maintain their own consensus without relying on a more foundational blockchain beneath them, such as Bitcoin and Ethereum), the project founders consistently play an outsized role in the fate of the project–especially its governance. This is a product both of the initial token economics established by the founders and the ongoing dependence on developers for decisions about project direction, since most of these projects are launched in an unfinished state and, by design, will be in an eternal state of not-quite-done-but-good-enough-to-use. The central role of developers driven by technical developments over present end-user needs is directly responsible for the slow adoption of the technology to end-users who are not, themselves, also highly computer-literate people. The decisions made at a project's establishment and after invariably serve technical needs and the financial interests of the current leaders much more than the end-user community.

A central goal of EASE is to provide a ready-made technology stack for user communities that do not wish to hand over all technical design to developer-minded project founders, but rather to the community and its self-defined needs. Consider this analogy: when a town needs a new storm sewer

system, it does not generally have to surrender all key decisions about the city's budget, economic development, emergency services, and the like to the contractors who built and operated the storm sewer system. Instead, they contract this service out and take reasonable operational advice from the experts where needed. They pay to keep it going and if the contractors who operate the system no longer meet the town's needs, they replace them with another operator who does.

As in this scenario, too many blockchain-based communities just hand over the keys to the town to those who build and operate the blockchain system, at the expense of all the other systems that are necessary to run a functioning community. This is one reason why DAOs so often fail. EASE is designed to stop this by providing a full set of tools that can be configured to meet all of the community's needs simply and efficiently. Just as end-users should not need to learn clumsy interfaces and security measures just to enjoy the benefits of blockchain, community leaders should not have to bring in experts and give them an ongoing, greatly outsized voice in all future community decisions. The appropriate role for blockchain developers and operators is to deploy and maintain the system, not to run the community it serves.

The Role of Antelope.IO Software

In 2018, a highly anticipated blockchain protocol was released following a contentious four billion dollar ICO by a company called Block One. The free and open source software (FOSS), named EOS.IO, was built on the C++ Boost Library that had been used for previous implementations of similar blockchains, referred to as Graphene, and was maintained by Block One for about two years and then abandoned four years after launch. At this point, a group lead by Yves LaRose of EOS Network, Ted Cahall formerly of Block One, Douglas Horn of Telos, Guillaume Babin-Tremblay of UX Network and Lukas Sliwka of WAX, created a coalition of blockchain communities built on the EOS.IO software to take over the software which suffered from at least two years of software development that was crucially needed but not performed, a.k.a. "technical debt". The codebase was revitalized and renamed Antelope.IO, with the C++ implementation, previously known as "eosio," was dubbed "Leap." Leap began at version 3.0 which addressed most of the critical technical debt. Over another two years, Leap developed to version 5.x, which addressed the majority of identified software flaws in the original design and added some long-promised features such as Inter-Blockchain Communications (IBC) which would allow for secure, decentralized token bridging across different Antelope.IO-based blockchains as well as horizontal scalability across multiple joined implementations of a "single" blockchain, meaning that blockchains could grow at enormous scale in ways invisible to all but the most technical users.

Antelope Leap 5.x is the terminal implementation of the software with future versions developed under the name Spring and offered under a license that is no longer free or open source. The Leap 5.x software, however, is a marvel of design with more features, security and configurability than any competing technology. A large base of users and developers exist, meaning that the "terminal version" does not suggest that it's dying, but rather that it has arrived at its intended destination.

The adoption problem for Antelope.IO blockchains was never that it lagged on a technical basis, but instead that it was based on a faulty assumption that its consensus mechanism, DPoS, could function in a live decentralized environment without capture and a resulting shift of all network governance and resources towards serving the capturers. ("Capture," in this context, refers to a state where a small group of users–typically the developers and/or largest token-holders–gain persistent control of its governance and macro-economic functions, particularly in cases where the original aim of the blockchain was to be decentralized.) There are several reasons for this, a key one being the constant inflation of the system token supply to pay validator nodes, who are elected based on token-weighted voting and whose actions can control all aspects of the blockchain.

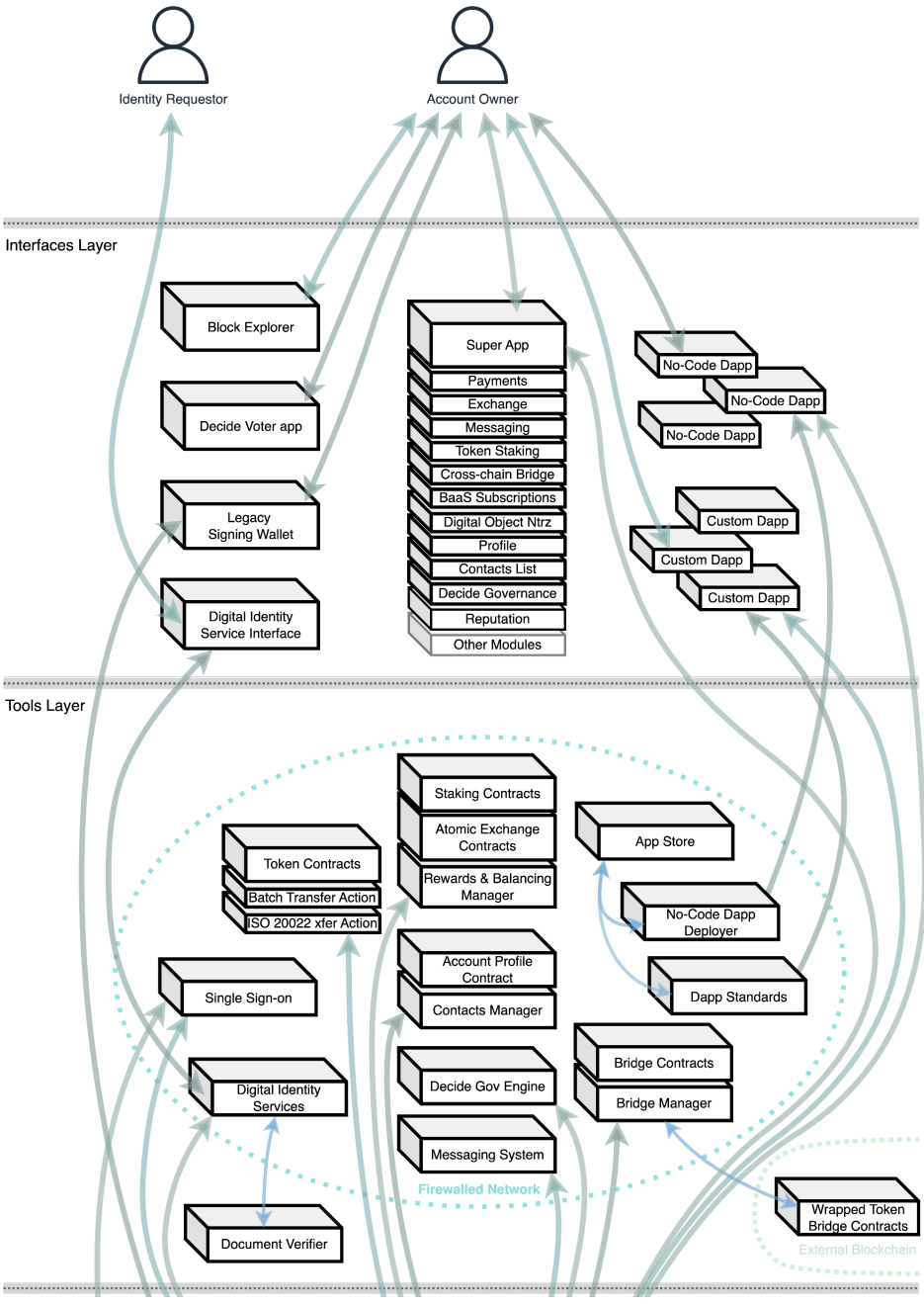
While Antelope.IO proved to be unsuitable for the purpose of running fully decentralized blockchains with system token inflation, it is, actually very useful and accomplished as a back-end for centralized computing systems or, in the case of EASE, for blockchain systems with limited centralized features and no system token inflation. AntelopeIO Leap is well suited to this purpose,, managing several perennially

difficult problems of computer system architecture such as network resiliency, immutability and resolving concurrency issues across multiple nodes. For this reason, EASE Protocol blockchains employ the Antelope.IO Leap 5.x software and view it as a complete software implementation, addressing all system needs, so as not to require maintenance or updates in the future, except to support version updates of underlying software dependencies.

A Visual Tour of EASE Protocol Design

The technical design of the EASE Protocol can be understood by viewing the five layers that make up its operational stack: Network, Node, API, Tools, and Interfaces. This section will look briefly at the system as a whole and then describe each element of its stack, layer by layer.

EASE Protocol System Architecture



(continued on the next page)

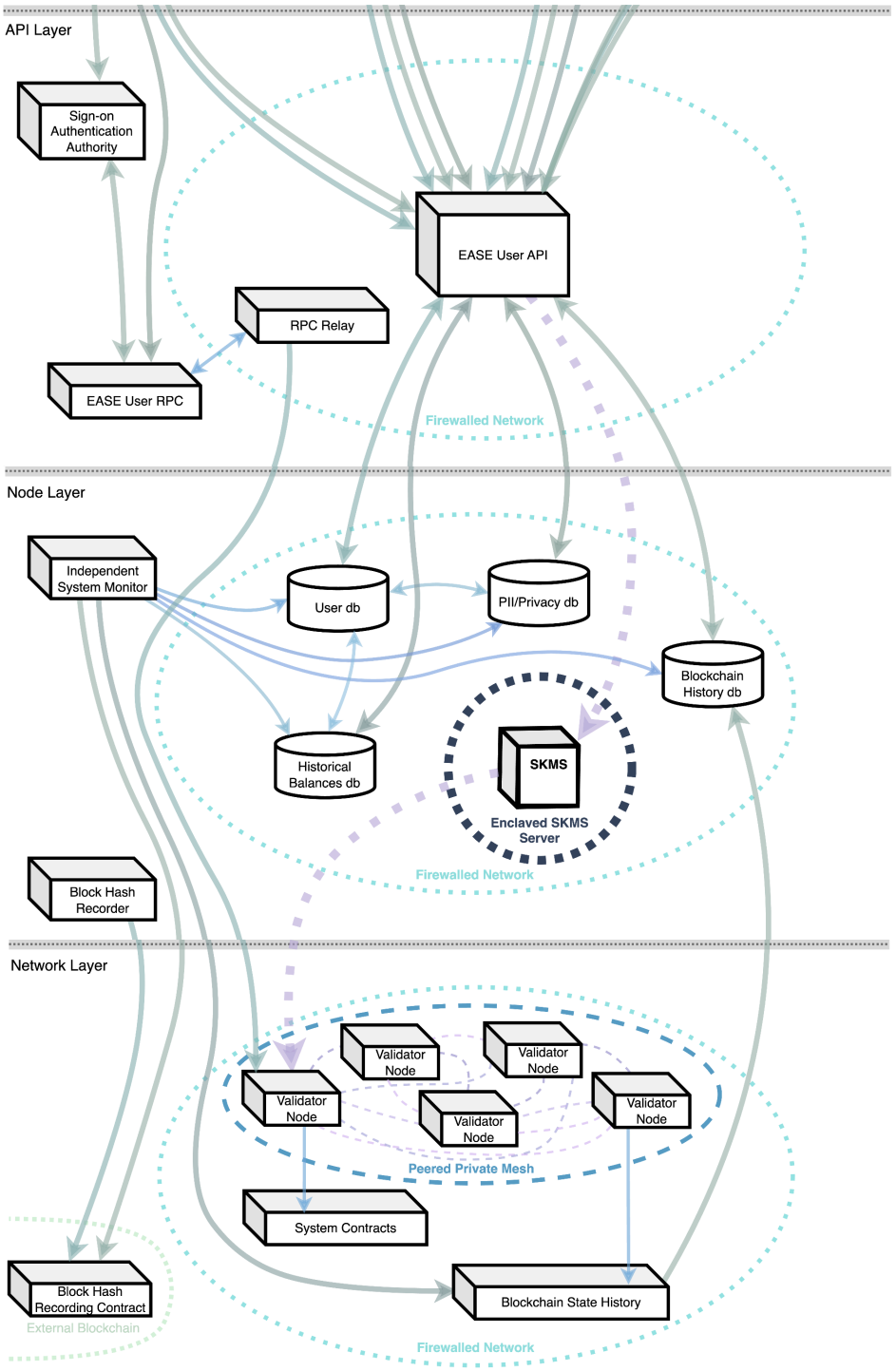


Fig 3. (in two parts) The EASE Protocol System Architecture Diagram

Network Layer

The layer underlying the system consists of a firewalled, permissioned blockchain built on the Antelope.IO Leap 5.x software and ideally operated by a system of homogeneous, ultra-high performance Validator Nodes all running identical versions and configurations of all OSs and software including the EASE System Contracts.

Optionally, other public blockchains may be used to record the block number and block hash of each block on the EASE chain to add further transparency and immutability to the blockchain. For example, the Ayetu Mainnet has recorded its block hashes to the Telos Mainnet since its inception.

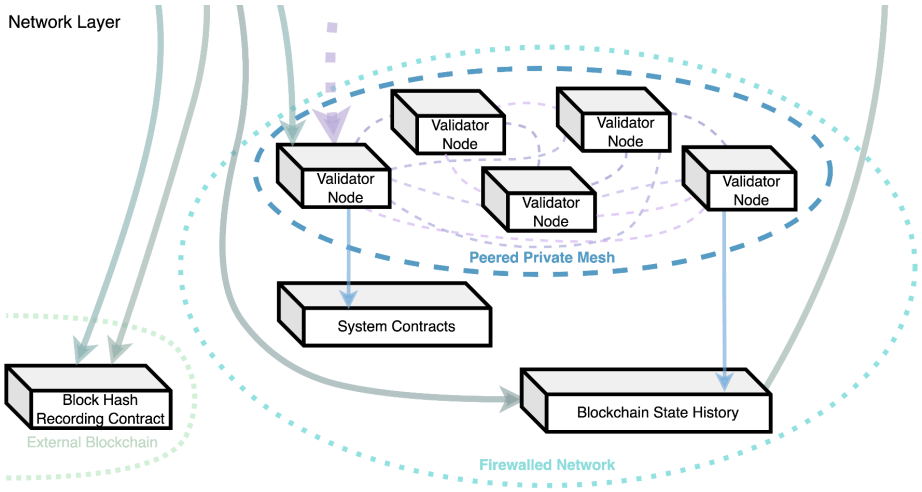


Fig. 4 The EASE Network Layer

Validator Nodes

Validator nodes each run a full copy of the blockchain, maintaining the state of all accounts with each validator node operating on high performance, homogeneous hardware and software ensuring no conflicts. Using the base Antelope.IO Leap 5.x Delegated Proof of Stake (DPoS) consensus model, consensus is achieved when one more than two-thirds of the active validator nodes agree on a valid block. A valid block is one that conforms to operational rules built into the Antelope.IO Leap 5.x code and the core System Contracts (e.g. eosio, eosio.token, etc.) as configured for the specific network. Since all validator nodes are running the same hardware and software, there should be few instances of discord. These nodes can be housed in different regions or countries, or all in the same room, depending on the needs of the system. Geographic separation of the nodes is more secure, of course, in terms of losing validator nodes to local interruptions.

Validator nodes are peered together in a private mesh network using the open-source WireGuard VPN tunneling protocol with no direct connection to the Internet. They communicate with the outside world using seed nodes, with each validator node linked to a seed node that is connected to the Internet and is able relay information to and from the blockchain validator nodes via API calls (at the API Layer). This limits the surface area for potential blockchain consensus attacks.

EASE System Contracts

The System Contracts for any EASE blockchain contain the programming for how to address any action that may occur on the blockchain. These contracts exist on a variety of contract accounts, most using ‘eosio’ as part of their name. Their function is to ensure that consistent consensus rules are followed by all validator nodes.

EASE Blockchain Mainnet

The mainnet of any EASE-based blockchain is distributed across multiple validator nodes, history nodes, API nodes, with each containing a full copy of all blockchain transactions. This ensures the security of the blockchain transactions and history. Individual blockchain communities may opt to run a testnet as well, but these are generally not needed or beneficial since the EASE Protocol developers operate testnets for any development and testing needs.

Block Hash Recording Contract

A smart contract that exists on a *different* public blockchain for recording a block number and a SHA256 hash of the block's entire contents (a.k.a a block hash) provides security against the remote possibility that blockchain operators would change the historical block records. This simple contract could run on any blockchain, but ideally will run on another Antelope.IO blockchain due to low transaction cost and similar or identical block times. A Block Hash Recorded program on the Node layer provides the data and triggers the blockchain action.

Node Layer

EASE nodes operate key systems beyond the blockchain itself including a number of auxiliary databases for privacy, speed and efficiency, and the Sequestered Private Key Management System servers which hold all system private keys in an enclaved, inaccessible environment and handle signing operations. These all reside inside the same firewalled system as the Network Layer servers. Outside of this firewall is an independent system monitor to alert the system any time the Network and Node Layer servers are inaccessible.

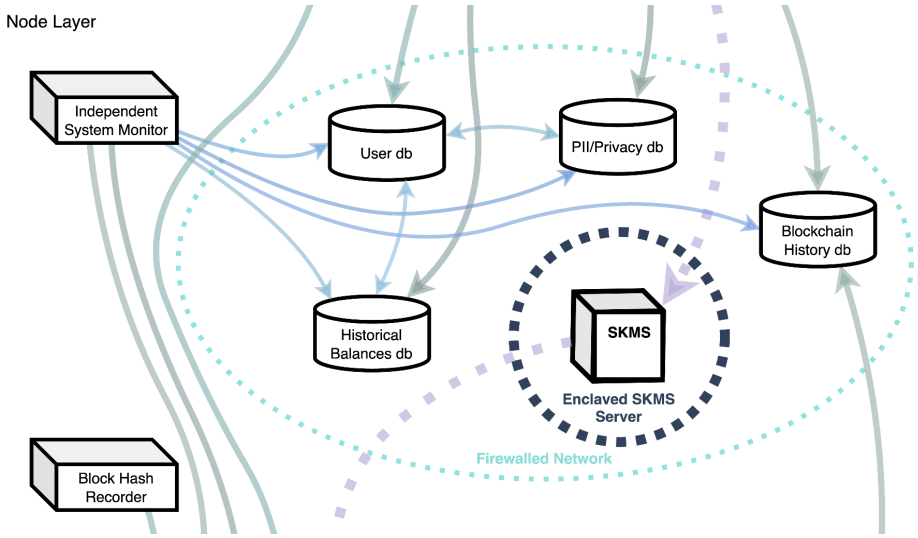


Fig 5. The EASE Node Layer

User Database

This database of users holds various details about access, preferences, and other necessary information about each user account.

Blockchain History Database

This database continuously scans the blocks verified in the system, transcribes the information, and records it into a large database format that can be queried much more quickly and powerfully than the blockchain record itself. This is a common occurrence in blockchain systems where the blockchain is a source of immutable data but not easily retrievable data and a secondary system is used to increase query abilities and performance.

Historical Balances Database

When data is ingested for the blockchain history database, certain data is also fed to a database that records only account token balance information for whatever tokens have been set to be tracked. This allows any account to instantly recover their balance amounts of any tracked tokens at any block height or date and time. Users can execute searches in the Historical Balances Database, for their own accounts, and return data about their balances of any tracked token at any

time. Despite blockchains being ledgers, this type of service is practically unknown on any L1 blockchain.

Mutable PII/Privacy Database

This database follows a similar operational model as the others, but it deals with personally identifying information (PII) that is afforded further protection and isolation due to it containing information of this nature. This information is not recorded on the blockchain where it would be immutable but instead, stored in a distinct database and indexed to the owner's blockchain account as supplemental data storage. Each EASE community can decide what data should be recorded on or off the blockchain, but there will be a set of recommended practices.

SKMS

The Sequestered Key Management System (SKMS) controls all signing operations. It is entirely distinct from the rest of the system servers by the nature of its task: to generate, and securely retain in encrypted form, and momentarily decrypt accounts' private keys when directed to by their owners. The SKMS can be instructed to generate new private keys for an account as a way to halt or change control, when directed to do so by appropriate judicial authorities. It is believed that without this ability, an EASE system is unlikely to be deemed acceptable or legally compliant in most jurisdictions. However, implementation of the key changing feature is configurable by any EASE community that wants to disable it.

Independent System Monitor

To constantly monitor the operation and accessibility of the system to users, one or more system monitors operate outside of the firewalled network where the majority of the system nodes are located. The system can also be monitored from within the firewall, but that arrangement does not alert the system when there are problems with data crossing the firewall, itself. The Independent System Monitor has no impact on the system operations or access to data. It simply reads the status of each system it monitors and reports when conditions are outside norms.

Block Hash Recorder

This node reads each new block created by the validators and sends its block number and block hash to the Block Hash Recording Contract (Network Layer) that exists on a separate blockchain, which adds immutable evidence that each block has not been altered from its original state. (If the block were changed in any way, the block hash would not match what had been recorded on the blockchain contemporaneously.) Thus, any block can be proven to be unaltered for as long as the public blockchain it's written to remains available.

API Layer

This layer acts as a connector between the various smart contracts and related infrastructure on the Tools Layer and the databases and SKMS on the Node Layer.

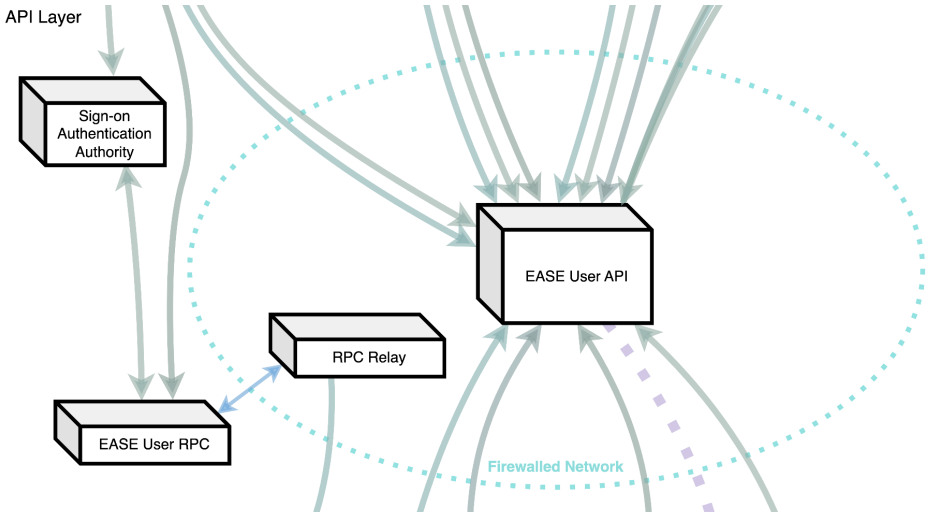


Fig 6. EASE Protocol technical stack API Layer

User API

The primary element of the API Layer is the EASE User API. It acts as the central router and clearinghouse for actions on the blockchain. Users' actions and data requests are received from interfaces (Interface Layer) such as the Super App directly, or indirectly by one of the various smart contracts (Tools Layer) which an interface interacts with. API calls are organized as endpoints that can be reached via HTTP commands or program commands from the EASE SDK.

Actions and data requests are addressed by the User API by consulting the various database nodes (Node Layer) and returning requested data to be formatted by the Super App or other interface (Interface Layer). All data returned by the User API has been filtered by the privacy controls to ensure that data is only presented to those who are allowed to receive it (typically the account owner). All filtering is handled at the API Layer prior to sending.

The User API possesses a protected communications tunnel with the SKMS and is, in fact, the only component of the system that can send commands to it. When performing an action for a signed-in user, the User API retrieves the decryption key token from the user and passes it to the SKMS along with the proposed blockchain transaction to be signed. Within the SKMS the decryption key then decrypts the account's private key, which is stored in encrypted form. The decryption only occurs while the SKMS is signing the proposed transaction and the private key is not revealed or accessible in any other way. The signed transaction is then transmitted directly to the validator nodes for evaluation and execution.

The User API contains a command to reissue private keys for an account. This can be done in the highly unlikely event of compromised keys (they never leave the SKMS so it's difficult to conceive of how they would be compromised, however, this is a reasonable protection against unknown unknowns), or when compelled to do so by a valid court order from a relevant government jurisdiction (typically, this would mean the nation of citizenship or residence of the account owner). The execution of the reissue keys action causes a new private key to be generated within the SKMS and the permissions of the account to be modified with the corresponding public key. The SKMS stores the new private key internally, encrypted with a private key held by the owner.

Relay RPC System

The Relay RPC System is included as a method for account owners who are well steeped in the Antelope.IO system to continue to use the third-party signing wallets they are familiar with to perform limited allowed actions, or for existing Antelope.IO smart contracts to be deployed on an

EASE blockchain without code changes. (Configuration changes such as RCP server connections will of course be necessary as they would be on any new blockchain deployment.) The signer wallets favored by Antelope.IO users are dated compared to the EASE interfaces and it's expected that this offering will not be needed or used by most EASE communities.

The relay system presents a typical RPC interface existing outside the EASE system's firewall. Without this, wallets would be prevented from accessing the information inside the firewall. A tunnel between the EASE Public User RPC server outside the firewall and the Relay RPC within it, is strictly limited to passing RPC requests and filtered responses.

The Relay RPC has the ability to query the Blockchain History Database (Node Layer) to respond to data requests, such as account balances and transactions. Of course, these are all filtered by the same rules that apply to requests from the User API.

The Relay RPC can send signed blockchain transactions to a blockchain validator node for evaluation and execution. These transactions are highly limited compared to the standard Antelope.IO RPC calls. They are largely limited to actions like transferring tokens and staking, unstaking liquidity tokens and claiming staking rewards. As noted, this system is a concession to legacy users that few EASE blockchain ecosystems are expected to use.

Sign-On Authentication Authority

A Sign-On Authentication Authority is any provider of Single Sign-On service that is enabled on each particular EASE blockchain system. Examples include Google, Facebook, X/Twitter, Github, Apple, etc. Each authority deploys and maintains their own infrastructure and connection details and most also have setup requirements before they can be enabled. This must be performed by the community leaders of each specific EASE Protocol blockchain.

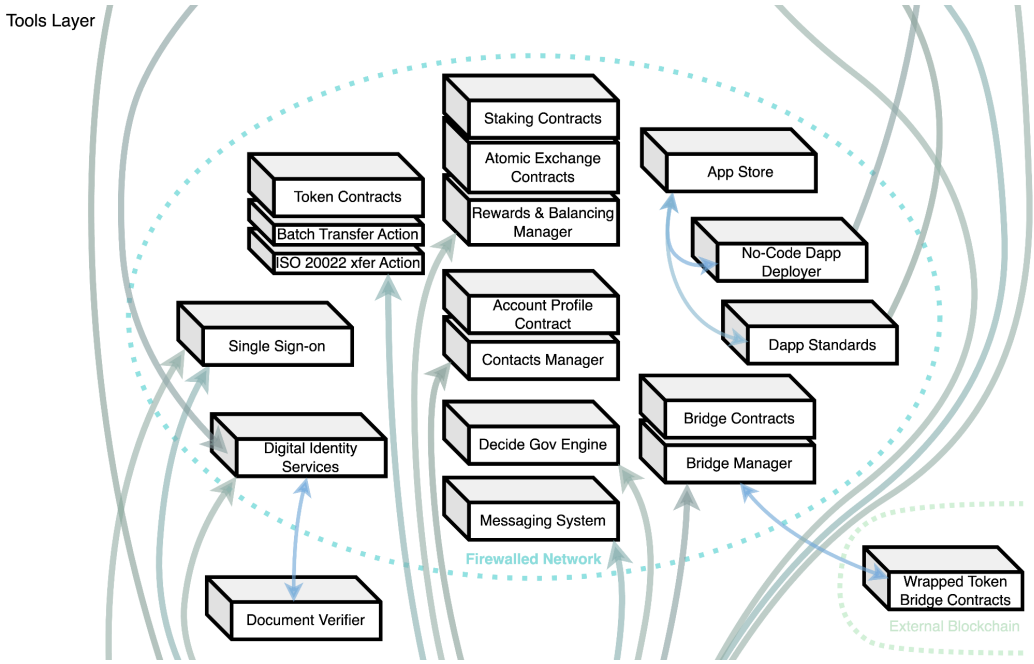


Fig 7. EASE Tools Layer

Tools Layer

The Tools Layer consists of the various programs used to take advantage of the blockchain system and associated nodes. These include any smart contracts (beyond the system contract) as well as programs running ancillary systems to support these

Token Contracts

EASE introduces new core transfer actions to the Leap 5.x token contracts: `batchtransfer()` and `isotransfer()`.

Batch Transfer Action

This new action provides a simple method for aggregating multiple token transfers into a single action. This was originally developed to support the Liquidity & Staking System's need to make large numbers of transactions as efficiently as possible, but it is available to other systems and users as well. The performance increase over multiple single `transfer()` actions is on the order of 20 to 500 times more efficient. It's possible to create non-breakable transfers without notifying the recipient.

ISO(-20022) Transfer Action

To support the ISO 20022 banking standard, a new transfer action is needed that can not only record the ISO 20022 XML data, but perform compliant smart contract functions such as halting or payment return transactions.

Single Sign-On

The Single Sign-On tools consist of an account registry with one or more login credentials for each account. These can be social sign-in services like Google, Facebook, X/Twitter, Apple, or other methods such as a hardware key or MetaMask crypto wallet/signer program. Accounts can have more than one verified sign-in method and they can manage these sign-ins. Once verified, a JSON Web Token (JWT) is transmitted to the user's device which allows continued access until it expires or the user signs out from their account. While the Super App is the primary EASE interface, other interfaces compliant with the EASE SDK will be able to use the JWT to verify identity without requiring their own sing-in operations.

Messaging System

The EASE Messaging System connects users with one another. It is intended that the primary function of messaging will be to discuss financial transactions, but the value of integrating social interactions into a Super App cannot be discounted. The system tracks message sending and reading to keep users up to date on the status of each individual message. Text-based message objects such as emojis and locations (as map coordinates), plus internal links such as transfer requests or no-code contracts can be included in messages. It will also be possible to include other objects such as data files, voice messages, and image files on systems where this will be enabled. Because these create potential gluts of data for the system to store and retrieve, EASE ecosystem creators will each have to decide what can and cannot be stored and therefore, included in the Messaging System. For example, a creator team may decide not to allow large files due to storage concerns, but to allow shrunken and/or compressed low-data versions to facilitate the value of the messaging system. A different creator team may decide to configure large data and file storage by including a fee-based system to pay for resources and prevent abuse.

The planned Messaging System will utilize end-to-end encryption for direct messages that even system administrators will not be able to read, although that feature is not yet enabled at this time.

Liquidity and Exchange System

The Staking Contracts, Atomic Intermediate Exchange Contracts, and Rewards & Balancing Manager comprise the Liquidity and Exchange System–one of the core systems of the EASE ecosystem. The chart shows the flow of operations controlled by each smart contract and manager program in the system. The individual components are described below.

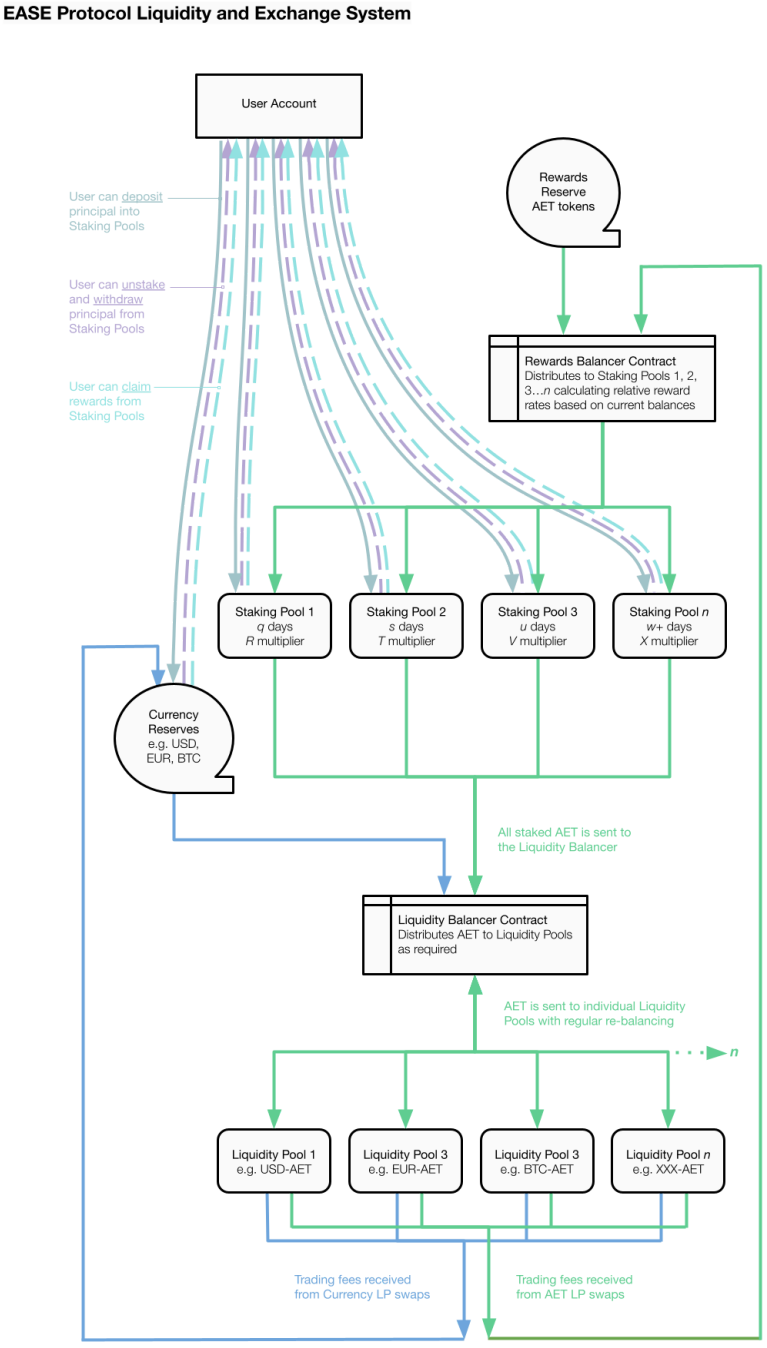


Fig. 8. Diagram of the EASE Liquidity & Exchange system showing components and flow.

User Account

Any user account can contribute their AET to any combination of rewards pools using the `stake()` action. They can initiate the commitment period for unstaking AET using the `unstake()` action or remove tokens immediately, minus a pool-dependent penalty using the `fastunstake()` action. Once unstaked tokens reach the end of their commitment

period the account owner may withdraw them using the `withdraw()` action as they will no longer earn staking rewards. (EASE systems can optionally be configured to automatically withdraw tokens from staking to the owner's liquid funds at the end of the unstaking period.) Finally, the account owner can use the `claim()` action to receive their accrued staking rewards once per Calculation Period. The Calculation Period can be configured to any number but the default is once per day using midnight UTC as the day end. Removing earned rewards by claiming does not affect the principal amount staked into the various staking pools.

Rewards Reserve

Every EASE ecosystem can have a portion of its initial AET token supply allocated to the Rewards Reserve pool to supplement the exchange fees revenue as a way to kickstart fees for liquidity providers during the early periods of the ecosystem when exchange fees alone may not appropriately incentivize liquidity providers. The particulars of amount in the Rewards Reserve and the formula for how they are injected into the system and over what period of time are configurable for any EASE blockchain based on its specific tokenomics.

As tokens are released from the Rewards Reserve, they are transferred to the Rewards Balancer Contract where they join liquidity fees earned from exchange operations.

Staking Pools

A number of AET liquidity pools can be created and each configured with its own minimum commitment time (the number of days required to unstake tokens to enable withdrawal), rewards multiplier, and fast-unstaking penalty. For example, a system could have two pools, the first pool with a ten-day unstaking period, a 1X multiplier on the relative rewards rate, and a 3% penalty for fast withdraw unstaking, the second pool with a 30-day unstaking period, a 2X multiplier, and a 6% penalty for fast withdrawal.

The fast withdraw unstaking penalty is deducted by the system only for the amount of tokens that are unstaked. The penalty is contributed to the pool as staking income and is earned by all the people staked in the system at that time who did **not** use fast withdraw unstaking, as an additional reward for remaining invested in the system. Essentially, this feature is designed to have liquidity providers who do not stick with their commitments to pay their penalty to those who do.

Staking Pools are funded by users providing liquidity via `transfer()` actions and by receiving exchange operations fees. Funds are removed either by withdrawing staked tokens (principal) at the end of their unstaking commitment period, or by claiming earned rewards. Tokens also exit the Staking Pools by being sent regularly to the Liquidity Balancer Contract, which, in turn, distributes AET tokens to the various token asset pairs (e.g. BTC-AET, USD-AET). These are regularly rebalanced by the Staking Manager program to maintain optimal liquidity.

Currency Reserve

The Currency Reserve is a token pool holding aggregate tokens of each asset that is contributed by users for liquidity, excluding the system's AET. These provide the other side of the AET token pairs and earn rewards for the exchange operations where they are used. Staking rewards are calculated for each token asset based on a ratio of how many tokens are staked as liquidity and how much exchange activity that token generates. When there is a shortage of liquidity relative to the amount of transactions, LP income will rise, attracting new liquidity providers in that asset. Assets in the Currency Reserve also flow

into the Liquidity Balancer Contract where it is paired with staked AET as needed by each liquidity pool.

Currency Trading Contract

The Liquidity balancing functions occur within the Currency Trading Contract using the `rebalance()` action, which combines AET and staked assets from the Currency Reserve to fund each of the liquidity pools, optimizing ratios of AET as needed to optimize system liquidity.

Liquidity Pools

A single liquidity pool exists for each token in the system, paired with AET to provide stable liquidity operations. As exchange operations occur, currency balances change within the contracts, adding additional trading incentive to users. These funds later flow into the Trading Contract for reallocation to unbalanced pools. Exchange fees flow into the Rewards Load Balancer Contract. Once per Calculation Period, these fees are distributed to Liquidity Pools for eventual claiming by liquidity provider accounts.

Rewards Balancer Contract

AET from the Rewards Pool (if any) are evaluated and rebalanced at least once per hour. Funds flow into the staking pools and back to owners through this contract. The Rewards Load Balancer aggregates tokens from Currency Trading Contract and Reward Reserve Contract and divide them between the various liquidity pools based on the amount of tokens staked and the Rewards Multiplier of each pool (e.g. 1x, 2x, 3x, 4x).

Intent of the Token-Staking Model

Token staking serves multiple purposes which must be carefully considered when designing a successful token-staking model for an EASE community. The primary goal may be to ensure there is an ongoing supply of the internal liquidity needed to operate the system efficiently. Other goals may include providing a token supply for providing an initial economic value locked within the community or to provide the initial funding needed to create the community in the first place. Once users have invested time and resources into the system, another intent for the AET-staking system, may be to provide valuation stability for the token by disincentivizing panicked selling in down-markets.

There is an ecosystem benefit if staked tokens remain staked in predictable patterns to increase stability of staking pools: volatility is reduced, token trading becomes more predictable and therefore there are fewer reasons to try to time markets. Most token-staking systems have a single set of parameters for how long tokens must be illiquid, when that period begins, and the rewards earned by the token owner for contributing their tokens to the liquidity pool. EASE offers a better approach by allowing the creation of multiple staking pools with different commitment requirements, so that owners who are willing to commit to longer periods of token illiquidity (“commitment periods”) can earn a higher reward. This allows token holders to create token commitment profiles that match their own financial needs. A portion of their allocation can be in pools with long commitment periods to maximize returns, while other allocations are in short- commitment pools to allow for faster unstaking when needed.

Whatever the commitment period, the clock does not start until the owner calls the `unstake()` action. Once the unstake action is called, the amount unstaked begins the countdown until it becomes liquid. During this time, it continues to earn staking rewards exactly as it did before until the commitment period expires, at which time the tokens are liquid but no longer accrue rewards. At launch, Ayetu uses four pools with distinct unstaking commitment periods and associated rewards multipliers.

This system applies to both AET tokens and counter-assets, however, only AET tokens will earn additional funds from the Rewards Reserve (unless an EASE community opts to configure their system differently). Every token is staked as a single token as opposed to a pair. The system itself balances the various token pools to eliminate impermanent loss or other risks associated with liquidity pair staking.

Distributing Rewards

The value of staked tokens does not compound. No matter how long a user has their tokens staked, the staked balance will never increase automatically. Instead, each day, the account earns rewards, which are calculated as *pro rata* share of all exchange fees and Rewards Reserve allocations that day multiplied by the Rewards Multiplier for each bucket. For clarity, the Rewards Multipliers are calculated for buckets before *pro rata* shares. The equation is:

$$StakingPoolRewardsShare_n = \frac{1}{(\sum_{x=1}^{p} StakingPoolMultiple_n)} \times StakingPoolMultiple_n$$

Where:

$StakingPoolRewardsShare_n$ - the share of rewards applied to Staking Pool n
$StakingPoolMultiple_n$ - the rewards multiple assigned to any Staking Pool n (e.g. 4x)
p - the total number of staking pools
n - the number of any given Staking Pool to be calculated

Claiming Rewards

Rewards can be claimed once per day and they will accrue for each day not claimed. When claiming, all available rewards are transferred to the user's liquid balance. Rewards are all paid in AET tokens whether for staked AET or other tokens. Tokens that have low staking relative to demand may receive higher rewards.

There's no requirement to claim rewards daily, however, users who wish to convert the system's simple interest-like mechanism into one closer to compound interest can add their claimed rewards to any staking bucket as a way to maximize returns.

Bridge System

To facilitate the most secure form of value transfer of tokens native to EASE systems, bridges will connect to other chains. Initially these will be EASE-EVM bridges and later EASE-EASE bridges with additional functionality such as conveying ISO 20022 transactions. Other blockchain systems will likely be added in the future, such as an EASE-Cosmos bridge. The components of the bridges are the smart contracts on each blockchain being bridged and a manager program that monitors these and facilitates the off-chain actions required.

Bridge Smart Contracts

One smart contract is required on each side of any bridge. The controlling contract is on the EASE system. It holds tokens that have been wrapped on another blockchain and then distributes them to the destination EASE account when they are returned by burning. This contract manages all tokens registered on the bridge. Registration is not automatic when creating a new EASE blockchain-based value token due to the need to deploy smart contracts on corresponding blockchains

On the EVM side, each wrapped token from the given EASE blockchain requires its own ERC-20 standard-compliant smart contract, to which the `bridgeburn()` action is added. The BridgeBurn action allows a number of the wrapped tokens to be burned on the bridged chain along with a destination EASE account. The smart contract also manages the paying of any bridge transfer fee, which might be incorporated into to pay for gas needed on the other blockchain, or as a revenue-generating feature, depending on each EASE community's needs.

Bridge Manager

The Bridge Manager listens across multiple blockchains for the specific contract address associated with each wrapped token deployed, in order to find BridgeBurn actions. Doing so initiates periodic checks (every few seconds) of the proposed transaction to verify that it has, indeed, been transferred successfully, and that any necessary finality period has passed. Once this has occurred, the EASE-side smart contract calculates any fee due using on-chain price oracles, deducts the fee, and transfers the balance to the destination account.

When tokens are transferred from the EASE blockchain to an EVM, the verifications all occur within the native smart contract and the Bridge Manager only mints the wrapped tokens in the destination address and verifies that the action was successful and reached finality.

Digital Object Notarizing/Verifying Service

The Digital Object Notarizing Service accepts any file (within configurable file size parameters) along with information about the block number at which it was sent and the sender account. This information is hashed using a SHA256 hashing algorithm. A record of this is also kept in the off-chain database for fast recall. The original file is not retained. Later, the owner may verify that a submitted digital object file is exactly the same as the one recorded on the blockchain by once again creating the deterministic SHA256 hash of the file or object and comparing it with the hash value written on the original block. This can be independently verified outside of the EASE ecosystem using any file hashing program. The Super App will include interface screens for both of these operations.

Account Profile and Contacts

Account creation is accomplished by a specially-permissioned account-creator smart contract and off-chain servers that interact with the necessary API infrastructure of each approved method of creation. Each EASE system will have the ability to configure their process from which verification services are allowed to what information is retained. One account may have multiple sign-in methods. Any account name selection, identity verification, or payment rules that the EASE community requires or allows can be configured on this server.

The account profile server is also a combination of an on-chain smart contract called `profiles()` which is permissioned for open reading by any user (accounts are private by default) in order to deliver on-chain personalization without revealing PII. Accounts have an initial profile avatar

based on a custom-created set of features, creating a safe profile not based on a photograph. The initial configuration is derived programmatically from the account name, but users can customize any of the image elements to create a version they like. Multiple avatar-creation sets can be configured for any EASE blockchain.

Personalization also allows users to set their own commonly used name, which can be edited by default (unless configured not to be editable). These names can also be required to be unique on the system at the time of creation. Users also have a status which can be expressed in any way the EASE community creators choose. On the Ayetu Blockchain, for example, the Status parameter is configured to only show one to three emojis as a non-text way to describe one's status or mood. There is also a text "bio" field that can be edited. The interfaces render Markdown format text (.md) and include a simple Markdown and emoji editor interface. This allows links to be included in contract bios. One exciting use of this feature already in use on Ayetu is letting system contracts describe their workings to users, add any important notes, and include links to their documentation right from their profile screen. This greatly improves user understanding in the (generally rare on EASE) occasions when they may interact directly with a smart contract outside of the Super App.

Code Deployment System

Deploying smart contracts on EASE blockchains is generally very easy. The components involved are the no-code smart contracts and deployer wizard, the EASE SDK and standards documentation, and the App Store that will be integrated into the Super App once enough apps are available on the system.

No-Code Smart Contract Deployer Wizard

Compiled smart contract executable files and their descriptor file (wasm and ABIs) are available as tested, audited code to deploy to any user account. These are organized into families of similar functionality and different levels of complexity so that users can run through the deployer wizard without having to know in advance which level of complexity will be required. The wizard consists of a simple decision tree with inputs to be used in the initialization of each smart contract immediately following deployment. The interfaces for deploying and utilizing no-code smart contracts will generally be built into the Super App.

Dapp Standards/SDK

For developers who wish to deploy more complex apps than the no-code system is meant to support, there is a set of features documentation including all API calls. The SDK contains pre-built components in different development frameworks that developers can directly incorporate into their app interfaces. There is also a best practices guide and prescribed names for certain types of actions used in the ABI to allow for a consistent interface. These correspond to industry best practices and are required for any app that wishes to be allowed in the App Stores.

EASE App Store

Each EASE blockchain ecosystem will have its own App Store where standalone apps using the EASE blockchain or additional modules to the Super App can be easily discovered and acquired. Apps will need to comply with best practices and naming conventions from the EASE Standards and SDK documentation in order to appear in the App Store and will need to comply with further, more stringent requirements if they are intended to be incorporated into the Super App as downloadable modules.

No-Code Governance System

In 2019, GoodBlock created what is still the world's most advanced digital governance platform, the Decide Governance Engine (DGE). It has been operating in a live environment on Telos as "Telos Decide" since that time with countless decentralized elections, referenda, community proposals, and by-laws amendments in that time. Since GoodBlock elected to make DGE free and open source software, other blockchain projects have also adopted it as the basis of their governance systems. In 2021, GoodBlock released the Decide Voter app as a proprietary full-featured interface for enabling all the various tools available through DGE. These tools will be integrated into Ayetu to allow communities to add a wide variety of voting tools and methods to empower internal governance.

The Decide Governance Engine works as a no-code, configurable, modular system for managing the full lifecycle of a ballot, from creation to dissemination to voting and finally to vote-counting and, in upcoming versions, to results contesting and review. A ballot can be tied to a number of potential deterministic outcomes, such as the gaining of a position that comes with certain powers (budget direction, council voting), the approval of a community proposal and distribution of funds to fulfill it, or the passage of a revision in the by-laws of a group that is then automatically enacted. These outcomes are enshrined in smart contract code to occur following a given outcome, without people in the middle needing to activate any action. For example, if an election is configured to give the winner the ability to vote a seat on a board or council, the winner of the election will receive that without any other person being in a position to block or delay the handover. If the passage of a new by-law clause is approved, the revised text will be inserted directly into the by-laws, without requiring further approval by anyone (unless such approval is configured as a step in the group's governance process, naturally). If funds are approved for a proposal, a successful vote will transfer them to the holding account for the proposers to access via whatever performance milestones are set up for that project. All of these parameters are configurable with a high level of granular control.

This engine is available to use for all kinds of groups on Ayetu, and groups using the system can do so in a no-code manner similar to how no-code smart contracts deployment works. Creating governance groups will be easy, fast, and affordable. (Like no-code smart contract deployment, Decide governance features will be at additional costs due to their consumption of network resources and to offset initial integration costs.) The primary interface will be the Ayetu Super App or a dedicated governance app like Decide Voter. DGE offers three kinds of ballot processes: Elect, Works, and Amend:

Decide Elect

Elections are managed by the Decide Elect module of DGE. These can be single seat elections where each seat can have multiple candidates, or "leaderboard elections" where all candidates vie for more than one seat and the top *n* vote-receivers win the available seats. Like everything in DGE, thresholds for minimum voting participation or quorum and for requirements for supermajorities are available in the configuration of each election. Voted seats can be configured for a specific term, and term-limits can be enacted, as can the circumstances required to trigger off-schedule elections.

Decide Works

Funding initiatives are managed by DGE's Decide Works module. Any group with a treasury and membership can configure rules for access to community funds. Typically, the steps involved are: submitting a proposal and funding request, determining the amount that will be allocated if the proposal is approved and where the money will flow. For example, funds could flow directly to the proposer for smaller projects, or to an account governed by an elected or appointed oversight

board for disbursing funds as the board deems milestones to have been met. In some cases, deterministic on-chain actions could unlock funds without human intervention. For example, a fund could become unlocked automatically when a certain amount of matching funds have been contributed.

Decide Works allows the voting period, voting type, amount of funds and post-approval management, minimum voting quorum and majority or supermajority threshold all to be configured at the outset of the ballot. Each ballot's parameters are clearly available to any member to review.

Decide Amend

Every organization needs some form of by-laws, constitution, charter, or member agreement, by whatever name these fundamental governance documents may be called. These are the commonly held and understood rules that tell every group member what can and cannot be done. Every viable organization needs the ability to evolve their by-laws with changes in the culture and law, and with the ever-changing makeup of their membership. Amendment is invariably a complicated process fraught with disagreements and frustration. In nearly every group, there will always be those who see the need for change and those who see the need for continuity with the existing rules.

The Decide Amend module of DGE is a unique tool for bringing clarity to decentralized amendment of by-laws (and other documents). Existing documents can be imported into the system and viewed by any member as a single source of truth. At the outset, groups will determine the conditions by which changes can be made. Typically, this will involve a long voting period, high quorum and significant supermajority. To propose an amendment, new clauses are proposed or changes to existing ones are offered (or a combination of these within one proposal). Each proposal can be voted on based on the group's voting configuration. Proposals that meet the passage conditions will replace the existing clauses with the new amendments, or add any new clauses to the official document. All changes are clearly documented and the official version of the by-laws will reflect the changes to anyone accessing them.

Interfaces Layer

The Interfaces Layer consists of all the ways that users can interact with the blockchain. These primarily interface with items on the Tools Layer. For example, the front end for any custom Dapp or No-Code Dapp deployed on the system, any modules of the Ayetu Super App, block explorer, external wallet, or digital identity services.

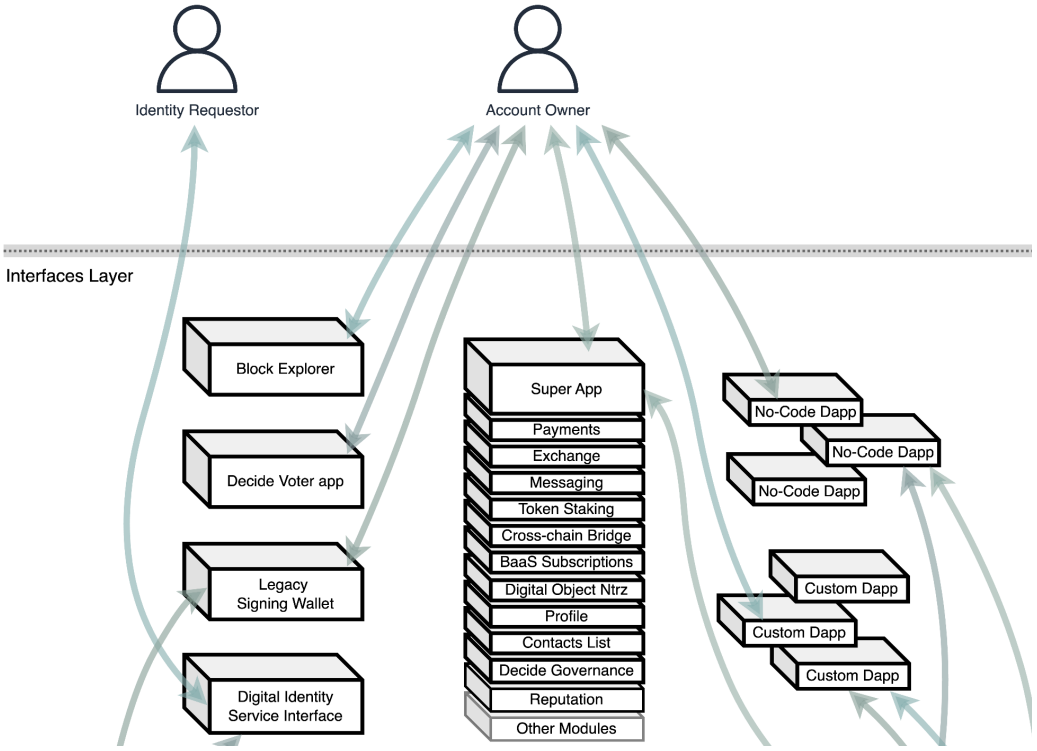


Fig 9. EASE Protocol Interfaces Layer

Super App

The Super App is the primary interface for all users. It encompasses interfaces for payments, token exchange, token staking, messaging, bridges, BaaS subscriptions, digital object notarization, profile and contacts management, reputation, no-code deployment and more. It can incorporate approved add-on modules from outside developers once they are approved.

No-Code Dapps

No-Code dapps utilize the EASE no-code smart contracts with their own custom interfaces built from the EASE SDK. They can exist as web sites or standalone apps on app stores like Apple or Android, if approved by the administrators for those app stores. These dapps can utilize the EASE single sign-on features for user security and convenience and will ensure protections offered by the precompiled smart contracts.

Custom Dapps

Custom dapps use both custom interfaces and smart contracts deployed on the EASE blockchain of their developers' choice. In general these must comply with certain elements of the standards and SDK in order to allow them to utilize the EASE single sign-on features for user security and convenience. However, these may be using code beyond the control of the EASE community creators.

Block Explorer

EASE includes a block explorer that utilizes the EASE sign-in system and privacy protections. It is run on dedicated servers that manage blockchain history and search. Only "public" accounts will be viewable to users who are not signed-in to their accounts. Account privacy is filtered internally within the Relay RPC, so no private data will reach the requester. Third party block explorers can be created, but none will be able to show anything beyond what is shown on the official block explorers.

External Wallets

Some third-party wallets for the AntelopeIO ecosystem are able to interact with the Relay RPC. These may manage a parallel private key for their account that is not used on the EASE blockchain's servers. Different EASE community creators can decide to configure how these are allowed access and how much to filter. The default level of filtration allows any kind of action that does not potentially break the privacy protections or manage user resources, which are not used on EASE blockchains.

Digital Identity Services

The Digital Identity Service requires its own whitepaper description, which will be delivered in due course. The intent of the system is to create the ability for users to prove their identity as needed for things like government interactions or employment, without needing to leak unneeded PII. Additionally, the system will allow a variety of independent document verifiers to participate in a verify-once mode.

Device Voter Governance App

When extensive Decide Governance tools are implemented in an EASE community, it may make sense to release a bespoke version of the Decide Voter app for that community, particularly if a large number of users are part of the community primarily to engage in voting features rather than financial features.

Building with EASE

For the first time, new communities of any size, can organize themselves and provide advanced computer services to their members without needing any hard to use tools and without needing to hand over most aspects of their community organization to the developers configuring and operating the computer network it runs on. EASE has removed all of the impediments of blockchain systems: the over-complicated interfaces, the so-secure-it's-actually-risky private key self-custody practices, and the eternal dependency on blockchain developers, who are never done tinkering with their favorite gee-whiz features (which are rarely the features that the communities are actually requesting).

EASE is easy to use, it's loaded with features that are needed yesterday, not *possibly* needed tomorrow, and it frees communities to manage their own systems and configurations on a BaaS payment basis like any other professional software. Users are empowered to manage their businesses, finances, accounting, payments, data management, currency exchange, no-code smart contracts, and advanced governance tools, to name just a few of EASE's features.

Communities looking to organize themselves into DAOs or similar decentralized organizations, can compare the feature set and ease of use and setup to any system claiming similar features. The differences will be stark.

The time has come for new and better ways to organize ourselves to tackle the pressing problems facing our civilization and threatening our future safety and prosperity.

Ayetu: the first EASE Community

The Ayetu community in Africa is the first to be built using the EASE Protocol tools. In fact, the EASE Protocol and the Ayetu Network are being built in tandem. Their community has provided generous support, needed feedback, and constant encouragement throughout the design and building of the software described in this paper. Because of that vibrant and growing community, EASE is able to be built to the demands of real users, which accelerates its useful development.

Looking ahead, over the next eight to sixteen months as all the described features are likely to be introduced, there will, no doubt, be many EASE communities created across the world to solve specific needs. Ultimately, these communities will all be able to interface with one another via EASE-to-EASE bridges. People will be able to finally receive the long-promised benefits of blockchain technology, perhaps without ever even thinking about it as blockchain technology. This is the dream we are working to build.

Ω

Glossary

ABI	(Application Blockchain Interface) A file that describes the actions and tables contained within a smart contract to make it easier to interact with it
AET	(Atomic Exchange Token) the core liquidity token used within an EASE community. AET is the default token symbol and name. Live EASE communities may use different names and symbols.
API	(Application Programming Interface) An interface designed to allow computer programs to easily interact with each other. An API can be used to request information from a program or to cause it to perform actions. APIs may be public or private/permissioned.
Atomic	Something that cannot be further divided. An atomic transaction is designed so that every action that is part of the transaction must occur, or none of the actions are allowed to be performed.
Atomic Intermediate Exchange	A form of token exchange that uses an intermediate token that is paired to each asset within the intended exchange instead of using a single transaction from a dedicated token pair. Where a traditional token pair would trade: *AAA* → *BBB*, a version of this exchange using an intermediate token would trade: *AAA* → *Int. Token*, *Int. Token* → *BBB*. Because the exchange is atomic, all actions in the transaction must execute for any of them to be recorded on the blockchain. The generic symbol for this is AET though it may have different names on each EASE community.
BaaS	(Blockchain as a Service) A blockchain operated as a for-fee service, paid to the operators in some form of payment. Typically these are permissioned and not fully public, but this is not a requirement.
Block Explorer	A human interface for viewing transactions and current balances on a blockchain.
CBDCs	(Central Bank Digital Currency) Digital currencies issued by a nation's central bank. Most often, this is envisioned as a token that is fungible with the country's fiat currency, although a country can issue other forms.
EVM	(Ethereum Virtual Machine) A distributed Turing-complete computer system operated by other validating nodes that can run smart contracts written in Solidity or compatible smart contract programming languages. The Ethereum Mainnet is the best known EVM, but many others exist.
Generation	In blockchain technology, fundamental paradigm changes are viewed as generations, with the first generation of blockchain consisting of Bitcoin and all the various cryptocurrencies that used the same basic features for recording value transfers using unspent transaction outputs (UTXOs). Generations typically inherit the useful features of their precedents.

	Second generation blockchains, such as Ethereum and its various clones add the ability to operate smart contracts. Third generation blockchains typically have account abstraction allowing for accounts that are not deterministically derived from their private keys (meaning an account's private keys can be changed). EASE Protocol represents the fourth generation of blockchain technology due to its sequestration of signing actions and greatly improved user adoption and ease-of-use.
Enclaved Space	A trusted execution environment where secure data can be used by programs such as a private key being used to sign a blockchain transaction. The space can exist on a physical chip or just a portion of, or "space" on the chip. A dedicated secure server can also be an enclaved space, a.k.a. A Secure Enclave
ISO 20022	A standard by the International Standards Organization describing a set of standardized messages about every type of banking action. It has been designed to replace the previous standard, known as SWIFT.
JSON	(Javascript Object Notation) A widely used open file format for recording computer data in human readable format.
JWT	(JSON Web Token) a web security token that includes encrypted information that is used to securely extend permissions to a computer user. A component used in secure protocols such as Single Sign-On. Pronounced "Jawt."
KYC/AML	(Know Your Customer/Anti-Money Laundering) Protocols for verifying the identity of a customer of a financial service provider to comply with legal requirements for ensuring they know the true identity of their customers as well as their status regarding international sanctions.
Layer One	(L1) A blockchain like Bitcoin or Ethereum that operate their own consensus protocol and have a unique genesis block. cf. L2 or Layer Two blockchains which exist solely to add missing functionality to a L1 blockchain. EASE Protocol blockchains are L1s.
Nonce	(Number used ONCE) A number added to information to be run through a hashing algorithm to create a unique hashing result. A nonce is incremented every time it is used by a specific blockchain address to prevent EVM transactions from being "replayed" or repeated.
PII	(Personally Identifying Information) People's private information that can be used to discover their identity and other details about their life. Many countries regulate how PII may be recorded and what disclosures and procedures are necessary, often including the right to have PII removed.
RPC	(Remote Procedure Call) A computer action that lets one computer trigger a procedure on a remote computer. Blockchain wallets often use RPCs to communicate with blockchains.
SDK	(Software Development Kit) A collection of development tools for a computer environment to ease the creation of new software.

Sequestered Encryption	An architecture that allows for signing private key actions using a remote key management system on an enclaved secure server. The server can generate private keys and never allow them to leave the sequestered space, the private keys are encrypted with a password that only the account owner controls, which is sent to the enclaved server to allow secure remote signing functions. Account owners have sole control over their accounts, except that system operators have the ability to cause a new secure key pair to be created.
Single Sign-On	(SSO) is a method of computer authentication that allows a user to log in with a single credential to any of several independent software systems.
SKMS	(Sequestered Key Management System) A system for remotely managing private and public keys using sequestered encryption.
Super App	A mobile application designed to perform numerous and varied tasks from a single master interface, typically offering features for messaging, social media, payments, and other modules. WeChat is a well known Chinese Super App.
The Greater Fool Theory	A theory for explaining why people will buy assets at seemingly "foolish" prices and yet sell them at a profit to another person, who is a "greater fool" than the seller (the "lesser fool"), who may go on to also sell the asset at a higher price to an even greater fool. These "fools" manage to make profits until the market collapses.
User/Password Credential	A common security credential for logging into computer systems with a user name and corresponding password. Email addresses are often used as the user name.
Web2	A name for the second generation of the World Wide Web, where user-generated content, social media, and responsive websites and apps increase engagement.
Web3	A term used in many different and often conflicting contexts. In general, it suggests whatever is the next iteration of the Web. Decentralization, data privacy, digital currencies and AI are various components of the fuzzy definition of Web3.
Wrapped Tokens	Tokens that represent tokens native to another blockchain but bridged so as to be tradeable on other blockchains, generally due to lower liquidity on their native blockchains. The letter "w" before the token's native name often denotes a wrapped token, but other letter prefixes are often used to designate the exact flavor of wrapped token. For example, "wBTC" is a wrapped token representing native BTC tokens held 1:1 in some other account on the Bitcoin Mainnet. The wBTC tokens are in an ERC-20 token on Ethereum. This makes it easy to trade BTC on Ethereum DeFi.
XML	(eXtensible Markup Language) A format for marking-up and storing data that is both human- and computer-readable. The ISO 20022 standard uses XML for recording its messages.