Trisk 2.5

Hybrid low-code toolkit for developers bridging Web2 and Web3

May 5, 2022

Abstract

Traditional business processes rely on manual actions from one or more parties involved in information collection, execution, communication, and verification/approval. The successful execution of these processes may involve automated workflows which are typically complicated in their nature and require case-specific knowledge from a technical development team to create/adjust/maintain the application for the business. Often such systems require integration with third-party services involved with a process in their respective stages (regulatory entities, CRM systems, etc.). Historically to launch and maintain the sustainability of such applications, the business must have both industry and technology expertise.

Data protection, proof of ownership, proof of action, security, and auditability are also mandatory. Unfortunately, traditional centralized applications are susceptible to incompetence, hackers, fraud, scams, etc.

Blockchains and Smart Contracts ("SC") could address all these issues. SCs are poised to revolutionize many industries by replacing the need for both traditional paper contracts and centrally automated digital contracts. They become a distributed proof of ownership and proof of state that can be easily verified and cannot be changed since its placement on the chain.

Blockchains on which SCs run cannot support native communication with external systems because of their underlying consensus protocols. Furthermore, there is a lack of understanding of blockchain technology and a struggle with the applicability as well as simplicity of use by traditional companies, even those with IT expertise.

Recognizing these obstacles, Trisk is evolving from a no-code software as a service that allows users to quickly build/automate/execute business processes to a hybrid no-code developer toolkit, taking into account and keeping the importance of the convenient web2 interface.

This paper presents Trisk's vision of Magniflow – an isolated, hardened, and highly constrained environment to operate security-critical data. A Studio Store – a digital environment that connects

creators, consumers, and developers. We describe existing architecture and componentry, and introduce a brand new Blockchain Manager that facilitates connectivity between web2 and web3 spaces. We also describe the Activity History – an audit component that helps businesses and developers achieve robust application security.

Finally, we showcase existing areas of use for creators in digital arts ("NFTs"), the real estate and escrow process, governance and compliance for DAOs, and nail down the roadmap of future deliverables.

Contents

1 Introduction

The traditional application source code and its logic (contracts) run and exist in a centralized manner that leaves it subject to alteration, termination, and even deletion by a privileged party. SC, on the other hand, is a self-verifying and self-executing "decentralized code" that guarantees the binding of all parties to an agreement as written. It provides a new powerful type of trust relationship that does not rely on trust in any one party. Nobody (even an SC creator) can alter the code or interfere with its execution.

The vast majority of SC applications rely on data about the real world that comes from key resources, specifically integrations and APIs, that are external to the blockchain. Because of the mechanics of the consensus mechanisms underpinning blockchains, a blockchain cannot directly fetch critical data. This introduces a new technical challenge: connectivity.

When we talk about connectivity, we also must stress the inability of SC to output data into off-chain systems. Such output often takes the form of a payment message routed to traditional centralized infrastructure in which users already have accounts, e.g., for bank payments, PayPal, and other payment networks. Trisk provides an ability to securely receive data back from the SC and deliver it to third-party systems.

Trisk introduces a solution that connects data sources from web2 applications to decentralized applications ("DAPPS"). It differs from other solutions by having a convenient and simple UI for business process creation, the granularity of accesses and permissions, and flexibility to integrate with all third-party data points. Trisk has a powerful conditional logic builder, similar to SC nature of if/then conditions, that allows configuring a set of actions based on when something happens with the data/process/chain or even events triggered from the SC execution.

As SC relies in many cases on external data sources, any decentralized application ("DAPP") must demonstrate ease of use and simplicity for the end-user. Trisk counts that as a rule from the existing web2 and brings a simple user interface of configuration, creation, and execution.

Blockchain transactions that are tied with SC executions still lack proper UI representation creating a struggle for DAPP users and even their administrators. Trisk intends to solve that issue with the existing toolkit consisting of activity history and notifications. Given the fact that it is easy to configure access for each step in a workflow, there is solid confidence as to who can view/execute/observe and be notified when something happens. This is crucial for DAO and governance issues.

As a result, making SCs externally aware, meaning capable of interacting with off-chain resources, is necessary if they are going to replace the digital contracts in use today, such as Docusign, PandaDoc, and others.

Examples of where the technology evolves include:

- NFTs smart contracts must inform the actual trademark or IP owner about the successful sale of a digital asset (e.g. art) that references their proof of ownership.
- Securities smart contracts such as bonds, interest rate derivatives, and many others will require access to APIs reporting market prices and market reference data, e.g. interest rates.
- Insurance smart contracts will need data feeds about IoT data related to the insurable event in question, e.g.: was the warehouse's magnetic door locked at the time of the breach, was the company's firewall online or did the flight you had insurance for arriving on time.
- Trade finance smart contracts will need GPS data about shipments, data from supply chain ERP systems, and customs data about the goods being shipped in order to confirm the fulfillment of contractual obligations.

2 Architecture Overview

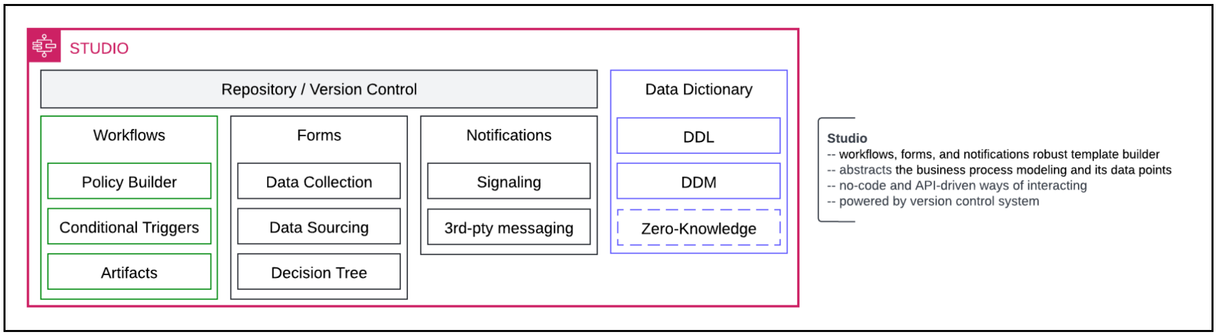
2.1 Basics

To eliminate the gap between web2 and web3 space and provide a robust environment for developers, the platform has been designed with modular microservice architecture. Below, we briefly explain each core component of the platform and its purpose. Keeping in mind the evolution speed in tech, each component is fully separated and modular so that we can achieve isolation, scalability, and the possibility of replacement as better and modern techniques are available.

Trisk has five general concepts in the core architecture - **Studio**, **Engine**, **Orchestration** and **Governance, API Gateway** along with **Artifacts** and other components.

2.1.1 Studio

Studio - workflows, forms, and notifications no-code template builder. Essentially, the user designs the business process and its data points with the power of conditional logic without writing any line of programming code. The configuration granularity provides tremendous flexibility and transparency for the business process modeling. Each step in a workflow may be configured against the responsible role, permissions to view/modify/submit the data, due dates, notifications, and artifact-specific attributes. Studio components utilize the absolute power of Trisk's versioning subsystem to track any changes that have been made to the template by any party involved and address the consistency or data immutability concerns. Once the workflow is designed and tested, the user may submit the version and lock (protect) the workflow from accidental changes.

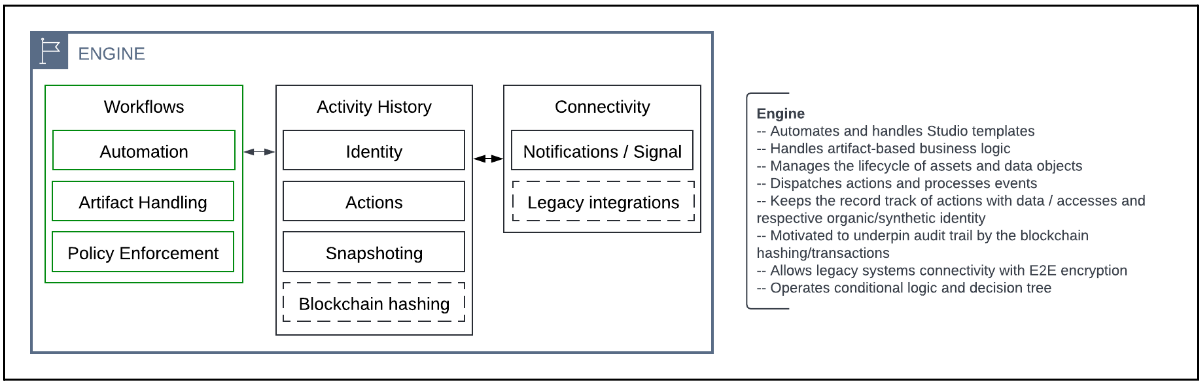


Studio
-- workflows, forms, and notifications robust template builder
-- abstracts the business process modeling and its data points
-- no-code and API-driven ways of interacting
-- powered by version control system

Figure 1: Illustration of Studio, its composition, and componentry.

2.1.2 Engine

Engine — a core component that operates Studio's assets. The launch process consists of the first required and essential part — defining the responsible parties against template-defined roles called assignees. This action involves a user with a specific set of permissions in the Tenant that are configurable and stand as a top-level access management component in Trisk. Once all the assignees are defined and users launch the workflow, the engine snapshots the template in its Instance representation. The engine ensures that the workflow Instance cannot be modified and reflects the same template submitted in Studio. The engine engages the assignee with Instance only when required and in alignment with the permissions from the workflow template. With a powerful engine component, Trisk provides such a high confidence level that each user-defined artifact and sensitive data are only available to the necessary parties and only when scheduled.



Engine
-- Automates and handles Studio templates
-- Handles artifact-based business logic
-- Manages the lifecycle of assets and data objects
-- Dispatches actions and processes events
-- Keeps the record track of actions with data / accesses and respective organic/synthetic identity
-- Motivated to underpin audit trail by the blockchain hashing/transactions
-- Allows legacy systems connectivity with E2E encryption
-- Operates conditional logic and decision tree

Figure 2: Engine and core modules – Workflows, Activity History, Connectivity

2.1.3 Orchestration and Governance

As all components are isolated from each other with regard to security and riskless replacement or enhancement, Orchestration connects them with solid security and governance utilities. The Orchestration module is responsible for intercommunication between Studio, Engine, API Gateway, Blockchain Manager, and Assets (Garage).

Authentication – unique identity verification by issuing and managing access tokens that are tailored to client/device hardware. Authentication is designed to log every inbound action from the outside and monitor suspicious activity, as well as managing active sessions, and supplying SSO capabilities.

Governance – a second and incredibly robust security layer. Each granular action within Trisk can be configured by the administrator of the platform in three levels of engagement: read, write, delete.

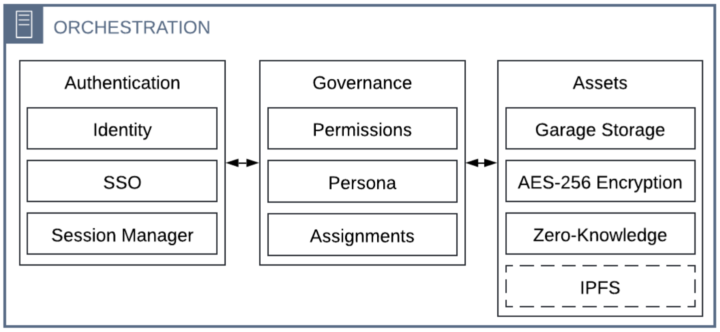
Governance module ensures that an authenticated client goes through a permissions check on each inbound request, internal components processing, and the crucial one – assets access.

Within permissions, Trisk applies Persona and Assignments concepts.

Persona essentially scopes the access to internal entities, pretty much like assigning a person to a client/account in a traditional business. Business owners are assured by the platform that Persona will filter all the information and unlock only the necessary data portion.

Assigments are the last but not least data protection layer. Workflow component allows defining assignments granularly for each Artifact by the known three criteria: read, write, delete. When the request passes Authentication and Persona check, Assignments verify actual access to the data point on Artifact level.

Assets – is a supervising data layer. Handles storage encryption, zero-knowledge, and access control. Our roadmap has IPFS-like storage support vision that addresses the single point of failure issue. Customers will have an opportunity to store their data in a distributed manner, mitigating risks related to traditional centralized applications – hackers, fraud, scams, etc.



Orchestration
-- Maintains the identity and connectivity of the platform members
-- Validates and enforces governance-related boundaries: permissions, accesses, persona
-- Handles microservices and components intercommunication

Figure 3: Orchestration definition

2.1.4 API Gateway

API Gateway provides secure, monitored, and modern external communication and connectivity access. Trisk uses API Gateway for its UI/UX purpose internally and unlocks it for external use: developers, SMBs, tech-savvy people, development shops, etc. API Gateway allows communication with almost all platform components without understanding the internals. Abstraction philosophy is crucial nowadays because flexibility pays off in a short period of time and becomes a long-term investment. Instead of installing software, spending time on training and onboarding, and understanding connections across components, developers are able to get a rapid start, with results in hours, just by utilizing API routes.

Inbound/outbound traffic is protected by E2E encryption, API Keys, and ACL layer. Along with protection, traffic is being monitored and logged.

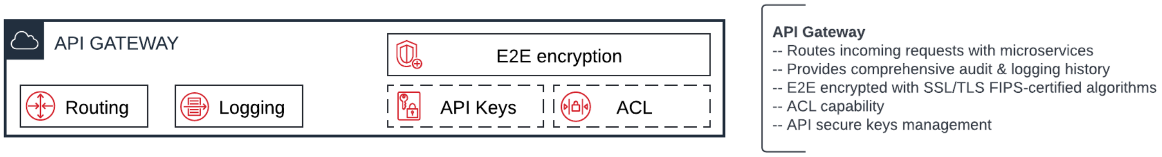

API GATEWAY
Routing
Logging
E2E encryption
API Keys
ACL
API Gateway
-- Routes incoming requests with microservices
-- Provides comprehensive audit & logging history
-- E2E encrypted with SSL/TLS FIPS-certified algorithms
-- ACL capability
-- API secure keys management

Figure 4: API Gateway

2.1.5 Artifacts philosophy

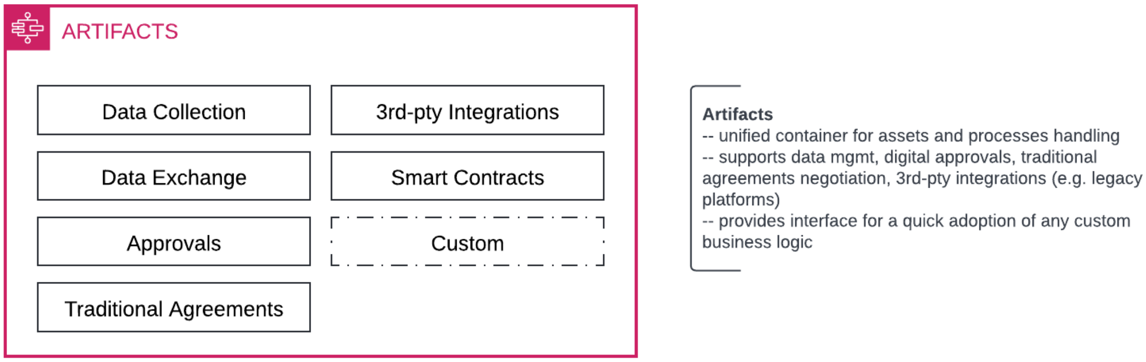
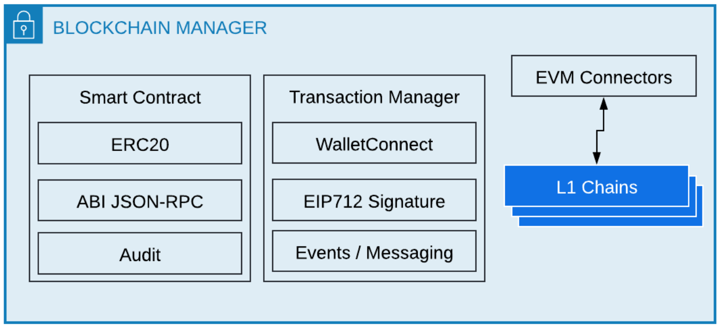

ARTIFACTS
Data Collection
3rd-pty Integrations
Data Exchange
Smart Contracts
Approvals
Custom
Traditional Agreements
Artifacts
-- unified container for assets and processes handling
-- supports data mgmt, digital approvals, traditional agreements negotiation, 3rd-pty integrations (e.g. legacy platforms)
-- provides interface for a quick adoption of any custom business logic

Figure 5: Supported artifacts

Considering the dynamic business environment and rapid technological evolution, Trisk applied the artifacts approach to its heart of Studio and Engine components. Artifacts abstract away implementation and integration burdens of the newer necessary business functionality and requirements. As artifact componentry is fully isolated, potential risks of regression in the rest of the platform are mitigated, stability increased, and maintenance costs reduced. As of now, Trisk supports all the common artifact implementations:

- **Data Collection**: Utilizing powerful custom Forms from the Studio component, Artifact handles all of the rules defined by the creator, providing a robust way of collecting information and controlling conditional logic.
- **Data Exchange**: Artifact can obtain and deliver data across multiple responsible parties, applications, and 3rd-pty integrations.
- **Approvals**: The major aspect of risk assessment and management is how information can be controlled and verified. With the approvals support, Artifact can engage respective assignees with collected data at the optimal time in a process. The actual verification result can be hashed and stored on the blockchain.
- **Traditional Agreement**s: Any common business use case can be facilitated and addressed via Artifact definition scheme.
- **3rd-pty Integrations**: Artifact components could embed and handle external systems/platforms connections. A sufficient flexibility level allows for the secure movement of information back and forth between Trisk and existing customer platforms.
- **Smart Contracts**: Trisk now supports EVM SCs and allows creators to establish connections between web2 and web3 world by mapping data in and out with convenient and simple UI. Each function and argument within a SC can be mapped with the respective data point in Trisk. Each preceding Artifact can securely supply the necessary information into a SC. Events triggered by the SC can also be observed and stored in protected off-chain storage for further processing.
- **Custom**: Trisk offers developers a layer in Artifact components that unlock a wide range of custom business implementations. Effectively, developers can create an external module that can then be adopted or embedded by the Artifact component. Having this robust connectivity, Trisk and developers are able to cover a comprehensive volume of use cases.

2.2 Blockchain Manager



Blockchain manager
-- Supplies L1 chains connectivity and communication
-- Interacts with explorers for Smart Contract engagement
-- Manages wallet connections and session
-- Executes and monitors Smart Contract invocations
-- EMV-compatible and vendor-agnostic

Figure 6: Definition and key concepts

To connect web2 and web3, Trisk defined and implemented a new isolated component that abstracts away the blockchain complexity. We call it Blockchain Manager. In its early stage, Blockchain Manager supports L1 chains such as Ethereum and Celo, EVM-compatible SCs (ERC20), connectivity with blockchain explorers such as Etherscan, and Wallet management with Walleconnect integration.

With the blockchain explorer abstraction, Trisk provides an application binary interface ("ABI") of the desired SC in an understandable way of representation. Typically, ABI files are not human-readable but necessary for a definition of which function in the contract to invoke, as well as to get a guarantee that the function will return data in the expected format.

From now on, any SC can be securely invoked with dynamically mapped data to its functions and arguments. The execution is furtherly observed by Blockchain Manager and reported back to the respective Artifact for processing/chaining.

Blockchain Manager exists as an isolated microservice protected by Governance, API Gateway, and Orchestration components mitigating the data compromising risk, unauthorized or permissionless access.

By having this component in place, we unlock access to and benefits of blockchain SCs to traditional businesses, developers, and tech-savvy people. We are lowering the barriers, reducing in-house development costs, reducing time to market, wiping out hiring the right and expensive talent, and simplifying training/onboarding. This will vastly accelerate development time, as teams can focus on their application's core business logic instead of struggling with connectivity issues and reporting.

Furthermore, as part of the roadmap described in the Future work section, we plan to decentralize the handling process by establishing the next level of security - hardware enclave and multi-party computation (“MPC”).

2.3 Integrations with legacy systems

In today's world, APIs are everywhere — they control each piece of the information from social media, GPS, analytics, data collection tools, CRM systems, logistics tools, etc. Those environments are integral to today’s businesses. By definition, Trisk is an API-driven environment that creates opportunities for various applications and external solutions. Keeping in mind our modular architecture, any component can be an isolated or compound feature that stands as a standalone value-add proposition. Furthermore, existing Trisk’s UI interfaces rely on API Gateway that proves ease of use and integration.

When we talk about API and integration, it is not always internal but a third-party. There is no escaping third-party integrations: every company has applications they use from vendors that need to read and write data. Integrating with third-party services is often complex and error-prone. There are many factors to consider when tailoring applications for a specific need.

Nowadays, third-party APIs are ubiquitous, and as a company grows and becomes more sophisticated, the demand to integrate with third-party applications will continue to scale.

Trisk stands here as a simplification and cost-saving solution to speed up the interaction process with such APIs. The Artifacts concept allows businesses to easily and quickly onboard a wide variety of integrations. The data securely flows back and forth within the Engine component, accompanying Activity History, and Garage components. Behind the scenes, Trisk handles rate limits, batching, session management, and handling errors, allowing businesses to focus on the real needs instead of getting buried with routine technology specifics.

Considering the increasing interest in blockchain technology, the complexity of being integrated with multiple applications, and the struggle with insufficient blockchain developer expertise, Trisk exclusively creates a simple automated data bridge between third-party applications and the benefits of the blockchain SCs. Any party can quickly gain control over information residing in an external application by bridging it to the fully web-like customizable process with underpinning blockchain benefits.

3 Security, audit, and data management

3.1 Activity History – a blockchain-based audit trail

Audit trails are information records that collect and represent important events, providing supporting documentation and valuable history for security and operational actions or mitigating challenges, including details on the date, time, and user information associated with the event. The **Activity History** component provides a baseline for audit or analysis when necessary (an error has been detected, risk management, etc.) In the web2 space, we must consider a lack of transparency in any change in the process and the data. Furthermore, the auditability in the web3 ecosphere requires Trisk to address traditional system weaknesses such as unauthorized alteration of data.

In the basics, the **Activity History** component tracks each user's or even the Engine's action in a workflow. The intelligence for the blockchain-based audit trail mechanism is intended to be implemented through SCs deployed on the blockchain network.

The use of SCs backboned by a blockchain network is taking advantage of the following inherent features of blockchains:

- **Integrity**: The information recorded on a blockchain cannot be manipulated. As the data is distributed across multiple nodes, it cannot be modified. In this way, fraud prevention and detection are almost unnecessary.
- **Authenticity**: Blockchains provide an unalterable assurance of origin; thus, nonrepudiation is guaranteed.
- **Long-term maintenance of audit information**: Once the information has been entered into the blockchain, it can never be deleted; consequently, the audit trail will always be traceable.

The **Activity History** component provides a user-friendly web-based interface to access the information recorded on the blockchain. In this way, users unfamiliar with the underlying technology will also be able to use the auditing system, enhancing the mass adoption of this solution.

3.2 Garage – distributed data storage

Conceptually, traditional centralized systems still act as a potential point of failure - breach, hack, hardware issue, loss of control. The recent cybersecurity statistics study (https://www.varonis.com/blog/cybersecurity-statistics) revealed that data breaches exposed 36 billion records in the first half of 2020. Businesses and individuals must invest in decentralized solutions to store and manage data. Blockchain technology allows private information to be shared securely while users remain in complete control of their data.

Trisk has established adequate precautions to protect data privacy and let individuals control how and when their data is shared. Each piece of user's data resides in the permission-driven Garage component that uses AES-256 encryption to securely store sensitive and non-sensitive information.

The Garage component takes the decentralization concept as the essential step for protecting user data and privacy. That requires the implementation of IPFS-like or SWARM-like distributed zero-knowledge data storage. The underlying architecture is strongly tied with **Instance,** which means that each step and its data are stored in a secure, encrypted, and distributed manner. Access to encrypted data doesn't sit with the businesses prone to breach, so the data is safe in the event of a breach. Only parties with signed consent can access sensitive data.

The Garage component is envisioned where all aspects of cloud storage, such as transport, processing, or data storage, are entered on the blockchain. Later, anyone who has access to the blockchain can verify what happens to the data. Such a system provides complete traceability, accountability, and transparency to the cloud. It enables the possibility to store data in a secure and decentralized manner. The decentralized aspect ensures there are no central servers to be compromised.

The fundamentals include the consideration of the following principles:

Resilience — Computation has to continue even if some number of failed nodes
Efficiency — The performance must be excellent for the end-user, even if the nodes involved are heterogeneous.
Performance — A distributed network must comply with linear performance
Security — Data protection, confidentiality, and information security must be adequately addressed.

Sensitive data exchanges between members are not suitable for sharing with all participants. The Garage component may use isolated peers subnets and support private ledgers in such an environment. Peers can only join the chain when approved by the organization on that chain, which leads us to consortium or permissioned blockchains that are best suited for business use. The right to read/submit transactions may be public or restricted to participants. The distributed nature of blockchain means it is almost impossible to hack.

As a result, the Garage component reduces risk and fraud, creates trust, puts the user back in control over their data and devices, and helps with governance and compliance.

4 Solutions and existing niche

4.1 DAO – decentralized governance

A decentralized autonomous organization (DAO) is a blockchain-specific organizational model targeting a common challenge that affects a broad spectrum of public and private entities — misaligned incentives when an authorized party is incentivized to act in their own benefit over the benefit of the person, entity, or group, or those they are enacted to represent. There is an inherent risk in the divergent goals, priorities, or access to crucial information of the respective parties that DAO solves.

The impact of implementing DAOs on a broad scale is enormous, perhaps equivalent to the real-world challenges of effectively applying technology that enables a DAO.

Wyoming became the first U.S. state to recognize DAOs as legal entities in 2021, and CryptoFed DAO was the first corporate entity to do so. A DAO can function as a corporation without legal status, typically and legally referred to as a general partnership.

On April 20, 2022, Tennessee Governor Bill Lee signed into law a bill to allow decentralized autonomous organizations (DAOs), to register as a type of limited liability company

At the same time, the challenges of designing and deploying highly sustainable, value-generating DAOs are significant. A DAO's rules set (includes airdrops, voting, incentive rewards, funding processes, etc.) is backboned by SCs that are used to lay the groundwork for the DAO's operations. They are evident, verifiable, and publicly auditable, allowing any potential member to comprehend how the protocol will operate at all times fully. Each implementation varies from the others, but they all share a set of common features — permissionless proposals for certain initiatives, voting, treasury, and token governance.

While those features are the basic pillars of decentralized governance, they also present downsides that could and must be addressed by the off-chain secure solutions.

4.1.1 Permissionless proposals

Typically, proposals are suggested by the community but may also come from DAO's members and are intentionally aimed at continued growth and success. Those proposals are aggregated to a board that efficiently manages DAO's strategy and vector. In that scenario, entirely permissionless proposals may dilute contributor attention. E.g., spamming the proposal board with incorrectly structured proposals or unrelated topics posted by unwanted actors. Potentially, it creates a mess of irrelevant ideas, ultimately harming the DAO's ability to operate efficiently.

Trisk encourages developers to overcome the problem by utilizing Form Builder (Studio's component). Form Builder enables DAOs to standardize the proposal submission experience and provide comprehensive guidance to the community member who wishes to submit a proposal. DAO could expect streamlined off-chain data collection that leads to a drastically reduced proposal bounce rate. Versioning subsystem used by Form Builder expands DAO's flexibility and significantly reduces building effort and development burdens.

Once created, the Form can be engaged with **Workflow Builder** (**Studio component**) and with **Engine** component automation to be delivered to the member via Trisk's UI or a custom UI representation implemented on top of Trisk's **API layer**.

Solid user experience with the Form not only facilitates the data collection part of the process. **Workflow Approvals** can significantly filter undesirable proposals before they get delivered to the entire DAO. **Workflow Approvals** could be utilized as the off-chain voting mechanism and effectively create a layer that controls proposal validation. It means a mandatory rule in reaching the threshold of a certain number of votes before a proposal enters the on-chain voting process.

4.1.2 Voting mechanism

DAO voting mechanisms rely on the token-holding members who participate in the decision-making process. It means that even if DAO has a tremendous number of members, its core is still fully dependent on only a small percentage of these members participating in votes. On-chain operations, such as voting, require payment of a transaction fee. Resulting high network activity may cause sky-rocketing fees which may ultimately have a negative impact on the voting process and dependent decision-making. This may become a potential downside that can even impact DAO's decentralized philosophy. These circumstances potentially disincentivize DAO members from participating in proposal votes.

Proposal making DAO members would be required to gather support for their proposal before publishing it to the public. As a result, extra voting could produce a substantial transaction gas fee problem.

Trisk addresses the problem by **Workflow Approvals** described above and unlocks the cost-efficient off-chain voting processing underpinned by **Workflow SC** engagement and on-chain results delivery.

Workflow Approvals, SC, and **Forms** can be auto-piloted by the **Engine** module that handles **Workflow** execution and processing.

Having a powerful **API layer**, Trisk enables developers to rapidly create a superior user experience or utilize Trisk's UI as an interim step for a quick DAPP delivery.

4.1.3 Off-chain governance and permissions

Off-chain governance resembles real-world politics in many ways. Various stakeholders inform their agreement or disapproval of a proposal through private or community conversation. Different groups aim to take control of the system by trying to persuade everyone else to support their cause. Typically, there is no automation or code that ties these groups to precise actions. While SCs address agreements and trustless issues, off-chain governance is essential to regulate access to critical or valuable resources.

Trisk unlocks new possibilities for DAO to determine off-chain policies and permissions in a reliable, auditable, and scalable way. The **Governance** module has 3 levels or granularly-configurable components: **Permissions**, **Persona**, and **Assignments**. Each component sets up the access boundary tied to the authorized user identity for each individual component within Trisk.

Furthermore, Trisk wants to extend **Governance** module capabilities as we advance for **MPC** and Engine's decentralization. This means that DAO could control node permissions and participation in the network.

4.1.4 Smart Contract automation

The major limitation impacting DAOs is the nature of SC — they cannot automatically trigger themselves. Specific conditions have to be met along with data provided by oracles in order to trigger the desired method of SC. In the context of a DAO's voting mechanism, it reflects the presence of an intermediary entity that initiates the method of SC when the proposal vote is passed.

There are solutions that address continuity and connectivity problems but Trisk stands as an ideal platform for no-code designing, maintaining, and enforcing policies. **Engine** component along with **Studio** and **Blockchain Manager** provides the bridge for off-chain data and traditional processes with on-chain computation. Any EVM-compatible SC can be mapped with the off-chain information and automatically triggered when necessary conditions are met. Developers can focus on the no-code business logic configuration rather than programming efforts.

Cost-efficient operation reduces gas fee burdens, backend development effort, and provides a robust toolkit for rapid product delivery.

4.2 NFT for artists

Web2 has opened up the possibility for anyone with an Internet connection to create any content. For example more than 500 hours of content are uploaded to YouTube every minute. Spotify's library contains more than 70 million tracks, 4.5 billion playlists, and more than 2 million podcasts, made by over eight million creators. But this has led to the problem that content very quickly loses its uniqueness by copying. There are no reliable tools to determine the authenticity of the authorship of this or that media.

Web3 solves this problem with NFTs - cryptographic tokens, each copy of which is unique (specific) and cannot be exchanged or replaced by another similar token which is assigned to one or another media. Blockchain allows the possession of such a token to confirm content ownership.

There are already about 80 million digital goods - NFT tokens - on powerful platforms for artists, such as opensea.io. And this market in 2021 generated over $23 billion in trading volume. Over 2.7 million Unique Active Wallets (UAW) connect daily to a blockchain dapp hosted in any 30+ blockchain protocols.

In addition to transforming the art market and solving the problem of digital scarcity, NFTs can introduce a new marketing tool for many businesses. NFTs can be used for any of the following:

- Giveaways - McDonald's celebrates the McRib's 40th anniversary,
- Promos - Warner Bros launched campaign for "The Matrix" and for "Batman",

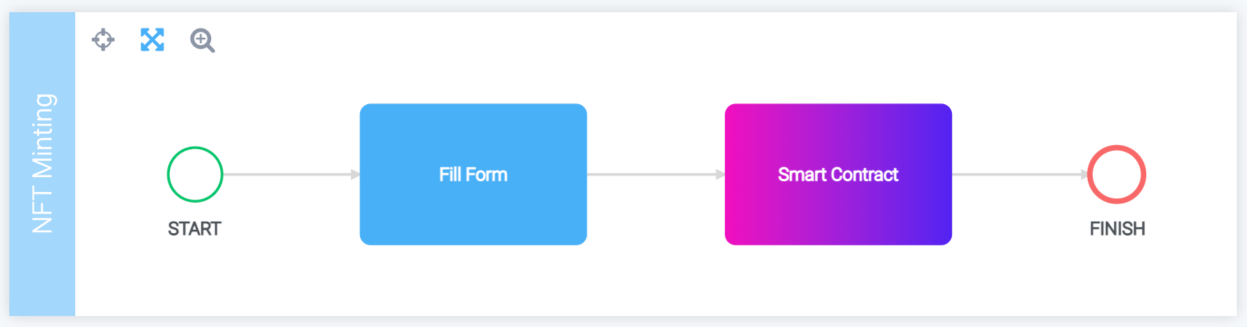
- Limiting goods editions - K-Swiss created a limited-edition sneaker with 55 pairs, each with an NFT digital rendition that also serves as proof of purchase,
- Licensing - you can sell/buy incredible moments from NBA as collectible cards,
- Getting celebrities autograph, and
- Gaming industry - for in-gaming accessories.

The work with NFTs can be broken down into two large blocks - minting and transfer of ownership - sending NFTs to wallets. The process of minting NFT tokens is not simple. This is evidenced by keyword searches such as 'nft & mint' in the biggest open source code base as GitHub, where we get roughly 3,500 results. This brings us back to the problem that if your business wants to try web3, it needs specialists, who are not always easy to find and incredibly costly.

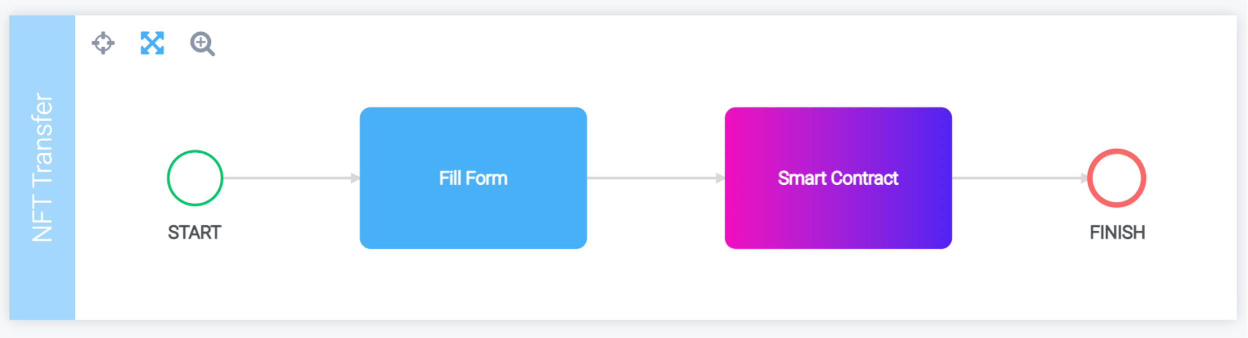
Let's look at how Trisk addresses this problem. For the minting process, the following actions are necessary:

- choose a file,
- upload it to Interplanetary File System (IPFS) - a decentralized protocol and peer-to-peer network for storing and sharing data in a distributed file system,
- using a smart contract, generate a unique token based on the file address in IPFS, and
- save the token into your wallet.

With Trisk, you can do it all with a workflow consisting of two steps. The Fill Form step - where you upload the file and specify the wallet where the token will be saved. With the second step, the SC will generate an NFT and write the token to your wallet.



Within Trisk, the transfer of ownership can be very simple. It is again a workflow consisting of two steps. First, in the Fill Form step - you choose which token and to which wallet it will be transferred. Then, with the use of the SC, you engaging web2 part with blockchain.



Trisk unlocks end-to-end solutions development for NFT creators by providing robust flexibility at all stages.

Using Trisk's native UI, the Studio enables creators to model a workflow for NFT minting / ownership transfer in minutes. For developers wanting to create custom customer facing UI, using the API gateway, it can be developed in hours. Whichever path a user wants to take, Trisk abstracts away the blockchain complexities and accompanying costs.

4.3 Real estate and escrow process

In the United States, the process of buying or selling real estate is complicated. Home buyers spend hours signing documents they don't read or understand, relying in most cases on the expertise of real estate agents and title company officers to get everything right. Commercial real estate transactions also get bogged down by agents, lawyers and title company officers. Other participants in the process, including county clerks, lenders, inspectors, insurance agents, and more, play minor, but important roles in the process.

Because the process is so dependent on people, the risk of error is high. The financial implications of mistakes or omissions can be significant. While each participant in the process may have internal controls to mitigate risks, because the end-to-end process entails multiple hand-offs of information and funds between parties, and the participants in the process who have the most to lose is the actual buyer and seller of the real estate, who knows the least about the process, there has been little incentive to implement an integrated solution.

Smart contracts and Trisk are the solution.

Each state and local jurisdiction has its own requirements to protect the public. So while there are at least fifty versions of these laws, each such variation can be codified into a series of smart contracts. State mandated automation would ensure performance, compliance and transparency.

To be in compliance, each participant identified above implements policies and procedures to comply with applicable state and local regulations. In lieu of the seemingly limitless documents required by real estate agents, title companies, lenders, insurers, and others, participants would create smart contracts to automate their internal processes. Furthermore, these participant level on-chain activities would connect with 1) state mandated smart contracts to achieve speed, accuracy and auditability that is presently not possible, and 2) other participants' smart contracts to ensure accurate and timely transfers of information amongst other users of information.

Trisk is the bridge connecting the on-chain and off-chain functions. In the present SaaS environment, Trisk's no-code developer platform enables the low cost development of DAPPs to interact with both the parties and the smart contracts. Trisk's integrated forms, workflows, communications and scheduling, enable participants to a transaction to not only reduce the errors and omission risk, but also provide the parties with up to the minute status visibility. Trisk's data dictionary serves as a database for information commonly used by participants.

Trisk will offer distributed document management through its garage feature. Incorporating Trisk's existing three dimensional data access management matrix into all aspects of this proposed end-to-end process, each participant is assured of requisite data privacy while data is in flight in addition to complete control of the data/documents permanently stored on-chain.

In the future, Trisk anticipates offering its solution not just in a SaaS environment but also as on-chain workflow processing.

Our vision, as described above, starts at the state level and works down through all participants in real estate transactions. By adopting Trisk's powerful development tool as a standard and establishing state and county level smart contracts connected by DAPPs built on Trisk, independent developers, as well as participating organizations, will be able to realize business efficiencies and mitigate error and omissions risk.

5 Future work

5.1 Super Smart Contract / Magniflow

Highly sensitive content such as personally identifiable information (PII), healthcare, financial, and intellectual property data requires further protection and secure processing. Traditional platforms typically store encrypted content, but at the moment of processing, data gets decrypted which may become a point of failure.

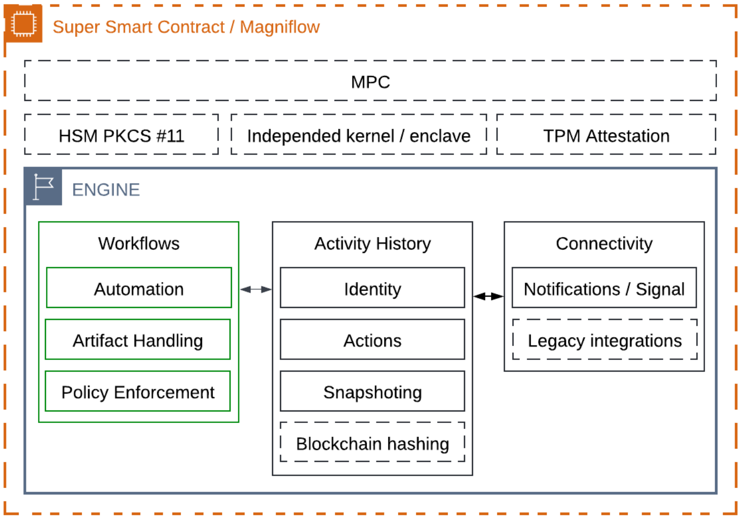
Keeping with a decentralized philosophy, we see MPC as a way to solve the problem where multiple parties join and process highly sensitive data without having to disclose or share the actual data with each individual party. MPC can even be done within the same organization to establish separation of duties. To reduce the attack surface area for their most sensitive data processing and ensure safety for customers, Trisk plans to offer an isolated, hardened, and highly constrained environment to operate security-critical data called Magniflow.

Magniflow consists of Trisk's Engine component that is embedded into a Secure Enclave.

A Secure Enclave is a fully isolated virtual machine, hardened and highly constrained with no persistent storage, no interactive access, and no external networking. Communication between blockchain node and enclave is done using a secure local channel. Even a root user or an admin user on the node will not be able to access or SSH into the enclave. A hypervisor is also used to further isolate the CPU and memory of the enclave from users, applications, and libraries on the parent node. These features help isolate the Engine and significantly reduce the attack surface area.

The encryption process that converts highly sensitive data such as credit card numbers or health care data into a token can become much more efficient because with Mangiflow, encrypted data can be sent to the Secure Enclave, where it is decrypted and processed. The node, network, database, and storage will not be able to expose or access the sensitive data throughout the entire path.

As part of the further evolution, we consider Magniflow as a decentralized network that facilitates MPC across all members (nodes) of the network with tamper-proof, secured, and encrypted data transition. Along with the decentralized Garage storage, built-in integration with other Trisk components and 3rd-party systems, customers are able to receive a high performance and cost-effective solution while developers create robust applications using Trisk's hybrid no-code ecosystem.



Super Smart Contract / Distributed enforcement
-- The intent is to migrate existing on-premises workloads, isolate engine computations in a secure and encrypted environment/enclave
-- Further to protect highly sensitive data such as personally identifiable information (PII), healthcare, financial, and intellectual property
-- Utilize modern HSM PKCS key management
-- Provide cryptographic proof of integrity
-- Establish resilience, redundancy, recoverability, and monitoring

Figure 7: Illustration of Magniflow concept and Super Smart Contract vision

5.2 The Studio Store

In the original vision of the Trisk platform, we considered subject matter experts (SME) as the crucial aspect of the business process modeling and successful execution. Every industry has seasoned experts who know and understand the who, how, what, when and where that makes their business environment thrive. The Studio Store enables these experts to become creators and monetize their expertise by creating content (i.e., forms, workflows, governance, and conditional logic) to be licensed to others.

Because of the hybrid no-code nature of Trisk, these SMEs are not required to have technological expertise, programming skills, or knowledge in the blockchain. SMEs are able to model the process using the Trisk Studio components and launch for themselves or for use by others.

From the web3 perspective, Smart Contract developers create very unique content – business processes enclosed into the source code. Developers can also deploy Smart Contracts on the Ethereum network and Trisk will immediately make these Smart Contracts available for use using the information from the blockchain explorer.

Throughout this whitepaper, we stress the difficulty of finding the right talent to apply web3 in traditional businesses. Typically, each party exists separately and struggles to find the other. SMEs want to monetize their expertise and IP developers search for real-world applicability and adoption of Smart Contracts. Last but certainly not least, consumers want quick access to content so they can quickly and easily go to market. The Studio Store will be the hub through which all interested parties interact.

We see the Studio Store as a significant opportunity to change the way many businesses and professionals operate. Whether it's a hotshot young developer wanting to create business applications for herself rather than her employer or a retired CPA with a knack for technology and years of experience who wants to monetize his experience, Trisk offers a significant opportunity.

SMEs (Creator) will be able to develop and publish their custom Studio content as an Application on the Studio Store and effectively make it widely available. Trisk customers (Consumer) are able to select the Application that facilitates their business needs and pay the fee *directly* to the SME. Developers (Developers) can monetize connections with Creators and Consumers who have real-world needs for Smart Contracts and web3. Should there be no existing Application on the Studio Store, a Creator or Consumer could post a description for the desired Application or Smart Contract and pick the right Developer for the implementation.

The Store in this scenario facilitates three main web3 adoption hurdles:

- Acceptance and adoption by the traditional companies who are in search of pre-package content
- Accelerate time to market by adapting existing and useful content quickly into real-world applications
- Matching Creators, Consumers, and Developers

6 Conclusion

Web3 world is rapidly expanding its footprint and will no doubt ultimately revolutionize compliance, financial industry, data ownership and governance. As with any previous technological revolution, it will likely take at least a decade to see big changes in the adoption rate by traditional companies. While blockchain technology, Smart Contracts, and DAOs are unfortunately associated with cryptocurrency and tokens, the biggest opportunities are related to information exchange, its protection, and audit.

Trisk has a first-mover advantage of lowering the barrier for web2 businesses to enter blockchain at virtually no-cost by utilizing a convenient hybrid no-code toolkit from Trisk. By bringing web2 companies into the web3 space, Trisk can rapidly extend real-world opportunities and use-cases for web3 developers, unlock new areas of blockchain implementations, significantly increase communities, and associate the creators' economy with not only NFTs and arts, but with industry-specific needs.

Having significant experience in compliance, Trisk can become a standard in governance for DAOs and provide efficient aid in off-chain policies with always connected on-chain execution and enforcement.

Trisk's abstraction philosophy brings ease of use into the complex web3 world, reducing development burdens, facilitating talent acquisition and training, and simplifying application maintenance.

And finally, with MPC and Magniflow on our sites, Trisk will establish a new level of secure workflow processing, information exchange, and data handling.