TTYH

Abstract—**Over the past decade, artificial intelligence has promised to reshape our society in many unique ways through a new technological revolution, and the impetus created by innovations like self-driving cars, real-time voice assistants, and other similar products has formulated a new axiom: *everything around us has to be smart*. Current market expectations are accelerating demands for -based solutions, but progress has been slowed due to high implementation costs, integration problems, and technical meta-complexity. However, there are several global communities of developers, researchers, and innovators who are working round-the-clock to solve real, complex problems. Harnessing the power of community-driven progress could be the key to extending the current boundaries.**

The current paper presents TTYH , an ecosystem which aims to encourage these communities to unite, monetize their intellectual property, and compete or collaborate to unlock the road that will lead to the Internet of Artificial Intelligence.

The TTYH platform abstracts the technical complexity of systems, providing an enhanced user experience and business integration. Through the use of a trusted execution environment and advanced cryptography methods, the underlying framework empowers scalability, guarantees data privacy, and protects intellectual property. Adaptive ontology models create a seamless data integration environment which nurtures cooperation and cross-organization interoperability.

Economic activities are coordinated by game theory-inspired rules and a Bayesian reputation system, while the infrastructure is based on a decentralized network governed by a federated blockchain.

Keywords—**artificial intelligence, blockchain, ontologies, technological innovation**

I. Introduction

In the recent past, unprecedented economic growth and social prosperity resulted from enhancing human labor through the use of machine power [1], and increasingly complex and abstract notions represented the primary catalyst of progress. The prosperity of human civilization correlates with the capacity to easily manipulate information-dense structures.

Today, humanity is currently witnessing the Fourth Industrial Revolution, a transition stage which is pushing things towards the era of cyber-physical social systems [2], [3]. In this dynamic context, analysing and controlling the complexity of the information systems becomes a burden, a challenge, and a priority. Alongside the growing complexity of these systems, the data also becomes the new oil [4], with those who have the tools to create, extract and understand it gaining a significant advantage over their competitors.

Artificial Intelligence ( ) is the enrichment of machines with human-like intelligence. Concurrently, intelligent machines are built to augment and enhance human capabilities. drives the transition to a new type of society in which intelligence is the governing factor, as well as a new vehicle for exchanging value. Any progress made in this area has a disruptive potential on everything [5], [6] by addressing the hardest problems we are currently facing, such as: climate change [7], [8], the energy crisis [8], disease-fighting [9], etc.

The abstraction layers, tools, products, and ecosystems that are currently being built, improved, adapted, or redesigned are the real trendsetters which will significantly drive the revolution forward [10]. There is a growing global community of promoters: developers, academic institutions, corporations, small companies, and government initiatives [11], [12] that are continually innovating and solving real and complex problems. Unfortunately, however, their innovation is not being leveraged at scale, because they choose competition instead of cooperation. Big tech companies and a handful of well-funded startups overwhelmingly set the direction and future of by building moats around their technologies and hindering future innovation. Since the current state of affairs implies that prevailing ecosystems exhibit severe limitations, particularly concerning fragmentation, isolation, and lack of an environment to stimulate evolution and mass-adoption, the only real path to growth is to create the Internet of Artificial Intelligence (IoAI). IoAI is an ecosystem that encourages internal coordination and cooperation among participants, rewards the real creators, motivates the contributors, and fosters seamless integration with well-established and emerging businesses.

II. State of the Art and Beyond

A. Centralized Approaches

Centralized platforms are dominating the current landscape. They accommodate users with the tools required to build intelligent applications. By combining smart decision-making algorithms with data, they enable developers to create advanced solutions. Some platforms offer pre-built algorithms and simple workflows, allowing for drag-and-drop modelling and visual interfaces, as compared to others that require in-depth knowledge of software development and coding.

IBM Watson is based on cognitive computing technology, and was created to support -driven research and development for enterprise products [13]. It integrates everything from big data manipulators to data analytics and infrastructure configurations. Through its series of tools, the platform is tailored for large corporations and data-intensive applications [14]. IBM Watson operates in a classic service infrastructure, where clients pay for all operations directly to a service provider which allocates resources and controls the flow.

Google brings with it the most comprehensive suite of tools, and offers an end-to-end cycle (from data ingestion to deployment) for building applications [15]. Aside from its smooth integration with all other Google services and tools, this platform also favors the usage of Kubernetes - the open-source container-orchestration system - to guarantee safe and flexible development-deployment processes [16]. Endorsed by many major tech corporations (Intel, Nvidia, others), the platform aims to offer its services not only to big enterprise clients, but also to small business solution providers.

Microsoft Azure has evolved into an intelligence system built right on top of the Microsoft Azure platform, with mostof its features being designed to focus on three pri mary areas:

apps & agents, Knowledge mining, and Machine learning. An intensively-addressed topic in the context of Microsoft Azure

be used for real-world problems, starting from the input dataset to the final result. The declared goal is to enable non-technical

Amazon Machine Learning presents a full stack cloud-based solution for machine learning development by providing infrastructure, frameworks, and services for Artificial Intelligence ( ) and Machine Learning (ML). Amazon ML quickens the customer's business process by allowing them to smoothly integrate machine learning algorithms. It also provides simplistic integration with the user's application and frameworks for deep learning and a comprehensive documentation [20]. It integrates existing frameworks such as TensorFlow, PyTorch, and Apache MXNet, and comes with pre-installed deep learning frameworks. Combining services from different ML stack levels is possible by using workloads.

B. Decentralized Approaches

As a response to the rigidity of centralized environments controlled by big corporations, several projects have begun using blockchain as a foundation for new services. These new models follow the principles of decentralized autonomous organizations, and empower communities to unite and contribute to a swarm-like ecosystems.

SingularityNET is a platform where users create, share, and monetize services at scale on top of a decentralized economy. All services revolve around a marketplace that connects creators with consumers. Although it plans to become blockchain agnostic, SingularityNET currently runs on Ethereum, and uses a consensus called Proof of Reputation which was derived from Proof of Stake [21]. Aside from the components of the SingularityNET platform, Singularity Studio [22] uses external tools such as inter- collaboration framework, OpenCog Artificial General Intelligence engine, and TODA secure decentralized messaging protocol, and many others. Although SingularityNET utilizes a Proof-of-Reputation system, the token holders still control the democratic decision-making process.

Effect integrates three components: a marketplace for outsourcing small tasks(Effect. Force), an registry with algorithms (Effect. Smart Market), and an infrastructure layer (Effect. Power). Effect is building a computational environment to apply automation processes on flexible data sources. Its blockchain structure was initially designed for NEO [23], but currently runs on EOS [24] and represents the foundation for interoperability of algorithms and services. Effect is focusing on gathering a global network of freelancers to offer on-demand integrations for clients in different industry areas (sentiment analysis, language translation, chatbot training, etc.) [25].

providers, data consumers, service providers, marketplaces, and network keepers. The network architecture has 5 components: Frontend, Data Science Tools, Aquarius, Brizo, and Keeper Contracts. The core innovation behind the protocol lies in the decentralized layer which offers Service Execution Agreements (SEAs) as a fundamental method for enabling tracking, rewards, and dispute resolution in a Web3 data supply chain. The Ocean token is at the centre of the economic model, and allows actors to share and monetize data while at the same time ensuring control, auditability, transparency, and compliance [26]. As clearly mentioned in the technical papers, Ocean Protocol is a substrate for Data & Services which focuses extensively on the infrastructure and governance of the protocol representing the foundation of all other components.

DeepBrain Chain was launched as an open platform in China, slowly evolving into a global platform which aims to reduce the cost of hardware for processing by 70%. DeepBrain decentralizes the neural network operations needed for training models. The incentive model relies on mining (processing), and rewards infrastructure providers with DBC tokens. Already-confirmed use-cases in areas such as Driverless Cars, Speech Recognition, and Tumor Detection reinforce the capabilities of the platform to match requests and resources

[29].

C. Current Challenges

The centralized and decentralized approaches presented above offer a broad viewpoint concerning the current state of the field. The selected platforms represent the starting points on understanding the key ingredients, constraints, and pitfalls towards advancing the domain. Fig. 1 provides a pragmatical and condensed side-by-side visual perspective depicting the critical traits which require attention. All values are collected by researching the official documentation, or by directly testing the solutions. The information is structured to show the main differences between ecosystems and the TTYH approach. TTYH aims to use this model as a guideline to empower community contributions, while keeping the environment safe for seamless business integration.

By surveying the current platforms related to the project scope, we have identified several challenges that need to be addressed, in order to be able to take things further:

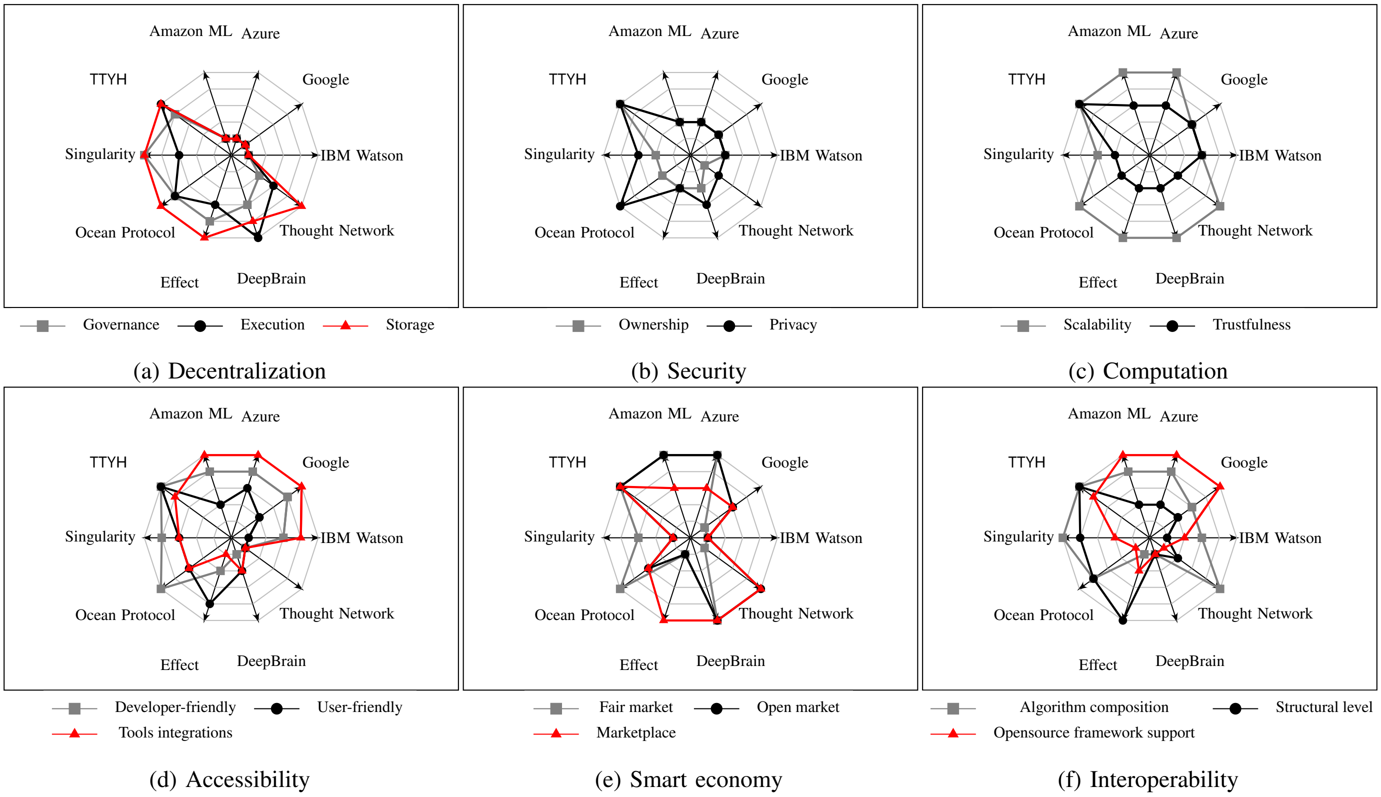
- Decentralization - no central entity that controls the location of data or information processing;
- Security - protect end-user privacy and guarantee intellectual property rights;
- Interoperability - use of standardized interfaces to allow multiple agents to cooperate and connect in providing relevant answers to complex problems;
- Accessibility - simplify the interactions between end-users and AIs by providing straightforward, non-technical flows;
- Smart economy - create a built-in robust exchange medium that facilitates fair transactions between supply-and-demand of services;
- Computation - expand network capabilities by allowing network participants to rent their computing power for execution and training of AIs;
- Datastreams - provide mechanisms to distribute and consume vast amounts of data for training and execution.

D. Conclusions and Next Step Forward

After a complete overview of the domain (Sections A, B) and side-by-side technical comparisons identifying the challenges (Section C), the path towards the Internet of seems to be guided by two main conclusions:

1) The centralized platforms are secure and reliable, but their components are private, the information is siloed, and the ecosystem is divided among the few big players.
2) The decentralized platforms are slowly evolving into mature products with many use cases and relevant business models, but the need for standardization, flexibility, and open community has to be more present in order to produce a larger impact wave.

In this context, TTYH is learning from existing approaches by using the positive outcomes from both worlds (centralized and decentralized) to move things closer to the genuine concept of the Internet of that can provide the long-awaited visible progress.



(a) Decentralization

(b) Security

(c) Computation

(d) Accessibility

(e) Smart economy

(f) Interoperability

Fig. 1: platforms review

III. TTYH Ecosystem

A. Architecture Overview

TTYH is an ecosystem which harnesses the power of blockchain [30], distributed ontologies [31], advanced cryptography [32], and a trusted execution environment by developing a novel infrastructure that is capable of supporting the authentic revolution. Inspired by the primordial soup that stimulated the interaction between replicators [33], our system provides a consistent medium wherein agents can combine seamlessly to support increasingly more advanced features. Agents' intercommunication is defined by ontological schemes.

The evolutionary pressure [33] that controls agents progression is governed by a distributed secure rating mechanism [34]. Fig. 2 shows the overall architecture of TTYH . The design is composed of several structural elements that provide an open, practical, scalable, and extensible system. The logical structure of the architecture implies several layers. At the top are the stakeholders interacting with the platform through specialized tools connected to the underlying network services operating on DLT (decentralized ledger technologies).

B. Stakeholders

Aiming to create a self-sufficient ecosystem, TTYH envisions holistic interactions [35], [36] between four significant stakeholders:

- Service Consumer
- Innovator
- Data Provider
- Infrastructure Provider

The service consumer represents the end-user who needs to solve particular business problems to enrich their products or services with an extra layer of intelligence. According tothe level of integration and expertise, the service consumer can utilize services via the TTYH marketplace or the SDK.

The platform simplifies the experience by eliminating the prerequisites of having technical skill or owning the hardware equipment required for running and training . The innovators utilize their expertise to create elaborate and practical algorithms capable of solving complex business problems. The platform incentivizes the innovator to focus on delivering high-quality solutions and engaging in cooperation instead of competition, by reusing algorithms deployed by others to build ever-more intricate solutions.

The data providers ensure the vast amount of data required to train and test algorithms. Companies that possess consistent datasets can make a profit by licensing them on the platform to be used by the innovators and data providers.

The infrastructure providers supply the platform hardware capabilities required for running and training AIs. Solving challenging and complex problems requires a considerable amount of computing power that can only be achieved through the joint participation of multiple providers.

C. TTYH Platform

The TTYH platform includes the high-level system components; it abstracts the infrastructure complexity from the uppermost levels, facilitating the business use cases and integration with extension modules. The main components are:

- TTYH store (DApp) - marketplace providing the end-user with access to platform functionalities;
- TTYH toolkit - contains the collection of IDE extensions, CLI tools, libraries, frameworks, and development tools used by innovators.
- TTYH SDK - provides programmatic access to the platform functionalities;
- TTYH daemon - represents the client maintaining the connection to the network;

D. OSAK Network

The network layer contains nodes implementing the core service and the execution environment. Operating on top of a federated P2P network, the TTYH protocol provides core services by integrating modules which manage storage, execution, interoperability, rating, etc. The TTYH workers are specialized nodes running on top of the Infrastructure providers' hardware. At this level, the algorithms are being executed inside a secured environment.

E. Distributed Ledger Technology / Blockchain

The system replication, decentralization, and communication are ensured by the DLT layer. Representing the communication backbone of the system, a set of decentralized technologies is integrated, providing support for storage [37], [38], [39], state management [40], [41], resource allocation, etc. From a technical perspective, the platform should be agnostic to the DLT solution. With the use of blockchain, the inherited features of trust, security, and transparency are ensured. Maintaining the state of the ledger is secured by a consortium of companies, with each company running a set of peer nodes. Adding new members to this network requires complaining to the governance model parameters.

F. Decentralized Operating System (DOS)

Coordinating a set of services of such scale and complexity requires a novel approach towards the entire system architecture [42]. The fair and secure governance of the platform is enforced by a set of protocols and procedures. The network services, resources and processes are supervised by a decentralized / distributed operating system (DOS) [43], [44]. Composed of a trusted P2P [45] network of nodes, the DOS ensures the proper functioning of the system. As depicted in Fig. 3, the P2P network acts as a host for a virtualized multidimensional consensus mechanisms. Through the use of a dynamic consensus mechanism [46], the network operates numerous decisions in parallel required to run the DOS services. According to the importance of the resolution, degree of security, and impact, the number of nodes needed to reach consensus may vary.

Resembling the logical structure of a traditional OS, the nodes are organized into logical layers:

- Application layer - operates protocols intermediating the interaction between users and the system. This isolates the user from the implementation details, in conjunction with offering an improved user experience.
- Kernel layer - encapsulates protocols and procedures for searching, rating, executing, and registering of AIs, datasets, and infrastructure.
- Resource layer - adapts the underlying technological dependencies, such as IPFS [47], blockchain, and operational smart-contracts used to allocate, register, and rate AIs and infrastructure providers.

IV. Economy

In most economic systems, the products and services carry a built-in layer of knowledge that is indistinctive from the rest of the system, blurring the real value of the intelligence. TTYH creates a novel form of economy which exploits the intelligence in its purest form. In TTYH 's economy, knowledge represents the primary governing factor and fundamental value system. The critical element in propelling the law of accelerated returns [48] is to design appropriate incentive mechanisms [49] that empower innovators to monetize their work and encourage collaborative exploration and research. Building a genuinely sustainable economy of intelligence [50] requires securing a high degree of quality and performance for all network services.

A. Economic Interactions

TTYH provides a built-in peer-to-peer marketplace that facilitates transactions between main stakeholders, as depicted in Table I:

- The service consumer uses the network to get solutions toward particular business problems carrying the expense for and execution services.
- The infrastructure provider rents the execution environment to run AIs, and is getting paid on a per-execution basis, in accordance to the amount of hardware resources being utilized.
- The Innovator adds value to the network by developing and deploying solutions capable of solving demanding problems, being paid each time that the implemented is executed.
- The data providers own the relevant data that can be used during the process of training; they are getting paid each time their datasets are downloaded.

Stakeholder	Offers	Receives
Service Consumer	Payment	Solution
Innovator	Algorithms	Per-execution payment
Data Provider	Training datasets	Per-download payment
Infrastructure Provider	Exec environment	Per-execution payment

Table I: Economic model

The economic exchange between parties utilizes a pay-per-execution model based on an intrinsic token. Each actor that performs commercial operations on the platform possesses an account (a simple public/private keypair, used to sign transactions) and - optionally - a wallet (smart-contracts that allow advanced features, such as transaction logging, multisig, withdrawal limits, and more) [51].

B. Payment System

The payment system must guarantee the fair, fast and secure transfer of funds between parties, when doing transactions. The protection of all involved parties needs to be enforced at the protocol level. The receiver needs the guarantee that the sender has the available funds to pay, and that they will get paid once the job has succeeded. The sender requires that they pay only if the job is concluded successfully, in a specified time frame. Fostering an improved user experience can be accomplished by diminishing the response and bootstrap (execution startup) times. For the end-user, the system response should be almost instantaneous, which is achievable by the combination of an escrow smart-contract [21], [52] and unidirectional atomic payment channels. A payment channel [53], [54] enables the secure off-chain transactions between parties without blockchain delay. Considering the simple flow displayed in Fig. 4:

1) Alice (service consumer) deposits funds into the escrow smart-contract with a specific withdrawal timeout. She can recover the funds only after the timeout has passed;
2) Alice opens a payment channel with the system (DOS);
3) Alice submits a signed request to run an algorithm employing a time-bound execution environment configuration *EE*;
4) The system checks whether Alice has the required funds in the escrow;
5) The system will send a message to Bob (infrastructure provider) to prepare the execution;
6) The system will lock the funds required for the execution for a time frame;
7) Alice sends the input data over a TLS connection, Bob starts the execution of ;
8) If Bob has successfully concluded the task, the payment channel will be closed, and funds will be released into his account;
9) If Bob was unable to perform the job in the allocated time frame, the funds would be unlocked.

The above-presented design poses some limitations in enforcing the closing of the channel before the funds can be released. This approach affects the overall performance, and may introduce some significant delays [55], especially in the case of batch execution.

The path to solving this limitation is to add a nonce to the channel, which will act as a wrapper containing the general commitment and expiration timeout. Inside the main channel, subchannels will be spawned, with each one possessing its commitment and timeout. The channel nonce and expiration timeout will be updated each time a new request is initiated. Alice can submit multiple execution requests, with each one being processed separately. Following this approach, the payment channel possesses the following favorable characteristics:

- The channel between parties can persist considerably;
- The sender can add funds to the channel and extend the expiration timeout, as needed;
- The recipient can claim the agreed-upon amount at any time;
- The underlying blockchain system delays will not affect the transactions between parties.

C. Incentive Mechanism

Achieving the Nash equilibrium [36], [56] in a decentralized ecosystem that fosters collaborative innovation implies the use of a built-in, self-regulatory reward system. Fueling the chain reaction which propels the ecosystem toward fast and sustainable growth, the incentive mechanism behind it must solve the well-known network-effect bootstrapping problem [57].

TTYH empowers innovators on creating novel solutions by piecing together functionalities of the previously-deployed algorithms. Fig. 5 depicts the dependency structure in which the root algorithm u makes use of the children sub-algorithms q_i. In turn, each child algorithm may integrate further sub-algorithms.

TTYH utilizes an incentive tree mechanism [58], [59] that encourages cooperation over competition. All algorithms in the tree at any level will be rewarded, while keeping a fixed cost for the end-user. T_u represents the sub-tree rooted at node u having k children q_1,...,q_k, T_{uq} represents T_u's first level children. The total cost of the T_u can be computed as:

$$Cost(T_u) = p_u + \sum_{i=1}^{k} C(q_i) * p_{q_i} \tag{1}$$

where p_u represents the node cost (specified by the innovator), and $C(q_k), (0 \leq C(q_k) \leq 1)$ represents the contribution function of the node q_k. Computing the $C(q_k)$ function requires considering the general rating of the algorithm, the relative position in the current tree, and how often other algorithms use the algorithm q_k.

Given the total contribution $C(T)$ and $\pi(x)$ functions defined below:

$$C(T) = \sum_{u \in T_q} C(u) \tag{2}$$

$$\pi(x) = \beta x + (1 - \beta) x^{1+\rho} \tag{3}$$

the reward function $R(u)$ [60] is defined as:

$$R(u) = \Phi C(T) \left[\pi(\frac{C(T_u)}{C(T)}) - \sum_{T_{qi} \in T_{uq}} \pi(\frac{C(T_{qi})}{C(T)}) \right] \tag{4}$$

where $\Phi(0 \leq \Phi \leq 1)$, $\beta(0 \leq \beta \leq 1)$ and $\rho(\rho > 0)$ are system parameters controlling participant reward distribution.

The designed reward mechanism possesses the following properties:

- Truthfulness: no one could increase its utility by acting maliciously;
- Sybil-Proofness: no one could benefit from generating multiple fake identities;
- Individual Rationality: no algorithm has a negative utility in being used as a sub-algorithm.

V. Ontology-based Data Integration

The heterogeneous format of the data represents the top impediment in creating a genuinely interoperable cross-organization system. Developing a common data format to act as a universal language represents one of the largest challenges that needs to be addressed in order to create a truly interconnected and interoperable ecosystem.

A. Data Format

Based on [61], TTYH utilizes a protocol composed of interoperable ontology models [62], [63] representing the inputand output of the agents. Although each organization hasits semantics, context, and perception of the data, this protocolwill act as a translator/ abstractor fostering internal and externalcollaborations. As depicted in Fig. 6, the model proposesa layered architecture in which each l ayer is composed ofmachine-readable semantic data structures that provide contexton a pa rticular dimension of the ontology concepts. The most important aspect of this approach is that data structures can be used, deployed, and updated in a decentralized manner. The core layers are designed to store machine-targeted semantic information. The optional layers provide human-readable information in the communication between agents, but are compelling in cases which involve human-computer interactions. From a high-level perspective, the architecture is composed of the following layers:

- Structural layer - the formal specification of the ontology in its purest form, composed of concepts, properties, and relations;
- Connection layer - contains information about the location of concepts from external ontologies and mappings between multiple versions of the same ontology;
- Encoding layer - specifies the used encoding format, like UTF-8, ISO, or any other chosen format;
- Defaults layer - used to define the fallback values for specific properties;
- Validation layer - used to add formal validation rules for schema properties;
- Restriction layer - contains a set of contextual restrictions between schema properties;
- Naming layer - tags schema properties classes and relations in human-readable format;
- Instruction layer - includes guidance information on how the user should provide the input data;
- Versioning layer - contains community proposals about schema structure changes and future evolution;
- Template layer - used for the contextual fragmentation of the schema.

B. Compatibility and Versioning

Efficient ontologies need to be plastic, subject to constant change and improvement, but at the same time, they also need to be sufficiently stable for consistent communication. The current architecture [61] ensures the plasticity and stability of the structure by using versioning and property mapping mechanisms. Further, the formal model of the transformations and the specific characteristics are defined from a mathematical perspective. Considering the following notations:

- property P with definition domain D on version X
- property P' with definition domain D' on version Y
- from a temporal perspective $X < Y$
- f is the forward transformation and f' is the backwards transformation

mapping M can be defined as

$$M(P, P') = \{< f, f' > | f : D \to D', f' : D' \to D\} \quad (5)$$

Depending on the evolution of the domain between versions X and Y, the following cases emerge:

- $f(P) = P'$ and $f'(P') = P \Rightarrow$ no loss of information between versions X and Y then $M(P, P')$ is called a stable mapping, noted as $Ms(P, P')$;
- $f(P) = P'$ and $f'(P') \neq P \Rightarrow$ there is a loss of information on the backward transformation between versions X and Y then $M(P, P')$ is called a forward-stable mapping noted, as $Mf(P, P')$;
- $f(P) \neq P'$ and $f'(P') = P \Rightarrow$ there is a loss of information on the forward transformation between versions X and Y then $M(P, P')$ is called a backwards-stable mapping, noted as $Mb(P, P')$.

Considering $T = \{M(P, P')\}$ as the set of all mappings between versions X and Y, the information between versions may be transported bidirectionally with ease when

$$T = \{M(P, P') | M(P, P') \in Ms(P, P')\} \quad (6)$$

Even though in practice implementing a full set of stable mappings is a difficult task, it provides an excellent, ideal goal to aim for. Of course, the ideal case presented above is unlikely to be found in practical cases, which is why the case of partially-stable mappings is closer to reality:

- partially backwards stable

$$T = \{M(P, P') | M(P, P') \in Mb(P, P') \cap Ms(P, P')\} \quad (7)$$

- partially forward stable

$$T = \{M(P, P') | M(P, P') \in Mf(P, P') \cap Ms(P, P')\} \quad (8)$$

Information might be lost in the partial stability cases, which is why it is necessary to consider a contextual and gradual approach that is tailored to a domain's specific needs.

C. Algorithm Composition

Aside from the formal definition of data structures, allowing cross-organization interoperability, the model presents a more subtle - but strikingly powerful - feature. The consistency and uniformity achieved by ontology models [61] facilitate integration by providing a clear data-contract at the algorithm boundaries (input and output).

This feature opens up new possibilities for creating smarter solutions which combine the functionalities implemented by existing algorithms. As depicted in Fig. 7, the proposed mechanism uses gRPC [64] stubs to allow remote invocation and ontology concepts to ensure communication and data consistency. The presented architecture enables incorporating functionality from multiple AIs with custom logic blocks. From the development standpoint, this will be perceived as including an external library. The TTYH toolkit will generate all the boilerplate required to perform the integration.

VI. Execution Environment

Aiming to create a planetary scale supercomputer requires the intensive participation of as many infrastructure providers as possible. Supporting the implementation of such scalability and openness requires a new approach toward security, data privacy, and execution models. Isolation can be achieved through the use of virtual machines or containers [65]. Virtual machines ensure a higher degree of isolation and security [66] by emulating the entire hardware, but with a more significant execution overhead. Containers [67], [68] are able to reduce this overhead through the use of the underlying host operating system's kernel, at the cost of sacrificing some isolation in the process. The security level provided by the traditional isolation approach operates on the premise that the host environment is trusted. Given the decentralized nature and the open approach towards infrastructure providers, it is possible that certain actors may act maliciously. The conventional isolation mechanisms need to be amended with new security policies and mechanisms. The designed system must secure protection at multiple levels:

- at the service consumer level, it must guarantee data privacy and provide accurate results in a fine time frame;
- at the infrastructure provider level, it must protect the execution environment against malicious code;
- at the innovator and data provider level, it must secure intellectual property and protect against unauthorized access.

A. Trusted Execution Environment

To achieve all the security requirements presented above, the TTYH execution container utilizes the SGX enclavemechanism . Intel's Software Guard Extensions (SGX) [69] isa mechanism which ensures an application's confidentialityand honesty, even if the OS, hypervisor, or BIOS are compro-mised. The SGX mechanism even protects against attackers that are able to physically access the machine. The primary SGX concept is the enclave [70], a fully-isolated execution environment in terms of process space and virtual memory. The enclave memory containing the application code and data does not leave the CPU package unencrypted. When the memory content is loaded into the cache, a specialized hardware mechanism decrypts the content, and checks cache integrity and the virtual-to-physical memory mapping. Upon startup, SGX performs a cryptographic check on the integrity of the enclave, and provides attestation to remote systems or other enclaves. [71], [72]. SGX provides two major benefits:

- the remote system cannot modify the program that is executed in the enclave;
- the code being executed is in plain text only inside the enclave, and it stays encrypted anywhere else.

The proposed solution protects the execution environment against attack vectors or unauthorized access that may arise from the host machine (it ensures that the infrastructure provider cannot view/alter user data or the code at runtime).

Inspired by [71], [73] and [74], the TTYH trusted execution environment (TEE) architecture combines a secure Docker [75] container with SGX enclave mechanisms. The TTYH TEE architecture is portrayed in Fig. 8.

At the operating system level, specialized kernel drivers ensure SGX integration. The enclave abstraction layer intermediates a secure conversation between the enclave and the system. Inside the enclave, the application code and data is secured by the use of the built-in SGX hardware encryption mechanisms. The library stack includes an OS shield, a set of OS libraries, libc, and other user binaries. It provides mechanisms that allow access to the standard library, operating system calls, and dynamic library loading.

The environment utilizes a manifest file describing the type of resources that are required by the application to run (declining the execution of the application with a questionable manifest). The manifest can also specify the hash (SHA-256) of trusted files and directories accessible from the enclave.

B. Secure Execution Flow

As depicted in Fig. 9, the trusted execution is ensured by the interaction between a set of system components, processes, and actors, as follows:

- The consumer initiates the process by submitting an execution request specifying the to run, and the hardware requirements to run it;
- The DOS broadcasts the request toward available infrastructure providers and performs the selection matching the request;
- After the infrastructure provider is selected, it starts the process of bootstrapping the Enclave;
- Inside the Enclave, the encrypted binary is downloaded from the decentralized storage (e.g. IPFS [47]);
- Once the Enclave is ready, it connects to the DOS;
- The DOS checks the authenticity of the Enclave through remote attestation [76], [77];
- If attestation has succeeded, the DOS will add a time-bound ACL (access-control list) entry, enabling the Enclave to run the specified ;
- The consumer will receive the Enclave connection details;
- Through a secure TLS connection, the consumer sends the input data to Enclave;
- The Enclave requests the key required to decrypt the binary from the Secret Store;
- If the Enclave ACL entry is valid, the Secret Store sends the key;
- With the received key, the Enclave decrypts and executes the binary;
- The process ends after the response is sent to the consumer.

The presented design guarantees that all security requirements are met. Keeping the binary accessible only inside the attested Enclave limits any unauthorized access, while also preserving the innovator's intellectual property. The TLS communication channel between Enclave and consumer assures that data privacy is kept.

VII. GOVERNANCE

Even though blockchain technology is immutable by nature, TTYH has to be able to address any market challenges and continuously adapt. The ecosystem must include mechanisms that can apply adjustments to its components, when needed. Conceptually, there are two categories of changes that can be applied: changes in system parameters, and changes at the protocol level.

The system parameters guide the proper functioning of the platform. The actual values for these parameters should be the subject of system-wide voting sessions. Some examples of parameters that can change over time are the execution fees and the minimum number of peers required for a specific task.

The protocols themselves need to be adjustable and - with sufficient agreement - it should be possible to introduce new rules and constraints. Amendment of the existing rules will be necessary for the future. As the TTYH ecosystem is decentralized, there cannot be a single organization or person performing these changes, so the network has a governance system that allows prominent actors in the community to propose and vote for improvements.

A. Improvement Proposals

Protocol level and system parameter adjustments should be submitted to an enhancement proposal system. Each proposal contains logic for adjusting parts of the ecosystem. A proposal is only executed if a majority of the council members have voted in favor of it within a time limit.

B. Governance Model

The governance is ensured by a council formed by a group of individuals and organizations that are allowed to cast a vote on improvement proposals. This council is dynamic in size; one can leave the group at any time, and new members can join if the majority approves them. The council members are responsible for continuously applying changes to the network, so that it can adapt to the frequent changes in the market.

C. Consensus

From a scientific perspective, a decentralized system represents a replicated state machine [78], [79]. The replication ensures the system's consistency, liveliness, and fault tolerance. Traditionally, the transitions in system state are endorsed by a unique consensus mechanism that is operated by the entire network. This approach has proven to be slow [80] and inefficient, in terms of energy consumption [81], since every decision requires the involvement of the entire network.

In order to ensure a fast, scalable and responsive system, TTYH is utilizing the dynamic consensus [46], a superset of the traditional consensus architecture. As depicted in Fig. 10, the physical network is virtualized; a node represents a computing element engaging into multiple consensuses in parallel. The Dynamic Consensus represents a new architecture extension which allows multiple, complementary, consensus algorithms to run on the same platform. This approach ensures that several decisions covering different topics can be reached in parallel.

In a distributed system, any component of the network may be faulty at any moment, so the system design shouldincorporate mechanisms to prot ect against these faulty nodes[82].

In order for the system to be fault-tolerant, it must introduce redundancy and replication of the information, as well as the capacity to isolate the faulty nodes [79].

From an infrastructure perspective, the TTYH DOS runs the dynamic consensus model on top of a federated network. The DOS represents the backbone of the ecosystem, and is responsible for all governing aspects.The dynamic consensus guarantees byzantine fault resilience by adding the replicas in the communication protocol; it will provide a minimal number of nodes on any level of the consensus.

D. Regulation Compliance

The GDPR (General Data Protection Regulation) [83] is an EU regulation for protecting an individual's fundamental right to privacy; it broadly defines personal data as ”any information” that relates to an identified or identifiable living person Art. 4(1). The GDPR's core principles are lawfulness, fairness and transparency, data minimisation, data storage and purpose limitation, accuracy, accountability, integrity, and confidentiality. Enforcing GDPR in a decentralized system is a controversial and challenging task; it creates a tension between protection of the fundamental rights and technology innovation [84]. Principal difficulties that should be resolved are blockchain data removal (blockchains are immutable), automated decision-making in a smart contract that can be contested, and deciding on who the data controller is. Various studies [85] [86] [84] tried to show recommendations on how DLTs can be GDPR compliant, but there is no standardization on this subject. In TTYH , the user is the owner and controller of their sensitive information. For publicly-disclosed user attributes, the GDPR rules are still applicable, and the user should grant consent for allowing to have that data revealed on the platform. The user can revoke access to their data. To ensure privacy and security, users encrypt data before sending it to the platform, and in addition, the off-chain storage pointer (address) is encrypted. When the user requests data removal, the platform will destroy the encryption keys. Even if data continues to exist on the off-chain storage (when the deletion is impossible, e.g. IPFS), the content cannot be accessed without decryption keys. TTYH implements a logging mechanism wherein activities that include personal data are recorded.

VIII. Enterprise Integration

In the enterprise context, managing and optimizing resources is achieved through a service-oriented approach leveraged by the use of the Cyber-Physical Systems paradigm. Cyber-Physical Systems are systems of cooperating computational agents that possess a high degree of connectivity with the surrounding physical environment and ongoing processes [87]. Cyber-Physical business processes orchestrate the actions of IoT (Internet of Things) devices and embedded ICT (Information and Communications Technology) systems (e.g., smartphones, sensors), and in doing so, they strongly determine the coordination of real-world entities (e.g., humans, robots, etc.) [88].

The TTYH platform provides integration tools for augmenting the enterprise environment with a cognitive layer. Through this, the enterprise application can access TTYH services that are used for the better management and optimization of Cyber-Physical Systems processes, resources, and activities.

Promoting innovative changes in business ecosystems often represents a disruptive, costly, and inefficient process, and is one of the most significant impediments in the adoption of new technologies. The proposed integration model offers a symbiotic, non-invasive approach with a minimal impact on existing processes. The integration is achieved by providing extension points at the Workflows and Microservices [89] level. Fig. 11 presents a simplified enterprise model covering the integration part. From a logical perspective, this can be structured into the following layers:

- Cyber-Physical layer - represents the connection with the surrounding physical environment;
- Service layer - contains computing systems used to orchestrate and manage enterprise resources;
- Cognitive layer - leverages services through the enterprise.

A. Cognitive Layer

The cognitive layer provides a set of microservices, connectors, adapters, and REST services, granting access to the TTYH ecosystem. From an infrastructure viewpoint, these services may run on the company infrastructure, or they may be offered by external providers.

B. Service Layer

The service layer contains enterprise domain-specific business rules, data models, and workflows. Preparing the company's unstructured data for ingestion requires passingthem through a normalization / cleanup process. This process makes use of ontology-based data abstractions [62]. From a user perspective through the use of workflows (sequences, flowcharts, and transactional business processes), complex business problems could be formally described and automatized with ease [90], [91]. Workflows operating in an unattended (without human supervision) or assisted (with human guidance) context can be enhanced with capabilities that are provided by the cognitive layer. The workflow activities, tasks, and decision blocks could be connected to TTYH AIs through the use of microservice connectors.

C. Cyber-Physical layer

The Cyber-Physical layer represents the connection with the surrounding physical environment. It supervises digital twin instances of IoT devices, robots, industrial machinery, processes, and humans [92], [93].

IX. Reputation System

The quality of services transacted on the TTYH platform is subject to the quality of the ratings collected from end-users. The fundamental challenge is that a user can provide ratings that are not truthful to the actual experience that they had with the . When users provide ratings outside of the control of the relying party, it is difficult to know a priori when a user has submitted a dishonest evaluation. However, it is often the case that unfair evaluations diverge in their statistical patterns from those of the accurate and honest reviews [94]. TTYH utilizes a Bayesian rating system [95] based on an analytical filtering technique, ensuring the exclusion of unfair ratings [96] [97]. The reputation score represents an indicator of how a particular , infrastructure, or dataset will behave in the future. Mathematically, the beta probability density function (PDF) can be defined using the gamma function Γ as:

$$beta(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{9}$$

where α and β represent the amount of positive and negative ratings. As depicted in Fig. 12, when nothing is known, the beta PDF has a uniform distribution where $\alpha = 1$ and $\beta = 1$.

The distribution readjusts after observing r positive and s negative evaluations. For example, the beta PDF after observing 7 positive and 1 negative ($\alpha = r+1$ and $\beta = s+1$) outcomes is illustrated in Fig. 12. $E(p)$ represents the probability expectation of the $beta$ function defined as:

$$E(p) = \frac{\alpha}{(\alpha+\beta)} \tag{10}$$

The rating system is composed of vectors $\rho = [r, s]$ where $r \geq 0$ and $s \geq 0$. The aggregated rating of service Z at time t_R performed by reviewers X from the community C is defined as:

$$\rho^t(Z) = \sum_{X \in C} \rho^X_{Z,t_R} \tag{11}$$

Taking into account that users may change their behavior over time, it might be advisable to favor more recent ratings over the ones that were cast further in the past. This can be achieved by including a survival factor λ controlling the speed at which old ratings are ”forgotten”. The definition updates to:

$$\rho^t(Z) = \sum_{X \in C} \lambda^{t-t_R} \rho^X_{Z,t_R} \tag{12}$$

where $0 \leq \lambda \leq 1$ and t is the current time. The reputation score of the agent Z at time t is defined as follows:

$$R^t(Z) = E[beta(\rho^t(Z))] = \frac{(r+1)}{(r+s+2)} \tag{13}$$

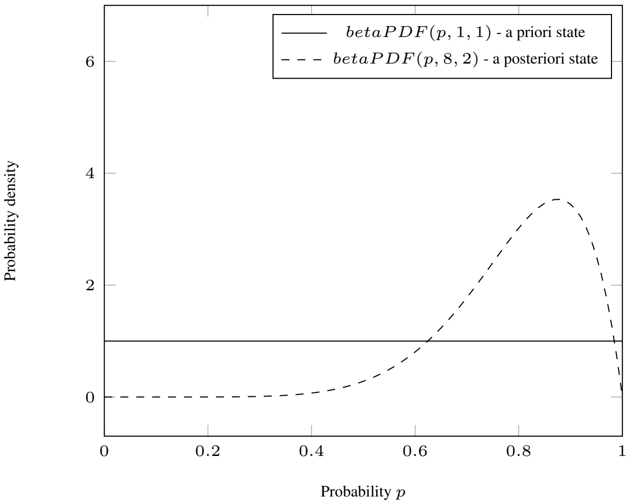


Fig. 12: $beta$ PDF a priori / a posteriori state

The pseudocode of the rating function is presented below:

```
$C$ is the set of all evaluators
$F$ is the set of all assumed fair raters
$Z$ is the evaluated agent
$F \leftarrow C$
**while** F changes **do**
  $\rho^t(Z) \leftarrow \sum_{X \in F} \rho^t(X)$
  $R^t(Z) \leftarrow E(\rho^t(Z))$
  **for** rate R in F **do**
    $f \leftarrow$ beta($\rho^t(R, Z)$)
    $l \leftarrow$ q quantile of f
    $u \leftarrow$ 1-q quantile of f
    **if** $l < R^t(Z)$ or $u < R^t(Z)$ **then**
      F $\leftarrow \backslash R$
    **end if**
  **end for**
**end while**
return $R^t(Z)$
```

The flexibility and robustness of this algorithm is ensured by variable distributions, rather than by a fixed threshold. If the spread of ratings from all reviewers is wide, then it will tend not to reject individual evaluators. If a rating vector $\rho = [r, s]$ is frequent among reviewers (e.g. 85% positive, 15% negative ratings) except for one reviewer (e.g. 50% positive, 50% negative ratings), then the exceptional rating will be rejected. The algorithm’s sensitivity can be increased or decreased by modifying the q parameter.

X. Security

TTYH utilizes cryptographic methods aiming to establish a secure environment that is hardened against malicious attacks or any arbitrary misuse of the system by either external or internal entities. Cryptography assumes that any protocol execution is subject to permanent aggression from malicious actors that are striving to obtain information about the system and users, tamper with data, or get unprivileged access to any resource.

From a security perspective, the fundamental requirements which must be addressed by any system are privacy and correctness. Private data must be kept undisclosed to any unauthorized actor. The successful execution of a protocol should produce correct, consistent, and predictable outputs for all participants.

The main security features that any decentralized system should possess are:

- Privacy - there should be no party learning, accessing, or modifying information that is not intended for that particular party;
- Correctness - it must be guaranteed that each party receives the correct output from the execution of the algorithms;
- Independence of Inputs - the inputs of the used algorithms must be independent between the honest and the corrupt parties;
- Guaranteed Output Delivery - this property states the resistance of the system to a denial of service attacks;
- Fairness - all of the parties should receive the output of the algorithms, regardless of their behavior (fair or malicious).

A. Permission Levels

TTYH transparently regulates interactions between multiple stakeholders, but also ensures privacy and protects personal information, intellectual property, and other sensitive data. In order to preserve confidentiality and security, Openfabric implements an access control mechanism over the DLT based on smart contracts. It uses similar approaches to the one from [98] (implements a GDPR-compliant personal data management scheme in a decentralized environment) and in [99] (enhances privacy at the blockchain level), where the Access Control List (ACL) is implemented by a smart contract.

Access control flow in TTYH :

- Resource submission (e.g. algorithm from an innovator). The platform checks the requester certificate, generates encryption keys, and stores the encrypted resource off-chain. The ACL smart contract is updated to grant the requester full access to their resource.
- ACL update. Resource permissions can be modified only by the resource owner; when some other contracts are executed, the owner can decide to grant/revoke access to their resources.
- Resource retrieval. The requester interrogates the ACL smart contract. The smart contract will verify whether the requester has the necessary permissions to access the specified resource, and will grant access to it in that case.

B. Identity Management

Currently, various privacy and security regulations try to protect the user's identity, reduce identity fraud, and strengthen online anonymity. The state of the art in identity management is the Self-Sovereign Identity (SSI) [100] model, which aims to ensure the user's control and sole ownership over their digital identity. To implement SSI, the majority proposed solution is based on decentralized technology [101] [102] [103]. There is a trade-off between using the ideal self-sovereign identity model - in which users are the exclusive owners of all their data - and having users' public attributes continually available, even when they are offline.

The TTYH ecosystem uses an identity management scheme based on the Public Key Infrastructure (PKI). Personal information is encrypted with the platform's key (E), which is uniquely generated for each case separately. Fig. 13 describes stakeholders or assets identification and how the payload regarding their identity is stored. In TTYH , the pointers to the off-chain data (H) are encrypted in a manner similar to the one described in [86]. The following steps should be performed for user registration:

- The user creates a pair of secret / public keys (sk, Pk) on their local device. The secret key should never leave the client's device. The user's public key (PK) represents his / her identification across the system.
- After the local key generation step, the user requests new certificates from the KYC / CA module. Certificates are used for granting and revoking access to the platform.
- The user demands an encryption key from the DOS. The DOS prompts the Secret Store to produce a pair of keys, and the newly-generated secret key will remain in the Secret Store, split among the Store's nodes - whereas the public key (E) is forwarded to the user. On chain, the ACL is updated to link the user to this encryption key.
- The user encrypts information with the Secret Store encryption key (E), then makes a request to the DOS for storing this information. The DOS persists the payload off-chain, and on-chain, it stores a tuple with the following information: user's public key (Pk), encryption public key (E), resource pointer (hash/address (H) of the resource), and - optionally - an identifier (required for asset identification).

The asset (algorithms, datasets) identification model uses a design similar to the stakeholder model, with the following difference: it does not need to generate a pair of keys (sk, Pk), because assets will use the author's keys. Identifiers for the assets are unique across the system, and will be regenerated with each update.

C. Remote Attestation

Remote attestation [77] allows a server (Infrastructure Provider Enclave) to convince the others that the software is secure and that it is running inside an up-to-date SGX enclave. Successful attestation assures the requester of the software's identity, and also informs about possible software tampering. For Intel SGX, there are currently two supported types of remote attestation: Intel SGX attestation service based on Intel Enhanced Privacy ID (Intel EPID), and ECDSA-based attestation based on Intel SGX DCAP.

The first solution uses Intel EPID provisioning services, and requires that the platform has internet access. In contrast, the second one uses a custom implementation for SGX attestation, either because internet-based services cannot be accessed, or because the attestation process should remain in-house.

D. *Key Management in* TTYH

Managing keys across a decentralized system is an endeavor that each decentralized platform should handle, if private processing of data is involved. Most centralized solutions use specialized hardware for secure storage (HSM) or custom implementations that require a trusted authority - but for the decentralized platforms, there are fewer solutions. Ethereum, Ocean Protocol [26] use the Secret Store [104] for key management, which is based on threshold cryptography and secret sharing schemes. TTYH uses a model akin to the Secret Store for managing the platform's keys. Fig. 14 describes the platform key generation process. The actor (any stakeholder, smart contract, or other entity) generates an ID, and requests a new key for it, specifying the key threshold. The threshold (t) represents the maximum number of nodes that cannot reconstruct the key. The actor signs this request with their secret key (A_{Sk}). Using the ECDKG [105] algorithm, a new key is generated, and each peer from the Secret Store will receive a fragment of the key. Any node belonging to the Secret Store can compute the public key (S_{PK}) related to the shared key.

The process of retrieving the key from the Secret Store is described in Fig. 15. TTYH has a permission mechanism that performs Secret Store key access control. Any actor who requests a key for a specific identifier (ID) signs the request with their private key (A_{Sk}). Any of the Secret Store's nodes can check permissions for the actor's key and selected ID; if the approval is declined, the actor will not receive the key. Otherwise, the Secret Store network computes the key (at least $t + 1$ peers from the network are required, where t is the threshold of the key), and it then encrypts the key, along with the public key of the actor (A_{Pk}).

The distributed verifiable secret sharing scheme is a mechanism which ensures the generation, storage, and retrieval of keys in a decentralized context. ECDKG [105] represents an enhancement of the DF-VSS [106] that can tolerate halting, eavesdropping, static malicious, replay, and adaptive adversaries. Each involved party chooses its random polynomials, computes and broadcasts shares, verifies shares from the other parties, and computes its private share of the secret. None of the parties can determine the secret by itself. To successfully reconstruct a secret, the Secret Store peers must collaborate and combine $t+1$ shares.

Elliptic curves represent a particular subset of mathematical equations of the following form: $y^2 = x^3 + ax + b$, and displaying some unique characteristics with regards to cryptographic operations. In the equation above, the coefficients a and b control curve behavior. Operations such as addition, multiplication by scalars, etc. applied on any curve point $T(x, y)$ will keep the results inside the group $(G, \bigoplus)$.

ECDKG algorithm

1) **Notation:** All arithmetic operations are done in a finite field $GF(q)$, G is the additive group derived from point T, and $\bigoplus$ is the addition operator over G. $\sum^{\bigoplus}$ represents point summation.
2) **Key distribution:**
   - **Setup:** For generating a second point T', each of the n parties chooses a random number $r \in GF(q)$, then sets:
$$T' = \sum_{i=1}^{n \bigoplus} r_i T \tag{14}$$
   - **Polynomials generation:** Each party p_i generates two random polynomials f_i and f'_i over GF of degree t.
$$f_i(z) = \sum_{k=0}^{t} a_{ik} z^k \ , f'_i(z) = \sum_{k=0}^{t} b_{ik} z^k \tag{15}$$
   - **Compute public shares:** Compute P_{ik} for $0 \leq k \leq t$. Broadcast them to all parties.
$$P_{ik} = (a_{ik}T) \bigoplus (b_{ik}T') \tag{16}$$
   - **Compute private shares:** Each p_i secretly shares the s_{ij} and s'_{ij} with p_j, for $j = 1, 2, \ldots, n$:
$$s_{ij} = f_i(p_j) \, mod \, p \ , s'_{ij} = f'_i(p_j) \, mod \, p \tag{17}$$
3) **Key verification:**
   - **Verify:** Every party p_j verifies the shares received from the other parties p_i.
$$(s_{ji}T) \bigoplus (s'_{ji}T') = \sum_{k=0}^{t \bigoplus} p_i^k P_{jk} \tag{18}$$
with $i = 1, 2, \ldots, n$. If validation fails for a specific index i the p_j party broadcasts complains against p_i.
   - **Dispute:** If more than t participants complain against p_i, then p_i is considered faulty, and is excluded. Otherwise, the p_i will reveal shares s_{ij} and s'_{ij} for each complainer. If formula (18) is not valid for all disputed shares, the p_i is disqualified, and will be excluded from further computations. Let Q_i be a set of all non-disqualified players.
4) **Key check:** If party p_j is found faulty, share s_i is updated by removing it from Q_i and recomputing the sum:
$$s_i = \sum_{j \in Q_i} s_{ji} \tag{19}$$
5) **Key generation:**
   - **Compute public key:** Each party can compute A_{i0} and broadcast it.
$$A_{i0} = a_{i0}T \tag{20}$$
Compute public key:
$$y_i = \sum_{j \in Q_i}^{\bigoplus} A_{j0} \tag{21}$$
   - **Secret share:** Each party their can compute their key share:
$$x_j = f_i(0)T \tag{22}$$
And public key share:
$$s_j = \sum_{i \in Q_j} s_{ij} \tag{23}$$
   - **Recover:** More than $t+1$ parties can recover the secret by using interpolation, as described in the following section.

E. Polynomial Interpolation over Elliptic Curves

Lagrangian interpolation

Given a set $\{(x_0, y_0), \ldots, (x_n, y_n)\}$ of points, the Lagrange polynomial F is the lowest degree polynomial which assumes that all previous points in the set belong on the function's plot.
$$F(x_i) = y_i, \text{ for } j = 0, \ldots, n. \tag{24}$$
Define the Lagrange basic polynomials:
$$L_{n,j}(x) = \prod_{k \neq j} \frac{x - x_k}{x_j - x_k}, k = 1, \ldots, n. \tag{25}$$
Compute the interpolation polynomial:
$$F(x) = \sum_{j=0}^{n} y_j L_{n,j}(x). \tag{26}$$

Secret sharing interpolation

Knowing $t+1$ points: $s_i = F(p_i)$ using Lagrangian interpolation, the coefficients of the F(x) can be uniquely identified. The authors from [105] consider the following polynomial for interpolation:
$$F(z) = \left(\sum_{k \in Q} a_{k0} \right) + \left(\sum_{k \in Q} a_{k1} \right) z + \cdots + \left(\sum_{k \in Q} a_{kt} \right) z^t \tag{27}$$
The shared secret is computed as $y = F(0)T$, wherein T is a generator of the group of the curve.

XI. Conclusion

A. Protocol Performance

The agility of the TTYH protocol represents one of the critical factors that affects the speed and the scalability of the platform. The cumulative platform performance is affected by multiple factors, such as execution environment, ontology-based data integration, data storage, and data transfer.

An essential platform component of significant impact on the overall performance is ontology-based data integration. The TTYH protocol was designed to be agnostic to the DLT implementation; and consequently, we performed simulations utilizing multiple DLT implementations.

Fig. 16 depicts the results of simulating multiple write and read operations using a dataset with ontology concepts containing up to 10,000 properties. The simulation results display the performance outputs for IPFS, Cosmos Tendermint, Hyperledger Fabric, and Etherum.

B. Conclusions

The TTYH ecosystem provides a novel foundation which is capable of sustaining the genuine revolution of artificial intelligence. TTYH 's mission is to nourish the required synergy uniting all relevant stakeholders, facilitate their interactions, and empower the creation and usage of intelligent algorithms with ease.

Securing intellectual property and stimulating fair competition among innovators are the key factors that coagulate large, vibrant, and collaborative communities. This aspect embodies the real catalyst that is driving the evolution of intelligent algorithm solutions. High quality, valuable, and reliable results require the support of an economic environment that covers innovator expenses through the monetization of their work. By satisfying the financial aspect, innovators can then dedicate their time and effort towards exploring, formulating, and creating elaborate solutions, and thus accelerating the ecosystem's growth.

TTYH lowers the adoption barrier by reducing the infrastructure demands and technical know-how required to utilize algorithms. This aspect empowers the end-users to operate with a new generation of intelligence-driven products and tools that are made accessible through the built-in peer-to-peer marketplace. The TTYH marketplace provides a uniform, intuitive, and simplified user experience, allowing for execution of AIs without the need to install, configure, or customize anything. It consolidates the business relationship between the supply-and-demand of services, between innovators, infrastructure providers, end-users, and businesses. Considering the fact that enterprise adoption of edge technologies is slow, expensive, and disruptive, TTYH provisions connectors that minimize the integration friction.

Motivated by the goal of decentralization, TTYH brings together the concepts of scalability and algorithm execu-tion. Any infrastructure provider that adheres to the requirements of the ecosystem will also take part in this endeavor. The trusted execution environment (TEE), in combination with distributed cryptographic keys management, creates a sandboxing environment for securing intellectual property,user-data privacy, and isolating execution hosts. Privacy is an essential attribute of TTYH , which stems from the fact that algorithms and datasets are decrypted only inside the TEE, so that neither the platform nor the executor have access to them.

The distributed ledger ensures undeniable contracts and unforgeable history between the platform's stakeholders. Furthermore, it is also the underlying layer for access control and identification mechanisms. The platform is orchestrated by a decentralized operating system (DOS) which manages network resources, services and processes, and coordinates the proper functioning of the system.

In TTYH , a Bayesian reputation model supervises the quality and performance of products and services through a reputation score that is computed based on community feedback. It also increases collaboration between participants in a safe environment, without having to rely on any centralized authority. Nash equilibrium is achieved when infrastructure providers offer excellent services, innovators generate high-quality algorithms for which the community is willing to pay, and service consumers efficiently combine algorithms to obtain solutions for their specific use cases.

C. Ongoing and Future Research

TTYH provides the technical solution, encouraging innovation to unravel its true potential through harnessing the power of communities. As a research project, TTYH pushes the boundaries of the current technological and scientific advancement, and the road ahead will continue to unravel exciting research and implementation challenges. The existing architecture has a flexible, future-proof design, and - as innovation is surely going to bring to light new technologies - some system components might be swapped with more suitable concepts (cryptography, secure computation, virtualization, etc.). The technical updates should not change the main project goal of creating an ecosystem in which algorithms are executed securely and participants are rewarded in accordance to their contribution. Further research can be done in order to improve the secure computation from the perspectives of performance and security. Other points of interest include integration with various platforms, as well as accommodating entire suites of algorithms in the TTYH ecosystem.

References

[1] E. Karvonen, *Informational Societies: Understanding the Third Industrial Revolution*. Tampere, Finland: Tampere University, 2001. I

[2] "Navigating the fourth industrial revolution to the bottom line," Report, Manufacturing Institute, PWC, 1997. I

[3] L. L. Halse and B. Jæger, "Operationalizing industry 4.0: Understanding barriers of industry 4.0 and circular economy," in *Advances in Production Management Systems. Towards Smart Production Management Systems*, F. Ameri, K. E. Stecke, G. von Cieminski, and D. Kiritsis, Eds. Cham: Springer International Publishing, 2019, pp. 135–142. I

[4] T. Economist. (2017, May) The world's most valuable resource is no longer oil, but data. [Online]. Available: https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data I

[5] A. Pavaloiu, "The impact of artificial intelligence on global trends," *Journal of Multidisciplinary Developments*, pp. 21–37, 12 2016. I

[6] "The impact of artificial intelligence on work. an evidence synthesis on implications for individuals, communities, and societies," ISBN: 978-0-85672-626-2., British Academy, The Royal Society, 2018. I

[7] A. D. C. Change, T. Blair, and R. Pachauri, *Avoiding dangerous climate change*. Cambridge University Press, 2006. I

[8] V. V. Quaschning, *Renewable energy and climate change*. John Wiley & Sons, 2019. I

[9] K. D. Miller, L. Nogueira, A. B. Mariotto, J. H. Rowland, K. R. Yabroff, C. M. Alfano, A. Jemal, J. L. Kramer, and R. L. Siegel, "Cancer treatment and survivorship statistics, 2019," *CA: a cancer journal for clinicians*, vol. 69, no. 5, pp. 363–385, 2019. I

[10] "Tech trends. beyond the digital frontier," Article, Deloitte insights, 2019. I

[11] "The national artificial intelligence research and development strategic plan." Article, USA - Select Committee On Artificial Intelligence Of The National Science and Technology Council, 2019. I

[12] "A european approach to artificial intelligence," Report, European Commission, 2019. I

[13] D. A. Ferrucci, "Introduction to "this is watson"," *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 1:1–1:15, 2012. II-A

[14] (2019, January) Ibm watson. enterprise ready . [Online]. Available: https://www.ibm.com/watson/about II-A

[15] Google platform. [Online]. Available: https://cloud.google.com/ -platform/ II-A

[16] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, omega, and kubernetes," *Commun. ACM*, vol. 59, no. 5, p. 50–57, Apr. 2016. [Online]. Available: https://doi.org/10.1145/2890784 II-A

[17] I. Guyon, L. Sun-Hosoya, M. Boullé, H. J. Escalante, S. Escalera, Z. Liu, D. Jajetic, B. Ray, M. Saeed, M. Sebag *et al.*, "Analysis of the automl challenge series," *Automated Machine Learning*, p. 177, 2019. II-A

[18] M. Salvaris, D. Dean, and W. H. Tok, "Microsoft platform," in *Deep Learning with Azure*. Springer, 2018, pp. 79–98. II-A

[19] A. Del Sole, "Introducing microsoft cognitive services," in *Microsoft Computer Vision APIs Distilled*. Springer, 2018, pp. 1–4. II-A

[20] "Machine learning lens," aws, Amazon, 2020. [Online]. Available: https://d1.awsstatic.com/whitepapers/architecture/wellarchitected-Machine-Learning-Lens.pdf II-A

[21] SingularityNET, "Singularitynet a decentralized, open market and network for ais," 2019. [Online]. Available: https://public.singularitynet.io/whitepaper.pdf II-B, IV-B

[22] "Singularity studio," Extension, SingularityNET, 2020. [Online]. Available: https://public.singularitynet.io/whitepaper.pdf II-B

[23] "Neo ecosystem," Website, NEO, 2020. [Online]. Available: https://neo.org/ II-B

[24] "Eos, the next-generation, open-source blockchain protocol with industry-leading transaction speed and flexible utility," Website, EOS, 2020. [Online]. Available: https://eos.io/ II-B

[25] "Effect ," Website, Effect , 2018. [Online]. Available: https://effect. /download/effect_whitepaper.pdf II-B

[26] O. P. Foundation, B. GmbH, and N. Circus, "Ocean protocol:a decentralized substrate for data & services - technical whitepaper," 2019. [Online]. Available: https://oceanprotocol.com/tech-whitepaper.pdf II-B, X-D

[27] C. DeepBrain, "Deepbrain chain artificial intelligence computing platform driven by blockchain," *Whitepaper*, 2019. [Online]. Available: https://deepbrainchain.org/assets/pdf/DeepBrainChainWhitepaper_en.pdf II-B

[28] C. Zednik, "Solving the black box problem: A normative framework for explainable artificial intelligence," *Springer Philosophy & Technology*, 2019. II-B

[29] T. N. Ltd, "Thought white paper," 2018. [Online]. Available: https://thought.live/assets/thought-white-paper.pdf II-B

[30] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf III-A

[31] K. Kaneiwa and R. Mizoguchi, "Distributed reasoning with ontologies and rules in order-sorted logic programming," *SSRN Electronic Journal*, 2009. [Online]. Available: https://doi.org/10.2139/ssrn.3199422 III-A

[32] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006. III-A

[33] U. C. Schreiber and C. Mayer, *The First Cell*. Springer International Publishing, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-45381-7 III-A, III-A

[34] W. J. Yeager and R. Y. Chen, "Distributed trust mechanism for decentralized networks," May 22 2007, uS Patent 7,222,187. III-A

[35] R. Aumann and A. Brandenburger, "Epistemic conditions for nash equilibrium," *Econometrica: Journal of the Econometric Society*, pp. 1161–1180, 1995. III-B

[36] E. Maskin, "Nash equilibrium and welfare optimality," *The Review of Economic Studies*, vol. 66, no. 1, pp. 23–38, 1999. III-B, IV-C

[37] A. Haeberlen, A. Mislove, and P. Druschel, "Glacier: Highly durable, decentralized storage despite massive correlated failures," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, 2005, pp. 143–158. III-E

[38] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-based, decentralized access control for ipfs," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 1499–1506. III-E

[39] Q. Zheng, Y. Li, P. Chen, and X. Dong, "An innovative ipfs-based storage model for blockchain," in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 2018, pp. 704–708. III-E

[40] O. López-Pintado, L. García-Bañuelos, M. Dumas, and I. Weber, "Caterpillar: A blockchain-based business process management system." in *BPM (Demos)*, 2017. III-E

[41] Y. Lu, “The blockchain: State-of-the-art and research challenges,” *Journal of Industrial Information Integration*, vol. 15, pp. 80–90, 2019. III-E

[42] F. Baiardi and M. Vanneschi, “Design of highly decentralized operating systems,” in *Distributed Operating Systems*, Y. Paker, J.-P. Banatre, and M. Bozyiğit, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 113–145. III-F

[43] A. S. Tanenbaum and R. Van Renesse, “Distributed operating systems,” *ACM Comput. Surv.*, vol. 17, no. 4, p. 419–470, Dec. 1985. [Online]. Available: https://doi.org/10.1145/6041.6074 III-F

[44] J. Fitzer, P. F. Menna, F. Musse, and S. Seif, “Kubernetes as a distributed operating system for multitenancy/multiuser,” Jun. 11 2020, uS Patent App. 16/216,602. III-F

[45] O. A. Ismael and H. Tews, “Secure communications between peers using a verified virtual trusted platform module,” Mar. 17 2020, uS Patent 10,592,678. III-F

[46] A. Butean, E. Pournaras, A. Tara, H. Turesson, and K. Ivkushkin, “Dynamic consensus: Increasing blockchain adaptability to enterprise applications,” in *Applied Informatics and Cybernetics in Intelligent Systems*, R. Silhavy, Ed. Cham: Springer International Publishing, 2020, pp. 433–442. III-F, VII-C

[47] J. Benet, “Ipfs-content addressed, versioned, p2p file system,” *arXiv preprint arXiv:1407.3561*, 2014. III-F, VI-B

[48] R. Kurzweil, “The law of accelerating returns,” in *Alan Turing: Life and legacy of a great thinker*. Springer, 2004, pp. 381–416. IV

[49] C. F. Camerer, *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press, 2011. IV

[50] X. Kong, Q. Xu, and T. Zhu, “Dynamic evolution of knowledge sharing behavior among enterprises in the cluster innovation network based on evolutionary game theory,” *Sustainability*, vol. 12, no. 1, p. 75, 2020. IV

[51] S. F. Mjølsnes and C. Rong, “On-line e-wallet system with decentralized credential keepers,” *Mobile Networks and Applications*, vol. 8, no. 1, pp. 87–99, 2003. IV-A

[52] A. Permenev, D. Dimitrov, P. Tsankov, D. Drachsler-Cohen, and M. Vechev, “Verx: Safety verification of smart contracts,” in *2020 IEEE Symposium on Security and Privacy, SP*, 2020, pp. 18–20. IV-B

[53] C. Egger, P. Moreno-Sanchez, and M. Maffei, “Atomic multi-channel updates with constant collateral in bitcoin-compatible payment-channel networks,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 801–815. IV-B

[54] C. Lin, N. Ma, X. Wang, and J. Chen, “Rapido: Scaling blockchain with multi-path payment channels,” *Neurocomputing*, 2020. IV-B

[55] C. Burchert, C. Decker, and R. Wattenhofer, “Scalable funding of bitcoin micropayment channel networks,” *Royal Society open science*, vol. 5, no. 8, p. 180089, 2018. IV-B

[56] C. W. Bach and A. Perea, “Generalized nash equilibrium without common belief in rationality,” *Economics Letters*, vol. 186, p. 108526, 2020. IV-C

[57] A. Ozment and S. E. Schechter, “Bootstrapping the adoption of internet security protocols.” in *WEIS*, 2006. IV-C

[58] Y. Lv and T. Moscibroda, “Fair and resilient incentive tree mechanisms,” *Distributed Computing*, vol. 29, no. 1, pp. 1–16, 2016. IV-C

[59] G. Pickard, W. Pan, I. Rahwan, M. Cebrian, R. Crane, A. Madan, and A. Pentland, “Time-critical social mobilization,” *Science*, vol. 334, no. 6055, pp. 509–512, 2011. IV-C

[60] J. R. Douceur and T. Moscibroda, “Lottery trees: motivational deployment of networked systems,” in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, 2007, pp. 121–132. IV-C

[61] A. Tara, A. Butean, C. Zamfirescu, and R. Learney, “An ontology model for interoperability and multi-organization data exchange,” in *Artificial Intelligence and Bioinspired Computational Methods*, R. Silhavy, Ed. Cham: Springer International Publishing, 2020, pp. 284–296. V-A, V-B, V-C

[62] P. Hofferer, “Achieving business process model interoperability using metamodels and ontologies,” 2007. V-A, VIII-B

[63] O. Banouar and S. Raghay, “Interoperability of information systems through ontologies: State of art,” *International Journal of Computer Science and Information Security*, vol. 14, no. 8, p. 392, 2016. V-A

[64] X. Wang, H. Zhao, and J. Zhu, “Grpc: A communication cooperation mechanism in distributed systems,” *SIGOPS Oper. Syst. Rev.*, vol. 27, no. 3, p. 75–86, Jul. 1993. [Online]. Available: https://doi.org/10.1145/155870.155881 V-C

[65] P. Sharma, L. Chaufournier, P. Shenoy, and Y. C. Tay, “Containers and virtual machines at scale: A comparative study,” in *Proceedings of the 17th International Middleware Conference*, ser. Middleware ’16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2988336.2988337 VI

[66] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh, “Terra: A virtual machine-based platform for trusted computing,” in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, 2003, pp. 193–206. VI

[67] “The speed of containers, the security of vms,” accessed: 15 May 2020. [Online]. Available: https://katacontainers.io VI

[68] “A container sandbox runtime focused on security, efficiency, and ease of use,” accessed: 15 May 2020. [Online]. Available: https://gvisor.dev VI

[69] V. Costan and S. Devadas, “Intel sgx explained.” *IACR Cryptol. ePrint Arch.*, vol. 2016, no. 86, pp. 1–118, 2016. VI-A

[70] S. Sultan, I. Ahmad, and T. Dimitriou, “Container security: Issues, challenges, and the road ahead,” *IEEE Access*, vol. 7, pp. 52 976–52 996, 2019. VI-A

[71] S. Shinde, D. Le Tien, S. Tople, and P. Saxena, “Panoply: Low-tcb linux applications with sgx enclaves.” in *NDSS*, 2017. VI-A, VI-A

[72] A. Baumann, M. Peinado, and G. Hunt, “Shielding applications from an untrusted cloud with haven,” *ACM Transactions on Computer Systems (TOCS)*, vol. 33, no. 3, pp. 1–26, 2015. VI-A

[73] S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O’keeffe, M. L. Stillwell *et al.*, “{SCONE}: Secure linux containers with intel {SGX},” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 689–703. VI-A

[74] C.-C. Tsai, D. E. Porter, and M. Vij, “Graphene-sgx: A practical library {OS} for unmodified applications on {SGX},” in *2017 {USENIX} Annual Technical Conference ({USENIX}{ATC} 17)*, 2017, pp. 645–658. VI-A

[75] D. Merkel, “Docker: lightweight linux containers for consistent development and deployment,” *Linux journal*, vol. 2014, no. 239, p. 2, 2014. VI-A

[76] T. Knauth, M. Steiner, S. Chakrabarti, L. Lei, C. Xing, and M. Vij, “Integrating remote attestation with transport layer security,” *arXiv preprint arXiv:1801.05863*, 2018. VI-B

[77] S. Johnson, V. Scarlata, C. Rozas, E. Brickell, and F. Mckeen, “Intel® software guard extensions: Epid provisioning and attestation services,” *White Paper*, vol. 1, no. 1-10, p. 119, 2016. VI-B, X-C

[78] A. Bessani, J. Sousa, and E. E. Alchieri, “State machine replication for the masses with bft-smart,” in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2014, pp. 355–362. VII-C

[79] M. Castro and B. Liskov, “Practical byzantine fault tolerance and proactive recovery,” *ACM Trans. Comput. Syst.*, vol. 20, no. 4, p. 398–461, Nov. 2002. [Online]. Available: https://doi.org/10.1145/571637.571640 VII-C, VII-C

[80] P. Thakkar, S. Nathan, and B. Viswanathan, “Performance benchmarking and optimizing hyperledger fabric blockchain platform,” in *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 2018, pp. 264–276. VII-C

[81] J. Li, N. Li, J. Peng, H. Cui, and Z. Wu, “Energy consumption of cryptocurrency mining: A study of electricity consumption in mining cryptocurrencies,” *Energy*, vol. 168, pp. 160–168, 2019. VII-C

[82] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, p. 382–401, Jul. 1982. [Online]. Available: https://doi.org/10.1145/357172.357176 VII-C

[83] P. Regulation, “Regulation (eu) 2016/679 of the european parliament and of the council,” *Official Journal of the European Union*, 2016. VII-D

[84] M. Finck, “Blockchains and data protection in the european union,” *Eur. Data Prot. L. Rev.*, vol. 4, p. 17, 2018. VII-D, VII-D

[85] G. Kondova and J. Erbguth, “Self-sovereign identity on public blockchains and the gdpr,” in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 342–345. VII-D

[86] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, “Gdpr-compliant personal data management: A blockchain-based solution,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1746–1761, 2019. VII-D, X-B

[87] F. Dai, Q. Mo, Z. Qiang, B. Huang, W. Kou, and H. Yang, “A choreography analysis approach for microservice composition in cyber-physical-social systems,” *IEEE Access*, vol. 8, pp. 53 215–53 222, 2020. VIII

[88] A. Marrella and M. Mecella, “Cognitive business process management for adaptive cyber-physical processes,” in *International Conference on Business Process Management*. Springer, 2017, pp. 429–439. VIII

[89] I. Nadareishvili, R. Mitra, M. McLarty, and M. Amundsen, *Microservice Architecture: Aligning Principles, Practices, and Culture*, 1st ed. O’Reilly Media, Inc., 2016. VIII

[90] P. Poizat and G. Salaün, “Checking the realizability of bpmn 2.0 choreographies,” in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC ’12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1927–1934. [Online]. Available: https://doi.org/10.1145/2245276.2232095 VIII-B

[91] R. Song, J. Vanthienen, W. Cui, Y. Wang, and L. Huang, “Context-aware bpm using iot-integrated context ontologies and iot-enhanced decision models,” in *2019 IEEE 21st Conference on Business Informatics (CBI)*, vol. 1. IEEE, 2019, pp. 541–550. VIII-B

[92] A. Fuller, Z. Fan, C. Day, and C. Barlow, “Digital twin: Enabling technologies, challenges and open research,” *IEEE Access*, vol. 8, pp. 108 952–108 971, 2020. VIII-C

[93] Y. Lu, C. Liu, I. Kevin, K. Wang, H. Huang, and X. Xu, “Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101837, 2020. VIII-C

[94] F. Cornelli, E. Damiani, S. D. C. Di Vimercati, S. Paraboschi, and P. Samarati, “Choosing reputable servents in a p2p network,” in *Proceedings of the 11th international conference on World Wide Web*, 2002, pp. 376–386. IX

[95] A. Josang, “Trust-based decision making for electronic transactions,” in *Proceedings of the Fourth Nordic Workshop on Secure Computer Systems (NORDSEC’99)*. Citeseer, 1999, pp. 496–502. IX

[96] C. Dellarocas, “Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior,” in *Proceedings of the 2nd ACM conference on Electronic commerce*, 2000, pp. 150–157. IX

[97] A. Whitby, A. Jøsang, and J. Indulska, “Filtering out unfair ratings in bayesian reputation systems,” in *Proc. 7th Int. Workshop on Trust in Agent Societies*, vol. 6, 2004, pp. 106–117. IX

[98] C. Wirth and M. Kolain, “Privacy by blockchain design: a blockchain-enabled gdpr-compliant approach for handling personal data,” in *Proceedings of 1st ERCIM Blockchain Workshop 2018*. European Society for Socially Embedded Technologies (EUSSET), 2018. X-A

[99] G. Zyskind, O. Nathan *et al.*, “Decentralizing privacy: Using blockchain to protect personal data,” in *2015 IEEE Security and Privacy Workshops*. IEEE, 2015, pp. 180–184. X-A

[100] C. Allen, “The path to self-sovereign identity,” *Life with Alacrity*, 2016. X-B

[101] A. Tobin and D. Reed, “The inevitable rise of self-sovereign identity,” *The Sovrin Foundation*, vol. 29, no. 2016, 2016. X-B

[102] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, “Uport: A platform for self-sovereign identity,” *URL: https://whitepaper. uport. me/uPort_ whitepaper_DRAFT20170221. pdf*, 2017. X-B

[103] A. Ebrahimi, “Identity management service using a blockchain providing certifying transactions between devices,” Aug. 1 2017, uS Patent 9,722,790. X-B

[104] “Secret store,” accessed: 13 Aug 2020. [Online]. Available: https://openethereum.github.io/wiki/Secret-Store X-D

[105] C. Tang, “Ecdkg: A distributed key generation protocol based on elliptic curve discrete logarithm,” *sE· CURECOMM*, pp. 353–364, 2005. X-D, X-D, X-E

[106] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, “Secure distributed key generation for discrete-log based cryptosystems,” *Journal of Cryptology*, vol. 20, no. 1, pp. 51–83, 2007. X-D