UvA-DARE (Digital Academic Repository)

The ARTICONF approach to decentralized car-sharing

Saurabh, N.; Rubia, C.; Palanisamy, A.; Koulouzis, S.; Sefidanoski, M.; Chakravorty, A.; Zhao, Z.; Karadimce, A.; Prodan, R.

The ARTICONF approach to decentralized car-sharing

Nishant Saurabh [a], Carlos Rubia [b], Anandakumar Palanisamy [c], Spiros Koulouzis [d], Mirsat Sefidanoski [e], Antorweep Chakravorty [c,f], Zhiming Zhao [d], Aleksandar Karadimce [e], Radu Prodan [a,*]

[a] *Institute of Information Technology, University of Klagenfurt, Kärnten 9020, Austria*
[b] *Agilia Center, Aljaraque, 21110, Spain*
[c] *bitYoga AS, Stavanger, 4021, Norway*
[d] *Institute for Informatics, University of Amsterdam, Amsterdam, 1012 WX, Netherlands*
[e] *University of Information Science and Technology Ohrid, North Macedonia*
[f] *University of Stavanger, stavanger, 4021, Norway*

ARTICLE INFO

ABSTRACT

Social media applications are essential for next-generation connectivity. Today, social media are centralized platforms with a single proprietary organization controlling the network and posing critical trust and governance issues over the created and propagated content. The ARTICONF project funded by the European Union's Horizon 2020 program researches a decentralized social media platform based on a novel set of trustworthy, resilient and globally sustainable tools that address privacy, robustness and autonomy-related promises that proprietary social media platforms have failed to deliver so far. This paper presents the ARTICONF approach to a car-sharing decentralized application (DApp) use case, as a new collaborative peer-to-peer model providing an alternative solution to private car ownership. We describe a prototype implementation of the car-sharing social media DApp and illustrate through real snapshots how the different ARTICONF tools support it in a simulated scenario.

1. Introduction

Social media platforms are key technologies for next-generation connectivity [1,2]. They have the potential to shape and mobilize patterns of communication, practices of exchange and business, creation, learning and knowledge acquisition. Typically, social media are centralized platforms with a single proprietary organization controlling the network and posing critical issues of trust and governance over created and propagated content. This is particularly problematic when data breaches are a regular phenomenon at the hands of centralized intermediaries. Addressing this problem requires innovative solutions at the user level (i.e. consumers, prosumers, businesses) and the underlying social media environment. This facilitates global reach, improved trust, and decentralized control and ownership [3,4].

1.1. Decentralized social media

The ARTICONF project [5] [1] funded by the European Union's Horizon 2020 program proposes to research and develop a novel decentralized social media platform based on a novel set of trustworthy, resilient, and globally sustainable tools. ARTICONF addresses issues of trust, time-criticality and democratization for a new generation of federated infrastructure with the following goals:

- Simplify the creation of decentralized applications (DApps) using a two stage permissioned blockchain architecture;
- Automatically detect interest groups and communities using semantic contextualization and abstraction [6] of dynamic, diverse social media facets;

- Elastically auto-scale time-critical social media DApps through an adaptive orchestrated cloud infrastructure meeting their runtime requirements;
- Enhance monetary inclusion in collaborative models through cognitive and interactive visualization.

1.2. Car-sharing

Car-sharing is a new collaborative model providing an alternative solution to private car ownership. This model allows customers to temporarily use a vehicle (on-demand) at a variable fee, charged depending on the distance traveled or time used. This sharing economy example, which can be business-to-business (B2B) or business-to-consumer (B2C), intends to satisfy transportation demand in a sustainable way by lowering the CO_2 emissions in cities through:

- Fewer vehicles thanks to a combined car pooling and car-sharing approach;
- The use of electric cars that require destinations closer to the charging stations, which facilitates planning;
- Reduced vehicle lifetime (below the 13-year average), which encourages the faster introduction of eco-friendly innovations on the market;
- Reduced traffic and parking congestions.

Current car-sharing solutions rely on a central entity that operates the platform and charges customers (e.g. escrow, deposit) in a non-transparent manner. This entity unilaterally decides about the service level agreement (SLA), prices and penalties without considering other users' opinions, contents or reputation. In the absence of transparency and market competition, this can lead to high prices and even monopoly.

We aim to overcome this problem by shifting the focus from a B2B and B2C model to a more traditional setting (resembling the "good old market" from centuries ago), when business happened in a social and direct consumer-to-consumer (C2C) fashion. We propose a new car-sharing model based on a decentralized social network implemented as a DApp, which builds a community of car owners and fleet managers that jointly operate a pool or cars, rented to passengers in a transparent and peer-to-peer (P2P) fashion based on time, route and distance without central intermediaries. The success of the new model relies on a community per city, where users' trust and reputation are the basis for any successful business transaction, and the prices directly depend on these values. Our proposed model collects information from different sources, including vehicle devices, user ratings, user contents, and their past behavior to evaluate the user reputation. The platform evaluates the SLA by decentralized consensus, which has a major effect on the new interactions within the community. For instance, a user with a good reputation can rent a car without escrow or with a better price, and her content ratings in the social network have more trust than the ones coming from users with a poor reputation. The validity of the model relies on mutual trust and has an impact the community growth.

1.3. Outline

We describe the ARTICONF architecture applied to a car-sharing DApp use case, as a new collaborative model [7] and alternate software-as-a-service solution to private car ownership. The new car-sharing model allows customers to temporarily rent a vehicle on-demand and at a variable fee, charged depending on the travel distance or used time. This paper extends our initial work [8] with a related comparison and further implementation details of the car-sharing DApp and its benefits from using the ARTICONF social media platform.

The paper has six sections. The next section outlines the ARTICONF decentralized social media platform, with focus on its modular tool architecture and its DApp use cases. Section 3 describes the design of the car-sharing DApp use case on top of the ARTICONF platform. Section 4

gives implementation details of the car-sharing DApp in a simulated use-case scenario. Finally, Section 5 summarizes the related work and Section 6 concludes the paper.

2. ARTICONF architecture for DApp development

To provide a high level of robustness and autonomy, while addressing issues of trust, time-criticality and democratization, the ARTICONF shares a decentralized architecture with four building block tools:

- Trust and Integration Controller (TIC);
- Co-located and Orchestrated Network Fabric (CONF);
- Semantic Model with self-adaptive and Autonomous Relevant Technology (SMART);
- Tools for Analytics and Cognition (TAC).

Fig. 1 shows CONF as the initial entry tool, in charge of deploying and scaling the entire platform when necessary. On the other hand, the TIC tool is the backbone for use case blockchain services that instantiates a hyperledger fabric-based network [9] with two alternative blockchain access modes: a standard fabric client SDK or an enhanced adapter provided by TIC itself. SMART interactively guides social media consumers and providers to cooperatively support the behavior of the DApp, decision-making and infrastructure, while TIC provides the underlying permissioned blockchain that verifies and stores the pseudonymous of the user transactional activities and associated personal data in an encrypted form. Finally, TAC gives intelligent insights and relevant information about the platform to use case providers and end-users through an interactive dashboard. TIC tool initializes a SMART user with each instantiated hyperledger fabric-network to handle access permissions of end-user data required by the SMART and TAC tools. Essentially, the SMART user initiates a request to end-users on behalf of SMART and TAC tool for accessing encrypted data associated with their transactional activities. Upon getting the access permissions, SMART and TAC tools utilize the data for gathering and visualizing intelligent insights. This allows ARTICONF to analyze transactional activities of end users in a decentralized manner without violating privacy principles.

2.1. ARTICONF integrated tools

This section outlines the services of the four tools.

2.1.1. TIC

TIC provides support for creating and integrating fragmented DApp platforms with increased participation using an open application programming interface (API) based on a decentralized blockchain network. TIC allows customers, consumers, prosumers and businesses to engage in a safe, transparent and trustful interaction with monetization opportunities using four microservices.

Permissioned consortium blockchain is a fundamental service that allows actors in the network to develop consistency, accountability and traceability of their shared data.

Cloud storage of distributed large shared data items, such as copies of the blockchain transactions, ensures the same version of truth with efficient indexing and traversals.

Relationship system is a Turing-complete programmable unit of the blockchain that allows users to define conditions on data sharing and use rights through smart contracts. This enables them complete control over their content.

Certificate authority is a client software that manages user groups, securely shares the keys with their members, keeps records of shared keys, and encrypts shared data before broadcasting and storing it on the blockchain and the Cloud.

2.1.2. CONF

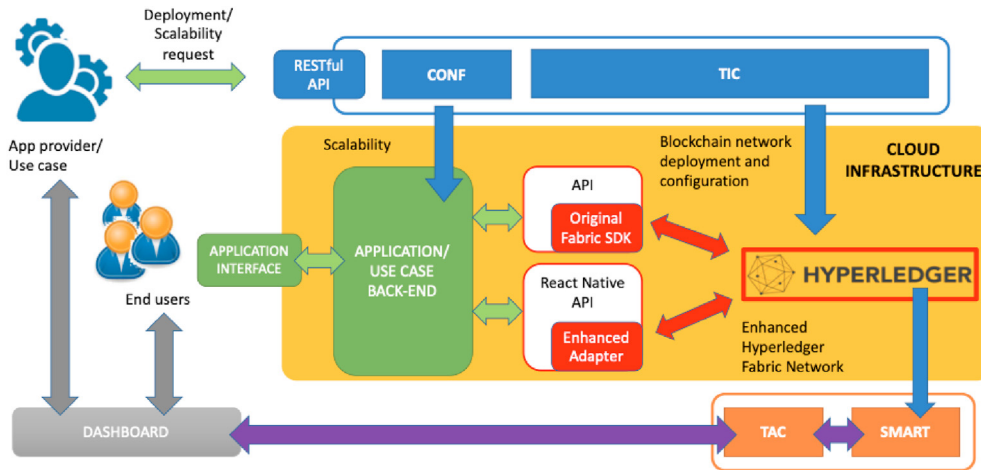CONF provides effective customization and provisioning of virtual

Fig. 1. ARTICONF integrated tool architecture for decentralized applications (DApps). TIC: Trust and Integration Controller; CONF: Co-located and Orchestrated Network Fabric; SMART: Semantic Model with self-adaptive and Autonomous Relevant Technology; TAC: Tools for Analytics and Cognition.

infrastructures based on specific DApp requirements. It also automates the DApp deployment, monitoring and adaption at runtime. The functional components of CONF follow a microservices design, connected via a message bus with a common interface to the other components. CONF includes four microservices.

Infrastructure planner provides functionality for DApp microservices to handle the time-critical constraints based on algorithms like critical path. The planner selects virtual machines from available data centers, and customizes their capacity and topology based on the performance requirements and cost.

Provisioning agent automates the infrastructure plans created by the planner onto the selected cloud data centers. The provisioning agent effectively decomposes the infrastructure description and provisions it in parallel across multiple cloud data centers with transparent configuration on the network topology.

Deployment agent automates the deployment of DApp microservices onto the virtual infrastructure provisioned by the provisioning agent by considering the quality constraints of the DApp and of the blockchain services.

Monitoring and control agent monitors the specific metrics of the cloud infrastructure and DApp network at runtime. In consequence, it takes effective actions to adapt them, such as scaling DApp microservices or cloud infrastructures, or adapting the network topologies based on the runtime status diagnosis.

2.1.3. SMART

SMART is a data-driven tool, capable of finding relevant interest groups through democratic and tokenized decision-making and reputation mechanisms, solving disputes in collaborative models and preserving the trustful and autonomous users-centric environment. SMART provides five microservices.

Semantic framework for federated social media involves large-scale entities at three levels of abstraction based on a conceptualization model:

- Concrete perception of the users and of the associated smart objects in a global domain;
- Structure of the perceived relationships;
- Communication among entities, exploiting decentralized reasoning and relevant communities with interest groups.

Autonomous and adaptive user-centric model encompasses role-stage programming techniques and a human-agent collective model describing, reasoning and conceptualizing consumer, prosumer and business processes at model description and at runtime.

Pseudonymous trace abstraction exploits the experiential pseudonymous activities embedded in the blockchain that take advantage of trace comparison and retrieval for effective and quick adaptation.

Smart matching with community detection engages with the relevant audiences and communities based on semantic abstraction and DApp requirements.

Decentralized decision making and reputation mechanisms apply to all entities irrespective of their role for enhancing the efficiency of collaborative business and prosumer models. This eradication of disputes and dissatisfaction through decentralized participation and incentivization opportunities.

2.1.4. TAC

TAC is a guided analytics and knowledge extraction tool for consumers, prosumers and businesses that aggregates contextualized data over spatial-temporal boundaries based on socio-cultural abstractions. TAC supports an analytic system for decentralized social media DApps, injecting additional information to improve operational tasks. This provides meaningful insights to DApp providers for improving their businesses and profits and to users for enhancing their experience and earning extra revenues. TAC consists of three services.

Augmented cognition data model consists of two microservices:

- **Geospatial microservice** gathers, displays and manipulates the data consisting of longitude and latitude information and provides meaningful geolocation information to prosumers for boosting their experience and improving their economic benefits.
- **Temporal microservice** supports complex analyses of the ARTICONF social network and allows users to benefit from actionable insights from a large amount of data processed over a short period of time.

Social-contextual model optimizes the platform functionality through empirical research that engages developers, end-users, digital right activists, non-governmental organizations, and scholars in relevant fields (e.g. social sciences and humanities, computer science). The goal is to understand how to promote different forms of community value, considering improvements in end-user engagement, transparency, fairness, trustworthiness, and sustainability.

Guided analytics for collaborative economy uses an interactive interface to assist social media consumers, prosumers, and businesses in injecting intelligent insights in data aggregation and cognition. The visualization guides social network users by focusing on the analysis and visualization of the parameters of interest and allowing different configurations for each use case. This way, the visual analysis moves beyond reporting shallow summary data to provide strong and actionable insights from users.

2.2. ARTICONF DApp use cases

A large variety of individuals, social entrepreneurs, civil society organizations, research centers, small and medium enterprises, as well as startups can strongly benefit from the ARTICONF platform. The project gathered a selection of four complementary social media DApp use cases to validate its goals.

2.2.1. Crowd journalism

Crowd journalism with news verification is a DApp providing opportunities to independent journalists and the news broadcasting industry to create content outside the mainstream media by gathering crowdsourced news with public participation. Two challenges faced by DApp providers are to validate the crowdsourced news with precise and trustworthy participation, and to provision time-critical infrastructure resources closer to the news location.

2.2.2. Car-sharing

Car-sharing is a form of person-to-person lending or collaborative consumption in a sharing economy, where existing car owners rent their vehicles to other people for short periods of time. Two challenges faced by this DApp are low public awareness on shared mobility, and geographical constraints with detailed routes, flexible offering, precise planning, reliable execution, and optimized costs.

2.2.3. Video opinion discussion

Video opinion discussion is a collaborative platform for publishing and subscribing to online videos. It allows non-professional users to record videos, share them on platforms, and earn rewards from their viewers. Two challenges faced by this DApp are the contextualized, thematic search of audio-visual metadata in large video libraries, and the security of a scalable business model that rewards users for their interactions, including content generation.

2.2.4. Energy marketplace

Energy marketplace uses a P2P monetized utility platform to reduce the energy bill of the prosumers by stimulating their energy sharing and demand response. The DApp encourages consumers to become prosumers who generate and consume energy in response to the increasing distributed energy generation at the demand side. Such human-agent models face two challenges: lack of intelligent techniques to identify the behavioral prosumer decisions over a specific smart appliance, and lack of an efficient data management plan to track the energy produced by each user for efficient reward allocation.

3. Decentralized car-sharing

The last decade saw an increased popularity of car-sharing adoption resulting in an explosion of providers offering shared-mobility services in the market (e.g. Uber [10], ZipCar [11]). In contrast to Uber, ZipCar and other centralized shared-mobility providers, the ARTICONF's car-sharing model creates a new decentralized platform based on blockchain [12] and smart contracts to face this growing market and to meet the appropriate service requirements related to user trust, resilience and cost. The car-sharing use case does not cover a single business model, but allows direct user interaction (C2C), or with car-sharing companies and fleet providers (B2C).

3.1. Car-sharing DApp architecture

Fig. 2 displays the high-level conceptual architecture of the car-sharing consisting of three main services.

Social network for each city serves the users who interact, plan (where and when a vehicle is available), hire a service, or share contents like photos and short videos. Sharing contents has various purposes, such as showing operators vehicle damages, fuel level, or battery status, or asking other users for a route. Customers use the social network to reduce risks related to their data, protected by a democratic mechanism that controls the malicious use of the network and its contents. Users have a reputation score assessed based on their actions, such as contracts fulfillment without penalties, reporting of real network and car-sharing issues, or publishing contents and service policies.

Blockchain network allows the users to easily create and deploy smart contracts using a contract generation service, automatically verified, resolved, and equipped with coded penalties in case of breaches. The customers can use the cars without financial worries since the money is available and visible only in the smart contracts, executed only upon triggering certain prerequisites.

Geolocation monitoring service [13], installed on each operating vehicle, tracks and verifies the real-time location of each vehicle and user smartphone, and resolves the clauses of the smart contract. The smart contracts have an escrow that depends on the users' reputation, measured based on the service policies.

Artificial intelligence (AI)-based rewarding and fleet allocation algorithms provide economic benefits for users and companies by eliminating fleet idle times, reducing the impact of external events, and enhancing the user experience [14,15].

3.2. Car-sharing DApp deployment and operation

Fig. 3 shows an architectural workflow for deploying and operating the car-sharing DApp using the ARTICONF tools. CONF represents the main tool for a DApp developer and provider to utilize the underlying cloud resources, which deploys the TIC backbone blockchain for the provisioning and management of different networks in different cities. Car-sharing clients use the TIC adapter wrapped in a mobile app to access, create, and manage smart contracts. TIC's identity and portability management allow the movement of users between cities without requiring the creation of new credentials. Moreover, TIC allows higher throughput by automating the smart contract generation process and the availability of the blockchain network. Additionally, CONF manages the deployment and the scalability of different car-sharing microservices by interacting with the SMART and TAC tools.

CONF provides a back-end service for the car-sharing provider to deploy the DApp microservices and to dynamically control them at runtime. CONF creates a resource configuration plan based on the car-sharing DApp requirements, including the number of virtual machines and the detailed deployment steps. The DApp uses a specific API to integrate the client SDK for the TIC's hyperledger fabric and to instantiate the decentralized network, its storage database and the prescriptive system. CONF also deploys the required virtual machines and containers, and performs continuous monitoring to detect infrastructure and resource performance anomalies hindering the microservices scalability.

TIC offers two communication methods upon deployment. On the one hand, TIC provides a RESTful API that uses the client SDK of NodeJS for the hyperledger fabric to communicate with the blockchain, grouping all the functions invoked in the chaincode. On the other hand, TIC utilizes a React Native[2] framework to integrate the TIC adaptor as a special client SDK that encrypts the information at the client side and allows data anonymization before sending it to the blockchain network. Moreover, this adapter allows calling every function in the chaincode and registering new users on the platform. TIC also registers diverse user activities and communicates to SMART with a RESTful API.

SMART utilizes the experiential traces embedded within TIC's blockchain network to provide a semantic mapping that captures car-sharing providers and user requirements originating from spatio-temporal mismatches between demands (e.g. trips, travelers, requests) and supplies (vehicles). Such a contextual mapping enables car-sharing providers to enhance customer escalation through personalized end-to-
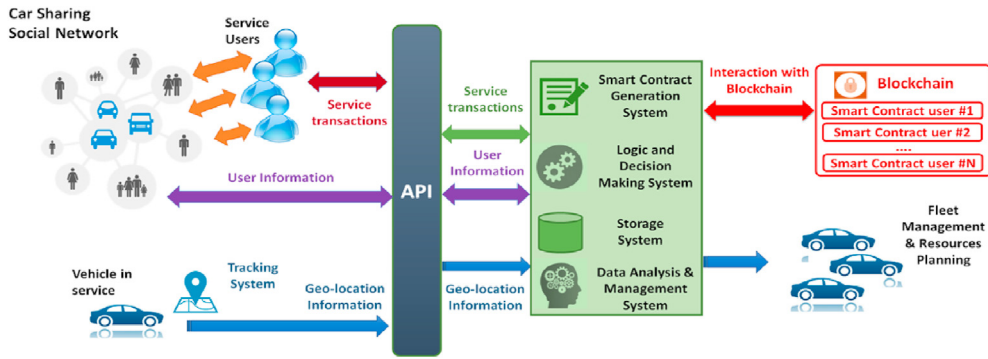
---

[2] https://reactnative.dev/.

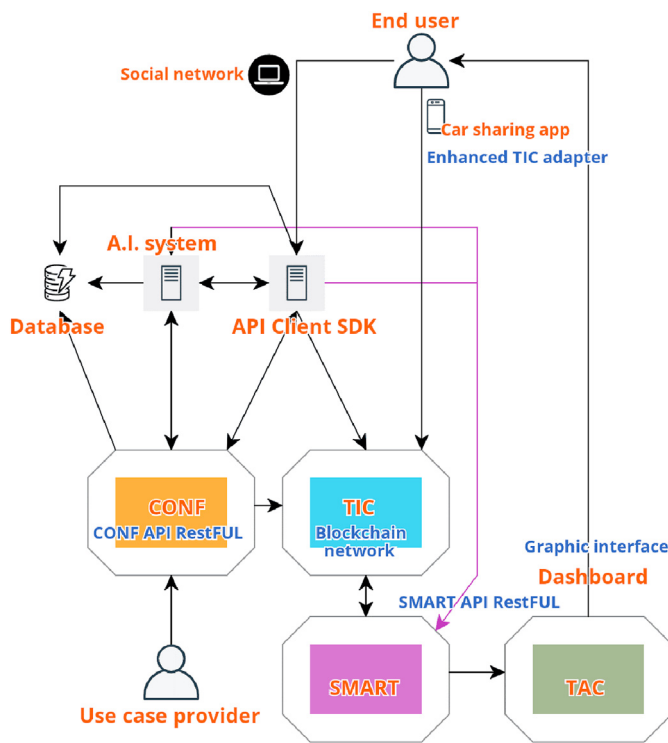Fig. 2. High-level car-sharing use case architecture.

Fig. 3. ARTICONF car-sharing deployment.

end travel experiences. Additionally, SMART's spatio-temporal community detection empowers car-sharing companies to identify semantically similar customers (e.g. based on lifestyle preferences) for optimizing fleet management and predicting travel demands with customized mobility options. Moreover, SMART provides trust metrics to calculate and query the user reputation through a RESTful API. An important objective of the reputation model is to detect lobbies or interest groups, which support malicious users in improving their reputation or ratings. Upon detecting such behavior in the community, SMART decreases the weight of its impact in the reputation model and further removes or mitigates it by learning from previous facts.

TAC exploits the knowledge processed by the SMART tool for aggregation, prescriptive analytics and visualization, offered to providers, prosumers and consumers through a dynamic visualization dashboard (e.g. car rental heat-map zones). The insights offered by the TAC geospatial microservice include the coordinates and addresses of the parked cars, the places allowed for ending a trip (without penalties), the rewards for ending the trip to suggested destinations, the list of users who started or wait for joining a trip, and the users' locations when sending messages to the platform. The TAC temporal microservice supports complex

analyses of the car-sharing social network, helping in avoiding traffic jams and using seasonality for accurate traffic prediction [16]. TAC's ultimate goal is to display guided analytics on the behavior and engagement of social media actors, which helps them diagnose performance risks and improves collaboration, successful sustainability and revenue growth. TAC also interacts with the car-sharing social media DApp to identify external factors (e.g. weather forecast) affecting the optimized rental of cars.

3.3. Car-sharing workflow scenario

Let us assume a customer called Alberta who asks for a car or to join travel. Her interaction with the DApp triggers the following workflow scenario:

1. The DApp first checks the certificate delivered by TIC, which entitles her to use the wallet, and handles the request afterwards.
2. CONF scales up the platform in case of a large number of concurrent requests and keeps the QoS by balancing the demand, replicating APIs or increasing the capacity of the database if the request comes along with the use of the social network.
3. SMART analyses Alberta's previous behavior and establishes a reputation rating before responding to the request. At the same time, it analyses the demand prediction at the destination and sets the price accordingly.
4. The DApp uses TAC to suggest the best price to Alberta and several options based on the analytics provided by SMART. If she decides to change her destination to a location closer to the more demanding zones, she will receive a better price.
5. Finally, Alberta decides the price and the travel, stores them in a smart contract together with the escrow for the vehicle and publishes the route for other users to join. However, she can modify the smart contract setup for the travel until the start time.
6. After Alberta starts the travel, the DApp collects geolocation from several sources, such as the car, the driver, and other passengers.
7. SMART analyses this data to detect potential inconsistencies or fake information. This data represents inputs to the smart contract executed on the blockchain network provided by TIC.
8. When Alberta finishes the travel, the platform evaluates the SLA through TIC and the smart contract. The escrow needs information from the next user of the same vehicle, who checks its status and validates it using a new entry for the smart contract.
9. Finally, SMART uses the new data stored on the blockchain for the next assessments of Alberta, affecting her future reputation and behavior.

4. Car-sharing DApp implementation

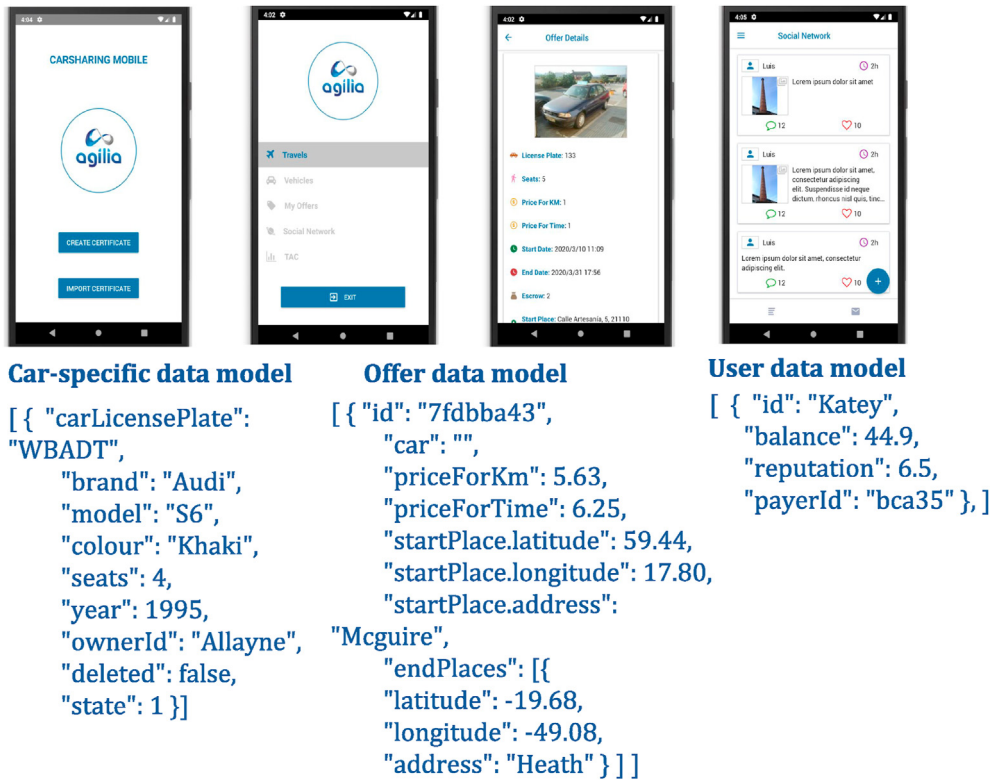Fig. 4 shows the graphical user interface of the car-sharing mobile DApp providing four interfaces to:

Car-specific data model

[{ "carLicensePlate":
"WBADT",
    "brand": "Audi",
    "model": "S6",
    "colour": "Khaki",
    "seats": 4,
    "year": 1995,
    "ownerId": "Allayne",
    "deleted": false,
    "state": 1 }]

Offer data model

[{ "id": "7fdbba43",
    "car": "",
    "priceForKm": 5.63,
    "priceForTime": 6.25,
    "startPlace.latitude": 59.44,
    "startPlace.longitude": 17.80,
    "startPlace.address":
"Mcguire",
    "endPlaces": [{
    "latitude": -19.68,
    "longitude": -49.08,
    "address": "Heath" }]]

User data model

[{ "id": "Katey",
    "balance": 44.9,
    "reputation": 6.5,
    "payerId": "bca35" },]

Fig. 4. Car-sharing mobile application interface and its associated data model.

1. Import or create a certificate pertaining to travel;
2. Import the travel details;
3. Display offers by different car owners specific to travel plan;
4. Provide user access for uploading content (e.g. images, videos, comments).

The following sections describe the data models and the prototype implementation, preparation and integration of the car-sharing DAPP through its interaction with the ARTICONF tools.

4.1. Car-sharing data model

To validate the car-sharing prototype, we integrated the Mockaroo[3] random data generator and API mocking tool into its mobile application interface. Mockaroo enables the creation of realistic test data in CSV, JSON, and SQL formats. We used Mockaroo to generate 100 000 cars, 100 000 users, and 100 000 offers according to the schema presented in Fig. 4. The test data set is openly available in JSON format on the Zenodo archive[4] for reproducibility purposes.

Car-specific data schema has eight fields: license plate, brand, model, color, owner identifier, number of seats, manufacturing year, and availability status.

Car offer data schema comprises eight fields: offer identifier, car license plate, price per km, price per time, start coordinates (latitude and longitude), as well as start and end locations (addresses).

User data schema contains four fields: user identifier, balance, payment source (generated using the PayPal RESTFul API[5] and reputation (i.e., numeric score).

Travel data schema is a combination of car, offer and user data models, instantiated by the users, registered onto the blockchain, and used by the SMART and TAC tools for analytic purposes (see Fig. 5).

[{"id": "c3a22ae7",
  "carLicensePlate": "3D7JB",
  "users": [{"userId": "Alberta", "passengers": 2},],
  "offerId": "bbf22efb",
    "startPlace.latitude": -22.84,
    "startPlace.longitude": -47.22,
    "endPlace.latitude": 49.83,
    "endPlace.longitude": 20.97,
    "startDate": 1571595235,
    "endDate": 1571642035,
    "suggestedEndPlaces": [
  {"endPlace": {"latitude": 14, "longitude": 121},
  "reward": "92.57", "suggestedBy": "Gerda"},],
  "kmTraveled": 183.44, "status": 2,
  "priceBalance": 0, "depositBalance": 137.37,
  "rewardBalance": 0, "startedBy": [{
  "coordinate": {"latitude": 31, "longitude": 120},
  "moment": 1573166855, "user": null},],
  "seats": 2, "totalPrice": 50.58,},]

Fig. 5. Car-sharing travel data model.

4.2. TIC for car-sharing

TIC facilitates a configurable blockchain platform with support for dynamic addition and removal of new organizations to the car-sharing consortium, as the business scales across different geographies. The TIC blockchain services described in the following sections provide to car-sharing DApp providers and developers an advanced environment to quickly deploy (Section 4.2.1) and configure (Section 4.2.2) a secure permissioned blockchain network for rapid DApp development and integration. The policy management in Section 4.2.4 provides an interface to add and update the rules that govern the access permissions of the participating nodes in the deployed permissioned network. The security model in Section 4.2.5 describes the pseudonymization of personal user data through encryption, while highlighting the user data sharing permissions in a two-stage permissioned blockchain architecture. The network management and blockchain explorer microservices in Section 4.2.6 manage and monitor the deployed blockchain network. Finally, the SDK and API described in Section 4.2.6 simplify the smart contract development process with rapid DApp integration.

4.2.1. Blockchain deployment

TIC uses Ansible [17] to manage the deployment of blockchain microservices. Ansible is an open-source configuration management, deployment, and orchestration tool that uses a secure shell to connect to the configured servers and run the tasks. Furthermore, Ansible is agentless and eliminates the need to install any additional software or open firewall ports on the client. This helps the car-sharing DApp providers to configure and deploy TIC microservices on remote public or private cloud infrastructures using secure shell access.

Fig. 6 illustrates the key Ansible components used by the TIC tool for the blockchain deployment:

- Inventories are a list of host Internet protocol (IP) addresses managed by Ansible.
- Playbooks are simple files that contain the task descriptions executed by Ansible in YAML format.
- Modules control the system resources like packages, or execute system commands on the remote host machines using playbooks.
- Plugins execute Ansible tasks as a job build step.

The car-sharing DApp provider starts the blockchain deployment by configuring the inventory with a list of IP addresses of host machines, running across private or public cloud providers. The Ansible automation engine executes the playbook tasks that configure the host machines with the blockchain platform requirements, followed by the deployment of persistent storage support (GlusterFS) and Hyperledger Fabric microservices. To achieve this, TIC identifies several sequential phases associated with the deployment of a sample organization in the car-sharing consortium, described in the following sections.

4.2.2. Infrastructure configuration for blockchain organization

The first step in the TIC deployment involves the identification of a reliable and scalable architecture for an organization participating in the car-sharing consortium. Its requirements may depend on various factors, such as network traffic or geographical distribution. Afterwards, TIC delegates the allocation of resources to CONF. Fig. 7 shows a prototype architecture of a car-sharing organization comprising one manager and four worker machines. The deployment process starts with the orchestration of the GlusterFS open-source distributed file system on these machines, which scales out in a building-block fashion to store multiple petabytes of data [18]. GlusterFS acts as a cloud storage for the common credentials and certificates needed by the blockchain network, as well as any other data required by the car-sharing DApp.

4.2.3. Blockchain service configuration

A simple interface provides access to the blockchain developments and facilitates the configuration of the hyperledger fabric microservices with the required number and type (i.e. endorser, committer, anchor) of peers, orderers, and car-sharing blockchain network. Such an interface eases the creation and deployment of a blockchain for the DApp providers without dealing with the underlying infrastructure complexities. Fig. 8 illustrates the configuration of a car-sharing organization, including:

- Two CA services, as follows:
- ORGCA that generates the membership service providers (MSP) for the agents (i.e. peers, clients, administrators) to interact with the blockchain network;
- TLSCA that generates the MSPs for the same agents to establish an internal or external transport layer secure (TLS) communication;
- An ordering service that implements the Raft consensus algorithm;
- Two peer services: Peer1 serving as anchor and committer peer and Peer2 serving as the endorser peer;
- A command-line interface (CLI) service to install or instantiate the chaincodes (smart contracts) onto the blockchain.

4.2.4. Access control policies

Unlike public blockchains where any node in the network validates the generated transactions, a permissioned blockchain requires collective mechanisms for controlling the network administration and its operations. Organizations that participate in a permissioned blockchain network need a proper governance model that defines how they agree on operations, transactions, updates, access rights, and others. They achieve this through policies that contain rules needed by their respective nodes for making any changes to the network.

Policies allow members to enforce decisions on the choice of the organizations that can update or access the deployed blockchain network. They include a list of organizations with access to a resource, the number of organizations' agreements required to update a smart contract, as well as transactions and channels. The set of digital signatures from each organization attached to transactions, which satisfy the governance agreed by the network, ensures adherence to policies.

4.2.5. Security model

The blockchain has two certificate authority services ORGCA and TLSCA, as described in Section 4.2.3. The network participants or users of the organizations use these certificate authorities to implement cryptographic algorithms for verification, signing and identity attestation, achieved by the MSP process running on the channel levels and ordering service. The MSP process consists of a set of protocols and cryptographic mechanisms for validating and issuing identities and certificates throughout the deployed blockchain network. This ensures that the issued identities have access within their defined scope.

Furthermore, TIC's SDK adapter provides a option to encrypt data with an user key before adding it to the blockchain ledger, particularly useful in the pseudonymization of sensitive personal data. At the time of user registration or login, SMART requests user permissions to access their blockchain ledger data for analysis. The user grants this permission by sharing the key with SMART and enabling it to access the encrypted ledger data and use it for analysis.

4.2.6. Ansible deployment

4.2.6.1. Blockchain platform setup.
Four playbooks prepare the allocated resources with the prerequisites to set up a blockchain platform for the business configuration:

- Install all prerequisite software for the fabric blockchain network, depending upon the operating system type (e.g. Debian, Centos);
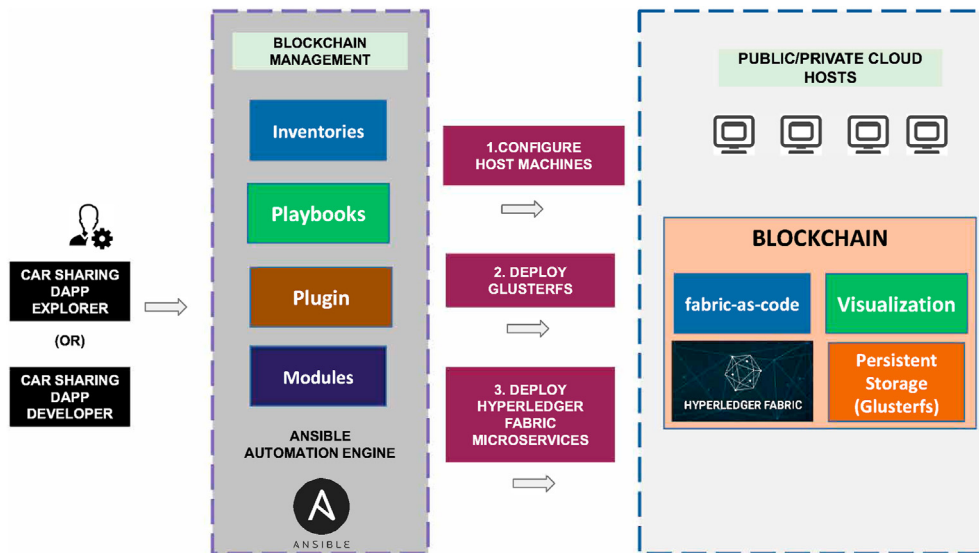
Fig. 6. TIC (trust and integration controller) blockchain deployment management.
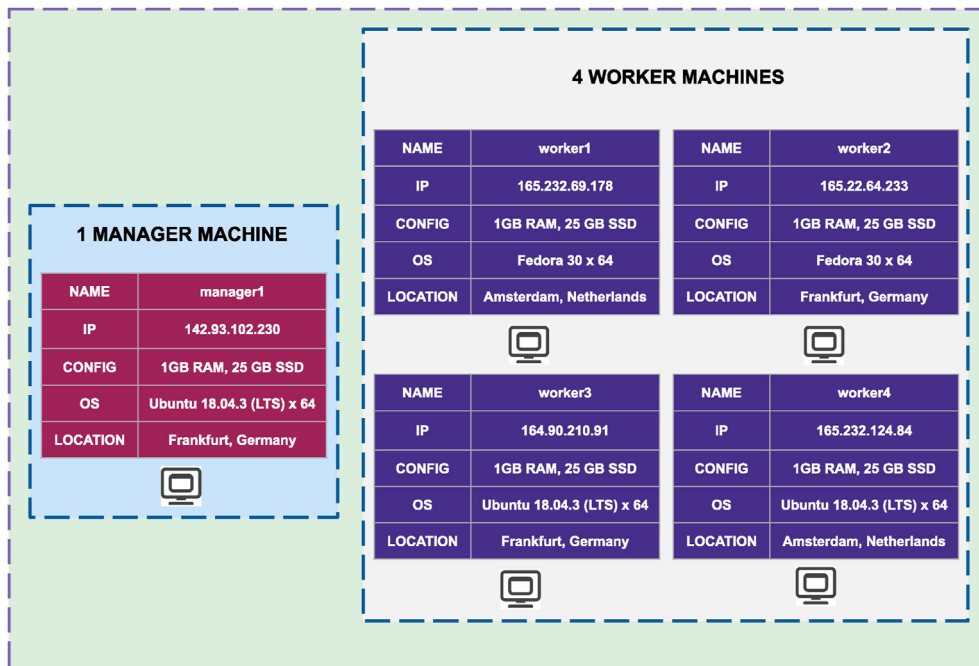

Fig. 7. TIC (trust and integration controller) car-sharing resource allocation architecture.

- Pull the required Docker images of the hyperledger fabric services, deployed as plug-and-play modular components;
- Deploy the GlusterFS cloud storage cluster for the blockchain network;
- Spawn the Docker services according to the business configuration.

4.2.6.2. Hyperledger fabric services. Multiple playbooks deploy the hyperledger fabric Docker service in the network, as follows:

- Spawn two certification authority services, as described before;
- Spawn the ordering service implementing the Raft [19] consensus algorithm;
- Deploy the peer services according to the business configuration, for example, one peer acting as anchor and committer (Peer1) and one peer acting as endorser (Peer2);

- Deploy a CLI service to install or instantiate the smart contracts onto the blockchain network, which define the transaction logic that controls the lifecycle of the car-sharing business object;
- Add or update the channel for conducting confidential and private transactions between several network members through private communication;
- Add or update system policies with the set of rules deciding which organizations can update or access the fabric network.

4.2.6.3. Visualization services. Two playbooks spawn the portainer and the portainer agent,[6] and two further playbooks spawn the hyperledger
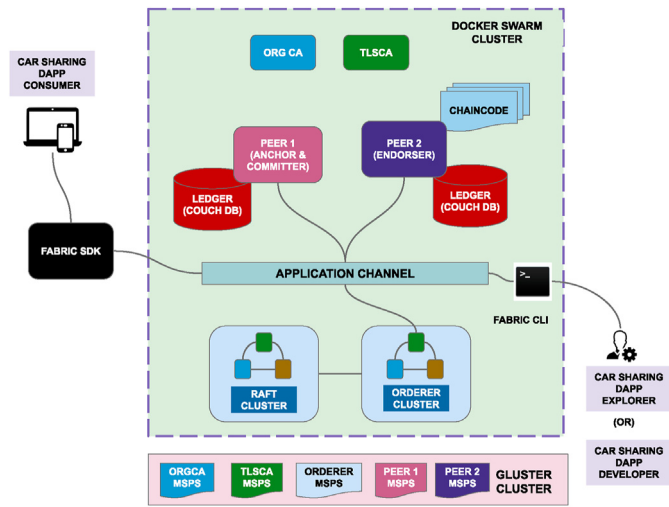
---

[6] https://www.portainer.io.

Fig. 8. TIC (trust and integration controller) hyperledger fabric service configuration for a sample car-sharing organization.

explorer[7] blockchain visualization services. Fig. 9 represents a snapshot of various Docker hyperledger visualization services deployed by TIC on the manager machines, and of the hyperledger fabric services deployed across four worker machines, as follows:

- Raft based orderer service on worker machine 1;
- CLI and ORGCA services on worker machine 2;
- CouchDB service storing the blockchain ledger on worker machine 3;
- Anchor and committer Peer1, endorser Peer2 and the TLSCA service on worker machine 4.

4.2.6.4. Deployment verification. To verify the deployment of a blockchain network according to its business configuration, the hyperledger explorer service provides visualizations of blockchain metrics like the number of participating nodes, transactions, blocks, and chaincodes (see Fig. 10). The car-sharing DApp provider accesses the hyperledger explorer service via the configured credentials and ports. The hyperledger explorer provides an interactive visualization interface for the deployed blockchain network used for monitoring and verification of the deployed car-sharing blockchain network. Primarily it shows a summary of the number of blocks and transactions created within the blockchain, and reports on the active nodes inside the network and the deployed chaincodes.

4.2.6.5. Car-sharing chaincode installation. TIC facilitates the installation and instantiation of the car-sharing chaincodes through a CLI service, verified using the hyperledger explorer. These smart contracts implement the business logic behind the car-sharing DApp. The car-sharing DApp provider can install, instantiate and update smart contracts through the portainer management service CLI.

4.2.6.6. TIC microservice management. The network management portainer provides a dynamic overview of all microservices of the deployed network (see Fig. 11). For each microservice, it reports their status, health and number of replicas. Furthermore, it provides facilities to dynamically update specific microservices based upon the business requirements, particularly useful in fine-tuning the microservices parameters for better performance (e.g. upscaling, downscaling).

4.2.6.7. Test node SDK. TIC supports the integration of mobile and web

Apps with the blockchain network through a NodeSDK. It also provides basic helper libraries for registering a user to the blockchain, querying or invoking a chaincode for easy integration, and verifying the car-sharing prototype implementation. Similarly, it can dynamically configure and add organizations to the car-sharing consortium, as the business grows across geographic areas.

4.3. CONF for car-sharing

We describe in the following a concrete car-sharing scenario with improved reaction time during peak hours, invoked using the Postman[8] collaborative platform for API development. In this example, CONF detected an increased load on the metrics database, which significantly affects the QoS and the user experience, mitigated in four steps.

Step 1. The DApp provider specifies a high-level description of the application in Topology and Orchestration Specification for Cloud Applications (TOSCA) without the underlying software dependencies or infrastructure.

Step 2. The application provider requests a plan using the identifier provided by CONF. CONF resolves all constraints and delivers a plan that contains the infrastructure and software definition to run the application. In this example, CONF detected an increased load in the application database and provisions virtual machines as close to its source as possible.

Step 3. The infrastructure provisioner executes this plan based on the available cloud providers.

Step 4. After provisioning, the client requests to deploy the car-sharing DApp on a Kubernetes cluster. Fig. 12 shows the Kubernetes dashboard together with the deployed metrics database and the application state (e.g., state of deployments, services, and pods).

4.4. SMART for car-sharing

SMART provides support for business predictions and recommendations to car-sharing DApp providers. To achieve this, it extracts information from the raw blockchain transactions and transforms it into a more expressive structured representation. Essentially, SMART exploits the experiential pseudonymous of car-sharing DApp user activities embedded in the TIC blockchain as immutable traces, such as the data models shown in Figs. 4 and 5.

4.4.1. SMART architectural workflow
The SMART architectural workflow follows three steps:

- Semantic linking [20] and contextualization [21] of blockchain transactions;
- Detecting communities based on contextual similarity and assign roles to DApp user groups;
- Temporal decomposition of contextual communities into stages.

Fig. 13 shows the SMART workflow for semantic linking, contextualization, community role detection, and temporal stage abstraction of DApp transactions. Initially, SMART collects semi-structured transaction data from the blockchain, splits them into unique contexts (e.g. location, timestamp), and semantically links them through diverse contexts. Afterwards, it identifies unique DApp user roles representing unique behavioral patterns in a specific context. Furthermore, SMART clusters transactions represented as nodes for each individual context, where each cluster represents transactions with similar properties (e.g. locations, times, prices, values). Within a geolocation context, for example, an explanatory cluster could span over a unique district D of a city C.
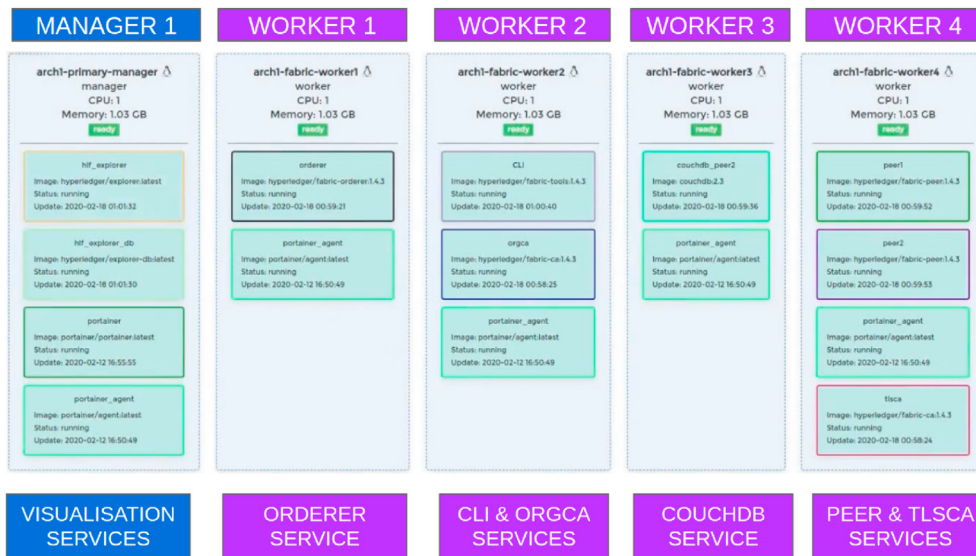
Fig. 9. Docker visualization of a TIC (trust and integration controller) deployment.
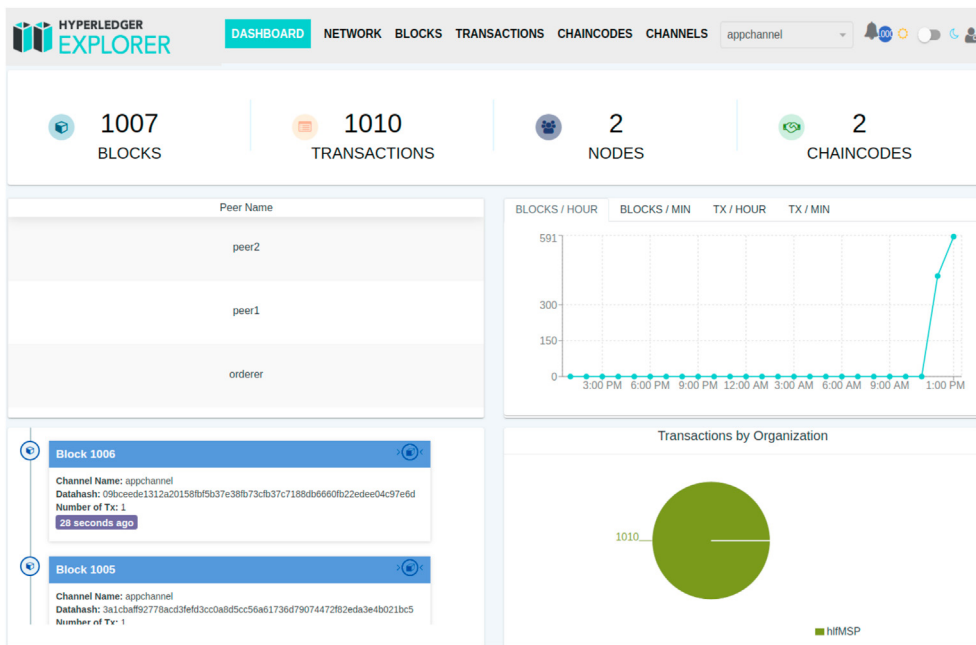

Fig. 10. TIC (trust and integration controller) hyperledger explorer deployment verification.

Hence, the cluster of users performing transactions receives the label $C - D$. We chose to label clusters instead of DApp users, as directly labeling users based on their transactions can be intrusive and compromise their anonymity. We further temporally partition the labeled cluster groups into stages, where a stage only contains transactions within specific time intervals. The temporally partitioned stages allow comparison of subsequent stages and discovery of the clusters' evolution (e.g. growth, shrinkage). This process allows SMART to not only understand the changing behavioral patterns of social media users in a single context, but also across contexts.

4.4.2. Semantic linking and contextualization

SMART explores multilayer contextualized semantic linking [22] and enrichment of raw blockchain transactions. For this purpose, it defines a set of layers showing different contexts of the DApp transactions. Additionally, SMART jointly considers the context of all network entities and

their social similarity strength. To define a semantic link, SMART considers different semantic labels for edges across contextual layers, where similar semantic labels stay within a single layer. SMART provides a fully interconnected network, where all layers contain all nodes, following a diagonal coupling model such that inter-layer edges only exist between nodes and their counterparts. This model also adopts a categorical coupling model, where inter-layer edges are present between any pair of layers, and links between pairs in each layer describe similarity strengths.

Fig. 14 shows the activity traces of the car-sharing DApp represented as a multilayer networked graph with six identified layers:

- *L1-Reputation* represents the credit points earned by each user for using the car-sharing application;
- *L2-Users* represent pseudonymous, as unique user identifiers;
- *L3-StartingPoints* is the source location of each user travel;
- *L4-Time* shows the timestamp at each travel source location;

Fig. 11. TIC (trust and integration controller) monitoring and portainer management of microservices.
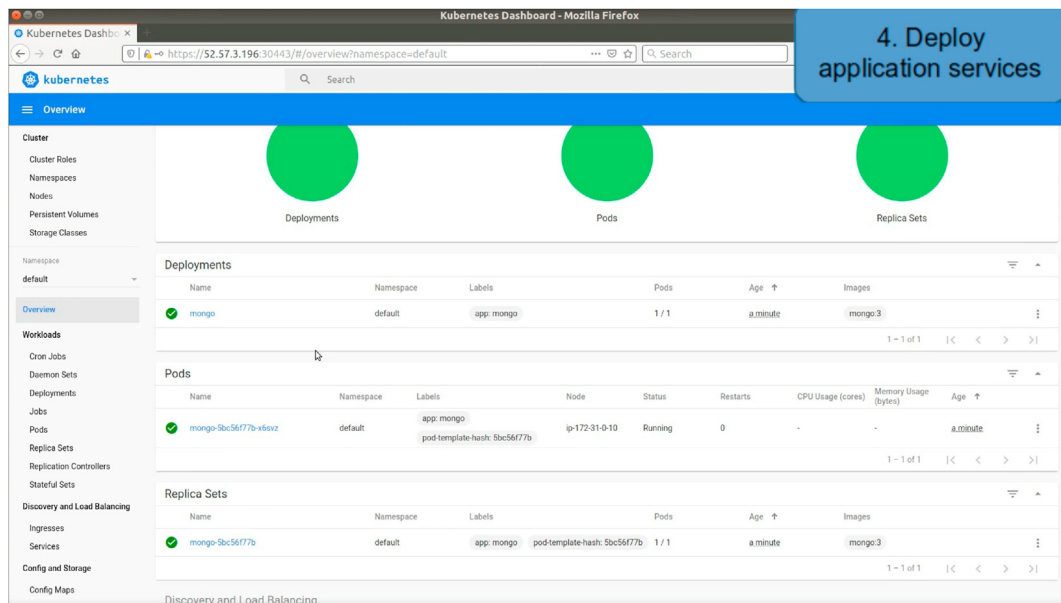

Fig. 12. CONF (co-located and orchestrated network fabric) Kubernetes application dashboard.

- *L5-Destination* represents the destination of each user travel;
- *L6-TravelPrices* is the price per-travel paid by the users.

Each layer in Fig. 14 represents a context embedded with multiple agents, where each agent is a car-sharing DApp user with a behavioral pattern within each layered context. For example, a behavioral pattern within the *L4-Time* contextualized layer varies the timestamp of different users at the source location. SMART identifies overlapping agents in different layers and semantically links them to understand the role of each user in different contexts, as shown in Fig. 14. SMART performs semantic linking at regular time intervals to obtain multiple multilayered graphs, stored in its knowledge base.

4.4.3. Car-sharing community and role detection

The multi-layer network contains raw, but contextualized information. Hence, the next step is to find similar agents (i.e. car-sharing DApp user) based on the characteristics of each layer. To achieve this, we applied a clustering technique that tags similar agents with the same arbitrary but fixed label. Initially, SMART fetches the multilayered contextualized graphs across temporal stages as input for community and role detection. Further, SMART utilizes the OPTICS [23] augmented cluster-ordering algorithm to find similarities among different agents in each layered context through a distance function and a minimum number of neighbors required as a unique community. Fig. 15 shows the clustering performed for the *L3-StartingPoints* layer, where the same color blocks represent a community. In this case, the distance-based similarity is a function of the longitude and latitude of the source location. Each contextualized layer repeats the
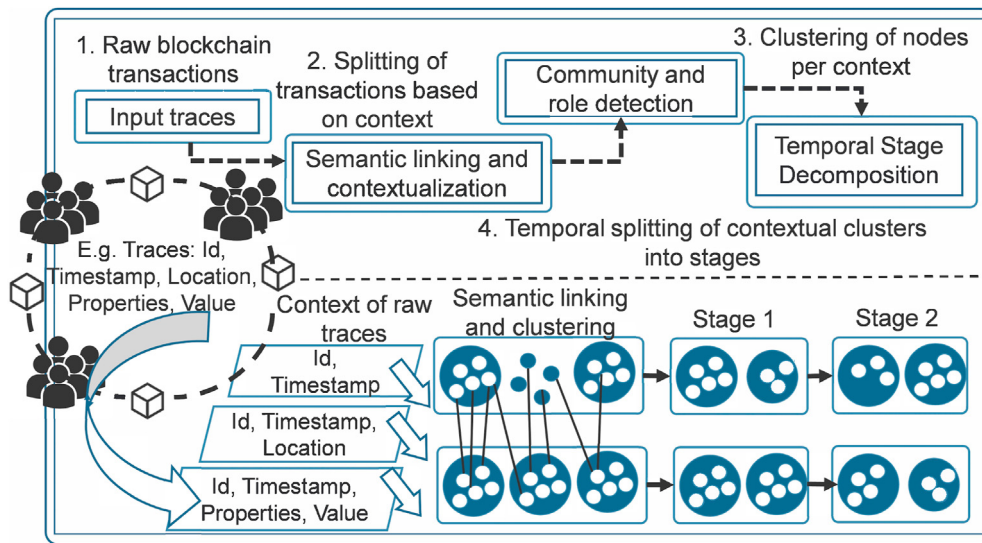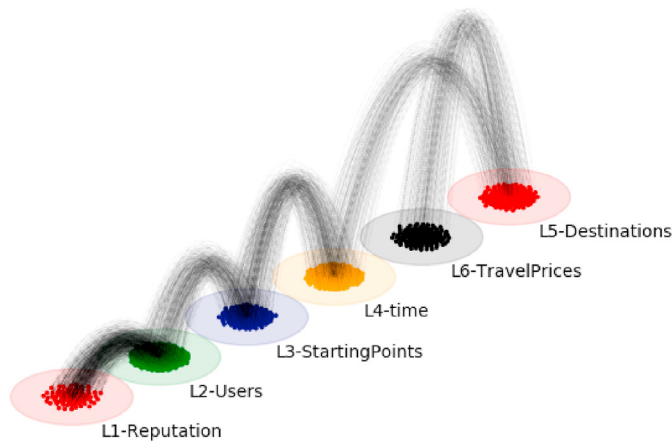
Fig. 13. SMART tool workflow.

Fig. 14. Multilayer car-sharing contextualization.

clustering algorithm with distance function models.

After identifying the clusters for each layer, we implicitly know the roles of the agents within a cluster. For instance, a cluster *C* representing the approximate location of a city *A* indicates that the agents in *C* have their location in *A*. If one agent is in *C* multiple times, we assign a label connecting it to the city *C*. SMART assigns roles only to clusters, as direct user labeling based on transactions can be intrusive and compromise anonymity.

4.4.4. Car-sharing temporal stage decomposition

After obtaining the clusters in each layer, the next step analyzes the evolution of the networked clusters in terms of their temporal, structural, and size changes. For example, a car-sharing DApp provider is interested in the changes in the number of users who connect from a city *A* over an interval. To formulate such an evolution, SMART defines stages that split the time interval and represent snapshots of the networked cluster. In the current implementation, SMART slices the data based on weeks for an average of 52 stages per year. We apply these time intervals to the ob-
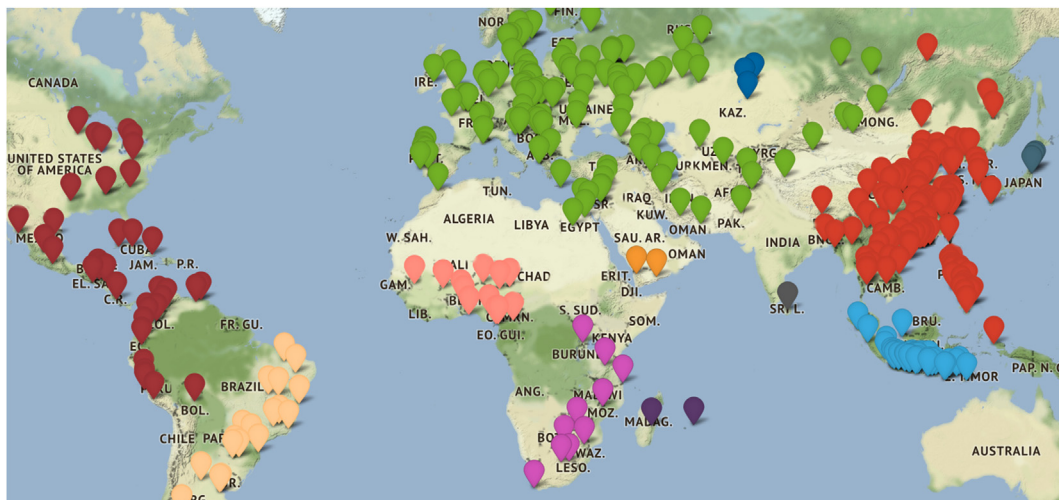

Fig. 15. Clustering for L3-StartingPoints layer, shown in Fig. 14.

tained clusters and analyze if single clusters grew or shrank from one stage to another. We plan in the future to dynamically create the stages by detecting and applying the natural peak use over time.

4.4.5. SMART car-sharing evaluation

We evaluate the SMART multilayered semantic linking, contextualization, community and role detection, and temporal stage decomposition for the car-sharing data model described in Section 4.1. We use an `Intel(R) Xeon(R) Gold 5218` server operating at 2.30 GHz using Ubuntu 18.04 (`x86_64`) operating system with an attached 9.6 TB solid-state drive. We used a MongoDB[9] database engine for managing the contextualized layer and clustered data due to its NoSQL and fast querying-based features.

Semantic linking and contextualization. SMART's multilayer car-sharing network has 100 000 trips corresponding to the data models described in Fig. 5. Fig. 16a shows that the multilayer network context creation and semantic linking time for this data set allocated across six network layers increases linearly with the number of journeys taken. The increase in the number of journeys also increases the analysis data volume required for creating the multilayer network, and results in a higher execution time.

Community and role detection. We used the L4-Time, L5-Destinations, and L6-TravelPrices car-sharing data set layers to evaluate the SMART community and role detection. We performed clustering of five input sizes with 1000, 5000, 10 000, 20 000, and 50 000 agents. Fig. 16b shows the execution time of the clustering-based role detection in SMART, where all three layers show an increase in runtime with a quadratic complexity. The clustering operation in the L5-Destinations layer across two dimensions (i.e. longitude and latitude) requires a higher execution time for large input sizes compared to the L4-Time and L6-TravelPrices layers clustered across a single dimension.

Temporal stage decomposition. We performed the temporal stage decomposition using clusters with computed within the L4-Time, L5-Destinations, and L6-TravelPrices layers with five input sizes of 1000, 5000, 10 000, 20 000, and 50 000 agents (i.e. car-sharing DApp users). Fig. 16c shows that the stage decomposition execution time increases with the number of agents, however, it is similar for all three layers requiring less than 0.5 s for 50 000 nodes. The reason is that stage extraction for clustered agents takes place across a single timestamp dimension (i.e. one week, described in Section 4.4.4), while keeping the number of agents constant in each layer.

4.5. TAC for car-sharing

The TAC interactive interface assists the car-sharing DApp consumers, prosumers, and businesses in injecting intelligent insights in data aggregation and cognition. Data aggregations are collections of data stored in buckets and grown from the geospatial and temporal microservices.

4.5.1. Data aggregations

Data aggregations generate analytical information over stored documents used for near real-time data analytics. There are two types of aggregations.

Bucket aggregations contain buckets created to store and group various documents based on the value of a specific field. The bucket aggregations usually combine with other types of aggregations, creating sub-aggregations.

Metric aggregations represent computations performed over several documents after creating bucket aggregations. Subsequently, the metric aggregations calculate and return the value of each bucket.

4.5.2. Geospatial microservice

The geospatial microservice handles the gathering, display, and

manipulation of geolocation data [24] to create categorized buckets based on geo_point fields. The geospatial map visualizes the starting positions of journeys that made the most revenue. This visualization requires metric aggregations on the data field containing geographic information (i.e. latitude and longitude) about the travel starting positions, and counts the corresponding total spending (see Fig. 17).

4.5.3. Temporal microservice

The temporal microservice handles the changing data over time. The temporal analysis coupled with the visualization microservice allows car-sharing users to follow their savings, check if a car is available for rent, and verify the status of travel (i.e. booked, started, finished, checked, canceled). The car provider can identify the rating of the user (which reveals their behavior during the trip), the date when the trip starts or ends, or penalize the user if the trip has not started or finished in a range of 5 min. Hence, the company can follow the distance traveled by a concrete car, the rating of the travel given by the passengers, the list of users who already started the trip with the dates and the coordinates, and the date of publications made by the users. Fig. 18 displays the time series trend of the aggregation data, which analyzes the change in maximum, minimum, average, and median rental price over time.

4.5.4. Visualization service

Visualization service is responsible for aggregating and exploiting the car-sharing content supply chain across providers, communities, groups, and users. It provides information that supports user engagement in collaborative economies with monetary inclusion, it increases the provider's awareness of users' activity on the platform, and it helps them track the rating and functioning of their application. TAC provides visual analytics of the car-sharing qualitative data obtained from SMART by applying filtering rules to select the most relevant parameters of interest specific to the car-sharing use case. For this purpose, it uses three types of visualizations implemented using three open-source software tools known as the elastic stack (Elasticsearch, Kibana, and Logstash). Fig. 19a shows a metric aggregation example that requests the average, minimum, maximum, and price per kilometer for the car-sharing data set. Fig. 19b shows the following aggregated result:

- Number of documents inside the index ("count" field of value 1000);
- Minimum value for priceForKm ("min" field of value 1.01);
- Maximum value for priceForKm (max field of value 10.0);
- Average value for priceForKm (avg field of value 5.49608);
- Total price paid for the travel ("sum" field of value 5496.08).

TAC offers several visualization types, including area chart, data table, line chart, markdown widget, metric, pie chart, tile map, and vertical bar chart. Fig. 20 visualizes the average, minimum, maximum, and median price per kilometer for the car-sharing metric aggregation given in Fig. 19. Fig. 21 shows the aggregated number of seats across diverse car brands visualized as a heat map. Similarly, the bar chart in Fig. 22 illustrates the top five car brands and the corresponding car types that achieved the most revenue.

5. Related work

This section presents related works on decentralized social media and car-sharing platforms.

5.1. Decentralized social media

Diaspora [25] is a non-profit, user-owned, distributed social network that addresses privacy concerns related to centralized networks. It consists of a network of nodes called pods, hosted by different individuals and institutions. Each node operates a copy of the Diaspora software running as a personal web server with social networking capabilities. Users of the network can host a pod on their own server, or create an
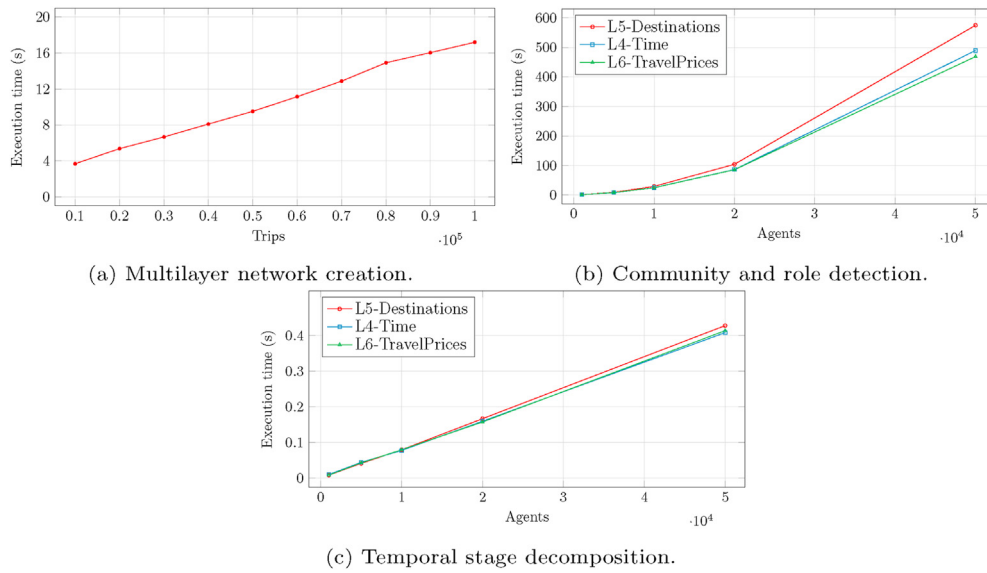

(a) Multilayer network creation.

(b) Community and role detection.

(c) Temporal stage decomposition.

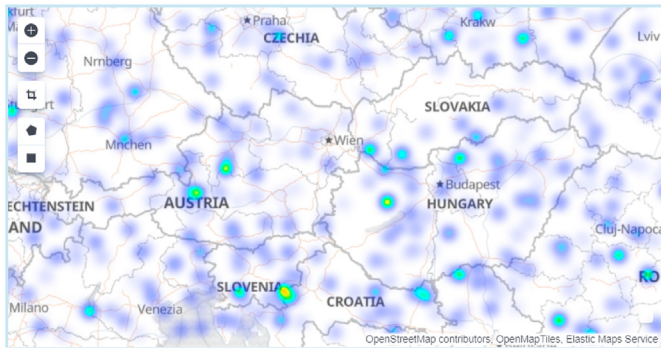**Fig. 16.** SMART car-sharing evaluation results.

Fig. 17. Coordinate map for the geographical points with most revenue.
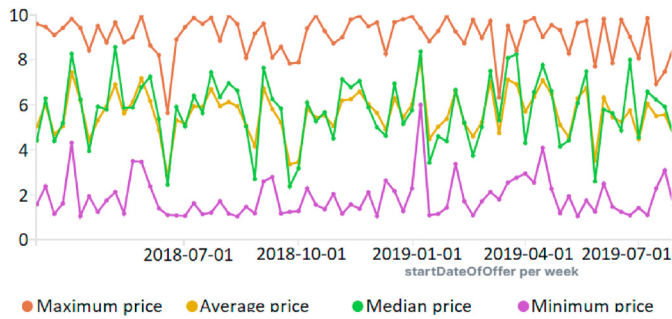

Fig. 18. Time series trend of rental price over time.

account on any pod to interact with other users. The Diaspora users retain ownership of their data and do not assign ownership rights.

Steem[10] is a social blockchain [26,27] that grows communities and generates revenue streams by rewarding users for sharing content. It supports community building and social interaction with cryptocurrency rewards. Steem provides a backbone to support social media and online communities by returning its value to the people who provide valuable contributions (e.g. content generation and propagation) and rewarding them with cryptocurrency. Through this process, it creates a currency

able to reach a broad market, including people who have yet to participate in any cryptocurrency-based sharing-economy.

UHIVE[11] is a blockchain-based, privacy-aware social media with token rewards for content producers and consumers, which respects privacy by not logging user activities and by permanently deleting data. The network distributes revenues and rewards in exchange for content, engagement with posts, and time spent on the app. It also provides a content discovery UX, powered by interest-based user selection. Finally, it supports public, personal, and interest-based user profiles.

Minds[12] is a decentralized social networking platform that rewards users with tokens for contributions to the community. Minds provides a free and open-source distributed crypto-social networking service that uses blockchain to reward the community with ERC-20 tokens. Minds users can use their tokens to promote content or to crowdfund other users through monthly subscriptions to exclusive content and services.

Matic[13] is a decentralized network that scales through sidechains for off-chain computation, and ensures security using the Plasma framework and a proof-of-stake validator. It provides scalability and superior user experience to DApps by using sidechains to guarantee fast, low-cost and secure transactions. Matic uses Ethereum, but intends to offer support for additional base chains. Additionally, it provides a developer abstraction from the main chain to a Matic chain, native mobile apps, and wallet support. Moreover, Matic uses public, permissionless sidechains capable of supporting multiple protocols.

Sapien[14] is a social news platform that offers users control over their data. Sapien utilizes a unique engine for rewarding content creators with SPN tokens by accepting micropayments or subscriptions, and allowing users to get SPN from advertisers. Additionally, it constructs a global reputation system that uses smart contracts to evaluate individual contributions.

MASTODON [28] is a free and open-source self-hosted social network that provides tools for community-based moderation [29,30], where each hosting server can limit or filter undesirable types of content [31, 32]. MASTODON also provides an interface for finding appropriate communities.

---

[10] https://steem.com/.

[11] https://www.uhive.com/.

[12] https://www.minds.com/.

[13] https://matic.network/.

[14] https://www.sapien.network/.

```
GET agilia-demo2/_search
{
  "size": 0,
  "aggs": {
    "Avg_min_max_med_priceForKm": {
      "stats": {
        "field": "priceForKm"
      }
    }
  }
}
```

(a) Price for distance (kilometer).

```
"aggregations" : {
  "Avg_min_max_med_priceForKm" : {
    "count" : 1000,
    "min" : 1.01,
    "max" : 10.0,
    "avg" : 5.49608,
    "sum" : 5496.08
  }
}
```

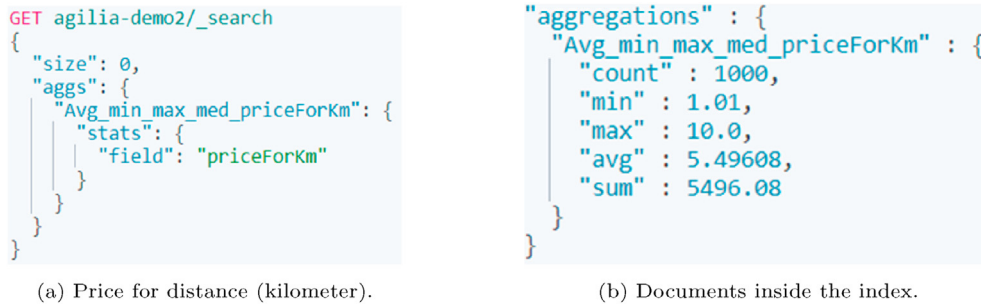(b) Documents inside the index.

Fig. 19. Metric aggregation.

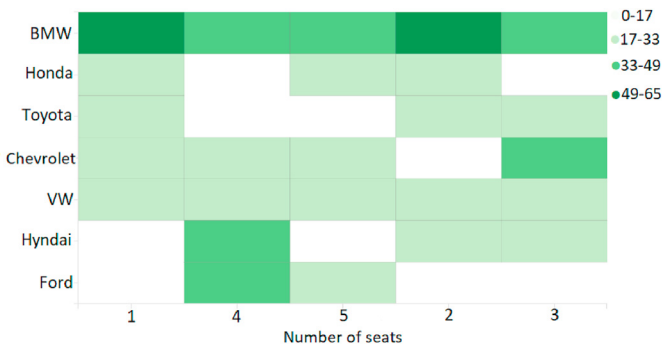**Fig. 20.** Data visualization for the distance metric aggregation (kilometer).

Fig. 21. Heat-map showing the number of seats in a car brand.
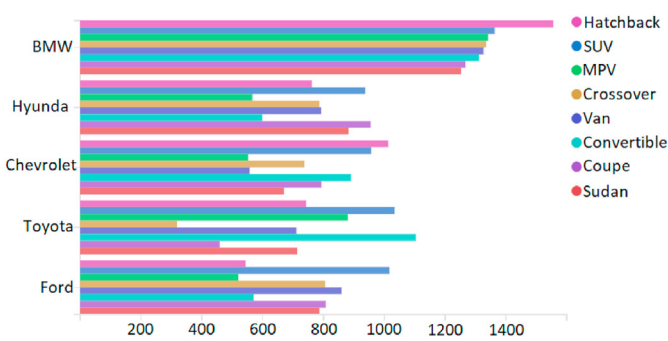

Fig. 22. Top five car brands and type of car with most revenue.

5.2. Decentralized car-sharing

The decentralization of social media platforms contributed to the rise of many business models, in particular for blockchain-based applications [7,14]. One such prominent model is the P2P mobility-as-a-service marketplace. We discuss in this section several existing car-sharing market DApps.

HireGo[15] is a car-sharing DApp based on the Ethereum blockchain. HireGO facilitates car-sharing and rentals between owners and passengers through smart contracts, and enables rewards by distributing virtual HGO tokens based on the non-fungible token standard ERC-721 [33].

Helbiz[16] integrates the mobile transportation systems and provides ERC-20 [34] tokens called HBZ. Contrary to HireGo, Helbiz also rewards end-users for providing mobility data to insurance companies.

DAV [35] DApp provides mobility services for a set of interconnected autonomous vehicles (e.g. cars, trucks, drones) in exchange of ERC-20 DAV tokens, similar to HireGo and Helbiz.

WONO[17] DApp provides a decentralized marketplace for car and house rentals. WONO combines a public Ethereum with a private blockchain network that uses a proof-of-stake algorithm for consensus and an IPFS-based decentralized file system for data storage (e.g. user profiles, videos) [36]. WONO uses the public key infrastructure to store and encrypt personal data on the client device.

6. Conclusions

We presented the approach taken by the ARTICONF project funded by the Horizon 2020 program of the European Union that researches and develops a novel set of trustworthy, resilient, and globally sustainable decentralized social media services. To achieve this goal, ARTICONF proposes an open decentralized architecture consisting of four tools (i.e. TIC, CONF, SMART, and TAC), addressing in collaboration the following five main objectives:

- Transparent and decentralized infrastructure creation and control (TIC) through a novel permissioned blockchain network, supporting pseudonymous identities and offering users a secure, permanent and unbreakable link to their personal data;
- Improved and trusted participation (TIC and SMART) by eliminating malicious actors in participatory exchanges, with improved collaboration, trust, and operational costs;
- Democratic and tokenized decision-making (SMART) through collective decentralized reasoning that improves the content quality by finding interest groups and incentivizing users for their participation;
- Elastic resource provisioning (CONF) for customizing, deploying and controlling distributed P2P and cloud virtual infrastructures required by time-critical social media applications;
- Cognitive guided analytics for improved collaborative economy (TAC) that injects intelligent insights into operational and mission-critical social media applications, with predictive models for consumers, prosumers, and business markets.

We presented the ARTICONF architecture applied on a car-sharing DApp use case, as a new collaborative model and alternate software-as-a-service solution to private car ownership. We described a simulated use case scenario accompanied by real snapshots illustrating how the different tools of the ARTICONF platform allow car-sharing customers to engage in trustful and secure interactions for renting a vehicle at a variable fee, charged depending on the distance traveled or time used, while

---

[15] https://hirego.io/.

[16] https://helbiz.com/.
[17] https://wono.io/.

keeping complete control over their personal data.

Future work aims to apply AI techniques to optimize the fleet allocation based on the demand location, predicted by SMART by analyzing community movements inside a city using geolocation and previous travel information. The challenge is to match and reward travel to destinations close to high demand locations, which reduces the operational costs of relocating cars.

The ARTICONF project plans to validate its results on three further industrial applications targeting crowd journalism with news verification, video opinion discussion, and a smart energy sharing marketplace. The project started in January 2019 and expects to achieve its first prototype release in 2021.

Funding

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] T.M. Nisar, G. Prabhakar, L. Strakova, Social media information benefits, knowledge management and smart organizations, J. Bus. Res. 94 (2019) 264–272.

[2] G. Appel, L. Grewal, R. Hadi, A.T. Stephen, The future of social media in marketing, J. Acad. Market. Sci. 48 (1) (2020) 79–95.

[3] G. Beigi, H. Liu, Identifying novel privacy issues of online users on social media platforms, ACM SIGWEB Newslett. (4) (2019) 1–7.

[4] G. Loukas, C. Patrikakis, A prototype framework for assessing information provenance in decentralised social media: the EUNOMIA concept, in: E-democracy – Safeguarding Democracy and Human Rights in the Digital Age, vol. 1111, Springer, 2020, pp. 196–208.

[5] R. Prodan, N. Saurabh, Z. Zhao, et al., ARTICONF: towards a smart social media ecosystem in a blockchain federated environment, in: U. Schwardmann, C. Boehme, D.B. Heras (Eds.), Euro-Par 2019: Parallel Processing Workshops, Springer, Cham, Switzerland, 2020, pp. 417–428.

[6] A. Fertier, A.-M. Barthe-Delanoë, A. Montarnal, S. Truptil, F. Bénaben, A new emergency decision support system: the automatic interpretation and contextualisation of events to model a crisis situation in real-time, Decis. Support Syst. 133 (2020) 113260.

[7] V. Valaštín, K. Kostal, R. Bencel, et al., Blockchain based car-sharing platform, in: 2019 International Symposium ELMAR; 23-25 Sep 2019; Zadar, Croatia, IEEE, Piscataway, NJ, USA, 2019, pp. 5–8.

[8] A. Palanisamy, M. Sefidanoski, S. Koulouzis, et al., Decentralized social media applications as a service: a car-sharing perspective, in: 2020 IEEE Symposium on Computers and Communications (ISCC); 7-10 Jul 2020; Rennes, France, IEEE, Piscataway, NJ, USA, 2020, pp. 1–7.

[9] C. Cachin, Architecture of the hyhperledger blockchain fabric, in: Workshop on Distributed Cryptocurrencies and Consensus Ledgers, vol. 310, 2016, p. 4.

[10] S. Cannon, L.H. Summers, How Uber and the sharing economy can win over regulators, Harv. Bus. Rev. 13 (10) (2014) 24–28.

[11] M. Hart, M.J. Roberts, J.D. Stevens. Zipcar: Refining the Business Model, 2nd ed., Harvard Business School, Boston, MA, USA, 2005.

[12] T.C. Dao, B. Nguyen, B. Do, Challenges and strategies for developing decentralized applications based on blockchain technology, in: International Conference on Advanced Information Networking and Applications, vol. 926, Springer, 2019, pp. 952–962.

[13] A. Patel, N. Shah, T. Limbasiya, et al., Vehiclechain: blockchain-based vehicular data transmission scheme for smart city, in: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC); 6-9 Oct 2019; Bari, Italy, International Conference on Systems, Man and Cybernetics, IEEE, Piscataway, NJ, USA, 2019, pp. 661–667.

[14] P. Helebrandt, M. Bellus, M. Ries, et al., Blockchain adoption for monitoring and management of enterprise networks, in: IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON); 1-3 Nov 2018; Vancouver, BC, Canada, 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEEE, Piscataway, NJ, USA, 2018, pp. 1221–1225.

[15] D. Rojas-Rueda, A. de Nazelle, O. Teixidó, M. Nieuwenhuijsen, Replacing car trips by increasing bike and public transport in the greater barcelona metropolitan area: a health impact assessment study, Environ. Int. 49 (2012) 100–109.

[16] M. Lippi, M. Bertini, P. Frasconi, Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning, IEEE Trans. Intell. Transport. Syst. 14 (2) 871–882.

[17] M. Zadka, DevOps in Python: Infrastructure as Python, Apress, 2019.

[18] A. Davies, A. Orsaria, Scale out with GlusterFS, Linux J. (2013) 235.

[19] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm, in: 2014 USENIX Annual Technical Conference; 19-20 Jun 2014; Philadelphia, PA, USA, USENIX Association, Berkeley, CA, USA, 2014, pp. 305–320.

[20] M. Lama, J.C. Vidal, E. Otero-García, A. Bugarín, S. Barro, Semantic linking of learning object repositories to dbpedia, J. Educ. Technol. Soc. 15 (4) (2012) 47–61.

[21] A. Wagner, P. Haase, A. Rettinger, H. Lamm, Entity-based data source contextualization for searching the web of data, in: European Semantic Web Conference, Springer, 2014, pp. 25–41.

[22] M. Kivelä, A. Arenas, M. Barthelemy, J.P. Gleeson, Y. Moreno, M.A. Porter, Multilayer networks, J. Compl. Netw. 2 (3) (2014) 203–271.

[23] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, Optics: ordering points to identify the clustering structure, ACM Sigmod Record 28 (2) (1999) 49–60.

[24] A. Karadimce, N.P. Dimoska, D.C. Bogatinoska, N. Marina, A. Nandal, Guided analytics software for smart aggregation, cognition and interactive visualisation, in: Innovations in Computational Intelligence and Computer Vision, vol. 1189, Springer, 2021, pp. 521–528.

[25] A. Bielenberg, L. Helm, A. Gentilucci, et al., The growth of diaspora-a decentralized online social network in the wild, in: 2012 Proceedings IEEE INFOCOM Workshops; 25-30 Mar 2012; Orlando, FL, USA, IEEE, Piscataway, NJ, USA, 2012, pp. 13–18.

[26] B. Guidi, A. Michienzi, L. Ricci, Steem blockchain: mining the inner structure of the graph, IEEE Access 8 (2020) 210251–210266.

[27] B. Guidi, A. Michienzi, L. Ricci, A graph-based socioeconomic analysis of steemit, IEEE Trans. Comput. Soc. Syst. 8 (2) (2020) 365–376.

[28] M. Liu, R. Gehl, D. Zulli, Rethinking the "social" in "social media": insights into topology, abstraction, and scale on the mastodon social network, New Media Soc. 22 (2020) 1188–1205.

[29] M. Zignani, S. Gaito, G.P. Rossi, Follow the "Mastodon": structure and evolution of a decentralized online social network, in: 12th International AAI Conference on Web and Social Media, AAAI Press, 2018, pp. 541–550.

[30] A. Rossaro, C. Surquin, Supporting the Portability of Profiles Using the Blockchain in the Mastodon Social Network, Ph.D. thesis, Louvain School of Engineering, 2019.

[31] A. Raman, S. Joglekar, E.D. Cristofaro, et al., Challenges in the decentralised web: the Mastodon case, in: 2019 Internet Measurement Conference; 21-23 Oct 2019; Amsterdam, Netherlands, ACM, New York, NY, USA, 2019, pp. 217–229.

[32] M. Zignani, C. Quadri, S. Gaito, et al., The footprints of a "Mastodon": how a decentralized architecture influences online social relationships, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops; 29 Apr-2 May 2019; Paris, France, IEEE, Piscataway, NJ, USA, 2019, pp. 472–477.

[33] W. Entriken, D. Shirley, J. Evans, N. Sachs, Erc-721 non-fungible token standard. https://eips.ethereum.org/EIPS/eip-721, 2018.

[34] P. Cuffe, The role of the ERC-20 token standard in a financial revolution: the case of initial coin offerings, in: IEC-IEEE-KATS Academic Challenge, 2018.

[35] N. Copel, T. Ater. Dav White Paper, DAV Foundation, 2017. Available online: https://static.coinpaprika.com/storage/cdn/whitepapers/6535565.pdf.

[36] B. Guidi, A. Michienzi, L. Ricci, Data persistence in decentralized social applications: the IPFS approach, in: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC); 9-12 Jan 2021; Las Vegas, NV, USA, IEEE, Piscataway, NJ, USA, 2021, pp. 1–4.