

Decentralized Social Media Applications as a Service: a Car-Sharing Perspective

Anandhakumar Palanisamy
bitYoga
Stavanger, Norway
anand@bityoga.com

Mirsat Sefidanoski
University of Information Science and Technology
Ohrid, North Macedonia
mirsat.sefidanoski@cse.uist.edu.mk

Spiros Koulouzis
Informatics Institute
University of Amsterdam, Netherlands
S.Koulouzis@uva.nl

Carlos Rubia
Agilia Center
Seville, Spain
carlos.rubia@agiliacenter.com

Nishant Saurabh, Radu Prodan
Institute of Information Technology
University of Klagenfurt, Austria
nishant, radu@itec.aau.at

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 to fulfil the 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 use case application, 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 application and illustrate through real snapshots how the different ARTICONF tools support it in a simulated scenario.

Index Terms—Social media, car-sharing, decentralization, blockchain.

I. INTRODUCTION

Social media platforms are key technologies for next generation connectivity [2], [14], [17]. 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 the propagated content. This is problematic when data breaches are a regular phenomenon at the hands of centralized intermediaries and requires innovative solutions at the user level (i.e. consumers, prosumers, businesses) and the underlying social media environment to facilitate global reach, improved trust, and decentralized control and ownership [3], [12].

The ARTICONF project (<https://articonf.eu/>) 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, to fulfil the privacy, robustness, and autonomy related

promises that proprietary social media platforms have failed to deliver so far. Namely, its goals are to:

- Simplify the creation of open and agile social media ecosystem with trusted participation using a two stage permissioned blockchain architecture;
- Automatically detect interest groups and communities through semantic contextualization and abstraction [8] of dynamic and diverse social media facets;
- Elastically autoscale time-critical social media applications through an adaptive orchestrated Cloud-Edge-based infrastructure meeting application runtime requirements;
- Enhance monetary inclusion in collaborative models through cognitive and interactive visualization.

In this paper, we describe the ARTICONF architecture applied on a car-sharing use case, as a new collaborative model [19] and alternate Software-as-a-Service (SaaS) solution to private car ownership. The new car-sharing model allows customers to temporarily use a vehicle (on-demand) at a variable fee, charged depending on the distance travelled or time used. This peer-to-peer (P2P) 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 CO₂ emissions per-city (due to fewer vehicles) and per-vehicle (encouraging the use of electric or hybrid cars), and reducing traffic and parking congestions.

The paper has the following structure. The next section outlines the ARTICONF decentralized social media platform, with focus on its modular tool architecture and use cases. Section III describes the design of a car-sharing use case on top of the ARTICONF platform. Section IV gives implementation details of the car-sharing application in a simulated use-case scenario. Finally, Section V concludes the paper.

II. ARTICONF ARCHITECTURE

To provide high level of privacy, robustness and autonomy, while addressing issues of trust, time-criticality and democratization, the ARTICONF shares a decentralized architecture with four building block tools: 1) Trust and Integration

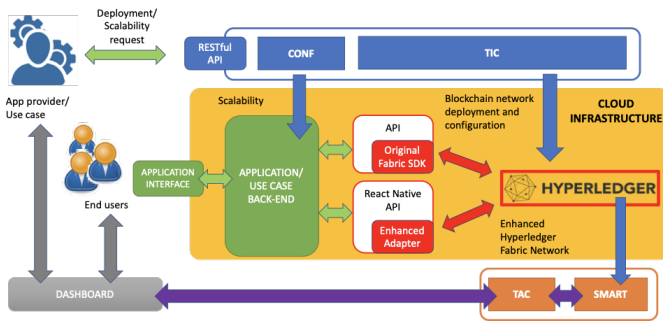


Fig. 1. ARTICONF integrated tool architecture.

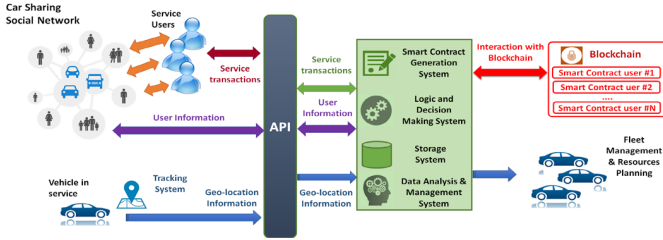


Fig. 2. High-level car-sharing use case architecture.

Controller (TIC); 2) Co-located and Orchestrated Network Fabric (CONF); 3) Semantic Model with self-adaptive and Autonomous Relevant Technology (SMART); 4) 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 [4] 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 behaviour of the application, decision-making and infrastructure, while TIC provides the underlying permissioned blockchain that verifies and records the user transactional activities using anonymous identities. Finally, TAC gives intelligent insights and relevant information about the platform to use case providers and end-users through an interactive dashboard.

III. CAR-SHARING USE CASE

Last decade saw an increased popularity of car-sharing adoption resulting into an explosion of providers offering shared-mobility services in the market, the most popular being Uber [5] and ZipCar [9]. In contrast to Uber, ZipCar and several centralized shared-mobility providers, the ARTICONF’s car-sharing model creates a new decentralized platform based on blockchain [6], [13] and smart contracts to face this growing market and to meet the appropriate service requirements related to user privacy, trust, resilience and cost. The car-sharing use case does not cover a single business model, but allows direct user interaction (B2C), or with car sharing companies and fleet providers (B2B).

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

a) *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 may have various purposes, such as showing to operators vehicle damages, fuel level, or battery status, or asking other users for a route. Customers anonymously 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 fulfilment without penalties, reporting real network and car-sharing issues, or publishing contents and service policies.

b) *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 contract breaches. The customers can then use the cars without financial worries since the money are available and visible only in the smart contracts, executed only upon triggering certain prerequisites.

c) *Geolocation monitoring service*: [16], installed on each operating vehicle is in charge of tracking and verifying the real-time location and clauses of the smart contract. In the validation process, the geolocation service tracks each vehicle and user smartphone to resolve the smart contracts. In addition, the smart contracts have an escrow that depends on the users’ reputation, measured according to the service policies.

On top of these services, advanced artificial intelligence (AI)-based reward mechanisms and fleet allocation algorithms aim to provide economic benefits for users and companies by eliminating fleet idle times, reducing the impact of external events and enhancing the user experiences [10], [18].

IV. PROTOTYPE IMPLEMENTATION

This section describes the use case prototype implementation, preparation and integration with the ARTICONF tools.

A. Car-sharing prototype scenario

Fig. 3 shows the car-sharing mobile application interface providing four interfaces to: (1) import or create a certificate pertaining to a travel, (2) import the travel details, (3) display offers provided by different car owners specific to an imported travel plan, and (4) provide user access for uploading content (e.g. images, videos, comments). The following sections describe the car-sharing data models and the realization of decentralized car-sharing application through its interaction with the ARTICONF tools.

B. Car-sharing data model

To validate the car-sharing prototype, we integrated into its mobile application interface a random data generator and API mocking tool called Mockaroo (<https://www.mockaroo.com/>) that enables the creation of realistic test data in CSV, JSON, and SQL formats with an upper limit of 1000 rows. We used Mockaroo to generate 1000 cars, 1000 users and 1000 offers

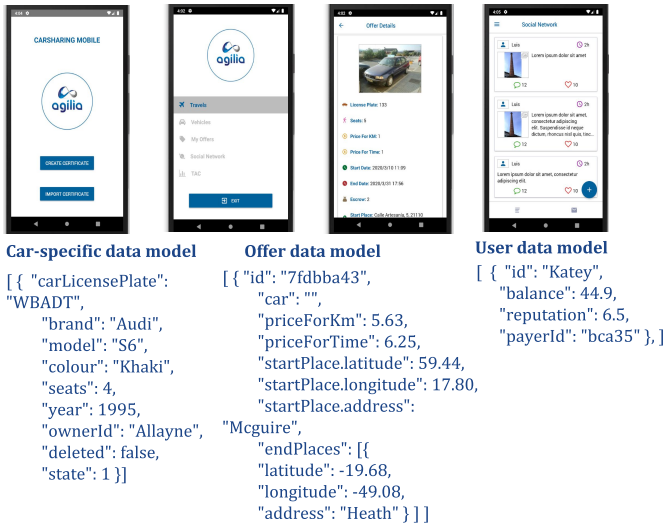


Fig. 3. Car-sharing mobile application interface and its associated data model.

according to a schema presented in Fig. 3. The test data set is openly available in JSON format on the Zenodo archive (<https://doi.org/10.5281/zenodo.3689521>) for reproducibility.

- a) *Car-specific data schema*: comprises license plate, brand, model, color, owner identifier, number of seats, manufacturing year and availability status.
- b) *Car offer data schema*: comprises offer identifier, car license plate, price per km, price per time, start coordinates (latitude and longitude), start and end locations (addresses).
- c) *User data schema*: contains four fields: user id, balance, payment source (generated using Paypal RESTful API and reputation (i.e. numeric score).
- d) *Travel data schema*: combines car, offer and user data models, instantiated by users, registered onto the blockchain and used by SMART and TAC for analytic purposes.

C. TIC

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. To achieve this, TIC identifies the following sequential phases associated with the deployment of a sample organization in the car-sharing consortium.

1) *Architecture planning*: The first phases in TIC deployment involves the identification of a reliable and scalable architecture for an organization participating in the car-sharing consortium, whose requirements may depend on various factors, such as network traffic or geographical distribution. Afterwards, TIC delegates the allocation of resources to CONF. Fig. 4 shows a sample prototype implementation architecture of a car-sharing organization comprising of one manager machine, four worker machines and two Gluster File System (GlusterFS) machines associated with two Gluster volumes. GlusterFS is an open source distributed file system that scales out in a building-block fashion to store multiple petabytes of data [7]. The two GlusterFS machines act as a Cloud storage,

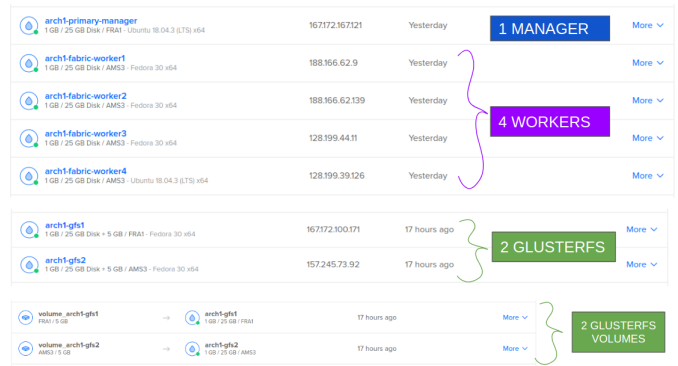


Fig. 4. TIC car-sharing resource allocation architecture.

TIC CONFIGURATION TOOL

MASTER MACHINES WORKER MACHINES GLUSTERFS MACHINES HYPERLEDGER SERVICES

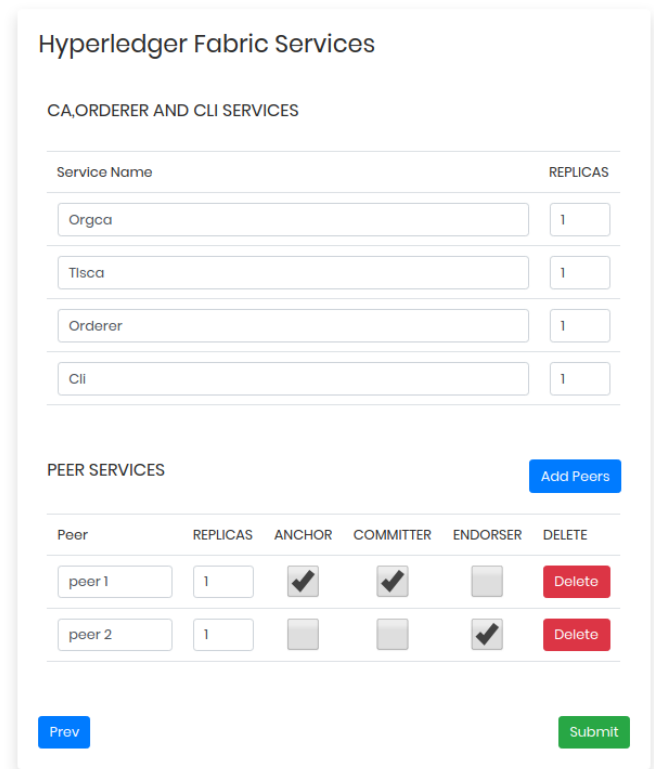


Fig. 5. TIC configuration tool, with an orderer service of replica 1, two CA services of replica 1, CLI services of replica 1, two peer services of replica 1 with peer1 acting as anchor and committer, and peer2 acting as endorser.

which stores the common credentials and certificates needed for the blockchain network, as well as any other data required by the car-sharing application.

2) *Update configurations*: A configuration interface displayed in Fig. 5 allows to configure the resource allocation by specifying the number of managers, workers and Cloud storage machines, as illustrated in Fig. 4. This interface also facilitates the configuration of the hyperledger fabric services, like the

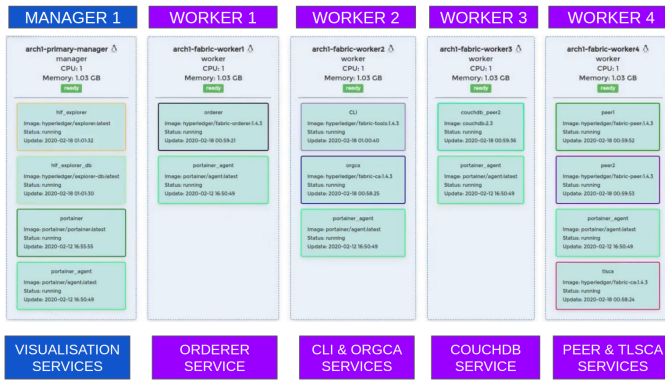


Fig. 6. Docker visualizer of a TIC deployment.

number of peers, orderers, and type of peers (i.e. endorser, committer, anchor) for the blockchain network.

3) *Deployment*: TIC facilitates the automatic deployment of the hyperledger fabric blockchain services through a modular deployment with a single click, achieved using open-source Ansible playbooks [20]. The deployment machine needs appropriate permissions to access the remote machines and triggers a linear execution of twelve Ansible playbooks with three functionality groups.

a) *Blockchain platform setup*: Four playbooks prepare the allocated resources with all the prerequisites needed to setup a blockchain platform for the business configuration:

- Install all fabric blockchain prerequisite software, depending upon operating system type (e.g. Debian, CentOS);
- Pull the required docker images of the hyperledger fabric services, deployed as plug-and-play modular components;
- Deploy the GlusterFS clusters acting as Cloud storage for the blockchain network;
- Spawn Docker services according to business configuration.

b) *Hyperledger fabric services*: Four playbooks deploy the hyperledger fabric Docker service in the network.

- Spawn two certification authority services: 1) ORGCA that generates the membership service providers (MSP) for the agents (i.e. peers, clients, administrators) to interact with the blockchain network, and 2) TLSCA that generates the MSPs for the same agents to establish an internal or external TLS communication;
- Spawn the ordering service that implements the Raft [15] consensus algorithm;
- Deploy the peer services according to the business configuration, for example one peer acting as anchor and committer and one peer acting as endorser;
- Deploy a Command-Line Interface (CLI) service to install or instantiate chaincodes (smart contracts) onto the blockchain.

c) *Visualization services*: Two playbooks spawn the portainer and portainer-agent, and two other playbooks the hyperledger explorer blockchain visualization services. Fig. 6 represents a snapshot of the various Docker hyperledger visualization services deployed across different resources. TIC deploys the visualization services on the manager machines and

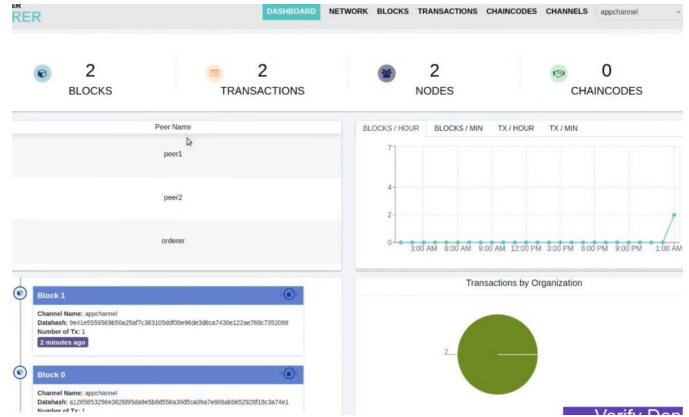


Fig. 7. TIC hyperledger explorer deployment verification.

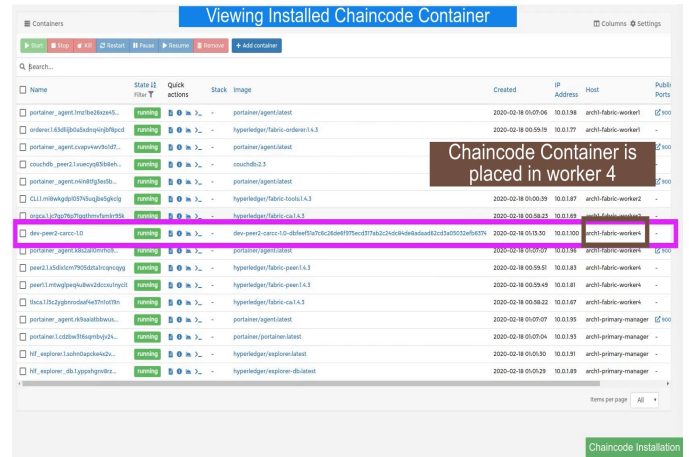


Fig. 8. TIC chaincode visualization.

the hyperledger fabric services across the worker machines, as follows: the orderer service on worker machine 1, the CLI and ORGCA services on worker machine 2, the CouchDb service storing the blockchain ledger on worker machine 3, and the peer and TLSCA services on worker machine 4.

d) *Deployment verification*: To verify the deployment of a blockchain network as per business configuration, the hyperledger explorer service provides visualizations of blockchain metrics like number of participating nodes, transactions, blocks and chaincodes, as shown in the Fig. 7.

e) *Car-sharing chaincode installation*: TIC facilitates the installation and instantiation of the car-sharing chaincodes through a command-line interface service, verified either using the hyperledger explorer or the Docker visualization service shown in Fig. 8. These smart contracts implement the business logic behind the car-sharing use case.

f) *Test node SDK*: The TIC tool supports the integration of mobile and web apps with the blockchain network through a NodeSDK and provides basic helper libraries for registering a user to the blockchain, querying or invoking a chain code for easy integration, and verifying the car-sharing prototype implementation. Similarly, it can dynamically configure and add

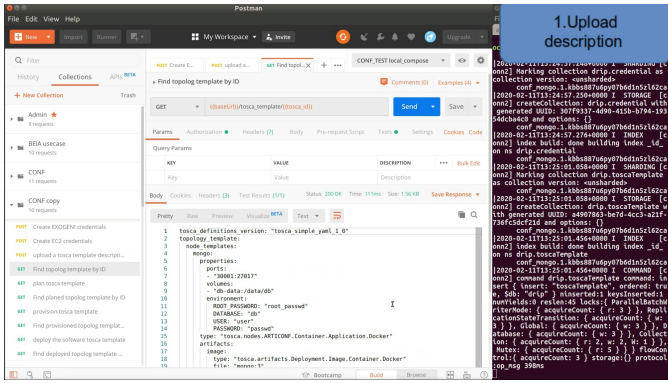


Fig. 9. CONF's RESTful API.

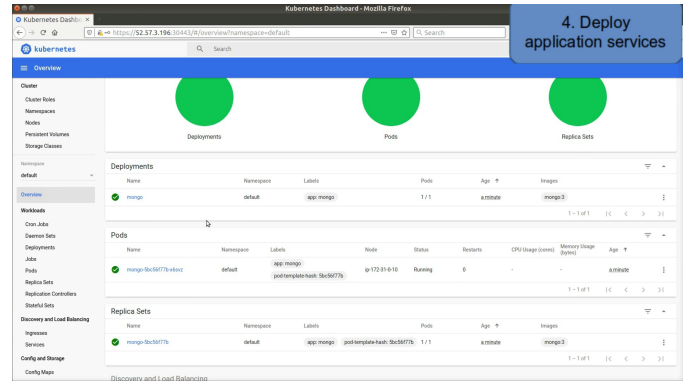


Fig. 10. CONF Kubernetes application dashboard.

organizations to the car-sharing consortium, as the business grows across geographical areas.

D. CONF

CONF provides adaptive infrastructure planning, provisioning, configuration, deployment and monitoring for social media applications. CONF intelligently plans and provisions services based on abstract application service requirements, operational conditions at the infrastructure level, and time-critical event triggering [11].

We describe a concrete car-sharing scenario with improved reaction time during peaks hours, invoked using the Postman (<https://www.postman.com/>) 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 application 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, as shown in the bottom-center part of Fig. 9.

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 VMs 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 application on a Kubernetes cluster. Fig. 10 shows the Kubernetes dashboard together with the deployed metrics database and the application state (e.g. state of deployments, services and pods).

E. SMART

SMART exploits the experiential anonymous user activities embedded in the TIC blockchain as immutable traces, such as the data models shown in Fig. 3. SMART takes advantage of trace comparison and retrieval to provide an effective and

quick adaptation to contextualized activity traces represented at different levels of abstraction in two steps.

a) *Semantic contextualization and abstraction:* SMART represents the activity traces of the car-sharing application as a multilayer networked graph, consisting of six identified layers presented in Fig. 11:

- *L1-Reputation* represents the credit points earned by each user for using the car-sharing application;
- *L2-Users* represent anonymous unique user identifiers;
- *L3-StartingPoints* is source location of each user travel;
- *L4-Time* shows timestamp at each travel source location;
- *L5-Destination* represent destination of each user travel;
- *L6-TravelPrices* is the price per-travel paid by users.

Each layer in Fig. 11 represents a context embedded with multiple agents, where each agent is a car-sharing user with a specific role, defined as 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. 11. SMART performs semantic linking at regular time intervals to obtain multiple multilayered graphs, stored in its knowledge base. This provides an abstract decomposition of changing agent roles into temporal stages, where each stage represents a fixed time interval. In the context of the *L4-Time* layer, different stages represent varying travel timestamps of an agent with unique identifiers.

b) *Community detection:* SMART fetches the multilayered contextualized graphs across temporal stages as input for community detection. Initially, SMART utilizes a density based spatial clustering algorithm (DBScan) to find similarities among different users in each layered context through a distance function and a minimum number of neighbors required as a unique community. Fig. 12 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 clustering algorithm with distance function models. Further-

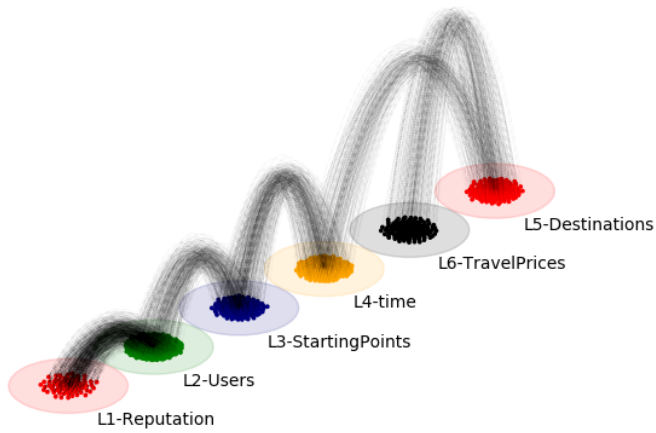


Fig. 11. Multilayer contextualization for car-sharing.

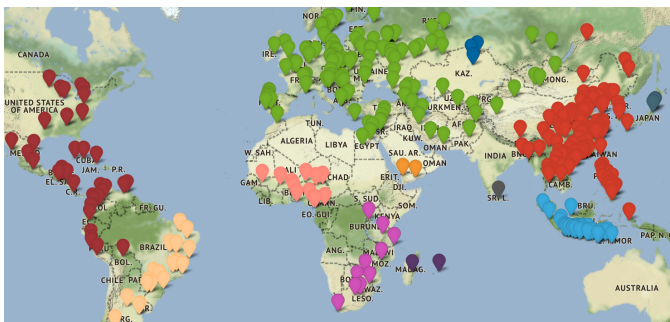


Fig. 12. Clustering for L3-starting point layer shown in Fig. 11.

more, SMART applies means shift clustering algorithms to find similarities between cross-contextual layered users, where each layer has a varying density, due to DBScan’s inability of detecting clusters over layers with varying density. This combination of clustering per context and cross-context analyzed across temporal stages allows SMART to predict demand and travel patterns over a period of time, and enables car-sharing providers to adjust with changing user behavior.

F. TAC

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 through three services.

a) *Geospatial service*: handles the gathering, display and manipulation of all data consisting of longitude and latitude as information [1]. For example, the coordinate map in Fig. 13 shows the starting positions of travels that made the most revenue. This visualization required metric aggregations on the data field containing geographical information (latitude and longitude) about the travel starting positions and counted the total spending based on the positions.

b) *Temporal aggregation*: handles the data changing over time. Fig. 14 depicts the time series histogram of the aggregation data, which analyzes the change in maximum, minimum, average and median price over time.

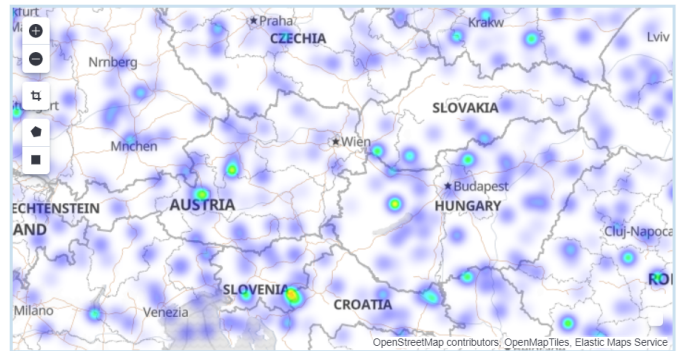


Fig. 13. Coordinate map by revenue per starting position of travel.

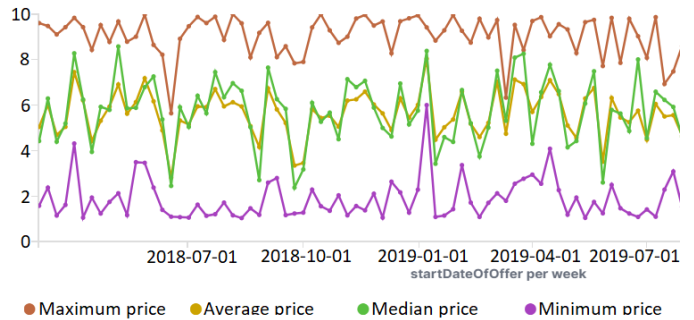


Fig. 14. Rental price across time.

c) *Visualization service*: implemented on top of three open-source software tools known as the elastic stack (Elasticsearch, Kibana and Logstash) dynamically identifies and aggregates specific data fields based on diverse provider, prosumer and consumer requirements. Fig. 15 shows the aggregated number of seats across diverse car brands visualized as a heat-map. Similarly, the bar-chart in Fig. 16 illustrates the top five car brands and the corresponding car types that achieved the most revenue.

V. 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.

a) *Transparent and decentralized infrastructure creation and control (TIC)*: through a novel permissioned blockchain network, supporting anonymous identities, offering users a secure, permanent and unbreakable link their personal data;

b) *Improved and trusted participation (TIC and SMART)*: by eliminating malicious actors in participatory exchanges, with improved collaboration, trust, and operational costs, without violating users’ privacy;

c) *Democratic and tokenized decision-making (SMART)*: through collective decentralized reasoning that improves the

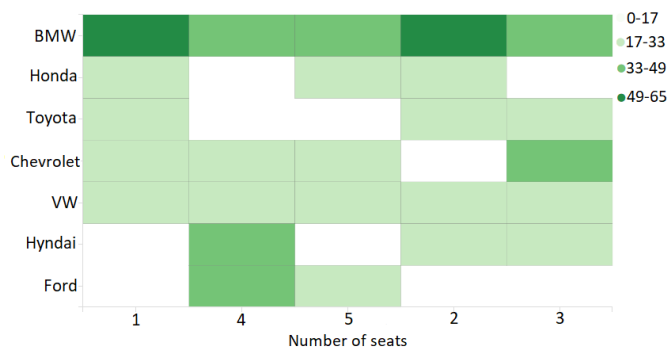


Fig. 15. Heat-map showing the number of seats in a car brand.

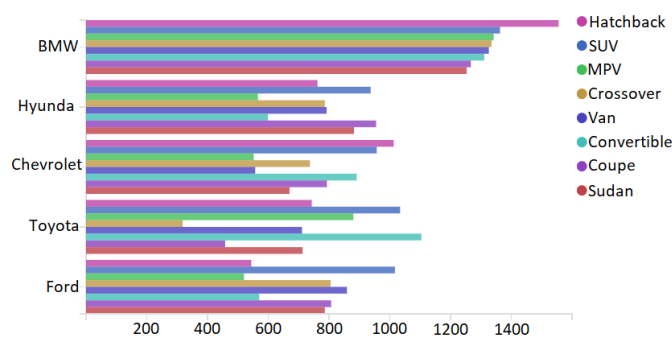


Fig. 16. Top five car brands and type of car with most revenue.

content quality by smartly finding interest groups and incentivising users for their participation;

d) *Elastic resource provisioning (CONF)*: to improve efficiency for customizing, deploying and controlling distributed peer-to-peer and Cloud virtual infrastructures required by time-critical social media applications;

e) *Cognitive guided analytics for improved collaborative economy (TAC)*: that inject 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 use case, as a new collaborative model and alternate SaaS 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 allows car-sharing customers to engage in trustful and secure interactions for renting a vehicle at a variable fee, charged depending on the distance travelled or time used, while keeping complete control over their personal data.

The project plans to validate its results on three further industrial applications targeting crowd journalism with news verification, video dialog and smart energy sharing.

ACKNOWLEDGEMENT

ARTICONF receives funding from the European Union's Horizon 2020 research and innovation program under grant agreement number 825134.

REFERENCES

- [1] D. Capeska Bogatinoska N. Marina A. Karadimce, N. Paunkoska (Dimoska) and A. Nandal. Guided analytics software for smart aggregation, cognition and interactive visualisation. In *International Conference on Innovations in Computational Intelligence and Computer Vision (ICICV-2020)*. Springer, 2020.
- [2] Gil Appel, Lauren Grewal, Rhonda Hadi, and Andrew T Stephen. The future of social media in marketing. *Journal of the Academy of Marketing Science*, 48(1):79–95, 2020.
- [3] Ghazaleh Beigi and Huan Liu. "identifying novel privacy issues of online users on social media platforms" by ghazaleh beigi and huan liu with martin vesely as coordinator. *ACM SIGWEB Newsletter*, (Winter):1–7, 2019.
- [4] Christian Cachin et al. Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310, page 4, 2016.
- [5] Sarah Cannon and Lawrence H Summers. How uber and the sharing economy can win over regulators. *Harvard business review*, 13(10):24–28, 2014.
- [6] T.C. Dao, B.M. Nguyen, and B.L. Do. Challenges and strategies for developing decentralized applications based on blockchain technology. In *International Conference on Advanced Information Networking and Applications*, pages 952–962. Springer, 2019.
- [7] Alex Davies and Alessandro Orsaria. Scale out with glusterfs. *Linux Journal*, 2013, 11 2013.
- [8] Audrey Fertier, Anne-Marie Barthe-Delanoë, Aurélie Montarnal, Sébastien Truptil, and Frédéric Bénaben. A new emergency decision support system: The automatic interpretation and contextualisation of events to model a crisis situation in real-time. *Decision Support Systems*, page 113260, 2020.
- [9] Myra Hart, Michael J Roberts, and Julia D Stevens. *Zipcar: refining the business model*. Harvard Business School Pub., 2005.
- [10] P. Helebrandt, M. Bellus, M. Ries, I. Kotuliak, and V. Khilenko. Blockchain adoption for monitoring and management of enterprise networks. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1221–1225. IEEE, 2018.
- [11] S. Koulouzis, T. Carval, P. Martin, B. Grenier, Y. Chen, J. Heikkinen, and Z. Zhao. Dynamic Optimization for Time-critical Data Services: A Case Study in Euro-Argo Research Infrastructure. In *EGU General Assembly Conference Abstracts*, volume 20 of *EGU General Assembly Conference Abstracts*, page 16012, apr 2018.
- [12] George Loukas and Charalampos 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: 8th International Conference, e-Democracy 2019, Athens, Greece, December 12-13, 2019, Proceedings*, volume 1111, page 196. Springer Nature, 2020.
- [13] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2019.
- [14] Tahir M Nisar, Guru Prabhakar, and Lubica Strakova. Social media information benefits, knowledge management and smart organizations. *Journal of Business Research*, 94:264–272, 2019.
- [15] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, USENIX ATC'14, page 305–320, USA, 2014. USENIX Association.
- [16] A. Patel, N. Shah, T. Limbasiya, and D. Das. Vehiclechain: Blockchain-based vehicular data transmission scheme for smart city. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 661–667, 2019.
- [17] Ashish K Rathore, P Vigneswara Ilavarasan, and Yogesh K Dwivedi. Social media content and product co-creation: an emerging paradigm. *Journal of Enterprise Information Management*, 2016.
- [18] D. Rojas-Rueda, A. de Nazelle, O. Teixidó, and M.J. Nieuwenhuijsen. Replacing car trips by increasing bike and public transport in the greater barcelona metropolitan area: A health impact assessment study. *Environment International*, 49:100 – 109, 2012.
- [19] V. Valaštín, K. Košťál, R. Bencel, and I. Kotuliak. Blockchain based car-sharing platform. In *2019 International Symposium ELMAR*, pages 5–8, Sep. 2019.
- [20] Moshe Zadka. *Ansible*, pages 139–145. 06 2019.