A Decentralized Trust-minimized Cloud Robotics Architecture

Alessandro Simovic, Ralf Kaestner and Martin Rufli

***Abstract*— We introduce a novel, decentralized architecture facilitating consensual, blockchain-secured computation and verification of data / knowledge. Through the integration of (i) a decentralized content-addressable storage system, (ii) a decentralized communication and time stamping server, and (iii) a decentralized computation module, it enables a scalable, transparent, and semantically interoperable cloud robotics ecosystem, capable of powering the emerging internet of robots.**

I. Introduction

With the rapidly growing number and interconnectivity of robotic devices, scalability of the underlying systems becomes a key concern. This particularly affects the interaction / cooperation among heterogeneous sets of robots; and indeed the very sharing of data. In such a setting, knowledge can increasingly no longer be effectively and verifiably curated by existing, efficient central entities alone. Yet no automated, robust and decentralized alternatives are available today. Our architecture addresses this void.

II. Architecture Overview

Our architecture is composed of a set of interlocking technologies enabling other applications to create, maintain and share an ever increasing corpus of trustable knowledge within a decentralized network. As key design choices, it implements interfaces to the following storage, communication and compute modules:

1) Decentralized Content-addressable Storage: The architecture's storage is designed to hold the entirety of accumulated collective data and knowledge. This ranges from raw, uninterpreted sensor data via model representations to semantically encoded knowledge and software modules. As its storage system, our framework interfaces to IPFS [1], a decentralized Content-addressable storage (CAS) system. In CAS files are addressed by their content hash rather than location. As a key benefit of this design files become highly available and downloads from untrusted sources trivially verifiable.

2) Trust-minimized Decentralized Communication and Time Stamping: CAS does not render stored data automatically discoverable, however. Since we serialize all data into files of suitable format within IPFS, it suffices to communicate only the hash of a file to other interested robots. The integration of a blockchain in this setting results in a single ledger of referenced tamper-proof and cryptographically signed hashes that can be monitored by connected robots for newly ingested data of interest.

A. Simovic is with the University of Zurich (e-mail: simovic@ifi.uzh.ch). R. Kaestner and M. Rufli are with IBM Research – Zurich.

3) Trust-minimized Decentralized Computing: The persistent need for verification of other robots' computations all the way back to the ingested raw data becomes quickly uneconomical in larger decentralized cloud robotics settings, however. For this purpose our framework interfaces with Ethereum [2], a universal decentralized computer, via a set of Decentralized Application (Dapp). Due to performance limitations inherent to that design, heavy computations – including data manipulations – cannot be executed on chain. In our framework, heavy-weight code execution is thus performed off-chain on device hardware, which requires the software modules to be universally executable whilst yielding reproducible output for later on-chain verification via e.g., replicated execution. Differing operating system configurations make universal computation a challenging objective, however. Containerization addresses this issue. Within our approach we distribute software modules in the form of Docker [3] runtimes. Analogous to other forms of data and knowledge, these containers may be stored in IPFS for easy access and distribution.

III. Application

One illustrative example application we implemented in our framework is a fully decentralized, transparent and verifiable knowledge base for robots. The application can be thought of as an extension to RoboEarth [4] with the following benefits: knowledge generation, storage and manipulation can be trusted and reused among robots even when they themselves do not trust each other. In this setting the entirety of sensor data, derived knowledge and software components is stored in a CAS such as IPFS. Containerized software programs are geared at the manipulation and enrichment of that knowledge. The execution thereof may be routed through Ethereum, making their output visible to and transparently verifiable by others with no further computations necessary.

This lays the grounds for a virtuous cycle characterized by trustable knowledge reuse and buildup, and thus the emergence of an internet of robots.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688652.

References

[1] IPFS. [Online]. Available: https://ipfs.io
[2] Ethereum. [Online]. Available: https://www.ethereum.org
[3] Docker. [Online]. Available: https://www.docker.com
[4] D. Hunziker *et al.*, "Rapyuta: The RoboEarth cloud engine," in *IEEE Int. Conf. Rob. & Autom. (ICRA)*, 2013.