A Decentralized Trust-minimized Cloud Robotics Architecture

Alessandro Simovic, Ralf Kaestner and Martin Rufli

Overview

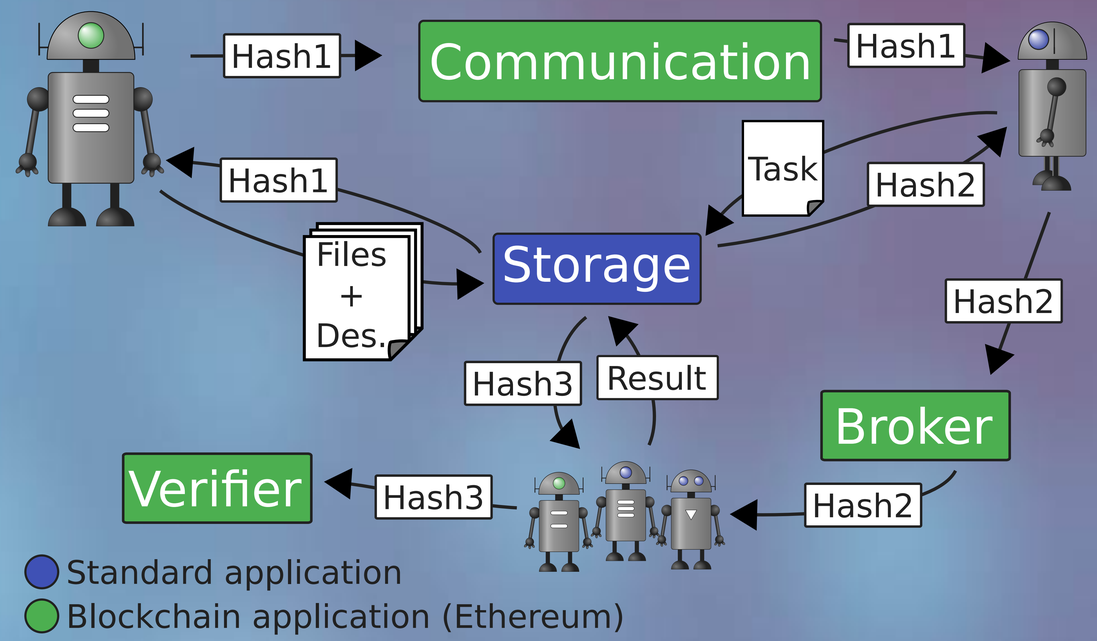
We introduce a decentralized architecture facilitating consensual, block chain secured computation and verification of data/knowledge. Through the integration of a decentralized content addressable storage system; a decentralized communication and time stamping server; and a decentralized computation module, it enables a scalable, transparent, and semantically interoperable cloud robotics ecosystem, capable of powering the emerging internet of robots.

Operation

1.Knowledge manipulation task (incl. participation rules and reward payout schedule) is sent to verifier
2.Agents register with verifier as prover candidates and stake a security deposit
3.Provers are chosen randomly by verifier
4.Provers perform computations yielding a result R
5.Provers chose a secret nonce to obfuscate result b = h(R, nonce)
6.Provers submit their respective blinded result b to the verifier
7.Provers reveal their nonce to verifier together with the result R
8.Verifier validates all blinded results from step 6
9.Verifier elects final result (and pays out rewards)

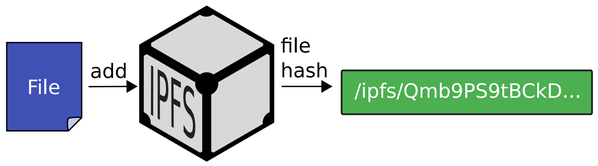
Applications

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 benefit that knowledge generation, storage and manipulation can be trusted and reused among robots even when they themselves do not trust each other.

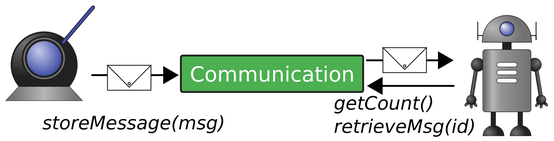


Components

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.



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 block chain 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.



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. For this purpose our framework interfaces with Ethereum [2], a universal decentralized computer, via a set of decentralized applications. Due to performance limitations inherent to that design, heavy computations cannot be executed on chain. In our framework, heavy-weight code execution is performed off-chain on device hardware, which requires the software modules to be universally executable whilst yielding reproducible output for later on-chain verification. 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.

References and Links

[1] IPFS: https://ipfs.io
[2] Ethereum: https://www.ethereum.org
[3] Docker: https://www.docker.com
[4] RoboEarth: https://roboearth.ethz.ch

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