***GRAVITY:* Project Overview**

Project Description

The Economic Space Agency is developing Gravity — a distributed computing architecture that emphasizes both resilience and interoperability, enabling a new way to create smart contracts. It allows for private contracts or contracts among small groups to be as secure, resilient, verifiable, available, and unforgeable as public contracts are on Ethereum, while also allowing data and transaction history to be kept private. Gravity is built from the ground up on object capabilities, a simple, mature and secure programming model. On this foundation, Gravity creates decentralized, verifiable applications modeled on blockchain cryptography and distribution technologies.

Technology

World Computing Fabric

Bitcoin introduced blockchain technology and cryptocurrency to the world. It created a decentralized network of trust, optimized for resiliency, verifiability, and anonymity. Ethereum then built upon the decentralized architecture of Bitcoin, encoding on the blockchain an executable language with which to build applications. It is fundamentally based on the computing platform of the "World Computer", where every node on the network processes every line of application code being executed in the system. This replication has tremendous benefits in terms of transparency and fault tolerance. However, the World Computer architecture is logically centralized: every node is acting on the same ledger, and duplicates the entire history of transactions. This introduces transaction cost, redundancies, and limitations that are inefficient and inconvenient for many applications, which would benefit from more flexibility than this architecture offers.

Gravity offers a platform for building interoperating networks of decentralized computers — a World Computing Fabric. This World Computing Fabric is a modular, object capability-oriented architecture for building resilient, interoperable, verifiable networks of virtual machines (VMs). The modularity of the Gravity platform offers more flexibility to developers and users than existing blockchain-based platforms, and enables the application domain of smart contracts to grow into new spaces. Gravity places what would be considered core functionality in most other blockchain systems into contracts, which can be read and modularly chosen to fine-tune an app’s functionality.

Features

- Interoperability

  Gravity enables multi-blockchain space integration seamlessly. Alongside the built-in transmittable authorization of object capabilities, Gravity can transmit the protocol types and specifications necessary to interact with a remote contract's API, eliminating the manual work of interoperating with remote services. Building on this, we are able to create higher-order abstractions such as the Remote Access Object that make interoperating with multiple remote heterogeneous contracts nearly as straightforward as programming to a local library.

  Gravity contracts are not restricted from communicating with external systems, as with Ethereum and Bitcoin. A developer could write their own protocol bridge linking any other blockchain system, such as Ethereum, to the Gravity Public Network. This allows for VMs within the Gravity ecosystem to exchange value with a range of other DAOs built on other protocols, from Bitcoin sidechains, to Ethereum, and beyond.

- Public and Private Contracts

  Gravity's logically decentralized post-blockchain protocol enables distributed atomic transactions among an interdependent network of contracts. In this approach, private agents and private contracts are possible, and the parties to a contract are in control of who can participate in its execution and verification. If an external authority is used for verification, the transaction data in the private contract can be revealed only to the verifier, who reveals only whether the given transactions are valid. This enables verifiability and non-repudiability (proof) on interactions among autonomous agents on the network, while preserving privacy, unforgeability, and auditability.

- Principle Of Least Authority

  The notion of capability-based security brings with it the principle of designing user programs such that they directly share capabilities with each other according to the principle of least authority. This enables fine-grained, well-tailored authority rights to be granted between VMs to create secure yet renegotiable boundaries for interaction.

- Data Sovereignty and Privacy

  User data control and self-sovereign identity and sharing are a natural result of the ability to create private capabilities-enabled networks on top of the Gravity Public Network. Following the Principle Of Least Authority, users can precisely calibrate how they wish to share their data and with whom, enjoying the flexibility of choosing which data to make public, and which to keep privately shared among a select set of parties. For instance, a network of AI's will be able to securely and privately share experience-derived knowledge graphs with one another without sharing the underlying data (e.g. photos, user information, voice recordings) that informed the knowledge graph's construction.

- Modularity

Flexible network capabilities make it possible to create custom topologies, consensus algorithms, and failure recovery methods. For some applications, speed is paramount. For other applications redundancy, recoverability, and verification may be more important. The choice of network topology, consensus algorithm, and persistence modules allow developers to tune the properties of their application.

- Scalability
  As a result of its flexible network capabilities providing the possibility to securely host both private and public contracts, Gravity allows much more scalable and resilient applications to be built on top of it. It also provides infrastructure for a much faster network in which thousands of *committed* transactions can occur every second, which is not possible with the logically centralized blockchain technology in which every node in the network participates in maintaining the single public state of the network, and wait for an eventually consistent result.

Roadmap 2017

Goal: develop the Gravity Contract and enable the application ecosystem around Gravity.

- Economy — Economic logic to make the Gravity Fabric work long-term. Issuance and use of Gravity tokens, e.g. within the Gravity Host Fabric.
- Library — All of the computational modules that, put together, allow the fabric to function.
- Market — A reputation layer for computational offerings — e.g. availability, processing power, network throughput — that is used to allocate a balanced set of distributed resource pools for the Gravity Host Fabric.
- Distributed Exchange — A fully distributed exchange platform, with no centralized control, that allows full liquidity and exchange of all tokens and any other value-captured forms with each other, as well as creation of any form of derivative financial instrument.

The bulk of the work initially is in the development of the Gravity Library, the computational modules upon which the functionality of the Gravity Contract rests.

1. Smart Contract Infrastructure

- Messaging — Object capabilities, object reference graph isolation layer
- Persistence — Serializing the object reference graph of the entire VM; saving and retrieval. Storing diffs of each state change into a log.

2. Distributed Contracts

- Networking — Inter-Process Communication for VMs running on the same machine or spawned from the same parent VM and running on the same hardware. Service discovery and routing for accessing remote contracts over the internet.
- Security — The cryptographic primitives to enable contract verification. VM state content hashing and encoding into object metadata. Cryptographic signing of the state hash for proving authorship, and auditing.

- Protocol Transmission — The ability to transmit all data necessary to create a Remote Access Object to fully interact with the capabilities that a contract is offering to the network.

3. Gravity Library

- Distribution Contracts — Allow multiple hosts to collaborate to verify correctness in contract execution, and increase availability of a particular contract.
    - Replication Contract — This is the basic Agency. Full replication by all nodes.
- Verification Contract — The methods of verification. Includes 100% verification and signing by all nodes in the network, x% verification, random verification, and the use of external verifiers.
- Consistent Transactions — Necessary to achieve consistency of state change in the execution of multiple dependent transactions across several (often remote) contracts. This is accomplished using the recursive 2-phase commit design pattern.

To dive deeper in all of these topics, please reference the Gravity White Paper.