TOM3: Distributed Compute Orchestration Framework Optimized for Web3

Abstract
Cloud computing is a model that allows network access to a flexible and scalable pool of shared physical or virtual resources, with on-demand self-service provisioning and management. On the other hand, distributed computing is a framework that allocates data storage and processing tasks across various devices or systems, instead of relying on a solitary central unit.

Although cloud computing generally functions as promised, we've discovered that inefficiencies continue to affect the system due to centralization by the major cloud providers such as AWS, GCP, Azure, OCI, etc. The offerings from leading cloud providers are functional, yet they are constrained by issues that can be addressed today through innovations in container technology and a robust token economy. The aim of this whitepaper is to present our proposal for a cloud services platform named TOM3, the first global P2P cloud computing infrastructure in the world.

We envision a future where the world's cloud infrastructure is decentralized and shared among all cloud service providers; a platform that deploys and repurposes data center computing in a safe, rapid, and transparently priced fashion. Services are offered in a unified yet democratic ecosystem accessible to all.

In this whitepaper, we introduce TOM3, a decentralized cloud infrastructure that is competitive and capable of distributing applications among various cloud service providers worldwide. The document will present the current market conditions, explain how we are leveraging the latest advancements in serverless container orchestration to address these challenges, cover the fundamentals and importance of the platform's native utility token, TOM3, and ultimately our launch roadmap.

Introduction
TOM3 is a decentralized, transparent, and secure platform for cloud computing that links clients needing computing resources with providers offering computing capacity for charter. TOM3 functions as an overarching cloud platform - delivering a cohesive layer over all platform providers to offer clients one cloud solution, no matter which specific provider they utilize.

Clients choose TOM3 due to its cost efficiency, user-friendliness, and adaptability to switch between cloud providers, along with the performance advantages of worldwide deployments. Providers utilize TOM3 as it enables them to generate profits from both dedicated and temporarily-unused capacity.

Today, cloud infrastructure suppliers are projected to represent 65% of worldwide internet traffic, with the Big 4 responsible for 80% of the total data transmitted. The future of the internet risks becoming consolidated, centralized, and vulnerable to these major providers.

The main reason for embracing cloud services is the potential for flexibility and cost savings; however, the truth is that the offerings from cloud providers are costly, complex, and confine customers within ecosystems that restrict their capacity to innovate, compete, and maintain control over their infrastructure requirements.

The distinction in capital costs for acquiring hardware versus leasing datacenters between utilizing the cloud and self-managing (on-premise) is negligible; nevertheless, cloud providers possess a substantial edge in operating expenses due to their investment in automation requiring minimal human involvement.

While on-premise computing provides superior flexibility, performance, and security, companies are moving away from their datacenter operations and transitioning to the cloud because they struggle to rationalize the operating costs, attributed to insufficient automation and low utilization rates. Inactive, underused servers turn out to be expensive and inefficient. Experts believe that up to 70% of servers currently in use have capacity that is not fully utilized.

Cloud providers enhance margins by creating hyper-scale facilities, meaning they consolidate resources in a limited number of datacenters for cost efficiency, while also cross-selling comprehensive managed backend services like databases, cache stores, API gateways, and more. Being hyper-scale enables them to oversubscribe their clients, thereby increasing margins but also introducing single points of failure. Geographically spread workloads provide significant reliability and user performance; however, cloud providers complicate multi-regional use for clients as it does not align with their interests.

Cloud providers favor clients who host their applications in one datacenter and impose penalties for utilizing cross-regional or multi-zonal setups, typically by charging significant bandwidth fees and differing regional costs. This is the reason AWS has a distinct pricing model for identical resources across different regions.

Although selling instances is profitable, cloud Providers often price instances lower than the premium they demand for managed backend services. These offerings marketed by providers often consist of white-labeled open source initiatives, where the original creators are never rewarded, and the cloud vendors lack motivation to advance the product.

Additionally, an increase in services results in greater reliance of the customer on the cloud provider. The intricacy brought about by greater feature sets, service accessibility, and standardization through non-standard APIs results in customer lock-in by cloud providers, hindering clients from seeking superior alternatives in the market and stifling innovation.

The model implemented by the providers hinders innovation by significantly decreasing the likelihood of a successful open-source project. Cloud providers essentially function as intermediaries that establish the industry's engagement rules without contributing to society at large.

TOM3 Cloud Infrastructure

The primary design goal of TOM3 is to keep entry barriers low for providers while also guaranteeing that clients can have confidence in the resources available on the platform. In order to attain this, the system needs a publicly-verifiable ledger of transactions within the platform. In this regard, TOM3 utilizes blockchain technologies to establish consensus regarding the accuracy of a distributed database.

TOM3 is primarily a platform enabling clients to acquire resources from suppliers. This is made possible by a distributed exchange powered by blockchain, where clients list the resources they need for providers to place bids on. The financial medium of this platform is a digital token known as TOM3, which has its ledger recorded on a blockchain. Here are the basic requirements for a platform like TOM3: Numerous workloads are distributed across

multiple datacenters; connectivity limitations that stop unauthorized access to workloads; autonomously managed, allowing operators to avoid constant oversight of deployments.

To facilitate the deployment of workloads on acquired resources, TOM3 features a peer-to-peer protocol that shares workloads and deployment configurations to and among a client's providers. In TOM3, workloads are characterized as Docker containers. Docker containers provide highly-isolated and customizable deployment environments and are already integrated into numerous cloud-based deployments today.

TOM3 Blockchain

The TOM3 blockchain offers a foundation of trust within a decentralized and trustless setting. Clients naturally place their trust in today's major infrastructure providers mainly due to the brand reputation they have established over time. TOM3 does not, and should not, necessitate that same level of belief, as any provider with the necessary capacity can vie to deliver services on TOM3. Rather, the blockchain gains trust through a clear and open platform. Information on the chain is a permanent and accessible log of every transaction, encompassing the deployment history of each Provider.

No individual organization governs TOM3, and no middleman assists in transactions. As a result, no organization has the motivation to manage or to generate additional income from the platform. Major corporations or cloud service providers can engage in the TOM3 platform as a Provider, supplying compute resources to a small or large corporation or to an individual developer, while all three entities maintain equal status within the ecosystem.

TOM3 Token

TOM3 Token facilitates value exchange and aligns economic motivations with appropriate user actions. TOM3 token serves as the currency for the platform, facilitating payments for rented compute resources on TOM3's decentralized ecosystem. Our token fulfills two main roles within TOM3's ecosystem: TOM3 Token's liquidity will align with the need for computational power. In line with this perspective, we are completely confident in TOM3's ability to secure optimal liquidity for its initial users and final state participants.

TOM3 Staking

The reliability of TOM3 depends on a staking mechanism designed to prevent malicious users from exploiting our system. A staking system creates a substantial financial deterrent for malicious individuals contemplating involvement in our ecosystem. The likelihood of fraudulent actions increases significantly when unfamiliar, new providers enter our ecosystem. Instead of needing a centralized or federated approval system for new accounts, TOM3 permits anyone to participate.

When a new provider decides to provide its resources on TOM3, instead of being approved, it must stake a significant amount on the ecosystem in TOM3 tokens. There is no minimum stake requirement, but involvement in TOM3 governance correlates with a provider's stake, considered as a portion of the total stakes. Moreover, stake contribution plays a role in determining a provider's reputation score, which tenants might consider as a criterion for deployment.

TOM3 Payment

TOM3 tokens help reduce the foreign exchange risk that often arises from international payments. Replacing fiat for these transactions, TOM3 tokens streamline the value exchange in the cloud infrastructure sector. Our matching

engine competitively prices every container compute against the current market rate of TOM3 tokens. When a tenant is paired with a provider, the tenant sends TOM3 tokens to the ecosystem, which are then disbursed to the provider as per the charter agreement.

TOM3 Platform

Infrastructure procurement - the method by which clients acquire infrastructure from suppliers - on TOM3 is executed via a decentralized exchange. The platform comprises a public order book along with a matching algorithm. Clients submit requirement specifications that specify their service requirements, and datacenters submit deployment orders to compete for those requirement specifications. Requirement specifications specify the highest price the client is prepared to pay for a set quantity of computing units (measured by memory, CPU, storage, and bandwidth) for a determined period; deployment orders state the cost at which the provider will supply the resources.

Requirement specifications are available for a client-specified duration, precisely measured in seconds. As long as the requirement specification is active, providers can submit deployment orders to bid on it. An deployment order can be paired with a requirement specification if it meets all the minimum criteria set by the requirement specification. When provided with a requirement specification and a collection of qualifying deployment orders, the deployment order that presents the lowest price will be paired with the requirement specification. If several deployment orders qualify for a match and present the same price, the deployment order that was placed first will be paired with the requirement specification.

Companies and individual consumers will seek and require to safeguard their public representation of compute power usage. To protect against competitor data mining and other attack methods, a layer of homomorphic encryption is implemented. A charter is established when there is a match between a requirement specification and a deployment order. The charter includes mentions of the requirement specification and deployment orders. Charters will act as the essential element in achieving a deployment.

Decentralized Deployment Mechanism

After resources have been obtained, clients need to allocate their tasks to providers so they can operate on the rented resources. We call the present condition of the client's workloads on TOM3 as deployment.

A user outlines their intended requirement specification in a workload. The workload is composed in a declarative file format that includes workload definitions, configurations, and connection rules. Providers utilize workload definitions and configurations to carry out the workloads on the resources they offer, and apply the connection rules to establish a virtual compute infrastructure along with firewall settings. A hash of the workload is referred to as the deployment version and is recorded on the blockchain-based distributed ledger.

Distributed Workload

The workload includes confidential information that should exclusively be shared with those involved in the deployment. This creates an issue for self-managed deployments - TOM3 needs to allocate the workload definition independently, without disclosing its details to irrelevant parties.

To tackle these problems, we developed a peer-to-peer file sharing system where charter participants exchange the workload among themselves as required. The protocol operates off-chain utilizing a Secure Sockets Layer (SSL)

connection; every participant can confirm the workload they obtained by calculating its hash and comparing it with the deployment version held in the blockchain-supported distributed ledger.

Along with offering private, secure, and autonomous distribution of workloads, the peer-to-peer protocol facilitates the swift distribution of extensive workloads to numerous datacenters.

TOM3 Ecosystem

A workload's ecosystem is isolated by default — no connections are allowed. Though this is safe, it is not feasible for practical use. For instance, take a basic web application: client browsers must be able to reach the web-tier workload, and the web-tier must interact with the database workload. Moreover, the web-tier might be located in a different datacenter than the database.

On TOM3, users can choose to permit communications to and among workloads by establishing a connection topology within the workload. Datacenters utilize this topology to set up firewall rules and establish a secure ecosystem among specific workloads when required.

To enable secure communications across datacenters, providers allow workloads to interact via an SSL tunnel. Every workload-to-workload link employs a separate SSL tunnel. Prior to creating these tunnels, providers produce an SSL certificate for every needed tunnel and share these certificates with the required peer providers. The root certificate of each provider is kept on the blockchain-based distributed ledger, allowing peers to confirm the legitimacy of the certificates they obtain.

After certificates are exchanged, providers create an authenticated tunnel and link the ecosystem of the workload to it. Everything of this nature is clear to the workloads - they are able to interconnect via consistent addresses and conventional protocols.

TOM3 Autonomous Scaled Deployment

The ever-changing qualities of cloud infrastructure present both advantages and challenges for operations management. The ability to provision new resources at any time is a blessing; however, the increasing management overhead and complexity of those resources is a curse. The objective is to ease the challenges of cloud infrastructure by utilizing TOM3's autonomous deployment protocol.

TOM3 was designed from scratch to equip engineers with an easy yet robust toolkit for establishing highly automated deployments. The toolkit includes the components that allow non-management applications such as general workloads and virtual ecosystems to be used to build autonomous, self-managed systems.

Self-managed deployments on TOM3 involve a straightforward process of creating workloads that autonomously handle their own deployments. An engineer might utilize a task that modifies DNS records as suppliers enter or exit the deployment, evaluates response times of applications in the web-tier, and adjusts infrastructure (according to the permissions and limitations set by the client) as required depending on various input metrics. The "management tier" can be distributed among all datacenters in a deployment, with global state upheld by a distributed ledger operating over a secure virtual ecosystem.

Numerous web-applications are sensitive to latency - reduced response times from application servers leads to a significantly enhanced end-user experience. Contemporary implementations of these applications utilize content delivery networks (CDNs) to swiftly provide static content like images to end users. CDNs lower latency by distributing content to be nearer to the users who are retrieving it. Deployments on TOM3 can not just replicate this method but surpass it - TOM3 allows clients to position dynamic content nearer to the users.

To establish a self-regulating dynamic delivery ecosystem on TOM3, an engineer would integrate a management layer in their deployment that observes the geographic positions of clients. This management level would deploy and decommission datacenters worldwide, allocating additional resources in areas with high user activity, while reducing resources in regions with low user engagement.

TOM3 is making a sizable impact in machine learning. Machine learning applications utilize numerous nodes to parallelize calculations that deal with extensive datasets. They perform their tasks in "batches" - there is no necessary "steady state" of capacity. A machine learning application on TOM3 can utilize a management layer to actively acquire resources in a single datacenter. At the start of a machine learning task, the management layer can "increase" the number of nodes assigned; once the task is finished, the allocated resources can be returned.

TOM3 is an ongoing research initiative, and updated versions of this paper will appear at tom3.xyz. For any comments or suggestions, please contact tom [at] tom3.xyz.

References

Liang, N.-Y., and G.-B. Huang. 2006. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward ecosystems. IEEE Transactions on Neural ecosystems 17 (6): 1411–1423.

Papazoglou, M.P. 2012. Cloud Blueprints for Integrating and Managing Cloud Federations. In Software Service and Application Engineering, 102–119. Berlin: Springer.

Lynn, T., P. Rosati, and P. Endo. 2018. Towards the Intelligent Internet of Everything: Observations on Multi-disciplinary Challenges in Intelligent Systems. Proceedings of the Research Coloquio Doctorados: Tecnología, Ciencia y Cultura: una visión global.

Masood, A., and A. Hashmi. 2019. AIOps: Predictive Analytics & Machine Learning in Operations. In Cognitive Computing Recipes, 359–382. Berkeley, CA: Apress.

Haller, S., S. Karnouskos, and C. Schroth. 2009. The Internet of Things in an Enterprise Context. In Future Internet Symposium, 14–28. Berlin, Heidelberg: Springer

Prasad, P., and C. Rich. 2018. Market Guide for AIOps Platforms. Gartner.

Iorga, M., L. Feldman, R. Barton, M.J. Martin, N.S. Goren, and C. Mahmoudi. 2018. Fog Computing Conceptual Model. Special Publication No. 500-325. NIST.

Cardoso, J. 2019. The Application of Deep Learning to Intelligent Cloud Operation. Paper presented at Huawei Planet-scale Intelligent Cloud Operations Summit, Dublin, Ireland.

Kleppe, A.G., J. Warmer, J.B. Warmer, and W. Bast. 2003. MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Professional.

Huang, Liang, et al. 2005. On-Line Sequential Extreme Learning Machine. The International Conference on Computational Intelligence.

Dang, Y., Q. Lin, and P. Huang. 2019. AIOps: Real-World Challenges and Research Innovations. Proceedings of the 41st International Conference on Software Engineering: Companion Proceedings, 4–5. IEEE Press.