

# ByteNite: A New Business Model for Grid Computing

Fabio Caironi ✉ 🏠 

ByteNite Inc., San Francisco, CA, USA

Niccolò Andrea Castelli ✉ 🏠

ByteNite Inc., San Francisco, CA, USA

---

## Abstract

Years and years of technological advancement have paved the way to cloud computing towards Industry 4.0, making it possible for a wide range of cloud solutions to become a reality, bringing innovation and efficiency to business processes and changing our lifestyles. With the benefit of hindsight in a fully digitalized era, have we ever wondered where does cloud computing come from? Furthermore, as the on-premise commercial model shifted to cloud computing with the advent of the internet, what will the increase in worldwide connectivity and the rise of 5G turn the cloud model into? This article describes in a model for a new commercial grid computing implementation, called “ByteNite”. We open the paper with the state of the art of the distributed computing models, including an overview of cloud and grid computing, their commonalities and history, and how they are topical in today’s world. We build the foundations of our work through a key insight that triggers powerful implications in connection with the current technologies. We address the new proposed model through a description of the system, its overall functioning, the underlying business concepts and the innovative value proposition. We finally then dive into its architecture and workflow design, delineating its structure and key features, and the chronological phases of its operation.

**2012 ACM Subject Classification** Computer systems organization → Grid computing; Computer systems organization → Cloud computing; Computing methodologies → Distributed computing methodologies; Software and its engineering → Software architectures; Computer systems organization → Dependable and fault-tolerant systems and networks

**Keywords and phrases** Grid Computing, Cloud Computing, Distributed Applications, High-Throughput Computing, dApps, Utility Computing

**Digital Object Identifier** 10.4230/OASICS.PARMA-DITAM.2023.1

**Category** Invited Paper

**Related Version** *Full v2.0*: <https://bytenite.com/bytenite-white-paper-full-version>

## 1 Introduction

In the IoT and Big Data era, cloud computing and distributed file systems are fundamentals for data management and processing. Big tech firms and their server farms are the most valuable resource we can rely on today for outsourced computations; edge computing has become indispensable in many applications as the volume of data produced daily by businesses is increasingly significant.

Cloud computing is more than renting someone else’s machines: it encompasses workload management, service orchestration, distributed storage, and much more. However, it all boils down to the target machine’s computing power provided by its processor when it comes to throughput and performance. After all, as B. Sosinsky [25] goes, “*cloud computing is revolutionary, even if the technology it is built on is evolutionary.*”

The invention described in this article mostly conforms to the techniques dictated by the model known as “grid computing”. However, several other topics and frameworks can be deemed relevant to this invention, including utility or on-demand computing,



© Fabio Caironi and Niccolò Andrea Castelli;  
licensed under Creative Commons License CC-BY 4.0

14th Workshop on Parallel Programming and Run-Time Management Techniques for Many-Core Architectures and 12th Workshop on Design Tools and Architectures for Multicore Embedded Computing Platforms (PARMA-DITAM 2023).

Editors: João Bispo, Henri-Pierre Charles, Stefano Cherubin, and Giuseppe Massari; Article No. 1; pp. 1:1–1:12



OpenAccess Series in Informatics

OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

high-throughput computing, distributed computing, and, most of all, cloud computing. Grid and cloud computing share several key traits, such as their reliance on distributed resources. Still, they differ slightly in many domains, including business model, architecture, resource management, and application model. Today, grid computing has evolved to become the basis of the more advanced cloud, offering more robust performance in a secure virtual environment. Yet, we claim that there is much value left behind in this transition, and no project or initiative has been able to seize it and implement it at scale so far.

## **1.1 Grid vs. Cloud Computing**

According to [17, 23], a grid can be defined as a large-scale geographically distributed hardware and software infrastructure composed of heterogeneous networked resources owned and shared by multiple administrative organizations, with the goal to create the illusion of a simple yet large and powerful virtual computer supporting a wide range of applications. Grids were developed in the mid-1990s to provide a solution for large-scale computational tasks that required significant processing power, only affordable by supercomputers back then. The emerging concept of virtualization turned out to be a big win in the utility computing model: it allowed applications to be abstracted from the underlying fabric (compute power, storage, network, etc.) and deployed on-demand to more exacting customers requiring stringent SLAs. That's how the grid computing model quietly shifted into what we call today *cloud computing*. The rapid adoption of the cloud from the mid-2000s was fostered by the decrease in hardware cost and increase in computing power and storage capacity, as well as the exponentially growing size of data and processing power used by modern internet applications and services. On the architecture level, grids and clouds share a fabric layer consisting of the raw hardware resources and the protocols to access them. While clouds provide a unified resource layer to virtualize such resources and expose them to end-user applications, grids feature a more complex set of standard protocols, middleware and toolkits to connect and manage the resources. Ensuring interoperability and security are fundamental both for grid and cloud infrastructures. While in grids interoperability comes built-in, as they are based on the assumption that resources are heterogeneous and dynamic, clouds have developed stronger security policies complying with regulatory standards. The combination of such properties in cloud-powered grid computing systems might prove a critical vision for the future of the cloud in the 2020s.

## **1.2 Grid computing today**

Nowadays, most grid computing initiatives around the world have given their way to more modern and service-oriented cloud computing applications. Plenty of grid middleware implementations and grid infrastructures built in the 2000s have either ceased operating, turned into cloud projects, or been acquired by cloud computing companies.

United Devices Inc., a commercial volunteer computing company offering high-performance computing services, was sold to a software company that developed cloud management products called Univa in 2007, which was in turn acquired by cloud software company Altair Technologies. DataSynapse was sold to TIBCO Software Inc. in 2009, a business intelligence software company, and their grid computing middleware was turned into a BI product powered by parallel computing. A different fate awaited companies like Entropia, Inc. and Popular Power, developers of distributed computing software for CPU scavenging, which were driven out of business. And so on: the list of companies born in the new millennium trying

to ride the wave of grid computing is long [18]. It is no mystery why they all succumbed in a matter of few years: while they were able to develop large-scale computing infrastructure by accessing the spare processing capacity of thousand of volunteered CPUs, these companies didn't offer a form of reward to their contributors. Consequently, the resource owners had no incentive for their continued contribution, and the economic model proved not scalable nor maintainable [19]. Given those years' computing and network capabilities, the only companies that managed to survive were those noticed and acquired by larger corporations, which could afford substantial infrastructure investments to keep up with the incoming cloud wave.

In the volunteer computing world, grids have made a name with some scientific projects that gained much attention in the academic community throughout the 2000s. Either infrastructure-based as TeraGrid [20], middleware-based like the Globus Toolkit [22, 14], or application-based like SETI@Home [7], all these kinds of projects were aimed at empowering scientific research in disparate fields (Physics, Medicine, Astronomy, Mathematics, Biology), making it possible to solve computationally intensive problems that would have been difficult or infeasible to tackle using standard computers. Some historical volunteer computing projects made their way through the 21st century and are still working in 2022. Their participation was primarily motivated by non-monetary prizes, fun, fame, or collaborative advantage.

The most representative one is BOINC [1, 15], a platform for distributed high-throughput computing where worker nodes are desktop and laptop computers, tablets, and smartphones volunteered by their owners. A fair number of applications or "projects" are linked to BOINC and use or have used its distributed computing infrastructure to solve large-scale scientific problems that could once be tackled only by supercomputers. SETI@Home was the first and foremost and gave BOINC the popularity it later had. It was devoted to the Search for Extra-Terrestrial Intelligence through distributed digital signal processing of radio telescope data. A week after its launch, SETI@Home scored 200,000 participants; after four or five months, it broke through a million, and later reached past two million users. In 2020 the project officially ceased operations. Other remarkable BOINC-powered projects include: Einstein@Home [4] for the search of weak astrophysical signals from spinning neutron stars; World Community Grid [10] for scientific research on topics related to health, poverty, and sustainability; Climateprediction.net [2] for climate models simulations. Distributed.net [3] was another volunteer computing project attempting to solve large-scale problems, governed by a non-profit US corporation. As of 2019, distributed.net's throughput was estimated at roughly 1.25 petaFLOPs. Lately, distributed.net has joined forces with BOINC with the aim of finding mathematical solutions to cryptographic algorithms. Another operating volunteer computing project is HTCCondor [5, 26], an open-source distributed computing software enabling the increase of computing throughput, developed at the University of Wisconsin-Madison. HTCCondor provides a job queueing mechanism, a scheduling policy, a priority scheme, and a resource monitoring and management tool, and can integrate dedicated resources (rack-mounted clusters) and non-dedicated desktop machines into one computing environment. Finally, a distributed computing project that has lately gained a broad consensus due to new discoveries regarding SARS-CoV-2 is Folding@Home [16]. The main aim of this project is to understand protein dynamics by means of statistically distributed simulations. In 2020 the computing speed of Folding@Home peaked at 2.43 exaFLOPS, which is a computing power in the order of one billion billion floating point operations per second, enough to mine a Bitcoin in ten seconds.

Although these projects are of great help for research, they won't be able to unlock the full potential of a worldwide grid. Their genesis and purpose keep them away from reaching a wider audience and becoming marketable products. The replicability of any of these models on the market is not only prevented by the lack of a well thought-out payment framework, but especially by the lack of a performance-oriented resource management system built with modern and widely adopted standards and protocols.

Starting in 2010, a new distributed technology started bringing collaborative computing back into the spotlight. A new global paradigm was established and many companies followed by building products on top of it or creating their own private sub-networks to capitalize on what proved to be more than a brand-new concept. I am referring to the blockchain and all the blockchain-powered dApps (decentralized applications) that have been implemented thanks to the wild proliferation of this technology. A dApp is an open-source software application that runs on a peer-to-peer blockchain network. dApps are built for disparate use cases across various industries, including finance and payments, gaming, supply chain, user-generated content networks, and distributed computing. The latter use case is relevant to our framework, as it involves dApps that exploit member devices' processing power and network to improve and democratize access to CPU- or GPU-intensive digital services. Some most notable implementations of decentralized computing involve video streaming (Livepeer [24, 6], Theta Network [9]), mobile blockchain mining (Sweatcoin [8], MinePi [12]), and general-purpose computing (Golem [11], Cudos [13], iExec [21]). These applications usually use Ethereum or owned coins for collecting and distributing payments, and they handle crypto transactions and task validation with smart contracts. Ethereum also provides these dApps solutions for guaranteeing distributed consensus and identity management.

A question that might arise is how Ethereum and, generally, blockchain technology actually empower distributed computing on the processing side. The answer is possibly that it doesn't. Uriarte, R.B. and DeNicola, R. (2018) [27], have analyzed the architectures of three blockchain-based decentralized cloud solutions. Their finding is that in all three projects, smart contracts, payments, and reputation are managed in a "transaction network" built on the blockchain, while the actual computing services are executed in a "side-chain network" charged with processing, negotiation, and verification of computing tasks. As the paper highlights, the results obtained from a collaborative, distributed computing network might be chaotic and heterogenous; hence, the side-chain network reveals a non-deterministic behavior that must be mediated in order to reach a consensus in the transaction network, and a specific component is needed to interface between the two networks. This adds complexity to the already high computational cost of running and maintaining a blockchain.

There are other elements holding back Ethereum and other blockchain technologies from implementing a large-scale, efficient grid like the one discussed in this White Paper. Two of them are the high transaction costs and the capped transaction throughput (Ethereum can process less than 30 transactions per second), both posing serious threats to performance and scalability. Another shortcoming is the almost absent definition of Quality of Service in most dApps' smart contracts, or even in their general terms and conditions. Besides signaling an inability to control and measure the average processing performance, the absence of QoS makes big customers, that are seldom unconcerned about quality guarantees, shy away from blockchain-powered computing solutions.

Finally, it is worth mentioning that, despite being the core philosophy of such dApps, the restriction to support only crypto wallets and cryptocurrency transactions cuts off the vast majority of both resource providers and cloud computing customers, who normally do business with fiat currencies and are still – and possibly forever – crypto-averse.

### 1.3 Fact

In 2023, an immense underlying computational power is widespread throughout the globe and sits idle for most of the time. Altogether, it overcomes the joint processor capacity of the biggest cloud providers by tens of times. More than 12 billion computers, smartphones, tablets, and other commercial electronic devices are hiding an immense potential, especially now that they're shipped with ever more performing hardware, and they're usually unexploited during the inactivity of their human owners, like during the night. Not only are electronic consumer devices underused: many businesses owning disparate types of hardware, from video production facilities to private data centers and office desktop computers, don't know how to use it when it's not at work.

Past and existing grid computing projects have shown us the potential of building a distributed computing farm by tapping into a category of machines not originally sold to fulfill utility computing purposes – the mass consumer technology. However, such vast unused computational power couldn't be easily gathered and connected until a few years ago because of major technological limitations, including the average network speed, network coverage, and the hardware capacity of common devices on the market. Plus, all the attempts to build a global grid have been held back by exclusively technology-gear strategies and major market misunderstandings, largely attributable to shortsighted or too-technical visions, that entailed failing executions or limited outcomes.

Today, the easy and fast access of any device to the internet and the virtualization provided by the cloud make it possible to collect and utilize the vast worldwide computing potential in a distributed computing system, reviving the already-known paradigm of grid computing and enhancing it with the reliability, scalability, and automation provided by the cloud. At the same time, the lessons learned from the past make us steer clear of development strategies that have the grid technology as the only guiding star: for such a massive commercial project to be successful, any development choice, from architecture to applications, must be driven by evident market demands and clear economic visions, that spur the adoption of grid computing as key to solving market-inherent cost-benefit problems.

## 2 A new model: ByteNite

ByteNite is a commercial, centralized, service-oriented grid computing system based on subscriber devices' processing capacity, realizing a high-throughput computing environment for utility computing purposes. Rather than an online marketplace, where buyers and sellers are directly put into contact, ByteNite creates two different and separate hubs that are accessible by the purchasers of computing services ("users" or "customers") and by the suppliers of computing power ("workers" or "suppliers"), respectively, brokering the management of computational resources to keep the two segments well coordinated and functioning.

The three components that build up ByteNite's grid computing system are the following:

### ■ Core System

The core middleware, or backend layer, responsible for managing, scheduling, retrieving, transforming, transitioning, sending, organizing, and validating the users' computational jobs. It stores and makes accessible at any moment all the users' and workers' data, including job history, activity, wallet balances, and device info. It also generates quotes, collects users' payments, and distributes rewards to workers.

### ■ ByteNite Computing Platform

A user-level middleware available as a software-as-a-service platform, accessible through a web UI or an API, exposing both ready-made and custom-made computing services ("applications") to customers. On the platform, users can configure, submit, and pay

## 1:6 ByteNite: A New Commercial Model of Grid Computing

for computing jobs, as well as upload and download their data (inputs and outputs), and watch their job history, jobs states, and summary usage. They can automate the execution of their jobs via recurring tasks and automation pipelines.

### ■ **ByteNite Worker App**

A piece of software that runs on workers' devices and enables them to receive, queue up, process, send back, and clear up computing tasks, according to programs shipped with each task and run inside the App. The Worker App also makes available and visible the summary of completed tasks and their credits; hence, it allows workers to redeem their credits by converting them into several forms of reward, including cash.

In other words, ByteNite provides software to connect the users to the system, schedule the workload, and connect the computational grid to the system. The workers supply the fabric layer consisting of distributed computing resources, and the users provide all the inputs that feed the applications, including data.

ByteNite stands in the market as a provider of high-throughput computing services. It targets small- and medium-sized companies seeking faster performance at more affordable prices than the cloud, and enterprises that operate daily with big volumes of data and need to speed up their workflows. In both cases, ByteNite helps fulfilling performance goals for specific applications that generate loosely coupled or independent tasks. ByteNite will develop three target applications that represent its core mission and an extraordinary market opportunity: Video Encoding, Graphics Rendering, and Computer Vision. In addition to being three of the most intensive commercial computing activities, these applications are well-suited for distributed computing as each of them generates workloads that can be divided into multiple, independent smaller tasks. On the other side, ByteNite's customers will be provided with the tools to develop their own distributed applications to run on the grid resources using ByteNite Computing Platform. It is possible to find a variety of use cases for such tailor-made solutions in the media & entertainment industry, as well as in the financial and healthcare sectors.

On the other side, ByteNite offers a solution to make passive income out of ordinary devices, like personal and office computers, smartphones, tablets, small servers, and eventually even a wider range of IoT devices like video game consoles, TVs, home appliances, and industrial electrical machinery. Whilst in 2022 we have online marketplaces to effortlessly sell or rent out almost everything, from material belongings to volatile goods like electricity, it is not yet possible to rent out our devices' exceeding computing capacity in the matter of a few minutes. ByteNite brings together the technology to enable such a monetization possibility with a smooth onboarding of the workers, by streamlining the workflow and condensing all the interactions into a single piece of software, ByteNite Worker App.

### **Innovation**

ByteNite is the first distributed computing solution to combine the following accomplishments:

- Uses heterogeneous, cross-platform, both mobile and desktop devices located anywhere as worker nodes; Creates a computing-capacity sharing economy based on the trade of distributed processing tasks with real money;
- Is open to everyone;
- Constantly monitors performance and automatically turns it into business requirements and price adjustments;
- Manages non-deterministic behaviors with a centralized scheduling system based on both a-priori and a-posteriori fault-tolerant techniques.

ByteNite has the mission of becoming the first worldwide grid powering a general-purpose high-throughput computing system, where everybody can build and run their distributed applications or use ready-made flagship computing products. ByteNite's values are enclosed in following attributes:

- *Availability*

The extension of ByteNite's grid, together with its devices' diversification, geographical distribution, and heterogeneous connectivity, allows and guarantees flexible provisioning of computing resources at any time.

- *Agility*

The commodification and customization of computing services, plus the existence of an optimal delivering pipeline, make the entire process from data ingestion to output upload extraordinarily agile.

- *Speed*

The more nodes are in the grid, the less time is needed to process partitioned jobs. This fact makes ByteNite competitive and preferable to the classic cloud and on-premise computing for various use cases.

- *Sustainability*

Deploying distributed computations on existing and commonly active devices is an environmentally-friendly alternative to using server farms, provisioning new hardware, and building new infrastructure. ByteNite's distributed computing model guarantees an inherent heat dispersion from devices' processors that are connected from different locations, removing the need for artificial cooling of rack-mounted servers. In addition, old or unused devices can be turned into ByteNite's workers instead of winding up in the trash, contributing to lowering the pollution caused by electronic waste.

- *Security*

Data is at the core of ByteNite's business, and so is cybersecurity. All data coming to and from ByteNite's system is encrypted and handled in isolated runtime environments, and workers are constantly monitored and readily excluded if deemed potentially malicious. In addition, ByteNite's reliance on a robust and certified cloud grants it ready and updated cybersecurity policies and implementations that are nowadays standards for all cloud-based software companies.

## 3 ByteNite's Core System

In this section, we shall give a brief overview of how ByteNite works from a backend perspective: how its Core System is structured, what the components responsible for running the services are, and what stages the general workflow is composed of.

### 3.1 Architecture

ByteNite's Core System has a micro-services architecture. Each service represents an independent and scalable backend component running in the cloud and interfacing with the Worker App, the Computing Platform, and the other components through dedicated APIs. The architecture diagram is depicted in Figure 1.

The following internal services run the business logic and are not exposed publicly:

- The *Partitioner* verifies the integrity of data uploaded by the users through the Computing Platform, and splits it into smaller chunks suitable for worker devices. A task record is created for every chunk, and the record ID is queued on a job-specific Redis queue.

## 1:8 ByteNite: A New Commercial Model of Grid Computing

- The *Feeder* manages and supervises the whole task scheduling system. It takes tasks from job-specific queues and puts them in a global task queue ready to be consumed by the Tasks API. Tasks are sorted according to a scheduling algorithm that considers the availability of computing resources in the grid, the job's requirements, and the user's preferences.
- The *Validator* verifies the integrity and correctness of results sent by the worker apps. Different jobs could use different validators.
- The *Assembler* collects completed and validated tasks from the Validator and assembles them into larger chunks until it has rebuilt the full processed data file, which is uploaded to a cloud storage bucket accessible from the Computing Platform.
- The *Reward System* is responsible for clearing ByteChip transactions between ByteNite and the workers and ensuring that all balances are constantly updated.

The customer APIs handle communications with the Computing Platform:

- The *Jobs APIs* allow the Computing Platform to create and configure new jobs, send input data, send and receive state updates, and fetch download links.
- The *Billing API* allows the Computing Platform to access billing and payment information.

Similarly, the worker APIs connect the Core System with the Worker Apps:

- The *Tasks APIs* allow the Worker App to fetch new tasks, download the data and programs, and send back results or abort the task.
- The *Wallet API* allows the Worker App to get the ByteChip balance and history and to request and record ByteChip expenditures in services or payouts.
- The *Devices API* connects to Firebase to fetch information about task and device states, user authentication, and device preferences. This is the only server-side component that connects to Firebase.

Finally, ByteNite's data is sorted and stored in the following components:

- The *Cloud SQL Database* is a SQL database that supports atomic transactions. It stores all data with persistence and consistency priorities over access performance.
- The *Firebase Database* stores all device-related information like hardware specifications and device state and handles authentication. This is the only database that directly interfaces with the devices.
- The *Redis Databases* are fast databases for internal usage that handle short-run storage for frequent reads, writes, and inter-service messages.
- The *Cloud buckets* are web-based folders with access restrictions that store files downloaded or uploaded by the users.

### 3.2 Workflows

ByteNite fulfills its twofold mandate of collecting users' jobs and distributing them to the grid through several recurring workflows. Each workflow is a set of rules and actions happening either in the Core System, on the Computing Platform, on the Worker App, or among them, that is well-coordinated with the other processes and designed to make the whole execution fault-tolerant and agile. From a 360-degree perspective, the processing of a job can be summarized as follows.

When a new job is submitted on the Computing Platform, ByteNite sets up a pipeline between the user and the grid. First of all, the Feeder builds the framework of the scheduling logic for that specific job, and the Reward System estimates its cost. Hence, the job starts and the Job Upload API streams the input data to the Partitioner, creating chunks on the

fly and passing each of them on to the Feeder. The Feeder wraps them with an executable, forming tasks that are scheduled and sent to the grid. The distribution logic established by the scheduling algorithm run by the Feeder guarantees the abstraction of the scheduling from the actual delivery so that the process is completely automated and reliable. In particular, the algorithm of the Feeder enforces a concept of “first come, first served”, thanks to which no data chunk needs to wait for a specific device to show up, but every chunk is appended to global queues from which the next available device can download it. Every device competes in the grid to process as many tasks as it’s eligible for, and its only assignment is to tune in with ByteNite’s server waiting for new tasks in the global queues, to process them and upload back the results. The grid responds asynchronously, sending back processed tasks from multiple devices. When node failures or delays are encountered, several measures are adopted to guarantee a hassle-free continuation of the processing. In any case, the workflow continues up to the moment when all tasks have been successfully processed, retrieved, and validated. Finally, the Assembler quickly rebuilds the integral output using indexes contained in tasks’ metadata and uploads it to a Cloud bucket immediately available to the user.

All data that goes through the Core System is temporarily stored and released as soon as a job is completed, except for the final output that could be stored in a Cloud bucket for 24h. This, together with the fact that neither the Partitioner nor any other services are tasked with the heavy lifting of data processing, make the execution of ByteNite very light, removing the need to maintain a high-capacity infrastructure. At the same time, ByteNite can control the inflow and outflow efficiently and take care of the integrity and security of data processing.

Figure 2 gives a representation of the general workflow described above.

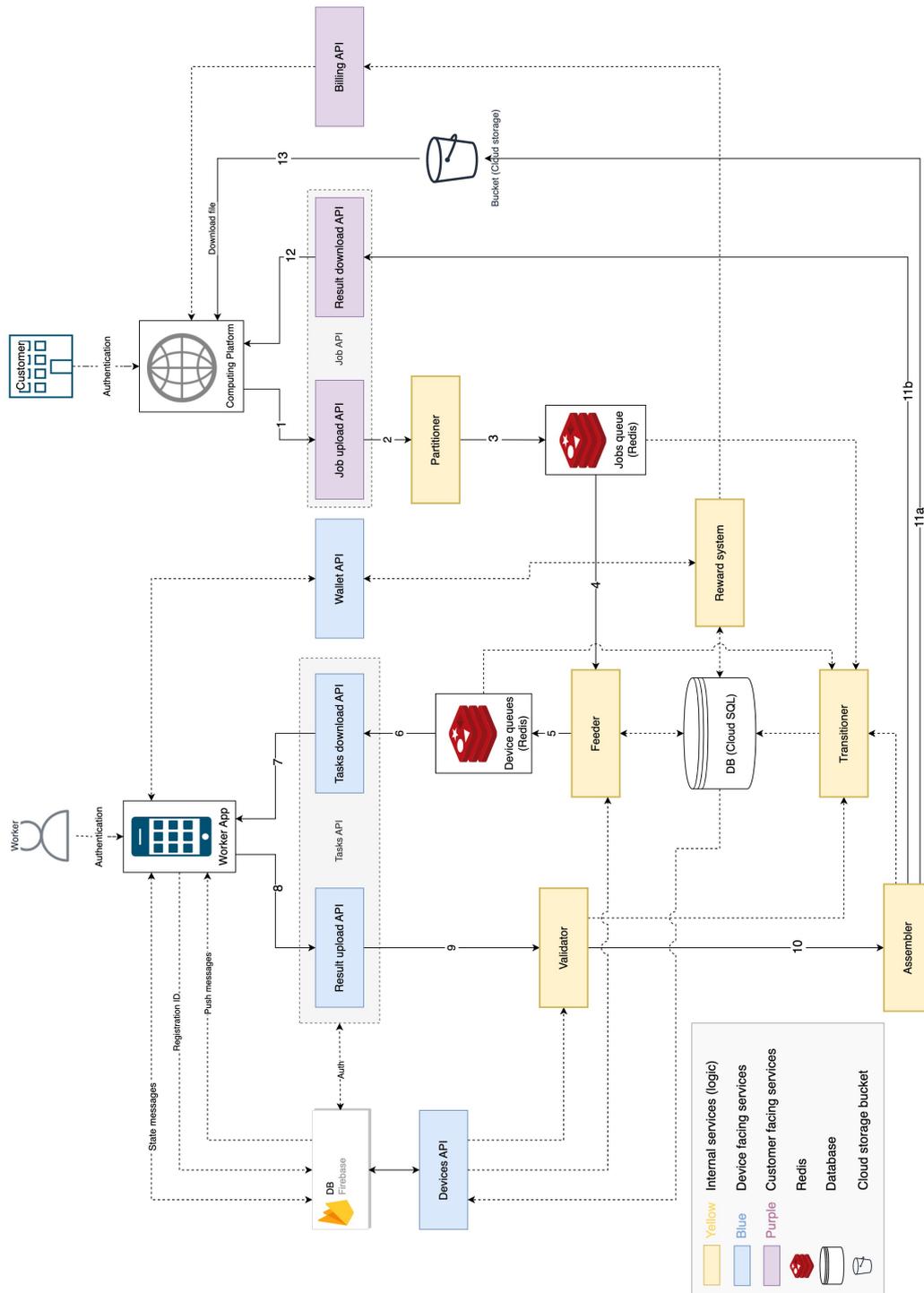
---

## References

- 1 Boinc. URL: <https://boinc.berkeley.edu/>.
- 2 climateprediction.net. URL: <https://www.climateprediction.net/>.
- 3 distributed.net. URL: <https://www.distributed.net/>.
- 4 Einstein@home. URL: <https://einsteinathome.org/>.
- 5 Htcondor. URL: <https://htcondor.org/>.
- 6 Livepeer. URL: <https://livepeer.org/>.
- 7 Seti@home. URL: <https://setiathome.berkeley.edu/>.
- 8 Sweatcoin. URL: <https://sweatco.in/>.
- 9 Theta network. URL: <https://www.thetatoken.org/>.
- 10 World community grid (wgc). URL: <https://www.worldcommunitygrid.org/>.
- 11 The golem project. Golem Factory GmbH, 2016. URL: <https://whitepaper.io/document/21/golem-whitepaper>.
- 12 Pi white paper. SocialChain, Inc., 2019. URL: <https://minepi.com/white-paper>.
- 13 Cudos white paper. Cudos Limited, 2021. URL: <https://www.cudos.org/wp-content/uploads/2021/11/cudos-white-paper.pdf>.
- 14 The Globus Alliance. The globus toolkit. URL: <http://toolkit.globus.org/>.
- 15 D. P. Anderson. Boinc: a system for public-resource computing and storage. In *Fifth IEEE/ACM International Workshop on Grid Computing*, pages 4–10, 2004. doi:10.1109/GRID.2004.14.
- 16 A. L. Beberg, D. L. Ensign, G. Jayachandran, S. Khaliq, and V. S. Pande. Folding@home: Lessons from eight years of volunteer distributed computing. In *2009 IEEE International Symposium on Parallel & Distributed Processing*, pages 1–8, 2009. doi:10.1109/IPDPS.2009.5160922.

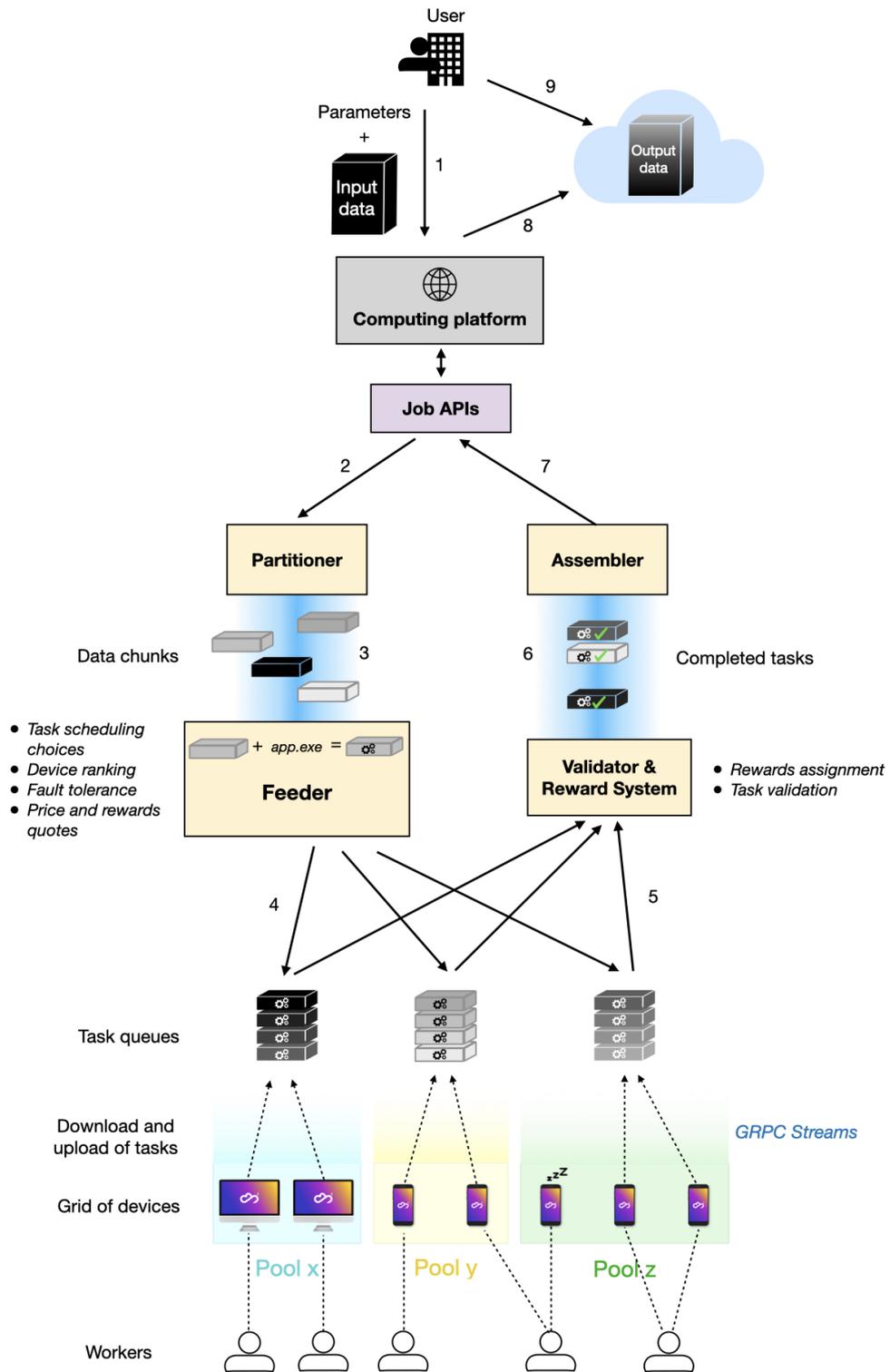
- 17 Miguel Bote-Lorenzo, Yannis Dimitriadis, and Eduardo Gómez-Sánchez. *Grid Characteristics and Uses: A Grid Definition*, pages 291–298. Springer, 2004. doi:10.1007/978-3-540-24689-3\_36.
- 18 Rajkumar Buyya. Grid computing info centre (grid infoware), 2000-2008. URL: <http://www.gridcomputing.com/>.
- 19 Rajkumar Buyya and Kris Bubendorfer. *Market-oriented grid and utility computing*. Wiley series on parallel and distributed computing. John Wiley & Sons, Hoboken, N.J., 2010. doi:10.1002/9780470455432.
- 20 Dane Skow Charlie Catlett, Pete Beckman and Ian Foster. Creating and operating national-scale cyberinfrastructure services. *CTWatch Quarterly*, 2, May 2006. URL: <https://icl.utk.edu/ctwatch/quarterly/print.php%3Fp=35.html>.
- 21 G. Fedak, W. Bendella, and E. Alves. iExec – blockchain-based decentralized cloud computing (whitepaper). Report, iExec, 2018. URL: [https://iex.ec/wp-content/uploads/2022/09/iexec\\_whitepaper.pdf](https://iex.ec/wp-content/uploads/2022/09/iexec_whitepaper.pdf).
- 22 Ian Foster and Carl Kesselman. The globus project: a status report. *Future Generation Computer Systems*, 15(5):607–621, 1999. doi:10.1016/S0167-739X(99)00013-8.
- 23 Bart Jacob and International Business Machines Corporation. International Technical Support Organization. *Introduction to grid computing*. IBM redbooks. IBM, International Technical Support Organization, United States?, 1st edition, 2005. URL: <https://1ccn.loc.gov/2006279225>.
- 24 Doug Petkanics and Eric Tang. Livepeer whitepaper. Report, Technical report, Livepeer, 2018.
- 25 Barrie A. Sosinsky. *Cloud Computing Bible*. Bible v.757. Wiley, Indianapolis, Ind, 1st edition edition, 2011. doi:10.1002/9781118255674.
- 26 Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed computing in practice: the condor experience: Research articles. *Concurrency – Practice and Experience*, 17:323–356, 2005. doi:10.1002/cpe.938.
- 27 Rafael Brundo Uriarte and Rocco DeNicola. Blockchain-based decentralized cloud/fog solutions: Challenges, opportunities, and standards. *IEEE communications standards magazine*, 2(3):22–28, 2018. doi:10.1109/MCOMSTD.2018.1800020.

**A** Figures



**Figure 1** ByteNite's Core System architecture diagram.

1:12 ByteNite: A New Commercial Model of Grid Computing



■ **Figure 2** An illustration of ByteNite's general workflow.