BIG DATA BLOCK

THE DEMOCRATIZATION OF BIG DATA

An accessible, affordable, and secure Big Data
solution serving a $130 billion market (and growing)

bigdatablock.com

“Software is eating the world...”

Marc Andreessen[1] 2011

The velocity of how true this is has become mind numbing. The amount of data being generated is exponential and all of it needs processing.

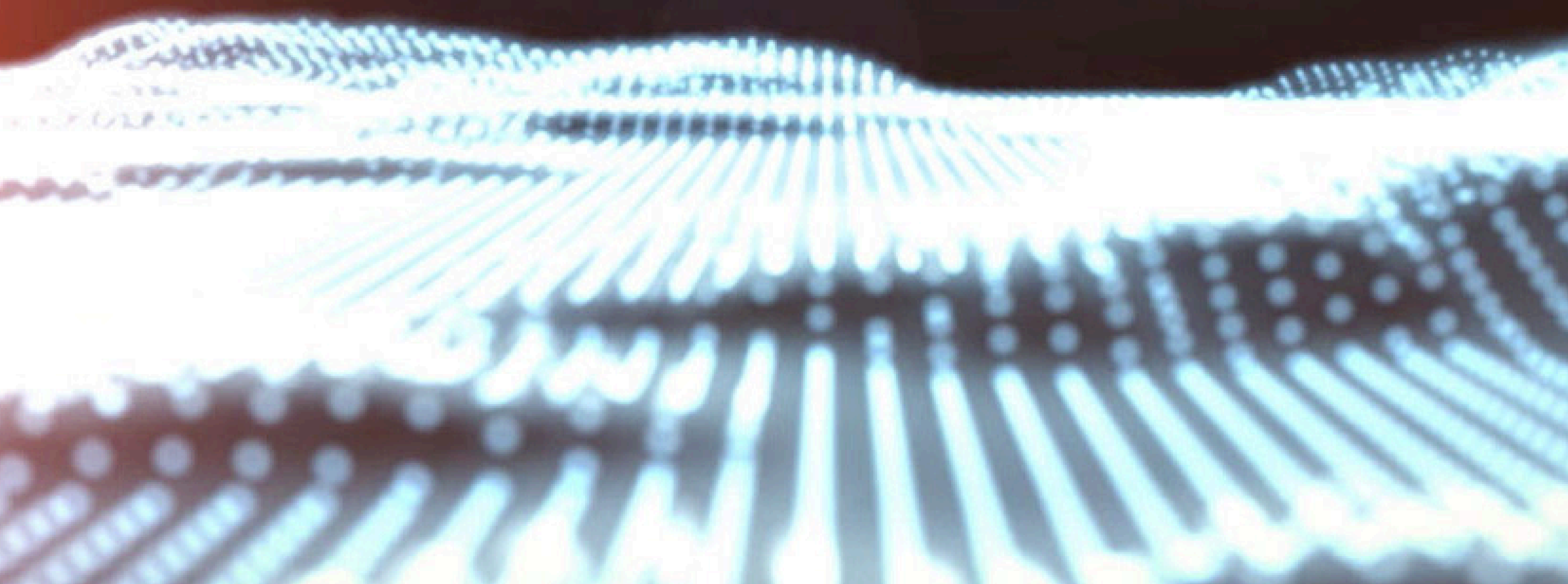


TABLE OF CONTENTS

Introduction

Big Data Block (BDB™) combines Big Data solutions and blockchain technology utilizing Ethereum's smart contract and token capabilities. BDB™ democratizes Big Data for everyone. Combining the best of Big Data and the best of blockchain we remove the deep technical skills and costs needed to leverage a Big Data environment and becoming the first BDaaS (Big Data as a Service) on a blockchain.

PROBLEMS WITH BIG DATA

There are a multitude of issues currently with Big Data including but not limited to cost–it's prohibitively expensive, accessibility–the vast majority of organizations in the world lack the capacity to implement any type of Big Data solution, and security–even the largest multi-hundred million dollar Big Data environments are susceptible to being hacked. Every day, we create 2.5 quintillion bytes of data. To put that into perspective, 90 percent of the data in the world today has been created in the last two years alone – and with new devices, sensors and technologies emerging, the data growth rate will exponentially accelerate.

BDB™ SOLUTION

We make Big Data computational analysis accessible, inexpensive, and more secure. Accessible–via a simple and streamlined GUI. Inexpensive–via our simple pricing tiers with 1000% annual savings over traditional Big Data. Secure–via BDB™ encryption. By using blockchain technology we spread the computing burden across a multitude of computing devices in the BDB™ global blockchain ecosystem. BDB™ is built and optimized to focus solely on Big Data analysis and removes all the associated headaches. In addition, in order to foster the sharing of knowledge, we've created the BDB™ Knowledge Exchange (KE), a portal that allows people with technical and/or data science skills to offer their services to anyone that needs this help. The marketplace matches our customers with technical support experts.

Market Opportunity

VISION

The vision for the BDB™ ecosystem is it to allow anyone to upload a dataset and the associated metadata to a centralized layer we manage. This job is then distributed to decentralized nodes on the blockchain for processing. The results returned are based on whatever analysis is being requested. The only limitation to what BDB™ is capable of is one's imagination.

PROBLEM

Big Data can come from anywhere. Today we capture data from millions of places including customers, sensors, websites, partners, social media, satellites, and more.

Data is being generated in places as close to our body as watches and phones, to other planets light years away. Before BDB™ an organization would have to grow capacity by expanding their data center to handle Big Data workloads. For smaller organizations it was near impossible to begin a Big Data initiative even at the smallest scale.

Regardless of size, from the largest corporation, to the smallest government to a single entrepreneur with a game changing idea, the argument to experiment and embrace Big Data and make it part of a one-off report or part of the greater culture is often lost due to cost, the inability to quickly evidence ROI (monetary, social, etc.), to bring others onboard as champions, advocates and beneficiaries, and the complexity of even asking the right questions to begin with. BDB™ demolishes all these concerns and allows experimentation to spark the fire supported by a community of data scientists (KE) and Big Data enthusiasts. All based on a tokenomics model that paves a purely utility-driven path forward towards better sustainability and a culture of maximum shared impact.

BDB™ solves these five key problems:

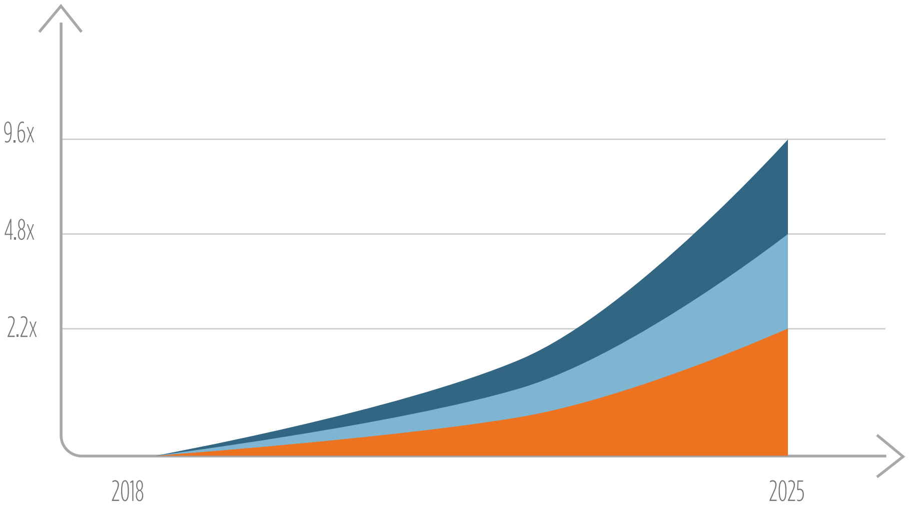
Growth Rate Of Data

From 2020 to 2025, the volume of traditional data will grow by 2.3x; the volume of data that can be analyzed will grow by 4.8x; and the actionable data will grow by 9.6x.

● Traditional

● IoT Relevant

● IoT Actionable



SOURCE: IDC / Getting Ready To Embrace It's Impact On The Digital Economy [1]

1. Skills Shortage

Every day there are new tools, frameworks, and protocols, that results in a massive skill gap. Organizations are struggling to answer how to make long-term technology investments, leverage existing skills, and obtain new ones.

2. Cost

Big Data installations require large clusters of servers with long setup times and extensive minute-to-minute monitoring. The exponential volume, variety, and velocity of data from existing datasets and those that have yet to be discovered can result in out-of-control costs.

3. Unpredictability Of Data

Big Data comes from a wide variety of sources, from legacy applications and transactional systems, to machine-generated data, sensors, mobile devices, satellites, web logs, and social media. A single event can cause major changes effecting every point of the infrastructure.

4. Security/Privacy

As organizations collect, store, and analyze increasing amounts of data from new and existing sources, security and privacy are a great concern. Left to determine compliance, governance, security and privacy protection is challenging, especially without compromising agility and performance. Inside and outside attacks are common in the monolithic data center or even in a multi-tenant cloud-hosted environment.

5. Provenance

Organizations have little assurance that the data has not been manipulated. How can I be sure the data I am using came from a trusted source. How many people have modified this dataset before I started using it?

The Effect On Business

Big Data entails much more than collecting large volumes of structured, semi structured, and unstructured data— that's just the beginning. In order to derive valuable insights from Big Data it must also be securely stored, cleansed, aggregated, sorted, joined, analyzed, etc. BDB™ provides a broad and deep set of easy-to-use tools at a fraction of the cost, providing capabilities that cover all possible Big Data analytics, including Big Data stores, data warehousing, distributed analytics (supporting Hadoop, Spark, HBase, Hive, Pig, and Yarn), machine learning, and artificial intelligence. With BDB™, there's no hardware to procure and no infrastructure to maintain and scale. As the first BDaaS (Big Data as a Service) on a blockchain organizations have on-demand access to compute, storage, and networking capacity. To provisioning, availability, durability, recovery, and backup services. Organizations can ingest data at any velocity, from a variety of sources (internal datasets and the millions of datasets freely available as part of the BDB™ community and our KE), and process and analyze data with tools provided by BDB™

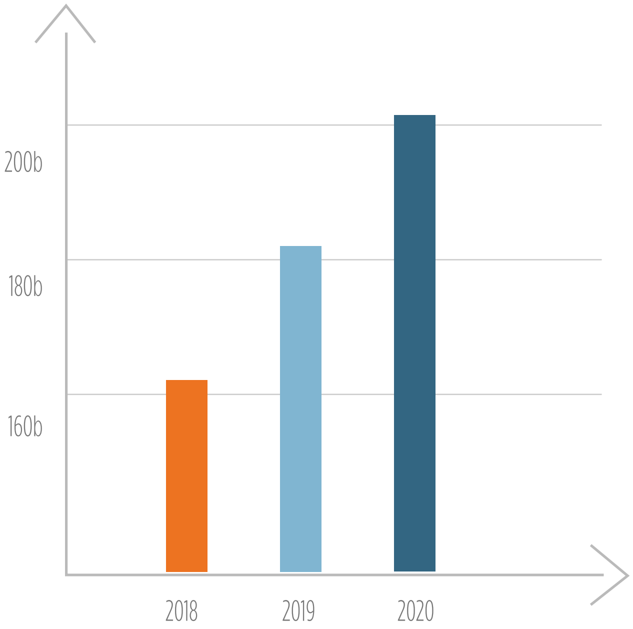
Addressing The Opportunity

IDC says that worldwide revenues for Big Data and business analytics will grow from $130.1 billion in 2016 to more than $203 billion in 2020, at a compound annual growth rate (CAGR) of 11.7%.

– This amount doesn't take into account those that aren't currently utilizing Big Data technologies due to the cost and complexity

–BDB™ addresses underserved markets with a solution that rivals revenues from traditional Big Data

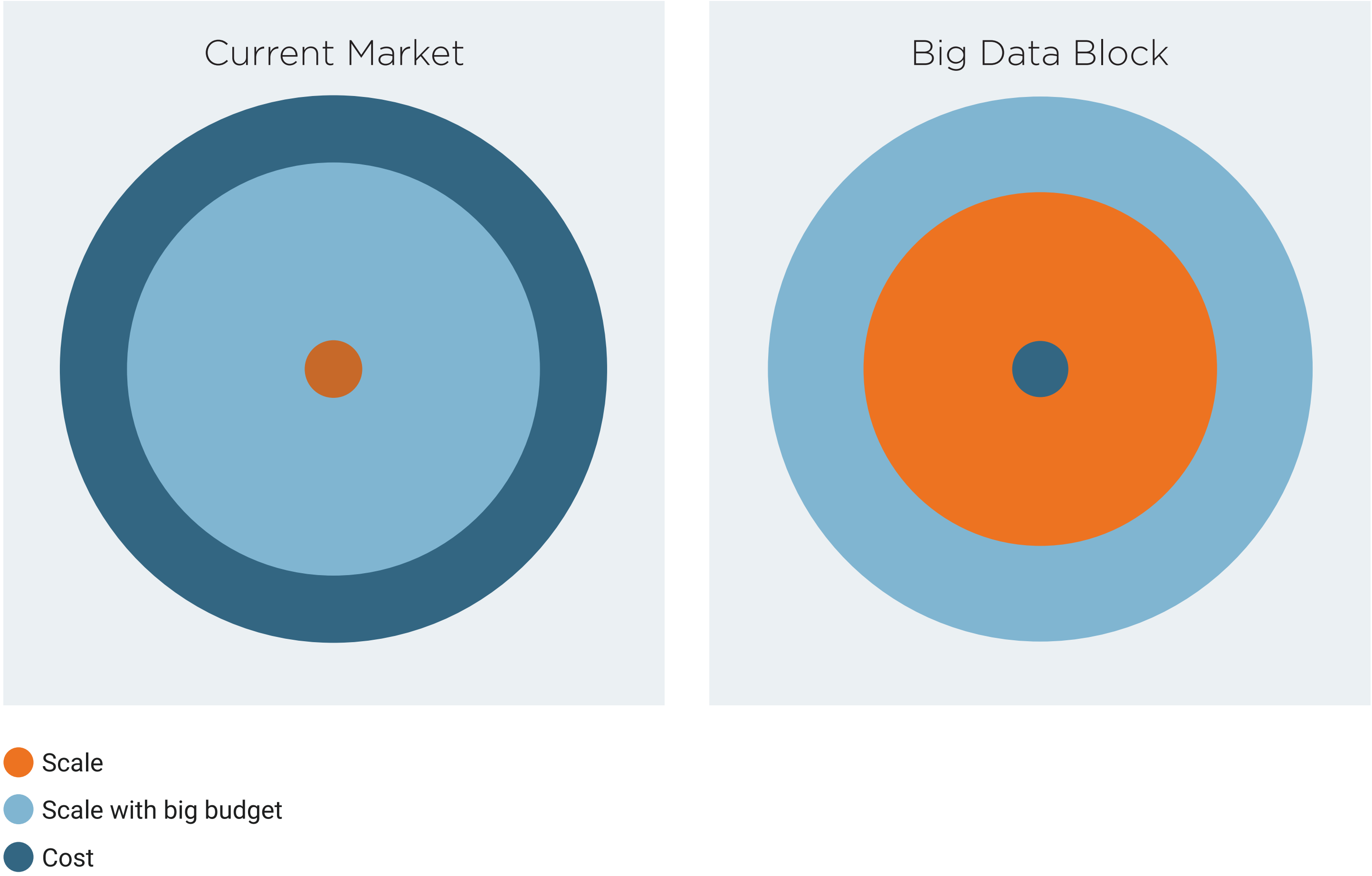
Growth Rate of Market



SOURCE: IDC / Double-Digit Growth Forecast for the Worldwide Big Data and Business Analytics Market Through 2020 Led by Banking and Manufacturing Investments [1]

We're currently in the middle of a giant data boom—which means more information is available to us than ever before. More information equals more opportunities to analyze data to find patterns and solutions that can help your organization grow and your mission to flourish. BDB™ is evolving to empower you to leverage the power of Big Data. At the end of the day, doing more with data starts with asking the right questions and having the tools to procure the best answers.

Cost vs. Scale



MSME Density Across the World

The median Micro, Small and Medium Enterprise (MSME) density indicates there are 32.18 MSMEs per 1,000 people.

● ≤ 30

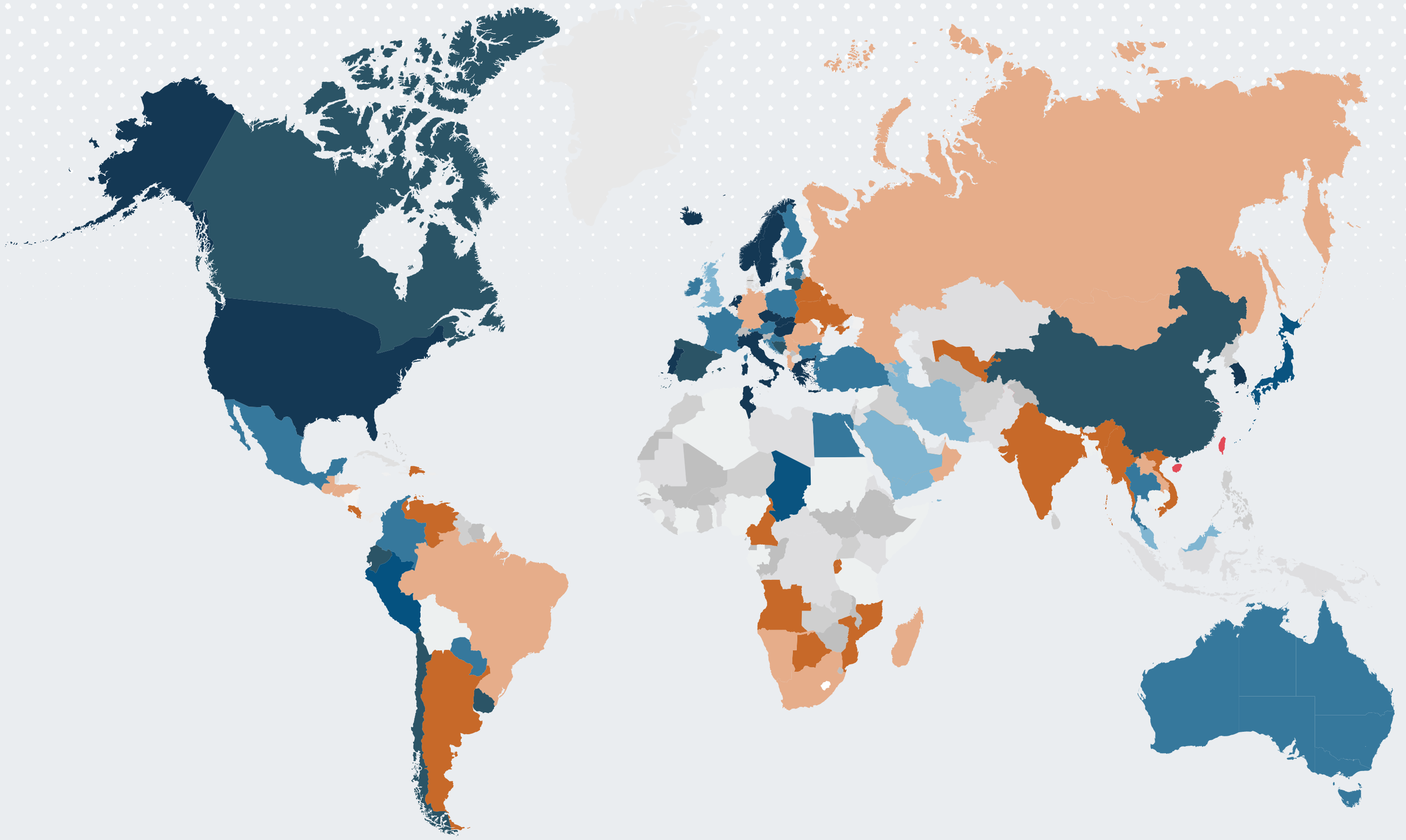
● > 30 ≤ 40

● > 40 ≤ 50

● > 50 ≤ 60

● > 60 ≤ 70

● > 70



SOURCE: MSME Country Indicators[1]

NOTE: The figure uses data from 124 economies. Number of MSMEs and observations per region are included in the table.

Region	Number of MSMEs	Observations
East Asia & Pacific	61,860,488	13
Europe & Central Asia	6,230,701	19
Latin America & Caribbean	13,737,962	17
Middle East & North Africa	5,858,026	9
South-Asia	7,534,153	4
Sub-Saharan Africa	1,105,190	12
High-Income non-OECD	5,949,612	21
High income OECD	60,529,338	30
TOTAL	**162,805,470**	**124**

BDB™ Token

UTILITY OF TOKEN

The job metadata publishes to the Ethereum blockchain using our token smart contract that is aware of all the job specifics. This job contract is then analyzed automatically, decomposed into subtasks then assigned out to the miners, i.e. hosts in our network running the BDB agent, based on which set of miners fit best for the task at hand. Ethereum is used to manage the end-to-end transaction and carry the details of the job in the smart contract. Job hits our grid running Apache Yarn, or similar, to determine all nodes available to process and then distributes the job. Nodes will have installed our data processing engine (Hadoop based) via a Docker container and then they will be announced to Apache Yarn to be included in the grid. Job executes on all the nodes and as each node is done processing it sends the data back to our grid that combines it all together. We control this middle layer to manage the data movement and assembly. Data is then placed on a secure location and customer is alerted that their job is ready for pickup.

HARD CAP

We want the community and supporters to maximize the returns for funds allocated. To achieve this, we will have a cap on the sale of tokens and this sale will be on a first-come, first-served basis to a limited number of participants. The token sale will begin in the autumn of 2018 and run for 30 days or until the total amount raised equals the equivalent of USD $29.5 million dollars.

STRUCTURE

Our goal from day one has been to offer a majority of our tokens to early adopters. Our structure allocates a majority of the tokens to the crowd sale for this reason.

Token Sale – Tokens sold during the crowd sale.

Founders and Advisers – Held by founders and advisers.

Reserve – Future incentives for employees, data scientists and the BDB™ community.

Private Sale – 50% discount for a limited period starting June 1, 2018 and runs until sold out. We are capping the amount and making a few tokens available to early adopters.

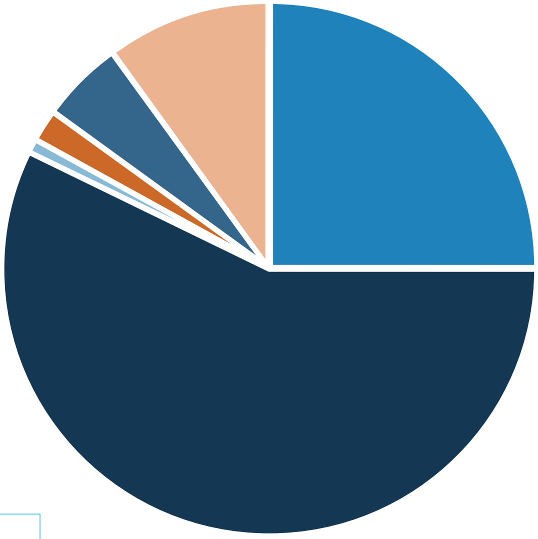
Bounties – Rewarding those that believe in BDB™ and help get the word out.

Non Profit Fund – Our goal is to find ways to help the world. Tokens will be issued to help students, nonprofits, data scientists, researchers, academic institutions, NGOs, governments, and others who can make a measurable impact in the world. The tokens held for this will be in a wallet that can be tracked by the community and all use of these tokens will be tracked as part of our governance process.

Tokens

The number of tokens is capped, and these tokens pay for access to BDB™. Therefore, the more successful BDB™ becomes the more demand there is for our tokens.

- ● Token Sale - 55% or 55 million
- ● Founders and Advisors - 25% or 25 million
- ● Reserve - 10% or 10 million
- ● Non Profit Fund - 5% or 5 million
- ● Private Sale - 4% or 4 million
- ● Bounties - 1% or 1 million



Token Name	BDB™
Lock Up (Advisors)	6-Months
Lock-Up	18-Months (tiered 1/3 released every 6 months)
Total Supply	100,000,000
Token Sale Distribution	59,000,000

TOKENOMICS

Customers

Payment

Our token provides an easy way to transact that leverages the inherent logic that allows for easy tracking and auditing.

Component Reuse

Our smart contract carries with it the instructions used to setup all phases of the job. This allows us to easily recreate any component from scratch at anytime without actually storing the components. Storing job information for the most current jobs as well as the entire history of all jobs every executed on BDB™.

Trust/Provenance

The sharing economy easily translates to every component created in BDB™. BDB™ makes offering your components or data frictionless. Sharing for free or selling for BDB™ tokens. Our ecosystem gives you the ability to know what components are being used by whom, who created these components, who has modified these components, and validates the provenance of the data. This seamless tracking is one of the many advantages of the BDB™ token. Some movement of components or data have the option of becoming a blockchain event, one that can be tracked even if they were free. This might result in a small amount of gas costs for BDB™ which aligns with our mission to democratize Big Data and foster the exchange of data.

Ecosystem Economy

Those offering services like data mapping, data cleansing algorithms, machine learning logic, basic analytics help, and data science analysis help are all part of an equation that also provides payment, tracking, auditing, and trust.

Data Processors/Miners

Payment

There are many payment events tied to the processing of data with payment in token as the preferred method.

Profit

Type of Token	Mining Profit (Monthly Average)
ETH	$75
BDB *	$630

* Based on full capacity of 2,520 jobs completed at a two-hour average job with a payout of $0.25 per job

Tracking
All jobs have the option of being tracked through the process, and to be easily auditable via blockchain events and contract states.

Support
Job failures happen and systems can go offline during processing. In order to inspect these failures we have end-to-end traceability that can be traced back to the source of the issue. We provide this info to any processor we did not pay due to issues in the processing.

Token uses cases

Data Processing
Miners process data for BDB™ tokens.

Payment/Fees

BDB™ charges small fees on top of the cost to process the job and any sales on the exchange which the token facilitates.

Tracking
Audit everything that happens in the system.

Metadata Analysis
Providing interesting reports, insight and new datasets about how the system is used. What industries are most popular, are people sharing data, what algorithms are most popular are all examples of BDB™ reporting. Uniquely identifying or sensitive customer info is never exposed in these, by safe default.

Use of Funds

Product Development	50%
Marketing & Communications	30%
SG&A	15%
Security & Audits	5%

Committed Customers

We already have several customers committed to BDB™.

Participate

Please visit bigdatablock.com and pre-register for the ICO.

Bonus Schedule

Additional tokens will be allocated during these time windows:

	Discount
Private Sale	50%
Day 1	25%
Week 1 (-Day 1)	20%
Week 2	15%
Week 3	10%
Week 4	5%

INITIAL COIN OFFERING

Token Sale	March 2019 and run for 30 days or until the total reaches hard cap
Currencies Accepted	BTC, ETH, LTC
Token Purpose	BDB™ tokens pay for access to the BDB™ platform, which allows users to run Big Data jobs
Supply	100 million total (59 million available in token sale)
Hard Cap	USD $29.5 million

The Big Data Block Network

DATA CLEANSE

The ability to set thresholds around how clean your data is and have rules available to cleanse some of the data where possible

DATA MANAGEMENT

The ability to manage all the metadata for any job created allows for the use of BDB™ to create a data dictionary for all your data assets

DATA ANALYTICS

The ability to run analytics against your data

DATA LEARN

Create or plug-in existing machine learning algorithms to start taking advantage of self-learning algorithms

DATA EXCHANGE

The ability to share data from all the modules for free or fee allowing the easy exchange of data assets from cleansing rules to the analytics output

DATA TRUST

The ability to track any data component through the chain of ownership within BDB™ to prevent data theft and allow for data purchasers to feel they can better trust the quality of the data

Knowledge Exchange

Big Data Analysis is Hard. We Get That. That's Why We're Building a Community of Big Data Experts.

Big Data Block is evolving to become the engine that drives large-scale data analysis globally. The adoption of BDB™ creates an ecosystem around data and the creation of a truly democratized data economy. The users of BDB™ have the ability to share any component they have built with others either for free or in exchange for BDB™ tokens. Anything created on the system has the ability to be shared on the platform. The ability to share all creates new economic models for some as well as provides the ability to move much faster and team with others to gain a much richer view. This ecosystem supports our user driven support model, which helps everyone reach their data goals.

The BDB™ Knowledge Exchange (KE) allows people within the BDB™ ecosystem to offer their data components, data services, and/or raw data with anyone they choose to.

Some examples of typical BDB™ shared components are:

Data Cleansing Algorithms

Standard algorithms used to cleanse for duplicates or incorrect formatting for example.

Analytics Logic

Logic used to determine the output parameters such as high-value customers or operational statistics.

Data Mapping Templates

Many industries use standard data formats. Those knowledgeable in those formats can create these template mappings.

Visualization Logic

The creation of standard data visualizations.

Data Analytics & Support

Raw input or output data can easily shared.

We recognize that there are times when additional help may be needed to assist our users to reach their ultimate data analysis goals. This is provided by the community with a marketplace that matches our users with technical support experts.

BDB™ helps match those needing assistance with a data expert to discuss the details of the proposed project, including the fee, which will be paid in BDB™'s. After the arrangement has been agreed upon and once the work is accepted, the detailed info on the individual users will be available so the work can be completed.

BDB™ will also be provide tier two support if and when needed. This will be a paid service provided by BDB™ for those needing help outside of the community.

Some examples of typical BDB™ support needs are:

Data File Creation

Non-technical users may have data sitting in multiple places and they aren't sure how to structure it for upload and will seek out technical assistance.

Data Analysis Setup

Setting up the analysis side to determine what items the user is looking to better understand might require additional help from someone with data science experience.

Data Management

Where is all my data and how do I create a data dictionary to track it going forward?

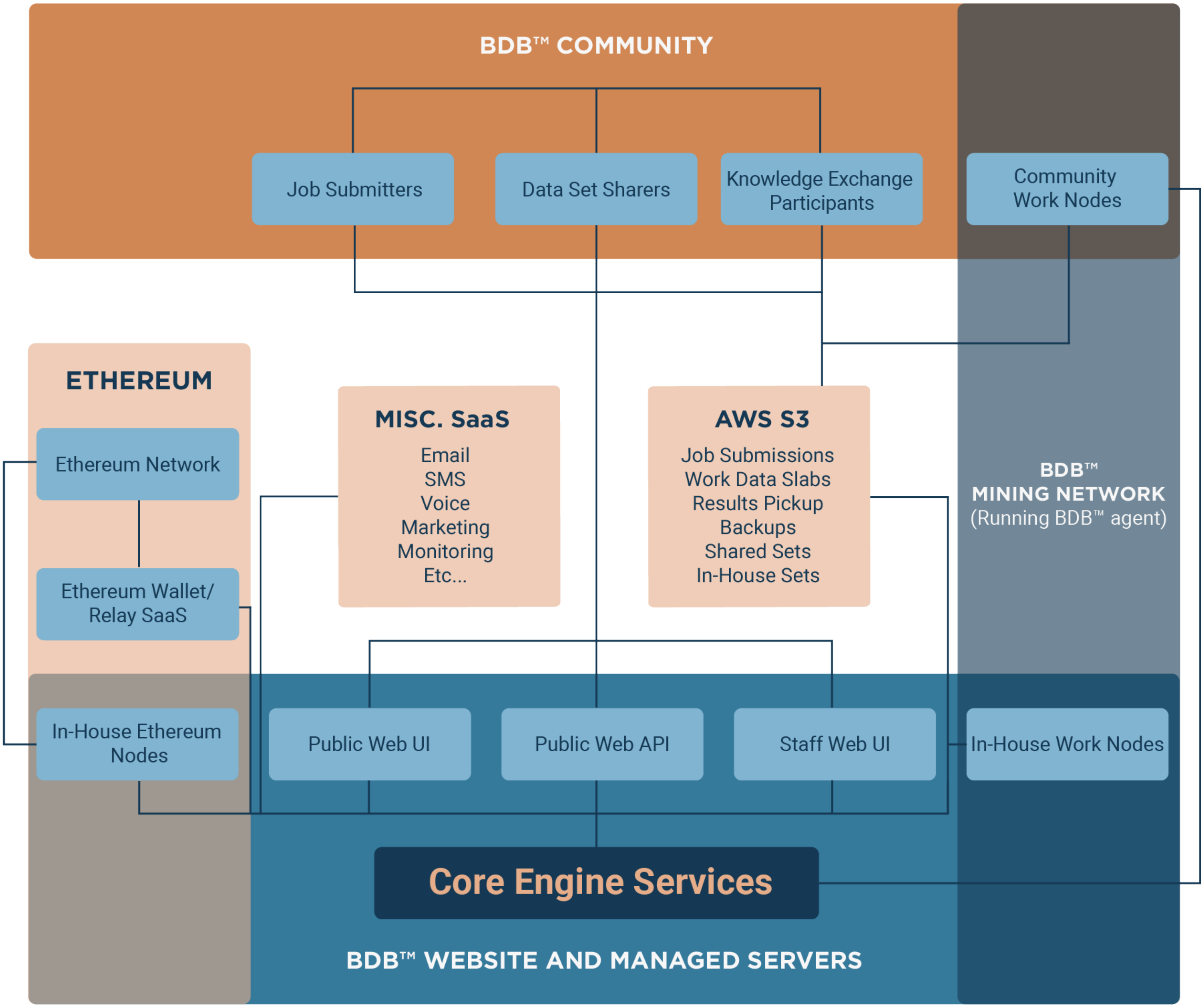
Data Presentation Assistance

Help creating presentations that map the data to real business issues.

Advanced Statistical Analysis

More deep data science needs.

Technical Details



JOB INPUT

There are multiple ways to register jobs to be worked by the BDB™ network. A traditional HTML/HTTPS-based web UI, at the BDB™ site, is likely the simplest way to get started, especially for the less technically adept users, and therefore is the default we recommend. There is also a modern RESTful JSON/HTTPS web API. Lastly, job submission can also be done via the Ethereum blockchain, on the main public network. This is done by creating, signing and broadcasting transactions out to the Ethereum network, which call certain methods on a single official BDB™ gateway contract responsible for accepting new job submissions. There are advantages and trade-offs to each submission method and therefore we let users decide for themselves. Under the hood and behind the scenes the system has a unified database, a common job pipeline and it provides controls and views over all submitted jobs: their status, history and outcome. One big advantage of the Ethereum blockchain mechanism for job submission, however, is that it provides a public, globally shared and distributed record of whether any given job was requested or done. For enterprise customers that require more privacy or anonymity, BDB™ will allow a private option leaving the job specific data encrypted.

Due to the nature of data processing work some jobs require input data in very large sizes, measured sometimes in gigabytes or terabytes. It would be impractical and unwise to devise ways to cram this input data (or any fat results) into the Ethereum blockchain. Therefore they will be stored and made available elsewhere. A job submitted via the blockchain will normally not have its input data embedded in the blockchain, and instead only identify the location of that data, e.g. the URL, any credentials needed and ideally a content hash to confirm authenticity of that data when fetched later for processing (in order to verify it). This is also to ensure no raw data is ever exposed on the blockchain. We would only ever put job metadata in the blockchain.

DATA SETS

Composite Input

A job can draw upon one or more distinct data sets for its raw input. For example, a single huge set of weather data for the last 30 years in North America. Or that set plus a data set with curated statistics on real estate sales over the same period. Or the weather, plus the real estate data, plus statistics on crime and political voting patterns.

Size Variety

Small data sets might not need a Map/Reduce approach to processing, and they do not need to be sliced up for placement across a distributed file system. For example, in the KB to MB scale or even smaller. But they are still needed as one of the several sources of raw input or call parameterization for a particular job, and thus our system must support them just as much as the traditionally larger set sizes usually measured in GB, TB or PB.

Multiple Formats

Our system supports multiple dataset formats including but not limited to all Hadoop compatible formats.

Multiple Origins & Terms

A job can specify data sets that the job requesting user provides, though they are never embedded directly in the submission itself -- instead it points to where the data is hosted elsewhere, just as long as our system has a way to fetch it when needed. Some data sets are provided by BDB™ as free samples that we host (especially for trial and testing), or made available only for a nominal access fee expressed in BDB™ tokens or as a job fee percentage. Members can also choose to register additional data sets with us, in a persistent way, and then make them available to the entire BDB™ community for use in their jobs, again either for free or for an access fee in BDB™ tokens. Any data set that is large enough–and cleaned and curated enough—on a topic of broad relevance will surely be in demand by users because it will be useful for answering many types of questions.

Provenance, Rights & Ratings

When known we recommend that data sets shared with the community describe the provenance of the raw data contained therein, and whether/how it was curated or cleaned, who/what gathered it originally, and how. Also, declarations of any rights or restrictions upon the data—though at a minimum the implied & required rights & restrictions must be such that any BDB™ job running in the network can draw upon the data, and potentially share out their own results or derived data sets, if and how they wish, and within the BDB™ network. Additionally, each data set has statistics tracked on how it's been used, how much and how recently, as well as both numeric ratings on quality and usefulness, and textual feedback and reviews. Basic objective statistics for each data set, like byte size, record counts and dimensionality, are published where provided or calculated automatically.

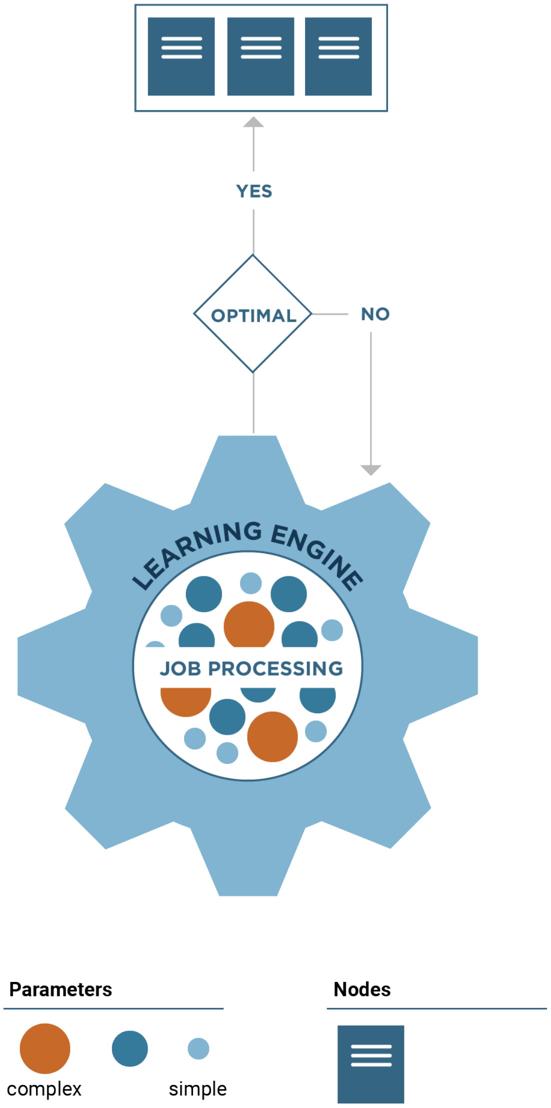
Job Scheduling

When a job is submitted the user can specify whether they want it to begin execution immediately, or as soon as possible, or, simply to have it completed and results ready by no later than a certain date and time. There are tradeoffs to each, therefore this strategy lets each user decide what fits their own needs best. Generally the more flexible a job is on when exactly it can begin execution, or finish, means that the BDB™ token fee demanded can be lower. If a job needs to be done quickly, then the total fee demanded will be higher. Recurring jobs can also be configured, e.g. to run once per hour, or daily or monthly, etc. The BDB™ system makes a best effort attempt to schedule everyone's job execution to satisfy their requirements and preferences. Where a job's requirements cannot be satisfied, or some kind of exceptional event occurs, a notification will be made available to the job's owner, using whatever medium they've registered with us as their preference, e.g. in-site popup/badge, email, SMS, automated voice call, or Slack bot message. Though BDB™ wishes in the ideal case to have all work host capacity provided by others—by the community who've chosen to integrate with us and run our software in order to earn tokens—it is also important to ensure there is always sufficient work capacity available to run all requested jobs. Therefore, if at any time it appears to the system that there is not enough organic work capacity available from community hosts, to run a particular job, then the system will make a best faith effort to spin-up in-house instances of BDB™ work nodes (hosted at a cloud provider that allows dynamic provisioning on demand, such as AWS, and who otherwise meets quality requirements).

Job Fee Matching & Requirements Optimization Engine

A fundamental challenges for a system that provides data processing as a cloud service, using a heterogeneous grid (or “fog”) of hosts with potentially widely varying hardware capabilities and availabilities is what to charge, what is the right fee to perform the work, this specific job and at this specific time, especially when considering the tradeoffs caused by the differences in node capabilities and data pre-positioning. The job submitter naturally wants to pay as little as possible. The work host provider wants to charge as high as possible. The submitter wants to know that the work will be done, but probably has some upper-bound on what they’re willing to pay. And perhaps a time limit, e.g. results ready in less than 1 hour after submission, or preferred time window for execution, e.g. result ready no later than Friday, or that the work should execute during whatever periods will likely demand the lowest BDB™ token fees, such as during off-peak or otherwise idle hours.) And almost always some minimum expectation of quality, and validity. Or privacy and anonymity. Therefore our system is designed to say yes to all these demands, and to strike a middle-ground that satisfies all these goals.

Default: Mode 1



By default the system will do the calculation on the backend using no particular logic to speed up or slow down any job. It’s FIFO and processes based on fee logic noted above. We recognize more flexibility and granularity will be needed at times. In order to provide maximum flexibility two modes for job execution will be provided:

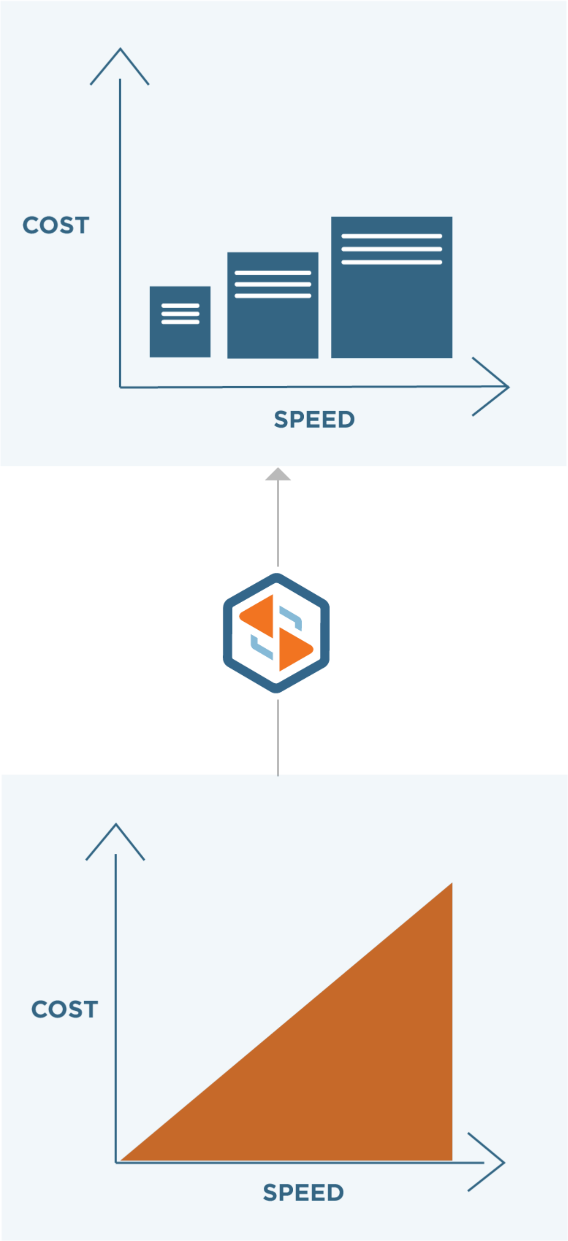
Mode 1—This is the default and uses our job costing algorithm to continuously monitor the multiple facets that go into the processing of each job to determine optimal nodes and costing. Costing algorithm will be similar to ((Data Size x Number of Modules Used) x Complexity Score).

This will continuously be tweaked as we use our own system to update the model and ultimately get extremely refined in our costing.

Mode 2—This will allow the user to control some parameters of the jobs performance. If they want faster processing for a higher fee or slower processing for a lower fee they can control these dials. This will be matched to auction type logic based on the input from the data processors combined with the system specs on each node. Higher end machines will be better suited for the faster processing and smaller nodes for the slower jobs. Both groups can be successful in this model

The entire job fee matching and preference optimization process is 100% automated in the BDB™ network. The user submitted jobs and the work host providers all register their requirements and preferences and our data processing system does the work to figure out the best solution, all in real time, behind the scenes. Optional prompts for manual overrides, and status change notifications are also provided. But in the “happy path” case BDB™ is designed to make reasonable choices for everything, by default, out-of-the-box.

Mode 2



Token Architecture & Lifecycle

BDB™ is using an ERC-20 utility token contract, built on top of Ethereum. There is also a single official gateway contract for new job intake, for those who wish to use that interface to submit jobs. The gateway contract is a distinct, separate contract from the contract that manages the ERC-20 token balances. The token balance contract is immutable, with no upgrade-in-place mechanism—in order to boost trust by users. Whereas the gateway contract does have an upgrade-in-place mechanism (based on proxy/delegate/lib patterns which can preserve desired state) to make it easier to add features or fix issues over time, and do so in a way that is seamless and therefore more convenient to users. The token balance contract is an unmodified, off-the-shelf codebase which is free, open source, widely used, automatically tested, aggressively audited and trusted by experts with the relevant skill sets.

Ethereum Network Integration

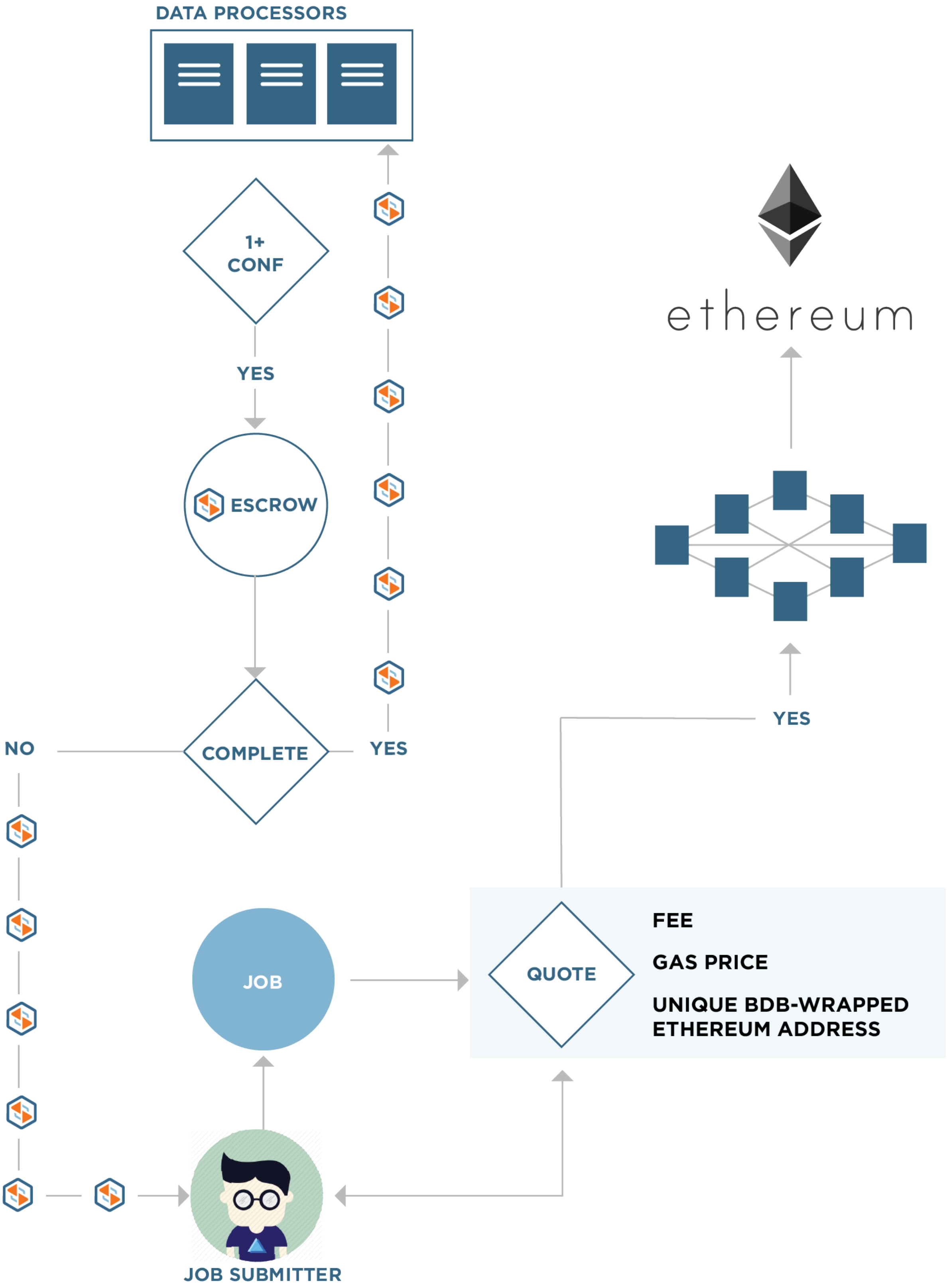
Once the total BDB™ fee needed to complete a job is decided a quote is shown to the job submitting user, including both the gas cost (if any, and in Ether), and a unique (BDB™-wrapped) Ethereum address (in a Big Data Block-controlled wallet), freshly assigned to this job run. If the user submits their acceptance of the quoted fee then the next step is Big Data Block waits to see at least the first zero-conf (unconfirmed) broadcast of the kind of transaction expected. Our system listens to the main public Ethereum network via a set of redundant network links. Upon seeing the 1st zero-conf appearance of the expected transaction the job preparation tasks may begin. But no real work begins until the first (at least) single-conf event is observed—that is, until the expected transaction has made it into a newly mined block. At this moment the BDB™ fee is considered held in a state of escrow by Big Data Block, and so it will be available to pass on to the involved host providers upon job completion, or, possibly refunded back to the job-submitting user in the case where the job was not completed.

The system is designed to keep the minimum amount possible, at any time, of non-escrowed Ether and BDB™ tokens available for programmatic access by the production system, or in so-called hot storage. Periodically, or whenever certain levels are triggered, an automatic process runs to look for excess Ether/tokens and if found they are swept out to cold storage addresses for safer-keeping. Some buffer amount is held back to ensure the system has enough Ether to cover transaction fees and any gas costs and an automatic process also observes the buffer amounts and if they look too low then additional amounts are moved back online to top it off.

All movement of Ether and BDB™ tokens is logged for audit/accounting purposes, and written out to multiple redundant, secure locations. Watchdog processes also monitor all movements, independently, along with the audit logs, looking for any concerning patterns and notifying staff (redundantly) as needed.

The escrowed fee mechanism helps to ensure that all parties get paid what they expect, no matter what hap-

Network Integration



pens with the job. The fees will be there to pay the work hosts, or, to refund back to the job owners. However, our system takes it a step further in terms of redundancy and safety in that our design also holds an amount in an Escrow Loss Reserve (ELR) fund. The ELR is some minimum amount of Ether and BDB™ tokens held back in a separate, offline, air-gapped cold wallet, which can be used to issue an absolute worst case full refund to all job submitters, in the event of a scenario where all outstanding escrowed fees were lost.

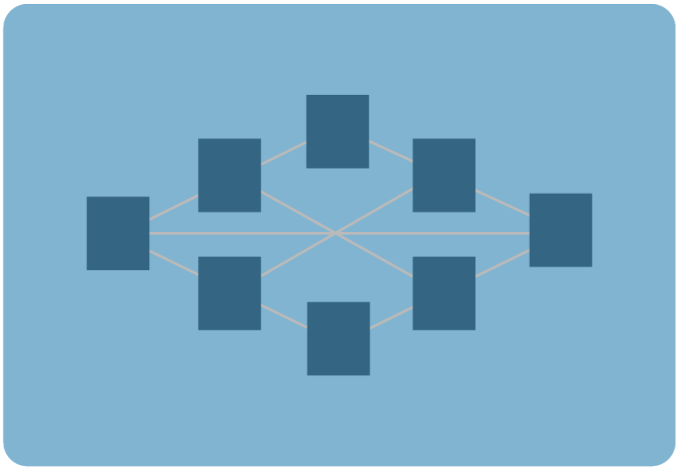
For the wallets themselves we also feature a redundant architecture to increase availability and throughput, and reduce risk. First, our bias is to use a professional, third-party wallet service, dedicated to its feature space, and whose entire mission is to provide a secure, private, highly available and performant service. However, even then we know that by relying on a single vendor we're at increased risk of a single failure on their end impacting us significantly. Therefore we'll integrate with at least two. Each must otherwise meet our requirements around feature set, security, performance and scale, and track record. Additionally, each must appear to be capable of handling our entire traffic load if needed. Though in the normal case we'll split our needs across them evenly. We'll also require that each has a sandbox mode that lets us run full integration tests against their interfaces, including load, stress, and "white hat" simulated attack edge-cases. We will also deploy our own in-house managed wallet capability, and ensure it too will run satisfactorily under all expected conditions in our automated tests. Though normally, in production, they'll hold no real actual Ether or BDB™ token balances. Our in-house capability will only be as a fallback mechanism, in case of a failure of most/all of our third-party vendor services—we simply want to have the capability ready to go, immediately, if ever needed. Our fallback wallet will use 2+ distinct wallet/node codebases, running on 2+ distinct OS'es (on latest security patches), running on 2+ distinct CPU/board hardware architectures, in 2+ different data centers, in 2+ widely separate geolocations and jurisdictions, etc. Also, all wallets anywhere that our system touches require M-of-N multisig, e.g. two of three, to move Ether or BDB™ tokens.

We use HD (therefore, all keys recoverable if have the root seed) wallet regeneration where possible. Paper keys, memorized seed phrases, airgaps, USB hardware dongles and a standard set of personal OpSec practices by staff.

In addition to using a redundant set of Ethereum network services, and clusters of backend processes, which allow us to horizontally scale up our total traffic handling capacity by the central Big Data Block management engine, we also maintain the BDB™ utility token balances in a sharded architecture. We've already explained how we'll reduce risk and increase trust by using the ERC-20 standard for the token interface, and using an off-the-shelf codebase with a successful production track record, positive feedback by experts, etc. But we want to take this a step farther, via sharding. This means that we'll maintain the total amount of existing BDB™ tokens across a set of public contracts. A small number of contracts like 3 to 5, but otherwise identical. Rather than just a singleton. With the total token supply split evenly between them. Again, also immutable and totally transparent to the public. The reason for this sharding approach is to reduce the downside risk to token holders in certain kinds of theoretical scenarios where a security breach causes a key to be compromised. The ownership key associated for each shard, for each of these 3 to 5 contracts, is distinct and maintained in separate locations, with separate access & authorization funnels, at all times, and with tight security controls, and yet also while deployed on heterogeneous environments—to further reduce risk of unauthorized staff access, or from zero-day exploits by remote black hat attackers. Big Data Block will declare in public the identity (contract address and sig) of each of these shard contracts at all times, as well as maintain a single signpost contract—meaning, no functionality, and is purely informational—which also points at them, and is immutable.

For any transactions broadcast automatically by the system, using M-of-N multisig, the intent is to mitigate scenarios where any given single management host fails, or becomes compromised. If any single host, belonging to the set of N, fails and its key is lost or otherwise inaccessible then the remaining surviving M of the N set can still sign the transaction and the system runs smoothly. Likewise if any one of the N hosts is compromised by a malicious actor, and its key compromised, then that malicious

Wallets With Redundant Architecture



actor will not have enough authority to issue a bad send, e.g. a theft attempt. The goal of Big Data Block is zero touch, meaning for normal production use the live site doesn't require manual attention by engineers. Therefore many of the Ethereum/BDB™ transactions are automated in software. Some transactions need to be reviewed by human staff and given approval to complete. And certain transactions require manual authorization, by staff, of an M-of-N multisig transaction, and in the most important cases: by 2+ pre-designated officers of Big Data Block.

Escrowed Fees

Users give BDB™ tokens to compensate those who provide the compute hosts who work their submitted jobs. However the nature of Ethereum, and most cryptocurrencies, is that once coins are sent, as part of a transaction, and confirmed in the blockchain, that the transaction cannot later be cancelled, and the coins cannot later be refunded. They're gone. If users sent their BDB™ tokens directly to the work host providers, upfront, and subsequently it turned out that the job was not ultimately performed or completed, or a particular provider or host delivered corrupt or fake results then there would be no way to cancel or refund the payment. Likewise, if the user wished to delay payment until the job was completed, there is a risk to the work host providers that they won't actually get the pay, because there is a chance the user ends up spending their tokens on something else in the meantime, or loses access to his wallet, or loses connectivity, and so on. But this issue is not a problem with BDB™. Because we feature an escrow mechanism to ensure a best of both worlds outcome: a job submitting user does pay tokens upfront, into a temporary holding account, controlled by BDB™, in order to ensure the host providers that the tokens are truly there and will be paid, and then, next, after the work finished and job completed, BDB™ sends these escrowed tokens onward to the host providers. Minus BDB™'s fee. This ensures that all parties get what they expected.

Results Validation

The secret ingredient to tackling the processing of compute tasks upon giant input data sets, and especially to have them complete as fast as possible, is via a hyper-parallelized approach across a massively distributed architecture, with the more nodes—and the beefier, and the fastest bandwidth—the better. The weakness with this approach is that as you increase the number of work nodes you also increase the chance that any single node might let you down at any moment. Either via a hardware or software fault, or actively malicious behavior. BDB™'s network features a heterogeneous community of a variety of host providers, with varying hardware capabilities, quality, and trust. Though in the normal happy path case most nodes will be reliable and trustable and clean of malware. However, a good system should assume that from time-to-time a node might be faulty or malicious. (This is also what Google does, and our system is heavily inspired by their architecture, especially in our use of the Map/Reduce algorithm, via Hadoop.) Yet, users still want their jobs to finish and to be able to trust their results. Therefore BDB™ supports the ability to allow any job, or sub-task, to be rerun some number of times, redundantly. The redundant runs would occur on different work hosts, to help ensure that they would not all be impacted by a host-specific or platform-specific flaw. And the redundant runs can happen either in serial order, or in parallel, depending on the job. And the total number of redundant runs may also be specified. The user submitting the job gets to choose whether and how much redundant execution occurs, because there are trade-offs. To get higher confidence in the correctness of the job's results then greater redundancy is needed, but that means a higher total price for the job (because all hosts must be compensated for their time.) To get the lowest possible price then there would need to be no redundant execution—a single best- faith execution of any task, and only on a single node. The job submitter can turn the dial to their preferences, but the system defaults to a single-run.

Results Authenticity

Beyond being valid, meaning correctly and honestly calculated, there is also a desire to know that a job's results are authentic. Meaning that the version of the results you're seeing now is exactly the same as the version you first saw, and the same as first completed and delivered. If a job completed and the results were correctly calculated and stored but then, for any reason, something happens to those results to alter them, to mess them up—again, either through hardware or software fault, network transport fault, or malware—then the results might be just as useless, or at least not trustable. Therefore the BDB™ system ensures that the moment a given candidate snapshot of a job's results has been assembled and considered completed and valid, that also a signature is calculated, based on that results snapshot via a hash algorithm appropriate for fingerprinting. Computed from both the contents and all relevant metadata of the results, including the job itself and its parameters. This helps to record a moment in time where a single unique version of them are considered authentic, and as part of a matched complete set of all job-relevant data. These fingerprints get stored along with the results. Later when a user goes to view or get delivery of the results—and especially after they've fetched the results across the wire to some new destination under their own control—that this user retains an ability to independently recalculate this fingerprint, using the same algorithm, and therefore independently verify they're looking at the right version of the data expected.

Cached Results

When a new job is submitted the system checks first if there happens to be results already sitting in a cache. If the job is such that it behaves like a traditional function—meaning its output depended only on its input data set and nothing else, and its behavior was deterministic, meaning not impacted by or relying upon any randomness—and if that input data has not changed, and the job's algorithmic work code also has not changed, then it is possible on subsequent requested runs of the same task that we can take a shortcut and avoid doing unnecessary work. If it has been run before and the system still has the prior results cached then it simply makes them available again. There will still be a small BDB™ token fee, going solely to BDB™ for providing the cache, but it will be a much lower fee than otherwise, and as a bonus the results are made available much faster—perhaps immediately. Lastly, this caching feature also helps free up additional work capacity on the BDB™ network.

Likewise, if any partial sub-task of a newly requested job is cacheable and we do have those partial results in cache, then the system makes those available to the new run, with the rest of the sub-tasks (the ones not cacheable or which simply lacked their own cached sub-results) having to be worked fresh as normal.

Fetching Results

Initially shared via a repository on Amazon S3 that's secure and auditable with the long-term goal of looking to a possible partnership with one of the file sharing blockchain companies.

Visualizing Results

Using open source tools we provide the ability to do data visualizations on the data returned, with options to pull in the results to the visualization tool or to simply download the data to be used outside of BDB™.

Work Host Capacity

The work host capacity in the BDB™ data processing network comes from a variety of sources. Some may come from otherwise amateur or hobbyist contributors to the community, who might provide only one or a few work nodes. And their nodes might be dedicated to BDB™, and available 24x7, or, they might run only during otherwise idle time periods or compute cycles on their computing host—perhaps a spare office server, a home desktop or personal gaming rig, or tablet, and only when Internet connectivity is present. And some will likely come from more professional operations that provisioned dozens to hundreds of robust servers and then deployed the BDB™ software upon them as a smart strategy to earn additional revenue and profit over and above their investment in the hardware or infrastructure services. Lastly, BDB™ itself will also provide some work host capacity that we ourselves pay for and manage. The reason for this latter capacity type is both to help bootstrap the BDB™ network's work capacity at launch, and, post-launch to help ensure the network always has some minimum level of work capacity available at anytime—in the case where the normal providers are having their hosts come online and offline, with normal fluctuations over time including over daily, weekly and seasonal cycles, as well as due to some hosts having intermittent network connectivity.

BDB™ will be analyzing the node processing on a regular basis to flag nodes consistently having processing issues. These nodes will be quarantined and put into a probationary period until they can be validated as functional. If this is a repeated issue for a particular node it will eventually be blacklisted from the network.

Scalability

The overall scalability of BDB™ is driven primarily by three specific areas.

First, the total global data processing capacity of the system is driven heavily by the number of participating work nodes, their capabilities, reliability and network communication speeds. In broad-strokes scaling this aspect up is nearly linear: approximately speaking, to handle 10x the workload we'll simply need to grow the processing network to have 10x the number of nodes. Luckily from an engineering perspective this is a kind of self-solving problem because as long as BDB™ is popular and worthwhile to integrate with then more and more people and hosts will join and run our software over time. We only have to ensure they do have that incentive, on the host capacity side even more so than the job demand side, though we want both to grow massively. However, our advantage in this area is the fact that even in an extreme edge case outcome where the system ends up with a huge amount of job demand, but zero participation by the community on the work host side, we will still be fine because BDB™ itself can ensure we always have enough work nodes in the network, simply by provisioning them ourselves.

Secondly, the nature of Big Data processing is that the system will need to move very large blocks of data across the network—it will try to minimize any movement which is unnecessary, having a bias to moving task code out to where the data already lives—and that movement will have a cost. For smaller or more amateur participants in the work host community they might be connected to the Internet using expensive data plans, ones that bias to low bandwidth need or higher latency, and perhaps with hard caps. So if the most efficient solution for completing some big job might involve moving a 10 GB data slab out to somebody's node, but moving it there even just once would cost them say $60 in one-time mobile data fees then its harder and less likely that that participant can make a profit based on his share in BDB™ token fees earned. Therefore we'll ensure the system makes a good effort to minimize the movement of data slabs, as well as provide controls to allow a participant to throttle and cap the bandwidth usage of their BDB™ work agent. This will help dramatically to reduce the costs to have a BDB™ work host. Overall, we're confident we can serve this segment of the market and give them ways to participate where it maximizes their incentive.

And third, there is Ethereum itself. Part of the long-term scalability of BDB™ depends on Ethereum. At launch and in the near term we're confident Ethereum won't have significant impact but in the long-term we will want to see major upgrades to it. Architectural enhancements and implementation tuning, or total rewrites, to allow it to either reduce the time between new block generation by miners, or increase the capacity of a block, or the total storage capacity of the main global blockchain—ideally all of these things. Three of the more promising development efforts underway to do this are sharding (of the blockchain itself), Plasma/Lightning, and a switch from proof-of-work (PoW) to proof-of-stake (PoS). We're confident that if they truly roll out any one of these new features there will be dramatic improvements to the Ethereum network's performance and scalability. All three of these upgrades, together, would be even better, and help eliminate any lasting concerns we'd have at this time about the long-term scalability of BDB™ on Ethereum.

Availability

24 x 7 x 365.

Zero-touch: no human engineering staff needed to operate the production site smoothly under regular conditions.

Periodic automated backups (of everything: code, config, data, docs) to 3 distinct locations. Backups periodically and automatically tested to successfully restore.
Servers are cattle, not pets.

No single-point-of-failure (SPOF). Everything and everyone has redundancy. Servers in multiple locations to have geographic & jurisdictional disaster mitigation. Also gives lower latency to nearest users, and higher availability during partial/point outages or traffic bursts on the Internet.

Throttling to reduce DoS.

Crash-only software: architectural design bias such that a normal shutdown is by hard-killing processes or cutting power.

NetFlix-style chaos monkey is active in production.

Security

Chessboard move potential-based analysis as a security strategy, if someone can do a thing, assume they will. If an element of the arch can do a thing, assume it eventually will. For example, if we leave a critical secret in plaintext on disk at AWS and its possible for that disk to be reclaimed and recycled for use by a next customer of AWS, and its possible that customer is malicious (or allows inadvertently the running of malware) then is wise to assume that WILL happen, eventually, and plan accordingly.

Nemesis automation, production is under constant white hat attack—friendly, allied, known, approved—both attempts to breach and DoS. Because it's far better that we and the white hats succeed at finding and exploiting any of our vulnerabilities, giving us a chance to both learn of it first and fix ASAP, rather than having a black hat do it, silently, and causing a disaster.

Homomorphic encryption or other encrypted processing techniques will be researched and used, as they're ready for primetime.

Isolation of the jobs running on each node will be done via a controlled Docker container that doesn't allow unauthorized access (looking in or out) of the job code, its data, or the BDB™ engine itself.

Geography based restrictions will need to be an available feature. Many countries restrict the use of data outside of their borders. The BDB™ system can easily handle this via a geo/IP based system so that if data is marked as geo-restricted BDB™ will only offer that data up for processing to nodes within that country. It's possible this will lead to jobs not being able to be processed. We can augment this by using Amazon's different regions but there could still be cases where there is no local cloud option for us to augment processing and jobs can't be run.

Roadmap

OCTOBER 2017
Big Data Block born

NOVEMBER 2017
Founders team assembled

DECEMBER 2017
Prototype development started

JANUARY 2018
Big Data Block website launched

MARCH 2018
Whitepaper completed

APRIL 2018
Phase 1 prototype deployed

MAY 2018
Phase 2 prototype deployed

JUNE 2018
Token private sale

JULY 2018
Beta customer registration

AUGUST 2018
Phase 3 prototype deployed

OCTOBER 2018
ICO & Beta launch

NOVEMBER 2018
Beta Customer Requirement Gathering

DECEMBER 2018
Knowledge Exchange Development Planning

JANUARY 2019
Data Cleansing Development

FEBRUARY 2019
Data Cleansing Testing

MARCH 2019
ICO Launch

Team

Jason Cohen

CEO

Simeon Schnapper

Director of Business Development

Jim Falvey

Legal and Compliance

Brett Singer

Director of Marketing

Dan Fisher

Creative Director

Stacey Billups

Director User Experience

Mike Kramlich

Architect

Bridget Groves

Project Manager

Bryan Maxwell

Graphic Designer

Amy Davila

Office Manager

Zach Lutes

Community Manager

Kaitie Zhee

Director Media Relations

Advisors

Atif Farid Mohammad, Ph.D.

AI/Data Science Advisor

Peter Bergstrom

Advisor

Gino Yu

Advisor

Rajesh Johnny

Advisor

Enzo Villani

Advisor

Phillip Nunn

Advisor

Nikolay Shkilev

Advisor

Vladimir Nikitin

Advisor

Mark van Rijmenam

Advisor

Legal Disclaimers

The purpose of this White Paper is to present Big Data Block and the BDB™ token to potential token holders in connection with the proposed ICO. The information set forth below may not be exhaustive and does not imply any elements of a contractual relationship. Its sole purpose is to provide relevant and reasonable information to potential token holders in order for them to determine whether to undertake a thorough analysis of the company with the intent of acquiring BDB™ Tokens.

Nothing in this White Paper shall be deemed to constitute a prospectus of any sort or a solicitation for investment, nor does it in any way pertain to an offering or a solicitation of an offer to buy any securities in any jurisdiction. This document is not composed in accordance with, and is not subject to, laws or regulations of any jurisdiction, which are designed to protect investors.

The product token is not a digital currency, security, commodity, or any other kind of financial instrument and has not been registered under the Securities Act, the securities laws of any state of the United States or the securities laws of any other country, including the securities laws of any jurisdiction in which a potential token holder is a resident.

The BDB™ token cannot be used for any purposes other than as provided in this White Paper, including but not limited to, any investment, speculative or other financial purposes.

The BDB™ Token confers no other rights in any form, including but not limited to any ownership, distribution (including, but not limited to, profit), redemption, liquidation, property (including all forms of intellectual property), or other financial or legal rights, other than those specifically set forth in this document.

Certain statements, estimates and financial information contained herein constitute forward-looking statements or information. Such forward-looking statements or information involve known and unknown risks and uncertainties, which may cause actual events or results to differ materially from the estimates or the results implied or expressed in such forward-looking statements.

This English language White Paper is the primary official source of information about the BDB™ token. The information contained herein may from time to time be translated into other languages or used in the course of written or verbal communications with existing and prospective customers, partners etc. In the course of such translation or communication some of the information contained herein may be lost, corrupted, or misrepresented. The accuracy of such alternative communications cannot be guaranteed. In the event of any conflicts or inconsistencies between such translations and communications and this official English language White Paper, the provisions of this English language original document shall prevail.

Citations

PAGE 2 / Andreessen Horowitz
https://a16z.com/2016/08/20/why-software-is-eating-the-world/

PAGE 5 / ID.
https://www.forbes.com/sites/gilpress/2016/08/05/iot-mid-year-update-from-idc-and-other-research-firms/#118585bd55c5

PAGE 6 / IDC
https://www.forbes.com/sites/gilpress/2017/01/20/6-predictions-for-the-203-billion-big-data-analytics-market/#a6d52f920838

PAGE 8 / MSME
https://www.smefinanceforum.org/sites/all/modules/custom/sme_custom/datasites/analysis%20note.pdf