Black Forest Distributed Ledger™

Craxel, Inc.

ABSTRACT

Craxel has developed a massively scalable, distributed database technology to support trustless and immutable transactions. This breakthrough is based on innovations in searchable encryption, and offers unprecedented speed, security and scale. Whereas the majority of distributed ledger systems are currently in Proof of Concept stage, Craxel's technology, based on a patented zero knowledge encryption database, has been architected for immediate, scalable deployment. As a result, this innovation massively accelerates the opportunities to explore alternatives to existing blockchain and distributed ledger technologies.

Blockchain was developed to host a ledger of immutable transactions for a digital currency, Bitcoin, without reliance on a trusted administrator. Although the decentralization of blockchain is often cited as the disruptive innovation, it is in fact the **trustless** and **immutable** nature of blockchain transactions that promises to disrupt existing value chains in many industries. Decentralization on a massive scale and an elegant use of cryptography (Proof of Work) were the techniques for ensuring the trustless and immutable characteristics needed for Bitcoin to function and to attract interest and support. Regrettably, strong consistency, security and performance were sacrificed to achieve these objectives.

Emerging efforts to use blockchain technology beyond virtual currencies are struggling due to the unavoidable implications of these trade-offs. Proof of work and massive decentralization, the key mechanisms which make blockchain such a unique and useful technology for the creation and transfer of a virtual currency (like Bitcoin), are being left behind. Various attempts to adapt the technology for use in ways it was never designed for are resulting in a convoluted exercise of making tradeoffs within the tradeoffs. This adaptation may lead to iterations which are good enough in isolation for this use case or that; however, the **ideal solution** is a technology that not only creates trustless and immutable ledgers, but one which can be massively distributed and scaled while maintaining computational efficiency and allowing broad interoperability.

1. INTRODUCTION

Given the tradeoffs mentioned above, Craxel has made a timely breakthrough in searchable encryption that allows its massively scalable, distributed database technology to support trustless and immutable transactions. Craxel's solution, the Black Forest Distributed Ledger (BFDL), enables organizations to leverage the promises of blockchain technology at unprecedented speed, security and scale. The security of each ledger is compartmentalized, solving many of the vulnerabilities of the existing permissioned blockchain schemes. Additionally, BFDL offers a new paradigm for tracking and creating transparency around offers and acceptances, by creating the framework for two-party ledgers. Finally, since validation occurs outside the distributed ledger, BFDL offers tremendous flexibility for accommodating a wide range of use cases. Black Forest Distributed Ledger is:

Trustless: Encryption of data and decryption of ledger query results occur at the application layer and can be done using whatever encryption method applications choose. BFDL is completely trustless - the Craxel technology only ever sees strongly encrypted ciphertext, and never possesses or accesses the encryption keys.

Immutable: Once validated, encrypted transactions are cryptographically linked (hashed) forming verifiable and immutable ledgers.

Consistent: Transactions occur with fully ACID properties. Ledgers are created and stored in a massively parallel manner and their integrity is assured by a highly efficient Paxos consensus algorithm. Any attempt to alter a prior transaction would be immediately captured.

High Performance: BFDL can perform transactions at massive speed, with orders of magnitude advantages over most blockchain technologies. BFDL can be deployed on cloud or private networks with only small server clusters being required to provide massive performance, security and interoperability.

Secure: All data in a BFDL is cryptographically secure and can be compartmentalized down to the individual record level. Data access is completely granular and can be shared with other applications or parties, such as regulators, on a need to know basis. Unlike blockchain, you control who sees any set of your data.

Flexible: BFDL provides total business logic and transaction validation flexibility by allowing transaction proposal and validation to occur independently of the ledgers. This enables use case optimization and interoperability of ledger data across transaction types, asset classes, transaction venues and trust environments. BFDL can accept a myriad of data types, including temporal and spatial, along with more standard documents as appropriate for smart contracts, and other legal and regulatory documentation.

2. TRUSTLESS LEDGERS THROUGH SEARCHABLE ENCRYPTION

Zero knowledge, searchable encryption is a superior alternative to proof of work and massive decentralization for achieving trustless, distributed ledgers containing immutable transactions. Proof of work is expensive due to the computational effort required for mining hashes and requires tremendous latency by design. Massive decentralization requires eventual consistency which creates complexity around correctness, contributing additional and significant latency. In contrast, zero knowledge, searchable encryption allows BFDL to provide trustless distributed ledgers without the same downsides as blockchain implementations. The ledgers are trustless because BFDL can't read the contents of the ledgers since they are strongly encrypted and BFDL never has the encryption keys needed to decrypt them.

Moreover, BFDL provides strongly consistent transactions instead of eventual consistency, supports thousands of transactions per second, can host millions of ledgers, and provides unprecedented security. Only participants with authority to access a ledger and with possession of the necessary encryption keys can query, create, or validate the ledger's immutable chain of transactions. Since all ledgers within BFDL are completely trustless and zero knowledge, BFDL administrators, insider threats, or hackers cannot view, add, or modify transactions unless they also steal the encryption keys from somewhere else and overcome all of the mechanisms that provide transaction validation and immutability.

3. IMMUTABLE TRANSACTIONS

Bitcoin's method to achieve a high degree of immutability is massive decentralization of its blockchain. Thousands of servers host blockchain and rewriting part of the blockchain would require controlling 51% of those servers. Even if 51% of the servers were controlled, fraudulent activity would almost surely be detected. However, successful rewriting of any blocks would likely be catastrophic for the trust model of Bitcoin.

However, massive decentralization of a ledger to thousands of servers is impractical for most distributed ledger use cases for two reasons: 1) Proof of work and eventual consistency are required when large numbers of

servers are involved. 2) Large latency is a required feature of most blockchains, including Bitcoin. Permissioned ledgers can't be massively decentralized across thousands of servers because proof of work doesn't scale and they can't securely operate on a small set of replicas. In addition, eventual consistency leads to complex and incorrect operation. These problems are holding back the promise of distributed ledgers.

Current permissioned distributed ledger efforts are stuck trying to move away from latency and lack of scale of the Bitcoin blockchain but can't solve the security, scale and consistency issues involved with using a small number of replicas. Although very tightly controlled pilots are ongoing for several use cases, they haven't solved these key challenges to widespread deployment. While it is possible to get a high degree of immutability through cryptography and multi-master replication, the fundamental distinction is that the servers involved must be 100% zero knowledge. The data they hold must be strongly encrypted and the encryption keys must not be present on those servers. Otherwise, the attack surface is too large and it would be too easy for attackers and insider threats to inject fraudulent transactions.

BFDL offers an immediate solution to these obstacles: trustless, immutable, strongly consistent transactions on a massive scale with multi-master replication, and with the highest levels of security:

Ease of Replication
In BFDL Replicas can be provisioned within a single data center, across data centers in different geographical regions, or on servers across multiple cloud providers. While the choice of replica distribution may affect latency for a single ledger (due to physics), BFDL's replication is highly parallelized so high throughput can be maintained for multiple ledgers. Thirteen replicas can be assigned to each BFDL partition and BFDL is able to operate if a majority of the replicas for a partition are operating. This provides a high degree of availability and redundancy.

Protection Against Hacks
If an attacker managed to control a majority of the thirteen servers, they could not rewrite the ledgers in order to create fraudulent entries unless they also had the encryption keys for a specific ledger. This is a critical security benefit of the trustless nature of BFDL, i.e. these encryption keys are not present within BFDL. Depending on the application validation mechanism, an attacker would also have to control all of the cryptographic keys for at least a majority of participants required to generate confirming transactions. In practice, an attacker would need to simultaneously attack and control all of the servers for a ledger, rewrite all backup copies of the ledger, have all necessary encryption keys to generate all required digital signatures, defeat all user access and mandatory access controls, and delete all audit logs to escape detection.

If all of the immutability protections are deemed insufficient for a particular use case, ledgers could also be check-pointed using a variety of mechanisms, including storing checkpoints on public blockchains. The trustless nature of BFDL, derived from searchable encryption, makes a high degree of immutability feasible when operating across a small number of replicas as compared to Bitcoin.

4. CRAXEL'S BREAKTHROUGH

Until Craxel's proprietary innovation, searchable encryption was not a feasible approach for creating a trustless distributed ledger. For years, the computer science community believed that high performance, searchable encryption that doesn't leak information would be prohibitively expensive and completely impractical[1]. Techniques that are considered impractical include two-party computation, fully-homomorphic encryption and oblivious RAM. Computer scientists have believed that it would be possible for some searchable encryption schemes to be efficient but only if they were allowed to "leak" some information about the encrypted contents[2]. For instance, range and spatial queries are assumed to require the leakage of ordering information.

Craxel's methodology overcomes these limitations and creates a new paradigm for searchable encryption that maintains the ability for high performance search of encrypted data. Craxel's technology pushes the notion of encrypted queries to include not only exact match queries (such as those found in deterministic encryption schemes), but also includes range, bounding, and even spatial intersection queries. Because Craxel does not use deterministic encryption, the BFDL is not susceptible to the information leakage identified against deterministic encryption schemes.

5. MASSIVE HORIZONTAL SCALE

BFDL is a distributed, multi-master transactional database that can be partitioned to achieve massive scale. An instance of BFDL consists of one or more servers that are operating within a cluster. The partitions within BFDL are arranged as a tree structure. Each partition or shard is actually a branch of this tree structure. Tables are overlaid and their contents indexed using this single, sparse, fixed tree structure. Each BFDL cluster includes one or more root servers. The root servers maintain the partitioning, table, and user metadata. Root servers also operate as full members of the cluster, servicing their branch of the tree. Unlike distributed hash tables used by NoSQL databases, BFDL's tree structure elegantly provides an ordered index that efficiently supports range and spatial queries. Queries that span multiple branches are performed in parallel.

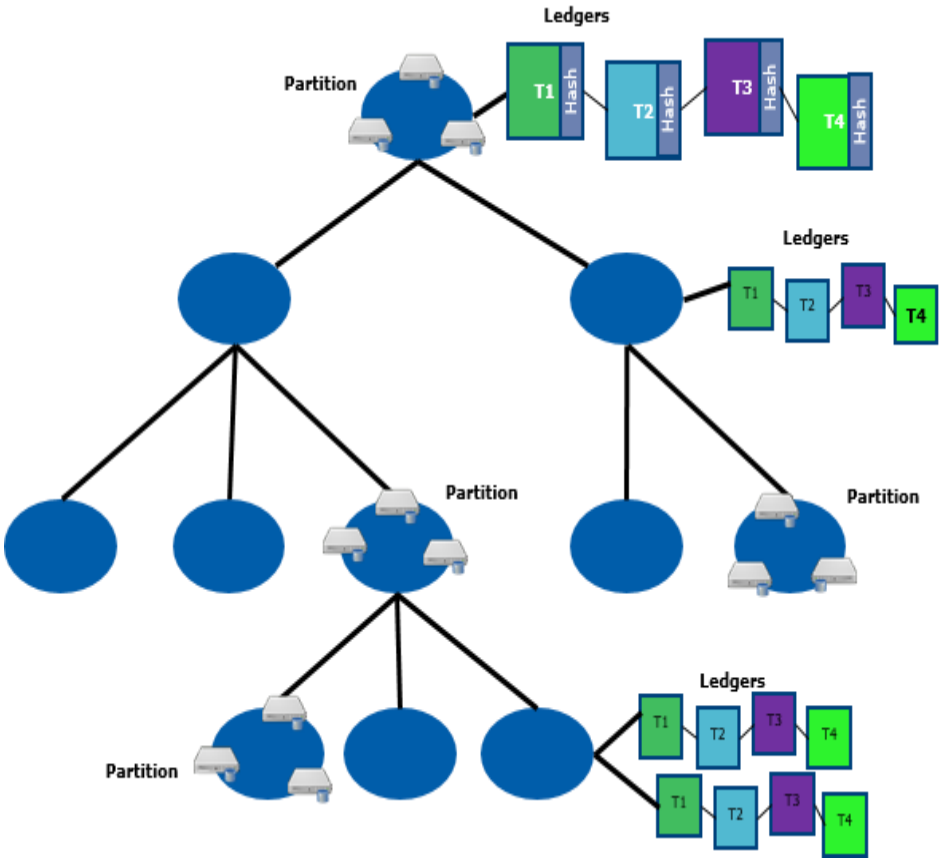

Figure 1 BFDL Partitioning

Figure 1 shows BFDL's partitioning scheme. Partitions are essentially branches of a tree structure. Each partition has a set of replicas. Those replicas use a consensus algorithm to provide serialized ordering of transactions within each tree node. BFDL provides multi-master replication of each partition. BFDL treats each partition as N

disjoint sub-partitions which allows for unprecedented scale and replication performance. BFDL uses a highly parallelized Paxos implementation for achieving consensus between the replicas of each sub-partition.

6. DEPLOYMENT

The zero knowledge nature of BFDL enables a wide variety of deployment models. BFDL can be deployed on-premises or in the cloud. In fact, BFDL could be deployed across multiple clouds, data centers, and locations. BFDL can even be deployed in a manner where different organizations operate different replicas. For instance, a partition with five replicas could be hosted by five different organizations on-premises, different clouds, or even the same cloud but in different accounts. BFDL is deployment agnostic, allowing for flexibility in how and where applications are deployed. In order to participate, a BFDL replica simply needs to be registered with the root node, have the appropriate security certificates, and be activated by the administrator.

7. APPLICATION ARCHITECTURE

BFDL provides a flexible and extensible architecture that supports many distributed ledger use cases, and which can easily be paired with existing application layer tools. Figure 2 shows the application architecture. Since BFDL has no visibility inside any transaction, validation is performed at the application layer. This allows an instance of BFDL to easily support any application use case. For instance, a payment use case and an identity management use case can co-exist within the same instance of BFDL. BFDL does not care or know about the contents of the ledgers it hosts. BFDL provides immutability and strongly consistent, atomic transactions. The application layer determines if a given transaction is valid. BFDL takes care of the complexities related to replication and ensuring serializable ACID transactions. It also provides fine-grain access control to ledgers using security labels and digitally signed user assertions.

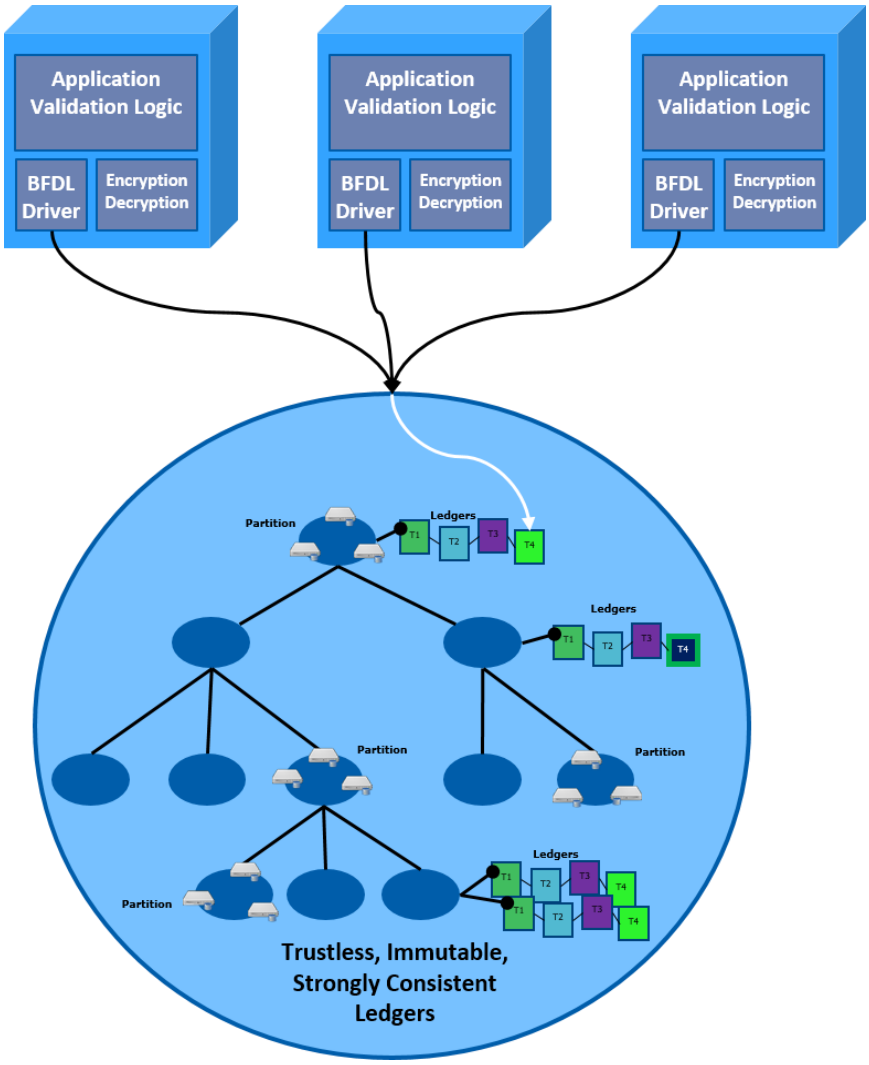

Figure 2 BFDL Application Architecture

Figure 2 shows an application with three application nodes. These nodes access BFDL through the BFDL Driver. Applications plug in an encryption provider that performs encryption and decryption before it is sent to BFDL. Applications can choose their favorite strong encryption algorithm, such as AES-256. When a transaction is presented to the encryption provider, the security label for the transaction is also included. This allows the

encryption provider to use different encryption keys for different security compartments, providing cryptographic compartmentalization. **With BFDL different ledgers can have different keys, AND different transactions within each ledger can be protected by different keys.** This creates possibilities for a ledger where some participants can only read some of the information within the ledger. This type of flexibility supports use cases across industries (e.g. health care, Fintech, supply chain, etc.), including those where regulators are participants within the ledger.

8. TRANSACTION CHAINS

BFDL's method of chaining transactions is the key to its powerful speed and performance. Instead of chaining together blocks of transactions, BFDL operates at a more granular level by chaining individual transactions together. Figure 3 shows a chain of four transactions. Each transaction contains a hash that includes the hash of the previous transactions and the contents of the transaction. The finer granularity of BFDL transaction chains improves retrieval performance dramatically over blockchain's approach of bundling hundreds to thousands of transactions into a single block. This bundling was a requirement for Bitcoin blockchain because of the cost and latency of proof of work.

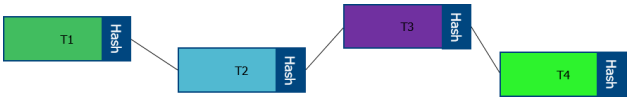

Figure 3 Transaction Chain

BFDL scales to millions of ledgers containing these transaction chains. Approaches that bundle transactions into large blocks do not support the performance required for large numbers of ledgers. Imagine the difficulty and inefficiency of accessing a ledger where transactions for multiple ledgers are bundled into blocks of thousands of transactions. To find all of the transactions for a specific ledger, every block would have to be opened and the transactions found and extracted. BFDL does not suffer from this performance overhead and in fact obviates the need for using the Merkel trees used in Bitcoin. Transactions for a single ledger are chained together and efficiently retrieved by BFDL. For many use cases, many ledgers will be required. For instance, tracking of large numbers of assets will be most efficiently handled by having a ledger for each asset. The flexibility to support both a large number of ledgers with relatively few transactions and fewer ledgers with many transactions is important to support multiple use cases and to facilitate interoperability. Existing efforts with blockchain-like distributed ledgers typically do not have this flexibility.

9. VALIDATION PROTOCOLS

BFDL approaches validation differently than blockchain type distributed ledgers. Specifically, since BFDL is completely trustless and cannot decrypt transactions, BFDL moves validation to the application layer. This provides an enormous amount of flexibility in adapting BFDL to a range of industries and use cases.

Transaction validation is incredibly important for distributed ledger systems. For example, in crypto-currencies, there must be mechanisms so that tokens such as coins can't be spent twice. For asset ledgers, participants can't arbitrarily change ownership of a given asset. In BFDL only participants with permissions for a given ledger and with possession of the encryption keys can query, create, or validate transactions. However, participants with access to a ledger and with possession of the encryption keys may not trust the other participants that also have possession of the encryption keys for that ledger. In these cases, trust among participants can be established by implementing a validation protocol at the application layer.

The BFDL Solution

BFDL provides the building block needed to easily implement these validation protocols: a serialized, strongly

consistent ordering of transactions within each ledger. This innovation is already proving to be transformative because BFDL can easily adapt to the different use cases in different industries that require unique validation models. For instance, a smart contract in the derivatives markets may require different validation than a payment transaction. The serialized, strongly consistent ordering of transactions within each ledger greatly simplifies agreement between participants that a transaction is valid. For instance, BFDL will not allow two transactions to be simultaneously committed. Every transaction to a BFDL ledger is atomic and must depend on the previously committed transaction. In the case of two simultaneous transactions, only one will succeed and the other will fail. Transactions that don't properly identify the previous transaction will fail. Like any strongly consistent, serialized transaction processing system, this approach ensures the integrity of each ledger, even with zero knowledge of the contents of each transaction. Figure 4 shows a transaction venue proposing a transaction and counterparties validation the transaction to ensure it meets the rules of the venue.

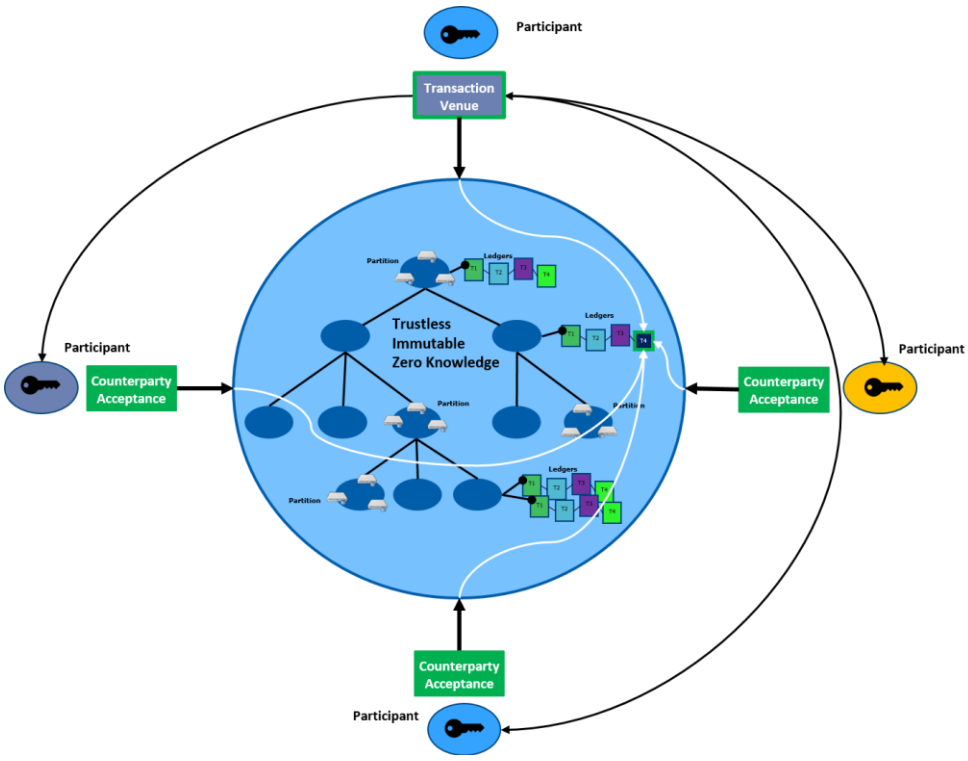

Figure 4 Application Layer Validation Protocol

Use of Simple Agreement and Validation Models

Given the serialized, strongly consistent building block provided by BFDL, simple agreement and validation models can be used, such as confirming transactions. For instance, a proposed transaction could be committed to the ledger and for it to be valid, it could require confirming transactions from a quorum of participants. In this model, fraudulent transactions would require hacking multiple BFDL accounts, each account requiring permission to access the target ledger, and possession of the private encryption keys of a majority of participants. These keys are not present in BFDL so they would have to be acquired from each participant. Also, since transaction chains are immutable, the fraudulent participant would not be able to modify existing transactions.

10. CONCLUSION

The disruptive nature of blockchain technology is easy to conceptualize in the world of finance, but the security, scalability and performance of the BFDL could be transformative across multiple industries including supply

chain, health care, and digital rights management. The BFDL is designed to reduce and even eliminate the friction and costs of intermediaries, and also lends itself well to creating a new framework for disjointed record keeping and lack of transparency. A BFDL approach to managing records will allow stakeholders to more effectively share information, enabling processes that are ultimately more efficient, disintermediated, and secure. By moving validation into the application layer, BFDL easily integrates with existing applications and can be quickly and easily bonded with smart contracts and other application layer tools.