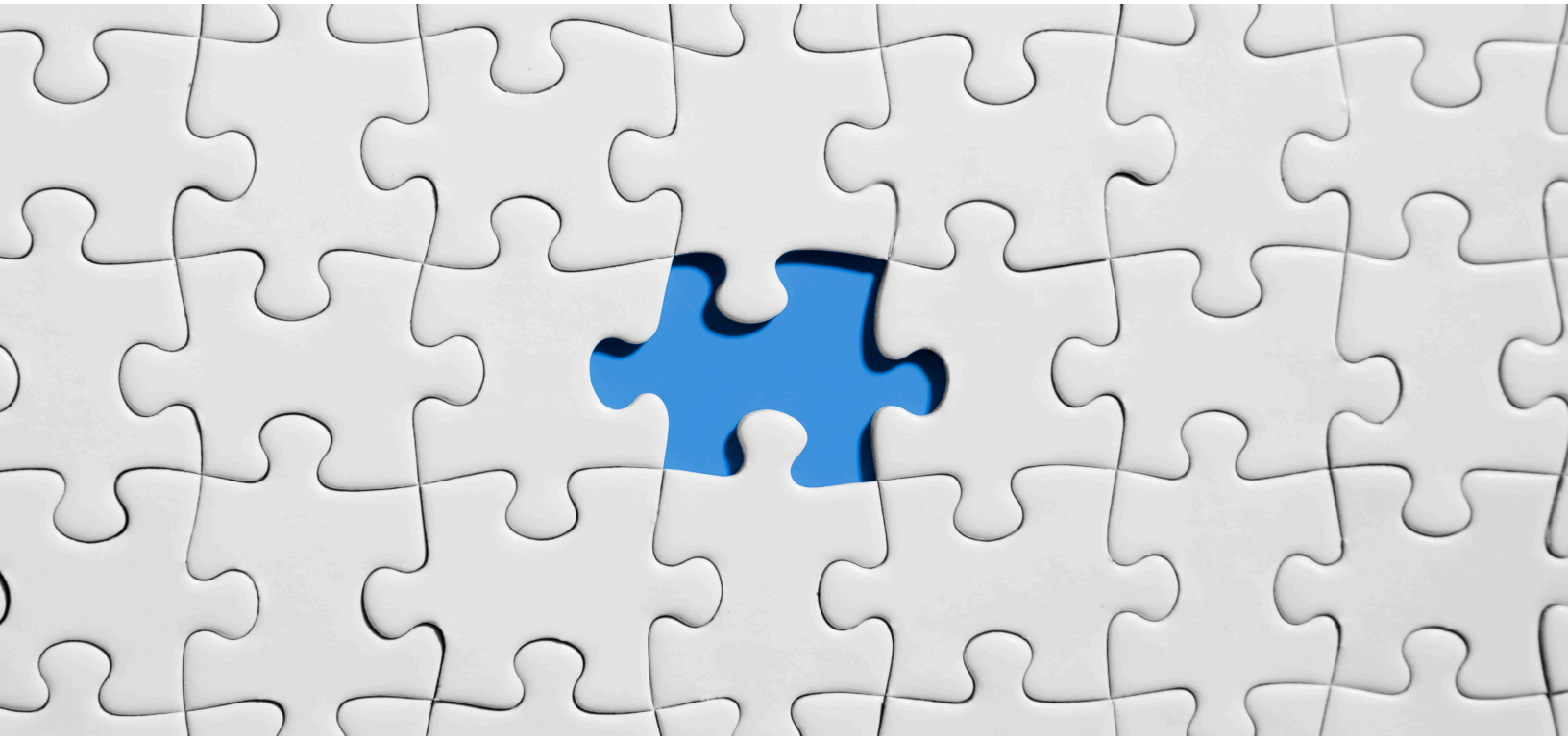
DELL EMC

# ROCK AROUND THE BLOCKCHAIN WITH DELL TECHNOLOGIES




## Steve Todd

Dell Technologies Fellow

Dell EMC

steve.todd@dell.com


DELL EMC PROVEN PROFESSIONAL

Knowledge Sharing Article

## Table of Contents

# 1 Introduction

Nearly ten years ago, Bitcoin developers came up with a new type of distributed database that spanned the globe. We know this as the Bitcoin blockchain. In subsequent years, thousands of copies of the Bitcoin software have spread across all six continents, continually contributing to and growing this database. Figure 1 highlights the remarkable geographical mesh of global contributors to Bitcoin's blockchain.

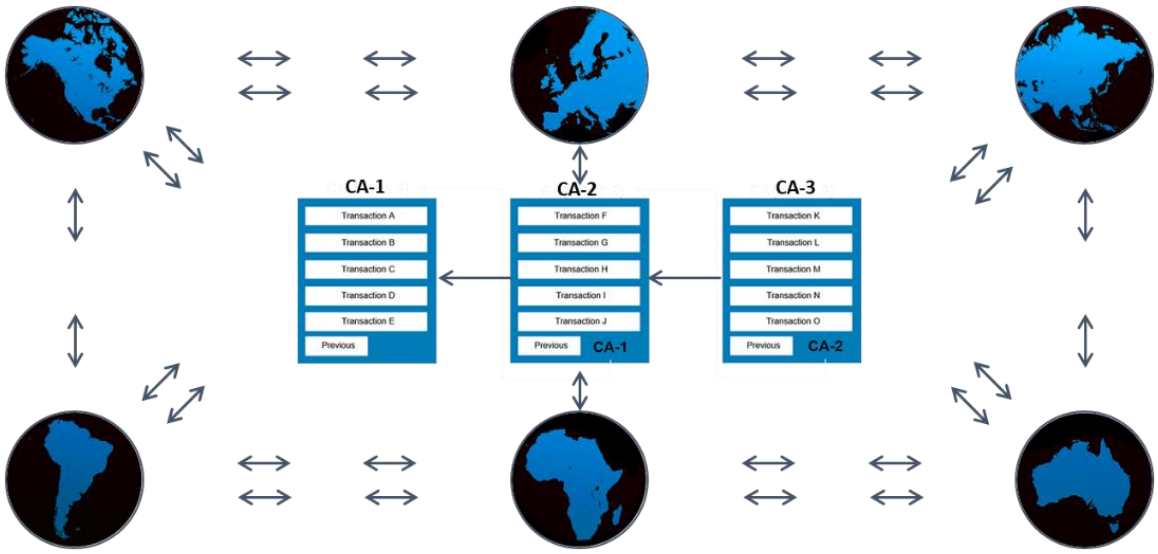



**Figure 1 - Global Scale of the Bitcoin Blockchain**

Figure 1 also captures a vital feature of the blockchain. In the center, there are groups of transactions (A-E, F-J, and K-O), stored in three separate "blocks." These blocks all have unique content addresses (CAs), which represent cryptographic hashes of the grouped transactions. Any one entity does not decide to "publish" new transactions on its own: decisions are consensus-based.

The Bitcoin blockchain is often called a "distributed ledger." One reason for the name is that it keeps track of financial transactions between wallets. However, the intention of this paper is not to explain what a blockchain is, but how it integrates into an enterprise environment. There are three areas of focus:

1. Understand why companies around the world are considering distributed ledgers.
2. Describe why implementing this type of database is so challenging in an enterprise context.
3. Explore how the Dell Technologies portfolio can be leveraged to build a system exhibiting enterprise-class ledger capabilities.

Placing transactions into a ledger would appear to be straightforward. Figure 2 highlights a simple picture of an application inserting an entry into a shared ledger.

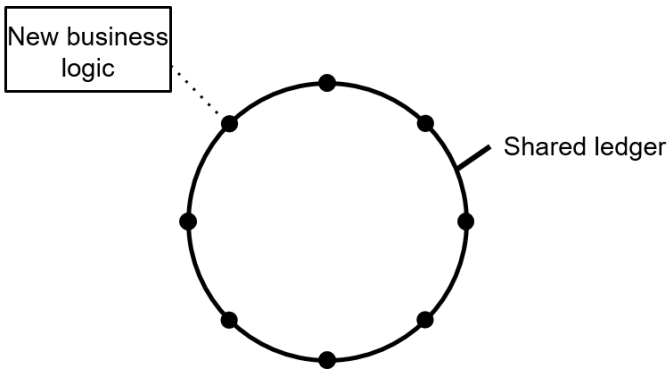



**Figure 2 - Inserting a Transaction into a Shared Ledger**

The nodes in the shared ledger could represent different business units within the same company. They could also symbolize proxy member companies in a consortium, cooperative vendors in a supply chain, or untrusted third parties in an open marketplace (note that ledger transactions may or may not include currency transfer).

Throughout the industry, enterprise companies are generating a hypothesis. The following sentence is a good summary of their goals for blockchain:

> *"The creation of new, ledger-based applications can significantly increase our revenues, reduce our costs, and reduce our risk."*

The adoption of shared ledger technology by enterprise companies has come on the heels of many years of innovation and startup activity. The global banking crisis in 2008 is seen by many as the spark that started the blockchain revolution. In 2009 the first forms of shared-ledger business logic appeared. These applications inserted financial transactions into the Bitcoin blockchain. Three important benefits include:

- Global accessibility (the nodes in the shared ledger span geographies).
- Security (transaction creation and validation using cryptography).
- Decentralization (no one government or business controls the ledger).

Lately, the number of new applications contributing to Bitcoin's blockchain has grown considerably. Figure 3 highlights new companies (e.g. Stampery, Blockstack, Factom) that deployed algorithms leveraging the Bitcoin blockchain.

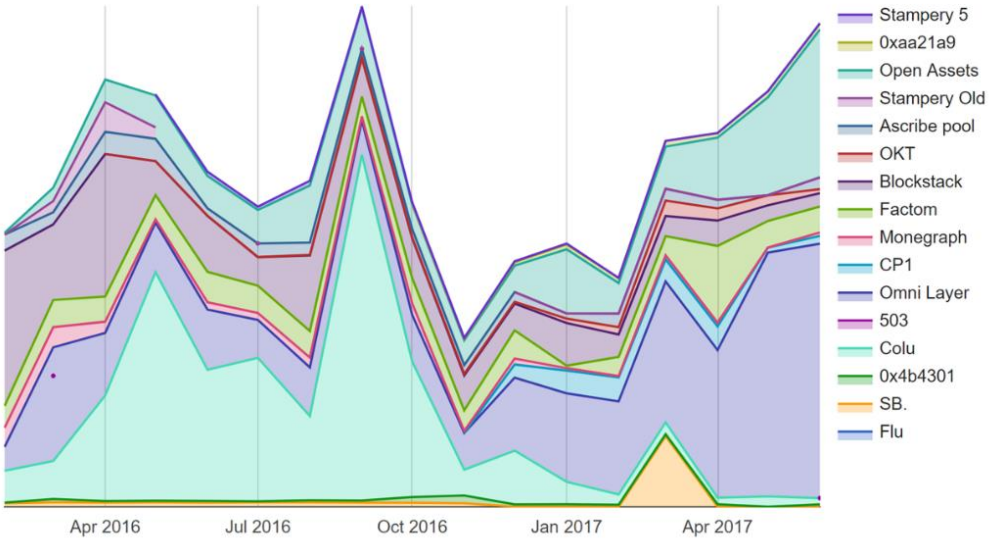



**Figure 3 - New Applications Leveraging Bitcoin's Blockchain (Source: opreturn.org)**

Recently some different blockchain implementations (Ethereum, Hyperledger, Corda, etc.) have surfaced. As a result, the number of startups writing new blockchain applications has also increased dramatically. Figure 4 highlights a 2017 map of global blockchain startups (released by Frost & Sullivan and Outlier).

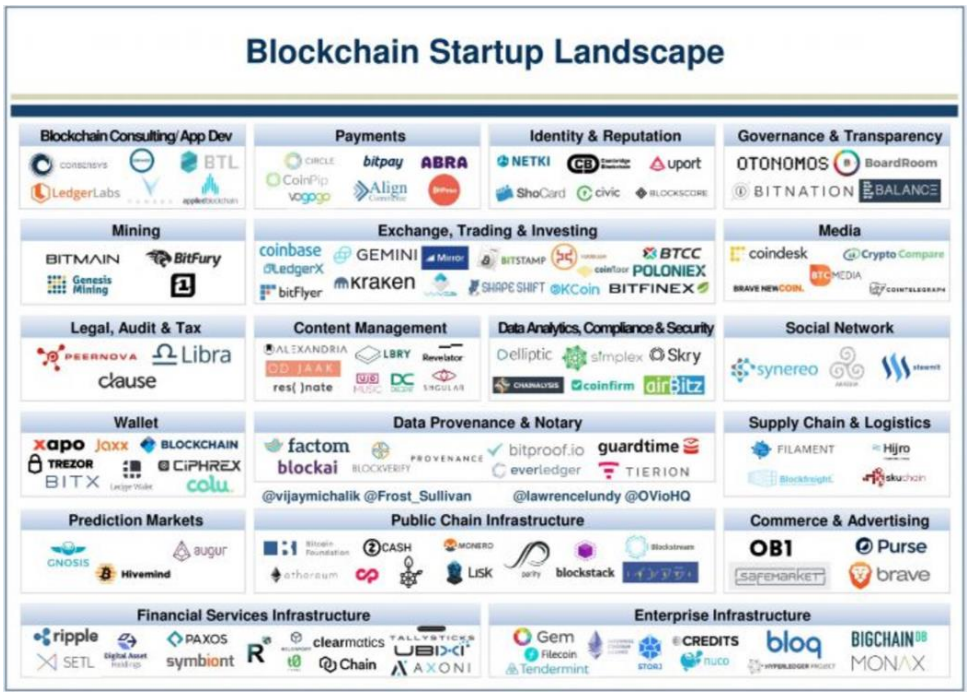



**Figure 4 - Blockchain Startup Activity (Source: Frost & Sullivan, Outlier Ventures)**

Many of these blockchain applications are not “enterprise class” regarding their performance, availability, scalability, and reliability. This capability shortfall has led to a rise in the number of enterprise-grade blockchain initiatives:

- The Hyperledger Community
- The Enterprise Ethereum Alliance
- The Corda Community
- Chain Enterprise

A large number of IT vendors are working alongside these enterprise blockchain initiatives and are advertising varying levels of success.

- IBM began building blockchain thought leadership as early as 2015 when they announced their participation as a founding member of the Hyperledger initiative. IBM contributed their blockchain code. IBM's open-source contribution was labeled "production ready" in 2017 as part of Hyperledger's Fabric 1.0 release. IBM has aggressively launched a blockchain-as-a-service initiative (based on Hyperledger) as part of their Bluemix catalog.
- Microsoft's Azure platform is also aggressively marketing blockchain-as-a-service (Deploy and configure a blockchain network in minutes). While IBM has focused heavily on Hyperledger ledger logic, Microsoft is focusing on other blockchains, including Ethereum and Chain.
- HPE announced a partnership with R3 in 2017 and an integration with R3's Corda distributed ledger. As part of the announcement, HPE mentioned that they had built a Corda proof-of-concept on HPE's Integrity NonStop Platform. HPE joined the Ethereum Enterprise Alliance in October 2017.
- In 2016 RedHat announced an OpenShift blockchain-as-a-service initiative, and in 2017 RedHat partnered with Ethereum application development platform partner, BlockApps.
- In 2016 Hitachi announced their intention to study blockchain in their new Financial Innovation lab in Santa Clara and in March 2017 announced their partnership with Tech Bureau on the PointFinity rewards point implementation.
- In January 2017 Cisco announced an IoT/blockchain partnership with a variety of companies, including Bosch. Cisco also became a premier member of Hyperledger and a member of the Ethereum Enterprise Alliance in July of 2017.
- Fujitsu has developed blockchain-based software as part of a secure data exchange network (announced in June 2017). Their research lab announced the accelerated performance of Hyperledger in July 2017.
- Oracle joined Hyperledger in August of 2017.

Dell Technologies is the broadest information technology company in the world. So, what is Dell's blockchain strategy?

In 2016, technologists from across Dell Technologies (Dell Client, Dell EMC, VMware, Pivotal, RSA, SecureWorks, Virtustream, and Boomi) formed a Blockchain Interest Group (BIG). The steering committee for BIG (Blocksteer) serves as a clearinghouse for assisting customers with information about Dell's blockchain capabilities.

Less than one year later, Michael Dell pointed towards some of the output from the BIG community.


The BIG community focuses specifically on industry development and deployment of enterprise-grade blockchain applications. There have been four central questions addressed by the team:

1. What business problems are these applications trying to solve?
2. What are the common obstacles to running blockchain applications in mission-critical environments?
3. How has VMware's research addressed these obstacles?
4. How can Dell Technologies' portfolio integrate blockchain applications into existing IT ecosystems?

This paper focuses on answering these four questions.

# 2 Why Are Companies Writing Blockchain Applications?

To better understand the application landscape, the BIG community launched an internal classification initiative to create a sample of customer (and industry) blockchain use cases.

BIG member Keith Regalbuti (Dell EMC) describes the collection strategy:

> *"Our Dell Technologies use case collection strategy uses the why, how, and what methodology. For each scenario, we document why the customer is interested in a ledger, how Dell Technologies can help them build it, and what it will mean for their business. The resulting catalog enables us to analyze what the market is asking for and how we can build it for them."*

The collection process resulted in the emergence of four distinct use cases.

- Management/Transfer of Data Assets
- Broadcast of Data
- Credential Verification
- Supply Chain Transparency

The following sections address each of these.

## *2.1 Management/Transfer of Data Assets*

As more and more assets are becoming digital (health records, deeds, etc.), the idea of using a shared ledger to manage those assets becomes more appealing. New ledger-based business logic can streamline one or more of the following tasks:

a. Ownership transfer of a digital asset
b. Permission to view a digital asset
c. Location tracking of a digital asset
d. Changing the attributes of a digital asset

One of the better examples of this use case is correspondent banking, which operates via correspondent accounts[1]:

> *"A correspondent account is an account (often called a nostro or vostro account) established by a banking institution to receive deposits from, make payments on behalf of, or handle other financial transactions for another financial institution. Correspondent accounts are established through bilateral agreements between the two banks."*

While correspondent banking allows financial firms to offer international banking services to their clients, it comes at a cost. Paula Roels, the Head of Market Infrastructure and Industry Initiatives at Deutsche Bank, describes one common problem[2]:

> *"Take, for example, cross-border payments, which currently can take up to three to five days to process end to end."*

Roels and many others took note of the Bitcoin blockchain's ability to perform cross-border payments in minutes as opposed to days. In the same article, Roels goes on to discuss many other benefits that the technology could bring to cross-border payments and correspondent banking:

> *"Proponents highlight that implementing the technology would lead to increased transparency, reduction in errors and greater transaction automation, leading to a decrease in cost and, ultimately, fees for the end user."*

In his article about the disruptive effects that blockchain can have on the financial industry, Daniel Jäger, the Head of Financial Close at Inplenion International AG, offers an illustration (Figure 5 below) of how the technology can cut through the layers of handshaking that plagues today's international financial transactions[3].

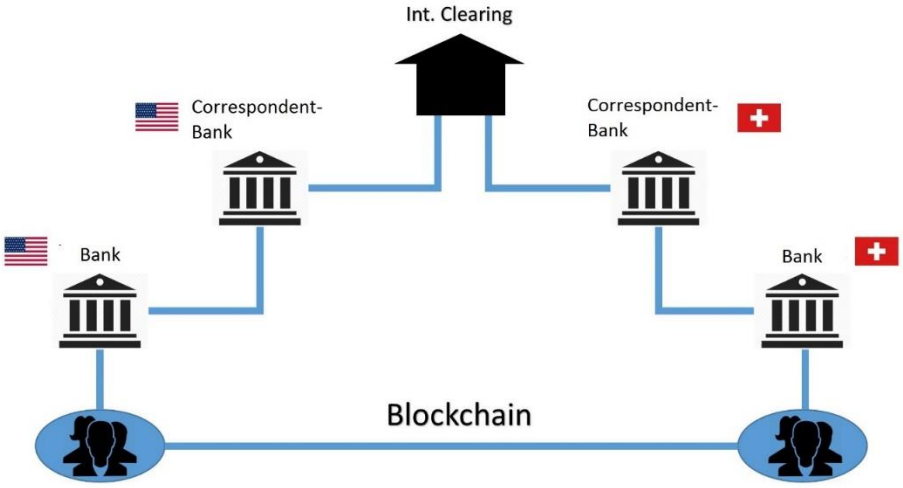



**Figure 5 - Blockchain Optimization of International Transfers**

The banks, correspondent-banks, and international clearing organization all typically have their own IT silos. These silos slow down transactions and increase the chance of human error. The blockchain depicted at the bottom of Figure 5 breaks down these silos by offering a common, shared ledger into which organizations can communicate more rapidly.

Correspondent banking is just one example of the management or transfer of data assets. The BIG community noted many other use cases as well, including new mobile money payment platforms and self-sovereignty for digital assets.

In all cases, the new applications considered for this use case aim to enable an increase in revenue, a decrease in cost, and a reduction in risk.

## *2.2 Broadcast of data*

The use cases also highlight that new applications are being written to broadcast data assets to a shared ledger. The ledgers often address a broad (but sometimes restricted) audience. These new programs focus on the broadcast of data to public, private, or consortia entities. Smart contract ledger APIs often control permission to access, view, and update these assets.

Some use cases broadly broadcast data to the world, forever timestamping their creation via blockchain's transaction timestamping mechanism.

One of the more interesting use cases of using the blockchain to broadcast digital data broadly is known as "ASCII Bernanke." A creative bitcoin user created an ASCII image[4] of Chairman of the Federal Reserve Ben Bernanke and stored it in the Bitcoin blockchain.

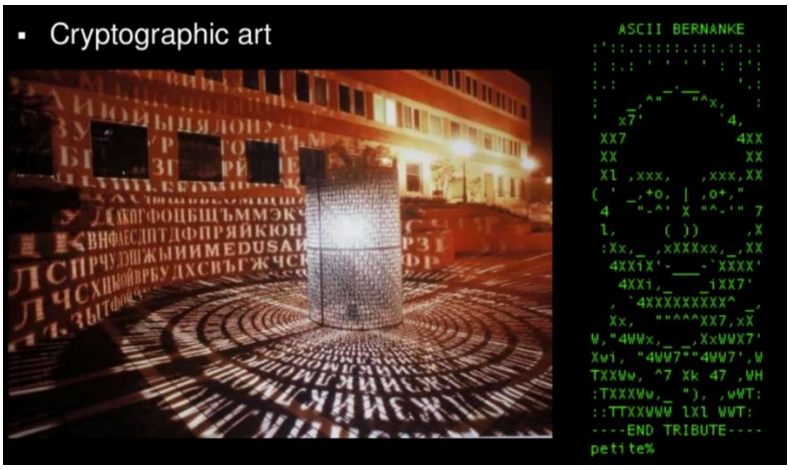



**Figure 6 - ASCII Bernanke: Broadcasting Data on the Bitcoin Blockchain**

However humorous the example, the ASCII Bernanke highlights the storage of digital content onto an immutable, cryptographically secure ledger, potentially establishing authorship and lineage.

One industry that took note of this capability was the modern film industry.

Author Paul Forrest of MBN describes the navigation (via a ledger) of the complexities arising from the emergence of digital content and online distribution platforms.[5]

> "*Traditional distributor and exhibitor networks have expanded to include international partners who control local distribution. Online distribution platforms have brought technology companies such as Netflix, Apple and Amazon into the mix. In such environments a Blockchain can serve as a decentralised and scalable solution to manage the increasing complexity of contemporary global networks and digital distribution models*".

Forrest goes on to describe how a ledger can address the business complexities of current film distribution models.

- Enforcement of distribution and release agreements
- Automated disbursement of revenue when collections are received and recorded
- Establishing and protecting copyright
- Registering ownership and provenance of content
- Combating digital piracy
- Enabling and encouraging a release channel for smaller players

Government agencies are also considering blockchain architectures for data exchange. These designs allow data consumers to broadcast content onto a blockchain for the use of data consumers. In a recent report[6], Australia's Commonwealth Scientific and Industrial Research Organisation (CISRO) proposed an architecture for data broadcast via the blockchain. Figure 7 outlines their approach.

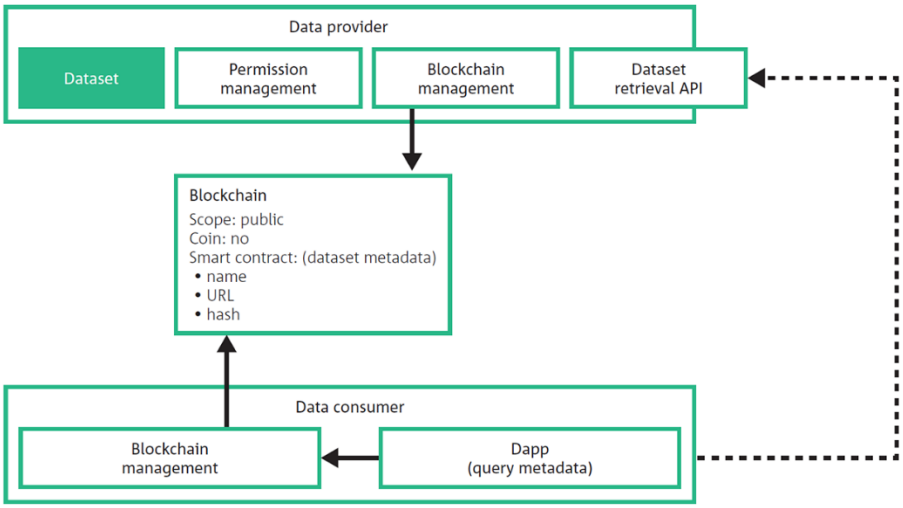



**Figure 7 - Using a Blockchain for Data Broadcast**

The exchange of data has always been tricky to implement from a security and provenance standpoint. Blockchain's cryptographic characteristics can assist in this regard.

## *2.3 Credential Verification*

As IT architectures continue to sprawl geographically, the centralized forms of credential management (e.g. Active Directory, LDAP) are no longer scalable enough and do not offer the authentication mechanisms required in sensor-based configurations. Startups such as Sensify use the technologies listed in Figure 8 to highlight a large number of communication protocols in those environments (OT or Operational Technology) and the difficulty in bridging to traditional enterprise deployments (IT or Information Technology).

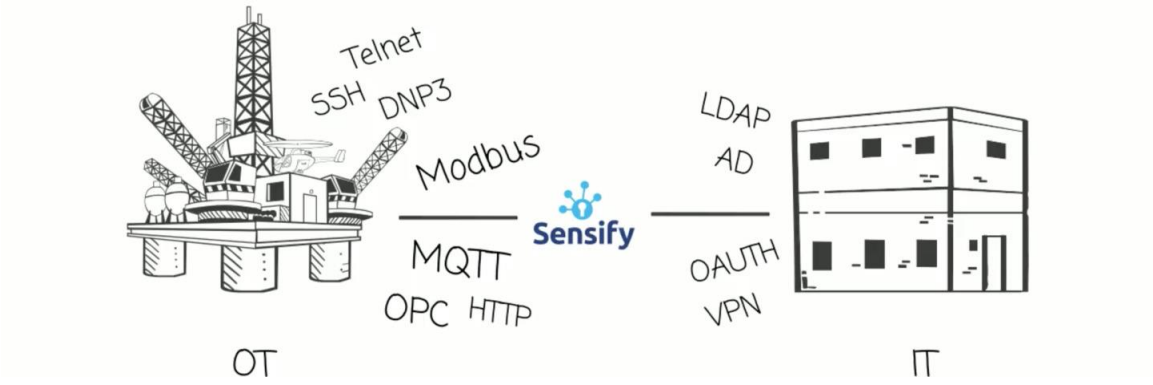



**Figure 8 - Challenges of Decentralized Identity Management**

One of the main problems with credential management in large-scale environments is the round-trip latency required to authenticate a "person" or a "thing" connected to the industrial internet.

The idea of decentralizing identities by securely storing and distributing them into a shared ledger can reduce that latency down to the length of time it takes to do a ledger lookup.

In 2016, members from Princeton and Blockstack Labs published a paper[7] that evaluated the validity of storing credential information onto a shared ledger. The Decentralized Identity Foundation (DIF[8]) was formed shortly after that to formalize standards for decentralized identity management.

Once identities are established on a blockchain, the ability to record and verify the credentials introduces intriguing possibilities for sectors such as education, healthcare, and the enterprise (e.g. Human Resources).

The Illinois Blockchain Initiative has begun to build a blockchain to assist in the medical credentialing process. Illinois Department of Financial and Professional Regulation (IDFPR) Secretary Bryan Schneider describes his hope that the use of a ledger can help with the complexities of their existing system[9]:

> "*In the short-term we anticipate this pilot will show how distributed ledger technology can help reduce the complexity of inter-state licensing processes in Illinois."*

In 2016 MIT's Media Lab and Learning Machine created Blockcerts[10]:

> "*Blockcerts is an open standard for creating, issuing, viewing, and verifying blockchain-based certificates. These digital records are registered on a blockchain, cryptographically signed, tamper-proof, and shareable. The goal is to enable a wave of innovation that gives individuals the capacity to possess and share their own official records.* "

MIT notes that in the job-seeking process, validating credentials (from a university, for example) is a painstaking and error-prone process[11].

> *"Job seekers have to request official transcripts from their alma maters (and typically pay a small fee), and employers still need to call the university if they want to be sure that a transcript wasn't faked. It's a slow and complicated process, which is one reason why degree fraud is a real issue. (A few years back, even our very own MIT Admissions office realized that its Dean didn't actually have the undergraduate degree that she had listed in her application). Making certificates transferable and more easily verifiable is one advantage of digital systems."*

Once digitized credentials land on a (nearby) trusted ledger, manual authentication processes can proceed much more smoothly and rapidly.

## *2.4 Supply Chain Transparency*

The movement of parts and goods through a supply chain and the ability to track those assets is a natural fit for blockchain's timestamped, append-only log.

One of the more illustrative examples of using blockchain for the supply chain is Oliver Wyman's example[12] of the supply chain for dry aged beef.

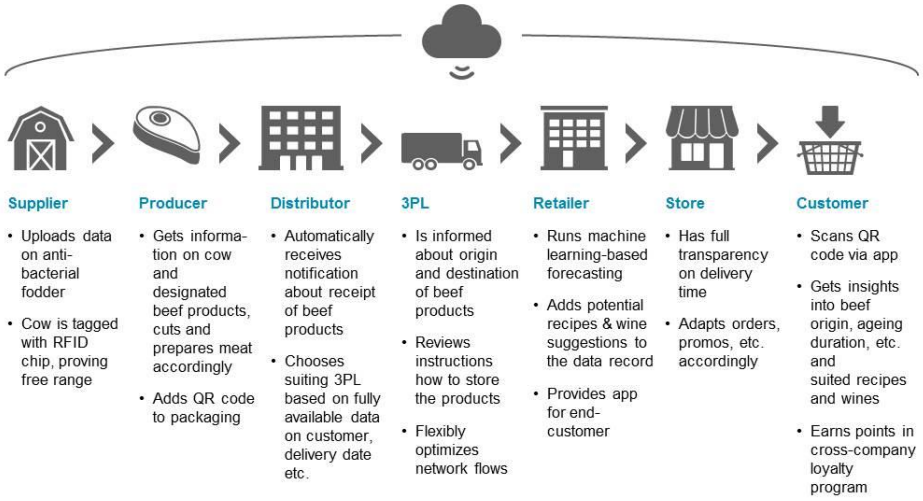



**Figure 9 - Cow-to-Consumer Blockchain Tracking**

The article goes on to highlight the benefits of the approach depicted in Figure 9:

> *"With a simple QR-code scan on their smartphone, customers could validate every step the beef has taken through the supply chain, and match that journey against their expectations. Any kind of historical as well as real-time data on the beef product, be it related to the origin (such as feed or breeding), timing (such as aging duration, time in transport, best before date), location (of the farm and of the beef throughout the supply chain) or additional information (such as recipes and wine suggestions) is continuously available from the blockchain database in a single, consistent version ("one source of truth")."*

Deloitte highlights[13] the benefits of deploying blockchain applications for each part of the supply chain.

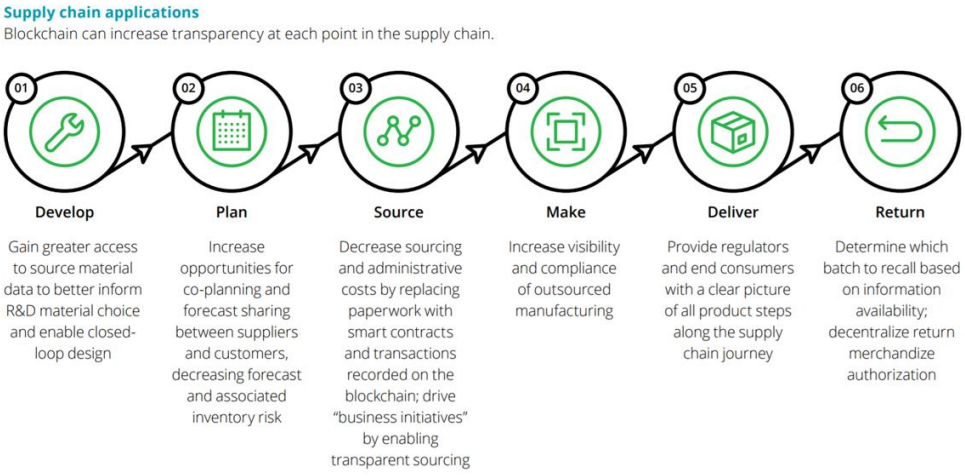



**Figure 10 - Deloitte Benefits of Blockchain Supply Chain Applications**

The use cases collected by the BIG Community all promise significant business advantages. As a result, the industry is being forced to confront the dizzying array of blockchain buzzwords, startups, emerging standards, and vendors.

For each use case in the catalog, a pattern of similar problems is emerging.

The BIG community has documented these problems below.

# 3 Obstacles to Enterprise Blockchains

As enterprise companies began to observe the business benefits of blockchains like Bitcoin and Ethereum, they began to create new applications that inserted transactions into a shared ledger. Over time, they began to observe significant shortcomings. The existing blockchain ecosystem fell far short of satisfying fundamental enterprise application requirements. The following weaknesses surfaced:

- Performance
- Time-to-Finality
- Data Consistency
- Multi-Chain or Multi-Ledger
- Secure and Portable Smart Contracts
- Smart Contract Instrumentation and Auditability
- Search Capabilities

These shortcomings are as follows.

## *3.1 Performance*

Most enterprise applications expect their data stores to process tens (if not hundreds) of thousands of transactions per second.

A blockchain transaction, by its very nature, must wait for an underlying network of nodes to come to a consensus before the transaction is acknowledged. Bitcoin, for example, commonly processes between 3-8 operations per second as a result of the chattiness of its protocol and dependency on mining (Blockchain.info provides a live view of Bitcoin transactions per second[14]).

The Initiative for Cryptocurrencies & Contracts (IC3) addressed the performance gap between blockchain and mainstream transaction processing in their 2016 paper On Scaling Decentralized Blockchains[15]:

> *"Today's representative blockchain such as Bitcoin takes 10 min or longer to confirm transactions, achieves 7 transactions/sec maximum throughput. In comparison, a mainstream payment processor such as Visa credit card confirms a transaction within seconds, and processes 2000 transactions/sec on average, with a peak rate of 56,000 transactions/sec [10]. Clearly, a large gap exists between where Bitcoin is today, and the scalability of a mainstream payment processor."*

Private enterprise blockchains are not as slow as Bitcoin. But how much faster are they? And how can performance for varied workloads on differing blockchain platforms be fairly evaluated?

The BLOCKBENCH project at the Computing Department of the University of Singapore has created a blockchain performance testing framework. In their paper, BLOCKBENCH: A Framework for Analyzing Private Blockchains, the authors make the following statement[16]:

> *"Multiple platforms for private blockchains are being actively developed and fine tuned. However, there is a clear lack of a systematic framework with which different systems can be analyzed and compared against each other."*

The BLOCKBENCH paper analyzed the performance of three popular blockchains (Ethereum, Hyperledger, and Parity) and recorded the number of transactions per second (y-axis) served while varying the number of issued requests per second (x-axis). The tests were performed with eight clients and eight servers[17].

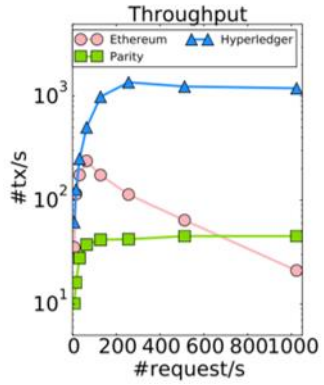



**Figure 11 - BLOCKBENCH Measurement of Blockchain Transaction Throughput**

For this testing scenario, Hyperledger consistently displays the highest throughput capabilities, but all implementations fall far short of standard enterprise performance requirements.

To continue closing this gap, Hyperledger has announced the formation of a Performance and Scaling Working Group[18]:

> *"The mission of the PSWG is to discuss, research, and identify key use cases and metrics that relate to the performance and scalability of a blockchain and blockchain related technology."*

The early meetings of the working group are indeed focusing on formulas or expressing latency regarding consensus delay[19].

While the performance problem is recognized, the performance deficiencies are still far too significant for consideration by many mission-critical applications.

## 3.2 Time-to-Finality

Enterprise applications expect fast response times on every transaction. It is not uncommon for blockchain applications to experience transaction response time on the order of minutes (recall the IC3 statement that Bitcoin transaction confirmation can take 10 minutes).

The Cambridge Centre for Alternative Finance studied enterprise and public sector use of blockchain and shared enterprise expectations for scalability in their report Global Blockchain Benchmarking Study[20]. Their findings are consistent with the assumption that adding more nodes solves latency problems.

> *"System needs to be able to scale immediately as more nodes join the network (latency issues), more transactions are performed (increasing processing power and memory usage required), and the transaction history grows (increasing storage requirements).*
>
> *Scalability can be more challenging to define, but in general refers to the ability of the system to sustain performance while growing and expanding (e.g. increase of the number of nodes and/or the number of concurrent workloads). This also includes increasing storage requirements and potentially higher latency (generally measured as the response time per transaction) as the network grows."*

BLOCKBENCH explored issues of scale in private blockchains by running their framework against an incrementally increasing amount of nodes. Not surprisingly, BLOCKBENCH discovered that adding more nodes did not improve latency in some blockchains (Parity) and significantly degraded performance in others (Hyperledger and Ethereum).

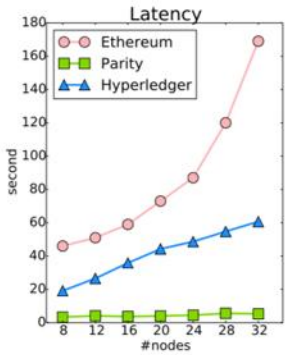



**Figure 12 - BLOCKBENCH Measurements of Latency at Scale**

The researchers for these use cases once again point to the underlying network (and the Practical Byzantine Fault Tolerant protocol, or PBFT) as a primary culprit for slow time-to-finality at scale[21].

> *"In fact, we also observe that as time passes, client requests took longer to return, suggesting that the servers were over saturated in processing network messages. We note, however, that the original PBFT protocol guarantees both*

> *liveness and safety, thus Hyperledger's failure to scale beyond 16 servers is due to the implementation of the protocol. In fact, in the latest codebase (which was updated after we have finished our benchmark), the PBFT component was replaced by another implementation. We plan to evaluate this new version in the future work."*

The underlying network consensus algorithm (PBFT for Hyperledger) is critical for time-to-finality.

The results provided by current implementations show improvement, but a large gap remains. This shortfall is an obstacle to enterprise deployment of blockchain applications.

## *3.3 Data Consistency*

The original Bitcoin paper describes a problem in which different nodes broadcast a different view on the most recent state of the ledger[22].

> *"If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer."*

This could result in two different applications (located on different nodes) "reading" different results from the local copy of their ledger. For some applications this is unacceptable. The Global Blockchain Benchmarking Study refers to this problem as Settlement Finality[23]:

> *"A legal concept that is mandatory for enterprise applications – once confirmed, transactions cannot be reversed (at least from a legal perspective). This does not apply to public blockchains where settlement finality is only probabilistic: an alternative, longer chain could replace the current chain and reverse all transactions that were previously confirmed."*

While all of the use cases described in Section 2 can have mission-critical dependencies on time-to-finality, perhaps the most significant industry concerned with finality is the financial industry.

Infosys related the criticality of settlement finality in their white paper Blockchain Adoption in Financial Services[24]:

> *"Without guaranteed settlement finality, there are risks of insolvency of one participant undoing the transactions that are otherwise deemed settled, creating myriad liquidity and credit issues for other participants."*

Developers of blockchain applications currently have no standard way of knowing the data consistency characteristics of their underlying blockchain. For use cases like the financial services industry, this situation is untenable and serves as yet another obstacle to overcome.

## *3.4 Multi-Chain*

The enterprise is trending towards a multi-blockchain world.

In September of 2017, Coindesk released their “State of Blockchain” quarterly report. The report highlights the number of blockchain implementations currently under evaluation by enterprise companies[25].

> *“Major firms like HP, Thomson Reuters, and the Moscow Stock Exchange have now launched testing and worked through integrations with Corda, Hyperledger Fabric, Sawtooth Lake, and Iroha, and permissioned version of Ethereum*.”

In the same report, Coindesk also highlighted Ripple’s demo of a financial transaction that crossed seven different ledgers.

> *“Ripple grows banking global network… sending a single transaction across 7 ledgers including public and private blockchains, a centralized ledger and a traditional channel*.”

One technology that deals with financial transactions across multiple ledgers is the Interledger Protocol (ILP)[26].

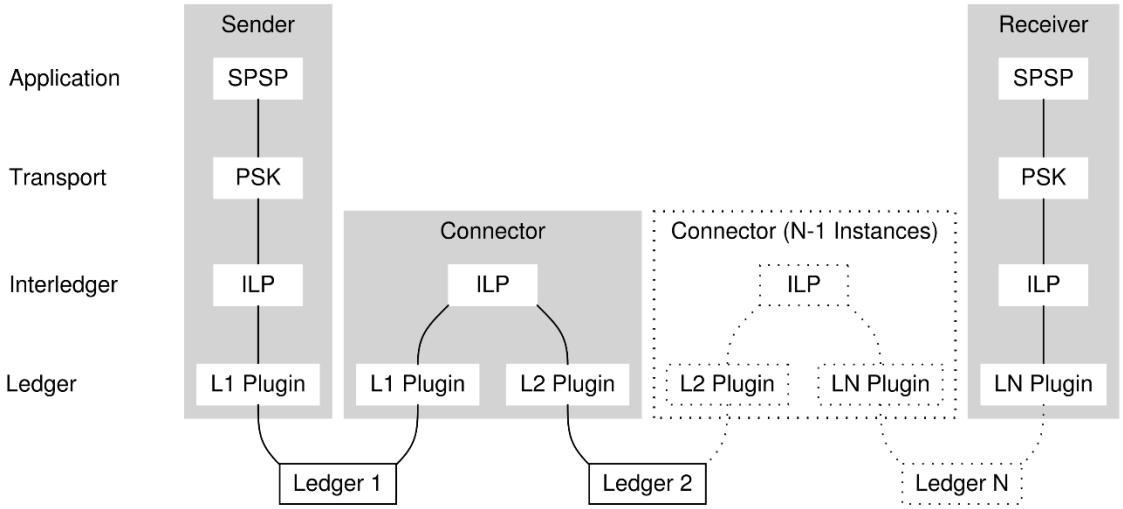



**Figure 13 - Architecture for the Interledger Protocol**

While Figure13 highlights a plug-in approach for transmitting a financial transaction across multiple ledgers, it does not address how an application can perform business transactions across all of the use cases mentioned above (e.g. data transfer, credential, supply chain, etc.)

One driver of multi-chain is solving for scalability. Many public blockchains address scaling issues by sharding a blockchain to parallelize transaction processing[27].

> *“A smarter approach is the idea of blockchain sharding, where we split the entire state of the network into a bunch of partitions called shards that contain their own*

*independent piece of state and transaction history. In this system, certain nodes would process transactions only for certain shards, allowing the throughput of transactions processed in total across all shards to be much higher than having a single shard do all the work as the mainchain does now."*

Sharding creates many "little blockchains" and should all vendors go this route then there will be ever-increasing numbers of "incompatible little blockchains." There are currently no standards for transferring information between chains. Some chains may encrypt, some do not. Some may use REST, JSON, or RPC.

To make matters worse, the smart contract interface to different blockchains varies wildly as well, and this variety will be a frequent target for malicious activity.

## *3.5 Secure and Portable Smart Contracts*

If an enterprise company commits to coding against the Ethereum blockchain (permissioned or public), they may invest their time writing smart contracts in the Solidity language. This investment cannot currently be ported directly to all other blockchain implementations (such as Go for Hyperledger and Java/Cotlin for Corda).

Usage of smart contracts is exploding. For example, smart contract security company Quantstamp predicts an explosion in the number of Ethereum smart contracts[28].

*"Between June 2017 and October 2017, the number of smart contracts grew from 500K[29] to 2M[30]. Within a year, we expect there to be 10M smart contracts."*

This represents two problems for the enterprise:

1. Smart Contract portability. Investments made in one language (e.g. Ethereum Solidity) cannot currently be ported directly into other blockchain implementations (e.g. Hyperledger and Corda).
2. Smart Contract vulnerabilities. Poorly-written smart contracts bring significant risk to the enterprise.

This second point is best highlighted by the DAO hack of 2016. In this example, the smart contract operated in a way that allowed re-entrant function calls to withdraw funds continually. Figure 14 highlights this code.

```
function splitDAO(
    uint _proposalID,
    address _newCurator
) noEther onlyTokenholders returns (bool _success) {
    ...
    withdrawRewardFor(msg.sender); // be nice, and get his rewards
    totalSupply -= balances[msg.sender];
    balances[msg.sender] = 0;
    paidOut[msg.sender] = 0;
    return true;
}
```

**Figure 14 - Poorly-written Smart Contract Code**

The smart contract developer called the “withdrawRewardFor()” function before subtracting the amount from the balance. This allowed a smart contract to continually re-enter this code and execute transfers without actually decrementing the “totalSupply” variable.

This re-entrant code is only one example of a large number of smart contract vulnerabilities. The academic paper Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab[31] highlights more weaknesses. The list below enumerates these problems (the parentheses point to the corresponding section within the paper).

- Errors in state machine design (4.1)
- Cleartext parameters (4.2)
- Misaligned incentives (4.3)
- Implementation-specific errors (4.4)

Smart contract execution environments (like Ethereum) can try and put safeguards in place to prevent these vulnerabilities. However, any effort to port one of these smart contracts to run in a different environment does not necessarily come with a framework that flags the same security violations.

Developers at companies like VMware and Dell EMC use a Secure Development Lifecycle (SDL)/Product Security approach to reduce vulnerabilities in the code that they produce.

The lack of security and portability of current smart contract technology is a significant barrier to enterprise adoption.

## *3.6 Smart Contract Instrumentation and Auditability*

Enterprise applications expect to consume APIs that produce rich metrics. Blockchain APIs currently do not provide such parameters at the smart contract execution layer.

The lack of smart contract execution metrics is another strike against guarding against malicious and/or vulnerable smart contracts. Without such support, it is impossible to build a security framework that “watches” smart contract invocation and learns to recognize potentially malicious smart contract execution.

In particular, the table below highlights the vast number of scripting languages used to access different blockchain implementations[32].

| Blockchain | Scripting Language |
|---|---|
| Bitcoin | Forth-like Script |
| Ethereum | Solidity, Serpent, LLL |
| Multichain | None (no smart contract support) |
| Hyperledger | Golang, Java (in progress) |
| Hyperledger Sawtooth | Python |
| Lisk | JavaScript |

**Table 1 - Smart Contract Scripting Implementations**

The multi-chain world described in Section 3.4 brings with it significant challenges for monitoring and auditing smart contract invocation and execution.

## *3.7 Search Capabilities*

Researchers from the Knowledge Media Institute (KMI) at the UK Open University have taken a thorough look at the difficulties associated with indexing a blockchain[33].

> *"Distributed ledgers based on blockchains do not have a central registry and, due to their structure, are not straightforward to search*."

> *"…the key point to note is that blockchains are strictly time-ordered structures. Where related data exists across multiple blocks (as inevitably it must), there is no inherent way to identify, group or query it."*

Blocks in a blockchain often contain blobs unaccompanied by metadata. This also makes it difficult to audit and search data across these blockchain environments (especially across multiple chains) while looking at the history of transactions.

The researchers at KMI believe that the solution lies in the creation of a standard vocabulary or ontology.

> *"To generate interoperable Linked Data, it would be helpful to use a standard ontology or vocabulary to represent blockchain concepts."*

Unfortunately, such a mechanism does not yet exist across multiple blockchains, which leaves developers at a distinct disadvantage when integrating ledger data into analytic environments. For enterprise companies, this is a significant roadblock to broad adoption of blockchain. There is an expectation of easy and timely retrieval.

# 4 The VMware Blockchain Stack

The VMware research team has looked at many of the problems described in Section 3 and has designed a blockchain stack capable of addressing these obstacles. In this section, we will briefly describe the stack (depicted in Figure 15) and then expand upon each layer in Section 5.

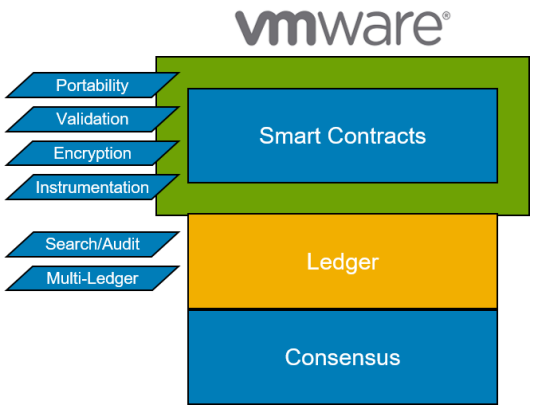



**Figure 15 - VMware's Blockchain Stack**

The bottom-most layer is VMware's consensus layer. This layer addresses the first three concerns listed in Section 3:

- Performance (Section 3.1)
- Time-to-Finality (Section 3.2)
- Data Consistency (Section 3.3)

The consensus layer contains a new Scalable Byzantine Fault Tolerant algorithm (SBFT) that provides faster throughput and 3f+1 node fault tolerance. Early research results indicate that SBFT can offer better performance, improved time-to-finality, and fault-tolerance than comparable industry approaches. Figure 16 highlights VMware's SBFT performance results as compared with a Practical Byzantine Fault Tolerant approach (PBFT).

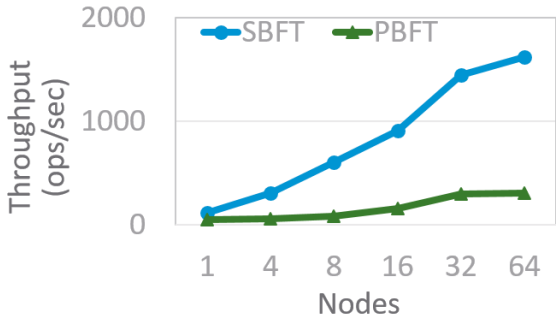



**Figure 16 - VMware's SBFT Performance Advantages**

The middle layer contains VMware's ledger. The ledger has been designed to address two of the challenges described in Section 3:

- Multi-Chain or Multi-Ledger (Section 3.4)
- Search Capabilities (Section 3.7)

One blockchain will not "rule them all," and the design of VMware's ledger implementation considers multi-chain compatibility. The ledger layer has strong ties to the smart contract layer (described below), which uses domain-specific and ledger-neutral features to span multiple blockchain implementations.

Multi-chain integration is described in more detail in Section 5.10.

The ledger layer also provides key-value store functionality that assists developers in searching and auditing the linear “chain of blocks” that make up a blockchain.

The final, top-level layer depicted in Figure 15 is VMware’s smart contract layer. The layer has high visibility (and provides significant functionality) to blockchain developers and system integrators. It addresses a specific set of problems described in Section 3, including:

- Multi-Chain or Multi-Ledger (Section 3.4). VMware has created a portable, domain-specific language (DSL) for smart contracts.
- Secure and Portable Smart Contracts (Section 3.5). The DSLs above enable code inspection for quality assurance, which protects against insecure contracts. Smart contracts can also be encrypted (as opposed to cleartext).
- Smart Contract Instrumentation and Auditability (Section 3.6). Statistics about smart contracts are tallied and query-able by the framework.

The VMware blockchain stack will, of course, be deployable to vSphere with push-button ease and compatible with VMware tooling.

Section 5 will describe how the features contained within this stack integrates with existing IT architectures by leveraging the rest of the Dell Technologies portfolio.

# 5 Dell Technologies and Blockchain

Enterprise companies wishing to create blockchain applications can easily deploy VMware’s stack in their lab and begin writing blockchain applications that experience enterprise-class behavior.

However, attempting to move the stack (and new applications) into a production environment (e.g. to run alongside their existing applications), a long list of issues will begin to surface. Many of these concerns are not specific to blockchain (and therefore did not receive discussion in Section 3). The challenges start at the application layer and extend all the way down to the infrastructure layer.

For some problems, the Dell Technologies portfolio has a “ready answer” and a clear path to implementation. In other cases, Dell’s customers and partners will need to innovate to completion. The Dell Technologies portfolio provides either an answer or a starting point for every one of these integration complexities.

Figure 17 below provides a “blockchain pinwheel” framework for discussing these challenges.

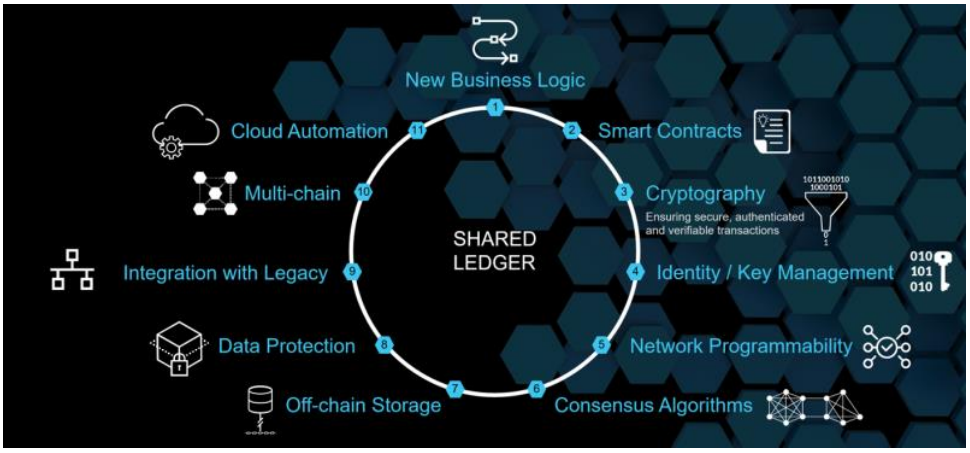



**Figure 17 - The Complexities of Integrating Blockchain Applications**

## *5.1 Writing and Deploying New Blockchain Business Logic*

The first problem often encountered when attempting to integrate new blockchain applications into an existing ecosystem is related to the speed at which these new applications must be written and deployed. Given the relative immaturity of blockchain ecosystems, continuous delivery of new blockchain features, functions, and bug fixes will be a priority.

Continuous delivery means that blockchain developers will frequently need to push out new software in days or hours (if not minutes).

This requirement conflicts with the extended release cycles that may be currently in place in traditional application development environments.

Because of this reality, a primary precondition for integrating blockchain applications is to transform how internal teams build software. A logical first step for developing blockchain applications would include a Pivotal Labs Professional Services engagement to introduce DevOps/Agile development concepts into the environment.

Without an enterprise-wide ability to develop and deploy *all* applications in an agile, cloud-native manner, inevitable bugs and problems in new blockchain logic will represent an increased risk to the business.

In addition to the requirement to use agile techniques for development, blockchain applications must be able to leverage cutting-edge services in the enterprise, including:

- The ability to run blockchain logic as part of a container framework (e.g. Kubernetes).
- The ability to call blockchain application logic in a server-less fashion (e.g. as part of a call-back function).
- The ability for blockchain application logic to leverage other services (e.g. cryptographic services, see below) from a marketplace.

All of the bulleted items described above are available to developers via Pivotal Cloud Foundry (PCF) 2.0, which is a must-have for blockchain development.

Figure 18 highlights the integration of new blockchain code into a standard agile development process (the “Pivotal Labs Approach” top layer) and a common deployment process (the “PCF 2.0” bottom layer).

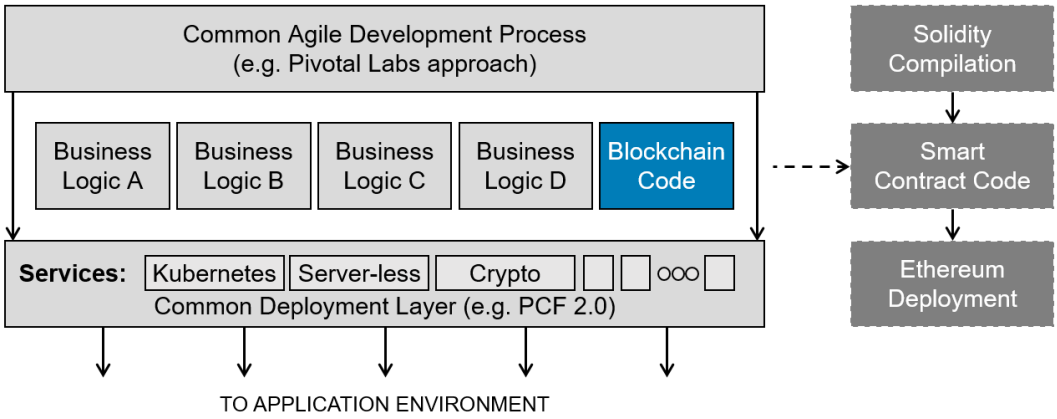



**Figure 18 - Common Development and Deployment of Blockchain Code**

The PCF 2.0 architecture depicted in Figure 18 allows for the development of any application (web, mobile, etc.) to integrate with a blockchain deployed directly within the Pivotal Application Services (PAS) layer (formally known as ERT, or Elastic Runtime).

With an agile development process and deployment layer in place, the next problem to solve is the integration of smart contract code.

## *5.2 Smart Contract Development and Deployment*

The deployment of smart contracts is typically platform-specific; Ethereum smart contracts deploy via the Ethereum framework, Hyperledger smart contracts deploy via Hyperledger, etc.

A corporation should strive to have one standard deployment framework for all of their code. Figure 19 highlights the integration of smart contracts into the structure depicted in Section 5.1.

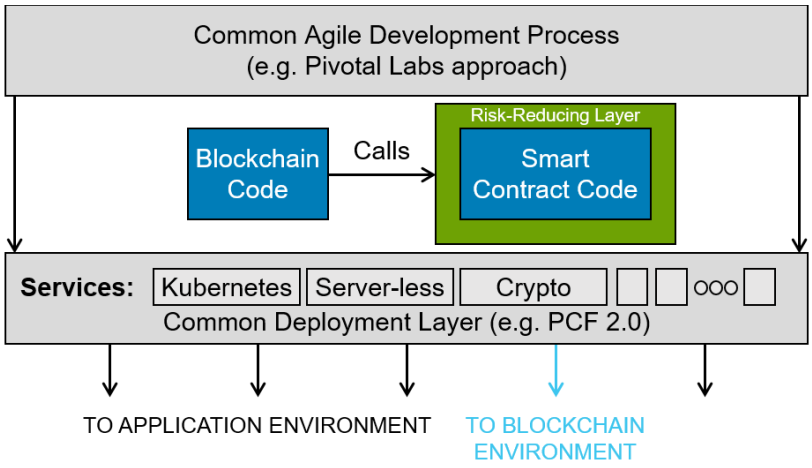



**Figure 19 - Risk-reducing Smart Contract Development and Deployment**

Note that the blue arrow leads to a blockchain environment. Smart contracts do not deploy like other applications; they must integrate with a specific ledger.

Note also that smart contracts embed within a risk-reducing layer (as implemented by VMware). Figure 20 highlights this approach.

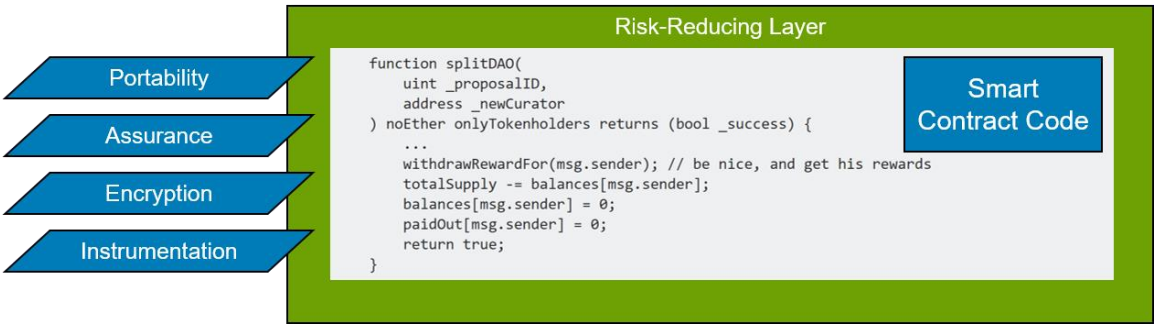



**Figure 20 – VMware Smart Contract Risk Reduction**

This figure highlights the (poorly-written) smart contract code from Figure 13 while also emphasizing the benefits that VMware's risk reduction environment offers:

- Portability: VMware's smart contract environment can accept natively written smart contract code (Ethereum in the example above) and run it on other blockchains (e.g. Hyperledger). This portability can preserve the investment that the business has made in the previous writing of existing smart contracts. More importantly, VMware supports the Domain-specific language (DSL) capability described in Section 4. The use of the DSL layer provides not only portability but assurance as well.
- Assurance: The use of a domain-specific language allows VMware to provide sound detection logic to guard against poorly-written smart contracts. At compile time this detection logic can alert developers to the risks that are common in specific domains.
- Encryption: VMware's environment supports encrypted smart contracts. Other implementations (e.g. Bitcoin) implement cleartext smart contracts, which allow

any developer to openly inspect contract logic that may reveal too much about corporate business agreements.

- Instrumentation: Should a hacker focus on a specific smart contract API to exploit its' business logic, VMware's smart contract layer keeps statistics about smart contract API calls and execution. These metrics will become critical in monitoring and managing risk from a security perspective.

With a risk-reduced smart contract development and deployment process in place, the next area of risk for blockchain application integration is the use of private keys during ledger transactions.

## *5.3 Cryptography*

Blockchain applications require private keys to create digital signatures. These signatures are a critical part of validating ledger entries. Constructing a strategy for the management of these keys is of primary importance to enterprise blockchain deployments.

In the consumer world, Bitcoin users will either store their private key in a local wallet or rely on a 3rd party to manage the key for them.

Researchers from Dell Technologies' SecureWorks division realized (several years ago) that this approach is an invitation for hackers. Figure 21 below[34] (published by Forbes) emphasizes SecureWorks' 2014 discovery that as the price of Bitcoin rose, the number of malware programs attempting to access the blockchain fraudulently rose as well.

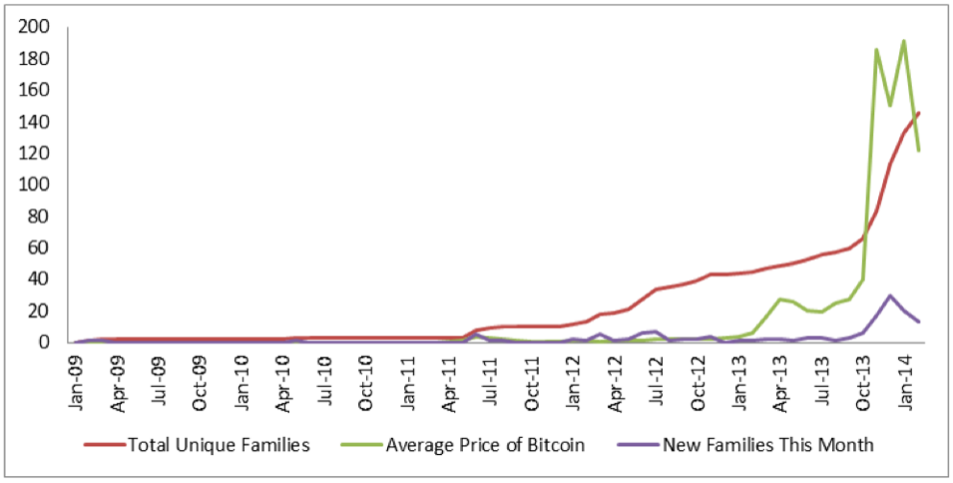



**Figure 21 - # of Bitcoin-stealing malware families detected by Dell SecureWorks**

The Forbes article also stresses the conventional approaches that these malware families used to steal from the blockchain:

> *To steal victims' bitcoins, most of the malware that SecureWorks found simply searches out common file types such as "wallet.dat" that might store private keys that control a user's coins. Any keys the malware finds are exfiltrated over FTP or*

*HTTP connections to a remote server, which uses them to transfer the victim's bitcoins to their own wallet.*

One might argue that the use cases described in Section 2 do not rely on Bitcoin (or cryptocurrencies in general) and that these types of attacks are therefore not relevant for enterprise applications. This argument misses the point. Asymmetric cryptography is at the core of most blockchain implementations; the theft of a private key would be disastrous for the business (whether the underlying ledger leverages cryptocurrencies or not).

The location of private keys is typically at the "endpoint" of a blockchain transaction (e.g. a desktop or mobile application). These endpoints are targets for hackers that wish to steal private keys and acquire the ability to insert transactions into a ledger maliciously. For cases where private keys are referenced locally by blockchain applications (e.g. private keys are in a "wallet.dat" file), malware detection software should be in place. Figure 22 highlights the addition of RSA Net Witness Endpoint Management software that detects the potential theft of private keys.

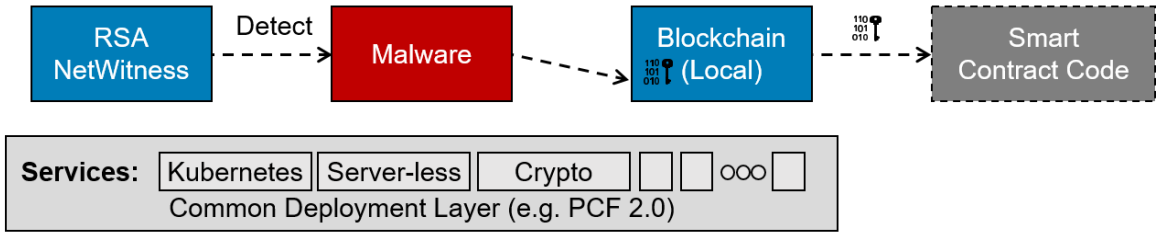



**Figure 22 - RSA NetWitness Endpoint Management Software**

Public-key vulnerabilities are also essential to address:

1. Key pair generation algorithms. A poor choice of algorithm (e.g. one that generates small key lengths) can make it easier for an attacker to exploit.
2. Protection of the root store used to validate public keys. Without this protection, an attacker may introduce their own private and public key pair and thus fool the consensus algorithm into believing the transaction is valid.

Technologists at Dell Technologies' RSA Labs recommend the creation of services for essential security functions (such as digital signatures, encryption, and hashing). These services could be developed and offered in-house via familiar and easy-to-use APIs that are well-known to developers. For example, security services can be delivered as a container and exposed via REST APIs to facilitate application development and deployment (e.g. the "Crypto" service depicted in Figure 19). These security functions can be grouped into common services and developed/maintained by a specialized security team. Alternatively, the APIs to these crypto services could be consumed "as-a-service" from a cloud provider.

The next section explores the relationship between corporate identity and key management.

## 5.4 Identity/Key Management

To explore the nuances of blockchain-based identity and key management, assume the following pre-conditions:

- A corporate identity "Bob" is maintained in a centrally-managed store (e.g. Active Directory).
- Keys exist within a public key infrastructure (PKI). A Certification Authority (CA) leverages this infrastructure.
- A CA registration process has already bound a key to Bob's identity.

Assume Bob runs a blockchain application in an attempt to insert a transaction into a ledger. There are three use cases to consider for Bob's identity.

1. Private corporate ledger. In this case, the blockchain operates within the context of one company. The organization's CA/PKI issues Bob's certificate and the blockchain transaction can proceed.
2. Hosted consortium ledger. An external entity (e.g. Virtustream, as discussed in Section 5.11) entirely manages the blockchain and the PKI. Bob can continue to use his corporate identity provided that the hosting company has access to his corporate identity management system. Otherwise, Bob must create a second, external identity with the hosting company.
3. Public ledger. Bob will leverage a third party to manage his identity and keys fully.

Use case #2 represents extra work for a company, but the third option poses the biggest challenge. If there is business value to be realized from interacting with public ledgers, how can corporate policies manage external identities and keys?

One approach being explored by technologists at RSA is known as decentralized identities, in which Bob manages his own identity (on a blockchain no less!), and his employer digitally signs that identity to verify employment.

RSA is currently a member of the Decentralized Identity Foundation.

## 5.5 Network Programmability

In this section and the section that follows (Consensus Algorithms), we arrive at a critical layer that is "out of sight" (and therefore "out of mind") for most enterprise blockchain developers.

One may argue that the configuration of the underlying blockchain network and the consensus algorithms used to make decisions across that network may be the most critical aspect to consider for mission-critical blockchain operations.

Figure 23 introduces a way to think about the need for network programmability of a blockchain network.

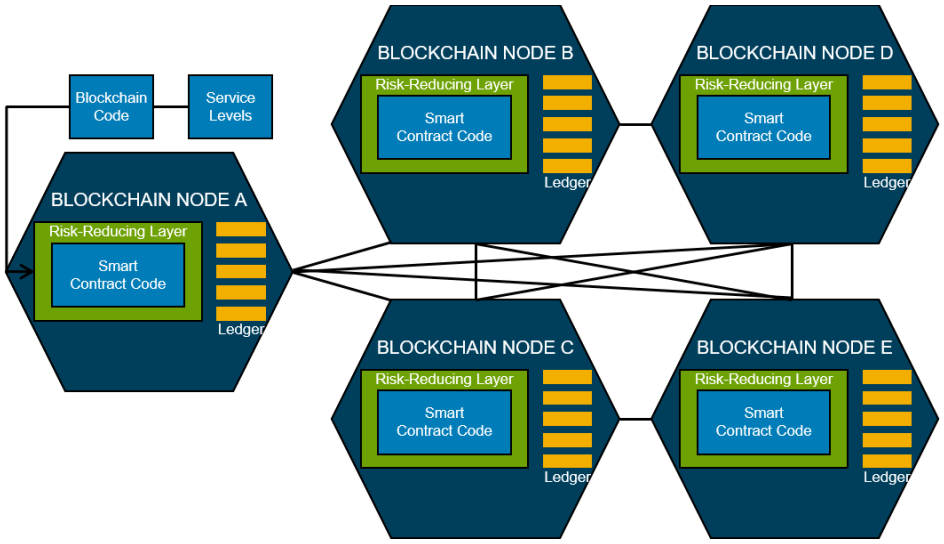



**Figure 23 - Network Programmability for Blockchain Configurations**

Figure 23 highlights an application ("Blockchain Code") that is calling a (risk-reducing) smart contract. The smart contract will attempt to insert a transaction into its local ledger (on Blockchain Node A). This insertion attempt will result in a broadcast of the transaction across all nodes.

Note that the application also comes with service level requirements that may specify enterprise-class needs regarding performance, resiliency, scalability, etc. These service level requirements may only be realized by appropriately configuring the underlying network. Consider the partial subset of network configuration options listed below[35]:

- Scalability: the number of network nodes can impact the choice of consensus algorithm across the nodes.
- Elasticity: surges in concurrent transactions may require the automated creation and addition of additional blockchain nodes.
- Security/Privacy: there may be requirements for multi-tenant privacy and encryption of data.
- Performance: specific bandwidth requirements may affect the configuration of virtual networks and also may impact the choice of higher- (or lower-) performing consensus algorithms.
- Fault tolerance: mission-critical blockchain applications may desire to ride through a high number of node failures without a significant decrease in performance.

This type of network programmability on a per-application basis can only occur automatically via the use of network services that sit on top of software-defined networking (SDN) APIs. For enterprise customers designing their networks for blockchain workloads, Dell Technologies offers two solutions to leverage (based on the environment):

- Dell Networking OS10: compliant with the Open Compute project[36].
- VMware NSX: delivers network and security services close to the application[37].

Either of these frameworks is interoperable with a wide variety of Dell networking hardware.

With a well-designed hardware and software networking strategy for blockchain in place, the next step is to consider the choice of industry consensus algorithms.

## *5.6 Consensus Algorithms*

Blockchain consensus algorithms function at the lowest layer of the blockchain ecosystem. It is critical to understand these nuances when considering the following questions[38]:

- How will decisions be made to accept/reject transactions?
- What is the "speed to finality" of these decisions?
- What are the scalability limits of the consensus algorithm?
- How much fault tolerance is built into the consensus?
- How much does performance suffer before and after crossing fault tolerance thresholds?

IT architects would do well to educate themselves on the advancements in consensus algorithms since the early days of Bitcoin. Fortunately, VMware researchers Ittai Abraham and Dahlia Malkhi have published a paper that traces this evolution: The Blockchain Consensus Layer and BFT.

Which algorithm works better under what circumstances? For many enterprise applications, the dominant issue is dealing with consensus at scale (dozens of nodes versus clusters of four to eight).

One of the first efforts to achieve consensus at a larger scale occurred in 2016 and is known as Byzantine Vertical Paxos (BVP). The protocol focuses on maintaining high-levels of throughput during elasticity and reconfiguration events (e.g. quickly scaling the number of blockchain nodes). The BVP paper starts as follows[39]:

> *"In this paper, we consider the challenge of driving a serious, industrial-grade infrastructure for Byzantine agreement and state machine replication. We focus on two aspects, elasticity (dynamic reconfiguration) and throughput."*

VMware's fast, elastic consensus algorithms are reaching a level of maturity where they are candidates for evaluation in production scenarios.

## *5.7 Off-chain storage*

Since the creation of the Bitcoin blockchain, application developers have looked for opportunities to store additional data into each transaction. Early developers, for example, leveraged the use of the Bitcoin OP_RETURN feature (as described on medium.com[40]):

> *"To store data on the Bitcoin blockchain we would enter the data in the OP_RETURN field of Bitcoin transactions. The OP_RETURN field allows a user to send a transaction that doesn't actually send money to anyone, but allows a small amount of data to be written to the Bitcoin blockchain. Each OP_RETURN output has a maximum size of 80 bytes, and each transaction can have one OP_RETURN output."*

Nearly all of the enterprise use cases mentioned in Section 2 involve a data asset. Many of them are much too large to insert into a ledger (as highlighted by Bitcoin's 80-byte limit). In fact, platforms such as Ethereum actually charge a fee for storing large objects in a transaction. This cost has caused many Ethereum developers to turn to object-based storage solutions. These solutions associate data with a unique hash ID. A trending storage solution often used with Ethereum is the Interplanetary Filesystem (IPFS). Blockchain technologist John Lilic explains the use of IPFS to counter "blockchain bloat"[41]:

> *"An interesting point here is the distinction between storing data on the blockchain and storing hashes of data on the blockchain. On the Ethereum platform you pay a rather large fee for storing data in the associated state database, in order to minimize bloat of the state database ("blockchain bloat"). Thus it's a common design pattern for larger pieces of data to store not the data itself but an IPFS hash of the data in the state database."*

Dell EMC offers Isilon and Elastic Cloud Storage, which both support object-based storage access (but with more enterprise features than IPFS). Consider Figure 24, which highlights the ability of a blockchain application to access a ledger (the inner ring) and an ECS object store (the outer ring) simultaneously.

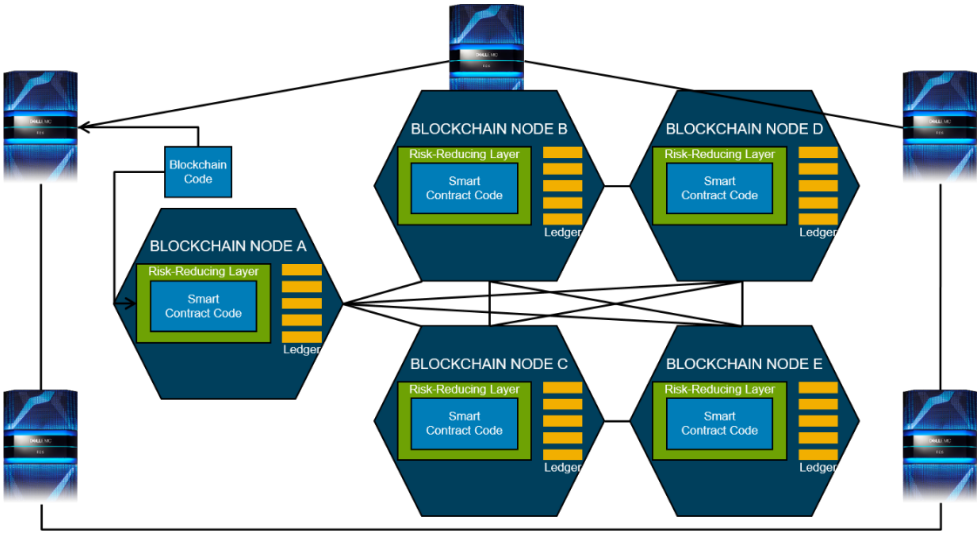



**Figure 24 - Blockchain Code Simultaneous Use of Ledger and Off-chain Object Store**

Once the design of off-chain storage is complete, there is another critical issue that remains: the relationship between ledger permissions and off-chain storage permissions.

Controlling access to off-chain storage assets has strong ties to the private key and identity management solutions proposed in sections 5.3 and 5.4.

The beauty of using enterprise-class storage solutions like Isilon and ECS revolves around the fact that it is a hardened system with a rich set of storage services (e.g. data protection). The enterprise data stored in either platform is protected and stored with integrity.

The same is not necessarily true for all blockchain ledgers. How are ledgers protected and managed with integrity?

## *5.8 Data Protection*

One might question the need for data protection tools for blockchain. The Bitcoin blockchain, for example, is replicated on every node in the system; handling corruptions and failures occurs by rebuilding the ledger and the index (using the "-reindex" or "—reindex" command line options for bitcoind). As time goes by, the bitcoin ledger gets larger and larger, and thus the restore times grow in length as well.

Ledger recovery concerns are not limited to just the Bitcoin blockchain, but all blockchains. Due to the append-only nature of blockchain, data stored within the ledger never gets deleted or expired. Blockchains can reach a considerable size, making the approach of "re-downloading" or "re-indexing" them unpractical.

The problem with the Bitcoin approach is that enterprise applications expect a level of backup and recovery integration that happens instantaneously and can't afford to wait for the download of potentially large chains.

Consider Figure 25, in which a copy of a ledger is "snapped" and consumed by an analytic application via the use of traditional Dell EMC techniques.

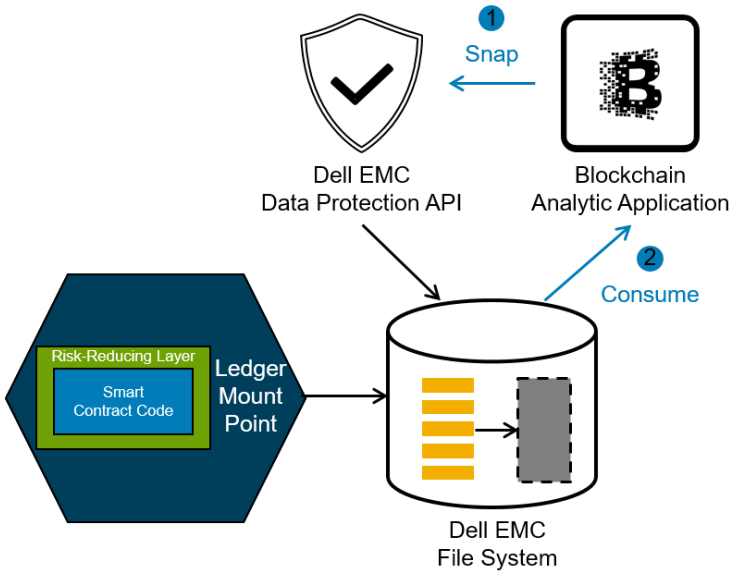



**Figure 25 - "Snapping" a File System Ledger**

Do these protection mechanisms still apply in the blockchain world? Ledgers are append-only, and operating on a copy can be as simple (in theory) as starting up an additional node.

Explore design considerations for blockchain data protection and integrity via the following questions:

- What happens when a ledger on a local node becomes corrupt or inaccessible?
- Will mission-critical blockchain applications experience seamless failover on transaction failures? If so, how?
- How can analytic applications access ledger copies without interfering with production operation?

The answer to these questions can guide application developers in the use of data protection functionality available from Dell EMC.

Similar integration issues arise when considering the integration of a full blockchain stack into an existing system.

## *5.9 Integration with Existing Architectures*

In addition to the identity management concerns described in Section 5.4, there are additional enterprise integration issues to consider when introducing a ledger into an existing IT architecture:

- Converged Infrastructure
- Security Monitoring
- Data Integration
- Risk Management

Figure 26 features the integration of the VMware stack to highlight integration strategies via relevant Dell Technologies solutions.

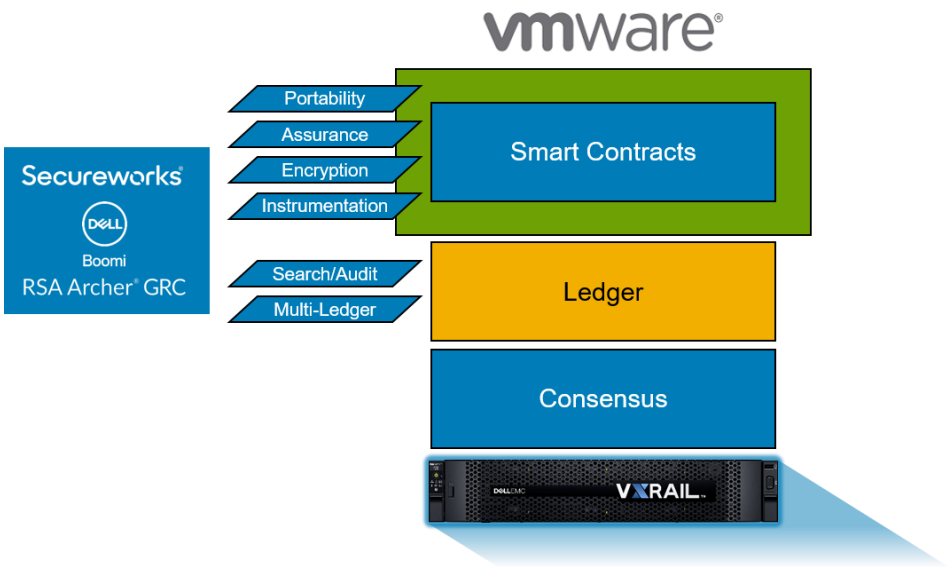



**Figure 26 - VMware Blockchain Stack Integration**

Converged Infrastructure

Figure 27 shows that blockchain, from a compute perspective, is just another workload, and as such it will benefit from running on the most modern hyper-converged infrastructure (in this case, VxRail).

VxRail is hyper-converged infrastructure that can be used to run a large number of heterogeneous workloads based on the PowerEdge server family[42]:

- General purpose HCI nodes that can support all-flash (or hybrid) storage needs (G410/G410F).
- Entry level nodes that can start small and scale big (E460/E460F).
- High-performance nodes that can run heavy workloads such as databases (P470/P470F).
- Dedicated graphics nodes for specialized (e.g. VDI) workloads (V470/V470F).
- Storage-dense nodes for workloads such as collaboration and data/analytics (S470).

The VMware ledger can be packaged and deployed (a) within the same hyper-converged infrastructure and (b) alongside all of the workload types listed above. VxRail has also been extensively tuned and optimized for VMware technologies[43].

Security Monitoring

With the VMware blockchain stack running as a full-fledged service on a VxRail infrastructure, the next integration challenge will be to ensure that security threats targeting this new stack (as well as any new blockchain applications) can be monitored, detected, and resolved in the context of existing security frameworks.

SecureWorks has decades of experience developing, tuning, and enforcing security policies on behalf of customers. Their technology performs frequent monitoring of low-level APIs and can detect anomalies in service usage. SecureWorks can be enhanced to

mine rich metrics from VMware's smart contract layer to inspect the boundary between applications and underlying blockchain logic continually. Figure 27 depicts this integration.

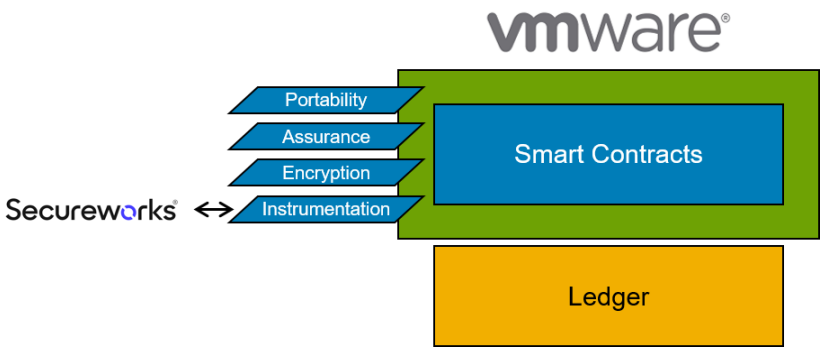



**Figure 27 - SecureWorks Monitoring with VMware Smart Contract Instrumentation**

This monitoring process at the API level is similar to what SecureWorks already does today for databases (and blockchain is, in essence, a highly trusted and decentralized database). SecureWorks, therefore, will be able to leverage VMware's Smart Contract instrumentation to detect unusual and malicious access patterns.

Data Integration

As rich information gets inserted into an enterprise blockchain, the ability to combine this new data with existing data sets and applications will be not only attractive but necessary. This integration can appear challenging as legacy applications may be present in both cloud and non-cloud forms.

The use cases listed in Section 2 of this paper will further exacerbate this challenge. Developers will need to integrate data from across a wide variety of verticals (e.g. the Supply Chain, IoT).

In other words, newly-integrated blockchain technology must interact with a hybrid landscape of applications and data sources.

One of the Dell Technologies family that will prove most critical for blockchain integrations is Dell Boomi. Boomi is a set of data integration services that can connect any two systems, whether they be cloud-based (or not), on-premises (or not) no matter what the vertical (e.g. Supply Chain, Medical, IoT, etc.). Figure 28 highlights the addition of VMware's blockchain into the Boomi data integration family.

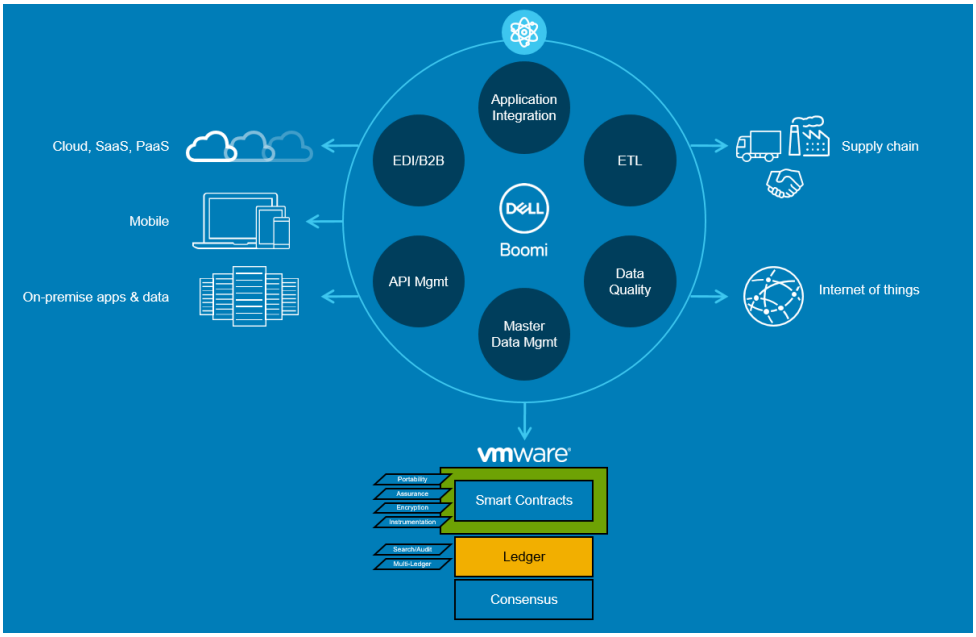



**Figure 28- Adding Blockchain Functionality to Boomi's Data Integration Framework**

As more and more blockchain smart contracts get written and deployed, Boomi will be able to discover and catalog these APIs for use across the enterprise (this will become especially useful in the multi-chain environment described in Section 5.10).

A catalog of smart contract APIs can also help an enterprise manage any risks associated with the introduction of blockchain applications.

Risk Management

The final area of consideration for blockchain integration is risk management. The adoption of blockchain into the enterprise is related to risk in two ways:

1. The introduction of any new technology into a corporate environment involves risk. This risk should be sized and monitored (especially for blockchains shared among a consortium of companies).
2. The reasons for introducing new blockchains may very well be to reduce risk (e.g. to accelerate regulatory reporting during audits). Any and all uses of blockchain to mitigate corporate risk should accompany new processes that measure risk reduction.

In both cases, risks need to be documented and managed. One such tool for codifying these business risks is RSA's Archer GRC (Governance, Risk, and Compliance) Platform. The Archer platform not only allows for risks to be documented, but the platform also supports data gathering and analysis to make sure that all systems are performing as advertised (thus managing risk).

Archer facilitates mapping between documented risks (encoded into a Risk Catalog) and programmatic scripts that periodically measure those risks. Figure 29 highlights a set of blockchain-related risk catalog (top row) that the Archer framework can verify by

programmatically calling blockchain-related plug-ins (e.g. smart contract APIs, supplied by Boomi for example).

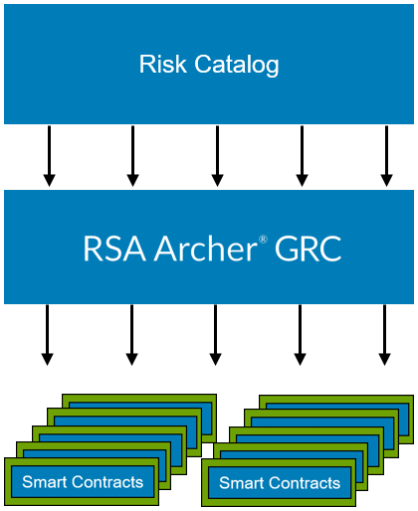



**Figure 29 - Managing Risk via Archer Integration with Smart Contracts**

## *5.10 Multi-chain*

Another challenge facing enterprise blockchain applications will be the co-existence of multiple blockchain implementations. One blockchain will not "rule them all." It will likely be common for an enterprise blockchain to come face to face with a ledger built by a different vendor (this may occur within one company).

There are two approaches to address this problem.

The first is to code smart contract business logic using VMware's portability layer (as highlighted below). This approach solves the problem at the highest level and relies on VMware's underlying plumbing to facilitate cross-ledger interactions. Figure 30 highlights this approach.

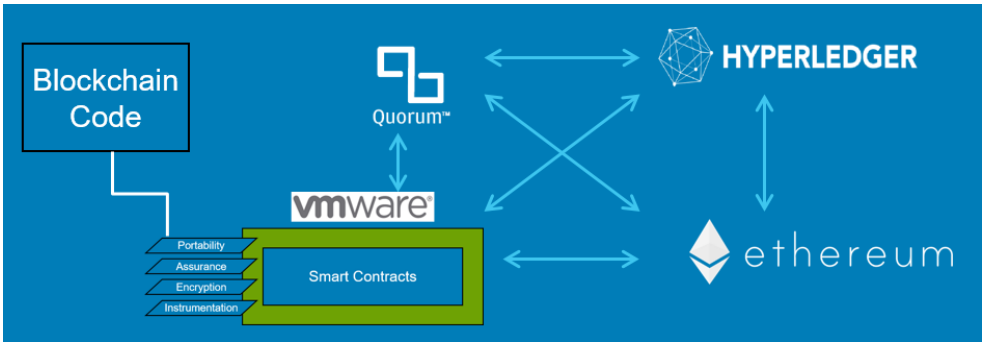



**Figure 30 - Example Permutations Requiring Ledger Interoperability**

The second strategy is to use Boomi as an integration platform across ledgers.

In Section 5.9 it has already been established that Boomi can discover and build a catalog of available smart contract APIs (e.g. for use in governance, risk, and compliance).

This approach also applies to multiple chains. In fact, Boomi supports not only query but also invocation. Figure 31 highlights the use of Boomi in a multi-chain environment.

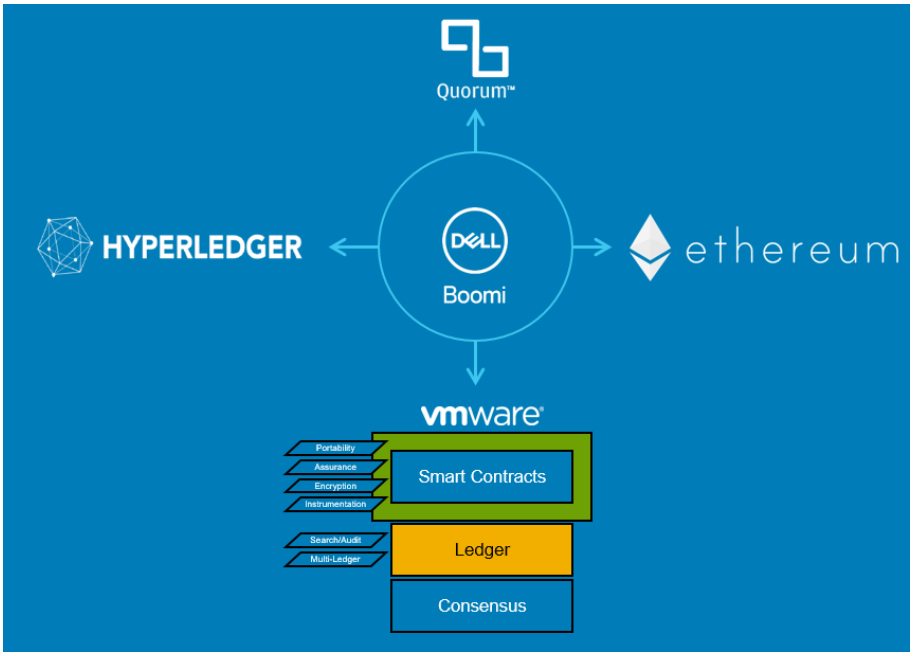



**Figure 31 - Multi-Chain Support via Dell Boomi**

## *5.11 Cloud Automation*

There may be many reasons why an enterprise may wish to consume blockchain-as-a-service (BaaS). The decision to build private blockchain infrastructure (using the guidance described in Section 5) often occurs because end-to-end enterprise-class BaaS capabilities and services (scale, performance, resiliency, management, etc.) are not currently available from some existing public or private cloud providers. Alternatively, many enterprise accounts would instead prefer to experiment with enterprise-class BaaS before deciding to invest in private cloud implementations. Many customers do not yet have the expertise to manage and operate the ledger software, for example, as is the case with both public and private cloud providers.

There are additional business reasons for BaaS: use cases that involve groups of participants (also known as consortiums) who want to work together but do not trust any one of their members to control the data or the system. Another variant of this use case is the establishment of non-repudiation for the financial sector.

For these use cases, Virtustream can provide hosted ledger solutions inside their enterprise-class cloud. Ledgers architectures can be automatically provisioned, and the networking between the nodes can be secured. Once the ledger is running,

Virtustream's Managed Services provides monitoring, patching, and backup for a constituent's node, along with custodian servers of the ledger. They can also reissue credentials in an audited, compliant fashion. Managed services can also be offered on premises for supported ledger software.

In these consortiums members securely exchange data (sometimes encrypted) on a shared ledger. The rationale is that each member has the same copy of the data. The data is immutable, preventing change and giving a picture of the consortium's data from the beginning of its history. A given party can attest to another party's data and stamp it with approval. If one party privately writes it to a database, the others can't know it was written correctly. By using encryption, both asymmetric and symmetric, Virtustream can create data masking rules to implement complex consortium requirements.

A Virtustream Secure Enterprise Ledger allows the offload of a significant number of complex system administration processes:

- Who manages the private keys?
- What happens if a member loses keys?
- How are keys cycled according to regulatory requirements?
- How do new blockchain nodes join the private ledger?
- How are outside malicious actors prevented from disturbing the network?
- How is ledger software patched and maintained?
- Who can add new members? Who can remove members?

These problems (and more) require an otherwise uninterested custodian to provide unbiased services. This phenomenon is similar to something already taken for granted on the internet: TLS/SSL. Companies like Digicert provide custodial services and attest that they are legitimate. Corporations then trust Verisign to revoke and add members to a trusted network of participants worthy of the padlock in industry browsers. Virtustream's implementation of a ledger also draws parallels with modern cloud computing operations; businesses who otherwise might be competitors all place the administration and monitoring of their infrastructure in the hands of a shared custodian who has no real competing interest in the data or applications on those resources.

# 6 Summary

This paper reviewed how enterprise customers are writing blockchain applications for the management and transfer of data assets, the broadcast of data, verification of credentials, and supply chain transparency.

Obstacles to enterprise blockchains were also reviewed, and include performance, time-to-finality, data consistency, multi-chain, secure and portable smart contracts, smart contract instrumentation and audit, and search capabilities.

The VMware blockchain stack was then described as a design that addresses many of the obstacles found in an enterprise application development environment.

Lastly, a Dell Technologies blockchain vision and capabilities overview illuminated construction approaches for building enterprise-class blockchain solutions.

Figure 32 represents a graphical look at the breadth of relevant Dell Technologies products that apply to the development and deployment of blockchain applications.

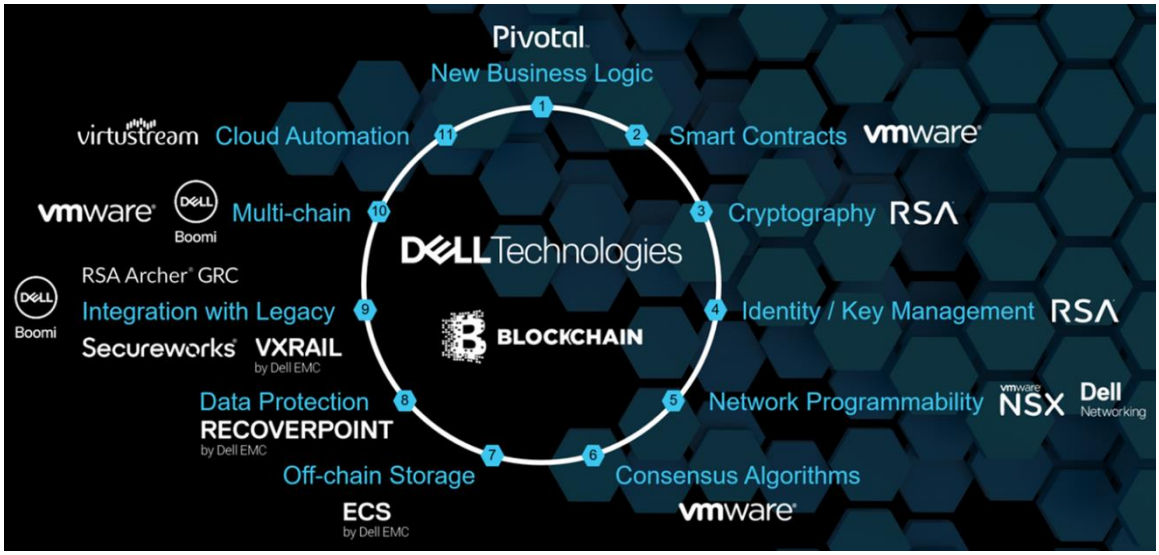



**Figure 32 - The Breadth of the Dell Technologies Blockchain Portfolio**

These products present themselves as a pinwheel of interwoven functionality.

In this final section, these capabilities are brought together into a solution. Figure 33 highlights this solution.

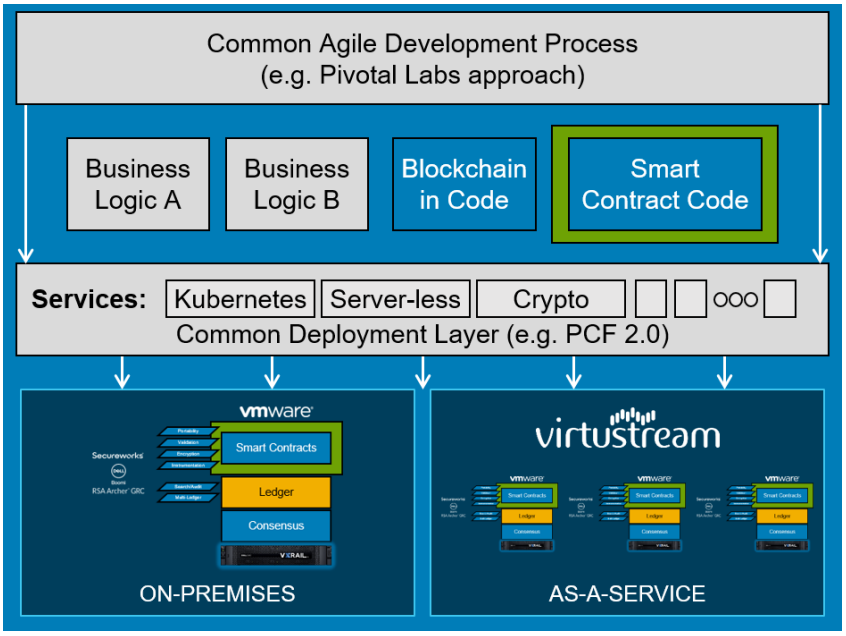



**Figure 33 - Dell Technologies Blockchain Solutions**

The elements of this solution consist of the following:

- Pivotal supplies an agile blockchain (and smart contract) application development process and deployment framework.
- VMware's blockchain stack provides secure, enterprise-class characteristics to these blockchain applications, with particular attention paid to the smart contract, ledger, and consensus layers.
- RSA enables key management, identity support, and blockchain-related governance (Archer) while also monitoring endpoint threats.
- Dell EMC provides secure data protection, scalable off-chain storage, powerful hyper-converged servers, network configuration, and overall infrastructure security support.
- SecureWorks delivers security monitoring at the blockchain smart contract API level.
- Boomi enables the integration of ledger data with existing application and data entities and brings smart contract APIs into a data integration ecosystem.
- Virtustream wraps all of the above into an automated cloud service.

The beauty of this architecture is that application development and deployment occurs consistently and quickly no matter whether the applications deploy on-premises or off-prem in the Virtustream cloud.

2018 will be a time of continued growth in blockchain application deployment.

The employees that make up the Blockchain Interest Group within Dell Technologies would like to collaborate on blockchain opportunities across the industry. To contact them, please email blocksteer@dell.com.

---

**Endnotes**

[1] Wikipedia. Correspondent Account. January 23, 2018. https://en.wikipedia.org/wiki/Correspondent_account.
[2] Lonnen, Bradley, and Roels, Paula. Correspondent Banking Reimagined on Blockchain. The Banker. July 2016. http://www.cib.db.com/docs_new/2016_07_The_Banker_Correspondent_Banking_Blockchain_P_Roels_B_Lonnen.pdf.
[3] Jäger, Daniel. Bitcoin: The Digital Currency and How Disruptive Is It for the Banking Industry? LinkedIn. September 12, 2016. https://www.linkedin.com/pulse/bitcoin-digital-currency-how-disruptive-banking-industry-daniel-j%C3%A4ger/.
[4] Swan, Melanie. Bitcoin and Blockchain Explained: Cryptocitizen Smartnetwork Trust. Slideshare.net. https://www.slideshare.net/lablogga/bitcoin-and-blockchain-explained-cryptocitizen-smartnetwork-trust. Slide 49.
[5] Forrest, Paul. Blockchain Technology – the Film Industry Use Case. MBNSolutions.com. May 18, 2017. http://www.mbnsolutions.com/blockchain-technology-the-film-industry-use-case/.
[6] Hanson, Rob. Risks and Opportunities for Systems Using Blockchain and Smart Contracts. CSIRO Data 61. May 2017. http://www.data61.csiro.au/~/media/D61/Files/Blockchain-reports/Blockchain-RisksandOpps-PDF.pdf?la=en&hash=D0765B85166B783E9FD13FC4EAEBDCB03E716631.
[7] Ali, Muneeb, Nelson, Jude, Shea, Ryan, and Freedman, Michael. Blockstack: A Global Naming and Storage System Secured by Blockchain. Usenix. https://blockstack.org/blockstack_usenix16.pdf.
[8] Decentralized Identity Foundation. Home Page. January 24, 2018. http://identity.foundation/.
[9] The Technology Headlines. Illinois Opens Blockchain Development Partnership with Hashed Health. January 24, 2018. https://thetechnologyheadlines.com/illinois-opens-blockchain-development-partnership-with-hashed-health.html.

[10] About Blockcerts. Blockcerts.org. January 23, 2018. https://www.blockcerts.org/about.html.
[11] Schmidt, Philipp. Certificates, Reputation, and the Blockchain. Medium.com. January 23, 2018. https://medium.com/mit-media-lab/certificates-reputation-and-the-blockchain-aee03622426f.
[12] Oliver Wyman. Blockchain: The Backbone of Digital Supply Chains. January 24, 2018. http://www.oliverwyman.com/our-expertise/insights/2017/jun/blockchain-the-backbone-of-digital-supply-chains.html.
[13] Deloitte. Using Blockchain to Drive Supply Chain Innovation. Deloitte.com. January 24, 2018. https://www2.deloitte.com/us/en/pages/operations/articles/blockchain-supply-chain-innovation.html#top.
[14] Blockchain.Info. Transactions Per Second. December 2017. https://blockchain.info/charts/transactions-per-second.
[15] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin G¨un Sirer , Dawn Song, and Roger Wattenhofer. On Scaling Decentralized Blockchains (a position paper). January 24, 2018. https://www.tik.ee.ethz.ch/file/74bc987e6ab4a8478c04950616612f69/main.pdf.
[16] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, Kian-Lee Tan. BLOCKBENCH: A Framework for Analyzing Private Blockchains. January 24, 2017. http://www.comp.nus.edu.sg/~ooibc/blockbench.pdf.
[17] Ibid. Figure 5(b).
[18] Wagner, Mark. Hyperledger Announces Performance and Scalability Working Group. Hyperledger.org. January 24, 2017. https://www.hyperledger.org/blog/2017/06/08/hyperledger-announces-performance-and-scalability-working-group.
[19] Hyperledger Meeting Minutes. Google Docs. January 24, 2018. https://docs.google.com/document/d/180YsNhqPUU7RMqLzDhkpSHuW-Go8etoN8XKoAY2Iftc/edit.
[20] Hileman, Garrick, and Rauchs, Michael. Global Blockchain Benchmarking Study. Cambridge Centre for Alternative Finance. https://www.jbs.cam.ac.uk/fileadmin/user_upload/research/centres/alternative-finance/downloads/2017-09-27-ccaf-globalbchain.pdf.
[21] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, Kian-Lee Tan. BLOCKBENCH: A Framework for Analyzing Private Blockchains. January 24, 2017. http://www.comp.nus.edu.sg/~ooibc/blockbench.pdf. Section 4.1.2
[22] Nakamoto, Hitoshi. Bitcoin: A Peer-to-Peer Electronic Cash System. Section 5. January 18, 2018. https://bitcoin.org/bitcoin.pdf.
[23] Hileman, Garrick, and Rauchs, Michael. Global Blockchain Benchmarking Study. Cambridge Centre for Alternative Finance. https://www.jbs.cam.ac.uk/fileadmin/user_upload/research/centres/alternative-finance/downloads/2017-09-27-ccaf-globalbchain.pdf.
[24] Vysya, Venkatesha, and Kumar, Anjani. Blockchain Adoption in Financial Services. Infosys. December 2017. https://www.infosys.com/industries/financial-services/white-papers/Documents/blockchain-adoption-financial-services.pdf.
[25] Coindesk. State of Blockchain 2017. Coindesk.com. https://media.coindesk.com/uploads/2017/09/State-of-Blockchain-Q2-2017-.pdf. Slide 53.
[26] MarcO. Is RCL an ILP Connector? Xrpchat.com. January 23, 2018. https://www.xrpchat.com/topic/12046-is-rcl-an-ilp-connector/.
[27] Jordan, Raul. How to Scale Ethereum: Sharding Explained. Medium.com. January 23, 2018. https://medium.com/@rauljordan/how-to-scale-ethereum-sharding-explained-ba2e283b7fce.
[28] Cepka, Angus. Quantstamp – ICO Review. Medium.com. January 28, 2018. https://medium.com/@acepka/quantstamp-ico-review-1e93f90680f3.
[29] Etherscan Ethereum Block Explorer. December 2017. http://web.archive.org/web/20170602184510/https://etherscan.io/accounts/c.
[30] Etherscan Ethereum Block Explorer. December 2017. https://etherscan.io/accounts/c.
[31] Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi. Step by Step Towards Creating a Safe Smart Contract: Lessons and Insight from a Cryptocurrency Lab. January 24, 2018. http://fc16.ifca.ai/bitcoin/papers/DAKMS16.pdf.
[32] Hintzman, Zane. Comparing Blockchain Implementations. December 2017. https://www.nctatechnicalpapers.com/Paper/2017/2017-comparing-blockchain-implementations/download.
[33] Third, Allan, and Domingue, John. Linked Data Indexing of Distributed Ledgers. Knowledge Media Institute. January 24, 2018. https://www.nctatechnicalpapers.com/Paper/2017/2017-comparing-blockchain-implementations/download.
[34] Greenburg, Andy. Nearly 150 Breeds of Bitcoin-Stealing Malware in the Wild, Researchers Say. Forbes.com. February 26, 2014. https://www.forbes.com/sites/andygreenberg/2014/02/26/nearly-150-breeds-of-bitcoin-stealing-malware-in-the-wild-researchers-say/#29f3f65a33d7.
[35] Todd, Steve. SDDL: Software-Defined Distributed Ledger. Information Playground. June 2017. http://stevetodd.typepad.com/my_weblog/2017/06/sddl-software-defined-distributed-ledger.html.

[36] Dell.com. Open Platform Software. January 24, 2018. http://www.dell.com/en-us/work/shop/povw/open-platform-software.
[37] Ayyar, Shekar. Leading our Industry into a Software-Defined Future with Strategic M&A. VMware.com. November 22, 2017. https://www.vmware.com/radius/leading-our-industry-into-a-software-defined-future-with-strategic-ma/.
[38] Todd, Steve. The Elements of Blockchain. Information Playground. October 2017. http://stevetodd.typepad.com/my_weblog/2017/10/the-elements-of-blockchain.html.
[39] Abraham, Ittai, and Malkhi, Dahlia. BVP: Byzantine Vertical Paxos. VMware.com. https://research.vmware.com/publications/bvp-byzantine-vertical-paxos.
[40] Omaar, Jamila. Forever Isn't Free: The Cost of Storage on the Blockchain Database. Medium.com. July 29, 2017. https://medium.com/ipdb-blog/forever-isnt-free-the-cost-of-storage-on-a-blockchain-database-59003f63e01.
[41] Lundkvist, Christian, and Lilic, John. An Introduction to IPFS. Medium.com. February 17, 2016. https://medium.com/@ConsenSys/an-introduction-to-ipfs-9bba4860abd0.
[42] Dell.com. Dell EMC VxRail Appliances. http://www.dell.com/en-us/work/shop/povw/vmware-vxrail.
[43] EMC.com. Dell EMC Accelerates IT Transformation with Extensive VMware-Optimized HCI. https://www.emc.com/about/news/press/2017/20170828-02.htm.