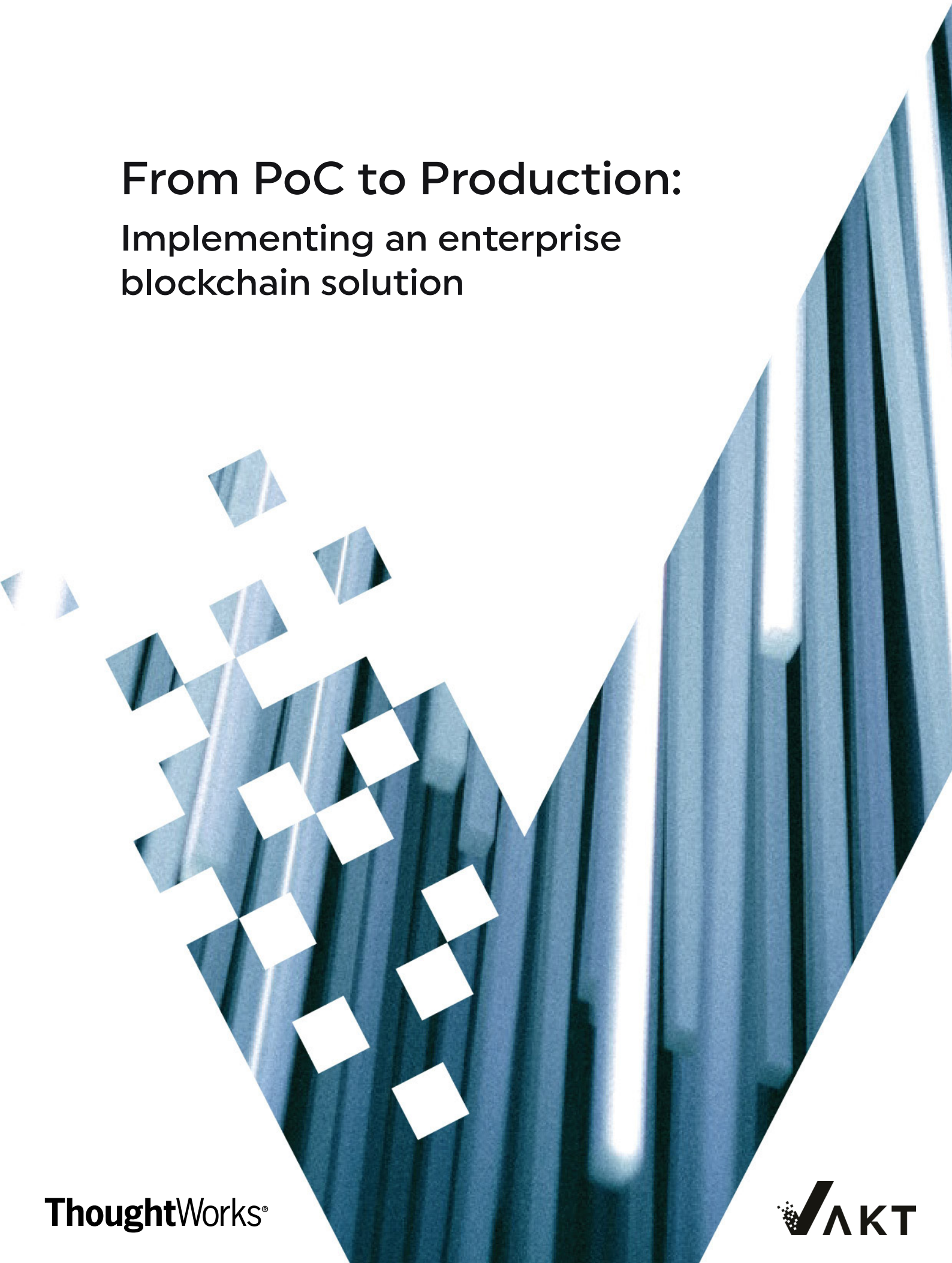
From PoC to Production:

Implementing an enterprise blockchain solution



ThoughtWorks®

AKT

INTRODUCTION

There have been innumerable blockchain proof of concept (PoC) initiatives in the past few years. These range across multiple industries and use cases. A PoC is a unique activity that needs a specific mindset and typically focuses on a small piece of the puzzle.

A PoC shows that something can be done, in a restricted context and in restricted time. Necessarily, it's about proving the technology and focus is on a specific instance of a business problem. The great thing about PoCs is that they give an opportunity to try out the technology safely, and – since they are framed as an experiment, with a limited budget – failure is often as good an outcome as success.

At some point however the training wheels need to come off, and we need to migrate from PoC to production.

The VAKT team have successfully pioneered the release of the world's first enterprise grade, production blockchain system for commodities trading, completed in an extremely short timeframe to initial launch. This whitepaper offers a high-level exploration of why a consortium of organisations chose to build a platform based on blockchain, and contrasts the stringent requirements of the enterprise with the technology available today, based on this experience.

ThoughtWorks® VAKT

PART 1:

Batteries not included

Deploying and operating a system to enterprise level requirements requires a shift in approach from PoC. Innovation and experimentation give way to concerns of usability, operability, confidentiality, integrity and availability. This is the point at which the solution has to address a very real business proposition, support live business critical transactions, and meet strict service level agreements (SLAs). There are many trade-offs to consider when moving into this phase, particularly if moving at pace.

Is there a blockchain solution that can do all of this out of the box? Or is it a case of "batteries not included"?

Before answering that question, let's ask another...

WHAT MAKES BLOCKCHAIN SO SPECIAL?

In our view, blockchain technology (as exemplified by Bitcoin and Ethereum) brings together the following characteristics:

- **Persistence:** Data and information are stored in a recoverable way
- **Permanence/tamper resistance:** Stored data cannot be changed or deleted without considerable effort and is cryptographically verifiable
- **Data integrity:** Data is accurate, valid, irrefutable and has cryptographically demonstrable provenance
- **Data consistency:** Wherever the data is accessed from, it is the same
- **Programmability:** Executable logic can be integrated with or directly created within the data
- **Decentralisation:** Data is not controlled by a single, central authority and is instead stored and processed on distributed, connected resources

It is this core set of characteristics that makes the application of blockchain technology politically appealing for a group of organisations that wish to transact digitally with each other. The term "blockchain" is becoming a mantra for change, even where the problem space may not need all of the characteristics.

However, when building enterprise transactional platform, there are two further key considerations:

- **Data confidentiality:** The enterprise context entails a number of operational challenges that don't currently apply to public implementations, such as Ethereum. The most striking of these is the need to preserve the confidentiality of transactional data (often referred to as "privacy" or "private transactions"). While some degree of confidentiality can be attained using encryption or zero knowledge protocols, organisations typically require stronger guarantees of data isolation, i.e. transactional data is only shared on a strictly "need to know" basis.
- **No requirement for completely trustless environment:** The complicated process of building a shared persistent store of securely verifiable data in a completely trustless environment is no longer a must. Legal agreements allow the hardened security ring to move to surrounding the blockchain, rather than being deeply baked-in. This opens the door to a wider array of mechanisms for ensuring data integrity and consistency (in other words, consensus) on the facts across nodes of the network.

In summary, blockchain (or more broadly Distributed Ledger Technology – DLT) is an implementation of technology that satisfies a set of social, political and business needs. Various companies and consortia are exploring the space, but for now the field is still maturing. As such, concerns remain when it comes to building production implementations that have blockchain at the core.

ENTERPRISE GRADE ARCHITECTURE AND BLOCKCHAIN –WHAT DOES THAT ACTUALLY MEAN?

The time has come for the IT community to get off the PoC fence and deliver an enterprise grade, fully private, politically apposite, technical solution that satisfies the true requirements.

Right now, as the maturity index increases on the emergent implementations (which could take several more years according to McKinsey) whatever choice of blockchain is made, in order to build a complete system it will need to be surrounded by tried-and-tested enterprise grade software "capabilities". These are all well-known and readily available but there will be some effort required to integrate them with the core blockchain implementation.

Let's think of this from a business point of view first of all. The product or platform needs to be able to satisfy the business service levels and to that end needs to be, to some extent, resilient, available, scalable, secure and consistent. There will be requirements for uptime, and for minimising P1 events, there will be a minimum level of performance required and an expectation of high levels of security, privacy and integrity.

In order to ensure all of this, the platform must be designed with operability in mind, to ensure SLAs are met, and that standard processes and procedures are usually enough to keep the lights on.

As an heuristic, we can apply the "3am test". When considering an architectural design, technology or approach ask the question "is there a possibility that because of this decision, at some point in the operation of the system the CTO will be woken up at 3am to deal with a production issue?".

Nothing passes through the 3am test gate if the answer is yes.

IS THERE A BLOCKCHAIN IMPLEMENTATION THAT IS AN ENTERPRISE GRADE SOLUTION (TODAY)?

We suggest that the answer to this question is: not yet. A number of enterprise blockchain initiatives are competing in this marketplace. These products follow one of two approaches to meeting the needs of an enterprise blockchain.

There are the top down initiatives, designed from the start with the enterprise in mind such as:

- Corda (R3)
- Uledger, radix

Secondly there are the bottom up approaches, trying to make an existing public blockchain solution fit for the enterprise such as:

- Quorum
- Pantheon (EEA based)
- Hyperledger Fabric
- Hyperledger Sawtooth

Corda promises a frictionless, private enterprise grade solution. Backed by banking (the R3 consortium) and focused on performance and scalability, it has promise. While it does have an open source version, many of the desirable characteristics are only available in the paid for, licensed "enterprise" version.

The blockchain community at large is generally not overly supportive of such commercialisation, yet there are positive benefits in having the security of support and maintenance fees that deserve consideration. A key distinction of CORDA from the other platforms is that it is specifically not a blockchain.

The Enterprise Ethereum Alliance (EEA) is driving the bottom up approach, and has issued a specification for enterprise blockchain that seeks to define, and ultimately answer, many enterprise and interoperability questions. While Quorum satisfies some of the standards already, we expect other reference implementations to emerge from this initiative, the first of these being Pantheon, from PegaSys (a Consensys company).

Hyperledger fabric (IBM led), and latterly sawtooth (driven by Intel), are strong attempts to bring blockchains to the enterprise. The approach here is to build around the Hyperledger community to create enterprise features. Hyperledger fabric has a pluggable architecture and a unique "channel" feature that allows private transactions.

While these initiatives are making good progress toward the goal, we suggest that none of them as yet would pass the 3am test, for two core reasons:

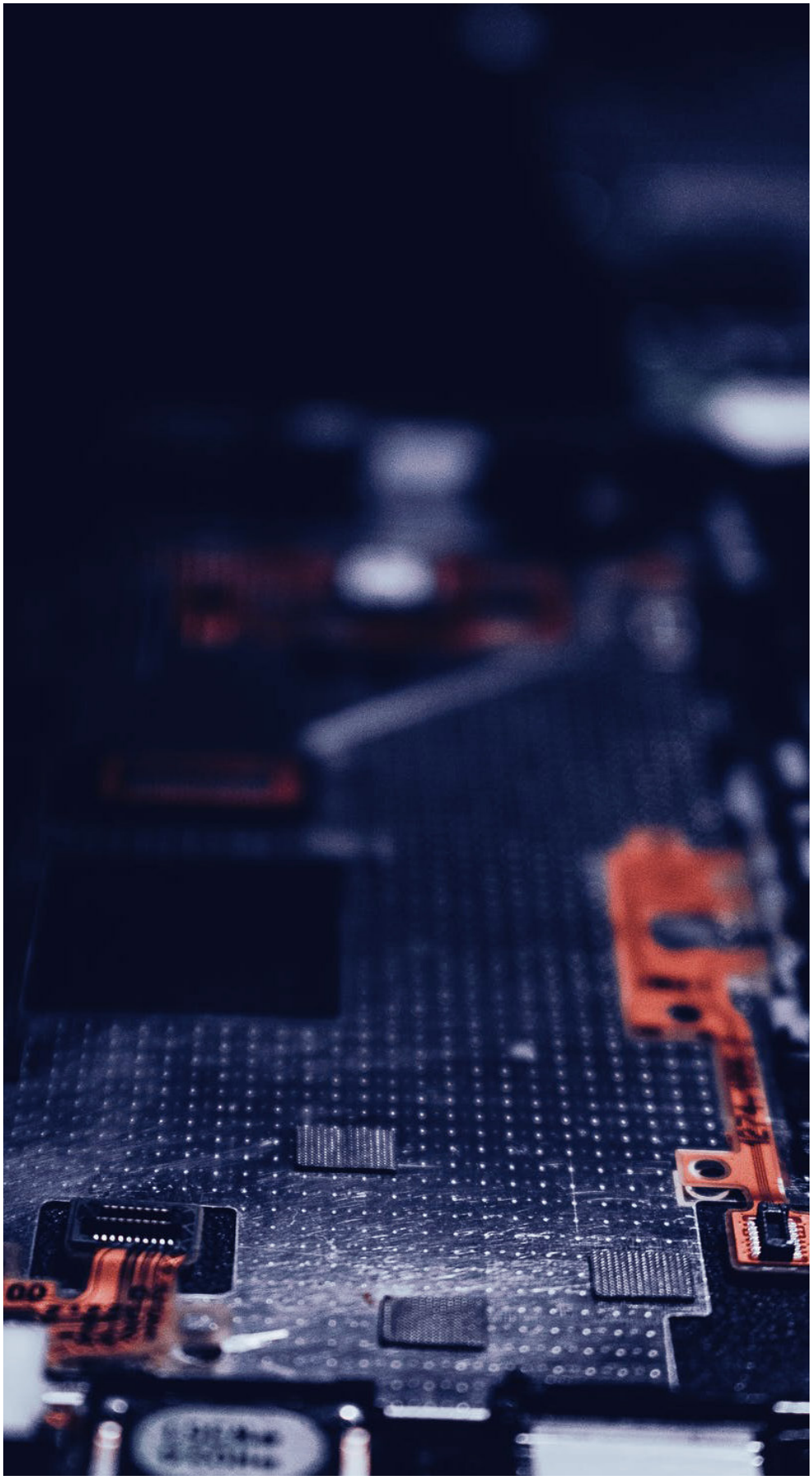
- Firstly, true blockchain implementations suffer from a number of well documented constraints around transaction throughput, scalability and privacy. While the products listed above are evolving to overcome these and the public Ethereum community is also making inroads – there is some way to go.
- Secondly, all of the products are relatively new in development (less than four years). Blockchain in particular suffers from immaturity regarding operational characteristics and resilience. Missing features such as identity and document storage also hamper 3am test performance.

BATTERIES NOT INCLUDED

Today, there is no out of the box solution for enterprise grade blockchain. Instead, it's necessary to bolt together blockchain and the best of boring standard enterprise level technologies.

As for which blockchain technology to place at the heart of the solution, there are a number of candidates, as listed above. However, it's worth remembering that the "best" solution is not necessarily architecturally best: be mindful of VHS vs Betamax. Choices made are likely to lead to the top three most adopted platforms spawning enterprise versions, or at least aspirations to create them.

For now, however, batteries are not included. So, it is worth asking: how do we go about gluing together old and new, and what is needed for it to be successful?



PART 2:

A framework for an enterprise grade platform, with the features of a distributed ledger

While it is easy to state that an enterprise grade solution built on blockchain must combine the best of the old with the best of the new, it is not easy to implement.

At VAKT, we have built such an enterprise ready platform, and can draw on our experiences to provide insight.

A current trend in blockchain is to try and create a one size fits all platform. We argue this ignores the lessons of technology history. Ultimately, tools, architectures and products have been developed that solve specific purposes (databases, File systems, ETL, MIS etc), and a model of enterprise architecture has evolved that plugs these tools together to create a best-of-breed system.

Guided by these lessons, we propose several principals and define several capabilities that we believe should be considered when building an enterprise grade architecture with blockchain at its heart. Such capabilities may not be provided out-of-the-box with any current ledger implementation, and will likely require significant development effort to realise.

KEY PRINCIPLES TO MITIGATE RISK

In our experience, the following guiding principles lead to a shortened, simplified delivery and a robust, operable solution, that focuses on the business problem while making good use of all necessary technology elements, including the ledger. We should note that these recommendations are particularly driven by our experience of working with Quorum, an Ethereum / Solidity based technology solution.

- **Minimise use of the ledger**
- **Minimise exposure to innovation**

We discuss each of them in detail overleaf.

PRINCIPLE I: MINIMAL USE OF THE LEDGER

When you have a hammer, everything looks like a nail. There is an argument that in a private, permissioned blockchain the notion of a block is unnecessary, and the requirements of consensus are minimal (both practically and theoretically). Still, the requirement for an immutable, indisputable, and cryptographically secure version of the truth points squarely to some form of distributed ledger technology for persistence.

The crucial thing here is not to use blockchain when it isn't necessary.

A number of blockchain offerings now provide frameworks for many applications, and in a B2C context, perhaps this makes sense. But in a B2B context, it is neither necessary nor desirable to put all of your eggs in one basket. In fact, it is problematic to do so.

Instead, there is a growing understanding of the kinds of processing that need to happen off-chain, and the patterns for this, to ensure that the ledger can scale and perform at the levels required for the business.

The use of smart contracts tends to polarise opinion. We lean towards the view that they should be used sparingly and only where it makes complete sense to do so. While it is possible to build an entire distributed application with smart contracts it is not necessarily desirable because of the difficulty of management of contract versions, security concerns and complexity in implementing simple business logic. We recommend that you carefully consider what logic you place in smart contracts. At VAKT we are moving more towards using smart contracts only to store state.

RESOLVING THE TRUTH VS STORING THE TRUTH

It is indisputable that the blockchain is – and should be – the single record and source of truth. Executable smart contracts allow not only the persistence and verification of that truth, but also the automatic generation of a state of truth: the resolution of truth.

While initially simple, smart contracts are now supported by full language semantics; no longer limited to replicating atomic transactions in a distributed space (such as executing a transfer of funds between accounts securely and guaranteed without an intermediate authority).

So, while it is now possible to resolve the agreement between the two parties in a smart contract, it is not necessary and it adds significant risk to an enterprise system. Smart contract language implementations such as Solidity are immature, open to security challenges (through the possibility of race conditions and re-entrant functions), difficult to test and verify, and hard to prove correctness for (that they are guaranteed to work as expected in the future). What's more, putting complex truth resolution on the ledger breaks Principle I.

Equally, the resolution of the truth often requires a shared "whiteboard" where human-to-human negotiation takes place. This whiteboard will need to be highly interactive, and can't wait for the chain to catch up. Furthermore, it is not always necessary to store intermediate state.

Our recommendation is to separate truth resolution as far as possible from the writing and management of the record of the state. This allows minimal use of smart contract language primitives, reduces the computational overhead on the ledger and allows flexibility. This is the approach increasingly taken in the enterprise ledger community, for example in the Microsoft enterprise smart contracts.

PRINCIPLE II: MINIMISE EXPOSURE TO INNOVATION

In a nutshell, Principle II is: choose boring tech.

Boring tech has been tried and tested over many years and across many implementations, and typically benefits from a large, well-established development community. This doesn't preclude open source, in fact open source is an excellent way to pull in established technology to your stack. So, consider a tech list with the likes of Java, JavaScript, databases, network file systems, cryptographic libraries, cloud infrastructure and messaging systems, for the majority of your stack.

While this is a generally applicable principle, it is particularly relevant in the case of a blockchain project. Consider a "portfolio of risk" across technology selection. You will be taking on enough innovation with the core technology, so you may want to consider keeping the rest of the architecture as boring and low-risk as possible.

It is, of course, hard to push back the general desire of developers to get deep in to the core of new technology. This is where governance comes in to play. However, we don't advocate stifling innovation. Far from it.

Innovation should be encouraged and new technology adopted judiciously while keeping a weather eye on the evolution of new solutions (especially in the fast-moving world of blockchain technology). The trick is managing a balanced portfolio of risk, particularly in relation to maintaining delivery momentum.

ISOLATION AND COMPONENTISATION

While it is vital to minimise exposure to innovation, it is also important not to preclude the potential to benefit from it.

In order to make the implementation as evolvable as possible, pioneers building enterprise grade platforms should consider ways to avoid lock-in. New technology will emerge, and it would be short-sighted to build a system that can't benefit from that.

It is generally well understood that defining interfaces and contracts, and building components that plug together is good enterprise architecture practice (except in extremely low latency situations, which is definitely not the case with a blockchain solution). In the case of ledger technology, it is especially important to minimise the use of complex primitives and where possible reduce the ledger surface.

The ledger debate has only just started, and while it mirrors the VHS/ Betamax competition in the 1980's it is important to remember two things from that example. First, that arguably the inferior technology won. Second, that everyone with a Betamax recorder had to throw it away.

In the case of the ledger, we are all making bets on the winner (or perhaps winners), and all hoping we are right, but since we don't want to throw away the baby with the bath water, and since we want to exploit new developments and evolutions as they emerge, isolation seems to be the most practical approach.

Our experience also tells us that whilst the ledger is a critical part of the technology landscape, the vast majority of code is concerned with building business features and, of course, infrastructure. It is important not to forget that 80 per cent of your effort will be "business as usual" enterprise development, with a decentralised twist.

SURROUNDING THE LEDGER WITH ADDITIONAL CAPABILITIES

If we assume that there is business benefit in placing a distributed ledger at the core of our architecture, there remain various necessary capabilities which are unlikely to be available "out of the box". As the market and products evolve, no doubt they will expand to provide these, or at least require minimal effort to customise (For example the EEA includes many of these capabilities in their vision). However, as things stand today, it is likely that these capabilities will require significant investment beyond simply running a few nodes and writing some smart contracts.

Different implementations will cover these capabilities to varying degrees and satisfy them with different architectural designs (for example, the Ledger Index will require more bespoke effort with technologies based on Ethereum), but from the perspective of requirements these capabilities provide a model to be able to compare and assess the platforms.

Capability	Description
Identity	Schemes for mapping corporate identities into the platform so that transactions can be signed and attributed to individuals.
Transaction privacy	Ability to control access to and distribution of transactions and related data on a "need to know" basis amongst participants.
Security	Models of authorisation with appropriate roles and permissions
Integration	Integrations through APIs with various third-party systems to expand the ecosystem [blockchain and non-blockchain].
Ledger robustness	The ledger is core to the platform, and current implementations are maturing but still have a way to go.
Document sharing	Mechanisms of storing documents [or blobs more generically] that are to be shared between one or many, but not necessarily all, participants in the platform.
Reference data	Schemes for making common reference data available to all participants.
Ledger index	Ability to query the data in the ledger effectively and quickly, for example in Ethereum based systems this requires construction of a separate data store that allows the underlying ledger transactions to be queried.
Platform robustness	There are a number of core components that ensure platform robustness: • **Availability** - The design of the system should account for standard redundancy and recovery approaches to ensure high levels of availability • **Operability** - Elements of the system must be manageable by participant technology teams, should conform to standard • **Deployment** - Models of deployment both on cloud and on premise, that are driven by automation **Tooling** - Tools for monitoring and remediating issues, as well as for building the deployment pipeline to ensure regular and seamless patching and updates
What to logic to encode in smart contracts?	Is your ledger used as a decentralised database/persistence layer only, or is some significant business logic encoded in smart contracts?
What to decentralise?	What are the consequences/impacts of decentralisation? A good example of logic to decentralise would be a voting mechanism.
Consensus model	There are some specific considerations in the applications of consensus mechanisms in private transactions. For example, how do you solve the double spend problem in a transaction that only has two participants – and do you need to at all? The raft consensus algorithm weakens the decentralisation as there are no checks and balances preventing corrupt nodes.
Immutability and tamper evidence.	While immutability is considered a desirable property of a blockchain, the fact that data (and code) can never be deleted has consequences all of its own. Consider the problems of immutability with a data retention policy that requires data to be deleted after a certain number of years. Equally problematic is the immutability of a smart contract, which can never be evolved or removed – instead it must be deprecated and replaced. In general, tamper evidence takes precedence over immutability in a business context. And before committing anything to the ledger it is important to consider how much immutability you really need – or indeed want?

IN CONCLUSION

Building an enterprise grade solution is not easy. A platform underpinned by a nascent but revolutionary technology like blockchain is difficult and pioneering.

But there are lessons to be learned from many years of distributed computing and enterprise thinking. Not everything that is needed yet exists and the ecosystem is evolving rapidly, so where we have no solution today one might appear in the near future. Until then, new technology will need to be built to solve specific problems.

It is time for the training wheels to come off and to move from PoC to production.

Our hope is that the framework outlined above provides a lens through which to view the problem. If you can crack each piece and the interconnectedness of them all, then you will have an enterprise grade platform which, running in a full production environment, exploits the ledger to not only solve business problems but create new ways of thinking and operating in business altogether.

If you have any questions, please do contact the authors:

ThoughtWorks®

Jim Barritt

Shodhan Sheth

info-uk@thoughtworks.com

VAKT

Adam Vile

Enquiries@vakt.io