2018 IEEE European Symposium on Security and Privacy Workshops

Blockchain as a Service (BaaS): Providers and Trust

Jatinder Singh
Department of Computer Science & Technology
University of Cambridge, UK
jatinder.singh@cl.cam.ac.uk

Johan David Michels
Centre for Commercial Law Studies
Queen Mary University of London, UK
d.michels@qmul.ac.uk

***Abstract*—Distributed ledger technologies (DLTs) are receiving much attention. As discussion focuses on the potential applications of DLTs, Blockchain-as-a-Service (BaaS) offerings are emerging to provide the underlying supporting infrastructure. BaaS entails a service provider supplying and managing aspects of a DLT infrastructure to facilitate and bring efficiencies regarding the development, experimentation, deployment, and the ongoing management of DLT applications. However, much of the interest in DLTs stems from their potential to decentralise, disintermediate, and enable 'trustless' interactions. At first sight, BaaS – being offered by a provider – appears to run counter to this. In practice, whether BaaS raises substantive trust concerns depends on the nature of the offering, the application's specifics, and the participants' goals and risk appetite. This paper elaborates the nature of BaaS and explores the trust considerations it raises, particularly regarding the role of providers as part of a wider infrastructure.**

I. Introduction

Blockchain and *distributed ledgers* (DL)[1] are the subject of much hype. Much of the interest is because ledgers that are visible to, and operate across, various parties have the potential to unlock a whole new range of applications [1].

Despite the technology being nascent – to date, there are few mature applications of substantial scale, the most prominent being Bitcoin – there is growing commercial interest in DLs, where claimed benefits include offering value and efficiency gains by, for example, assisting compliance, asset tracking, supply chain management, and generally displacing intermediaries [1]. The focus is particularly on multi-party scenarios (across organisations, departments, individuals, etc.), where the ledger provides a transparent and reliable source of facts across administrative domains [2].

In line with this, *"Blockchain as a Service*" (BaaS) offerings are emerging to make DLT more accessible, particularly for businesses, by reducing the costs and overheads of adoption. In essence, BaaS entails a service provider offering and managing various components of a DLT infrastructure. The precise nature of a BaaS deployment depends on the service provider, application specifics, and the customer goals.

Much of the early discussion of distributed ledger technologies (DLT) focused on the benefits of disintermediation and the operation of so-called 'trustless' interactions, where users rely on a network to maintain an accurate ledger, instead of relying on a trusted intermediary or the specific counterparty to a transaction [4]. However, an interesting characteristic of BaaS is that it reintroduces an intermediary in the form of a service provider – who often has a relationship (only) with certain participants in the network – that provides integral parts of the system. This 'recentralisation' introduces new trust considerations as they relate to the provider.

This paper explores the trust considerations of BaaS, particularly with respect to the role of service providers. As BaaS is a new and emerging trend, our aim is to highlight its trust implications to inform research into DLTs, particularly in areas of security, privacy and trust, as well as to indicate some practical architectural and governance considerations for potential users and providers of BaaS-driven systems

A. Relationship with Cloud

BaaS can be considered a form of cloud computing, and therefore offers benefits similar to those of cloud in general [5]. That is, the service provider leverages economies of scale to provide managed compute, storage, network, security and other services that can be used by customers on demand. This can result in significant capital and recurrent cost savings for adopters. Cloud also supports elasticity, dynamically able to handle increases/decreases in computational loads as and when required. This can be relevant in a DL context, for instance, to deal with a large number of events triggering a range of smart contracts, or providing the infrastructure to ensure some quality of service, e.g. reducing latency that might impact a particular consensus protocol. Further, DLs aim at interactions across multiple parties, which cloud infrastructure can help manage. The major cloud providers have a global footprint with regional access points, to assist in maintaining a certain quality of service.

Cloud also enables the outsourcing of skill and expertise with regard to technology deployment and management. Blockchain is an emerging technology, meaning that expertise

[1] As terminology in this space varies, we use the term 'blockchain' as synonymous with *distributed ledger technologies* (DLTs), to align with common industrial usage. We consider DL as comprising a ledger consisting of a chain of immutable blocks that record various facts/interactions; that has some degree of distribution/visibility amongst participants; where defined processes govern its operation, in particular regarding consensus on the addition of ledger entries [3]. We do not imply any specific incantation, e.g. consensus protocol, systems architecture, etc., as the specifics will vary depending on the particular application or service.

DOI 10.1109/EuroSPW.2018.00015

IEEE computer society

in the area is limited, and in high demand. As such, BaaS can facilitate technology access, providing abstractions over the lower-level technical details. Currently a key BaaS marketing focus is on providing the environment for rapid experimentation with DLTs, to support businesses in their desire to explore the technology's potential. Beyond experimentation, at present most BaaS applications appear to be specific applications for closed environments (see §III).

Similarly, the considerations and trade-offs concerning the adoption of cloud services are also relevant in a BaaS context. From security and governance perspectives, businesses relying on a BaaS solution may have less control over the application than when using their own in-house infrastructure. On the other hand, businesses benefit from an established service provider's 'best-in-class' security and resilience solutions, including identity management services. Of course, these considerations depend on the nature of the particular deployment.

B. *Movers in the Space*

Blockchain has the attention of the major IT firms. Regarding DLT-driven applications, Amazon has partnered with the Digital Currency Group (DCG) to explore an enterprise-focused blockchain experimentation environment, while Google has invested in various blockchain-related start-ups to explore particular applications of DLT.

In terms of BaaS, IBM and Microsoft appear to be leading the charge, already offering blockchain infrastructure services. There are blockchain infrastructure offerings on Amazon's AWS marketplace – an 'app store' for cloud software – but these are produced by third party developers rather than dedicated infrastructure offerings of the provider. It is also worth noting that the major audit firms – including PWC (DeNovo) and Deloitte (Rubix) – are promoting DLT-enabled solutions, driven by the propensity for blockchain to disrupt the audit industry.

Since BaaS is a new and rapidly emerging, it is too early to estimate the scale of adoption or forecast a potential market size. Nonetheless, given that estimates suggest that the blockchain market will grow from USD 411.5 m in 2017 to USD 7.7 bn by 2022 [6], BaaS may prove a sizeable subset of this large and growing market, especially as it aims reduce the uptake barriers of the technology. In terms of sectoral adoption, Microsoft Azure's blockchain services uptake data indicates that banking and capital markets will be the most likely early adopter, followed by government, insurance and consumer goods [7].

Note that our discussion focuses on BaaS as it relates to more traditional cloud offerings, where a provider offers a managed infrastructure. However, it is important to note that platforms such as Ethereum could be considered a provider of a decentralised BaaS. The Ethereum blockchain supports smart contracts and has standardised guidelines for creating new crypto-currency tokens. As a result, companies can launch their own crypto-currency application, using the public Ethereum infrastructure. In other words, the Ethereum public chain could be seen as offering a bare-bones, self-service platform, where Ethereum 'tenants' use smart contracts to assign computational tasks to the public Ethereum distributed infrastructure. Though some similar considerations might apply to such platforms, our focus here is on the security and trust considerations of a managed infrastructure, entailing some degree of 'recentralisaton', and involving a contractual service-provider relationship.

II. Applications and Platforms

A. *Classification of BaaS Types*

BaaS has various service models, similar to those of cloud [8]. First, a provider (directly, or with partners) works with client enterprise(s) to develop and tailor an application that leverages a blockchain infrastructure. This resembles a *Software as a Service* (SaaS) situation, where the applications and interfaces are provided over a managed DLT technical stack. Currently, this 'application-oriented' approach seems the dominant means of adoption. This may reflect the relative infancy of DLT, its perceived complexity, general low levels of understanding, and a scarcity of experts.

The other main BaaS approach is closer to *Platform as a Service* (PaaS), whereby tenants themselves select, use, integrate and customise components of a provider's managed infrastructure according to their needs. In a BaaS context, this entails providing tenants with the ability to deal with the technical specifics, such as selecting and managing the chain, defining what the ledger records, the consensus protocol, configuring the access/management regime (open-closed-permissioned), and so forth. In future, opportunities may also emerge for DLT-dedicated *Infrastructure as a Service* (IaaS) offerings, which could entail, for instance, renting out specific blockchain-tailored hardware (GPUs/ASICS [9]) for proof-of-work consensus operations.

Note that tenants can also leverage general cloud services for running and managing DLT-related activities. For instance, a tenant could use a traditional IaaS offering to run particular nodes or perhaps even an entire blockchain network, cloud storage services to host ledgers, and identity management services to enable access control. Our discussion here focuses on BaaS-oriented (provider-branded) services, rather than on using more general cloud services as infrastructure to support DL applications, though some similar considerations of trust may apply.

BaaS typically entails providing modular infrastructure and tools, including identity management services and middleware, to facilitate the building of applications and/or the integration of legacy systems. (Work is ongoing on interoperability and standards between different chains/ledgers [10]). However, since DL infrastructures are built as open platforms, not all components forming the DL infrastructure need to be exclusively provided by, or executed on, the provider's platform. Instead, an approach similar to hybrid cloud can be taken, where some components – such as certain nodes in a consensus protocol, the storage of some ledger replicas, specific contracts, and even other chains – have the potential to operate externally, for example, on a firm's "in-house" services, or even on a different BaaS provider. Indeed, it may be the case that a participant in a DL application uses a BaaS provider purely to run the participant's own node(s), which interact with the larger DL network (see §IV).

B. BaaS Platforms

In terms of the platforms underlying the two most established BaaS providers, IBM's BaaS is based on the outputs of the Hyperledger Consortium, an open-source initiative started by the Linux Foundation to develop business-oriented blockchain, based on a custom codebase governed by the Hyperledger Consortium. Microsoft's vision is to support various protocols. Initially it has shown some alignment with the Ethereum platform, being a founding member and on the rotating board of the Ethereum Enterprise Alliance (EEA). Both the Hyperledger and EEA consortia have many firms as members, across a range of industries. Hyperledger focuses on permissioned chains without a crypto-currency basis (tokens), whereas the EEA aims to build on and adapt Ethereum (which includes Ether, a token/bearer-asset) to address business needs such as permission management. Note also that the Hyperledger codebase is governed by its Consortium, whereas the EEA builds on Ethereum which is governed by the Ethereum Foundation, a separate organisation. It is said that Hyperledger and the EEA are not competitive initiatives [11], but start from different foundations and levels of openness, and that both have similar goals in addressing business needs (including standards), which may ultimately converge [12].

As we explore below, the design of a BaaS platform (and its constituent components), including how it is formulated and how it is governed, has direct trust implications.

III. Management and Governance

Trust considerations are closely related to questions of control over the DLT. This section highlights two key issues of control, namely: who can access the DLT application and who has the power to determine the nature of the application and infrastructure.

A. Private & Permissioned Chains: Opportunities and Need?

One aspect of governance in the BaaS context concerns access: i.e. who may participate. Many of the application examples for BaaS involve blockchains that are private, i.e. a 'separate' blockchain dedicated to a particular application/use, and/or permissioned, meaning that the participants and their actions are limited and governed.

The current interest in this comparatively closed use of DLTs reflects the enterprise focus of BaaS, as firms will seek to use the technology for specific business purposes, often with their pre-established business networks. This requires some mediation or assurance regarding the possible participants, (and may also serve to reduce the performance overheads of achieving consensus, where, for example, proof of work may not be necessary given the pre-existing trust relationships). For instance, an organisation might maintain a blockchain for asset tracking between departments, a banking consortium may have a closed membership, and a supply chain may only be open to pre-vetted participants with which commercial arrangements have been predefined. It follows that identity and confidentiality aspects are core components of BaaS offerings, e.g. with providers offering identity management services and where digital signatures are used to control the visibility of particular transactions [13].

However, these more closed DL formulations make it worth considering whether a closed/permissioned ledger differs from a (secure append-only) database. One can envisage similar functionality through a standard cloud-hosted application (or SaaS), with a suitable access control and identity regime, that records specific actions of those involved in an appropriate "secure" database. With software there are often several ways of realising similar functionality, and therefore it may be that some of the desired outcomes can be realised through other means, without a blockchain-based technical stack. This raises the question: *what does the use of blockchain add in these more closed contexts, over and above existing technology?*

The answer will depend on the scenario. In some situations there may be little need for DLTs, particularly where operations are more centralised or where certain parties maintain more administrative control/power. It may be that a database suffices; audit of the database's transaction log gives some level of assurance. If parties require more assurance, immutability and integrity constraints can be implemented, e.g. there is work on secure audit [14], forward integrity [15] in append-only data stores, which may suffice for simple audit and tracking applications.

The case for using DLTs is stronger where more parties are involved, e.g. in a large business network. DLs are useful for situations where parties have some need to oversee and verify the actions of others (e.g. competing interests in a marketplace), or where parties have (or require) a degree of autonomy, such as being able to define their own interests and functionality, e.g. by specifying smart contracts. In short, the appropriateness of DLT depends on the balance of power, trust concerns, and risk appetite of the participants involved.

There are business opportunities for service providers in supporting DLT infrastructures and in offering more traditional cloud services. In practice, we see that the major providers offer both, with a strong overlap between BaaS components and their more general cloud service offerings; for example, identity management services might service both BaaS and more traditional applications.

The relative infancy of DLT may underlie the focus on more 'closed' environments. As the nature of the technology becomes more widely experimented with and understood, there may be movements towards more open uses of the technology. Though the current focus is on B2B (business-to-business) applications, it may be that the more B2C (business-to-consumer – or perhaps "consumer-to-consumer" in a sharing-economy context) use-cases lend themselves to the more open/permissionless uses of DLTs. This would likely entail consumer-oriented BaaS opportunities, not unlike an email hosting service, e.g. where providers manage nodes on behalf of consumers that seek to participate in a particular (peer-to-peer) marketplace.

B. Systems Governance

Another governance aspect concerns who determines how the DL system works. This is important given that system design decisions have direct security and trust implications. BaaS involves multiple parties, including the BaaS provider and various tenants/users, each with their own interests and incentives. To facilitate cooperation, the parties need mechanisms for coordinating their actions. DLT provides such a mechanism in respect of parties' on-chain transactions by enforcing the rules of its protocol; for instance, Bitcoin has technical mechanisms to enforce the validity of transactions, e.g. such that no user can spend the same coin twice.

However, typically it is not the DL itself that determines the nature of the system, including what those rules should be or how the rules should be changed, nor does it establish whether there should be derogations from the rules in exceptional circumstances. To settle such matters, participants need to reach agreement on the 'rules' of the system and how the system behaves. Such agreement occurs 'out-of-band' or 'off-chain' – i.e. outside the ledger itself. The process for deciding on such issues is commonly referred to as 'blockchain governance'.

In practice, different blockchain applications rely on different types of governance mechanisms to resolve such issues. These include informal, community-driven mechanisms based mainly on off-chain communications; formal legal/contractual mechanisms; and technological governance mechanisms (on- or off-chain).

1) Crypto-currency Governance

Naturally, much of the blockchain governance discussion concerns crypto-currency, where recent events suggest the two major sources of contention relate to: (i) changing the software/protocol, e.g. the debates surrounding Bitcoin's block size; and (ii) changing the ledger's record of past transactions, e.g. as a response to the Ethereum DAO hack.

Bitcoin and Ethereum rely on several 'off-chain' governance mechanisms to determine design decisions and the direction the platforms take. Examples include developers debating Bitcoin and Ethereum improvement protocols, Ethereum users voting on possible responses to issues (e.g. leading to the post-DAO-hack hard fork), miners imposing soft forks, and, ultimately, hard forks resulting in a new, competing crypto-currency, in which case nodes and intermediaries (such as exchanges and wallet providers) need to decide which version to support [16].

Some authors have criticised the existing crypto-currency governance mechanisms as informal and opaque. One solution would be to implement traditional governance mechanisms similar to those imposed on companies under corporate law. This could include imposing transparency requirements and fiduciary duties on developers and giving users formal voting rights under a proposed 'Crypto-currency Governance Code' [17]. Several start-ups are instead proposing DLT applications with built-in, technical governance mechanisms; for instance through means for voting on changes to the software and/or changes to past blocks (examples include Tezos and Dfinity).

2) BaaS and Governance

BaaS raises similar governance issues, i.e. concerning the nature of the software, its configuration, and whether ledger amendments (forks) are allowed; however, in a BaaS context the parties involved are often more clearly identifiable, facilitating their coordination. Given the business-oriented nature of BaaS, governance issues will be a primary concern. A key factor is who controls the configuration of the infrastructure. In an application-oriented approach (similar to SaaS, described in §II), the BaaS provider tends to have control over the configuration of the DLT infrastructure. Conversely, in cases that more closely resemble PaaS, BaaS tenants generally have more control over the configuration of the DLT infrastructure. In cases of an IaaS-like service, or where generic (non-BaaS oriented) cloud services are used for DLT purposes, the tenant and other participants are likely to have much greater control over the system and its configuration and can decide governance issues amongst themselves.

Businesses using BaaS should arrange contractual terms with the BaaS provider and any other participants that cover governance issues, where possible. (In many cases, BaaS providers may offer commoditised services on standard terms). Such terms could cover, inter alia, (i) whether the BaaS provider can change the software unilaterally and (ii) whether the provider's software licence allows a client to fork its software and port the existing ledger. A further issue is whether a ledger is 'completely immutable' or whether there are circumstances under which the provider, tenants or other participants can (collectively or by majority vote) decide to override the ledger, e.g. forking the chain to deal with inappropriate transactions. Alternatively, there may be means to build in governance mechanisms – such as voting systems – into the technical infrastructure itself.

Another important consideration is whether the provider is offering a DL that is run on another blockchain (similar to layered cloud) that is governed by another community, e.g. as we see with "Ethereum-as-a-Service" offerings that could be affected by Ethereum's community governance mechanisms. Indeed, any layering of technology and services, as is common in cloud [18], can complicate issues of governance and risk assessments. Generally speaking, governance issues should be simpler to resolve where fewer parties are involved.

IV. Clouds of Suspicion: Trust Considerations

Much of the interest in DLs stems from their capacity to decentralise and disintermediate, removing the need for trusted third-parties. That is, in many cases it is the decentralised nature of DLs that bring security, resilience, and data integrity considerations. BaaS, however, involves introducing a provider to supply and/or manage (aspects of) DLT infrastructure. This may entail re-centralising aspects of the DL.

It follows that BaaS brings considerations regarding security and trust. In practice, whether BaaS raises significant concerns depends on the particulars of the service, the application's risk and threat profile, and the purpose of the DL. The following subsections indicate some of these considerations as they relate to tenancy, the role of service providers, emerging trust technologies and external ("out of cloud") interactions.

A. *Trust in the Tenant(s)*

Traditionally cloud services involve a contract between a (legal) person – the tenant – and a provider. The tenant selects and pays for the services consumed, and may have options to customise and configure the services.

The notion of 'tenancy' can bring complications in a DLT context. DLT applications typically aim at dealing with the trust and transparency concerns regarding the other participants in the application. Tenancy in a BaaS context can mean that some participants – through their arrangements with service providers – have more power to control the infrastructure than others. This raises questions in DL situations, where different entities interact through a common infrastructure, as to the trustworthiness of those who can configure, control (and indeed, who pay for) the provider's service. This issue is of particular importance if there is only a single tenant that is responsible for interacting with the provider, meaning that it can act unilaterally (at least at a technical level - contractual considerations with other participants aside).

Tenancy issues will vary depending on the situation and the nature of the chain: this is (i) likely to be a non-issue in a single-organisation's private blockchain, (ii) not much of an issue in a predefined consortium (where the rules/membership are defined and such issues can be dealt with through prior agreement), but (iii) more of a concern in more open environments if the power to manage the service is overly concentrated. In terms of architecture, it may be more straightforward if each participant manages their node(s) through their own BaaS tenancy arrangements (see §IV(B)) - or on their own infrastructure. There appear to be R&D opportunities for mechanisms that better enable multi-party control regimes, and tooling to better support and facilitate the deployment and management of these different architectures.

B. *Trust in the Provider(s)*

Clearly the trustworthiness of a BaaS provider, who supplies and manages aspects of the supporting DLT infrastructure, is also an important consideration. The nature of the infrastructure underpinning a ledger determines its functionality and integrity.

The established cloud providers rely on reputation, so it is reasonable to assume that providers will act honestly, and also invest heavily to secure their infrastructure. However, trust and transparency remain key concerns. Though we have seen a massive uptake of public cloud services for a range of business processes, trust concerns remain, for instance where data is particularly valuable or in more highly-regulated industries such as finance or in healthcare [19]. However, trust is a particularly pertinent consideration in a BaaS context, not least as much of the interest in DLT relates to issues of trust, and following on from §IV(A), there are extra considerations regarding BaaS that relate to the provider's business arrangements; i.e. stemming from the entity who holds the contract with the provider (or who pays!), and those who can manage and configure the hosted services.

Again, whether provider trust for BaaS services is a concern depends on the purpose of the DLT, and the application's risk and threat profile. As mentioned, a private, single-organisation chain for managing some internal process differs from a multi-stakeholder environment involving a wide range of possibly competing parties, where the provider-managed BaaS infrastructure maintains the processes around the ledger, which acts as the source of truth.

The Provider's Role

A key factor regarding provider trust is the role the provider plays as part of the broader system architecture.

Intuitively, trust is a greater concern where there is a dependence on the provider to 'run the entire world', in the sense that all aspects of the DL are provided and administered by a single provider, who manages all the nodes (i.e. holding all copies of the ledger), executes the consensus processes and smart contracts, and also manages the identity aspects (perhaps holding the keys) of the parties involved. This applies most strongly in the case of an SaaS-type service, where the provider controls the applications and interfaces (possibly in collaboration with other service providers [20]), with the exception of limited user-specific application configuration settings. Conversely, in a PaaS-like settings, or indeed situations where more generic cloud services are used to support DL applications, the tenant has more control over the deployed applications and possibly configuration settings and can select, use, integrate and customise components according to their security and privacy needs.

In other words, centralising the DL infrastructure appears to undermine the trust mechanisms that DLT aims to address. Increasing levels of operational visibility may mitigate such concerns, for instance, if the nodes/ledgers are all hosted by one provider, then participants could periodically check that the entry hashes of the provider's chain remain consistent over time, e.g. validating that the hashes of the blocks accord, and by caching some hashes locally for comparison with the provider's ledger, to ensure entries have not changed by way of the chain being surreptitiously "reinvented" (however unlikely this may be).

Naturally, more federated architectures lessen these concerns. This is where, for example, components are run across different tenant accounts (i.e. managed by different stakeholders), run across different providers, with some on private infrastructure. Federated architectures are analogous to the well-established concept of hybrid cloud, which given the modular nature of BaaS platforms, means that such approaches are readily supported in a cloud context. However, federating components brings considerations (benefits and risks) regarding security, performance and resilience.

In short, BaaS infrastructure-related trust issues involve architectural design, operational oversight and validation considerations.

C. *Silicon-based Trust*

There are generally strong incentives for providers to implement means for improving security and raising levels of trust in their platforms. This is because increased trust helps encourage greater service uptake, while reducing the need for firms to host 'in-house' infrastructure. Advances in hardware-backed trusted execution environments (TEEs) [21] may offer a means for providers to achieve increased trust, by effectively working to remove the provider from the 'chain of trust'. These technologies aim to provide a more trustworthy environment (often termed an "enclave") for isolated code execution, secure data storage (encrypted memory) and remote attestation (e.g., assertions about configuration and code that was executed). Some prominent examples include Intel's Software Guard eXtensions (SGX), ARM's TrustZone, the upcoming AMD Secure Encrypted Virtualisation/Memory, and the ongoing work on CHERI – an open-source approach.

Enclaves are relevant to DLTs; common enclave use cases include key and password management and validation. Enclaves offer much potential for BaaS, as the main service providers recognise. Key management is important, but there are also other DLT use cases. For instance, a smart contract's code might be executed in a cloud-hosted enclave providing guarantees that it was properly executed. This might remove the need, for instance in a standard Ethereum-like blockchain, to expose the smart contract code to those on the chain, bringing advantages where the contract contains commercially sensitive details. (Privacy-preserving smart-contracts is a topic of research [22]). In sum, enclaves can be used to raise levels of trust in higher-risk application scenarios, including more open environments, as well as in the BaaS infrastructure itself.

Of course, no mechanism can guarantee absolute security, and enclaves represent another tool complementing a provider's other offerings. TEE technology is still maturing, and issues have already been identified; for instance, side-channel attacks have been demonstrated to extract keys in SGX [23]. Further, enclaves are considered a rather 'heavyweight' approach, requiring special hardware and application design. There are also the more general concerns; Meltdown and Spectre being recent illustrations of wide-scale hardware-based security vulnerabilities [24], and given that enclaves are built to handle sensitive data, does one completely trust the hardware chip manufacturer?

That said, there appears to be a clear role for enclave technology to greatly raise levels of trust one places in cloud providers, both to protect and provide guarantees relating to tenant code and data, and for other security/management mechanisms, such as audit logs. However, this raises a more general question: *if the provider, with its technical security infrastructure, is considered trustworthy, is there still a need for BaaS, or indeed, DLTs more generally?* That is, perhaps a more traditional cloud-hosted application/service architecture (as previously discussed), where the critical underpinnings – such as a transaction processing or logging mechanism – are backed by enclaves and other security mechanisms, could provide the appropriate levels of assurance to support multi-party interactions in environments of mutual distrust. In practice, it will depend on the situation; though in a BaaS context, there is a clear role for secure enclave technology to complement other security, privacy, management and trust mechanisms – areas for future work.

D. *The Outside World*

Related trust and security considerations regard how external interactions, i.e. between the provider-managed service and the outside world, are managed.

Naturally, cloud providers maintain access control mechanisms to govern external interactions; i.e. to ensure that tenants are properly authenticated before interacting with the provider, and that their actions are authorised. In a BaaS context, access controls might relate to chain-related identity (key management/wallet) services, and will govern one's ability to manage the BaaS components. Access controls will also be in place at the 'chain-level', to ensure the proper identification and participation of the transacting entities, some of whom may run on external infrastructure (again, akin to hybrid cloud). There may well be scope for BaaS to drive new access control regimes, for example as a means for managing multi-party governance controls. Also relevant are the means for interoperability, where various components of a DLT infrastructure (i.e. nodes maintaining ledgers, part of the consensus mechanism), can be distributed and federated across a range of differently owned and managed infrastructure elements. BaaS platforms are designed to be modular, to facilitate integration with applications and legacy systems, though there is scope for more work on standards and mechanisms for enabling interoperability.

Another aspect is how to validate the data that is external to the DLT. If a smart contract exists, for instance, to transfer funds automatically on the arrival of a boat in a port, or to pay out an insurance premium in the event of a storm, how can one be sure if and when these events actually occur? In a DLT-context, these concerns are managed by way of entities, termed oracles, which operate to provide the interface to the outside world (potentially including other chains), and attest (sign) the veracity of the data. Oracles will play an important role in BaaS infrastructure, and have the potential to operate across a range of different applications, e.g. many financial trading applications rely on stock quotes.

It follows that trust in, and the security of, the oracles is also important, particularly as the world becomes increasingly instrumented with the emerging Internet of Things. The simplest approach is to have specified oracles as definitive 'sources of truth', though this may only be suitable for certain scenarios – e.g. a market's official stock prices feed. Another approach is to have a number of oracles whereby the data is validated by consensus between them. This requires the data to be generally (or publicly) accessible, so that all the oracles, which might be under different administrative regimes, have the ability to 'see' the data in order to attest to it. In more closed environments, where access to a sensor stream is restricted – e.g. a car or phone's GPS, both of which collect highly personal data – other approaches are needed. There is work on increasing levels of trust regarding oracles, for example through hardware infrastructures (including enclaves) that couple with data sources to attest their data feeds [25] (or potentially certain happenings through computation over these feeds, without revealing the data items/readings themselves), while preventing tampering. Moving forward, we expect to see oracles increasingly dealing with complex/composite event patterns [26], i.e. considering the combination of events, possibly from across different sources.

V. Clear Skies Ahead

It is possible to characterise DLTs as distributed computing environments, where storage and computation happen through the network of participating nodes. As such, there is potential for DLT applications to displace the more traditional (and centralised) cloud computing services.

There are already start-ups positioning themselves in this space. For instance, Sia and Storj offer DLT-enabled storage infrastructure, i.e. Dropbox alternatives, whereby files are encrypted, fragmented, with the fragments (and duplicates) distributed and stored throughout the network. Files are accessed on demand, where segments are reassembled. Payments for storage services are through 'coins' native to the service, which enables a marketplace – in which participants who share their unused space are rewarded. There are also services that aim at distributed computation, where code is executed on the machines of other participants in the network. Examples of start-ups include iEx.ec, and Golem, which describes itself as the "AirBnB for computing". The business models effectively entail participants renting their processing cycles, where smart contracts and virtualisation environments are used to manage code execution. Again, payments are managed through tokens (coins) specific to the network.

Platforms for distributed, P2P computation are not new. However, the potential for DLT-driven services to challenge more traditional cloud services is not just that they are decentralised, but also because DLTs enable better transparency, and are different from previous P2P computation systems, intrinsically providing the means (coins/tokens) to transact and create a marketplace.

Large bets are being made – the market-capitalisation for the coins underpinning these services can be in the hundreds of millions of dollars, though clearly these numbers are inflated by speculators. Indeed, one issue with token/coin-based services is the extreme price volatility, which could pose barriers for serious adoption. Another consideration is whether the specifics of the application/service are such that end-users are willing to place their trust in the network, as opposed to a managed solution provided by a centralised, reputable (or at the very least, readily-identifable) party that can be held to account when necessary. That said, there appears to be a real opportunity for blockchain-based approaches to displace more traditional cloud services [27], particularly as cloud computing evolves to include smaller, decentralised and federated services to support the emerging Internet of Things [28].

VI. Conclusion

BaaS offerings are emerging in response to the significant attention given to DLTs. Given that BaaS introduces service providers into the mix, at first glance, it appears to run counter to the popular discussion of DLTs as enabling decentralisation and 'trustlessness'. Yet in practice, the appropriateness of using BaaS will depend on the specifics of the application, its operational context, and where the perceived trust and security concerns and risks lie. As we have set out, relevant considerations in this regard include the specifics of the system's architecture and the nature of the hosted infrastructure, including the degree to which components comprising the wider DLT system are federated. These factors must not only be considered by potential users and providers of BaaS, but should also feature as part of the on-going research into DLT, particularly in the areas of security, privacy and trust.

BaaS is still in its infancy. At present, it appears to be a test-bed for organisations to experiment with the technology, or a means to support more permissioned/closed applications. It is clear that BaaS aims to facilitate and simplify access to DL technology. Whether BaaS will play a significant role in more open/public applications is not yet clear. That said, even in highly federated architectures, provider-managed DL services may well offer benefits in terms of improving access, and in managing security, performance, and scale.

It also remains to be seen how BaaS will fit with other advances in trust-related computing technologies, which in some cases might work to enhance BaaS offerings, and in others, perhaps displace the need for DLT altogether. There may also be real potential for DLTs to shape the future direction of cloud computing, by encouraging more decentralised, P2P infrastructures. We shall see…

Acknowledgements

We acknowledge the support of Microsoft (through the Microsoft Cloud Computing Research Centre) and Engineering and Physical Sciences Research Council UK (EP/P024394/1).

References

[1] UK Government Office for Science, "Distributed Ledger Technology: beyond block chain: A report by the UK Government Chief Scientific Adviser",(https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf).

[2] A. Deshpande, K. Stewart, L. Lepetit, S. Gunashekar, "Distributed Ledger Technologies/Blockchain: Challenges, Opportunities and the Prospects for Standards", report prepared for the BSI, 2017. (https://www.bsigroup.com/LocalFiles/zh-tw/InfoSec-newsletter/No201706/download/BSI_Blockchain_DLT_Web.pdf).

[3] J. Bacon, J. D. Michels, C. Millard, J. Singh, "Blockchain Demystified", 2017, (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3091218).

[4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System" (2009), (https://bitcoin.org/bitcoin.pdf).

[5] M. Armbrust, et al., "A View of Cloud Computing", Commun. ACM, 53.4, 2010, pp. 50–58.

[6] Research and Markets, "Blockchain Market: Global Forecast to 2022", December 2017 (https://www.researchandmarkets.com/reports/4438515/blockchain-market-by-provider-application).

[7] P. Junco, "Accelerating the adoption of enterprise blockchain", 9 November 2017, (https://azure.microsoft.com/en-gb/blog/accelerating-the-adoption-of-enterprise-blockchain/).

[8] US Department of Commerce National Institute of Standards and Technology, "The NIST Definition of Cloud Computing", Special Publication 800-145, September 2011,

(https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf).

[9] M. Bedford Taylor, "The Evolution of Bitcoin Hardware," in Computer, vol. 50, no. 9, pp. 58-66, 2017 (http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8048662&isnumber=8048614).

[10] ITU-T Focus Group on Application of Distributed Ledger Technology (FG DLT), "Terms of Reference", 2017 (http://www.itu.int:80/en/ITU-T/focusgroups/dlt/Pages/ToR.aspx).

[11] B. Behlendorf, "Hey – You Got Your Ethereum in My Hyperledger!", Hyperledger.org, 10 April 2017 https://www.hyperledger.org/blog/2017/04/10/hey-you-got-your-ethereum-in-my-hyperledger [accessed 25 July 2017].

[12] B. Summerwill, "Comments on Hyperledger & Ethereum - Compare and Contrast" https://redd.it/694jke [accessed 25 July 2017].

[13] J. Palfreyman, "Privacy Services & Blockchain", IBM Blockchain Dev Center, 2016 (https://developer.ibm.com/blockchain/2016/09/20/privacy-services-blockchain/).

[14] B. Schneier and J. Kelsey, "Secure Audit Logs to Support Computer Forensics", ACM Trans. Inf. Syst. Secur., 2.2, 1999, pp. 159–176. 2017; J. E. Holt, "Logcrypt: Forward Security and Public Verification for Secure Audit Logs", in ACSW Frontiers '06, 2006, pp. 203–211.

[15] M. Bellare and B. S. Yee, "Forward Integrity for Secure Audit Logs", 1997.

[16] P. De Filippi & B. Loveluck, "The invisible politics of Bitcoin: governance crisis of a decentralised infrastructure", Internet Policy Review, 5(3), 2016, p. 6.

[17] P. Hacker, "Corporate Governance for Complex Cryptocurrencies? A Framework for Stability and Decision Making in Blockchain-Based Organizations", 2017 (https://ssrn.com/abstract=2998830).

[18] Hon, W.K. & Millard, C., 2013a. Cloud Technologies and Services. In C. Millard, ed. Cloud Computing Law. Oxford, United Kingdom: OUP Oxford, pp. 15-16.

[19] Intel Security/McAffee, "Building Trust in a Cloudy Sky", January 2017; CloudPassage, "Cloud Security Spotlight Report", 2016.

[20] Hon, W.K. & Millard, C., 2013a. Cloud Technologies and Services. In C. Millard, ed. Cloud Computing Law. Oxford, United Kingdom: OUP Oxford, pp. 13-15.

[21] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted Execution Environment: What It Is, and What It Is Not", in 2015 IEEE Trustcom/BigDataSE/ISPA, 2015, i, pp. 57–64.

[22] A. Kosba. A. Miller, E. Shi, Z. Wen, C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts" IEEE Symposium on Security and Privacy (SP), San Jose, CA, 2016, pp. 839-858.

[23] M. Schwarz, S. Weiser, D. Gruss, C. Maurice, S. Mangard, "Malware Guard Extension: Using SGX to Conceal Cache Attacks", in Detection of Intrusions and Malware, and Vulnerability Assessment, Lecture Notes in Computer Science (presented at the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, Cham), 2017, pp. 3–24.

[24] https://meltdownattack.com [accessed 22 Jan 2018].

[25] F. Zhang, E. Cecchetti, K. Croman, A. Juels, E. Shi, "Town Crier: An Authenticated Data Feed for Smart Contracts", in SIGSAC Conference on Computer and Communications Security (CCS '16), ACM, 2016.

[26] O. Etzion & P. Niblett, "Event Processing in Action", Manning Publications, August 2010.

[27] A. Stanciu, "Blockchain Based Distributed Control System for Edge Computing", in International Conference on Control Systems and Computer Science (CSCS), 2017, pp. 667–71.

[28] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, "Fog Computing and Its Role in the Internet of Things", in Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC '12 (New York, NY, USA: ACM), 2012, pp. 13–16.