**A Blockchain Platform for the Enterprise**

기업용 블록체인 플랫폼

Enterprise grade permissioned distributed ledger platform that offers modularity and versatility for a broad set of industry use cases.

광범위한 산업 활용 사례에 대해 모듈성과 다기능성을 제공하는 기업용 허가받은 분산 원장 플랫폼

**CHAPTER 1**

**Introduction**

In general terms, a blockchain is an immutable transaction ledger, maintained within a distributed network of peer nodes. These nodes each maintain a copy of the ledger by applying transactions that have been validated by a consensus protocol, grouped into blocks that include a hash that bind each block to the preceding block.

일반적으로, blockchain은 *peer node*의 distributed network 내에서 유지되는 불변immutable의 transaction leger입니다. 이러한 node는 consensus *protocol*에 의해 유효성이 검증된 거래를 적용하여 각 block을 이전 block에 bind하는 hash를 포함하는 block으로 그룹화하여 원장 사본을 유지 관리합니다.

The first and most widely recognized application of blockchain is the [Bitcoin](https://en.wikipedia.org/wiki/Bitcoin) cryptocurrency, though others have followed in its footsteps. Ethereum, an alternative cryptocurrency, took a different approach, integrating many of the same characteristics as Bitcoin but adding smart contracts to create a platform for distributed applications. Bitcoin and Ethereum fall into a class of blockchain that we would classify as public permissionless blockchain technology. Basically, these are public networks, open to anyone, where participants interact anonymously.

blockchain의 가장 널리 인정된 application은 bitcoin 암호해독이고, 다른 사람들은 그것을 따라갔습니다. Ethereum은 bitcoin과 동일한 많은 특성을 통합하면서 분산 application을 위한 platform을 만들기 위해 smart contract을 추가하는 등 다른 접근방식을 취했습니다. bitcoin과 ethereum은 공공 허가되지 않은 blockchain 기술로 분류할 수 있는 blockchain class로 분류됩니다. 기본적으로 이들은 공개 network로서 참가자가 익명으로 상호 작용하는 모든 사람에게 열려 있습니다.

As the popularity of Bitcoin, Ethereum and a few other derivative technologies grew, interest in applying the underlying technology of the blockchain, distributed ledger and distributed application platform to more innovative enterprise use cases also grew. However, many enterprise use cases require performance characteristics that the permissionless blockchain technologies are unable (presently) to deliver. In addition, in many use cases, the identity of the participants is a hard requirement, such as in the case of financial transactions where Know-Your-Customer (KYC) and Anti-Money Laundering (AML) regulations must be followed.

bitcoin, ethereum 및 기타 파생기술이 널리 보급됨에 따라 blockchain, 배포 원장 및 분산 application platform의 기본 기술을 보다 혁신적인 기업 사용 사례에 적용하는데 관심이 커졌습니다. 그러나 많은 기업 사용 사례에는 무의미한 blockchain 기술로 인해 (현재) 제공할 수 없는 성능 특성이 필요합니다. 또한 많은 경우, 고객알기제도 (KYC) 및 자금세탁방지 (AML) 규정을 준수해야하는 금융 거래의 경우와 같이 참가자의 신원이 어려운 요구사항입니다.

For enterprise use, we need to consider the following requirements:

기업용으로는 다음 요구사항을 고려해야 합니다.

• Participants must be identified/identifiable 참가자는 식별되거나 식별가능해야 합니다.

• Networks need to be permissioned network에 *권한이* 필요함

• High transaction throughput performance 높은 거래 처리량 성능

• Low latency of transaction confirmation 낮은 거래 지연 확인

• Privacy and confidentiality of transactions and data pertaining to business transactions

transaction 및 business transaction 관련 데이터의 개인정보 및 기밀유지

While many early blockchain platforms are currently being adapted for enterprise use, Hyperledger Fabric has been designed for enterprise use from the outset. The following sections describe how Hyperledger Fabric (Fabric) differentiates itself from other blockchain platforms and describes some of the motivation for its architectural decisions.

많은 초기 blockchain platform이 현재 기업용으로 채택되고 있지만 Hyperledger Fabric은 처음부터 기업용으로 설계되었습니다. 다음 section에서는 Hyperledger Fabric (Fabric)이 다른 blockchain platform과 어떻게 차별화되며 architecture 결정에 대한 동기를 설명합니다.

**1.1 Hyperledger Fabric**

Hyperledger Fabric is an open source enterprise-grade permissioned distributed ledger technology (DLT) platform, designed for use in enterprise contexts, that delivers some key differentiating capabilities over other popular distributed ledger or blockchain platforms.

Hyperledger Fabric은 기업 환경에서 사용하도록 설계된 공개 소스 기업용 허가된 분산원장기술 (DLT) platform으로 다른 인기 있는 분산 원장 또는 blockchain platform과 비교하여 주요 차별화 기능을 제공합니다.

One key point of differentiation is that Hyperledger was established under the Linux Foundation, which itself has a long and very successful history of nurturing open source projects under **open governance** that grow strong sustaining communities and thriving ecosystems. Hyperledger is governed by a diverse technical steering committee, and the Hyperledger Fabric project by a diverse set of maintainers from multiple organizations. It has a development community that has grown to over 35 organizations and nearly 200 developers since its earliest commits.

차별화의 핵심 포인트 중 하나는 리눅스 재단에 설립된 Hyperledger는 강력하게 지속 가능한 community와 번성하는 생태계를 성장시키는 오픈 지배구조하에 오픈소스 project를 키우는 길고 성공적인 역사가 있습니다. Hyperledger는 다양한 기술 운영 위원회와 여러 조직의 다양한 유지관리자가 Hyperledger Fabric project를 관리합니다. 최초의 commit 이래로 35개 이상의 조직과 200명이 넘는 개발자로 성장한 개발 community가 있습니다.

Fabric has a highly **modular** and **configurable** architecture, enabling innovation, versatility and optimization for a broad range of industry use cases including banking, finance, insurance, healthcare, human resources, supply chain and even digital music delivery.

Fabric은 은행업무, 금융, 보험, 의료, 인적자원, 공급망 및 심지어 디지털 음악 전달을 포함하여 광범위한 산업 사용사례에 혁신, 다기능성 및 최적화를 가능하게 하는 고도로 모듈화되고 구성 가능한 architecture를 갖추고 있습니다.

Fabric is the first distributed ledger platform to support **smart contracts authored in general-purpose programming languages** such as Java, Go and Node.js, rather than constrained domain-specific languages (DSL). This means that most enterprises already have the skill set needed to develop smart contracts, and no additional training to learn a new language or DSL is needed.

Fabric은 제한된 domain 특정언어가 (DSL) 아닌 Java, Go 및 Node.js와 같은 범용 프로그래밍 언어로 작성된 smart contract을 지원하는 최초의 분산 원장 platform입니다. 즉, 대부분의 기업은 이미 smart contract을 개발하는데 필요한 기술을 갖추고 있으며, 새로운 언어 또는 DSL을 배우기 위한 추가 교육이 필요하지 않습니다.

The Fabric platform is also **permissioned**, meaning that, unlike with a public permissionless network, the participants are known to each other, rather than anonymous and therefore fully untrusted. This means that while the participants may not fully trust one another (they may, for example, be competitors in the same industry), a network can be operated under a governance model that is built off of what trust does exist between participants, such as a legal agreement or framework for handling disputes.

Fabric platform은 권한이 부여됩니다. 즉, 공용 허가되지 않은 network와 달리 참가자는 익명이 아니므로 완전히 신뢰할 수 없습니다. 이것은 참여자들이 서로를 충분히 신뢰하지 못하는 반면 (그들은 예를 들어, 같은 업계의 경쟁자일 수 있습니다.) 법적 합의 또는 분쟁 처리의 틀과 같은 참가자 사이에 어떤 신뢰가 존재하는지에 근거하여 구축된 governance 모델에 따라 network를 운영할 수 있습니다.

One of the most important of the platform’s differentiators is its support for **pluggable consensus protocols** that enable the platform to be more effectively customized to fit particular use cases and trust models. For instance, when deployed within a single enterprise, or operated by a trusted authority, fully byzantine fault tolerant consensus might be considered unnecessary and an excessive drag on performance and throughput. In situations such as that, a [crash fault-tolerant](https://en.wikipedia.org/wiki/Fault_tolerance) (CFT) consensus protocol might be more than adequate whereas, in a multi-party, decentralized use case, a more traditional [byzantine fault tolerant](https://en.wikipedia.org/wiki/Byzantine_fault_tolerance) (BFT) consensus protocol might be required.

platform의 차별화 요소 중 가장 중요한 기능 중 하나는 특정 use case와 trust model에 맞게 platform을 보다 효과적으로 사용자 정의할 수 있는 pluggable 합의 protocol을 지원 한다는 것입니다. 예를 들어, 단일 기업 내에 배치되거나 신뢰할 수 있는 기관에 의해 운영될 때, 전체 byzantine 내결함성 합의는 불필요한 것으로 간주 될 수 있으며 성능 및 처리량에 대한 과도한 부담이 될 수 있습니다. 이와 같은 상황에서는 충돌내 결함성(CFT) 합의 protocol이 적절하지 않을 수 있지만 여러 당사자의 분산된 use case에서는 보다 전통적인 비잔틴결함(BFT) 합의 protocol이 필요할 수 있습니다.

Fabric can leverage consensus protocols that **do not require a native cryptocurrency** to incent costly mining or to fuel smart contract execution. Avoidance of a cryptocurrency reduces some significant risk/attack vectors, and absence of cryptographic mining operations means that the platform can be deployed with roughly the same operational cost as any other distributed system.

Fabric은 비용이 많이 드는 mining으로 유인하거나 smart contract 실행을 지원하기 위해 원시 암호화를 필요로 하지 않는 합의 protocol을 활용할 수 있습니다. 암호화폐를 피하면 몇 가지 중요한 위험/공격 경로가 줄어들고 암호화 mining 작업이 없으면 다른 분산 시스템과 거의 동일한 운영비용으로 platform을 구축할 수 있습니다.

The combination of these differentiating design features makes Fabric one of the **better performing platforms** available today both in terms of transaction processing and transaction confirmation latency, and it enables **privacy and confidentiality** of transactions and the smart contracts (what Fabric calls “chaincode”) that implement them.

이러한 차별화된 설계기능을 결합하여 Fabric은 거래처리 및 거래확인 대기 시간측면에서 보다 우수한 성능의 platform 중 하나를 사용할 수 있게 되었으며, 거래 및 거래를 구현하는 smart contract (Fabric이 “chaincode”라고 부르는)의 개인정보보호 및 기밀성을 보장합니다.

Let’s explore these differentiating features in more detail.

차별화 기능에 대해 자세히 살펴 보겠습니다

**1.2 Modularity (**모듈성)

Hyperledger Fabric has been specifically architected to have a modular architecture. Whether it is pluggable consensus, pluggable identity management protocols such as LDAP or OpenID Connect, key management protocols or cryptographic libraries, the platform has been designed at its core to be configured to meet the diversity of enterprise use case requirements.

Hyperledger Fabric은 modular architecture를 갖도록 특별히 설계되었습니다. pluggable consensus, LDAP 또는 OpenID Connect와 같은 pluggable idnentity management protocol, 키 관리 protocol 또는 암호화 library 이건 간에 platform은 기업 use case 요구 사항의 다양성을 충족하도록 구성되도록 설계되었습니다.

At a high level, Fabric is comprised of the following modular components:

높은 수준에서 Fabric은 다음 모듈 구성 요소로 구성됩니다.

• A pluggable ordering service establishes consensus on the order of transactions and then broadcasts blocks to peers.

pluggable Ordering Service는 거래 순서에 대한 합의를 확립한 다음 블록을 peer에게 broadcast 합니다.

• A pluggable membership service provider is responsible for associating entities in the network with cryptographic identities.

pluggable membership service provider는 network의 entity를 암호화cryptographic identities와 연관시키는 역할을 담당합니다.

• An optional peer-to-peer gossip service disseminates the blocks output by ordering service to other peers.

선택적 peer-to-peer gossip service는 ordering service를 통해 다른 peer에게 블록 출력을 보급합니다.

• Smart contracts (“chaincode”) run within a container environment (e.g. Docker) for isolation. They can be written in standard programming languages but do not have direct access to the ledger state.

smart contract ("chaincode")은 container환경 (예: Docker) 내에서 isolation되어 실행됩니다. 표준 프로그래밍 언어로 작성될 수 있지만 원장 상태에 직접 access 할 수는 없습니다.

• The ledgercan be configured to support a variety of DBMSs.

원장은 다양한 DBMS를 지원하도록 구성할 수 있습니다.

• A pluggable endorsement and validation policy enforcement that can be independently configured per application.

application별로 독립적으로 구성할 수 있는 pluggable 승인 및 유효성 검사 정책 적용.

There is fair agreement in the industry that there is no “one blockchain to rule them all”. Hyperledger Fabric can be configured in multiple ways to satisfy the diverse solution requirements for multiple industry use cases.

업계에는 "모두를 통치하는 하나의 blockchain"이 없다는 공정한 합의가 있습니다. Hyperledger Fabric은 여러 산업 분야에서 사용되는 다양한 솔루션 요구사항을 충족시키기 위해 여러 가지 방법으로 구성할 수 있습니다.

**1.3 Permissioned vs Permissionless Blockchains**

In a permissionless blockchain, virtually anyone can participate, and every participant is anonymous. In such a context, there can be no trust other than that the state of the blockchain, prior to a certain depth, is immutable. In order to mitigate this absence of trust, permissionless blockchains typically employ a “mined” native cryptocurrency or transaction fees to provide economic incentive to offset the extraordinary costs of participating in a form of byzantine fault tolerant consensus based on “proof of work” (PoW).

허가되지 않은 blockchain에서 사실상 누구나 참여할 수 있으며 모든 참가자는 익명입니다. 그런 맥락에서 blockchain의 상태가 일정한 깊이에 이르기 전에 상태가 불변하다는 것 외에 다른 어떤 신뢰도 있을 수 없다. 이러한 신뢰의 부재를 완하하기 위해 허가되지 않은 blockchain은 일반적으로 "작업증명"을 (PoW) 기반으로한 비잔틴 결함허용합의의 형태로 참여하는 특별한 비용을 상쇄하기 위해 경제적 incentive를 제공하기 위해 “채굴된” 암호화폐 또는 거래 수수료를 사용합니다.

**Permissioned** blockchains, on the other hand, operate a blockchain amongst a set of known, identified and often vetted participants operating under a governance model that yields a certain degree of trust. A permissioned blockchain provides a way to secure the interactions among a group of entities that have a common goal but which may not fully trust each other. By relying on the identities of the participants, a permissioned blockchain can use more traditional crash fault tolerant (CFT) or byzantine fault tolerant (BFT) consensus protocols that do not require costly mining.

반면에 허가된 blockchain은 일정 수준의 신뢰를 낳는 governamce model로 운영되는 알려진, 확인된 그리고 종종 심사 받는 참가자들 사이에서 blockchain을 운영합니다. 허가된 blockchain은 공통의 목표를 갖고 있지만 서로를 완전히 신뢰하지는 못 할 수도 있는 entity group간의 상호작용을 보호하는 방법을 제공합니다. 참여자의 identify를 사용하여 허가된 blockchain은 값비싼 mining을 필요로 하지 않는 보다 전통적인 CFT(crash fault tolerant) 또는 BFT(byzantine fault tolerant) 합의 protocol을 사용할 수 있습니다.

Additionally, in such a permissioned context, the risk of a participant intentionally introducing malicious code through a smart contract is diminished. First, the participants are known to one another and all actions, whether submitting application transactions, modifying the configuration of the network or deploying a smart contract are recorded on the blockchain following an endorsement policy that was established for the network and relevant transaction type. Rather than being completely anonymous, the guilty party can be easily identified and the incident handled in accordance with the terms of the governance model.

또한 이러한 허가된 context에서 smart contract을 통해 의도적으로 악의적인 코드를 도입하는 참가자의 위험이 줄어 듭니다. 첫째, 참가자는 서로 알며 application 거래를 제출하거나 network 구성을 수정하거나 smart contract을 배포하는 등 모든 작업이 network 및 관련 거래 유형에 대해 설정된 보증 정책에 따라 blockchain에 기록됩니다. 유죄가 입증된 당사자는 완전히 익명으로 처리되는 대신 쉽게 파악할 수 있으며 사건은 governamce model의 조건에 따라 처리됩니다

**1.4 Smart Contracts**

A smart contract, or what Fabric calls “chaincode”, functions as a trusted distributed application that gains its security/trust from the blockchain and the underlying consensus among the peers. It is the business logic of a blockchain application.

smart contract 또는 Fabric이 "chaincode"라고 부르는 것은 보안/신뢰를 얻고 peer간의 근본적인 consensus를 얻는 신뢰할 수 있는 분산 application으로 작동합니다. blockchain 어플리케이션의 비즈니스 로직입니다.

There are three key points that apply to smart contracts, especially when applied to a platform:

특히 platform에 적용할 때 smart contract에 적용되는 세 가지 주요 사항은 다음과 같습니다

• many smart contracts run concurrently in the network,

많은 smart contract이 network에서 동시에 실행

• they may be deployed dynamically (in many cases by anyone), and

그들은 동적으로 (많은 경우에 누군가에 의해) 배치 될 수 있고

• application code should be treated as untrusted, potentially even malicious.

application code는 신뢰할 수 없고 잠재적으로 악의적으로 취급되어야 합니다.

Most existing smart-contract capable blockchain platforms follow an **order-execute** architecture in which the consensus protocol:

기존의 smart contract 가능한 blockchain platform 대부분은 합의 실행 프로토콜인 주문-실행 architecture를 따릅니다

• validates and orders transactions then propagates them to all peer nodes,

유효성 검사 및 order transaction은 모든 peer node에 transaction을 전달합니다

• each peer then executes the transactions sequentially.

그런 다음 각 peer는 transaction을 순차적으로 실행합니다.

The order-execute architecture can be found in virtually all existing blockchain systems, ranging from public/permissionless platforms such as [Ethereum](https://ethereum.org/) (with PoW-based consensus) to permissioned platforms such as [Tendermint](http://tendermint.com/), [Chain](http://chain.com/), and [Quorum](http://www.jpmorgan.com/global/Quorum).

주문-실행 architecture는 ethereum(PoW 기반 consensus 포함)과 같은 공공/허가되지 않은 platform[에서 부터 Tendermint](http://tendermint.com/), [Chain](http://chain.com/) 및 [Quorum](http://www.jpmorgan.com/global/Quorum)과 같은 permission platform에 이르기까지 사실상 모든 기존 blockchain 시스템에서 찾을 수 있습니다.

Smart contracts executing in a blockchain that operates with the order-execute architecture must be deterministic; otherwise, consensus might never be reached. To address the non-determinism issue, many platforms require that the smart contracts be written in a non-standard, or domain-specific language (such as [Solidity](https://solidity.readthedocs.io/en/v0.4.23/)) so that non-deterministic operations can be eliminated. This hinders wide-spread adoption because it requires developers writing smart contracts to learn a new language and may lead to programming errors.

주문-실행 architecture로 작동하는 blockchain에서 실행되는 smart contract은 결정적이어야 합니다. 그렇지 않으면 합의에 도달하지 못할 수도 있습니다. 비결정론 문제를 해결하기 위해 많은 platform에서는 smart contract을 비표준 또는 domain 특정언어 (예: [Solidity](https://solidity.readthedocs.io/en/v0.4.23/) )로 작성하여 비결정적 작업을 제거할것을 요구합니다. 이는 developer가 새로운 계약을 맺어 새로운 언어를 배우고 프로그래밍 오류를 유발할 수 있기 때문에 널리 채택 될 수 없습니다.

Further, since all transactions are executed sequentially by all nodes, performance and scale is limited. The fact that the smart contract code executes on every node in the system demands that complex measures be taken to protect the overall system from potentially malicious contracts in order to ensure resiliency of the overall system.

또한, 모든 거래가 이 모든 node에 의해 순차적으로 실행되므로 성능 및 규모가 제한됩니다. 시스템의 모든 node에서 smart contract code가 실행된다는 사실은 전체 시스템의 탄력성을 보장하기 위해 잠재적인 악성 계약으로부터 전체 시스템을 보호하기 위해 복잡한 조치가 취해지도록 요구합니다.

**1.5 A New Approach** 새로운 접근 방식

Fabric introduces a new architecture for transactions that we call **execute-order-validate**. It addresses the resiliency, flexibility, scalability, performance and confidentiality challenges faced by the order-execute model by separating the transaction flow into three steps:

Fabric은 **execute-order-validate**라고**하는** 거래를 위한 새로운 architecture를 도입했습니다. 거래흐름을 세 단계로 분리하여 주문-실행 model이 직면하는 탄력성, 유연성, 확장성, 성능 및 기밀성 문제를 해결합니다:

• execute a transaction and check its correctness, thereby endorsing it,

거래를 실행 하고 그 정확성을 확인함으로써 그것을 보증하고,

• order transactions via a (pluggable) consensus protocol, and

(Pluggable)합의 protocol을 통한 주문 거래를 하고

• validate transactions against an application-specific endorsement policy before committing them to the ledger

원장에게 commit하기 전에 application별 보증정책에 대한 거래의 유효성을 검사 합니다.

This design departs radically from the order-execute paradigm in that Fabric executes transactions before reaching final agreement on their order.

이 design은 Fabric이 주문에 대한 최종 합의에 도달하기 전에 거래를 실행한다는 order0execute 패러다임에서 크게 벗어나 있습니다.

In Fabric, an application-specific endorsement policy specifies which peer nodes, or how many of them, need to vouch for the correct execution of a given smart contract. Thus, each transaction need only be executed (endorsed) by the subset of the peer nodes necessary to satisfy the transaction’s endorsement policy. This allows for parallel execution increasing overall performance and scale of the system. This first phase also **eliminates any non-determinism**, as inconsistent results can be filtered out before ordering.

Fabric에서 application-특정specific 보증endorsement 정책policy은 smart contract의 올바른 실행을 보증하는 peer node 또는 그중 몇 개를 지정 합니다. 따라서, 각 거래는 거래의 보증 정책을 만족시키는데 필요한 peer node의 서브셋에 의해서만 실행(보증)될 필요가 있다. 이를 통해 병렬 실행이 시스템의 전반적인 성능과 규모를 증가시킵니다. 이 첫 번째 단계에서는 주문 전에 일관성 없는 결과를 걸러 낼 수 있기 때문에 **비결정성을 제거합니다.**

Because we have eliminated non-determinism, Fabric is the first blockchain technology that **enables use of standard programming languages**. In the 1.1.0 release, smart contracts can be written in either Go or Node.js, while there are plans to support other popular languages including Java in subsequent releases.

비결정성을 제거했기 때문에 Fabric은 **표준 프로그래밍 언어를 사용할 수** 있는 최초의 blockchain 기술입니다. 1.1.0 릴리스에서, smart contract은 Go 또는 Node.js로 작성될 수 있는 반면, 후속 릴리스에서는 Java를 포함한 다른 대중적인 언어를 지원할 계획이 있습니다.

**1.6 Privacy and Confidentiality** (개인 정보 및 기밀 유지)

As we have discussed, in a public, permissionless blockchain network that leverages PoW for its consensus model, transactions are executed on every node. This means that neither can there be confidentiality of the contracts themselves, nor of the transaction data that they process. Every transaction, and the code that implements it, is visible to every node in the network. In this case, we have traded confidentiality of contract and data for byzantine fault tolerant consensus delivered by PoW.

우리가 논의한 것처럼 PoW를 consensus model로 활용하는 공공, 허가되지 않은 blockchain network에서 모든 node에서 거래가 실행됩니다. 즉, 계약서 자체나 처리하는 거래 데이터의 기밀성이 있을 수 없습니다. 모든 거래와 거래를 구현하는 code는 network의 모든 node에서 볼 수 있습니다. 이 경우 우리는 PoW가 제공한 비잔티움 내결함성 합의에 대한 계약 및 데이터의 기밀성을 교환했습니다

This lack of confidentiality can be problematic for many business/enterprise use cases. For example, in a network of supply-chain partners, some consumers might be given preferred rates as a means of either solidifying a relationship, or promoting additional sales. If every participant can see every contract and transaction, it becomes impossible to maintain such business relationships in a completely transparent network – everyone will want the preferred rates!

이러한 기밀성 부족은 많은 비즈니스/기업 사용 사례에서 문제가 될 수 있습니다. 예를 들어, 공급망 파트너 network에서 일부 고객은 관계를 강화하거나 추가 판매를 촉진하는 수단으로 선호 금리를 제공받을 수 있습니다. 모든 참가자가 모든 계약 및 거래를 볼 수 있다면 완전히 투명한 network에서 그러한 비즈니스 관계를 유지하는것이 불가능해집니다. - 누구나 선호하는 요율을 원할것입니다!

As a second example, consider the securities industry, where a trader building a position (or disposing of one) would not want her competitors to know of this, or else they will seek to get in on the game, weakening the trader’s gambit. In order to address the lack of privacy and confidentiality for purposes of delivering on enterprise use case requirements, blockchain platforms have adopted a variety of approaches. All have their trade-offs.

두 번째 예로서, 증권업계, 포지션을 쌍은 거래자(또는 포지션이 없는)가 경쟁자들에게 이 사실을 알리는 것을 원하지 않을 경우, 또는 그들은 게임에 참가하려고 노력할 것이고, 거래장의 책략을 약화시킨다. 기업 사용 사례 요구사항을 전달하기 위한 목적으로 개인정보보호 및 기밀유지의 부족을 해결하기 위해 blockchain platform은 다양한 접근 방식을 채택했습니다. 이들 모두는 절충점이 있습니다

Encrypting data is one approach to providing confidentiality; however, in a permissionless network leveraging PoW for its consensus, the encrypted data is sitting on every node. Given enough time and computational resource, the encryption could be broken. For many enterprise use cases, the risk that their information could become compromised is unacceptable.

데이터 암호화는 기밀성을 제공하는 한 가지 방법입니다. 그러나 합의를 위해 PoW를 활용하는 허가받지 않은 network에서 암호화된 데이터는 모든 node에 저장됩니다. 충분한 시간과 계산 자원이 주어지면 암호화가 깨질 수 있습니다. 많은 기업 사용 사례에서 정보가 손상될 수 있는 위험은 용납될 수 없습니다

Zero knowledge proofs (ZKP) are another area of research being explored to address this problem, the trade-off here being that, presently, computing a ZKP requires considerable time and computational resources. Hence, the trade-off in this case is performance for confidentiality.

ZKP(Zero Knowledge proofs)는 이 문제를 해결하기 위해 연구중인 또 다른 영역이며, 현재 ZKP를 계산하는데 상당한 시간과 계산 자원이 필요하다는 점에서 절충점입니다. 따라서 이 경우 트레이드 오프는 기밀성을 위한 성능입니다

In a permissioned context that can leverage alternate forms of consensus, one might explore approaches that restrict the distribution of confidential information exclusively to authorized nodes.

대안 형식의 합의를 활용할 수 있는 허가된 context 에서 기밀 정보의 배포를 인증된 node로만 제한하는 접근방법을 모색할 수 있습니다.

Hyperledger Fabric, being a permissioned platform, enables confidentiality through its channel architecture. Basically, participants on a Fabric network can establish a “channel” between the subset of participants that should be granted visibility to a particular set of transactions. Think of this as a network overlay. Thus, only those nodes that participate in a channel have access to the smart contract (chaincode) and data transacted, preserving the privacy and confidentiality of both.

허가된 platform인 원장 Fabric은 channel architecture를 통해 기밀성을 보장합니다. 기본적으로 Fabric network의 참가자는 특정 거래 집합에 대한 가시성을 부여 해야하는 참가자의 하위 집합 사이에 "channel"을 설정할 수 있습니다. 이것을 network 오버레이라고 생각하십시오. 따라서 channel에 참여하는 node만 smart contract (chaincode)에 접근할 수 있고 데이터가 처리되어 양쪽의 개인정보와 기밀성을 유지할 수 있습니다.

To improve upon its privacy and confidentiality capabilities, Fabric has added support for [private data](https://hyperledger-fabric.readthedocs.io/en/release-1.2/private-data/private-data.html) and is working on zero knowledge proofs (ZKP) available in the future. More on this as it becomes available.

Fabric은 개인정보 및 기밀기능을 개선하기 위해 [개인 데이터에](https://hyperledger-fabric.readthedocs.io/en/release-1.2/private-data/private-data.html) 대한 지원을 추가했으며 향후 제공될 ZKP (zero knowledge proofs)를 위해 노력하고 있습니다. 이것이 가능해지면 더 많은 정보가 제공됩니다.

**1.7 Pluggable Consensus** (연결 가능한 합의)

The ordering of transactions is delegated to a modular component for consensus that is logically decoupled from the peers that execute transactions and maintain the ledger. Specifically, the ordering service. Since consensus is modular, its implementation can be tailored to the trust assumption of a particular deployment or solution. This modular architecture allows the platform to rely on well-established toolkits for CFT (crash fault-tolerant) or BFT (byzantine fault-tolerant) ordering.

거래의 순서는 거래를 실행하고 원장을 유지 관리하는 peer와 논리적으로 분리된 합의를 위한 모듈식 구성요소로 위임됩니다. 특히, 주문 service. 합의가 모듈화되어 있기 때문에 구현은 특정 배포 또는 솔루션의 신뢰 가정에 맞게 조정할 수 있습니다. 이 모듈형 architecture를 통해 platform은 CFT(crash fault-tolerant) 또는 BFT(byzantine fault-tolerant) 순서에 대해 잘 정립된 toolkit에 의존할 수 있습니다.

In the currently available releases, Fabric offers a CFT ordering service implemented with [Kafka](https://kafka.apache.org/) and [Zookeeper](https://zookeeper.apache.org/). In subsequent releases, Fabric will deliver a [Raft consensus ordering service](https://raft.github.io/) implemented with etcd/Raft and a fully decentralized BFT ordering service.

현재 사용 가능한 릴리스에서 Fabric은 [Kafka](https://kafka.apache.org/) 및 [Zookeeper로](https://zookeeper.apache.org/) 구현된 CFT ordering service를 제공합니다. 후속 릴리스에서 Fabric은 etcd/Raft 및 완전 분산된 BFT ordering service로 구현된 Raft [consensus ordering service를](https://raft.github.io/) 제공합니다.

Note also that these are not mutually exclusive. A Fabric network can have multiple ordering services supporting different applications or application requirements.

또한 이들이 상호 배타적이지 않음을 유의하십시오. Fabric network는 여러 application 또는 application 요구사항을 지원하는 여러 주문 service를 가질 수 있습니다.

**1.8 Performance and Scalability** (성능 및 확장성)

Performance of a blockchain platform can be affected by many variables such as transaction size, block size, network size, as well as limits of the hardware, etc. The hyperledgercommunity is currently developing [a draft set of measures](https://docs.google.com/document/d/1DQ6PqoeIH0pCNJSEYiw7JVbExDvWh_ZRVhWkuioG4k0/edit#heading=h.t3gztry2ja8i) within the Performance and Scale working group, along with a corresponding implementation of a benchmarking framework called [hyperledgerCaliper](https://wiki.hyperledger.org/projects/caliper).

blockchain platform의 성능은 거래 크기, 블록 크기, network 크기 및 하드웨어 한계 등과 같은 많은 변수의 영향을 받을 수 있습니다. Hyperledger community는 현재 성능 및 규모 작업 group 내에서 [초안 세트를](https://docs.google.com/document/d/1DQ6PqoeIH0pCNJSEYiw7JVbExDvWh_ZRVhWkuioG4k0/edit#heading=h.t3gztry2ja8i) 개발중이며, [HyperledgerCaliper](https://wiki.hyperledger.org/projects/caliper)라고 하는 벤치마킹 프레임워크를 구현했습니다

While that work continues to be developed and should be seen as a definitive measure of blockchain platform performance and scale characteristics, a team from IBM Research has published a [peer reviewed paper](https://arxiv.org/abs/1801.10228v1) that evaluated the architecture and performance of Hyperledger Fabric. The paper offers an in-depth discussion of the architecture of Fabric and then reports on the team’s performance evaluation of the platform using a preliminary release of Hyperledger Fabric v1.1.

IBM Research의 팀은 Hyperledger Fabric의 architecture와 성능을 평가한 [peer 검토 논문](https://arxiv.org/abs/1801.10228v1)을 출간하여 이 작업이 계속 개발되고 blockchain platform 성능 및 규모 특성의 확실한 척도로 간주되어야 합니다. 이 백서는 Fabric architecture에 대한 심층적인 토론을 제공하고 Hyperledger Fabric v1.1의 예비 릴리스를 사용하여 platform의 팀 성능 평가에 대해보고 합니다.

The benchmarking efforts that the research team did yielded a significant number of performance improvements for the Fabric v1.1.0 release that more than doubled the overall performance of the platform from the v1.0.0 release levels.

연구팀이 수행한 벤치마킹 노력으로 Fabric v1.1.0 릴리스의 성능이 크게 향상되어 v1.0.0 릴리스 레벨에서 platform의 전체 성능이 두 배 이상 향상되었습니다

**1.9 Conclusion** 결론

Any serious evaluation of blockchain platforms should include Hyperledger Fabric in its short list.

blockchain platform에 대한 심각한 평가는 Hyperledger Fabric을 짧은 목록에 포함해야 합니다.

Combined, the differentiating capabilities of Fabric make it a highly scalable system for permissioned blockchains supporting flexible trust assumptions that enable the platform to support a wide range of industry use cases ranging from government, to finance, to supply-chain logistics, to healthcare and so much more.

결합, Fabric의 차별화된 기능은 유연한 신뢰 가정을 지원하는 허가된 blockchain을 위한 확장성이 뛰어난 시스템을 구축함으로써 이 platform을 사용하면 정부, 금융, 공급망 물류, 의료 및 그 이상의것을 지원할 수 있습니다.

More importantly, Hyperledger Fabric is the most active of the (currently) ten hyperledger projects. The community building around the platform is growing steadily, and the innovation delivered with each successive release far out-paces any of the other enterprise blockchain platforms.

더 중요한것은, Hyperledger Fabric은 (현재) 10개의 Hyperledgerproject 중 가장 활발한 것입니다. platform을 중심으로 한 community 구축이 꾸준히 진행되고 있으며, 연속된 각 릴리스에서 제공되는 혁신은 다른 기업 blockchain platform 보다 훨씬 앞섰습니다

**1.10. Acknowledgement**

The preceding is derived from the peer reviewed [“Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains”](https://arxiv.org/abs/1801.10228v1) - Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolic, Sharon Weed Cocco, Jason Yellick