

# Blockchain Reference Architecture

---

A Smarter way to implement Agile  
& Effective Blockchain Solutions

Whitepaper

## Abstract

There is a lot of excitement around Blockchain technology these days as what was once dubbed as a maverick technology without a central authority is being adapted to benefit the mainstream businesses. There are a number of supporting technology elements that need to come together precisely to make this emerging technology quickly deliver business value. The technical skills and experience of the technical team also plays a decisive role in an implementation. The success of a typical implementation is likely to be a function of all these variables and hence, is likely to be unpredictable. A comprehensive business-ready Blockchain Reference Architecture offering along with pre-assembled Blockchain components from Coforge ensures unconditional success of a standards-based Blockchain implementation with minimal effort. With the reference architecture in hand, development teams can design and develop solutions to specific business problems without having to reinvent the wheel or losing time in experimentation.

## Challenges of Adapting an Emerging Technology

There are several popular Blockchain development platforms such as Hyperledger Fabric, R3 Corda and Ethereum. As Blockchain Platform standards are evolving, the technology keeps growing and changing, the development teams can find it difficult to be on top of the latest features without dedicated resources. In such a development framework, it does take their time away from specific business challenges that they are trying to address with Blockchain. In addition, they may come up with different solutions each time with potentially sub-optimal solutions.

Though such challenges are to be expected with any emerging technology, they are more pronounced with Blockchain because of its unique features such as open source development model, distributed computing platform and complete lack of central oversight.

## A Quick Primer on Blockchain

“A Blockchain, originally block chain, is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. By design, a Blockchain is resistant to modification of the data. It is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way. For use as a distributed ledger, a Blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks, which requires consensus of the network majority.”<sup>1</sup>

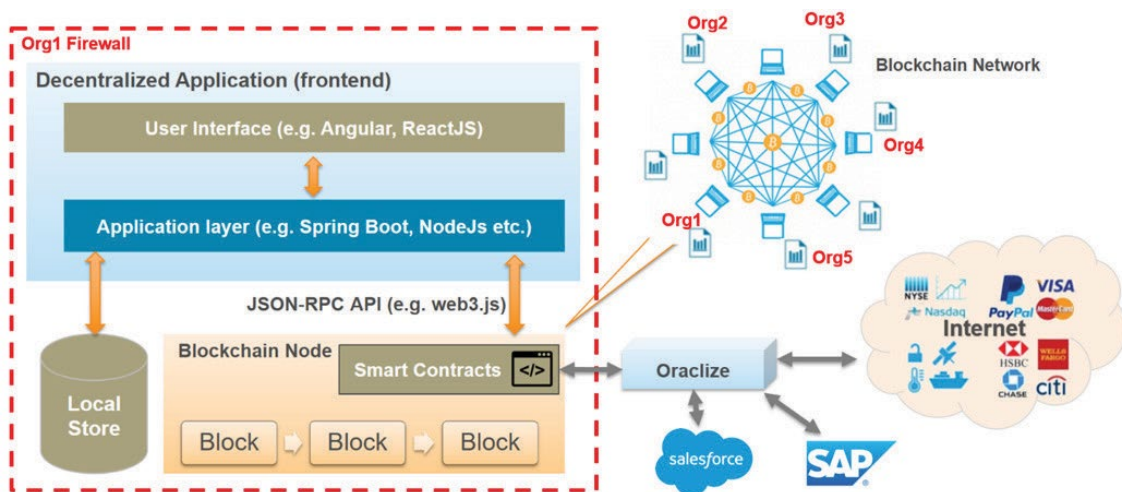


Figure 1: Archetype of a Blockchain network implementation

The above diagram illustrates the archetype of a Blockchain network-based system.

- A Blockchain network encompasses multiple organizations with each organization getting its own copy or view of the distributed ledger maintained on the Blockchain network.
- Instead of a centralized application, a distributed ledger-based system is employed with nodes spread across the participating organizations.
- Based on their role, the organizations in the Blockchain network may become participating nodes or may use a public or a shared node.
- Each organization gets a REST API interface to operate the underlying Blockchain layer. The users would typically work with enterprise applications or DApps (Distributed applications) that interface with the Blockchain network through these REST APIs.
- Off-chain storage is used sometimes to manage any data required for the overall application. A hash of off-chain data is stored on the Blockchain to maintain its validity & ensure trusted provenance. DApps or enterprise systems have the choice of using data from either the Blockchain or an off-chain database.
- Business Rules and contracts can be codified using Smart Contract feature of the Blockchain. Smart Contract code gets distributed with each node and gets compiled & executed on each node for each transaction (in some Blockchain protocols, the smart contract execution is need-based). Since each node has the same version of the smart contract, its execution result on local node can be compared with results coming from other nodes to validate against potential rule-tampering.
- The Blockchain node communicates with the rest of the nodes in the network using standard Blockchain protocols. This ensures that each node always has the latest and a consistent view of the distributed ledger.
- To integrate with external services and consume data thereof (Oracle), Oraclize services are implemented. Oraclize services help validate and sign the external data so that its validity is ensured and all nodes get the same set of external data.

As can be imagined, developing a distributed system using Blockchain network, DApps, Integrations with Enterprise systems, Oraclize services, and off-chain database to implement enterprise features like security, role-based access, faster time-to-market and micro-services-based architecture becomes a herculean task for the development team. The development teams tend to get bogged down with ensuring a functional implementation of all the plumbing and as a result, are often not able to focus on the actual business problem.

### **The Blockchain Reference Architecture Solution from Coforge**

A solution based on well-designed Blockchain reference architecture can alleviate the typical development and implementation challenges that technical teams face in implementing Blockchain. When the development platform is derived from the reference architecture, it can ensure a standards-based Blockchain implementation quickly and successfully. The reference architecture can also help organizations and customers in reducing the overall effort for Blockchain implementation through its ease of integration, accelerated time-to-market and faster development and hence, iron out the overall Blockchain technology adoption process.

The overarching guiding principles of the Blockchain reference architecture from Coforge are to:

- Provide a Standards-based implementation across all projects.
- Deliver the scaffolding code that would support the development teams to be productive quickly and to reduce their time-to-market.
- Abstract the application code from the Blockchain platform-specific idiosyncrasies so that development teams can follow consistent design patterns across all Blockchain development platforms.

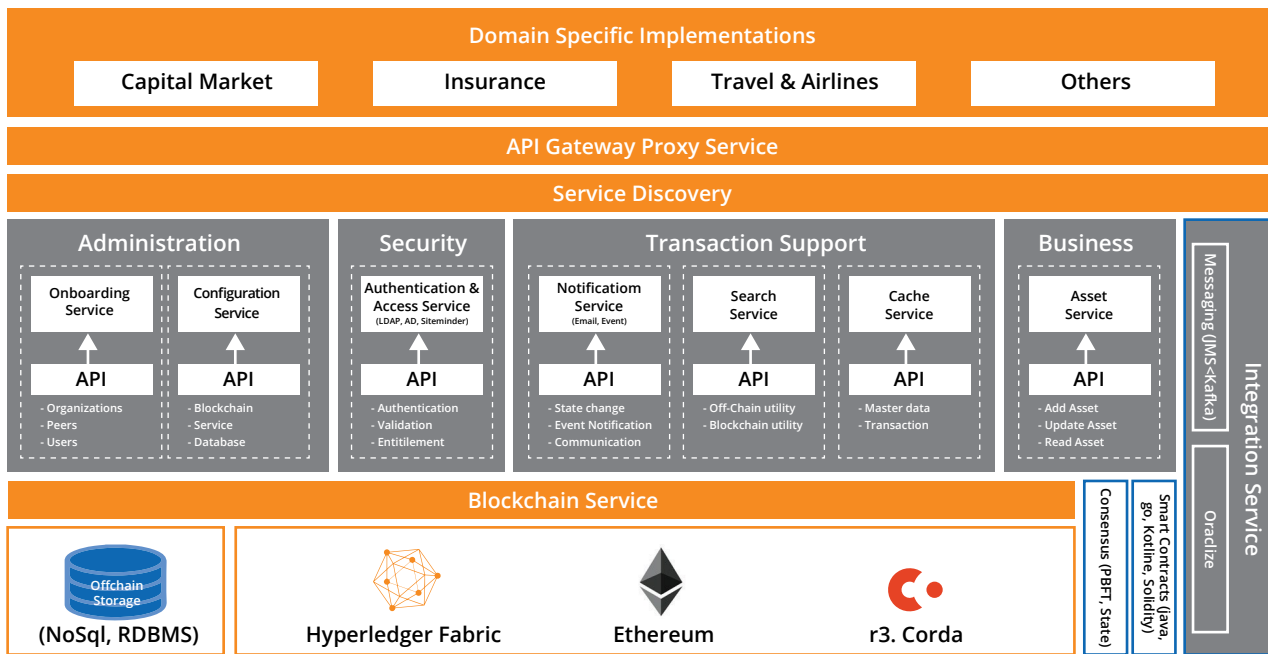


Figure 2: Blockchain Reference Architecture from Coforge

The reference architecture as outlined in the above diagram defines the various service components involved in enabling a complete Blockchain enablement platform. As the reference architecture framework and its reusable components and services are elaborated further, it should be noted that the framework is designed and developed to reduce the overall development effort needed for developing a robust Blockchain solution.



Figure 3: Components of the Blockchain Reference Architecture

### Technical Overview

The various technologies used to develop the accompanying platform for the Blockchain reference architecture are detailed below. The following general guidelines should be kept in mind as one goes through the technical overview of the various foundational technologies:

- All services are REST-based. They consume and produce JSON format.
- The services are developed using spring boot framework.
- A micro-service based framework is generally used.

Spring Boot Netflix Eureka library is being used extensively for the implementation of the microservices.

- Following client implementations are enabled:
- Hyperledger Fabric - Java SDK implementation.
- Ethereum - Web3 java SDK implementation.
- R3 Corda - Kotlin Client SDK implementation.
- Smart contracts are based on either Golang, Java, Kotlin or Solidity depending on the Blockchain platform.
- The implementation could be deployed on any of the cloud platforms (i.e. Azure, AWS, Digital Ocean, IBM Bluemix etc.) or on any on-premises system.
- Kafka queues are implemented for messaging support.
- The following design patterns are extensively used to develop various components:
  - Strategy
  - Observer
  - Adapter
  - Singleton
  - Façade
  - Command

## Architecture Service Components

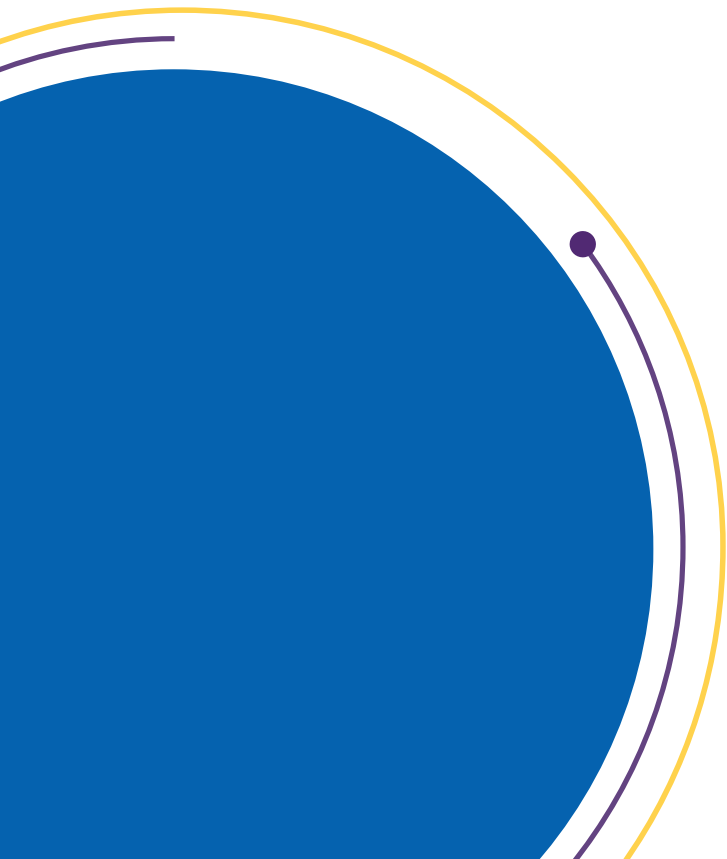
### API Gateway

API Gateway is the gateway to all the services provided by this platform. The services are exposed as REST interfaces and are published on the API Gateway. It is implemented using Spring Boot Netflix Zuul library. The API Gateway provides:

- Complete description of all the APIs available for the platform.
- Integration of all the security services of the platform to authenticate and authorize execution of each service.
- Throttling to mandate a cap on consumption of the services of the platform and prevent over-consumption.

### Service Discovery

Service Discovery allows the platform services to register themselves dynamically and keep their state constantly updated. It also provides the feature to locate them based on their updated state. API gateway is integrated with the service discovery to route incoming calls to the final endpoints for all the services. It is implemented using Spring Boot Netflix Eureka service discovery framework.



## Administrative Service Components

### On-boarding service

The on-boarding service helps in getting multiple organizations on to the Blockchain platform. The following features are supported:

- Facilitating consensus by existing organizations before accepting any new organization into the network.
- Introducing new Blockchain peer nodes for organizations.
- Introducing new user identities for the existing organizations on the platform as well as on Blockchain.
- Removing any existing node of an organization from the network, if needed.

### Configuration Service

- The configuration service maintains all the configurations required by the system. It primarily supports:
  - Off-chain database configurations.
  - Blockchain platform specific configurations:
  - Hyperledger Fabric's environment & smart contract specific configurations.
  - Ethereum's environment & smart contract specific configurations.
  - R3 Corda's environment & CordApp specific configurations.
  - Blockchain certificate configurations.
  - All service platform-specific configurations.

## Security Service Components

The security services provide the necessary features to authenticate & authorize any user or request.

### Authentication Service

The authentication service manages authentication for the platform. It comprises adapter implementations for various supported authentication protocols. It supports the following authentication types:

- **oAuth**: Configures the platform with an existing oAuth server. It accepts the authorization-specific details and verifies them against the oAuth server.
- **LDA**: It provides the features used by the LDAP-specific adapter to integrate with the existing LDAP system.
- **Siteminder**: It provides the features to use Siteminder-specific adapter to integrate with the existing Siteminder implementation.

### Access Service

The access service provides access management for the entire platform. Its features are:

- Managing the role-specific entitlements.
- Verifying whether the user has the required access for the operation he/she intends to perform.

This service is integrated with the API gateway to intercept platform-specific service requests and verify their access privileges.

## Asset Service (Business) Components

The Asset service is a specific implementation under Business Services and has been implemented as a generic service that provides the required asset management features. Every Blockchain implementation manages one or more asset types and these assets could be physical or liquid. The Asset service would allow management of all these assets and their attributes. The following features are supported:

- Manage any kind of asset on the Blockchain by interacting with the Blockchain engine:
- Introduce an asset.
- Modify existing asset and change their state.
- Query content of the assets saved on Blockchain.
- Transfer Ownership of an asset.
- Interact with the smart contract layer to perform asset operations.

The signatures of this service are kept generic so that any type of asset can be handled without requiring any service or signature change.

### Transaction Services

This set of services aid transaction management on the Blockchain platform. Following services come under its umbrella:

#### Notification Service

The notification service helps track notifications based on the operations performed on the assets maintained on the Blockchain and accordingly sends the notifications. It provides the implementations to send notifications in the form of SMS, Email and by using messaging queues. It also facilitates to send push notifications to mobile applications.

#### Search Service

The search service provides the capability to perform indexing on the data being stored on the Blockchain. Based on the generated indexes, it facilitates to perform search and fetch the content from the Blockchain platform.

#### Cache Service

The cache service provides the capability to cache the data which is continuously referred by various services of the platform. This helps to speed up many of the operations of the platform.

## Integration Service Components

The integration services facilitate the integration of the Blockchain-based system with external platforms used to share data with and to pull data from. It provides messaging queues and Oraclize services that can be integrated with external systems:

- Kafka-based queues are implemented for publishing the state change-based transactions.
- Additionally, Oraclize services are provided to integrate with existing external systems or external data.
- With an adapter-based pattern implementation, custom adapters could be readily developed based on the external system, plug them into this architecture and integrate them with external systems.

## Recommended Architecture Integration Flow

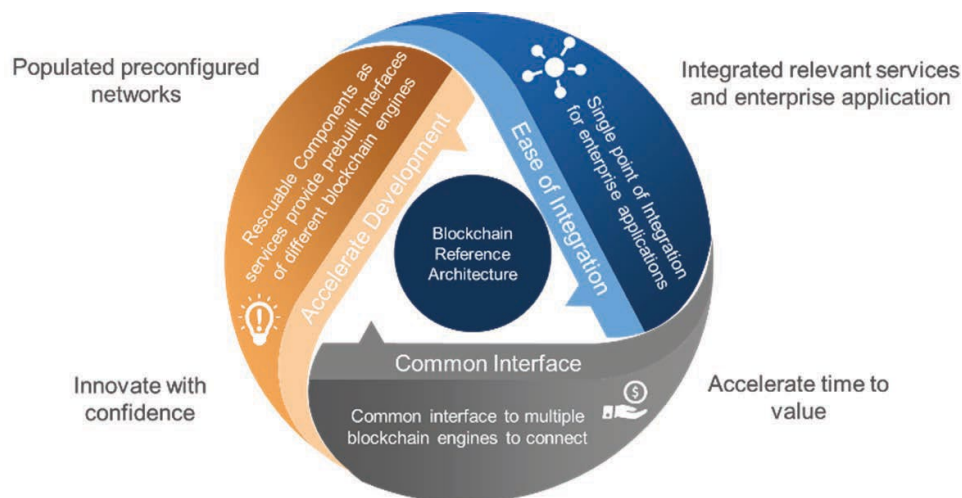
As the Blockchain reference architecture is implemented in the development platform, it is recommended that the organization follow the methodology below before enabling their specific Blockchain implementation:

- Determine the organizations participating in the overall use case.
- Ensure each organization has its own instance of the Blockchain platform deployed on their own infrastructure whether it is cloud or an on-premises system.
- On-board the organizations using the administrative services.

- Create the organization-specific configurations for database, authentication and messaging on the platform during the setup.
- Configure Smart contracts based on the business use case.
- Create identities specific to all the users for accessing their nodes using the on-boarding services.
- Integrate the organization's business services to configure the endpoints of all Blockchain services.

## Business Benefits

- Accelerated development and time-to-value.
- Reusable components as services providing pre-built interfaces for different Blockchain engines.
- Consistent and Efficient Standards-based implementation.
- Populated Pre-configured networks
- Common interface to multiple Blockchain engines.
- Ease of integration with all enterprise applications - native and external.
- Easier adoption of Blockchain into existing infrastructure.





## The Reference Architecture that makes a difference

Blockchain is a unique technology conceived as an alternative to what is considered mainstream. It has now matured to a point that its unique capabilities have become extremely attractive to mainstream businesses. Governments around the world are grappling with how to 'regulate' it as it was not designed with governance in mind. Technical teams trying to adapt Blockchain need to acquire the expertise in various unique supporting technologies around it. The technical intricacies can easily consume the bandwidth of the technical teams.

The Blockchain reference architecture with pre-assembled components designed and delivered by Coforge is production-ready with enterprise-grade capabilities. With its easier adaptability and faster development capability, the reference architecture fills a critical gap in expertise while enabling the business organizations and their technical teams to focus more on their business use cases. It will enable a consistent standards-based

solution and help the organizations accelerate their Blockchain adoption journey by driving solutions to the market quickly, cost-effectively and successfully



# The Coforge Thought Board: Blockchain Reference Architecture – A Smarter way to implement Agile and Effective Blockchain Solutions

## What are the unique design principles of Blockchain Technology?

Distributed shared ledger

Applies decentralized  
consensus mechanism to  
accept the transaction in  
shared ledger

Peer to Peer transmission of  
transactions

Access based transaction  
transparency

Immutability and  
irreversibility of data once  
recorded

Business rules can be coded  
inside the smart contracts

## What are the main methodologies to address the scalability challenges?



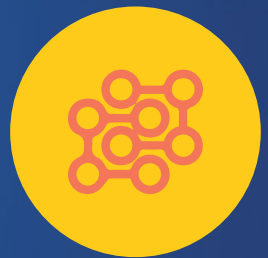
Technology on a  
constantly changing  
trajectory



Intricate technology  
and unique  
features



Lengthy development  
time and adaptability  
challenges



Being an emerging technology  
the available platforms are not  
yet standardized to facilitate and  
expedite enterprise adaptability

## How can Blockchain Reference Architecture contribute to an efficient implementation?

Consistently  
standards-based  
implementation

Deliver scaffolding code to  
support development  
professionals

Abstraction from application  
platform idiosyncrasies  
developed in voluntary open  
source model

Ensure consistent design  
and development flow

Accelerated Time-to-market

Easier technology adoption  
process

Ease of integration with existing and external systems

For more information, contact [information@coforgetech.com](mailto:information@coforgetech.com)

© 2020 Coforge. All rights reserved.

Coforge is a leading global IT solutions and services organization which believes that real transformation cannot be driven by thinking in technology terms alone. With a mission to “Transform at the Intersect” it aims to bring both deep domain and deep emerging technologies expertise to achieve real-world business impact. A focus on very select industries, a detailed understanding of the underlying processes of those industries and partnerships with leading platforms provides us a distinct vantage. We leverage AI, Cloud and Insight driven technologies, allied with our industry expertise, to transform client businesses into intelligent, high growth enterprises.

Learn more about Coforge at: [www.coforgetech.com](http://www.coforgetech.com)

Stay connected:    

---

**Coforge**