RiskBlock Strategic Framework Model

RiskBlock
Alliance
™

Table of Contents

Approach

Background

- The Institutes has developed its vision for bringing its diverse member base together to maximize the benefits of distributed ledger technology, and understands that there are first mover advantages to building a P&C Blockchain consortium
- The organization is focused on articulating its plan to drive the adoption of Blockchain in the risk and insurance space. At the plan's foundation will be a strategic vision and approach which will lay out technical and business requirements, next steps, and timing decisions that will optimize industry participation and market share
- The Institutes has engaged Deloitte Consulting to formulate a strategic plan for the creation of RiskBlock, a P&C Risk and Insurance Blockchain Consortium

Architecture Options Evaluated

Option 1

Single Permissioned Blockchain

Option 2

Federated Blockchain

Option 3

Federated Blockchain with Communications Hub

Architectural Guiding Principles

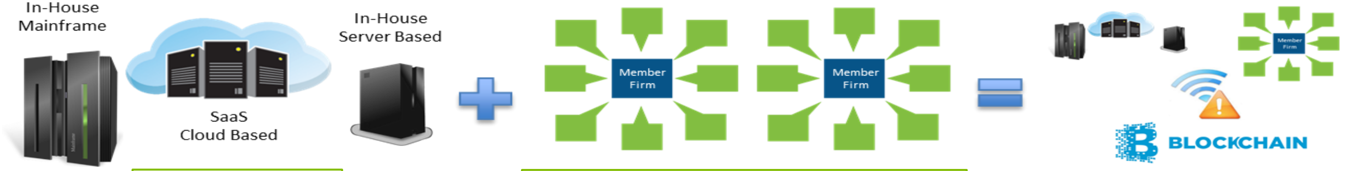
1. Cross-Industry Blockchain Foundation
2. Consortium for Interactive, Distributed Ledger Technology
3. Scalable and interoperable foundation to support a "**build once, use many**" approach
4. Approach for Regulatory and Governance for Data Sharing
5. Maximize Transaction Standardization & Adoption
6. Establish an extensible and scalable Blockchain architecture for transactions

Recommendation

Adopt **Federated Blockchain** at initial launch with a plan for evolution to **Federated Blockchains with Communications Hub** for efficient integration and scalability

Observations: Technology Landscape Across TI Member Firms

Member Firm Back Office Challenges



Diverse Policy Admin Platforms

Member Firms Have Multiple Policy Admin Systems (Most 5 or More!)

Creates Connectivity Challenges to Connect to External Systems Such As Blockchain

Blockchain Adoption Challenges

- Varying levels of Blockchain technology maturity
- Most mature Blockchain capabilities are at POC / exploratory level
- Path to production-ready Blockchain application is not clear yet for most member firms
- Independent directions in exploration efforts across member firms with no coordination across the industry

Technology Framework Success Factors

Higher Transactions Visibility at High Performance

- Visibility of data across shared Blockchain can lead to transparency

Back Office Connectivity Solution with High Scalability

- Standardized API's for connectivity to major Policy Admin Systems/Platforms
- Bespoke API connectivity for non-Major Policy Admin Systems
- Push/Pull DMZ Solution for Member Firms Not Comfortable with APIs

Effective Central Governing Body

- Control permission and access to data
- Control network membership and nodes participating in consensus

Framework Model

Architecture Options

Architecture: Objectives & Assumptions

Architecture | Interoperability | Membership Management | Identity Management | Permissions Management | Keys Management | Information Management | Usage Monitoring

Key objectives and assumptions considered to develop the operational and technical details of the recommended Architecture Solution

Objectives

Applications

- **Blockchain enabled** business processes
- **Interconnected business processes**
- Ability to support **Agile development methodologies**

Data

- **Data sharing** across member firms
- **Permissioned** data access
- **Standardized** transactions formats

Governance

- **Open Architecture** to allow fair participation to all member firms
- Adhere to best practice **IT security standards**
- **Future readiness -** Ability to adopt newer blockchain platforms

Technology

- Support leading **integration patterns**
- Ability to support **diverse application tech stacks**
- Allow for **simple and fast application development**
- Enterprise **SLA compliant** performance

Assumptions

- Transaction data standards will be defined
- Enterprises would like to host confidential data in their private data stores
- RiskBlock system roles limited to organization level
- Strong Consortium operations led by central governance

Architecture: Design Principles

Architecture | Interoperability | Membership Management | Identity Management | Permissions Management | Keys Management | Information Management | Usage Monitoring

Series of architecture design principles are considered that boost production and efficiency, and minimize potential risks

Open Architecture

Allow RiskBlock members to create applications in the ecosystem. RiskBlock Services should be reliably exposed to all applications in a non-proprietary manner

Business Process Subsystem is architected to allow app contribution from all RiskBlock members

White Labeling

Solution functionalities need to be exposed to through an externally facing API layer to allow for other institutions to re-use as required

'Build Once, Use Many' paradigm is encapsulated in Governance APIs

Efficient Scalability

All components within technology landscape should provide effortless, rapid, near-linear scalability without an exponential increase in associated costs

Multiple blockchain architecture ensures linear scalability.

Automate and Digitize Processes

Core business and technology processes should be automated and digitized wherever possible to promote efficiency and improve production

Reusable SmartContract libraries, strong **Dev Tooling** help automate several business and tech processes

Componentized Architecture

Architecture must ensure that as many of its core systems as possible are comprised of reusable, functionally granular components.

Subsystem based architecture ensures **Platform Modularity** and **Componentization**

Seamless Integration

Architecture should consider ability to effectively and seamlessly combine, analyze, and manage all data pertaining to business events and data

Interoperability layer and APIs provide seamless integration

Architecture: Options

The following Solution Options were considered and Architecture Design Principles applied for Architecture of the RiskBlock Platform

Single Permissioned Blockchain

A single Blockchain established across all member firms, used by all applications across various Business Functions

Single Blockchain Architectures are the current de facto standard for blockchain applications in the industry

Federated Blockchain

A network of Blockchains with each Business Function being powered by dedicated Blockchains, leading to higher throughput

An improved blockchain architecture utilizing multiple blockchains for increased scalability and security

Federated Blockchain with Communication Hub

Next level of improvement, targeting improved interoperability between the network of Blockchains

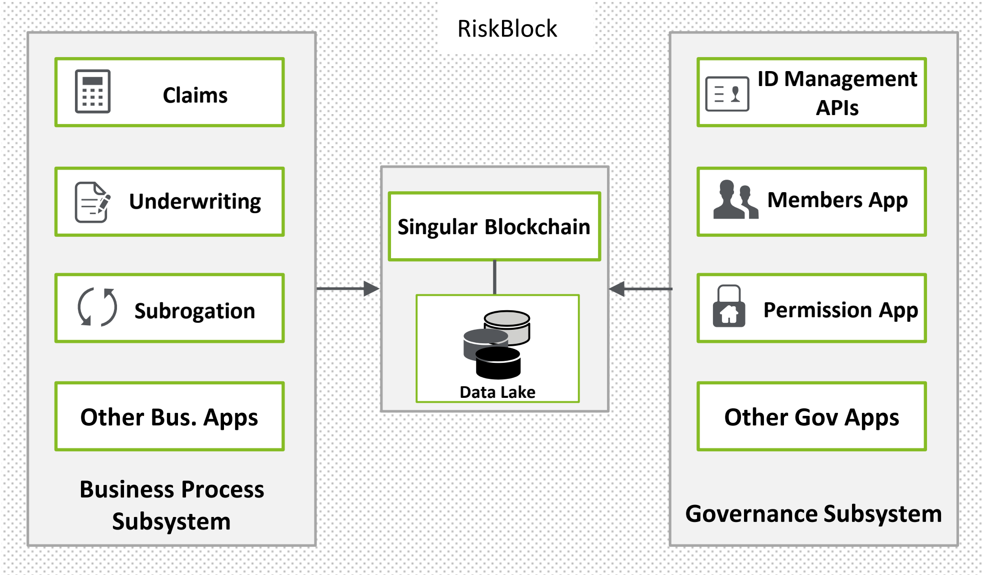
Messaging best practices are implemented for efficient data sharing across the various blockchains through a Blockchain based Communications Hub

Architecture: Option 1 – Single Permissioned Blockchain

Architecture | Interoperability | Membership Management | Identity Management | Permissions Management | Keys Management | Information Management | Usage Monitoring

Summary

A single Blockchain established across all member firms, used by all applications across various Business Functions



Solution Detail

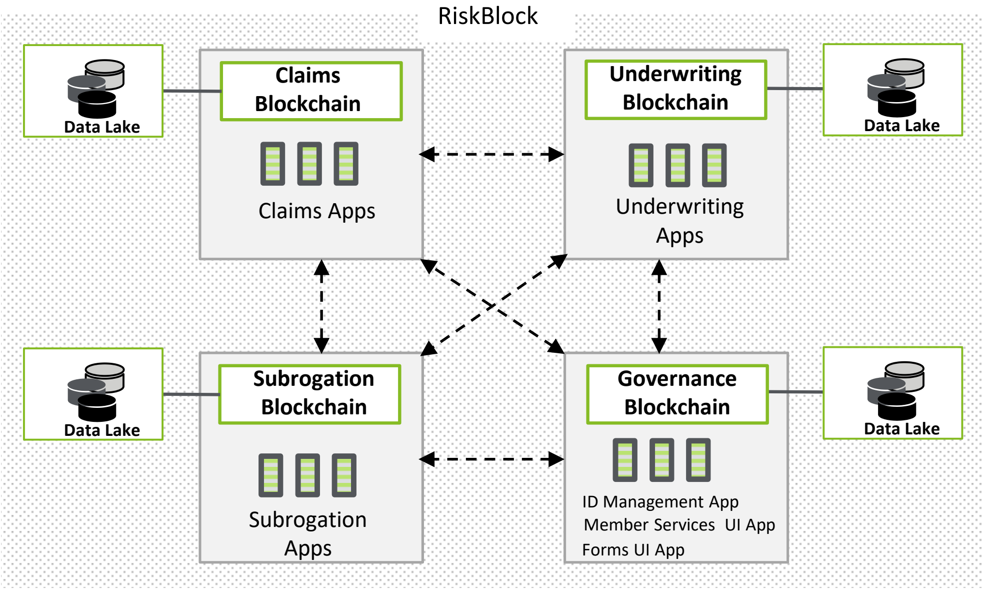
- Common, Single Blockchain across all member firms
- Transaction data resides on the common blockchain and is protected by permissions
- All Business and Governance applications run on the common blockchain
- Data sharing will occur between different applications via the common blockchain
- All smart contracts would be executed on the same Blockchain

Architecture: Option 2 – Federated Blockchain

Architecture | Interoperability | Membership Management | Identity Management | Permissions Management | Keys Management | Information Management | Usage Monitoring

Summary

A network of Blockchains with each Business Function being powered by dedicated Blockchains, leading to higher throughput



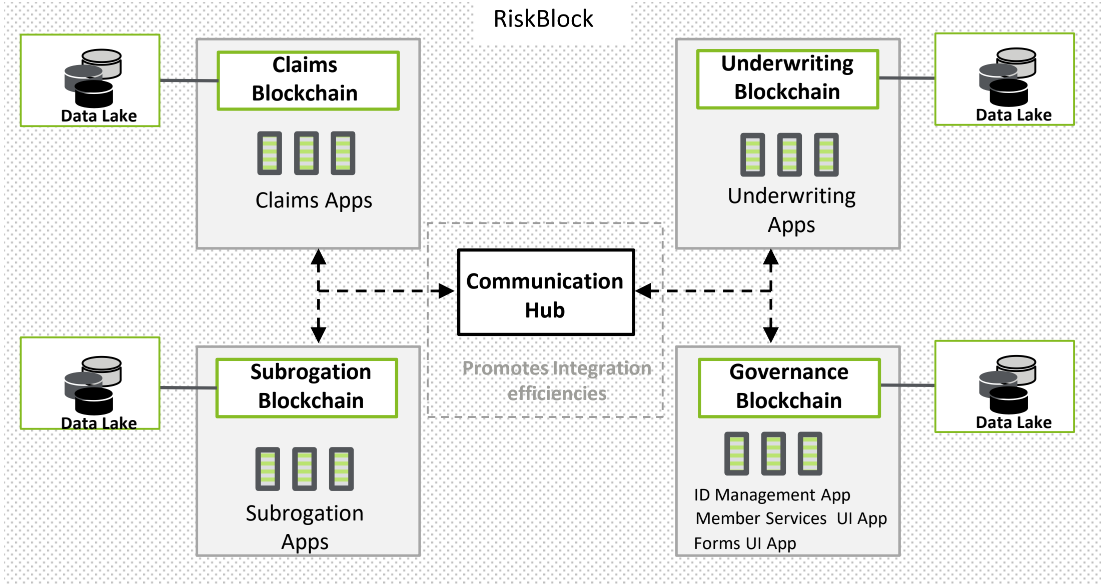
Solution Detail

- Federated Blockchains model provides for separate blockchains for each Business Function
- All business applications run on their respective Business Function blockchains (eg: Claims, Underwriting, Subrogation etc)
- Data sharing between business processes would be achieved via Blockchain interoperability (Blockchain to Blockchain communication)
- Business process specific Smart Contract execution will take place on the respective blockchain hosting that process
- Governance data will be confined to the Governance Subsystem blockchain
- Uniform operations due to homogeneity of Business Blockchains architecture

Architecture: Option 3 – Federated Blockchain with Communications Hub

Summary

A network of Blockchains with each Business Function being powered by dedicated Blockchains, leading to higher throughput
Messaging best practices are implemented for efficient data sharing across the various blockchains through a Blockchain based Communications Hub



Solution Detail

- A federated Blockchain model with a separate Blockchain for each Business Function would be deployed
- Data Sharing between business processes will be delegated to a decentralized Central Communications Hub
- Encrypted data would be stored in Data Lake with the verification records and data pointers on the Blockchain
- Business process specific Smart Contract execution will take place on the respective blockchain hosting that process
- Uniform operating models due to homogeneity of Business Blockchains architecture

Architecture: Recommendation

Architecture | Interoperability | Membership Management | Identity Management | Permissions Management | Keys Management | Information Management | Usage Monitoring

Planned Evolution

	Single Permissioned Blockchain	Federated Blockchains 1	Federated Blockchains with Communications Hub 2
BENEFITS	▪ Conceptually **simple architecture** ▪ **Seamless data sharing** between different functional areas ▪ **Lower costs** for initial rollout	▪ **High performance** due to data and smart contracts segmentation ▪ **Linear Scalability** as Blockchain-network can be expanded as required ▪ Data Segmentation allows for **faster time to market** ▪ Allows for **better security** models due to blockchain segmentation	▪ Seamless data sharing and efficient inter Blockchain communication through a **Communications Hub** ▪ Enhanced flexibility in supporting **diverse technology stacks** ▪ Further **Scalability improvements** with increased messaging efficiency
CHALLENGES	▪ **Limited capacity**, suitable for trivial use cases, POCs, etc. ▪ Need for **common IT Standards** across all member firms ▪ High Cost and **High Risk upgrades** ▪ Force **common technology stack** across all member firms ▪ **Increased security risks** due to scalability limitations	▪ Increased **system complexity** ▪ **Additional development** effort due to need for Blockchain interoperability ▪ Increased **maintenance effort**	▪ **Additional complexity** due to Communications Hub

1 Recommendation

Federated Blockchain model is recommended for the initial launch of RiskBlock

2 Planned Evolution

- Federated Blockchains with Communications Hub is expected to be a natural progression in the future for better supporting system efficiency
- This option is recommended to be considered as a future evolution and not a Go Live candidate
- Simple Federated Blockchains is a ‘Minimum Viable Product’ that can achieve a faster time to market

Architecture: Industry Trends and Best Practices

Single Blockchain architectures have been the traditional preference for decentralized platforms. To build more sophisticated platforms, the Industry is gravitating towards multiple blockchain architectures and allowing for Blockchain Interoperability.

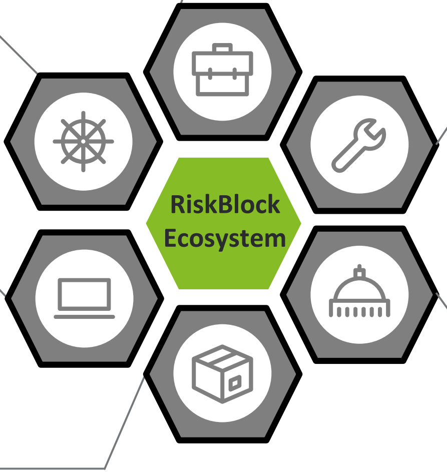
Industry Response		RiskBlock Adaptation
Cosmos → Network of Blockchains with a central blockchain 'Cosmos hub' to aid Inter Blockchain Communication. **Target launch date: Q4, 2017**	→	**Communications Hub** for efficient messaging
Plasma Solution → White paper published by Vitalik Buterin and Poon as a Scalability solution for blockchains using interoperable 'Baby Blockchains' **Release date: August 2017**	→	**Multiple blockchains** for higher scalability
Microsoft CoCo framework → Distributed governance model for blockchain networks to allow fine grained security and access control as per enterprise application requirements. **Release date: August 2017**	→	Reusable RiskBlock Services, Development Kit and Accelerators
Inter Ledger Protocol (ILP) → "Protocol for connecting ledgers" from Ripple for enabling payments across different blockchains **Public Demo: May 2017**	→	**Interoperability** via Blockchain Connectors

***Important Note**: Most of the relevant leading industry solutions are still emerging designs and are expected to evolve significantly in near to mid term. Also, these designs may not have a 100% overlap with RiskBlock requirements*

Architecture: RiskBlock System Components

Architecture | Interoperability | Membership Management | Identity Management | Permissions Management | Keys Management | Information Management | Usage Monitoring

Projected Target state for RiskBlock ecosystem is shown here with breakdown into key system components.



Business Applications

RiskBlock members can fortify their business processes with blockchain enabled use cases and create corresponding applications on the ecosystem

RiskBlock Operating Model

The RiskBlock Operating Model is the foundation with which the core architectural solution is aligned

RiskBlock Infrastructure

Infrastructure for Network of subsystems that comprise of blockchains and other runtime components required for application execution

Smart Contracts Development Kit

Smart Contracts SDK and the relevant Developer Documentation for building and rolling out business applications, by the member firms

RiskBlock Services

Set of Governance applications that allow member onboarding, permissions management, blockchain interoperability, analytics etc.

Development Accelerators

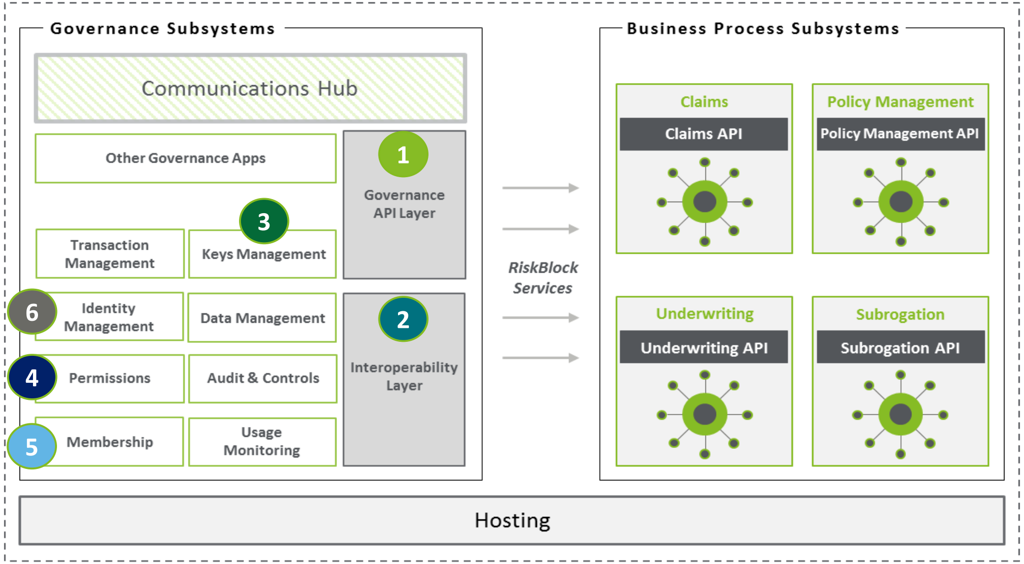
Robust development infrastructure to facilitate application development, consisting of stable Test environments and best of class app development tools

Important note: The key system components are consistent across all Solution Options

Architecture: Summary

With Federated Blockchains architecture, the RiskBlock ecosystem decomposes into 2 top level subsystems:

- Governance Subsystem: APIs and Applications that provide underlying services and utility functions for all RiskBlock applications
- Business Subsystems: The business processes encapsulated in the applications that will run on top of the RiskBlock architecture



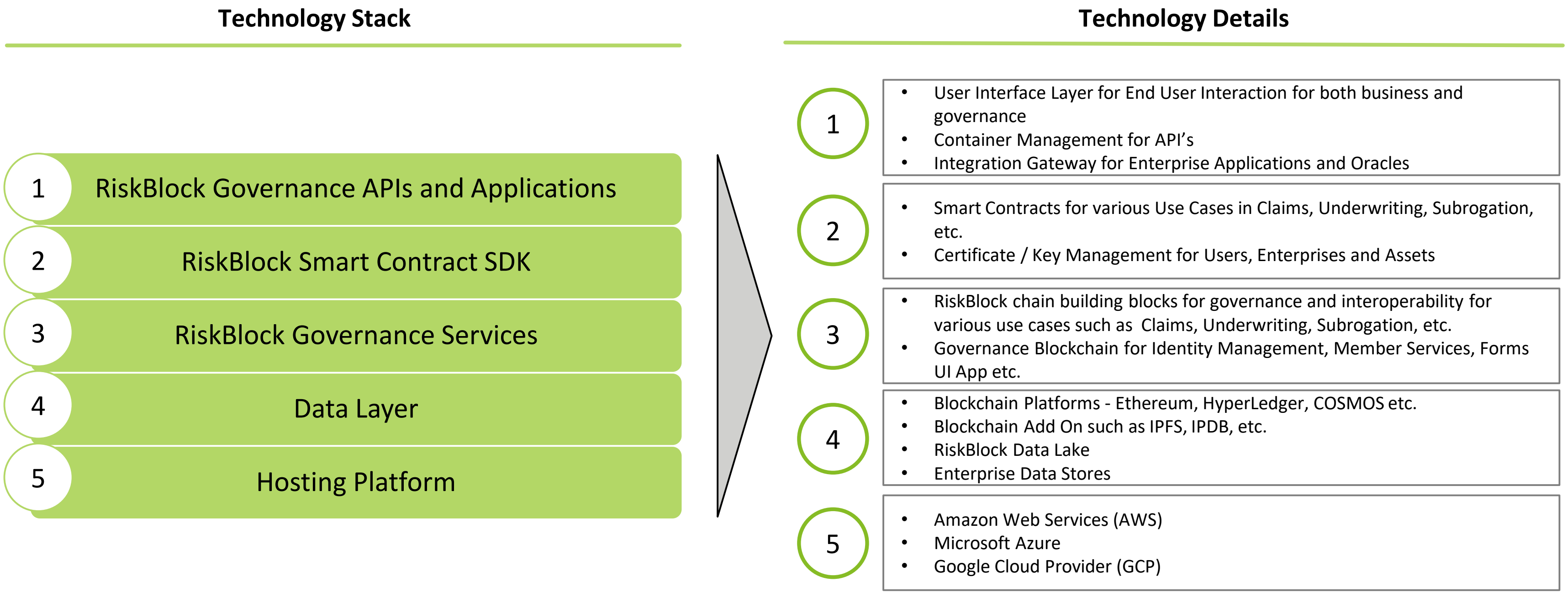
▨ Evolution State (Subsequent Phase)

Label	Subsystem Function
1	Governance API layer **exposes RiskBlock governance functions** to other subsystems to leverage in decoupled manner
2	Enables the **inter blockchain communication** and data sharing across RiskBlock subsystems
3	Keys Management APIs used to **securely store and use confidential blockchain keys**
4	Permissions subsystem is used to enforce **Data Sharing Agreements** and allow **secure data sharing** between RiskBlock members
5	Membership Management is used to **onboard and manage memberships** within the RiskBlock
6	Identity subsystem establishes **uniform identities for RiskBlock entities** to allow shared references throughout the ecosystem

Technology Architecture

Technology Architecture

Platform Management will Be Required Across the Different Layers of the Technology Stack



Technology Stack

1. RiskBlock Governance APIs and Applications
2. RiskBlock Smart Contract SDK
3. RiskBlock Governance Services
4. Data Layer
5. Hosting Platform

Technology Details

1
- User Interface Layer for End User Interaction for both business and governance
- Container Management for API's
- Integration Gateway for Enterprise Applications and Oracles

2
- Smart Contracts for various Use Cases in Claims, Underwriting, Subrogation, etc.
- Certificate / Key Management for Users, Enterprises and Assets

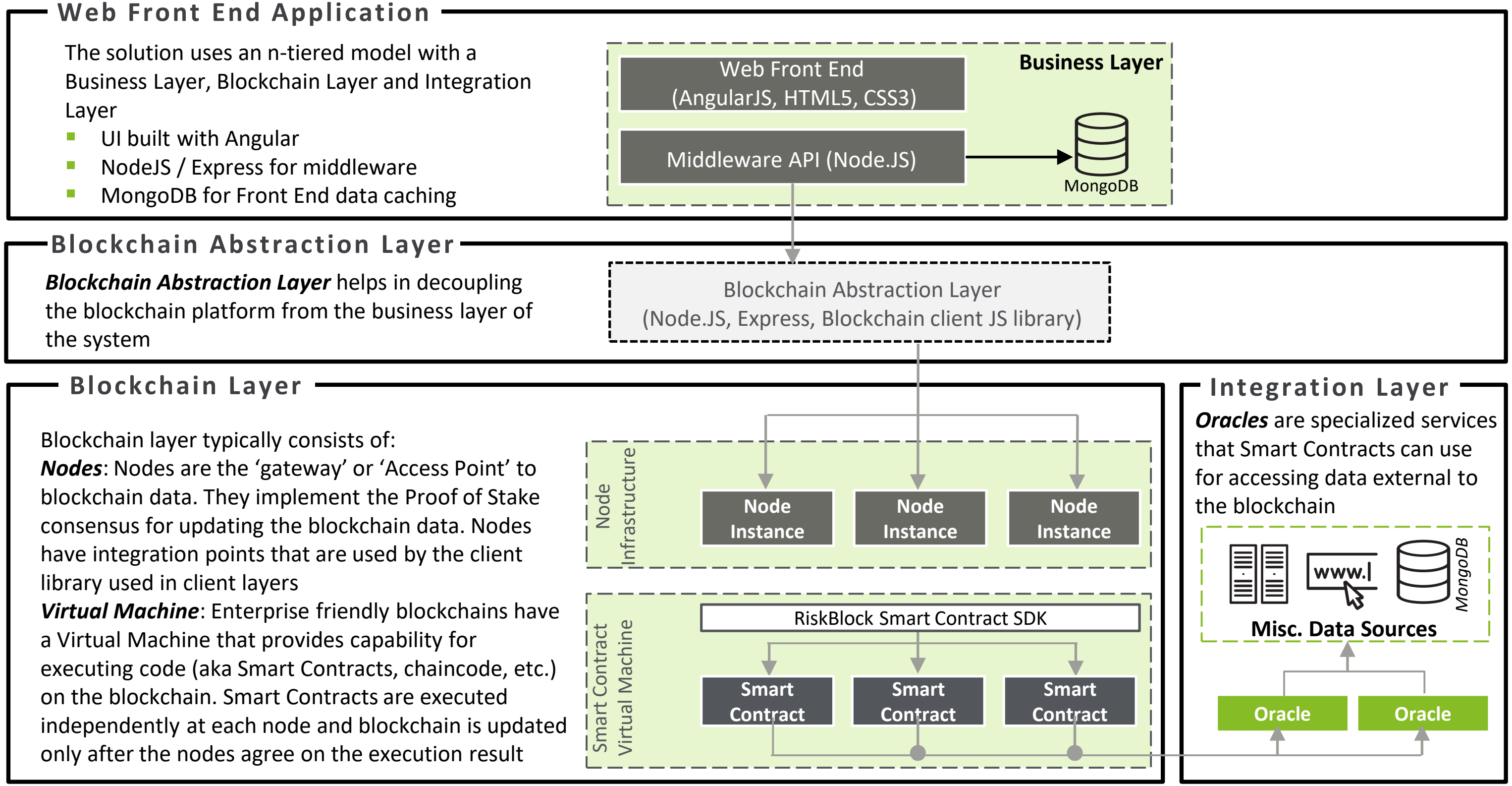
3
- RiskBlock chain building blocks for governance and interoperability for various use cases such as Claims, Underwriting, Subrogation, etc.
- Governance Blockchain for Identity Management, Member Services, Forms UI App etc.

4
- Blockchain Platforms - Ethereum, HyperLedger, COSMOS etc.
- Blockchain Add On such as IPFS, IPDB, etc.
- RiskBlock Data Lake
- Enterprise Data Stores

5
- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Provider (GCP)

Reference Application Stack

Business Applications on the RiskBlock will follow familiar development practices in use in various enterprises



Web Front End Application

The solution uses an n-tiered model with a Business Layer, Blockchain Layer and Integration Layer

- UI built with Angular
- NodeJS / Express for middleware
- MongoDB for Front End data caching

Blockchain Abstraction Layer

Blockchain Abstraction Layer helps in decoupling the blockchain platform from the business layer of the system

Blockchain Abstraction Layer
(Node.JS, Express, Blockchain client JS library)

Blockchain Layer

Blockchain layer typically consists of:

Nodes: Nodes are the 'gateway' or 'Access Point' to blockchain data. They implement the Proof of Stake consensus for updating the blockchain data. Nodes have integration points that are used by the client library used in client layers

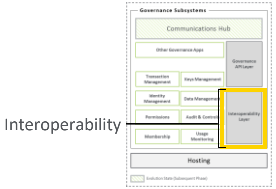
Virtual Machine: Enterprise friendly blockchains have a Virtual Machine that provides capability for executing code (aka Smart Contracts, chaincode, etc.) on the blockchain. Smart Contracts are executed independently at each node and blockchain is updated only after the nodes agree on the execution result

Integration Layer

Oracles are specialized services that Smart Contracts can use for accessing data external to the blockchain

Blockchain Architecture

Interoperability: Objectives & Assumptions



Architecture | **Interoperability** | Membership Management | Identity Management | Permissions Management | Keys Management | Information Management | Usage Monitoring

Why this is important

Blockchain Interoperability enables the **inter blockchain communication** and data sharing across RiskBlock subsystems

Objectives

Applications

- Enable inter blockchain **communication and data sharing**
- Incorporate leading solutions for **Blockchains Interoperability**

Data

- **Compliant with transaction standards** via Information Management subsystem
- Establish **data sharing** mechanism from private enterprise data stores

Governance

- **Phased delivery** approach for risk management
- **Modular architecture** ensures future upgrades
- **Permissions-aware** module

Technology

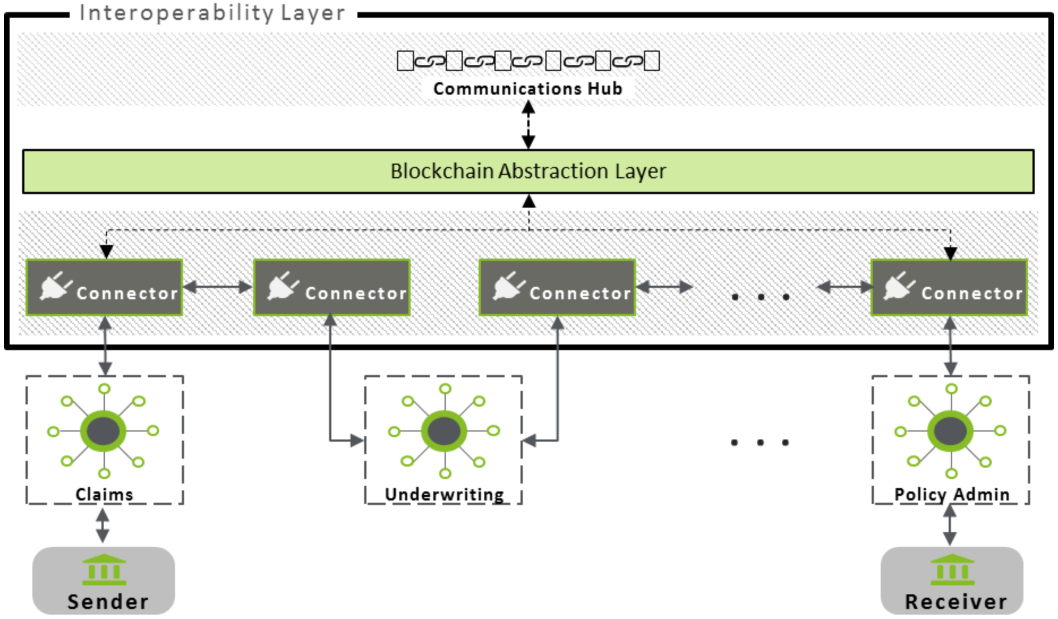
- Support both **push and pull messaging and event based** patterns
- **Interface driven integration** to avoid technology dependencies

Assumptions

- Business applications will need integration and data sharing with other applications on different RiskBlock blockchains
- Transactions consumed via Communications Hub will need to be reported via Usage Monitoring subsystem
- Only interoperability layer will be required for go live and the events blockchain will be rolled out in a later RiskBlock upgrade phase
- System events will need to be persisted for historical record keeping

Interoperability: Communications Hub

The Communications Hub and the Interoperability Layer enable cross blockchain communication across the all blockchain applications in the ecosystem



Interoperability Layer

Communications Hub

Blockchain Abstraction Layer

Connector

Connector

Connector

Connector

Claims

Underwriting

Policy Admin

Sender

Receiver

Interoperability Layer

- **Bi-directional Data sharing** across diverse blockchain population
- **Permissioned** data sharing
- Dynamic permissions management for representing inter organization **data sharing agreements**
- Modeled with **Connector pattern** to relay information between blockchains

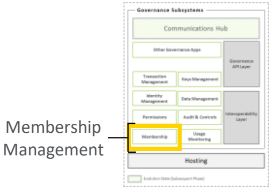
Communications Hub

- Communications Hub is a key **evolutionary step** for the RiskBlock Interoperability layer
- Encapsulates Interoperability Layer to **extend inter blockchain communication** for more powerful data sharing
- Supports **real time and historical events** information

Note

RiskBlock Interoperability design is independent of the blockchain products being used for different blockchains

Membership Management: Objectives & Assumptions



Architecture | Interoperability | **Membership Management** | Identity Management | Permissions Management | Keys Management | Information Management | Usage Monitoring

Why this is important

Membership Management is used to **onboard and manage memberships** within the RiskBlock ecosystem

Objectives

Applications

- **Onboard member firms** to RiskBlock post due diligence
- **Enable RiskBlock features** for onboarded member firms

Data

- Create **blockchain accounts** for member firms
- Establish **data sharing** mechanism from private enterprise data stores

Governance

- **Track membership status** for member firms
- **Modular architecture** ensures future upgrades

Technology

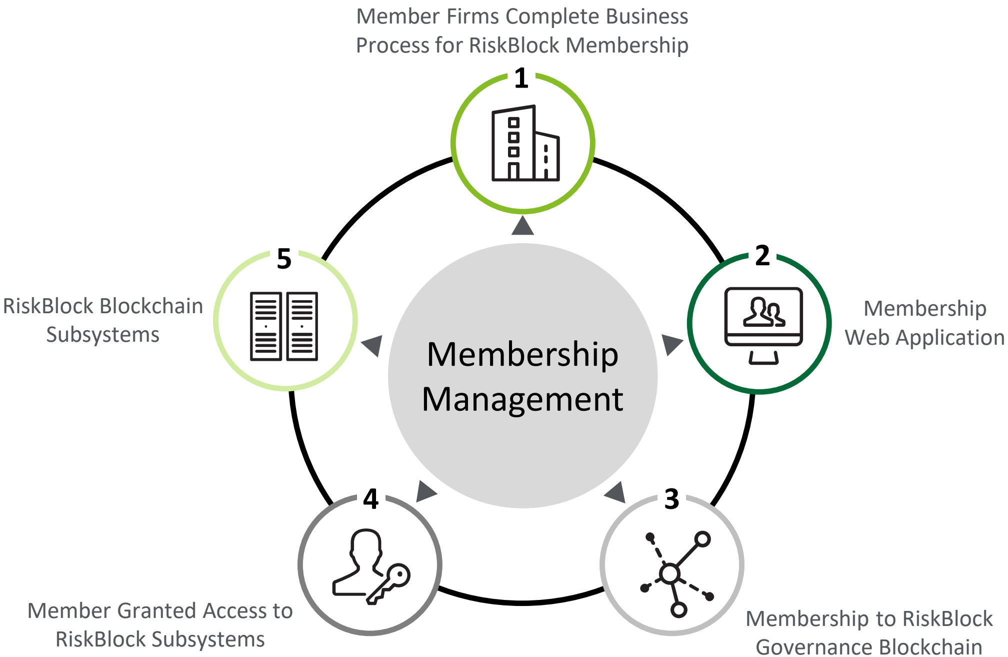
- **Web enabled** system for managing onboarding processes
- **Integration with other Governance** apps for holistic membership management

Assumptions

- Membership agreements will be stored on Membership services blockchain
- Usage limits will be viewable via Membership Management modules as reported by Usage & Monitoring subsystem
- Membership management will be a central application contained within the Governance Subsystem
- Member firms can participate in RiskBlock only with an active membership status

Membership Management: Solution Detail

Membership Management module will be a key part of RiskBlock Governance and will help manage membership status for all member firms


Membership Management
1
Member Firms Complete Business Process for RiskBlock Membership
2
Membership Web Application
3
Membership to RiskBlock Governance Blockchain
4
Member Granted Access to RiskBlock Subsystems
5
RiskBlock Blockchain Subsystems

Solution Detail

1 Membership to the Blockchain platform would be open to members and approved non members of The Institutes consortium upon completing the required **onboarding process** as defined by RiskBlock

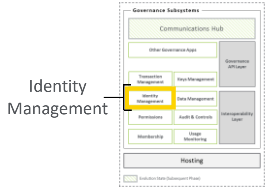
2 **Membership Management web application** will be used by RiskBlock for onboarding of member firms and ongoing membership services

3 Membership data to be stored on a **Membership Blockchain** in RiskBlock Governance Subsystem

4 Membership Management application will be **integrated with other subsystems** for ongoing membership services and usage monitoring

5 **Maintenance** of the Membership Management application will be required for any subsequent changes to membership rules

Identity Management: Objectives & Assumptions



Architecture | Interoperability | Membership Management | **Identity Management** | Permissions Management | Keys Management | Information Management | Usage Monitoring

Why this is important

Identity subsystem establishes **uniform identities for RiskBlock entities** to allow shared references within the ecosystem

Objectives

Applications

- RiskBlock ecosystem will benefit from **Unique identity** for various entities – Member firms, Insured Assets and Policy holders
- Enable easier **data sharing**

Data

- **Permissioned data access** to data attributes for each ID
- Onboarding process will create the required IDs for member firms

Governance

- **Insurable Asset** IDs would be generated in the context of business processing
- Detailed attributes for all IDs to be stored **off-chain for confidentiality**

Technology

- **Application integration** will be supported by ID Management APIs

Assumptions

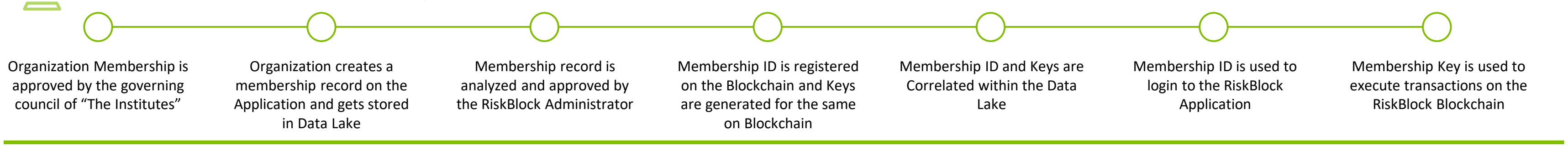
- All Member Firms would be on-boarded using the Membership Management Application
- ID and Identity data would be stored in the Data Lake

Identity Management: Generation Process

The following processes provide insight into ID generation and management for key RiskBlock entities:

- **Member firms** - Onboarding process generates their identities on RiskBlock
- **Insurable Assets** - These IDs will get generated via various different business processes

Member Firm ID Generation process



Organization Membership is approved by the governing council of “The Institutes”

Organization creates a membership record on the Application and gets stored in Data Lake

Membership record is analyzed and approved by the RiskBlock Administrator

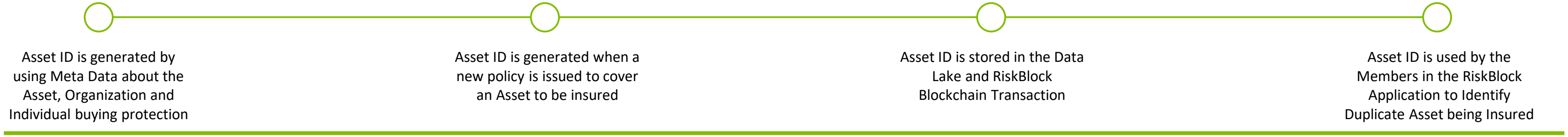
Membership ID is registered on the Blockchain and Keys are generated for the same on Blockchain

Membership ID and Keys are Correlated within the Data Lake

Membership ID is used to login to the RiskBlock Application

Membership Key is used to execute transactions on the RiskBlock Blockchain

Insurable Asset ID Generation process



Asset ID is generated by using Meta Data about the Asset, Organization and Individual buying protection

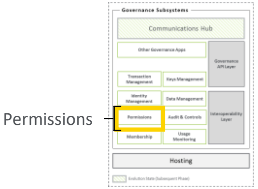
Asset ID is generated when a new policy is issued to cover an Asset to be insured

Asset ID is stored in the Data Lake and RiskBlock Blockchain Transaction

Asset ID is used by the Members in the RiskBlock Application to Identify Duplicate Asset being Insured

Permissions Management: Objectives & Assumptions

Architecture | Interoperability | Membership Management | Identity Management | **Permissions Management** | Keys Management | Information Management | Usage Monitoring



Why this is important

Permissions subsystem enforces **Data Sharing Agreements** and allows **secure data sharing** between RiskBlock members

Objectives

Applications

- Management of **Data Security Policies** at member firm level for data on RiskBlock

Data

- Protects all **transaction data** irrespective of storage location
- **Off-Chain processing** for increased confidentiality

Governance

- **Permissions framework** for Data Sharing Agreements
- Member Firms to manage permissions via **Permissions Application**

Technology

- **Real time** permissions updates across RiskBlock
- **High performance** system for fast system throughput

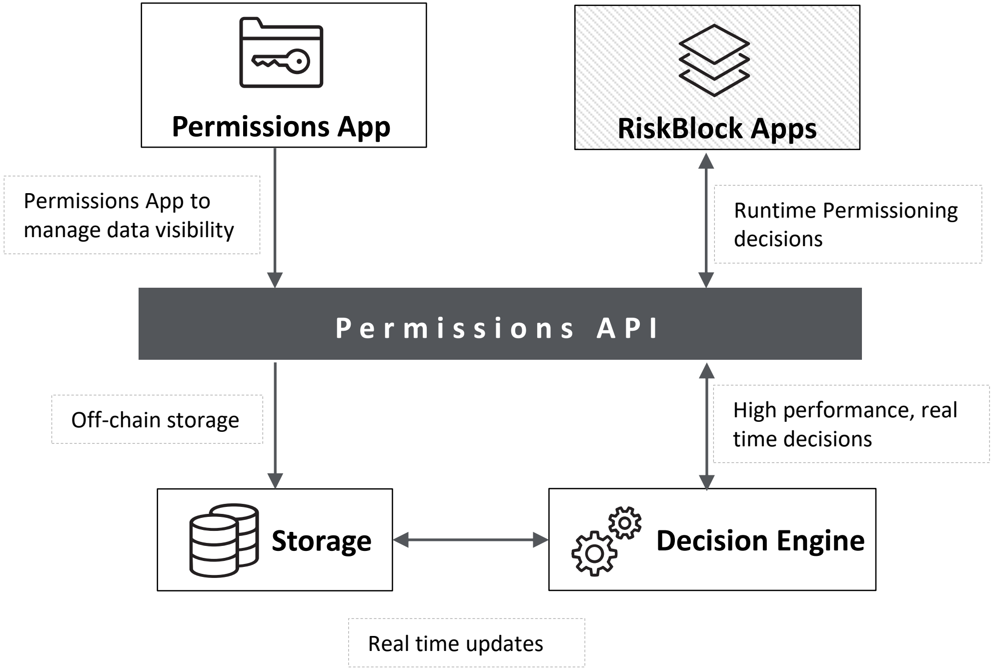
Assumptions

- Member firms would self manage their Data Security and Sharing policies vis-à-vis other member firms.
- Member firms would grant Permissions to other member firms as per their Data Sharing Agreements**.** **Data Sharing Agreements** are mutual agreements between member firms that define the level of access each has to the other's confidential data
- Permissions can be applied only at member firm level
- Permissions are confidential information and will need protection

Permissions Management: Solution Detail

Architecture | Interoperability | Membership Management | Identity Management | **Permissions Management** | Keys Management | Information Management | Usage Monitoring

Permissions Management is a critical part of RiskBlock and acts as security agent in all Data Sharing processes



Permissions App

RiskBlock Apps

Permissions App to manage data visibility

Runtime Permissioning decisions

Permissions API

Off-chain storage

High performance, real time decisions

Storage

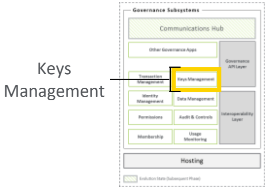
Decision Engine

Real time updates

Solution Detail

- RiskBlock provides a web based **Permissions Management application** to member firms to setup **data sharing agreements** with other firms
- Permissions APIs expose permissions for all the Subsystems to **connect and verify the permissions** for data access, as needed
- All Permissions would be stored in an **off-blockchain Permissions Store**
- A **high performance**, Permissions **Decision Engine** will evaluate access permissions and allow access to confidential data ONLY to authorized requests
- Updates to permissions will be effective in the ecosystem components in **real time**

Keys Management: Objectives & Assumptions



Architecture | Interoperability | Membership Management | Identity Management | Permissions Management | **Keys Management** | Information Management | Usage Monitoring

Why this is important

Keys Management APIs used to **securely store and use confidential blockchain keys** for signing transactions

Objectives

Applications

- Creation of **Private and Public Keys** for blockchain accounts
- **Secure transaction signing** with account keys

Data

- **Store and use** Keys securely
- **Reusable** across wide spectrum of blockchain products

Governance

- **Hardware Security Modules** (HSM) based key security
- **Secure Machine to Machine** interface

Technology

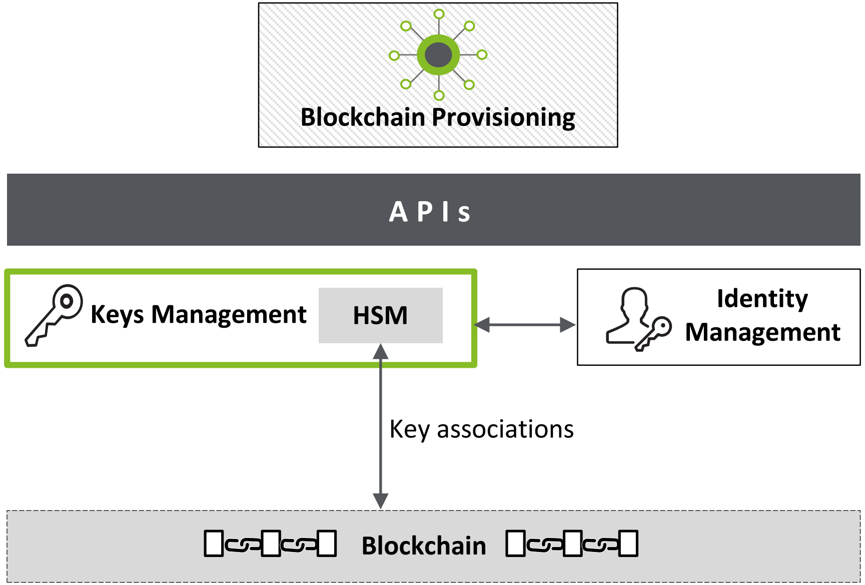
- **Cloud enabled**
- Wide range of supported **cryptographic standards** (SHA-256, SHA-512 etc)

Assumptions

- Account keys are required to transact successfully with the RiskBlock blockchains
- Account Keys will be specific to member firm and the blockchain
- Applications to have access to HSM appliances

Keys Management: Creation and Storage

Keys Management ensures confidentiality of member firms’ Blockchain keys



Blockchain Provisioning

APIs

Keys Management HSM

Identity Management

Key associations

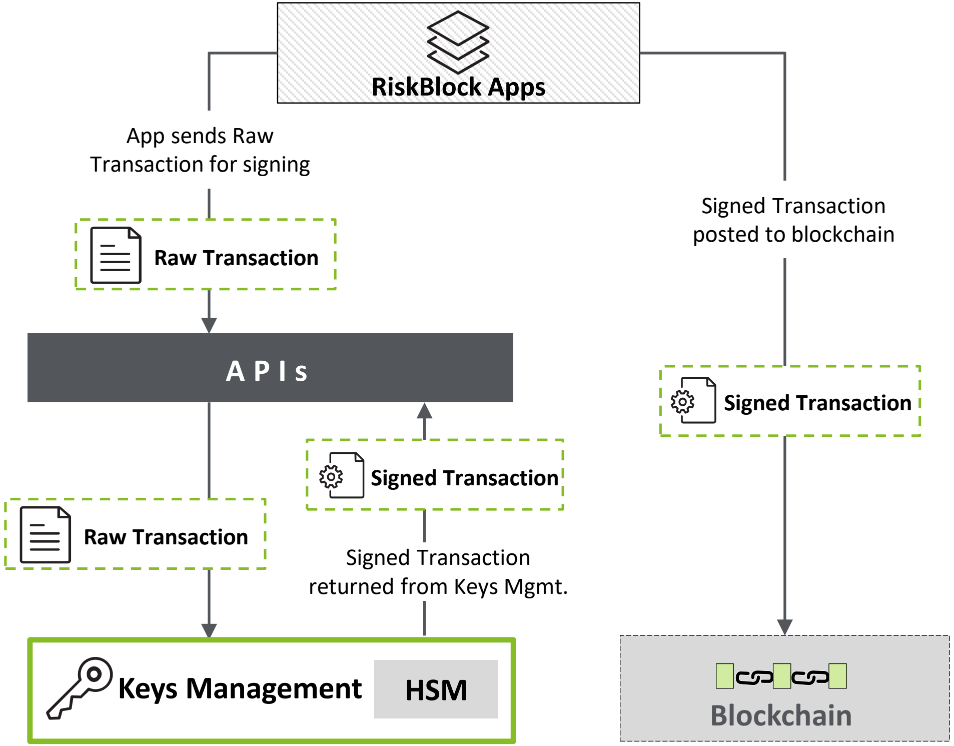
Blockchain

Key Creation and Storage

- Account Keys are extremely confidential and will be stored in **specialized** hardware called **Hardware Security Module** (HSM) for best of class security
- Keys once stored on HSM, will never leave the HSM
- Blockchain provisioning tools will also **generate the account keys** for all member firms and update them in HSM
- Account Keys are **exclusive to RiskBlock member firms** and are used for **Identification and Non Repudiation** of member firms
- Neither Governance nor Business Applications will ever need to extract Keys from the HSM, to comply with **RiskBlock Security Guidelines**
- Account Keys will be associated with the member firm identities while in HSM

Keys Management: Transaction Signing

Architecture | Interoperability | Membership Management | Identity Management | Permissions Management | **Keys Management** | Information Management | Usage Monitoring



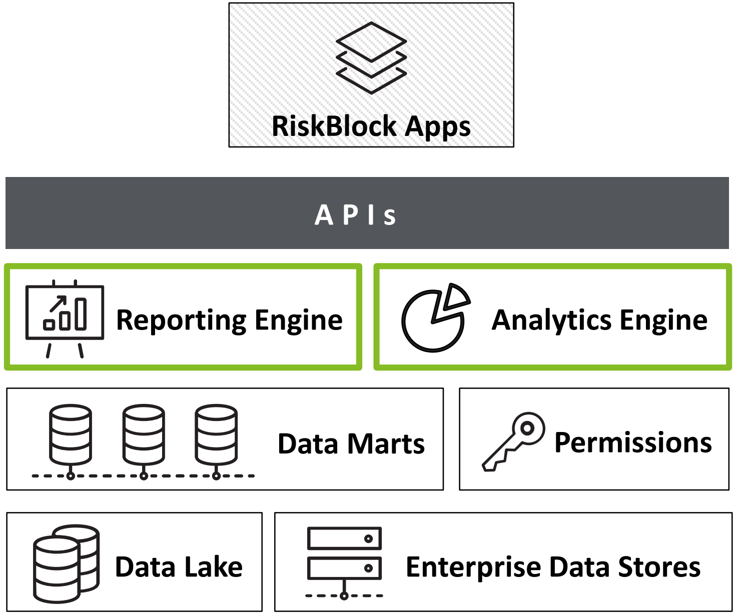
RiskBlock Apps
App sends Raw Transaction for signing
Raw Transaction
APIs
Raw Transaction
Signed Transaction
Signed Transaction returned from Keys Mgmt.
Keys Management
HSM
Signed Transaction posted to blockchain
Signed Transaction
Blockchain

Transaction Signing

- As an overarching **Security Guideline**, Account Keys once stored on HSM will never leave the HSM
- Keys Management module exposes a **robust and lightweight API** for application integration
- Applications will need to **sign transactions** via Key Management APIs before posting to respective blockchains
- **High performance** to ensure high system throughput

Information Management: Reporting and Analytics

Reporting and Analytics provide key insights into business and technical operations within the RiskBlock ecosystem



Reporting and Analytics

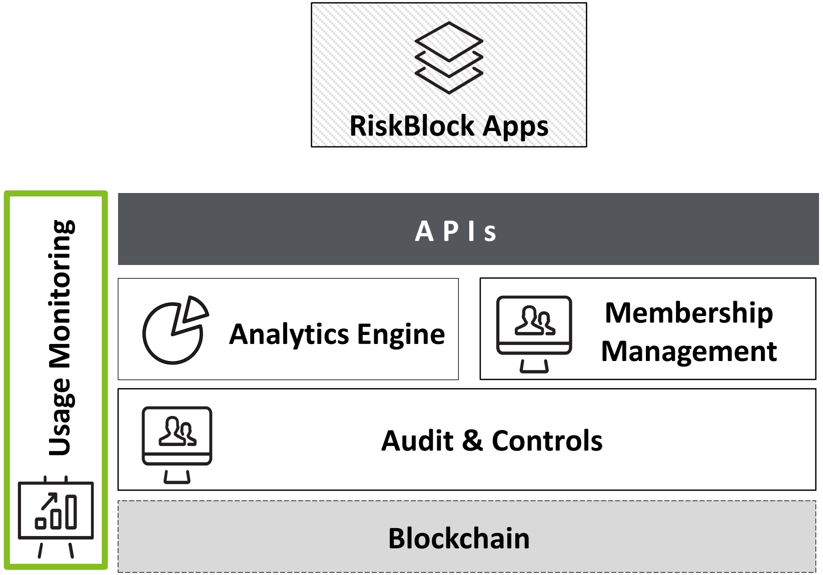
- **Reports Engine** to deliver custom, on-demand and periodic reports to RiskBlock Governance and member firms
- Reporting Engine can be used to setup **custom reports generation** by all RiskBlock member firms
- Reporting data will span repositories within **RiskBlock Data Lake and Enterprise data stores**
- Integration with Permissions for **data confidentiality**
- **APIs for integration** with RiskBlock applications
- Integration **support for external Data analysis tools** e.g. Audit & Controls, Fraud detection, Customer behavior pattern analysis, Predictive Analysis, etc.

Important: Audit & Controls are a specialized set of reports and analytics based alerts to target Compliance, Regulatory or Fraud detection processes

Usage Monitoring: Solution Detail

Architecture | Interoperability | Membership Management | Identity Management | Permissions Management | Keys Management | Information Management | **Usage Monitoring**

Accurate tracking and monitoring of each member firm's usage volume and pattern is critical to support several processes in RiskBlock Governance



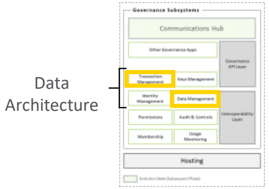
Usage Monitoring

- Usage Monitoring subsystem will **monitor and track** the metrics for Member Firm's **usage patterns and volume** of RiskBlock platform
- Usage Monitoring will **cover all subsystems** of the ecosystem
- Monitoring metrics will be **shared with Analytics Engine and Membership Management** subsystems as input into RiskBlock **Governance processes**, e.g. Cost Chargeback process
- Integration with **Audit & Controls reporting** within the Governance subsystem for Compliance, Regulatory, Fraud detection etc
- Monitoring data can be tracked in **real time or in time slices** as per the criticality of the usage metric datasets

Important: Audit & Controls are a specialized set of reports and analytics based alerts to target Compliance, Regulatory or Fraud detection processes

Data Architecture

Data Architecture: Objectives & Assumptions



Data Architecture → Transaction Standards → Data Storage → Data Retrieval → Transaction Inventory

Why this is important

Defines **standardized transaction** formats to support **data sharing** within RiskBlock and **secure data storage and retrieval**

Objectives

Applications

- Process **standardized transactional** and reference data in RiskBlock
- Support for **storage, retrieval and data exchange** between business applications

Data

- Support for **CRUD operations** on a wide variety of **diverse data types**
- **Data translators** ensure **transaction standards** compliance

Governance

- **Data Confidentiality** secured by permissions
- Compliance with **transaction standards and ensuring high Data Quality**

Technology

- **Integration** with Reporting, Audit & Control and Analytics
- Data stored in **Data Lake** with the blockchains storing **verification hashes**

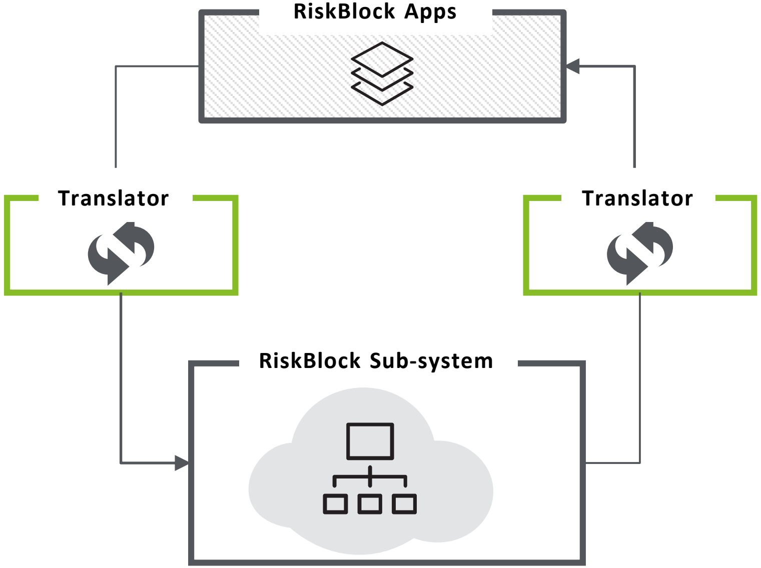
Assumptions

- Transaction data standards will be defined
- Enterprises to host confidential data in private stores
- Transactional data will accumulate over time
- With increased member firms Risk Block storage capacity will be scaled

Data Architecture: Transaction Standards

Data Architecture → **Transaction Standards** → Data Storage → Data Retrieval → Transaction Inventory

Translator components act as **data quality** inspectors and validate all transactions for compliance with the defined Transaction Standards

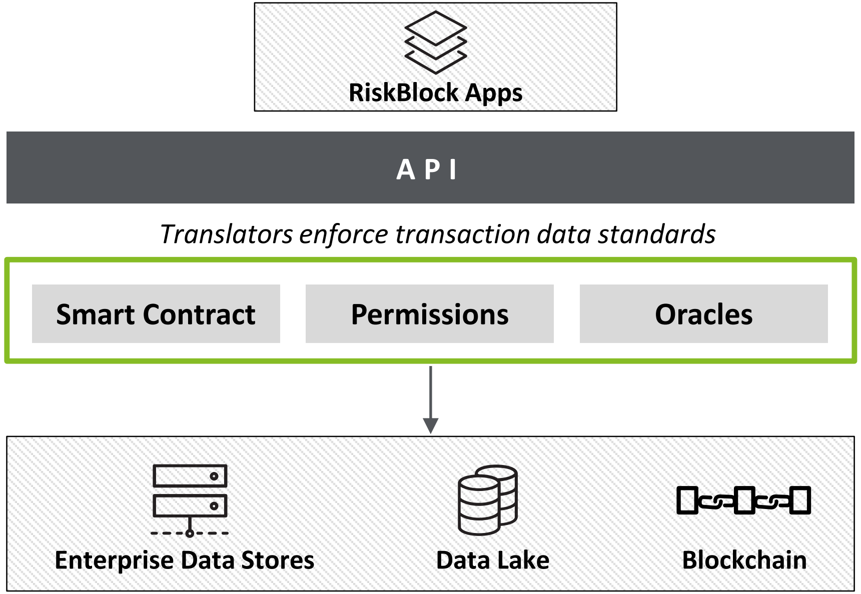


Transaction Standards

- Transaction Standardization enables **data exchange** in the proposed architecture
- **Facilitates interoperability** in the RiskBlock ecosystem
- **Enforce common** business transaction vocabulary
- **Translator** components act as **data quality** inspectors and validate all transactions for compliance with the defined Transaction Standards
- **Non Compliant transaction data is detected** and blocked from polluting the RiskBlock data repositories
- **Data Integrity checks** on inbound and outbound transaction traffic ensures good data quality on all data processing operations within RiskBlock

Data Architecture: Data Storage

Data Storage provides a consolidated view of standardized ‘inbound’ transactional data that is physically spread over 3 distinct types of repositories



Data Storage

Data flowing through RiskBlock will be stored in 3 separate repositories.

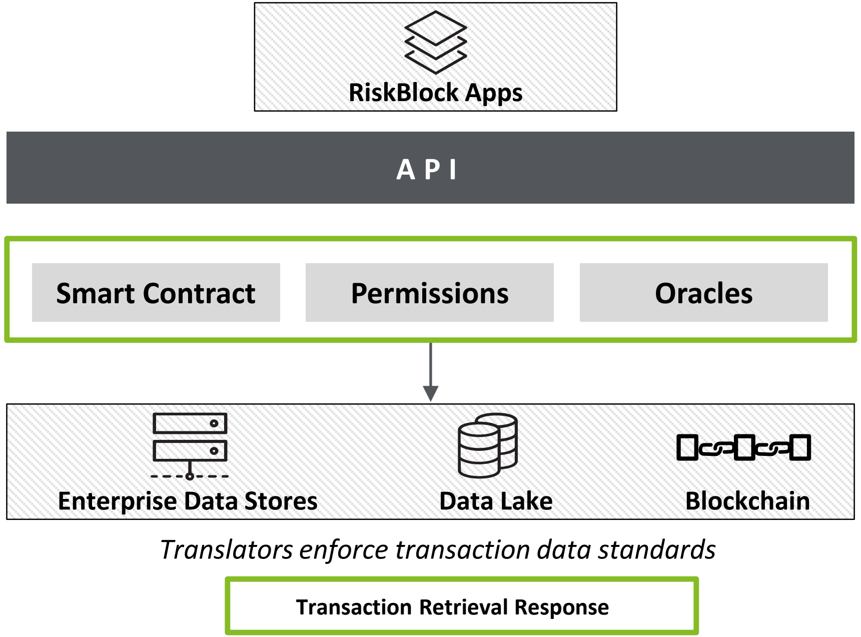
- **Confidential Data**: Organization owned confidential data within respective enterprises. Member firms will need to expose that data, as per their agreements with other members, via **Data APIs** hosted at their end.
- **Transactional Data:** To store shareable transactional information, IDs, Asset references in Data Lake.
- **Block chain Data:** One-way hash of information store is maintained on the Blockchain to validate off-chain confidential data

Off Chain Translator Components ensure Transaction standardization is followed for all inbound transactions

Data Architecture: Data Retrieval

Data Architecture › Transaction Standards › Data Storage › **Data Retrieval** › Transaction Inventory

Data Retrieval provides a consolidated view of standardized ‘outbound’ transactional data that is physically spread over 3 distinct types of repositories



Translators enforce transaction data standards

Data Retrieval

- Data Retrieval operations will utilize the Permissions API to validate the data sharing agreements
- Transaction Standardization enables data exchange in the proposed architecture
- Enables interoperability in the **RiskBlock** ecosystems
- Establish common **RiskBlock** vocabulary
- Allow data share among member firms
- Enable seamless value-exchange of information among network of block chains, data lakes and private stores.

Off Chain Translator components ensure Transaction standardization is followed for all incoming data

Data Architecture: Transaction Inventory

The following table illustrates the core transactions that make up the four founding use cases. These transactions will serve as the building blocks of RiskBlock as they define the data flow and dependencies among multiple stakeholders and systems collaborating on RiskBlock[1]

	Read Policy Details	Update Policy Details	Read Claim Details	Write Claim Details	Notify Claim Details	Read Driving License Details	Read Historic Rain Fall Data	Read Payment Details	Write Payment Balance	Notify Billing Request
First Notice of Loss	✓	✓	✓	✓	✓					
Proof of Insurance	✓	✓				✓				
Parametric Insurance[2]	✓		✓	✓	✓		✓			
Subrogation	✓		✓	✓	✓			✓	✓	✓

[1] The 10 listed transactions are considered as an input for Transaction Standardization work stream Phase II of the project.

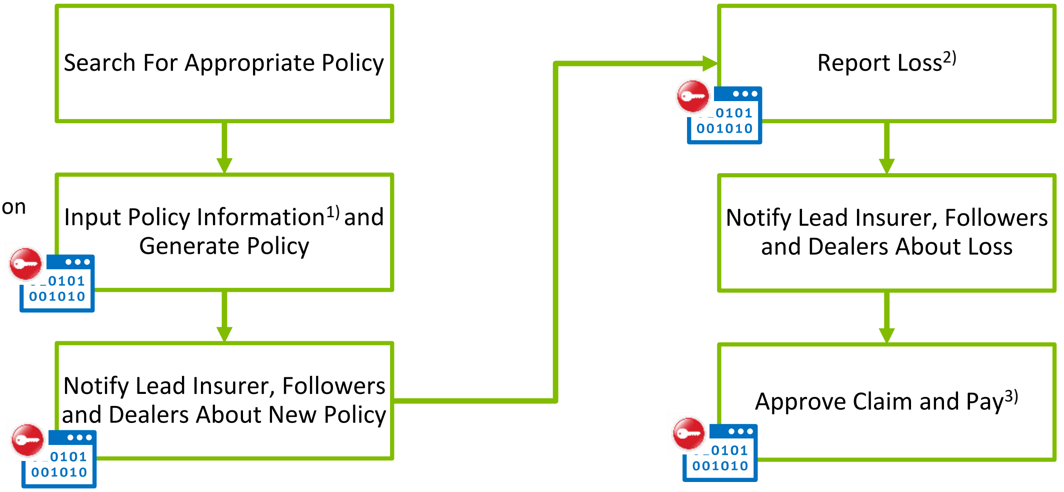
[2] Additional transactions for all use cases (e.g. “Write Policy Information” in Parametric Insurance) will be considered in the Design & Build phase.

First Notice of Loss: Business Process Flow

Permissions Protected, Standardized Transactions

Create Policy

First Notice of Loss



Search For Appropriate Policy

Input Policy Information[1] and Generate Policy

Notify Lead Insurer, Followers and Dealers About New Policy

Report Loss[2]

Notify Lead Insurer, Followers and Dealers About Loss

Approve Claim and Pay[3]

Standardized Transaction created:
- **Policy**

Trigger an e-mail notification to related parties based on contact information stored in blockchain

Standard Transaction:
- **Claim**

Trigger an e-mail notification to related parties when policy record is updated in blockchain

[1] Information includes: Insured, VIN, Date, Lead Insurer, Followers, Dealer, Policy Number, Car Manufacture Year, Car Make, Car Model, etc.

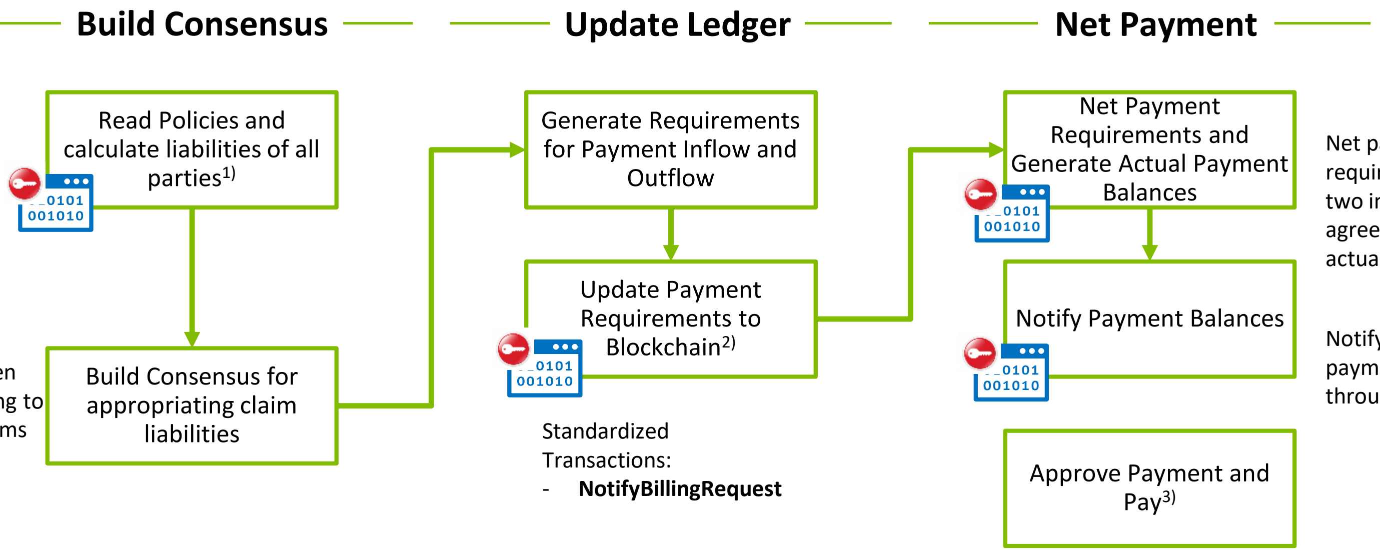
[2] Loss information includes date, description of loss, contact information. etc.

[3] Currently payment is an off-blockchain function

ILLUSTRATIVE

Subrogation: Business Process Flow

Permissions Protected, Standardized Transactions



Build Consensus

Standardized Transactions:
- **Policy**
- **Claim**

Read Policies and calculate liabilities of all parties[1]

Build Consensus for appropriating claim liabilities

Negotiation between stakeholders, leading to final agreement terms

Update Ledger

Generate Requirements for Payment Inflow and Outflow

Update Payment Requirements to Blockchain[2]

Standardized Transactions:
- **NotifyBillingRequest**

Net Payment

Net Payment Requirements and Generate Actual Payment Balances

Net payment requirements between two insurers based on agreed cadence. Generate actual amount of payment

Notify Payment Balances

Notify insurers about payment balances through oracle

Approve Payment and Pay[3]

1) Multiple providers accessing cross organization policy and claims data

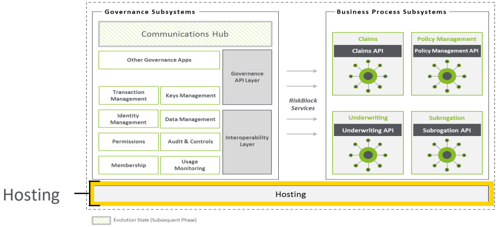
2) Only payment requirements are recorded in blockchain. Actual payments will happen later netting of payment requirements

3) Currently payment approvals and payments are done manually

ILLUSTRATIVE

Hosting Architecture

Hosting Architecture: Objectives & Assumptions



Governance Subsystems

Communications Hub

Other Governance Apps

Governance API Layer

Transaction Management

Keys Management

Identity Management

Data Management

Permissions

Audit & Controls

Membership

Usage Monitoring

Interoperability Layer

RiskBlock Services

Business Process Subsystems

Claims

Claims API

Policy Management

Policy Management API

Underwriting

Underwriting API

Subrogation

Subrogation API

Hosting

Hosting

▨ Evolution State (Subsequent Phase)

Key infrastructure objectives and assumptions are considered for availability, scalability, reliability, flexibility, security and configurability

Objectives

Applications

- **Hosting Strategy** for RiskBlock components
- Enable **Monitoring and Analytics** for API and infrastructure usage

Data

- Ability to withstand **data loss or corruption**
- **High Scalability** across entire RiskBlock ecosystem

Governance

- An **Open Platform** to allow enterprises to easily migrate on or off
- **Authenticate & authorize** access to resources

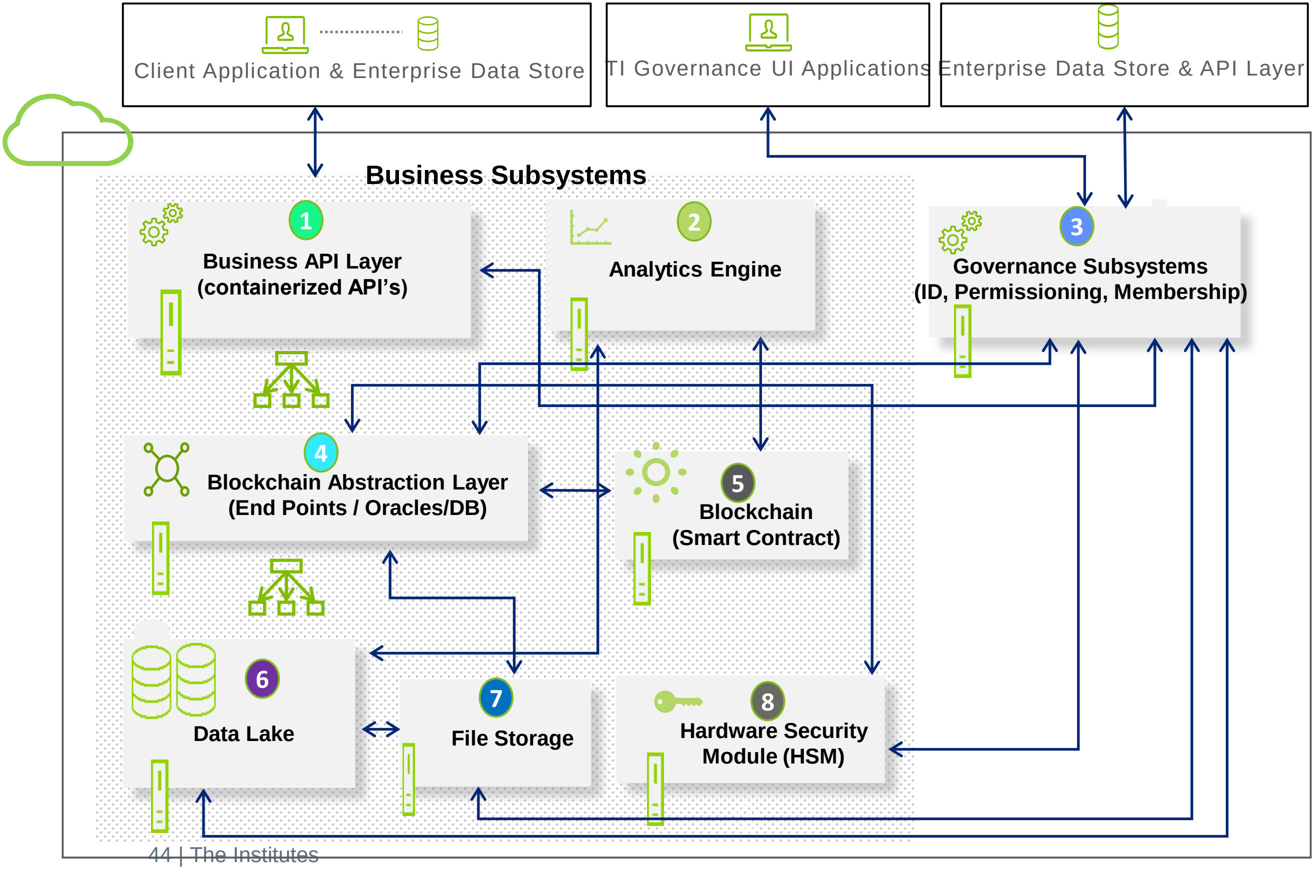
Technology

- Infrastructure to enable **scalability, agility and availability**
- **Configurability** of multiple application programs to share data where diverse applications communicate either directly or through third-party software

Assumptions

- During the detailed design phase specific cloud provider will be considered based on the maturity of Blockchain integration and frameworks
- All private and public keys are generated and managed in the cloud
- Detailed sizing of the initial servers to be determined at implementation based on the volume and number of customers
- All servers to be clustered and effective load balancing techniques to be considered

Hosting Architecture: Conceptual Architecture



Client Application & Enterprise Data Store

TI Governance UI Applications

Enterprise Data Store & API Layer

Business Subsystems

1 Business API Layer (containerized API's)

2 Analytics Engine

3 Governance Subsystems (ID, Permissioning, Membership)

4 Blockchain Abstraction Layer (End Points / Oracles/DB)

5 Blockchain (Smart Contract)

6 Data Lake

7 File Storage

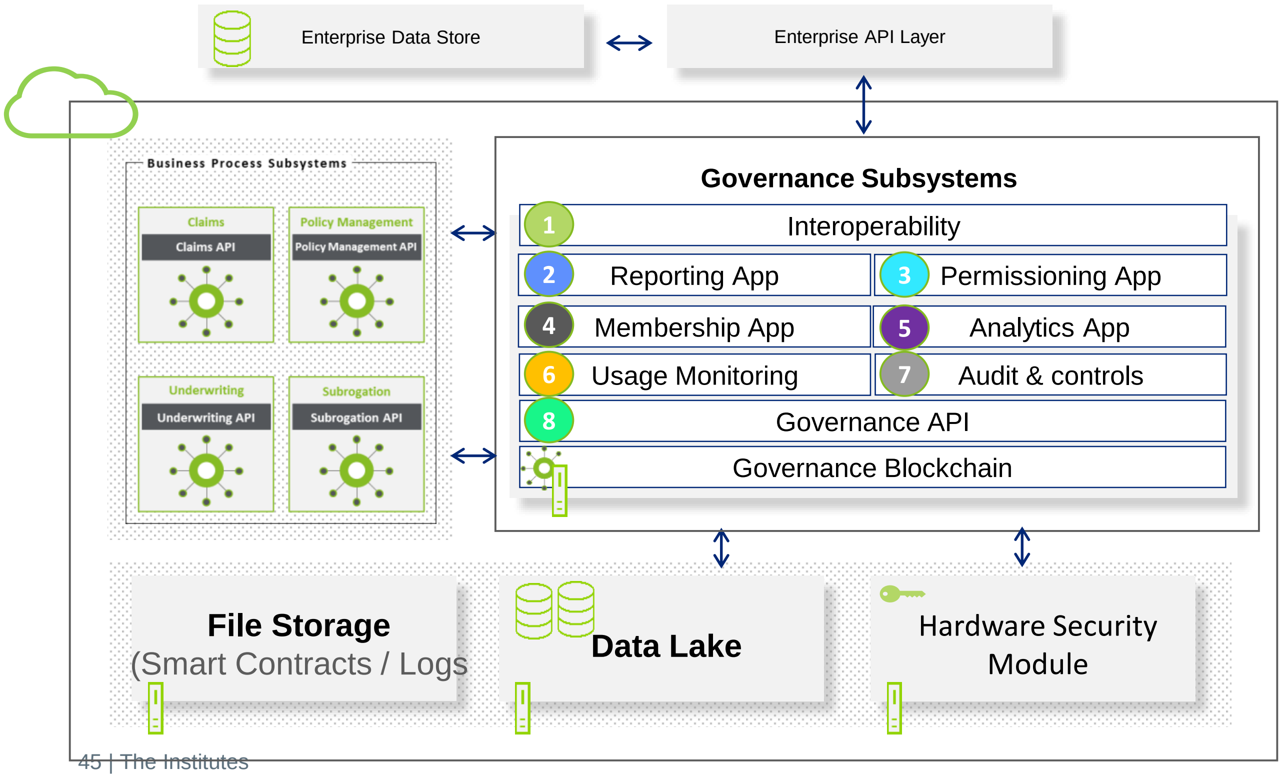
8 Hardware Security Module (HSM)

Label	Functionality
1	Business API Layer helps in **decoupling the underlying Blockchain platform** from the business layer of the system for better Platform isolation. This layer exposes the functional use cases such as onboarding, claims etc., and is containerized for scaling, clustered and load balanced. This layer connects to the Blockchain via API's exposed by the Blockchain Abstraction layer
2	Hosts the **analytics engine** that uses data from Blockchain and data lake
3	**Governance module** manages keys, Permissioning, Membership API's for the Blockchain consortium. e.g. create private/public keys, transaction signing etc.,
4	Blockchain Abstraction Layer (BAL) Integrates Business API requests to **blockchain end points, Smart Contract, Oracles, Data Lake.** Key feature includes manage Blockchain transactions, data integration, Oracles services for Smart Contract etc.,
5	Blockchain to **store transactions** across all member firms. Encrypted data would be stored in Blockchain
6	**Data Lake** will maintain ID's for assets, various transactions and co-relation between various entities
7	**Stores logs, and other unstructured data** such as configurations etc.,
8	Hardware security module (HSM) to generate and securely store **private and public keys** in the Cloud

Note: Governance and Business Subsystems will be built on the cloud platform of choice e.g. AWS, Azure, GCP

Hosting Architecture: Governance Framework

Governance Subsystem is composed of APIs and Applications that provide underlying services and utility functions for all applications built in RiskBlock



Enterprise Data Store

Enterprise API Layer

Business Process Subsystems

Claims

Claims API

Policy Management

Policy Management API

Underwriting

Underwriting API

Subrogation

Subrogation API

Governance Subsystems

1 Interoperability

2 Reporting App

3 Permissioning App

4 Membership App

5 Analytics App

6 Usage Monitoring

7 Audit & controls

8 Governance API

Governance Blockchain

File Storage

(Smart Contracts / Logs

Data Lake

Hardware Security Module

Label	Functionality
1	Enables the **interoperability** of inter blockchain communication and data sharing
2	Reporting app interacts with the reports engine to deliver **custom, on-demand and periodic reports** to the Institute and member firms
3	Member Firms use the permissioning app to **manages permissions** with other member firms
4	Membership Management app is used to **onboard and manage memberships** within the RiskBlock
5	Analytics app interacts with the analytics engine to generate **operational and executive dashboards** analytics
6	Usage monitoring tracks member firms **usage volume** for facilitating governance processes for charge back
7	Audit & controls enforces **compliance and regulation**
6	Governance API layer **abstracts the different governance management** functionalities such as Permissioning, Membership etc., for the Blockchain consortium

Glossary of Technical Terms

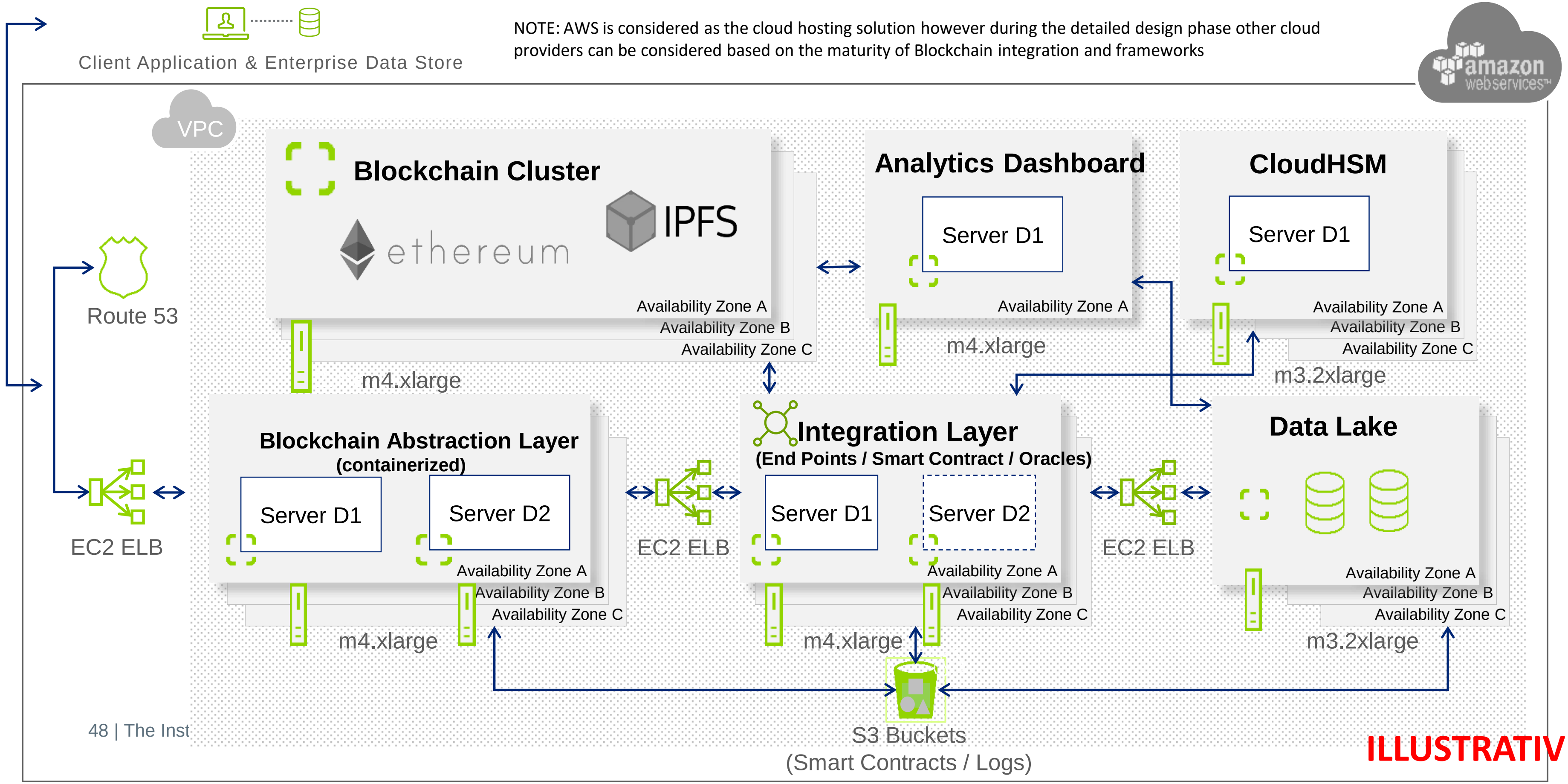
Glossary	
Application Programing Interface (API)	API’s provides definitions, protocols and tools to integrate disparate systems and technologies allowing the solution to seamlessly integrate through a variety of interface methods e.g., business services, governance services, platform services
Hardware Security Module (HSM)	HSM is a physical computing device that safeguards and manages digital keys for strong authentication and provides crypto processing
Data Lake	The Data Lake holds vast amount of raw data in its native format, both structured and unstructured
Analytics	The Analytics component is a software that improves efficiency and effectiveness in transforming and analyzing data and help identify risks on an ongoing basis e.g. Audit & Control, Fraud detection, Customer behavior pattern analysis, Predictive Analysis etc.,
Microservices	Greater modularity, loose coupling, and reduced dependencies in simplifying the integration tasks
Private Key	The private key is a secret key that is used to decrypt the message and allows to digitally sign transactions
Public Key	Public key is generated from the private key and allows to digitally verify transactions

Appendix

Assumptions

Hosting Architecture: AWS

NOTE: AWS is considered as the cloud hosting solution however during the detailed design phase other cloud providers can be considered based on the maturity of Blockchain integration and frameworks



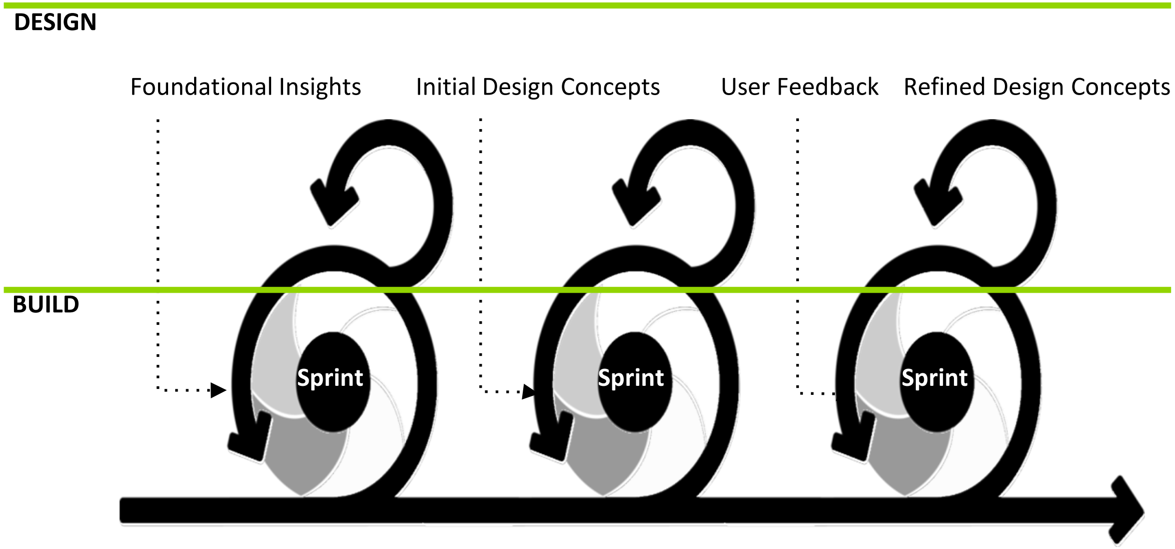
ILLUSTRATIVE

Agile Execution: Delivering the ‘ART OF THE POSSIBLE’

Delivering using Agile allows for iteratively test and incorporate user feedback

The Agile Approach

- Estimated scope with fixed time and resources
- Requirements (User Stories) are guided by initial view of Themes and elaborated during Sprints
- Sprints organized by User Stories which are sized, prioritized, and measured by User Stories Points
- Use of Sprints creates a flexible process that can easily ingest changes
- Highly collaborative and cohesive cross-functional teams across Design and Build work streams interact regularly during Sprints (daily stand-up calls, ad-hoc design sessions)
- Vertical slice(s) of functionality are completed during each Sprint to create demonstrable product that can be tested with users



Agile: Anatomy of a Sprint

Each Sprint uses interplay between Design and Build to advance Themes from insights to demo

DESIGN TEAM

Gather insights

- Ongoing generative and evaluative research in the field
- End of day debriefs to capture immediate insights to share with broader team

Refine ideas

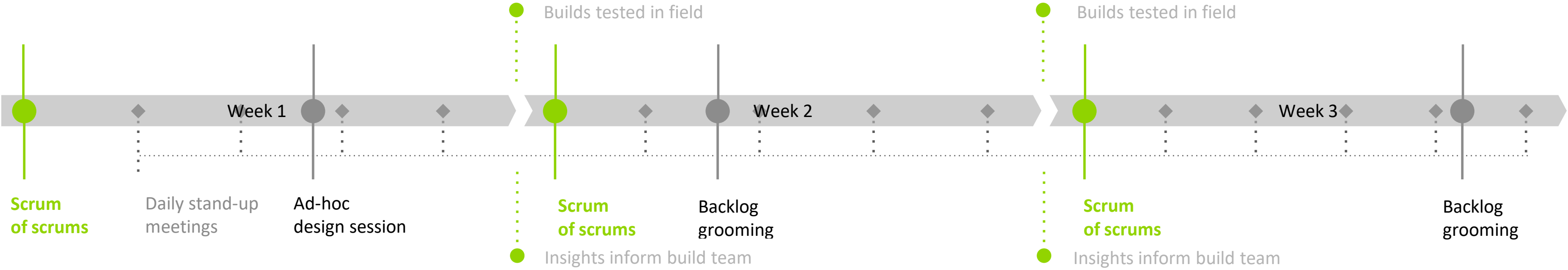
- Evaluative testing of emerging features and functions from early Sprints
- Evaluative testing of user interfaces and language system

Synthesize insights

- Cluster insights and identify implications for build team
- Synthesize insights from previous Sprints and new generative research output

Inform next Sprint

- Create potential options to test during next Sprint
- Define User Stories to feed into grooming and backlog



BUILD TEAM

Begin Sprint

- Sprint design session
- Backlog grooming

[Weekly cycle repeats]

End of Sprint

- Sprint Retrospective
- Plan next Sprint
- Demo