Blockchain Architecture Framework

Blockchain Reference Architecture Framework

The blockchain reference architecture frameworks provides supporting guidelines and tools for solution architecture development

Jump Start

Having an overview on the essentials to quickly start with Blockchain

Guideline

The framework serves as a basic guideline in setting up blockchain technology in an intra and extra organizational context

Library

Q3/2018

A collection of Blockchain pattern for various architecture domains, use cases and architecture domains. It supports the re-use of existing approaches and technologies

Best Practices

A reference architecture provides experience and proven approaches. Design pattern support the re-use of existing and proven knowledge

Methods & Tools

Surround selection and provision of best in class methods and tools

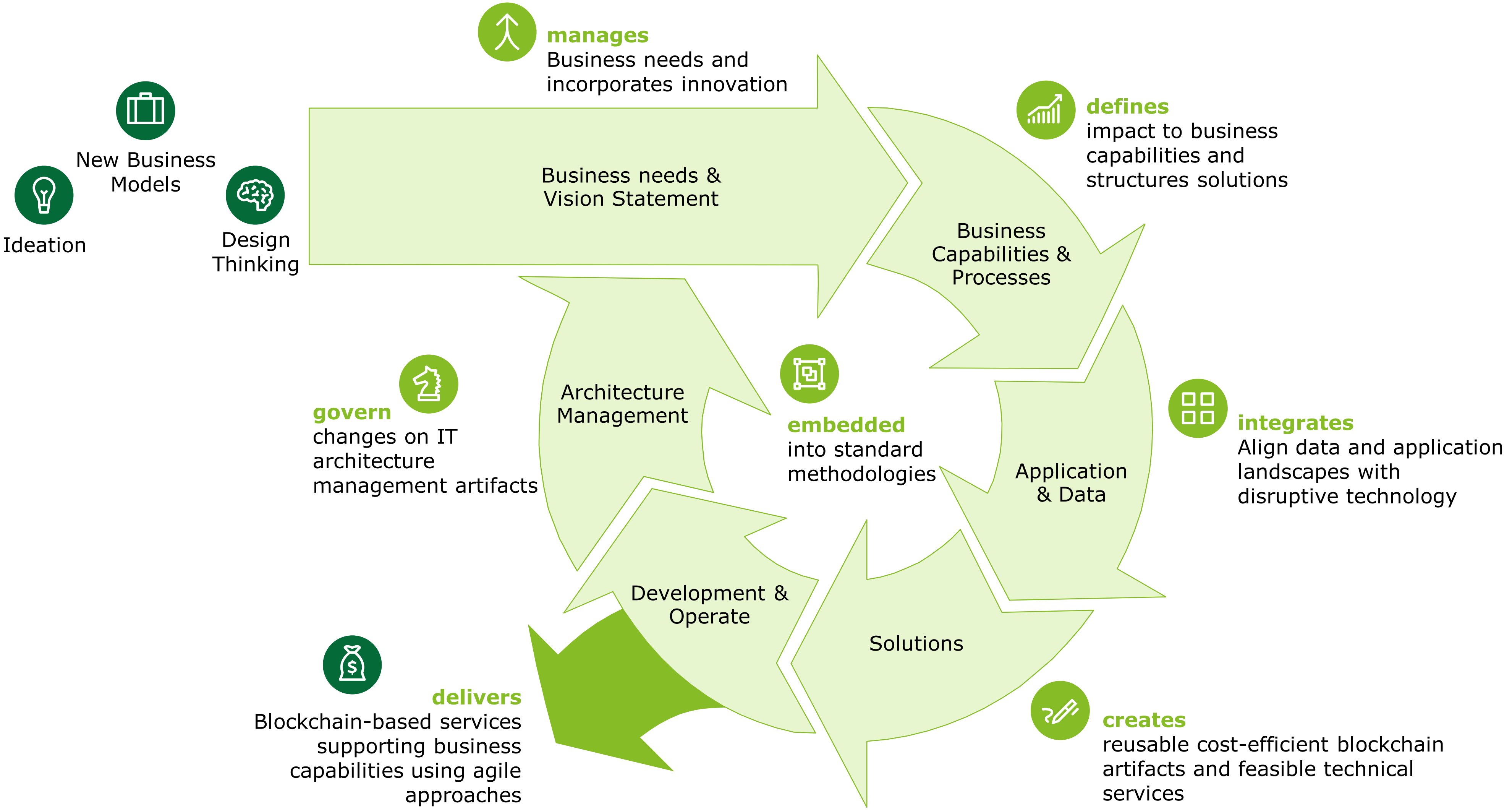
Maturity & Risk Assessment

Q4/2018

Discovery of the current level of maturity regarding the implementation and use of Blockchain. Consideration of risks in implementing and using Blockchain as a technology

Architecture driven approach to Blockchain

Making best use of Blockchain within ones organization requires a structured and fully proven approach – an architecture driven one



The basics about Blockchain

The fundamental concepts and DLT reshapes the roles of architects and requires holistic thinking across enterprise boundaries along the full network

- **Smart Contracts:** provide automated actions based on triggers ensuring integrity, who have the power to change and improve business processes and capabilities.
- **Decentralized database:** Each participating partner has access to a distributed database in its entirety at all times. No single party controls the database, which every party can verify or regenerate if required without having a central intermediary. Data structures and data hashing influencing architecture modelling.
- **Cryptography:** Blockchain's transactions achieve validity, trust, and finality based on cryptographic proofs and underlying mathematical computations between various trading partners
- **Immutability**: refers to the fact that blockchain transactions cannot be deleted or altered. This characteristic of data object ownership, data revision and data governance, typically structured and defined as part of a data architecture.
- **Distributed transaction-processing platform:** Options for combining IoT with Blockchain technology to deliver machine 2 machine interactions on immutability ledgers.
- **Distributed transaction-processing platform: L**edgers handle a range of transactions, including exchanging value, assets, or other entities. Transaction processing are moving from an monolithic approach towards an integrative, digital platform

(Cross-)Business Capabilities and Processes

Architecture Governance & Principles

Target Architecture Models

Solution Architecture Development

The development of an DLT-based solution architecture impacts enterprise architecture

Focus Areas

Business Arch. → Application & Data Arch. → Technology Arch. → Development and piloting → Migration plan and roll-out → Operations

	Typical Activities*	Key Deliverables*
Business Architecture	• Development of Blockchain vision and value contribution • Business capability assessment and process impact analysis • Impact analysis for compliance and regulatory requirements with DLT-based application	• High level target picture (e.g. vision statement or canvas) • Capability assessment report • Impact analysis report
Application & Data Architecture	• Technical use case decomposition and high level solution definition • Blockchain application pattern analysis and definition • DLT-specific architecture building block specification	• Revised DLT use case • Defined blockchain application patterns
Technology Architecture	• Technology research, rating and selection • Creation of solution architecture blueprints • DLT-specific solution building block specification	• Defined technology stacks • Architectural building blocks • Architecture blueprints (0.1)

- The planned **maturity and risk assessment** in Q4/2018 will cover all focus areas and includes overarching regulatory and compliance relevant dimensions to guide the client across the architecture implementation roadmap (e.g., GDPR, ePrivacy)
- Target format is an Microsoft Excel based questionnaire to determine the current as-is and to-be state for enterprises with recommended activities

* Depends on engagement

Technical Use Case Decomposition

The identified use case must be reviewed to align to fundamental properties of DLT-based technologies

DLT-specific Key Activities

- Distributed ledger technologies (DLT) technologies span the role of an architect beyond the enterprise. An detailed use case review regarding feasibility to make use of fundamental properties* of DLT and adhere to their capabilities is required:
- Actor & role review of target transactional patterns to determine architectural selection
- Creation of a logical data flow diagram and conceptual diagrams regarding DLT-based information storage
- Analysis and definition of required data governance and data management patterns
- Setup of an trust relay concept and role for information provisioning and injection outside DLT network

Outcome

- Defined (high level) architectural solution approach for an DLT-relevant use case
- Defined architecture building blocks with DLT-specific service requirements (0.1)

Architectural Options

Fully Centralized

- Single provider service where fundamental properties are not applicable
- Recommendation is to leverage of standard cryptographic & database technologies

DLT relevant

Partially Centralized / Decentralized

- Single / multi-provider service with requirements on fine-grained operations (e.g., for asset creation)
- Single / multi-provider service with permissioned blockchain with writers but permission-less readers

Fully Decentralized

- Single / multi-provider service with permission-less blockchain – everybody can read and write

+

Verifier

- A verification role can be introduced which needs to be trusted by the network to adhere the fundamental concepts of creating trust in un-trusted (business) networks.
- The concept of an oracle is an technical implementation of an verifier

* Consist of immutability, non-repudiation, integrity, transparency and equal rights in the target blockchain network

DLT Solution Architecture Design Framework

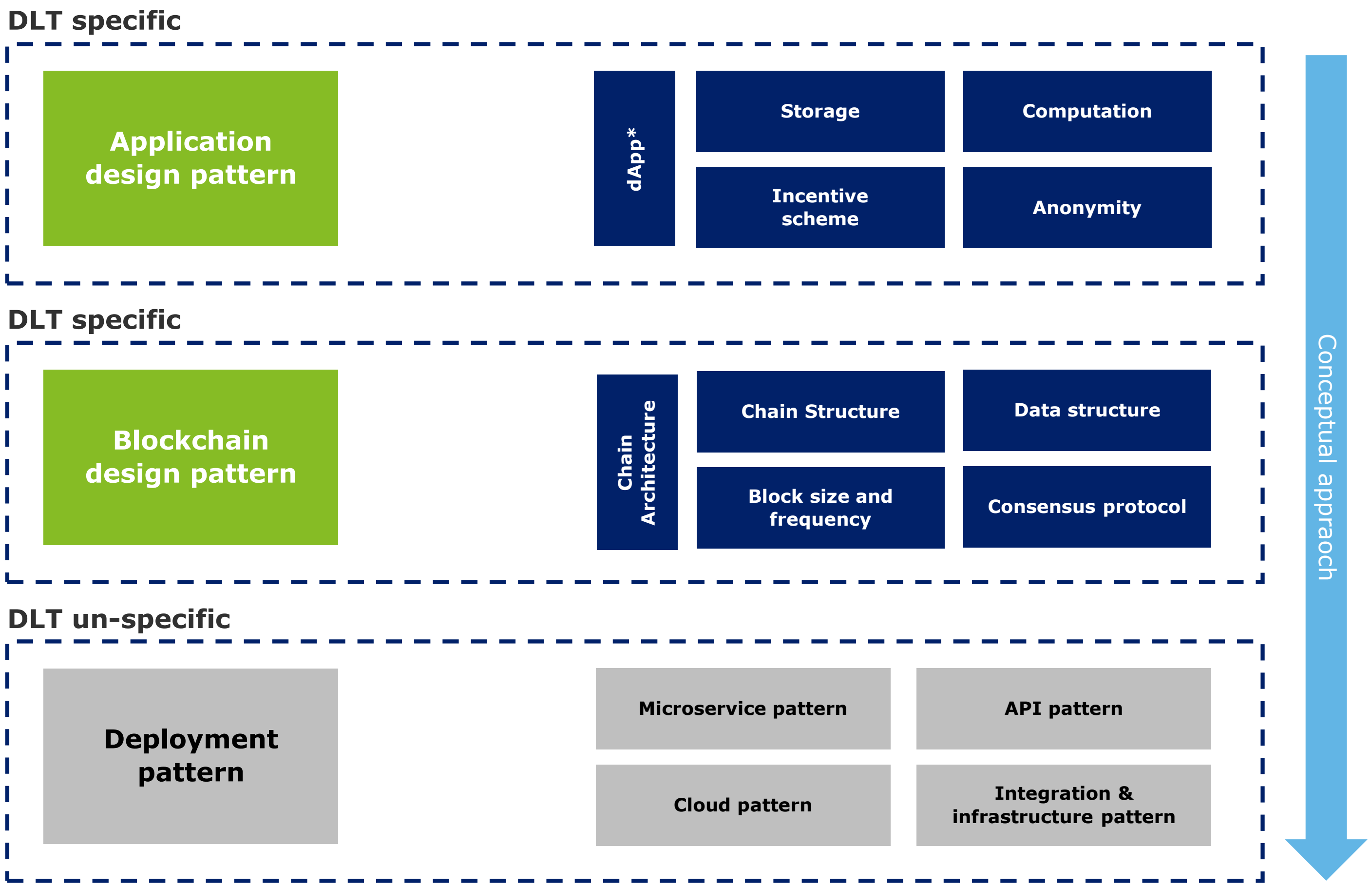
Architecting of specific DLT-based solutions can be improved with a standard taxonomy of relevant patterns

DLT based applications should be designed carefully with relevant architectural patterns based on the revised use case and to allow simplify additional iterations of a phased blockchain approach. The proposed taxonomy can be used during the process of architecting DLT-based systems.

- Application design patterns: Given the limitations of DLTs, a major decision is computation and data storage between on-chain and off-chain components as well as potential data privacy are a point of criticism of public blockchains as well as an incentive scheme, commonly used in public blockchains. Design decision for patterns depends on the value provided by the application.
- DLT design patterns: Configuration of a DLT requires design decisions for relevant patterns to configure the underlying technology according to the required use case specifications.
- Deployment pattern: General application and infrastructure design patterns influenced by DLT-based design decisions, such as microservices design patterns or API design patterns for integrations with legacy applications

Patterns are not independent ('pick and choose approach'), but knowledge of patterns is important to select the correct technology (solution) architecture and design a future-proof solution.

Not all patterns are required for all use cases.



* dAPP pattern are currently work in progress

Application Design Patterns

DLT-specific application design patterns provide a baseline for software architectural design decisions towards development of an application logic

Storage	• Method of how and what application data to store on a DLT-based architecture. • Common practice for data management in is to store raw data off-chain, and to store on-chain just meta-data and hashes. • There are various uses for wholly on-chain auxiliary data (e.g. colored coins method) on a blockchain. • Data governance required to ensure correctness, completeness and validity of data stored or hashed in the blockchain.
Computation	• Computation in a blockchain-based system can be performed on-chain (e.g., through smart contracts) or off-chain. • Different blockchains offer different levels of expressiveness for on-chain computation. • Capabilities range from simple scripting to Turing-complete languages. • On-chain benefits are the neutrality of the execution environment and immutability of the program code. • Off-chain computation offers the vast variety of languages and simplified integrative capabilities of existing environments.
Incentive Scheme	• Implemented incentive schemes have to cater the overall DLT and application design. Thus possible incentives either refer to reward network participants for distributing the ledger, appending the ledger or directly incentivize the participants needs. • Usually the incentives are distributed via crypto tokens, generated as a result of the value created by the nodes and / or users that contribute to the application. Incentives are distributed based on a verifiable proof for action such as a smart contract.
Anonymity	• Generally DLT based applications allow for anonymity by providing an API that allows creation of new addresses on the DLT. • Creation and allocation of new addresses can also be subject to an inauguration process the links the address to a users personal data thus making him identifiable. • Inauguration is also possible without linking the personal data, thus enabling the controlled joining of new participants

DLT Design Patterns

DLT-specific design patterns set the baseline for configuring the chain structure to fit the needs of the use case

Chain Architecture	• Using a public blockchain results in better information transparency and auditability, but sacrifices performance and has a different cost model. • Private or consortium-based consortium blockchain is used across multiple organizations / divisions, whereby the consensus process in a consortium blockchain is controlled by preauthorized nodes. • The right to read the blockchain may be public or may be restricted to specific participants.
Data Structure	• Chain structure is a chain of blocks, and when conflicts occur the longest chain is selected by participants. • The approach can been modified to improve scalability and reduction of conflicting transactions. • Shorter inter-block times, changing the chain structure to a directed acyclic graph (DAG) or selection rules could be configured to select the longest chain
Block Size & Frequency	• Specifies the limit of how much transactions fit in the blockchain's blocks regard to their size. The number of transactions is dependent on the amount of data that is contained in them. The size is important as it may limit the number of transactions • The block time specifies the time that passes between the creation of two blocks. Together with the block size it limits the transactional throughput of the blockchain. The block time can vary and its impact on the blockchain is strongly linked to the underlying consensus mechanism and network latency
Consensus	• The method of selecting the consensus is usually fixed for a particular selected blockchain. • Frameworks, such as Hyperledger, offer a modular architecture that caters for pluggable implementations of various • consensus protocols.
Chain structure*	• The data structure represents a back-linked list of blocks of transactions, that is ordered according to the time in which the latest block is created. Each block is identifiable a hash that results from the application of the SHA256 hash algorithm on the block header. • It can be stored as a flat file or a simple database, depending on the required functionality of the data structure

* Valid only for new blockchains

Example Evaluation of Blockchain Design Patterns

Design Pattern	Aspect	Option	Fundamental properties*	Cost efficiency	Performance	Flexibility
Chain architecture		Public chain	• • •	•	•	•
		Consortium/community chain	• •	• •	• •	• •
		Private chain	•	• • •	• • •	• • •
Data structure		Blockchain	• • •	•	•	•
		GHOST	• •	• •	• •	•
		BlockDAG	•	• • •	• • •	• • •
		Segregated witness	• • •	• •	•	•
Block size and frequency	Security	X-block confirmation	•	•	•	• • •
		Checkpointing	• • •	• • •	• • •	•
	Scalability	Original block size and frequency	• • •	n/a	•	n/a
		Increased block size / mining time	•	n/a	• • •	n/a
Consensus	Security	Proof-of-work	• • •	•	•	•
		Proof-of-retrievability	• • •	•	•	•
		Proof-of-stake	• •	• •	• •	• •
		BFT	•	• • •	• • •	•
	Scalability	Bitcoin-NG	• • •	•	•	•
		Off-chain transaction protocol	•	• • •	• •	• • •
		Mini-Blockchain	• •	• •	•	• •
Chain structure	Security	Merged mining	• • •	• •	•	•
		Chain hooking	• •	•	• •	• • •
		Proof-of-burn	•	•	• • •	• •
	Scalability	Side-chain	• • •	•	•	•
		Multiple private chains	•	• • •	• • •	• • •

* Consist of immutability, non-repudiation, integrity, transparency and equal rights in the target blockchain network

Technology Selection

Fit-for-purpose selection of DLT-based technologies requires a holistic understanding of all impacted service layers

Solution Evaluation

Business Architecture

Tech. Use Case Decomposition
- High Level Solution Approach
- Architecture Building Block(s)

DLT Design Pattern Definition
- Specified app design patterns
- Specified DLT design patterns

Non-DLT specific Pattern Definition
- Specified deployment patterns
- Specified integration patterns

Requirements Specifications
- Epic and user story definition
- Sprint planning (if applicable)

MVP and Solution Development

Various vendors offer DLT frameworks, platforms or software / hardware (IoT) stacks.
A best-fit selection for client's needs is required to ensure future-proof, functional & non-functional requirements compliance, commercial and technical viability as well IT operations demands:

Embedded / IoT
Riddle&Code

Client Layer
Web3.py
web3.js

Integration
oraclize
aelf

Blockchain Runtimes
HYPERLEDGER SAWTOOTH
HYPERLEDGER IROHA
HYPERLEDGER FABRIC
openchain
Quorum
ethereum

dAPP & Frameworks
TRUFFLE
Embark
POPULOUS

Business Logic Layer
SL

DLT Database
FlureeDB
BIGCHAINDB

Infrastructure
INFURA
IPFS

Blockchain-as-a-Service
BlockApps
SAP Leonardo
HPE Helion
ORACLE CLOUD
amazon webservices Blockchain
Chain
IBM Blockchain
Microsoft Azure Blockchain as a Service

Hosting Services
amazon webservices
Azure
Google Cloud Platform
IBM Bluemix

Not exhaustive

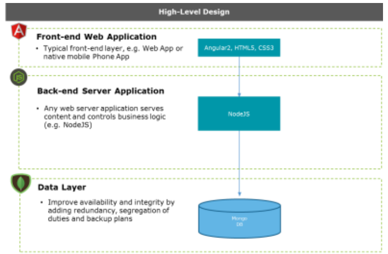
Technical architecture – Phased Approach

The depth of blockchain integration can play an increasing important role in use case definition

Phase approach benefit:
- Reduces development effort & reduced complexity per phase
- Easier incorporation of user / market feedback
- Better adoption from client’s IT organisation and IT operations teams as lower technical barriers

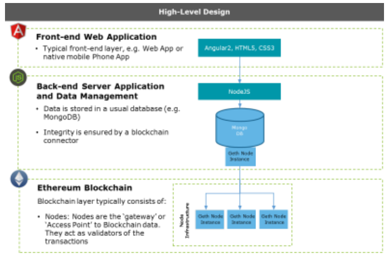
Traditional architecture

- Standard front-end web application
- Traditional server
- Central data base



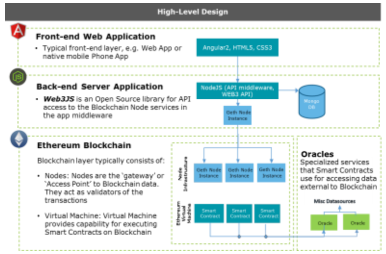
DLT as integrity service

- Traditional architecture as foundation of the service
- Inclusion of blockchain layer for additional security (eg: hash pointers written in the blockchain)



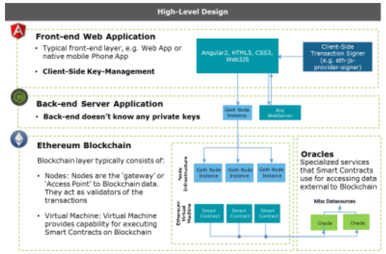
DLT infrastructure

- Traditional front end with controlled user access
- Blockchain back-end to act as transaction signer
- Ethereum blockchain as foundational layer, with node integration, virtual machine to execute contracts, and oracles



Full-blown DLT solution

- Blockchain front end with direct user access, including self-management of keys
- Integrated blockchain based back-end, with no signing possibility
- Full blockchain data layer (Ethereum)

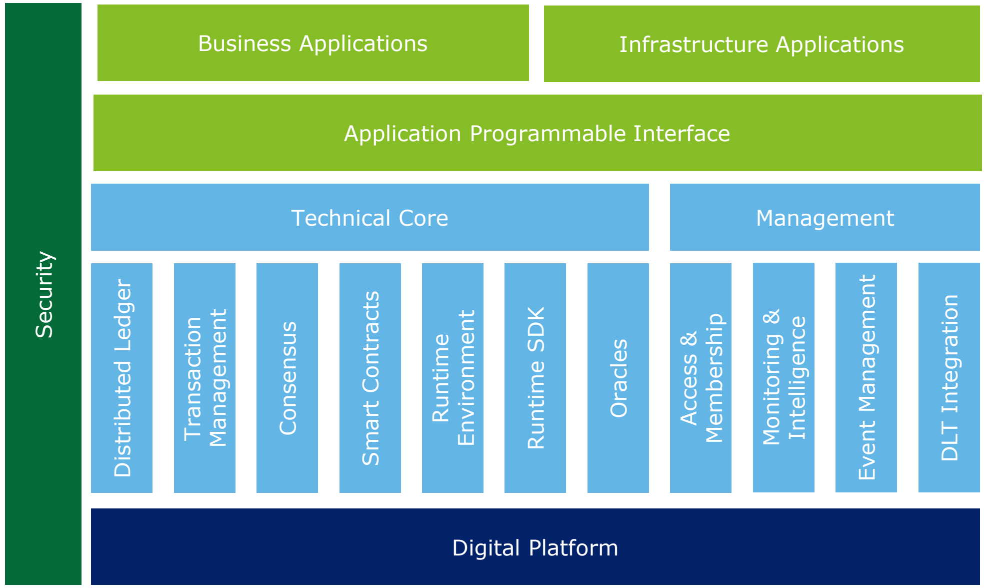


Architecture challenges:
- Multiple changes in data architecture due to phased approach can increase complexity levels
- Embedded architectural governance during all sprints and phases required to establish architectural management

Source: Blockchain Institute Germany

The Technical Reference Model

The technical reference model serves for functional and non-functional requirements gathering as well as an standardized taxonomy



- ✓ The Reference Architecture serves as basic guideline in the design, creation and operation of Blockchain technology.
- ✓ It follows the structure of the TOGAF® Reference Architecture TRM. Core of the reference architecture is the definition of services.
- ✓ The services define the required elements to consider and define when setting up Blockchain/ DLT technology. They are separated into a Technical Core and Technical Management services

The reference model provides the foundation architecture elements and functional logic for technical specifications as part of an architectural development process.

A DLT networks defines a „code is law“ environment, a completion of functional requirements and governance of their technical solution building is a fundamental part:

- Smart contracts part plays a major role in decentralized autonomous organization (environments and incomplete rule sets may allow for undesirable, but not forbidden, user behavior
- Organizational design decisions impact access restrictions and role differentiations, affecting the selection of a permitted or public blockchain

The technical reference model also provides the link to upper-layered requirements, such as legal and compliance:

- Legal and compliance challenges with liability in anonymous participation in smart contracts
- Security-by-design by the selection of the appropriate consensus mechanism (e.g. PoW, PoS, PoA, PoET)

Symbols Enabler | Architecture | Core Services | Technical Platform

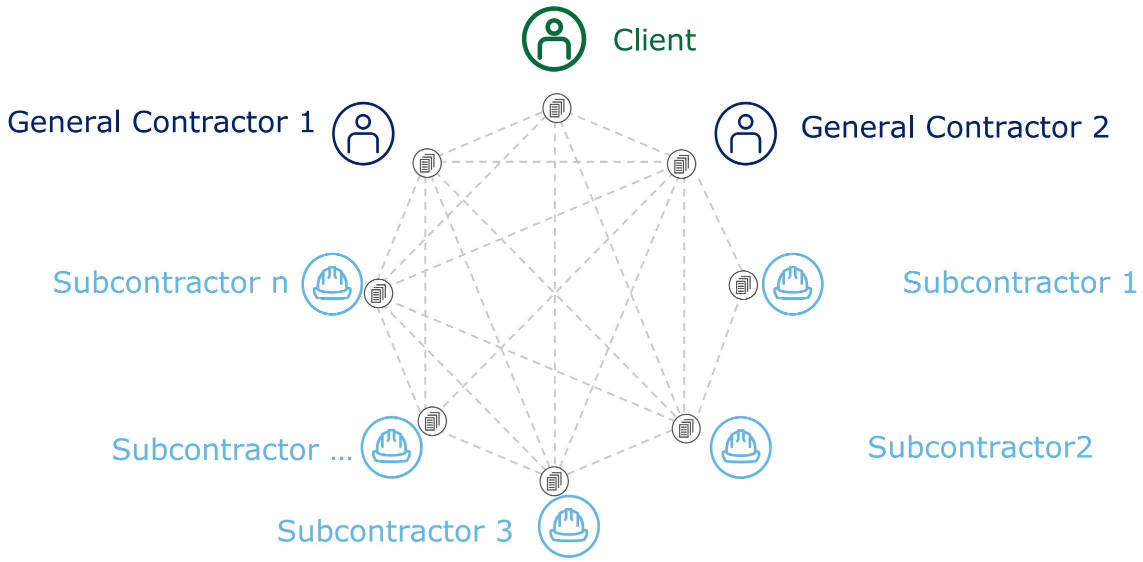
Solution Architecture Example

Contract Lifecycle Management System (LMS)

Example Use Case: Contract LMS – Vision Phase

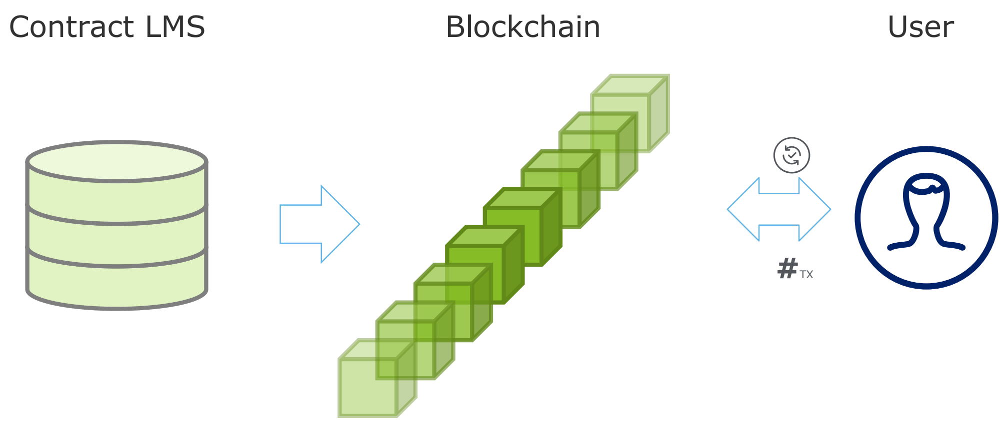
Long-running large-scale projects often give rise to legal uncertainty regarding the validity of contracts and the current status of negotiations.

Problem



- **With a lot of involved parties in negotiating contracts** of large scaling projects, the magnitude of complexity rises for **contract documentation**
- This could lead uncertainty regarding specific parts of the contracts or the contract in full. Existing revision solutions may not fulfill all legal requirements. Uncertainty leads to higher **operational costs**

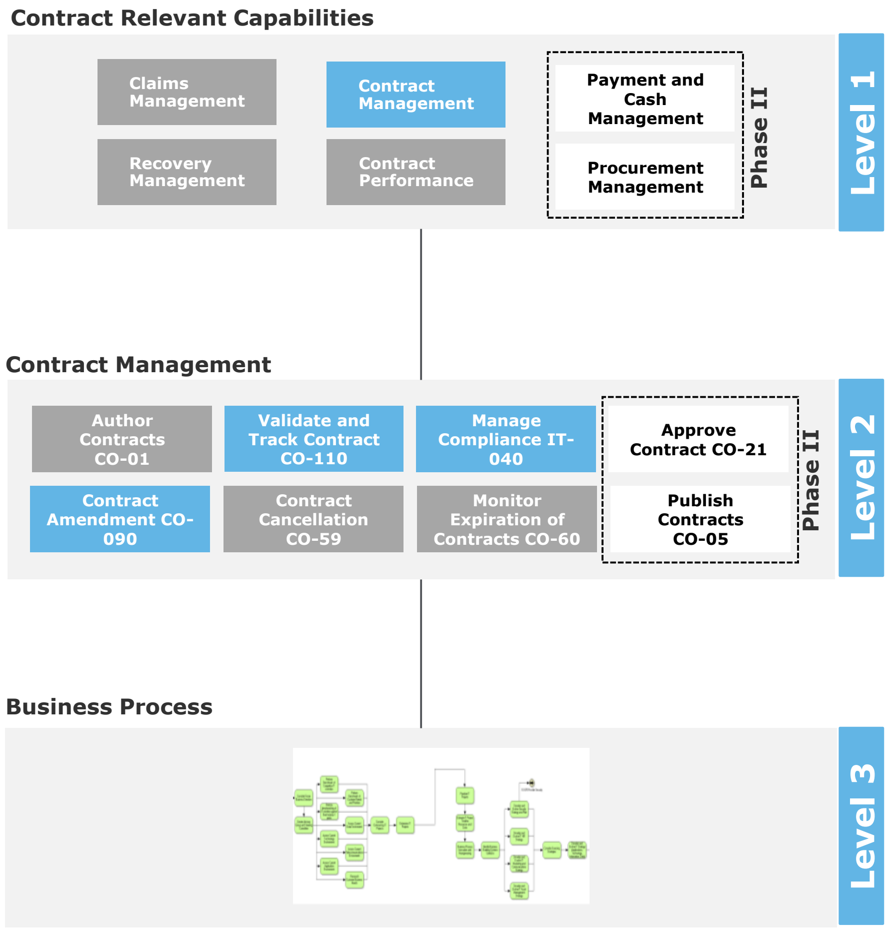
Vision



- Storing of contractual documents into a document management system using versioning with legal workflow management
- Ability to proof transparently with the **blockchain data validity** that each contract version was created and agreed upon a specific date & time
- Future **phase II with smart contracts** to provide contract automatization

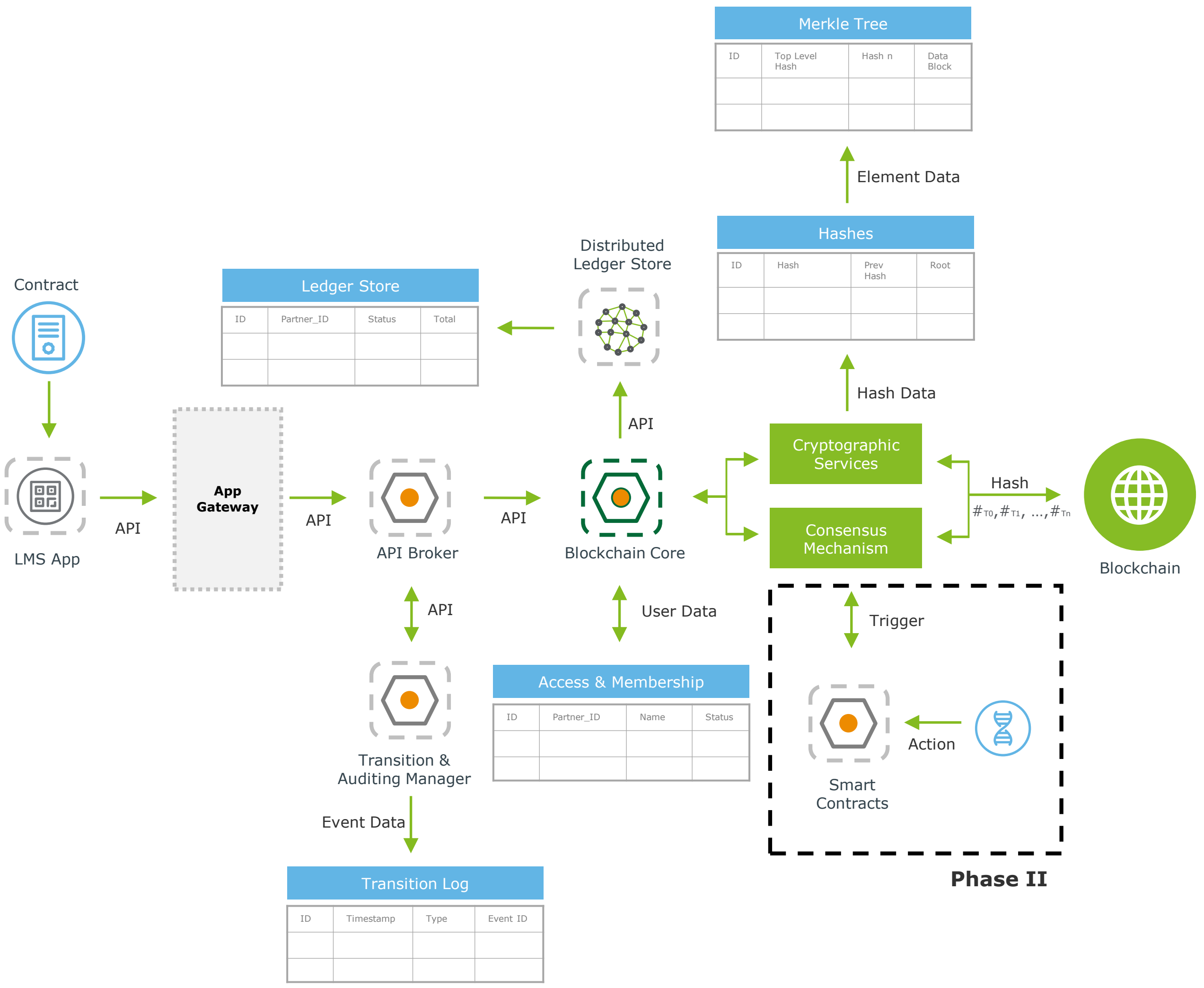
Example Use Case: Contract LMS – Business Architecture

- Effected capabilities are depending on the overall Business Architecture Capability map
- In Phase I, the capability **Contract Management** will change with switching to a LMS based platform
- The level 2 capability change include three core contract management capabilities
    - Validated and track contract capabilities reduce complexity in underlying processes
    - Manage compliance changes focus as data integrity can be assured, while still regulatory compliance must be checked
    - Contract amendments are facing simpler processes as contractual integrity is improved with Blockchain technology
- Phase II brings in smart contact functionalities which could automate contract handling
    - Changes could include automated payments or automated procurement processes effecting Level 1 and Level 2 capabilities



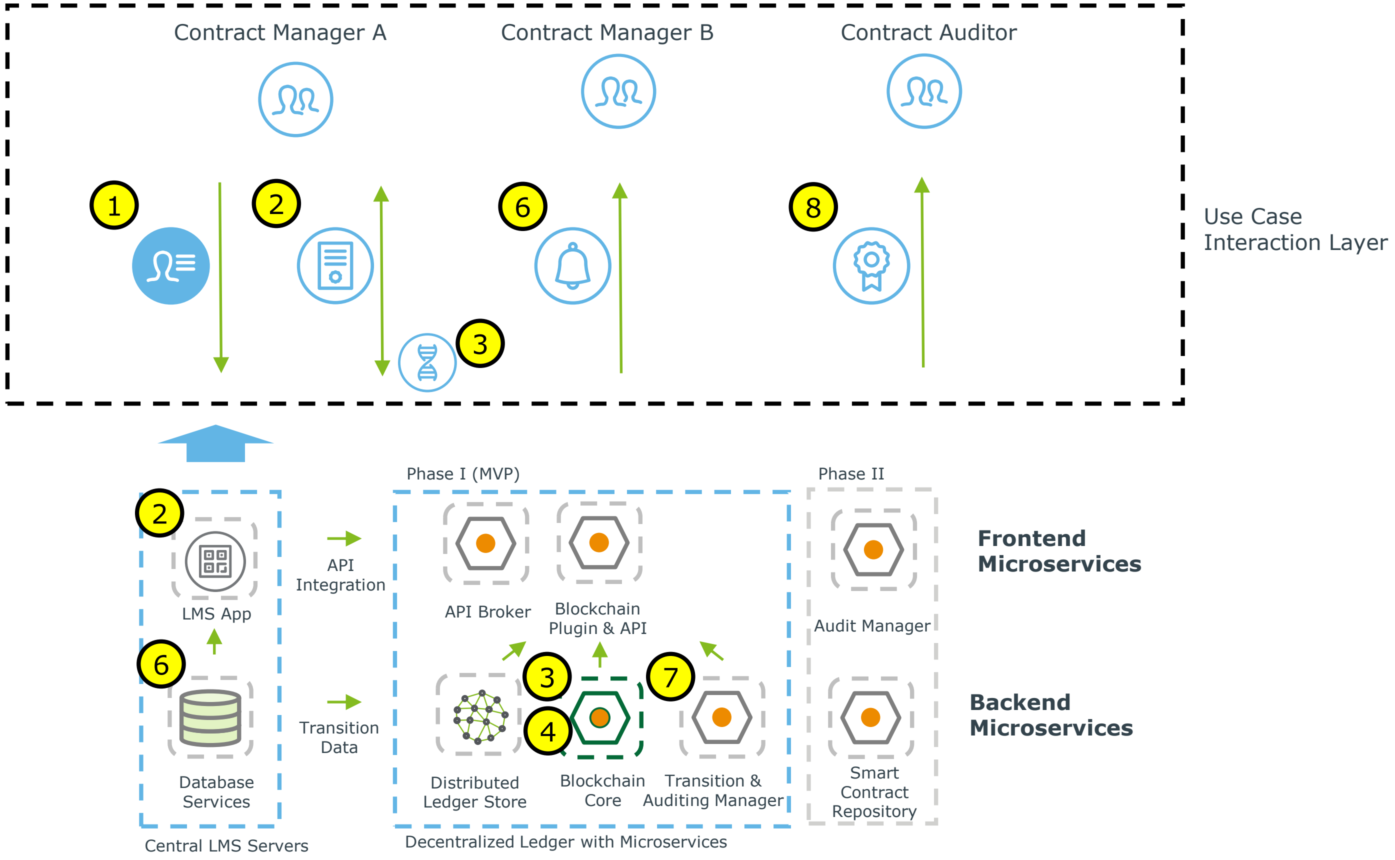
Example Use Case: Contract LMS – Data Architecture

- Data flows are using the embedded **application gateway** functionality with RESTful API interfaces between the LMS and the blockchain add-on
- Data is stored in individual data stores associated to microservices
- The cryptographic services of the blockchain core microservice create data hashes of contracts and store them in tables
- The hashes with meta data (editor, timestamp, etc.) into a blockchain to provide enhanced data integrity
- Transition & auditing service provides event data logging for event driven services (future option)
- Extensibility in phase II for smart contracts allow automatic behaviors of states triggered by events after a consensus have been applied



Example Use Case: Contract LMS – Application & Technology Architecture

- The user (Contract Manager A) gets authenticated 1 against the LifeCycle Solution Management (LMS) platform
- Contract Manager A modifies 2 an existing contract on the LMS, revision gets stored on the database
- The blockchain core creates a hash 3 of the contract version, using cryptographic services of the blockchain core
- New block on the Ethereum chain 4 is generated, initiating the consensus mechanism with Proof of Stake
- The hash gets validated 5 in the blockchain using the consensus manager and comms. protocol stack and gets stored in the distributed ledger store
- The LMS notification service sends an notification to other Contract Managers 6 to initiate legal review workflows
- Audit and transition logs hashed revision 7 state to provide an additional layer of security above usual document management revision capabilities
- Contract auditors can review 8 compliancy of contract landscape in case of legal cases



- ✓ Centralized LMS servers either on premise or cloud-based using COTS technology
- ✓ Blockchain services micro services architecture for simple scalability
- ✓ Extensibility to integrate Smart Contracts for autonomous decision making

Contact

Contacts

PHILIP PETERS

MANAGER – TECHNOLOGY STRATEGY & ARCHITECTURE, GERMANY

Philip Peters is Manager at Deloitte TS&A Service Line in Germany. His area of expertise are EAM methods, frameworks and tools and he leads the EA Setup capability within the Architecture & Integration cluster. He is managing the partnership with the EA software LeanIX. He consults his clients in complex transformations and enables the application of Enterprise Architecture management with the business.

MANUEL MICHEL

SENIOR CONSULTANT – TECHNOLOGY STRATEGY & ARCHITECTURE, GERMANY

Manuel Michel is working in Deloitte's TSA Service Line in Germany and focuses on architectural integration and infrastructure architectures in complex environments. Within this role, he consults clients in technology-enabled transformations leveraging disrupting technology stacks, organisational changes, as well as M&A integrations.

PATRICK REIBLE

SENIOR CONSULTANT – TECHNOLOGY STRATEGY & ARCHITECTURE, GERMANY

Patrick Reible is Senior Consultant in the Technology Strategy & Architecture Service Line. His know-how focuses on the development and implementation of IoT and microservice solution architectures in the automotive industry. In this context, he has build broad experience with leading edge technologies such containerized applications on the OpenShift Kubernetes platform and with regard to enabling shorter and more efficient development cycles through continuous integration and delivery processes.

Sources

Inclusive Block Chain Protocols

https://fc15.ifca.ai/preproceedings/paper_101.pdf [accessed May 10 2018].

On public and private blockchains

https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/ [accessed May 10 2018].

A Taxonomy of Blockchain-Based Systems for Architecture Design
https://www.researchgate.net/publication/314213262_A_Taxonomy_of_Blockchain-Based_Systems_for_Architecture_Design [accessed May 10 2018].

Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems

http://www.ofnumbers.com/wp-content/uploads/2015/04/Permissioned-distributed-ledgers.pdf [accessed May 10 2018].

Distributed Ledger Technology: beyond block chain

https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf [accessed May 10 2018].

IOTA Whitepaper

https://iota.readme.io/docs/whitepaper [accessed May 10 2018].

Making Sense of Ethereum's Layer 2 Scaling Solutions

https://medium.com/l4-media/making-sense-of-ethereums-layer-2-scaling-solutions-state-channels-plasma-and-truebit-22cb40dcc2f4 [accessed May 10 2018].