Co-funded by the
Erasmus+ Programme
of the European Union

A KNOWLEDGE ALLIANCE FOR BLOCKCHAIN IN ACADEMIC, ENTREPRENEURIAL AND INVESTMENT TRAINING

601063-EPP-1-2018-1-CY-EPPKA2-KA

October 2019

DLT4All Curriculum

Co-funded by the
Erasmus+ Programme
of the European Union

Table of Contents

Introduction

This document presents the curriculum for DLT4ALL. In summary, the curriculum that the DLT4All project proposes consists of the following Learning Modules:

- **LM0** – Introduction to the course and to the DLT world
- **LM1** – Peer-to-peer Database Design
- **LM2** – Encryption Techniques
- **LM3** – Consensus Mechanisms
- **LM4** – Digital Signatures
- **LM5** – Smart Contracts
- **LM6** – Privacy and Property Rights
- **LM7** – Blockchain-based Decentralized Applications
- **LM8** – Decentralized Autonomous Organizations

Learning module 0: Introduction to the course and to the DLT world

Description

Learning Module 0 is divided into two logical parts.

The first part provides an introduction to the DLT4All course, identifying different target audiences and parts of the curriculum that are addressed to each of them, and presenting an overview of the available certification options.

The second part gives an introduction to Blockchain technologies both from an historical perspective and from a technical point of view. An overview of the structure of Blockchains is presented and core concepts, which will be developed in the following modules of the course, are introduced. A taxonomy of Blockchains based on their reading and writing permissions is given, and the concepts of cryptocurrency, smart contract and decentralized application are introduced. An overview of blockchain use cases concludes the Learning Module.

Learning Objectives

- To present the structure and content of the DLT4All course
- To define the different target audiences of the DLT4All course
- To describe the certification options available to the participants
- To give an historical overview of the development of Blockchain technology
- To give an overview of the technological components of a blockchain-based system
- To present a characterization of Blockchains based on the user's reading/writing permissions
- To describe the concepts of cryptocurrency, Smart Contract and decentralized application in the context of a blockchain-based system
- To present a set of relevant use cases of blockchain-based systems

Learning Outcomes

- **Recall** the DLT4All course structure
- **Choose** which of the Target audiences they belong to
- **Understand** available certification options
- **Outline** the historical development of blockchain technologies
- **Describe** the technical components of a blockchain-based system
- **Explain** how blockchains can be classified
- **Explain** the concepts of cryptocurrency, smart contract and decentralized application
- **Name** and **describe** relevant blockchain use cases

Syllabus

1. Introduction to the DLT4ALL course
   1.1. Target audience
   1.2. Course structure
   1.3. Certification options

2. Introduction to Blockchains
   2.1. Historical overview
   2.2. Technological components of a blockchain-based system
3. Characterization of Blockchains
   3.1. Public vs Private, and Permissionless vs Permissioned
4. From Blockchains to dApps
   4.1. Cryptocurrencies
   4.2. Smart Contracts
   4.3. Decentralized applications
5. Use cases

Learning module 1: Peer-to-peer database design

Description

The possibility of having open, shared databases with certified and publicly verifiable data can lead to the involvement and empowerment of citizens in many areas.

Learning Module 1 provides an in-depth discussion of architectural design issues of data management. The challenges faced in designing a distributed and decentralized database are presented and analyzed in detail, by introducing the basic principles of database design and the mechanisms underlying peer-to-peer communication networks. Difficulties encountered in developing a reliable peer-to-peer database and potential solutions to such problems provided by blockchain technologies are presented.

Participants will understand how a database can be shared and synchronized between different nodes and why it is not easy to alter its structure afterwards. They will thus be able to determine when it is appropriate to use a distributed database. They will understand why a distributed database is reliable and how this technology can make transactions and data sharing within supply chains more secure. They will be able to recognize that in some cases it is worth creating a new ad-hoc blockchain, while in others it is better to use an existing blockchain to ensure the integrity of the data exchanged with their suppliers.

Dependencies

This module has no prerequisite.

Learning Objectives

- To explain what a database is and the basics of how it works.
- To introduce the basic concepts of peer-to-peer (P2P) communication technology and its advantages over the traditional paradigm.
- To explain the methods used to create a database in a P2P environment (*I.e.* the blockchain).
- To discuss criticalities, limits and drawbacks of using blockchain technology instead of a centralized database.
- To present use cases in which the use of blockchain technology is beneficial.

Learning Outcomes

- **Understand** the basics of database design.
- **Understand** the basic working mechanisms of peer-to-peer communication networks.
- **Understand** why blockchain technology has been developed as a solution to problems faced in designing a peer-to-peer distributed database.
- **Explain** in which cases peer-to-peer databases may or may not be useful.
- **Understand** why data integrity is not a guarantee of accuracy of information when the human factor is involved.

Syllabus

1. Explanation of what a database is and how it works
   1.1. Database principles, how a database works today
   1.2. Where databases perform better and why a new technology is needed

2. What peer-to-peer is and how it differs from the traditional communication paradigm
   2.1. Better performance on data distribution and data decentralization
   2.2. No point of failure
3. How to create a reliable database in a P2P network, the blockchain
   3.1. How block of information is linked together
   3.2. Why the chain data integrity is almost guaranteed
4. Criticality and limits of blockchain technology, use cases in which it is appropriate or not
   4.1. Why it is useful only when the information has to be exchanged between several subjects
   4.2. Best scenario: multiple actors with lack of complete trust
5. Successful practical cases and possible future implementations
   5.1. Quality and supply chain control
   5.2. More transparency in the private healthcare system

Learning module 2: Encryption Techniques

Description

Blockchain technologies rely on wide variety of cryptographic primitives and techniques to ensure their functioning. Successive blocks in an ever-growing blockchain are linked together by means of cryptographic hash functions. Accounts on a blockchain are identified by cryptographic public keys and the corresponding private keys are used to authorize transactions. Encryption techniques are a core concept for the understanding of DLTs in general, and blockchains in particular.

Learning Module 2 introduces and describes in depth cryptography concepts that are central to blockchain technology. The basics principles of cryptography and cryptanalysis are presented in an initial overview. The concept of hash function is introduced, and examples of hash functions and their applications are given. Symmetric and asymmetric cryptography techniques are discussed in detail, with examples and a dedicated lab session. The concept of Zero-Knowledge Proofs is introduced and applications of cryptography in the blockchain space are discussed.

Dependencies

This module has no prerequisite.

Learning Objectives

- To introduce the key characteristics of cryptography and its possible uses in Blockchain.
- To comparatively present different cryptosystems and their evolution.
- To describe the concept of hash function and comparatively present a set of Hash Functions used in blockchain-based systems.
- To explain the fundamental concepts of symmetric and asymmetric cryptography systems.
- To present privacy-preserving cryptographic techniques and Zero-Knowledge Proofs.
- To illustrate how cryptographic techniques are applied in blockchain-based systems.
- To present relevant use cases built around Blockchain Cryptography.

Learning Outcomes

- **Understand** the key concepts in cryptography.
- **Examine** which cryptosystem is most adequate depending on the intended use case.
- **Identify** and **analyze** the different hash functions.
- **Identify** and **analyze** applications of symmetric and asymmetric cryptography.
- **Discuss** and **analyze** privacy-preserving cryptographic methods and Zero-Knowledge Proofs.
- **Identify** and **assess** applications of cryptographic methods in blockchain-based systems.
- **Examine** how encryption techniques are being utilized in relevant blockchain use cases.

Syllabus

1. Introduction to Cryptography
   1.1. What is cryptography?
   1.2. Classification of cryptosystems
   1.3. Basic principles

1.4. Main classical cryptosystems and evolution
1.5. Perfect encryption conditions
1.6. Cryptanalysis

2. Hash Functions
   2.1. What is a Hash function?
   2.2. Types of Hash functions: MD5, SHA-x
   2.3. Lab: experimenting with hash functions
3. Symmetric cryptography
   3.1. Definition
   3.2. Vernam, Flow and Block encryption
4. Asymmetric cryptography
   4.1. Definition
   4.2. Key exchange algorithms (Diffie-Hellman)
   4.3. RSA
5. Lab: experimenting with symmetric and asymmetric encryption
6. Zero-knowledge proofs
7. Applications of Blockchain Cryptography
   7.1. Application in QR exchanges
   7.2. Managing Bitcoin and Ethereum addresses
   7.3. Practice of block theory
8. Use-cases: Finance Sector, Health, Legal Services, Defense, Public Administration, Industrial Digitalization, Social Projects, Tackling poverty, Individual identity management

Learning module 3: Consensus Mechanisms

Description

One of the central problems in the design of a blockchain system is the choice of the mechanism used by the nodes of the network to reach consensus on the state of the system in a decentralized manner, that is without resorting to having central trusted party.

Learning Module 3 explores the current mechanisms that handle the agreement among all the nodes that participate in a blockchain system. The most widely used consensus mechanisms will be discussed, with equal emphasis on the technical and business aspects of the topic.

At the beginning of the module, the need of a consensus mechanism in a distributed database setting is presented and solutions for permissioned systems are discussed. The role of Game Theory in the design of consensus mechanism is presented. A classification of the different consensus mechanisms that have been implemented in practice or proposed in theory within the two broad categories of Proof of Work (PoW) mechanisms and Proof of Stake (PoS) mechanisms is given and potential attacks on such mechanisms are described. Subsequently, a mapping of the different mechanisms to the trade-off between incentives to maintain the blockchain and security against malicious attacks that can compromise the blockchain's integrity, with special consideration of the position that each particular mechanism occupies in the centralization-decentralization space is presented. The final part of the module presents a set of practical case studies.

Participants who complete this module will be able to grasp the advantages and disadvantages of any consensus protocol in the context of any specific business model based on a distributed ledger as well as its limitations in terms of accuracy, cost efficiency, degree of decentralization, scalability, throughput rate and network sustainability.

Dependencies

This learning module has the following prerequisites:

- P2P database design (LM1)
- Encryption techniques (LM2)

Learning Objectives

- To explain the need for a consensus mechanism in a blockchain-based system.
- To present consensus mechanisms used in permissioned blockchain setups.
- To discuss the role of Game Theory in the design of a consensus mechanism.
- To present Proof-of-Concept (PoX) consensus protocols more widely used in blockchain systems
- To discuss possible attacks on distributed consensus protocols.
- To introduce and describe incentive structures used in distributed consensus protocols.
- To present attacks on consensus protocols related to incentives structures.
- To discuss costs of commonly used distributed consensus protocols.
- To discuss performance and scalability properties of widely used distributed consensus protocols.
- To present solutions to scalability issues based on alternatives to distributed consensus protocols used blockchain-based systems.
- To present use cases of different distributed consensus mechanisms in different blockchain systems.

Learning Outcomes

- **Understand** the role of consensus in DLTs and Blockchain systems.
- **Understand** the workings of PoX-type distributed consensus protocols.
- **Understand** possible attacks of distributed consensus protocols.
- **Compare** costs, performance, scalability and security between consensus protocols by analyzing their design specifications.
- **Appraise** the desired characteristics of a consensus protocol for a specific business model.

Syllabus

1. Introduction
   1.1. The need for consensus mechanisms
   1.2. Permissioned blockchains: Consensus through Byzantine-Fault Tolerant (BFT) voting mechanisms
2. The role of Game Theory in consensus
3. Proof-of-Concept (PoX) Protocols
   3.1. The concept of probabilistic consensus and its properties
   3.2. Cryptographic puzzle lotteries and their required properties
   3.3. Main protocols: Proof-of-Work, Proof-of-Stake, Delegated Proof-of-Stake, Proof-of-Authority
   3.4. A taxonomy of potential attacks: Sybil attacks, race attacks, Finney attacks, 51% attacks
4. Incentives
   4.1. Basic theory: sunk costs, principal-agent problems and incentive compatibility
   4.2. Incentive compatibility in PoX protocols: tokens, mining pools and mining cartels
   4.3. Markets in tokens
   4.4. Vulnerabilities of PoX protocols: selfish mining, block withholding, lie-in-wait mining pools, pool hopping
5. Costs
   5.1. The costly nature of PoX protocols
   5.2. Proof-of-Stake (PoS) protocols and the tragedy of the commons problem
   5.3. Security issues in PoS protocols: Nothing-at-stake attacks, grinding attacks
6. Performance
   6.1. The limited performance of permissionless blockchains
   6.2. Hybrid protocols
   6.3. Blockchain interoperability
   6.4. Non-linear blockchain networks: Greedy Heaviest-Observed Sub-Tree (GHOST) protocol
   6.5. Direct Acyclic Graph (DAG)-based protocols
7. Examples
   7.1. Hyperledger Fabric (BFT)
   7.2. Bitcoin (PoW)
   7.3. Primecoin (PoUS)
   7.4. Filecoin (UPoW)
   7.5. SpaceMint (PoSP)
   7.6. Bytecent (PoH)
   7.7. Peercoin (PoS)

7.8. Algorand (Hybrid protocols)

7.9. Teechain on Bitcoin (Side-chain networks)

7.10. Ethereum Casper implementation (GHOST)

Co-funded by the
Erasmus+ Programme
of the European Union

Learning module 4: Digital Signatures

Description

Digital signatures are the extension of paper signatures to the digital realm that is made possible thanks to the development of asymmetric cryptography. Like real signatures, they are a way to prove one's identity and to certify the origin of a message.

Learning Module 4 describes the properties and technical requirements necessary for the implementation of digital signatures and the mathematical prerequisites. The generic characteristics of a digital signature algorithms are then presented and specific algorithms that are commonly used in blockchain systems implementation are discussed in detail. Privacy preserving digital signature algorithms are discussed, focusing on their use in the cryptocurrency space and concepts of anonymity and pseudonymity of transaction on a blockchain are presented. Security of digital signatures and possible attack schemes are considered and an in-depth analysis of one possible attack vector is analyzed in a lab. The last part of the module discusses the future of digital signatures, presenting novel algorithms designed to be resistant to present and foreseen threats linked to the advent of quantum computers.

Dependencies

This learning module has the following prerequisites:

- Encryption techniques (LM2)

Learning Objectives

- To explain basic properties that a digital signature algorithm must satisfy.
- To present the digital signature algorithms that are most widely used in blockchain-based systems
- To present privacy-preserving digital signature algorithms.
- To discuss anonymity and pseudonymity features of blockchain systems.
- To present attacks and security issues related to digital signature algorithms.
- To present beyond state-of-the-art development related to digital signature algorithms.
- To present use cases of digital signatures in blockchain-based systems.

Learning Outcomes

- **Understand** fundamental properties of digital signature algorithms.
- **Understand** details of the most used digital signature algorithms
- **Acquire** technical skills related to the use of digital signatures.
- **Understand** the workings of privacy preserving digital signature algorithms.
- **Recite** possible attacks on commonly used digital signature algorithms.

Syllabus

1. Signature definitions, properties, and requests
2. Preliminaries
   2.1. Elliptic Curves
   2.2. Lab: the bitcoin-core/secp256k1 library

3. Digital signature algorithms
   - 3.1. Introduction
   - 3.2. DSA
   - 3.3. ECDSA
   - 3.4. Schnorr signatures
   - 3.5. Lab: implementation and analysis of Schnorr signatures
4. Ring signatures
   - 4.1. Properties
   - 4.2. The role of ring signatures in the Monero blockchain
   - 4.3. Lab: implementing ring signatures
5. Anonymity and pseudonymity in blockchain transactions
6. Security and Attacks
   - 6.1. Overview of security problems and attacks
   - 6.2. Lab: specific attack on Schnorr signatures
7. The Future of Digital Signatures: Lattices, hash signatures, and threshold signatures
8. Use cases

Learning module 5: Smart Contracts

Description

The term “Smart Contracts” is used with two separate interpretations:

- “A Smart Contract is a collection of code (its functions) and data (its state) that resides at a specific address on the blockchain.”
- “A Smart Contract (as a Contract) is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract.”

Learning Module 5 investigates design issues of blockchain-based applications (Smart Contracts) that can automatically execute the terms of a contract, focusing on both technological and business aspects of the subject.

The module begins with an introduction to the concept of Smart Contract and a description of its historical development. An overview of programming languages used for development of Smart Contracts for blockchain systems and execution environments is presented, followed by a detailed description of the Ethereum Virtual Machine and a thematic lab on development of Smart Contract for the Ethereum platform using the Solidity programming language. Oracles, a special category of Smart Contracts that communicate with trusted entities, are described in detail. Computational costs and security issues of Smart Contract programming are discussed, and best practices adopted to mitigate problems and risks presented. Finally, the Smart Contracts could be used as a contract, the main regulatory and legal issues about the Smart Contracts should are identified and described.

Dependencies

This learning module has the following prerequisites:

- Consensus mechanisms (LM3)
- Digital Signatures (LM4)

Learning Objectives

- To define and explain basic properties of a Smart Contract.
- To present different programming languages used to develop Smart Contracts and their execution environments.
- To describe the Ethereum blockchain and the Ethereum Virtual Machine.
- To explain how to use the Solidity language to develop a Smart Contract for the Ethereum Blockchain.
- To present the concept of Oracles their basic functioning principles.
- To discuss the computational costs of deploying and executing a Smart Contract and the role of Gas on the Ethereum Blockchain.
- To discuss security issues related to the development and use of Smart Contracts.
- To present regulation frameworks affecting the use of Smart Contracts.
- To present legal issues related to the use of Smart Contracts in a blockchain-based system.

Learning Outcomes

- **Describe** the basic concepts of Smart Contracts.

- **Recognize** different Smart Contracts' programming languages and their execution environments.
- **Identify** the key features of different Smart Contracts' programming languages.
- **Describe** the Ethereum Blockchain.
- **Explain** how the Ethereum Virtual Machine works.
- **Implement** Smart Contracts in Ethereum using Solidity.
- **Examine** the computational cost of deploying and using a Smart Contract in Ethereum.
- **Demonstrate** Smart Contracts' use cases in multiple domains.
- **Understand** current technology's restrictions and limitations.
- **Identify** issues related to the development of Smart Contracts.
- **Examine** advantages and disadvantages of using a Smart Contract.
- **Identify** concerns about security, stability and cost of a Smart Contract.
- **Assess** whether a Smart Contract solution is suitable for the problem under study.
- **Assess** how and when to apply Smart Contracts in real-life applications.
- **Implement** Smart Contracts in real-life applications.

Syllabus

1. Smart Contracts
   1.1. History, Definition, Simple use cases
   1.2. Introduction to "Smart Contracts" in Blockchain
   1.3. The lifecycle of a Smart Contract (theory)
2. Smart Contracts' Languages
   2.1. Reference to the different Blockchain environments and account of their key features: Ethereum, Hyperledger Fabric.
   2.2. Evolution of Blockchain Scripting Languages
   2.3. Overview of the key features of different high-level programming languages for various Blockchain environments
   2.4. Introduction to Smart Contract's Execution Environments
   2.5. Analysis of Ethereum Virtual Machine - EVM: Main Characteristics, Programming languages, Restrictions
3. Lab: Development in Ethereum with Solidity language
   3.1. Basic Data Types & Statements, Specific Data Types, Data Structures, Access Modifiers & Applications.
   3.2. Design Techniques
   3.3. A Smart Contract's lifecycle in practice: compilation, deployment, interaction and destruction
4. Oracles
   4.1. Introduction to oracles
   4.2. Develop contracts that will communicate with trusted entities in the real world
5. Computational Cost
   5.1. The role of gas in Ethereum
   5.2. Analysis of the cost of transactions
   5.3. Analysis of the cost of a Smart Contract
6. Security
   6.1. Issues: known 'hacks' and problems.

6.2. Solutions: security libraries, open known standards, best practices, analysis tools.
6.3. Key problems and Solutions strategies

7. Regulation Frameworks
   7.1. Overview of existing regulations for Smart Contracts
   7.2. Analysis of the laws that concern Smart Contracts

8. Legal Issues
   8.1. Clarify the Legal issues regarding Smart Contracts
   8.2. Political and environmental aspects

Learning module 6: Privacy and Property Rights

Description

Computing is no longer operating in a vacuum and as such it affects and is affected by society. Moreover, in most cases technology is ahead of society settings and legislation, and, in most cases both society and legislation need to catch up and adapt to the new situation as defined by the technology. In rare occasions, however, the technology is coming to the aid of legislation and society, and, allows certain operations to take place in a more transparent and faster fashion than current practices allow.

Learning Module 6 discusses how blockchain technology can support legislation on Property Rights, both tangible and intangible and describes privacy related issues in blockchain. The European legislation on property rights and licensing is presented and how blockchain technologies could help designing a fair remuneration scheme is discussed. The GDPR and issues related to privacy are then discussed, together with implication related to blockchain immutability for rights that need to be guaranteed under GDPR (like the right to be forgotten). Finally, technologies to enhance privacy of blockchain-based systems are presented and related issues discussed in detail.

Dependencies

This learning module has the following prerequisites:

- Introduction to the DLT world (LM0)
- Smart Contracts (LM5)

Learning Objectives

- To present the current European legislation framework related to Property Rights.
- To describe existing types of software licenses.
- To explain international copyright laws.
- To discuss license coordination and the role of registries.
- To discuss issues related to Property Rights that arise when using DLTs.
- To present the European GDPR and its implications for privacy and information freedom.
- To discuss issues related to GDPR that arise when using DLTs.
- To present a set of privacy-preserving encryption techniques.
- To discuss issues related to privacy that arise in a DLT-based system.

Learning Outcomes

- **Understand** the GDPR and its implications for blockchain technologies.
- **Understand** how to avoid conflicts with the law when utilizing blockchain technology.
- **Describe** property rights can be protected utilizing blockchain.
- **Understand** how personal data can be in the control of individuals thanks to blockchain technology.
- **Describe** how Angel Investors can valuate Blockchain-related companies.
- **Recite** the role of IPR rights in encouraging innovation and creativity and how blockchain can accommodate/facilitate IPR.
- **Understand** how the property rights can be protected utilizing blockchain technology.
- **Examine** security and privacy considerations of storage integration.

Co-funded by the
Erasmus+ Programme
of the European Union

1. Property Rights
   1.1. The legislation around Intellectual Property Rights (CPDA 1988)
   1.2. What a software license is
   1.3. License types
   1.4. International copyright law
   1.5. Token as a license?
   1.6. Private ordering
   1.7. Fragmentation
   1.8. Licensing coordination
   1.9. Registries
   1.10. Formalities
   1.11. Orphan works and the public domain
   1.12. Rights management information
   1.13. Fair remuneration
2. DLT Issues pertaining to Property Rights
3. GDPR – Issues with privacy and Information Freedom
4. DLT Issues pertaining to GDPR
5. Applied encryption techniques for privacy
   5.1. Chameleon hash functions
   5.2. Stealth addresses
   5.3. Confidential transactions through ring signatures
   5.4. Implementing privacy through zero-knowledge proof
   5.5. Private smart contracts – Enigma
   5.6. Off-chain storage
6. Privacy issues of DLTs

Learning module 7: Blockchain-based Decentralized Applications

Description

Decentralized applications differ from centralized alternatives, as they enhance a peer-to-peer network of participants. The need for transacting parties to communicate without the essentiality of a central authority is a common topic of discussion and evaluation among technology enthusiasts. Such Decentralized Applications (dApps) can disrupt various industries, which were presented to the audience as use-cases *e.g.* examples in finance, academia, supply chain, energy sector and others.

Learning Module 7 analyses dApps in all their aspects and is addressed to both technical and non-technical audiences, always keeping the topic challenging. The main aim of the module is to enable participants to evaluate which industries are ready to adopt Blockchain technology and to which extent. Decentralization and disintermediation are novel concepts and difficult to fully grasp. There are plenty of components that need to be taken into consideration such as the degree of security, privacy and interoperability. Each DLT network varies to the extent of satisfying these components. It is of ultimate importance for the content creators to evaluate the best practices and potential shortcomings of this technology. The basic structure and main design patterns of dApps are presented as an introduction to the basics of dApps development. Use cases for advanced dApps in various sectors a presented together with their relation to other disruptive technologies within the framework of the 4th Industrial revolution.

Dependencies

This learning module has the following prerequisites:

- Introduction to the DLT world (LM0)
- Smart Contracts (LM5)

Learning Objectives

- To comparatively present the conditions under which traditional centralized models and dApps can be used.
- To associate the key characteristics of dApps with the fundamental properties of blockchains.
- To explain the meaning of functional and non-functional requirements within the context of dApps.
- To comparatively present different blockchains as candidates for dApps development.
- To illustrate how the information flows at the architectural level of dApps.
- To present the technological stack of dApps.
- To present a number of indicative use cases built around dApps.
- To explain the possible synergies of dApps with other emerging technologies.
- To discuss the possible legal implications of dApps.

Learning Outcomes

- **Assess** whether a dApp is required as opposed to the traditional centralized model.
- **Analyze** the key characteristics of dApps with the fundamental properties of blockchains.
- **Identify** and **analyze** the functional and non-functional requirements of dApps.
- **Assess** the suitability of different blockchains for dApps.
- **Design** information flow architectures for dApps.

- **Identify** and analyze the main technological layers of dApps.
- **Examine** how dApps are being utilized in specific use cases.
- **Relate** dApps with other emerging technologies.
- **Identify** any requirements that may raise legal issues.

Syllabus

1. High-level anatomy of dApp
   1.1. Overview of the blockchain application stack
   1.2. Backend examples
   1.3. Frontend examples
2. dApp design patterns
   2.1. Patterns on Interacting with the External World
   2.2. Data Management Patterns
   2.3. Security Patterns
   2.4. Contract Structural Patterns
3. Basic dApps development
   3.1. Programming of public blockchains
   3.2. Programming of private/permissioned blockchains
   3.3. dApps lifecycle
4. Use cases of advanced dApps
   4.1. Decentralized exchange markets
   4.2. Decentralized data markets
   4.3. Blockchain-verifiable certificates and self-sovereign identities
   4.4. Emerging topics in the broader framework of dApps
5. Moving from dApps to the 4th industrial revolution
   5.1. The relation between IoT, AI and blockchain technologies

Learning module 8: Decentralized Autonomous Organizations

Description

The Decentralized Autonomous Organizations (DAOs) are organizations that run autonomously and could make decentralized decisions through the use of technology, e.g. Blockchain Technology, Directed Acyclic Graphs (DAG) technology, the Hashgraph algorithm, etc. A Decentralized Autonomous Organization (DAO) is typically an organization that is run through protocols encoded as various types of computer programs called smart contracts.

DAOs are sometimes also referred to as Decentralized Autonomous Corporations (DAC). Their financial transactions and program protocol records are maintained on blockchain or similar technologies. These types of organizations are similar to any organization in real world, however in digital world the rules of an organization (e.g. a company) are not enforced digitally. They are digital and already there by nature. DAOs are like a cryptographic democracy for an organization, where every stakeholder is able to vote to add new protocols, change existing protocols, or include and exclude a member among other such types of rights.

Learning Module 8 discusses the main characteristics of DAOs, as well as, their pros and cons. Particular attention will be given to the legal, cultural and political implications of the use of this disruptive paradigm. One case study of a DAO is analyzed in detail. Finally, a lab specifically designed for technical audience, will explain how to implement DAOs in the Ethereum infrastructure.

Dependencies

The learning module has the following prerequisites:

- Introduction to the DLT world (LM0)
- Smart Contracts (LM5)
- Privacy and Property Rights (LM6)

Learning Objectives

- To introduce the concept of Decentralized Autonomous Organization (DAO) as an extension of a dApp.
- To present the structure and governance mechanisms within a DAO.
- To discuss advantages and disadvantages of using a DAO to manage an organization.
- To present security issues, legal liability issues and risks of DAOs.
- To discuss cultural and political implications of DAOs.
- To analyze in-depth a specific use case of DAO.
- To present possible future developments of DAOs.
- To demonstrate how to implement a DAO on the Ethereum blockchain using Solidity.

Learning Outcomes

- **Understand** the basic concept of DAO.
- **Understand** the advantages and disadvantages of using DAOs.
- **Recite** legal and security risks of DAOs.
- **Describe** most important case studies using DAOs.

- **Implement** a DAO in Solidity.

Syllabus

1. Introduction to Decentralized Autonomous Organizations (DAOs)
   1.1. Defining the DAOs
   1.2. From dApps to DAOs
   1.3. Structure of DAOs
   1.4. Democracy within DAOs
2. Advantages and Disadvantages
   2.1. Advantages of DAOs
   2.2. Disadvantages of DAOs
   2.3. Challenges with DAOs
   2.4. Effectiveness of DAOs
3. Security, Legal Liability, and Risks
   3.1. Security of DAOs
   3.2. Legal liability of DAOs
   3.3. Risks related to DAOs
4. Cultural and Political Implications
   4.1. Cultural differences and implications of DAOs
   4.2. Political systems and implementations of DAOs
5. A Case Study of Decentralized Autonomous Organizations (DAOs)
   5.1. Exploring a case of DAO
   5.2. Lessons learned from the case study
6. Future of DAOs
7. Lab: Implementing a DAO in Solidity