Curriculum for

Certified Professional for Software Architecture (CPSA)®
Advanced Level

Module
BLOCKCHAIN

Low-Trust Consensus in Decentralized Applications

Version 2019.1-EN; April 21, 2020

Table of Contents

The abbreviation "e. V." is part of the iSAQB's official name and stands for "eingetragener Verein" (registered association), which describes its status as a legal entity according to German law. For the purpose of simplicity, iSAQB e. V. shall hereafter be referred to as iSAQB without the use of said abbreviation.

This version of this document has been produced with comments (like this one) enabled. It is **NOT** intended for public distribution or publication, but primarily for internal iSAQB purposes.

List of Learning Goals

Introduction: General information about the iSAQB Advanced Level

What is taught in an Advanced Level module?

- The iSAQB Advanced Level offers modular training in three areas of competence with flexibly designable training paths. It takes individual inclinations and priorities into account.
- The certification is done as an assignment. The assessment and oral exam is conducted by experts appointed by the iSAQB.

What can Advanced Level (CPSA-A) graduates do?

CPSA-A graduates can:

- Independently and methodically design medium to large IT systems
- In IT systems of medium to high criticality, assume technical and content-related responsibility
- Conceptualize, design, and document actions to achieve quality requirements and support development teams in the implementation of these actions
- Control and execute architecture-relevant communication in medium to large development teams

Requirements for CPSA-A certification

- Successful training and certification as a Certified Professional for Software Architecture, Foundation Level® (CPSA-F)
- At least three years of full-time professional experience in the IT sector; collaboration on the design and development of at least two different IT systems
  - Exceptions are allowed on application (e.g., collaboration on open source projects)
- Training and further education within the scope of iSAQB Advanced Level training courses with a minimum of 70 credit points from at least three different areas of competence
  - existing certifications (for example: Sun/Oracle Java architect, Microsoft CSA) can be credited upon application
- Successful completion of the CPSA-A certification exam

Essentials

What does the module “BLOCKCHAIN” convey?

The umbrella term "blockchain" describes a set of emerging, heterogeneous technologies for designing distributed systems that – while generally assuming little to no trust between parties – are able to establish consensus about stored data and procedures. Pioneered as a system to allow transfer of cryptographically secure monetary tokens, blockchains have since evolved to application platforms for executing smart contracts written in domain-specific languages. While cryptocurrencies are usually designed to lack a central authority, many industrial use cases admit partial cooperation between parties. The central idea common to all blockchain implementations is that transactions can be stored in an append-only ledger that is being kept on multiple nodes, increasing resiliency and decreasing the potential for fraudulent post-hoc modifications.

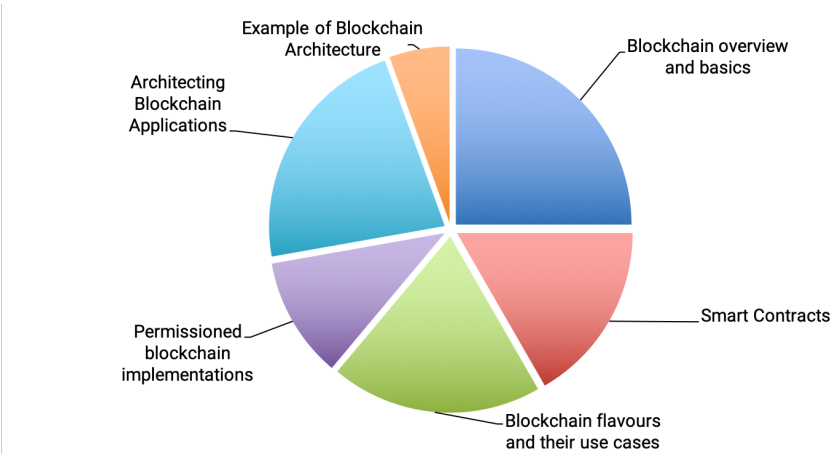
Participants of this module will learn to recognize and classify use cases of blockchain technology. They will gain a deep understanding the differences and trade-offs between their various flavours (public vs. permissioned, Proof-of-Authority/Stake/Work, smart contracts). Platform requirements and abilities, including languages for smart contracts and their testing and deployment, will be discussed.

The BLOCKCHAIN module focuses on the transfer of software engineering aspects from traditional methodologies to blockchain technologies. Participants will be able to make informed decisions about the tooling choices and be able to design decentralized applications.

Curriculum Structure and Recommended Durations

Content	Recommended minimum duration (minutes)
1. Blockchain overview and basics	270
2. Smart Contracts	180
3. Blockchain flavours and their use cases	210
4. Permissioned blockchain implementations	120
5. Architecting blockchain applications	240
6. Example of Blockchain Architecture	60
Sum	1080 (18h)

Allocation of time for the topic areas



Duration, Teaching Method and Further Details

The times stated below are recommendations. The duration of a training course on the BLOCKCHAIN module should be at least 3 days, but may be longer. Providers may differ in terms of duration, teaching method, type and structure of the exercises and the detailed course structure. In particular, the curriculum provides no specifications on the nature of the examples and exercises.

Licensed training courses for the BLOCKCHAIN module contribute the following credit points towards admission to the final Advanced Level certification exam:

Methodical Competence:	10 Points
Technical Competence:	20 Points
Communicative Competence:	0 Points

Prerequisites

Participants **should** have the following prerequisite knowledge:

- basic theoretical and practical database skills
- knowledge of any modern programming language

Additional knowledge that may be helpful, but is not required, for understanding advanced concepts:

- distributed systems and consensus algorithms
- basic JavaScript experience

We explicitly do not require knowledge in the following fields:

- cryptography, cryptocurrencies

Structure of the Curriculum

The individual sections of the curriculum are described according to the following structure:

- **Terms/principles**: Essential core terms of this topic.
- **Teaching/practice time**: Defines the minimum amount of teaching and practice time that must be spent on this topic or its practice in an accredited training course.
- **Learning goals**: Describes the content to be conveyed including its core terms and principles.

This section therefore also outlines the skills to be acquired in corresponding training courses.

Supplementary Information, Terms, Translations

To the extent necessary for understanding the curriculum, we have added definitions of technical terms to the iSAQB glossary and complemented them by references to (translated) literature.

1. Blockchain overview and basics

Duration: 180 min.	Practice time: 90 min.

This topic area introduces participants to the basic components of a blockchain. Training providers may choose to any blockchain to illustrate the concepts. Given that Bitcoin pioneered many of these components and combined them in a way that defines blockchains to this date, it is recommended to use it as a running example. While Bitcoin's focus and main application is squarely a cryptocurrency, it exhibits a structure on whose basis subsequent topics can be explained.

1.1. Terms and Principles

(cryptographic) hashing, public/private cryptography, addresses, wallets, transactions, blocks, mining, nodes, peer-to-peer networking, blockchain, cryptocurrency, (distributed) ledger, Proof-of-Work, Byzantine fault tolerance, Nakamoto consensus

1.2. Learning Goals

LG 1-1: Definition of Blockchain Terminology

- know the definition of Blockchain and DLT
- know the main concepts and terminology of Blockchain and Distributed Ledger Technologies (DLT)
- know the definition of Smart Contracts and Decentralised Applications (DApps)
- understand how Blockchain Technology enabled Cryptocurrencies and Tokens

LG 1-2: Basic operation of a blockchain

- understand the interaction of all components to form a coherent system
  - a peer-to-peer network composed of nodes that propagate transactions and blocks
  - a miner as a special type of node that forges blocks that include transactions
  - a wallet as a piece of software that manages addresses
- able to navigate the complex terminology of blockchains
  - the role of Bitcoin (or any other blockchain) as a stereotypical cryptocurrency
  - the generalized role of the blockchain as a distributed, append-only ledger

LG 1-3: Cryptographic primitives

- know the cryptographic properties of hashing and digital signatures
  - a hash function can be computed easily, but finding the inverse is computationally infeasible
  - a signature algorithm requires a private key to produce a signature, but only a public key to verify

LG 1-4: Nakamoto consensus

- understand the design constraints leading to Byzantine fault tolerance
- know the notion of "longest chain wins"

- understand the evolution of the protocol itself by majority
- understand the mechanisms of soft and hard forks

LG 1-5: Trade-offs

- able to appraise the trade-offs of a blockchain compared to a traditional architecture
- understand the hardware resources required for Proof-of-Work
- know the advantages and disadvantages of a distributed, no-trust approach

1.3. References

[Antonopoulos 2014]

2. Smart Contracts

Duration: 90 min.	Practice time: 90 min.

This topic area incrementally extends – based on the basic terminology – the notion of blockchains to include the ability for executing arbitrary, client-defined code. With Ethereum being the first and to date most mature example of this technology, it is recommended to use it as a running example.

2.1. Terms and Principles

Virtual Machine, instructions, gas, gas price, bytecode, online wallets, DAO, Truffle, security

2.2. Learning Goals

LG 2-1: Smart contracts basics

- able to create transactions
  - a wallet such as Metamask to interact with the network
- know main applications for smart contracts
  - Ethereum ERC-20 for arbitrary, transferable tokens
  - online wallets, multi-signature wallets

LG 2-2: Developing smart contracts

- implement and deploy smart contracts
- call smart contracts
  - estimating transaction fees based on gas required and the gas price
- develop, test and deploy smart contracts, for example using Truffle

LG 2-3: Virtual machines

- know the compilation pipeline from a high-level language to bytecode
- know basic instructions

LG 2-4: Security risks and implications

- know historic security flaws
  - unintended draining of the DAO contract
  - unintended destruction of Parity online wallet contracts
- able to estimate security risks for smart contracts
  - avoiding method reentrancy
  - avoiding double spending
- apply traditional methodologies to smart contracts
  - extensive unit testing with Truffle

2.3. References

[Antonopoulos+2018], [Consensys 2016]

3. Blockchain flavours and their use cases

Duration: 120 min.	Practice time: 90 min.

This topic area aims to collect and classify the abundant variations of blockchain technologies that have proliferated since Bitcoin. It is crucial for the understanding of the participants that the training provider carves out relevant features and compares them across technologies. Given the volatile nature of the field, this should be based on regularly updated market research; as such, the flavours listed below are open-ended and should be updated to reflect new developments.

3.1. Terms and Principles

Proof-of-Work, Proof-of-Stake, Proof-of-Authority, public blockchains, private blockchains, permissioned blockchains, consensus algorithms, sidechains, state channels, trust, decentralization, immutable history, business processes, onboarding, privacy, security, arbitrators

3.2. Learning Goals

LG 3-1: Feature axes

- understand block mining strategies
  - Proof-of-Work
  - Proof-of-Stake
  - Proof-of-Authority
- understand access control mechanisms
  - public, private and permissioned blockchains
  - sidechains
  - state channels
- know consensus algorithms
- know smart contract languages
- oracles
- naming services
- understand design trade-offs of blockchain flavours

LG 3-2: Choosing the best technology

- identify criteria for uses cases that benefit from blockchain technology
  - decentralization, trust model, malicious actors, competency, longevity
  - integrity and immutability of history
  - criteria for parties to join the network, access control, onboarding, arbitration
  - privacy and security of personal data
  - anonymity, pseudonymity, identity

- identify use cases based on functional and non-functional requirements
- apply traditional requirement engineering techniques to blockchain use cases
- transform business processes to a smart contract

3.3. References

[Bogensperger+2018]

4. Permissioned blockchain implementations

Duration: 90 min.	Practice time: 30 min.

For a lot of business use cases, permissioned blockchains are likely the only reasonable choice. Corporations generally have no incentive to publish internal processes to a public blockchain. But as opposed to the public blockchain space that is domineered by a few implementations with few degrees of freedom, the permissioned space usually offers a lot more flexibility to tailor a solution. This topic area should provide an overview of major concepts in this space.

4.1. Terms and Principles

Corda, Hyperledger Fabric, Ethereum, X.509, TLS

4.2. Learning Goals

LG 4-1: Permissioned blockchain implementations

- understand the benefits of permissioned blockchain regarding traditional businesses use cases
- understand the advantages of explicit governance
  - know the differences between in-chain and offchain governance
  - understand how to adjust access, verification, user rights and network structure to fit governance requirements
- know the trade-offs of reintroducing a certain level of centralisation while maintaining some of the inherent benefits of blockchain technology
- analyse quality attributes of permissioned blockchains (E.g. Corda, Hyperledger, Enterprise Ethereum)
  - know how to increase the performance by choosing different consensus algorithms
- know the technological differences in implementing major permissioned blockchains

LG 4-2: Infrastructure

- understand the infrastructure necessary for permissioned blockchains
  - X.509 and TLS are standards used for authentification and encryption
- understand the deployment patterns and their major implications on the functioning of the network
- be able to systematically comprehend and analyse the deployment patterns of major permissioned blockchains

5. Architecting blockchain applications

Duration: 150 min.	Practice time: 90min.

Smart contracts by themselves are not sufficient to build Blockchain-powered applications. They need an external infastructure powering interaction between users and contracts. Decentralization is the key pattern for designing robust applications.

5.1. Terms and Principles

Decentralized applications, sidechains, state channels, light clients, ipfs, off-chain storage

5.2. Learning Goals

LG 5-1: Architecting blockchain applications

- understand how Blockchain works as a software component of storage, computation and communication.
- know processes to design Blockchain applications
- know patterns for interaction, data management, security and contract structure.
- understand how to integrate Blockchain into a larger system.
- understand the principles of a Decentralised Application (DApp)
  - a smart contract that runs on the blockchain
  - an open source web application that interacts with the chain on behalf of the user
- undestand how to incentivize rational actors to behave a certain way using game theory

LG 5-2: Scalability

- understand the scalability challenges of public and permissioned blockchain implementations
  - understand the correlation between trust and bandwidth
  - know the performance trade-offs of PoW, PoS and PoA strategies
  - know hardware implementations of mining
- estimate the required computing power on server and client sides
  - able to design applications with light clients

LG 5-3: Storage

- understand the storage format and capacity of data on blockchains
- know the difference between on- and off-chain storage
- know the difference between replicated databases and decentralized storage
- know alternative storage methods
  - able to use distributed hash tables
  - able to understand peer-to-peer storages

LG 5-4: Privacy & Governance

- understand the durability of transactions on the blockchain
- design applications with regulatory compliance in mind
  - use hashes and salts for pseudonyms
  - know of zero-knowledge proofs

6. Examples

Duration: 60 min.	Practice time: 0 min.

This section is not examinable.

6.1. Terms and Principles

In every licensed training session, at least one example for BLOCKCHAIN must be presented.

Type and structure of the examples presented may depend on the training and participants' interests. The are not prescribed by iSAQB.

References

This section contains references that are cited in the curriculum.

A

- [Androulaki+2018] Elli Androulaki et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. EuroSys, 2018. https://doi.org/10.1145/3190508.3190538
- [Antonopoulos 2014] Andreas Antonopoulos: Mastering Bitcoin: Unlocking Digital Cryptocurrencies. O'Reilly, 2014.
- [Antonopoulos+2018] Andreas Antonopoulos, Gavin Wood: Mastering Ethereum: Building Smart Contracts and Dapps. O'Reilly, 2018.

B

- [Badr 2018] Bellaj Badr: Blockchain By Example. Packt, 2018.
- [Bartholomae+2016] Florian Bartholomae, Marcus Wiens: Spieltheorie: Ein anwendungsorientiertes Lehrbuch. Springer Gabler, 2016.
- [Bashir 2017] Imran Bashir: Mastering Blockchain. Packt, 2017.
- [Bogensperger+2018] Alexander Bogensperger, Andreas Zeiselmair, Michael Hinterstocker, Christa Dufter: Die Blockchain-Technologie: Chance zur Transformation der Energiewirtschaft? Studie der Forschungsstelle für Energiewirtschaft e.V., 2018. https://www.ffe.de/attachments/article/846/Blockchain_Teilbericht_UseCases.pdf
- [Brown 2018] Richard Gendal Brown: The Corda Platform: An Introduction. R3 Whitepaper, 2018. https://www.corda.net/content/corda-platform-whitepaper.pdf
- [Buterin+2015] Vitalik Buterin et al.: A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum Whitepaper, 2015 (with continuous updates, retrieved Apr 2020). https://github.com/ethereum/wiki/wiki/White-Paper

C

- [Consensys 2016] Consensys (various authors): Ethereum Smart Contract Best Practices. Consensys, 2016 (with continuous updates, retrieved Apr 2020). https://consensys.github.io/smart-contract-best-practices/known_attacks/

H

- [Hearn 2019] Mike Hearn, Richard Gendal Brown: Corda: A distributed ledger. R3 Whitepaper, 2019. https://www.r3.com/wp-content/uploads/2019/08/corda-technical-whitepaper-August-29-2019.pdf

K

- [Kanzow+2018] Christian Kanzow, Alexandra Schwartz: Spieltheorie: Theorie und Verfahren zur Lösung von Nash- und verallgemeinerten Nash-Gleichgewichtsproblemen. Springer, 2018.

N

- [Nakamoto 2009] Satoshi Nakamoto: Bitcoin: A Peer-to-Peer Electronic Cash System. Bitcoin Whitepaper, 2009. https://bitcoin.org/bitcoin.pdf

R

- [Rosenberger 2019] Patrick Rosenberger: Bitcoin und Blockchain. Springer, 2018.

T

- [Traub 2018] Eric Traub: Learn Blockchain Programming with JavaScript. Packt, 2018.

W

- [Wüst+2018] Karl Wüst, Arthur Gervais: Do you Need a Blockchain? Crypto Valley Conference on Blockchain Technology (CVCBT), 2018. https://doi.org/10.1109/CVCBT.2018.00011

X

- [Xiwei+2019] Xiwei Xu, Ingo Weber, Mark Staples: Architecture for Blockchain Applications. Springer, 2019.