



Blockchain Architecture Training Course: Consensus, Smart Contracts and Layer-2 Scaling

10 – 21 May 2027

London - Central London four-star



Blockchain Architecture Training Course: Consensus, Smart Contracts and Layer-2 Scaling

Ref: 1532_24840 **Date:** 10 – 21 May 2027 **Location:** London - Central London four-star **Fees:** 41400 SAR

Introduction:

Blockchain architecture with consensus, smart contracts and layer-2 scaling is a 10-day course for solution architects, senior developers and technical leads who design distributed ledger platforms, ending with a Ledger Reference Architecture and Threat Model for a case solution. Organisations stall when ledger projects reach production with untested consensus choices, exploitable contract code, unmanaged keys and no path to throughput or integration with core systems. Nominees already build or review software and integration designs, and the course runs as a lab on test networks and contract toolchains. CoreConcept Training Center teaches this blockchain architecture course at expert depth.

Course Objectives:

- Design ledger data structures, node topology and transaction models by comparing account and UTXO approaches for a stated workload
- Select a consensus protocol and network type by weighing finality, fault tolerance, throughput and validator trust
- Write, test and review smart contracts using secure coding patterns that close reentrancy, access control and arithmetic flaws
- Architect a permissioned network with membership services, channels, private data collections and endorsement policies
- Choose rollups, state channels, off-chain storage and bridges to meet throughput, privacy and interoperability targets
- Produce a Ledger Reference Architecture and Threat Model that an architecture board can review and a delivery team can build

Target Audience:

- Solution architects responsible for selecting ledger platforms and designing how they connect to enterprise systems
 - Senior developers responsible for writing, testing and deploying smart contracts and chaincode
 - Technical leads accountable for the security, performance and release quality of ledger-based applications
 - Integration and API engineers responsible for linking ledger nodes with ERP, identity and messaging platforms
 - Platform and DevOps engineers responsible for operating nodes, key stores and contract deployment pipelines
 - Security architects responsible for threat modelling and reviewing contract audits on distributed systems
-

Course Outline:

Day 1: Ledger Internals and Data Structures

- Cryptographic Hash Functions and Digital Signatures in Block Linking
- Merkle Tree Construction and Inclusion Proof Verification
- Block Header Fields, Timestamps and Chain Reorganisation Handling
- Account Model Versus UTXO Model Transaction Design
- Peer-to-Peer Gossip Propagation and Node Role Topology

Day 2: Consensus Families and Network Types

- Proof of Work Difficulty, Forks and Probabilistic Finality
- Proof of Stake Validator Selection, Slashing and Checkpoint Finality
- PBFT and Raft Ordering for Permissioned Validator Sets
- Public, Permissioned and Hybrid Network Trust Boundary Mapping
- Consensus Selection Matrix for Latency, Fault Tolerance and Throughput

Day 3: Smart Contract Design and Secure Coding

- EVM Execution Model, Storage Layout and Gas Accounting
- Checks-Effects-Interactions Pattern Against Reentrancy Attacks
- Role-Based Access Modifiers and Ownership Transfer Controls
- Proxy Upgrade Patterns and Storage Collision Prevention
- Fungible and Non-Fungible Token Interface Design at Concept Level

Day 4: Contract Toolchains, Oracles and Node Infrastructure

- Local Test Network Setup and Contract Compilation Pipeline
- Unit and Property-Based Testing of Contract State Transitions
- Oracle Design Patterns for Price and Event Data Feeds
- Off-Chain Storage With Content Addressing and Hash Anchoring
- Full, Archive and Light Node Sizing for Workloads

Day 5: Week-One Lab on a Case Ledger Application

- Case Brief for a Multi-Party Asset Registry Ledger
 - Transaction Model and Consensus Choice Applied to the Case
 - Registry Contract Build With Secure Coding Patterns
 - Oracle Feed and Off-Chain Document Anchoring Integration
 - Peer Code Review Findings Log for the Registry Contract
-

Day 6: Permissioned Platform Architecture and Interoperability

- Hyperledger Fabric Execute-Order-Validate Transaction Flow Analysis
- Membership Service Provider and X.509 Identity Hierarchy Design
- Channel Partitioning and Private Data Collection Configuration
- Endorsement Policy Design and Chaincode Lifecycle Approval
- Cross-Chain Bridges, Relays and Hash Time-Locked Swaps

Day 7: Contract Vulnerabilities, Audits and Key Custody

- OWASP Smart Contract Top 10 Vulnerability Walkthrough
- Price Oracle Manipulation and Flash Loan Attack Reconstruction
- Static Analysis, Fuzzing and Formal Verification Audit Workflow
- Hardware Security Modules, Multisignature and Threshold Key Custody
- Wallet Architecture, Key Rotation and Recovery Procedure Design

Day 8: Enterprise Integration and Consortium Network Operations

- REST and Event Gateway Design Between Ledger and ERP
- Identity Federation Linking Enterprise Directories With Ledger Certificates
- Consortium Onboarding Runbook for New Member Organisations
- Network Upgrade Proposals and Validator Voting Procedures
- Architecture Decision Records for Ledger Design Trade-Offs

Day 9: Scaling, Privacy and Performance Benchmarking

- Optimistic Rollup Fraud Proofs and Challenge Window Trade-Offs
- Zero-Knowledge Rollups, Validium and Data Availability Choices
- State Channel Design for High-Frequency Bilateral Transfers
- Zero-Knowledge Proof Privacy Techniques at Architecture Overview
- Throughput, Latency and Finality Benchmark Test Plan

Day 10: Lab Capstone on the Ledger Reference Architecture and Threat Model

- Case Enterprise Ledger Requirements and Quality Attribute Scenarios
 - Platform, Consensus and Scaling Layer Selection for the Case
 - Component Threat Enumeration Across Nodes, Contracts and Bridges
 - Control Mapping for Keys, Contracts and Integration Gateways
 - Ledger Reference Architecture and Threat Model Completion and Defence
-

Skills You Will Gain:

- Ledger Data Structure Design
- Consensus Protocol Selection
- Secure Smart Contract Coding
- Permissioned Network Configuration
- Contract Security Auditing
- Cryptographic Key Custody
- Layer-2 Scaling Design
- Ledger Performance Benchmarking

Why Attend This Course:

- Deliver a Ledger Reference Architecture and Threat Model, built in the lab, to the architecture review board and the delivery team lead
- Decide which consensus protocol, network type and scaling layer a proposed ledger solution needs before platform procurement starts
- Avoid contract exploits, lost signing keys and bridge failures that would otherwise surface only after deployment to production
- Pass on secure coding patterns, audit checklists and benchmark test plans to developers and operations colleagues on the same platform

Conclusion:

Back at work, the participant hands the Ledger Reference Architecture and Threat Model to the architecture review board, which uses it to approve the platform, consensus and scaling choices for the case solution or a live ledger initiative. Delivery and security teams can take the threat list and control mapping into sprint planning, contract audits and key ceremony design. After the first release, the unit should review benchmark results against the stated throughput and finality targets, audit findings against the threat model and whether consortium onboarding worked as the runbook planned.

Frequently Asked Questions FAQ:

What should participants know before a blockchain architecture course on consensus, smart contracts and layer-2 scaling?

Participants should already write or review code in at least one programming language, understand public-key cryptography in outline and be comfortable with command-line tools, APIs and integration design. No prior smart contract experience is assumed, but daily hands-on software or architecture work is expected.

How does blockchain architecture with consensus, smart contracts and layer-2 scaling differ from a blockchain strategy course?

It is an engineering course. A strategy course decides whether a ledger fits a business problem and how to pilot it; this course designs, codes, secures, scales and integrates the ledger itself, ending with a reference architecture and threat model that a delivery team can build.

When should a blockchain architecture use layer-2 scaling instead of a faster consensus protocol?

Layer-2 scaling suits workloads that need many low-value transactions while keeping settlement on a secure base chain. A faster consensus protocol suits permissioned networks with known validators. The choice depends on finality targets, data availability needs and how much trust participants place in operators.

What do participants take back from the blockchain architecture course on consensus, smart contracts and layer-2 scaling?

Participants return with a Ledger Reference Architecture and Threat Model for a case solution, covering transaction model, consensus and scaling choices, permissioned network design, contract security controls, key custody, integration gateways and a benchmark test plan ready for review.
