Blockchain Innovation Program Tutorial Framework

BIP BLOCKCHAIN INNOVATION PROGRAM

POWERED BY

BSV ACADEMY

Blockchain Innovation Program
Tutorial Framework

The Blockchain Innovation Program is a designed as an intensive 10-week programme which will provide students with educational and entrepreneurial support for them to develop a comprehensive understanding of what's involved in Bitcoin Application Development.
The programme will see the students complete the three bitcoin primitive courses Hash Functions, Merkle Trees, and Digital Signatures before they complete the newly refactored Introduction to Bitcoin Development.

Educators from the Bitcoin SV Academy team will prescribe weekly resource and question packs to stimulate the students to develop a deeper consideration for what is involved in creating a scalable Bitcoin application. The students will attend fortnightly tutorials where their answers to the question pack will be evaluated and discussed in greater detail.

Live session #1

Hash Functions	• Hash Functions and Hash Tables • Content Addressed Distributed Data Structures • Efficient Provable Data Possession for Hybrid Clouds

Course pre-requisite: Hash Functions (primitives)
Worksheet to complete prior to the live session: Week 1 – Hash Functions

Live session #2

Merkle Trees	• GitHub MerkleDAG • ForkBase: Immutable, Tamper-evident Storage • Substrate for Branchable Applications • Merkle-CRDTs - MerkleDAGs meet CRDTs • Merkle^2: A Low Latency Transparency Log System
Digital Signatures	• Digital Signatures • Legitimating Technologies: Digital Signatures Case Study. • Segwit, Mixing and Law • SigHash Flags

Course pre-requisite: Merkle Trees and Digital Signatures (Primitives)
Worksheet to complete prior to the live session: Week 2 Merkle Trees and Week 3 Digital Signatures.

Live session #3

Data and Databases	- What is DBaaS? - SQL vs NoSQL - What is Cloud Storage? - What is Object Storage? - Block vs File Storage - What is a Load Balancer? - Kubernetes vs Docker
API led Event-Driven & Microservices Architectures	- API vs SDK - What is API Management? - What is a REST API? - What is an API Gateway? - What is Event Driven Architecture? - What are Microservices? - Architecting a Cloud Native API Solution. - Blockchain Enabled Trustless API Marketplace - Unofficial API and Browser Extension Development for Augmenting Student Resources

Course pre-requisite: Bitcoin Development Chapter 1
Worksheet to complete prior to the live session: Week 4 Data and Databases and API led Event-Driven & Week 5 Microservices Architectures

Live session #4

Debunking the Blockchain Trilemma, CAP Theorem & Application Scalability	- Myths of Decentralisation - On Decentralisation - The Wizard of Blockchain - Cost Performance Trade-Off Evaluation in Microservices impacted by the CAP Theorem Limitations
Working Blockchain & Overlay Networks	- A Survey and Comparison of P2P Overlay Network Schemes. - Virtual Networking Explained - What is a Content Delivery Network - Mandala Network - SPV - Working Blockchain

Course pre-requisite: Bitcoin Development Chapter 2&3
Worksheet to complete prior to the live session: Week 6 Debunking the Blockchain Trilemma, CAP Theorem & Week 7 Application Scalability and Working Blockchain & Overlay Networks.

Live session #5

Intro to Git and Github	• Git and GitHub for Beginners • Getting Started With OpenSSH Key Management. • Setting up an Nx monorepo with Angular • Setting up CI/CD with Github Actions and Vercel
Constructing Transactions & Script	• Introduction to Bitcoin Transactions • MintBlue API, SDK and Integrations

Course pre-requisite: Bitcoin Development Chapter 4-5
Worksheet to complete prior to the live session: Week 8 Intro to Git and Github and Week 9 Constructing Transactions and Bitcoin Script.

Live session #6

Metanet	• Metanet Overlay • Dagda • The Birth of Ontology & the DAG • Tutorial on directed Acyclic Graphs • A.N.N.E preview.
End of programme	• Feedback on project completed by students • Wrap up of the programme

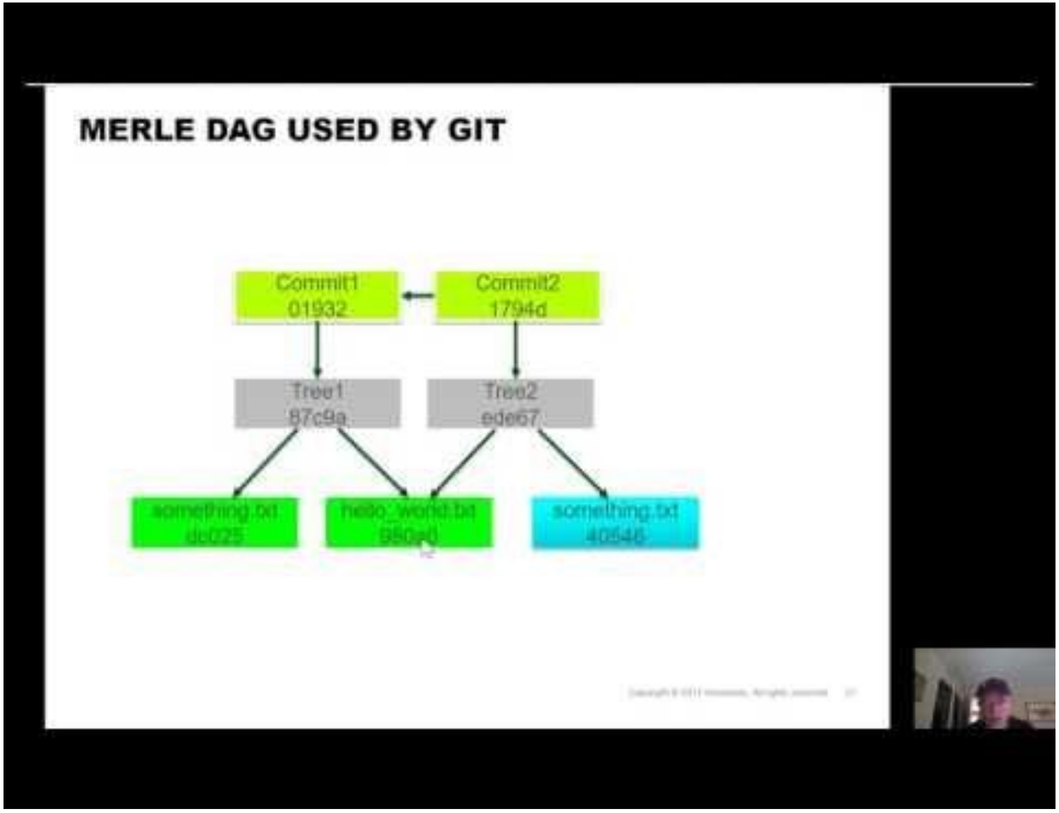
Course pre-requisite: Bitcoin Development Chapter 6
Worksheet to complete prior to the live session: Week 10 Metanet

Blockchain Innovation Program

Worksheets

Week 2 Merkle Trees

1. **Github merkle DAG**



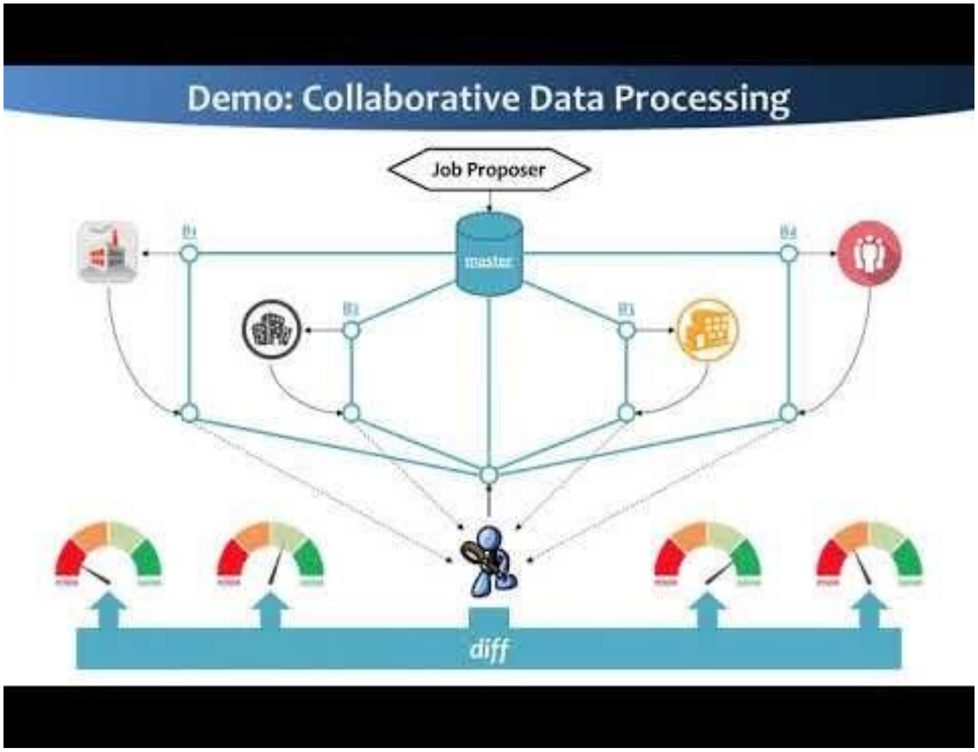
MERLE DAG USED BY GIT

a. If one version of something.txt contained the bytes 100101010101000100100000101001010101001001001010101010011010101010 and the newer version contained the bytes 100101010101000100100000101001010101001001001010101010011010100000 then what function could be used to find out how much data needs to be synchronised in the commit?

b. How would longest common substring and longest common sequence be useful in calculating the diff of?

c. Merkle trees are also used in various other distributed systems. How might a Merkle tree be used in the case of torrenting?

2. [986 ForkBase: Immutable, Tamper-evident Storage Substrate for Branchable Applications]



Demo: Collaborative Data Processing

3. [ForkBase: Immutable, Tamper-evident StorageSubstrate for Branchable Applications (Demo)]

About ForkBase

ForkBase: An Efficient Storage Engine for Blockchain and Forkable Applications
Sheng Wang, Tien Tuan Anh Dinh, Qian Lin, Zhongle Xie, Meihui Zhang, Qingchao Cai, Gang Chen, Beng Chin Ooi, Pingcheng Ruan. **Int'l Conference on Very Large Data Bases (VLDB), 2018.**

Demonstration using ForkBase web application

[Note: this video has no audio]

4. https://www.comp.nus.edu.sg/~ooibc/icde20forkbase.pdf

   a. How are Merkle trees used in ForkBase?
   b. How could a system like ForkBase be useful in managing data from a blockchain application?

5. **Merkle-CRDTs Merkle-DAGs meet CRDTs – Merkle DAGs as a transport and persistence layer for Conflict-Free Data Types (Part I, II (A,B,C,D) and Conclusion)**
   https://arxiv.org/pdf/2004.00107.pdf

   a. What assumptions must be made regarding using CRDT instead of a blockchain consensus algorithm to achieve eventual consistency?
   b. What is a logical clock?
   c. In what way does a Merkle CRDT model inherit the benefits of both Blockchain and CRDT?

6. [Merkle^2: A Low-Latency Transparency Log System]()

[redacted]

Merkle^2: A Low-Latency Transparency Log System

Yuncong Hu, Kian Hooshmand, Harika Kalidhindi, Seung Jin Yang, Raluca Ada Popa

University of California, Berkeley

[redacted]

Merkle^2 : A Low-Latency Transparency Log System (Sections I, II, IV, XI)
https://eprint.iacr.org/2021/453.pdf

   a. What is the unique technique used by Merkle^2?
   b. What are some of the shortcomings of a chronological Merkle tree in terms of efficiency?
   c. How does creating a multi-dimensional Merkle tree improve latency issues in look up?
   d. What is the difference between the chronological tree and the prefix tree?
   e. How are the trees nested within one another?
   f. What kind of blockchain applications could benefit from leveraging a multidimensional Merkle tree?