Blockchain Innovation Program Tutorial Framework

POWERED BY

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

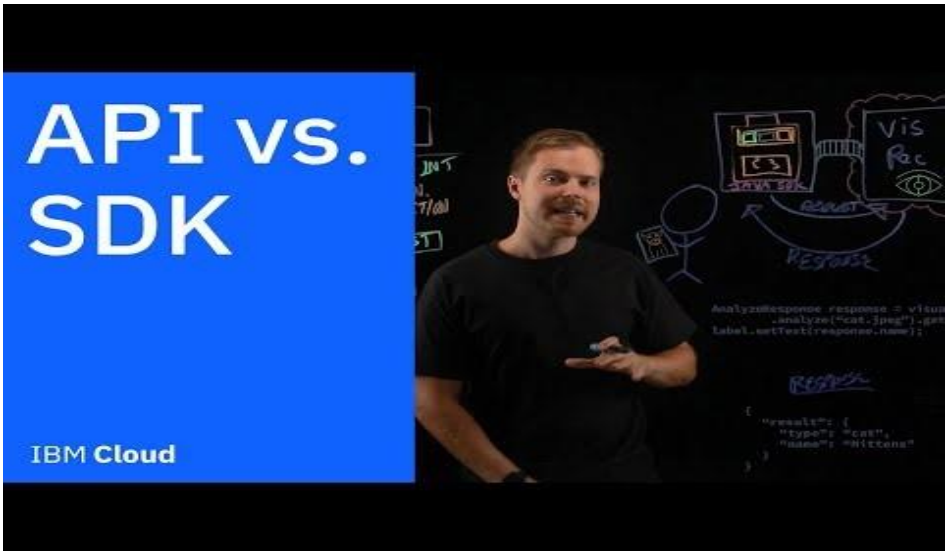
Course pre-requisite: Bitcoin Development Chapter 6
Worksheet to complete prior to the live session: Week 10 Metanet

Blockchain Innovation Program

Worksheets

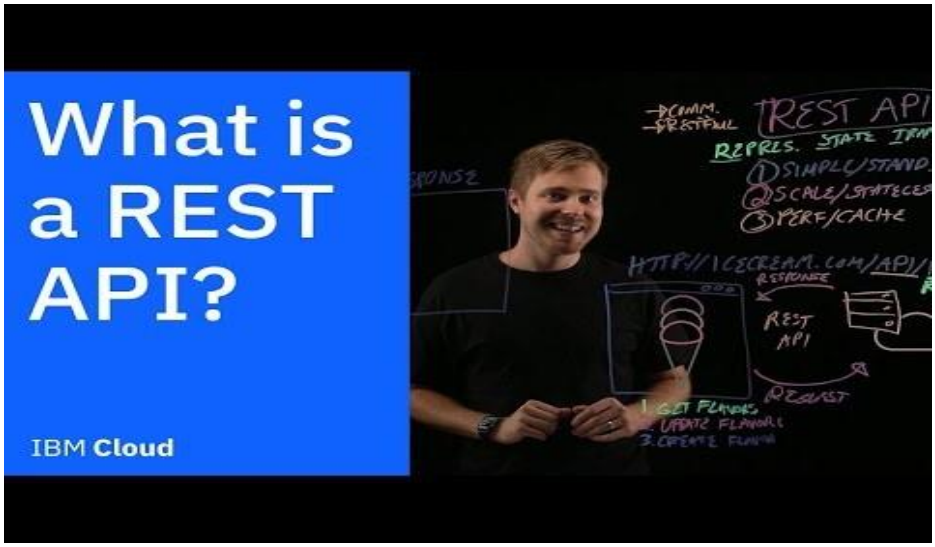
Week 5: API led Event-Driven & Microservices Architectures

1. API vs. SDK: What's the difference?



a. What is the high-level relationship between an API and an SDK?
b. How does bitcoin lend itself well to an API led architecture?

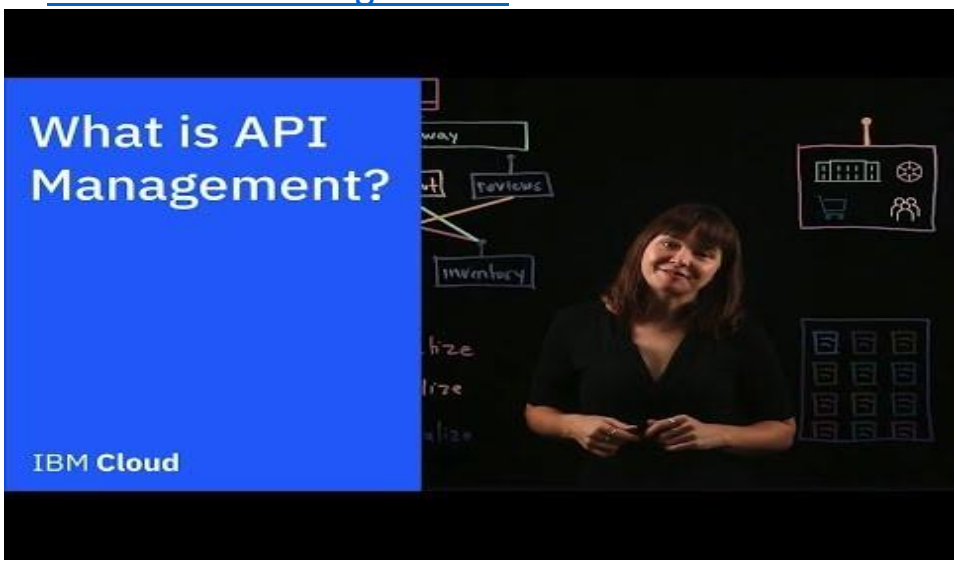
2. What is a REST API?



PI?

a. MerchantAPI (mAPI) uses a REST API which enables a merchant to utilise the SPV paradigm. What type of information might be relayed through the mAPI REST API which would enable an SPV transaction to be accepted?

3. What is API Management?



a. If an API is like a waiter, how could a bitcoin transaction work like a tip?
b. What are the four core elements of an API management system?

4. [What is an API Gateway?]

a. How could an API gateway make adding a rich feature set to your application much easier?

5. [What is Event Driven Architecture (EDA)?]

a. In the case of a bitcoin application, what would be some of the messaging events that leverage the properties of blockchain?
b. What would be the drawback of using the blockchain as your only event log?
c. In what way can a bitcoin transaction be considered an immutable statement of fact

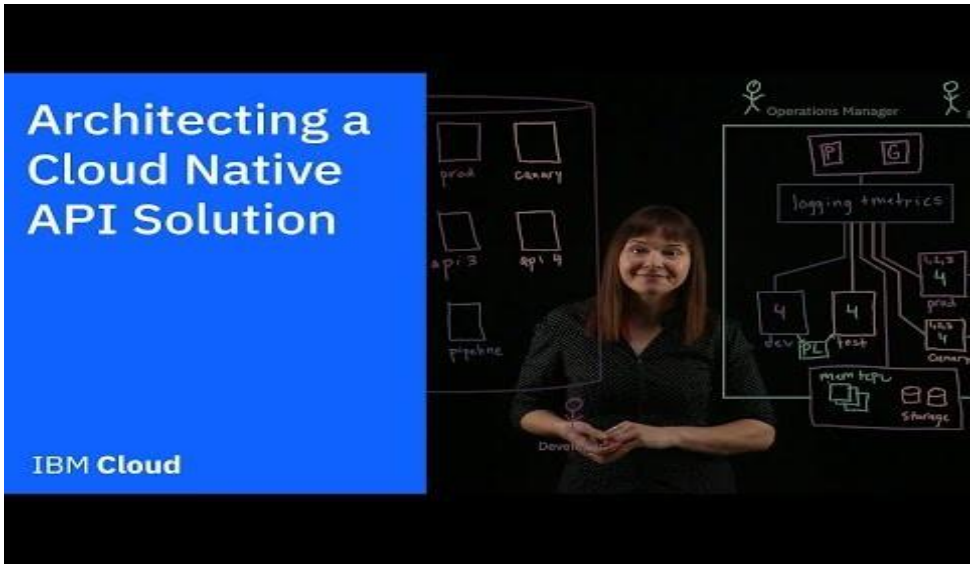
6. What are Microservices?

a. How could a pricing plan work for a microservices based bitcoin application?

b. How could a microservices architecture benefit a bitcoin application that eventually compiles transaction events down to bitcoin script?

c. If your bitcoin application had microservices for inventory, recommendations, cart and payments, which microservice would be least likely to fall over first?

7. Architecting a Cloud Native API Solution



a. What could be considered analagous to a canary environment with respect to bitcoin networks?

8. **Blockchain Enabled Trustless API Marketplace**

https://arxiv.org/pdf/1812.02154.pdf

a. How could bitcoin change the general structure of an API marketplace in terms of cloud vendors, API providers and API consumers?
b. What might represent the application layer endpoint for an API in a bitcoin application?
c. How might encapsulating the end points in this way lead to new markets around APIs?
d. How might an application developer be able to approach an API lead design without the conventional method of raising capital in exchange for equity?
e. How might enabling more p2p vending in an API marketplace minimise risk for API lead design?
f. Discuss the feasibility of the following quote with the ability to service $0.00001 transactions:
"Today I've been asked if I would build a new exchange as a webapp or a mobile app. Neither. I would build a Codugh API. Pay per call, include it everywhere." - Alex Agut

9. **Unofficial API and Browser Extension Development for Augmenting Student Resources**

https://american-cse.org/sites/csci2020proc/pdfs/CSCI2020-6SccvdzjqC7bKupZxFmCoA/762400b012/762400b012.pdf

a. Discuss the possibility of collecting low hanging fruit in the context of modifying browser extensions to incorporate bitcoin based API calls leveraging something like handcash....