Chapter 11

Integrating Non-Blockchain Apps with Ethereum

Episode 11.01

Blockchain and Database Storage

Comparing Blockchain and Database Storage

- Control
  - Databases – one authority controls data repository
  - Public blockchain – no central authority
- Data format
  - Databases – schema imposes data format
  - Blockchain – “freeform”, smart contracts help with data standards

Comparing Blockchain and Database Storage

- Updating data
  - Databases – CRUD operations
  - Blockchain – add or read only
- Optimizing performance
  - Blockchain – slower
  - Databases – writing to a highly optimized data repository

Comparing Blockchain and Database Storage

- Confidentiality
  - Blockchain – harder
  - Database – governing authority can protect data
- Paying for storage
  - Database – no direct payment for storage
  - Blockchain – pay for access
- Transparency
  - Blockchain – transparent
  - Databases – only access what owner allows

Comparing Blockchain and Database Storage

- Integrity
  - Databases – some add-on capability
  - Blockchain – technology lends itself to trust
- Resilience
  - Blockchain – built in
  - Databases – expensive and difficult

Episode 11.02

Execution and Flow in dApps and Traditional Applications

Execution and Flow in Databases and dApps

- Traditional database apps tend to be more centralized
  - Storage
  - Processing
- dApps are, by definition, decentralized
  - Invocations are asynchronous and detached

Blockchain Flow Considerations

- Response time tends to be longer for dApps
- Smart contracts should avoid making users wait
  - User input/output collected after user submits
  - Can always emit an event to let user know an action was completed

Episode 11.03

Blockchain Incorporation Design Goals

Implementing Blockchain Technology Goals

- Address application shortcomings
  - Fix some drawbacks of traditional applications
    - Ex: track ownership of digital assets
- Introduce previously unavailable features
  - Add features unavailable in existing application

Implementing Blockchain Technology Goals

- Enhance the user experience
  - Providing more details
- Reduce operational costs
  - Disintermediation
  - Increasing autonomy and automation
- Enhance auditability and compliance
  - Track all data and programming changes

Episode 11.04

Integration Considerations for Incorporating Blockchain

Integration Considerations

- Design
  - How will blockchain fit into existing structure?
- Decentralization
  - Timing issues
  - Data is more open
- Document all processes
  - Align existing and new blockchain processes

Integration Considerations

- Know what services are available
  - Look for existing services you can use
- Identity
  - Identities associated with accounts in Ethereum
  - Map identities from to an Ethereum account?
  - Does a single identity need to be mapped to multiple Ethereum accounts?

Integration Considerations

- Integration design pattern
  - Use the same look and feel as existing APIs

Episode 11.05

Interface Considerations for Incorporating Blockchain

Interface Considerations

- When do I need to call smart contract functionality?
- What do you want the interface to do?
- What data must you provide to the interface?
- What data will the interface return to you?

Interface Considerations

- Reliability
  - Work the same way every time with good performance
- Serviceability
  - Service and update throughout the life of the dApp
- Availability
  - Easy integration with existing system
  - Resilience