CSE5095-010: Blockchain Technology

Lecture 12

Ghada Almashaqbeh
UConn - Fall 2020

Outline

- A design framework for building decentralized blockchain-based service systems.
  - Centrally-managed service vs. P2P based.
  - Challenges of P2P based services.
  - A design framework for such services.

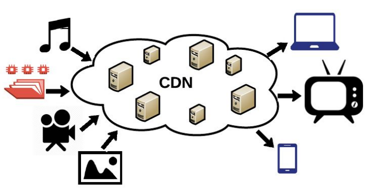
Traditional Service Systems

Central Management

Services

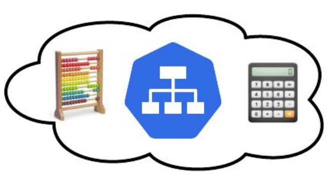
File Storage

justcloud.com OneDrive



Content Distribution

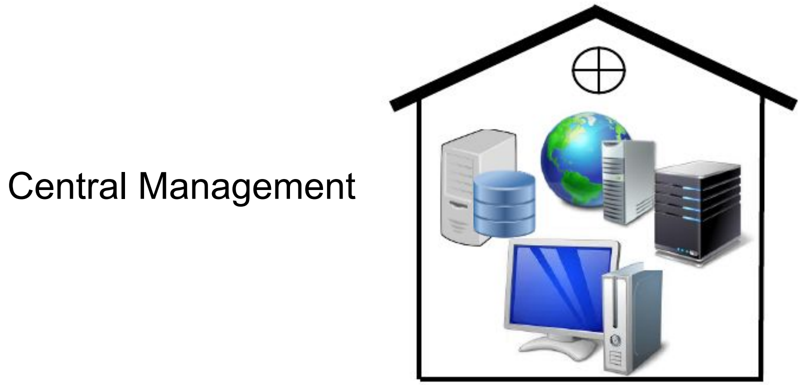
Akamai CLOUDFLARE fastly



Computing

Google Cloud Platform Microsoft Azure

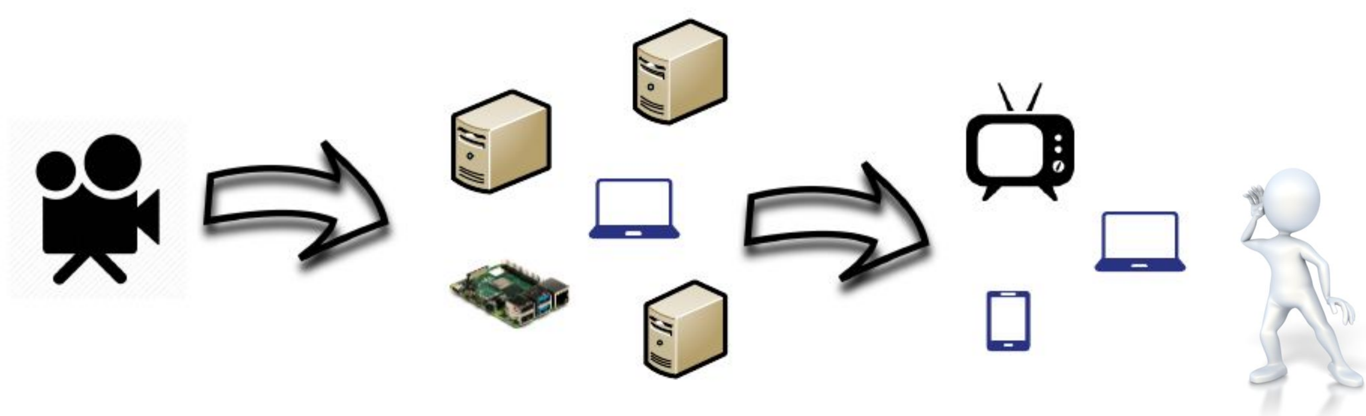
Traditional Service Systems



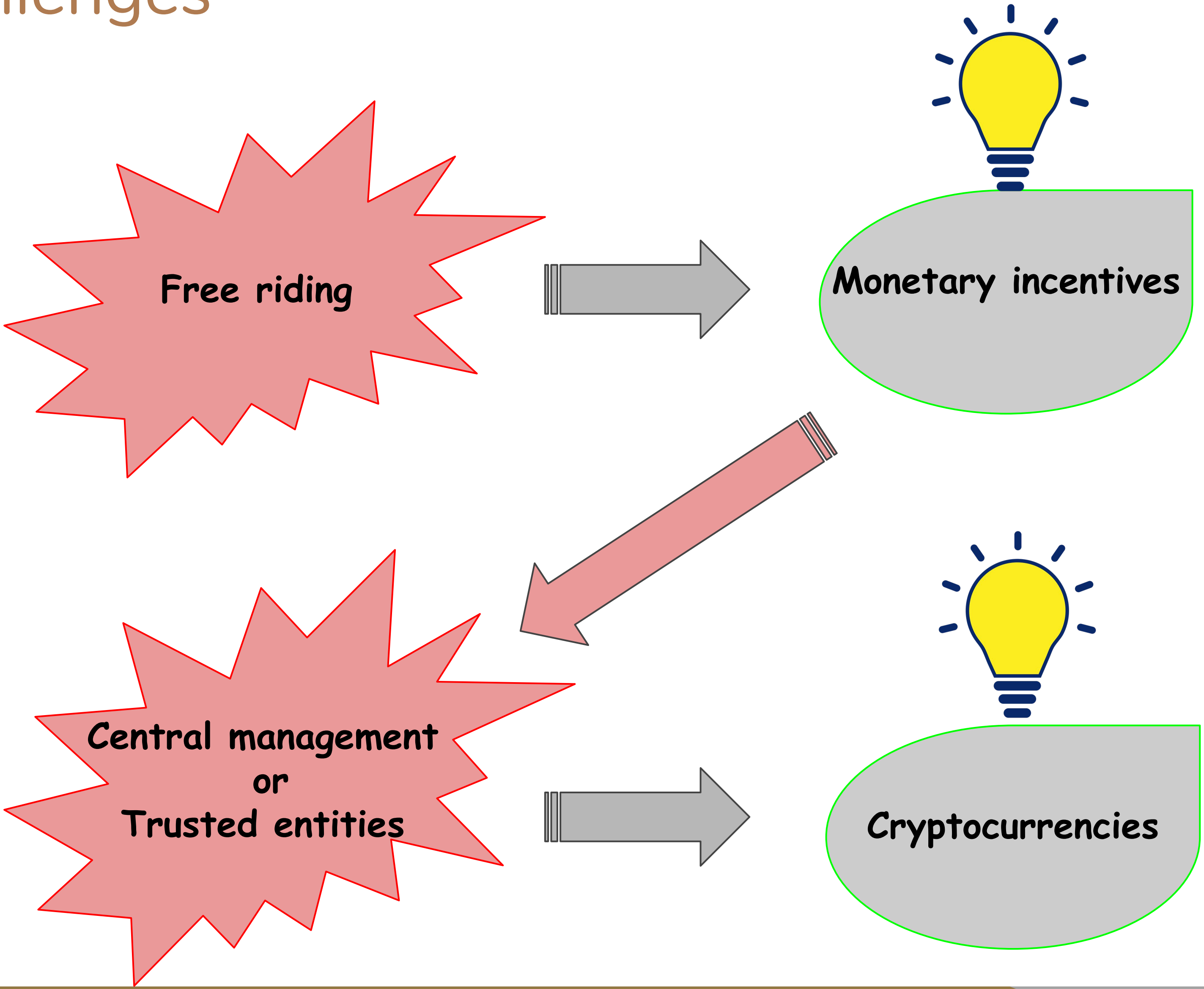
- **Drawbacks:**
  - Costly and complex business relationships.
  - Over-provisioning service needs.
  - Issues related to reachability, limited visibility, flexibility, etc.

Decentralized Services

- Utilize P2P-based models to build dynamic systems.
- **Advantages:**
  - Flexible services.
  - Easier to scale with demand.
  - Extended reachability and lower latency.
  - Democratized and transparent ecosystems.

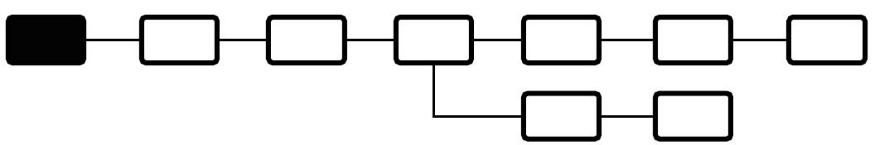


Challenges


Free riding
Monetary incentives
Central management
or
Trusted entities
Cryptocurrencies

Cryptocurrencies and Blockchain Technology

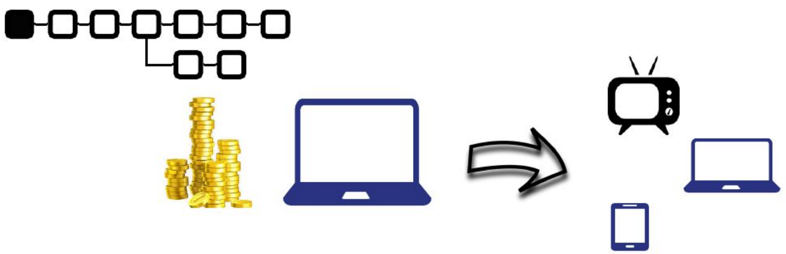
- Early systems, e.g., Bitcoin, focused on providing a currency exchange medium.
- Newer systems provide a service on top of this medium.
    - Create *distributed resource markets*.
        - Anyone can join to serve others and collect cryptocurrency coins (or tokens) in return.
    - E.g., Filecoin, Livepeer, NuCypher.



...

Examples

Service Type	**Traditional Solution**	**Blockchain-based Solution**
Payments	Banks	Bitcoin
File storage	Dropbox	Filecoin
Content distribution	Akamai	CacheCash
Key management system	Azure Key Vault	NuCypher

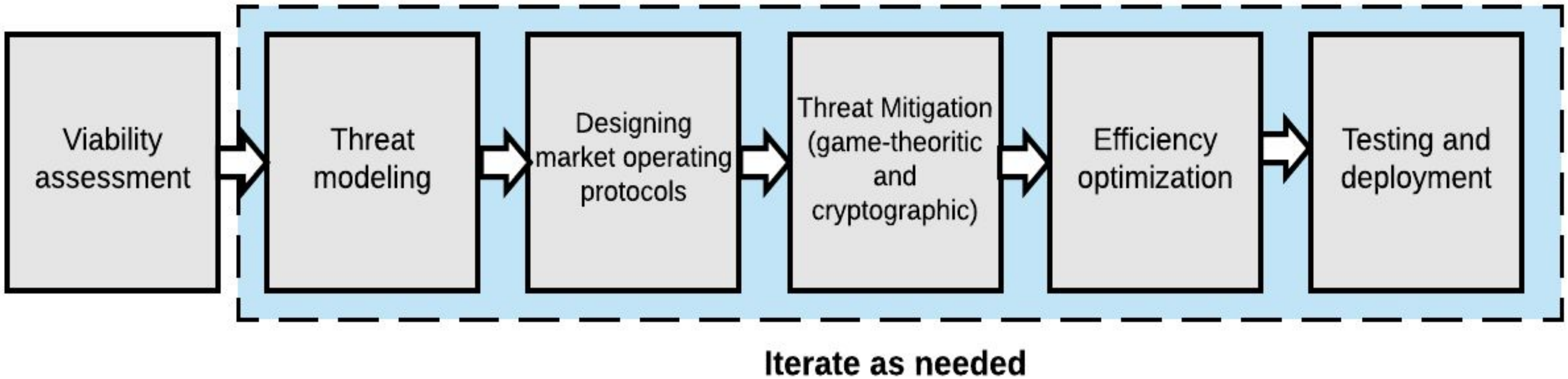


Problem solved?!

Open access work model, dealing with untrusted parties, large scale system with monetary incentives …

- These introduce a large number of security and performance issues.

A Design Framework for Distributed Resource Markets



Viability Assessment

- An important step to assess the potential for practical adoption.
- Two sides of the equation:
    - Service demand; does providing the service require specialized hardware or expensive resources?
    - Service supply; are there clients who are willing to replace traditional solutions with fully decentralized services?

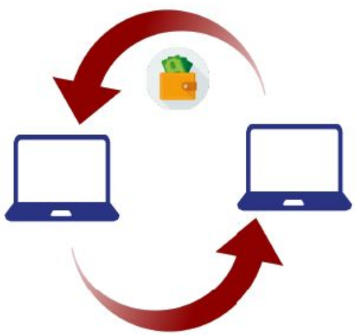
Threat Modeling

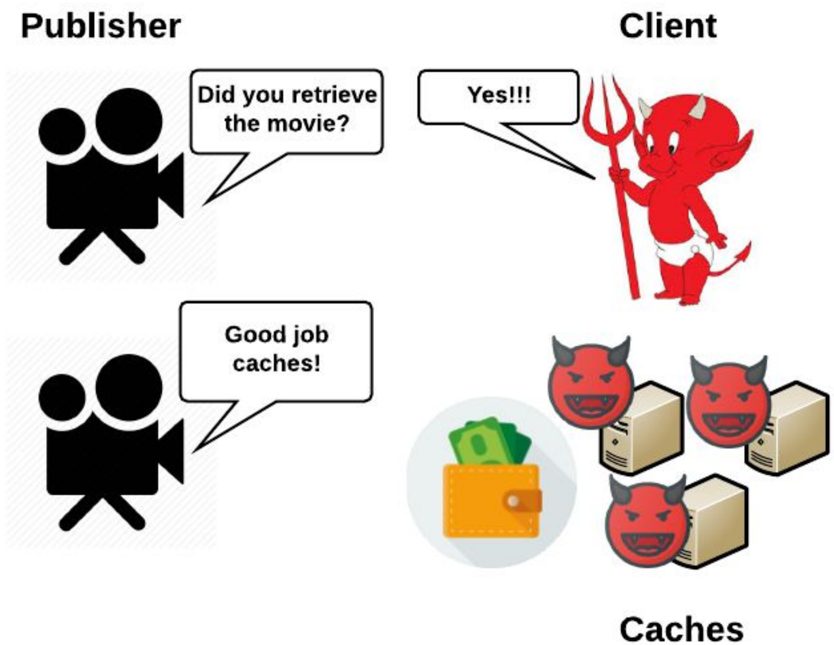
- An essential step to investigate all potential security risks.
  - A guiding map for designing a secure systems, as well as a tool for assessing security in the after design stage.
- Requires frameworks capable of:
  - Dealing with large scale distributed systems.
  - Explicitly account for financial motivations of attackers.
  - Explicitly account for collusion between attackers.
- We will explore ABC, a framework satisfies the above.

Unique Issues of Decentralized Resource Markets

- Fair exchange between two mutually-untrusted parties is impossible.
  - When to pay for the service? Before or after?
    - Malicious server may not deliver the service after being paid.
    - Malicious client may not pay after being served.
- Accounting attacks.
  - Parties may collude with each other to pretend that service had been delivered.
  - This allows servers to collect payment for free.
  - Also may lead to mismanagement of the network resources.
- Micropayments.
  - Payments of very small values.
  - Instead of paying for the whole service at once, pay in small amounts for each small service amount.

Unique Issues of Decentralized Resource Markets





Unique Issues of Decentralized Resource Markets

- Operating a decentralized resource market requires:
  - A careful design of a decentralized service-payment exchange protocol to reduce the risks of dealing with untrusted parties.
    - Represents the backbone of the market.
  - Mechanisms for service pricing, term negotiation for server recruiting, and matching protocols to match these servers with interested customers.

Cryptographic and Economic Security Measures

- Dealing with monetary incentives is challenging!
  - Not all threats can be addressed using cryptographic means.
  - Financially-motivated threats require economic mitigation techniques. Several examples.
    - Detect and punish,
    - service pricing,
    - algorithms that cost less resources when performed in an honest way than in a malicious way.
  - Usually rely on assuming rational players and require game theoretic analysis to configure incentives properly.
- Accounting attacks require proof of resource expenditure based on the service type provided.
  - E.g., proof of storage, proof of content delivery, etc.

Optimize for Efficiency

- Efficiency is a driving factor of practical adoption.
  - Testing and deployment.
  - Exploit every opportunity to boost system's performance.
  - Look for the right trade-off between security and efficiency.
- Among the main efficiency issues;
  - Reduce interactivity as possible.
  - Handle microyaments; they present a scalability and cost issues.
    - Will study this topic in details.

Iterate as Needed