Distributed Decentralized Domain Name Service

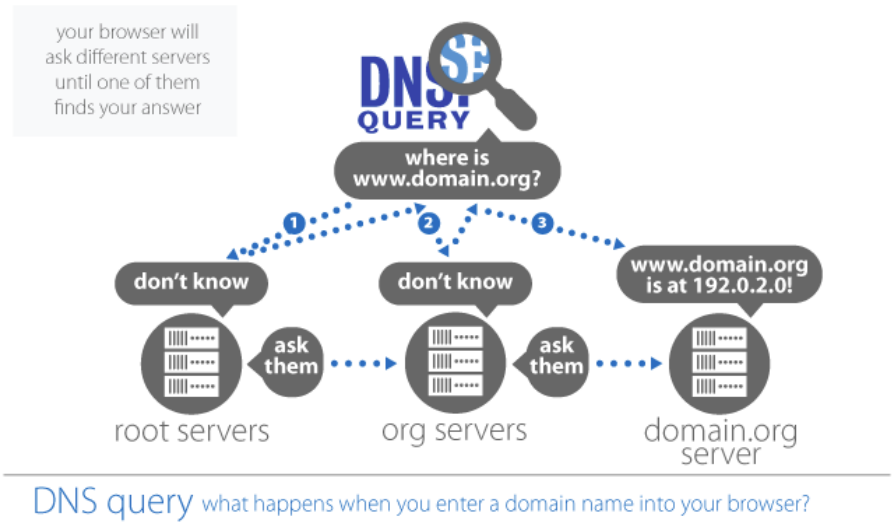
Brendan Benshoof Andrew Rosen
Anu G. Bourgeois Robert W. Harrison

Georgia State University

May 27th, 2016

Current DNS

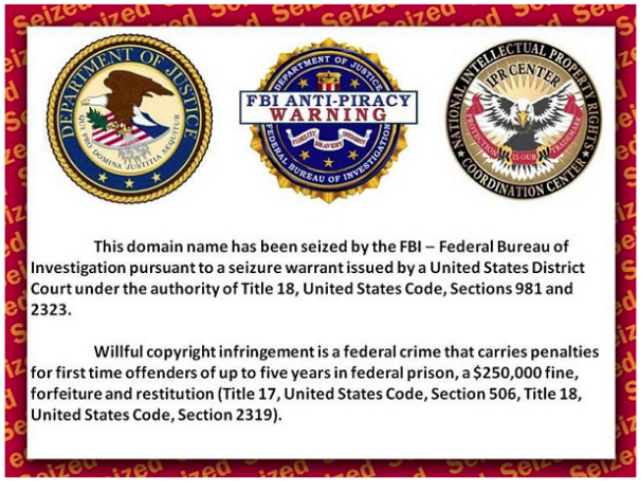
- ICANN is the final arbiter on who owns what domain
- ICANN maintains and organizes the TLD authoritative name servers
- Third party verifiers act to authenticate DNS records



Source: ICANN

Motivation

- Recent events have demonstrated that centralized authorities are not as secure a previously hoped
  - There is little cryptographic protection against the subpoena
  - Poorly constructed laws targeting DNS
  - SOPA and PIPA would have resulted in DNS blocking and compromised security
- A distributed approach for authentication is much less vulnerable



Related Work

Cox *et al.*[1] developed DDNS:

- Motivated by problem of expertise
- Fault tolerant, load-balancing, and scalable
- Easier to administer
- Found higher latencies in a P2P-based DNS
- Incentive problem – why store records for others?

[1] Cox *et al.*, "Serving DNS using a Peer-to-Peer Lookup Service" in *Peer-to-Peer Systems*, pp. 155-165, Springer, 2002

What is D^{3}NS

Distributed Decentralized Domain Name Service

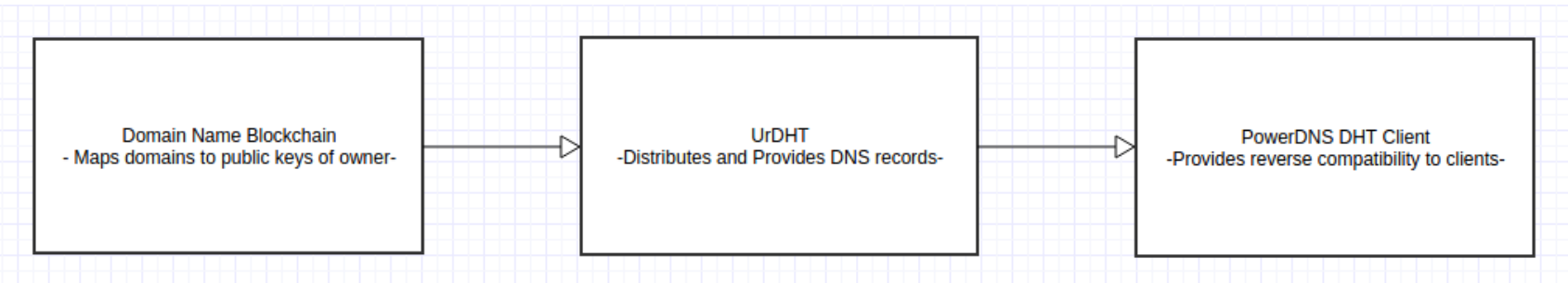
- Our Goals:
  - Decentralized authentication
  - Low latency
  - Incentive to participate
  - Backwards compatible
  - Transparent to the user

P2P-Based DNS

- The shared block chain is the final arbiter of who owns what
- The DHT organizes and maintains the authoritative TLD servers
- The block chain acts to authenticate DNS records

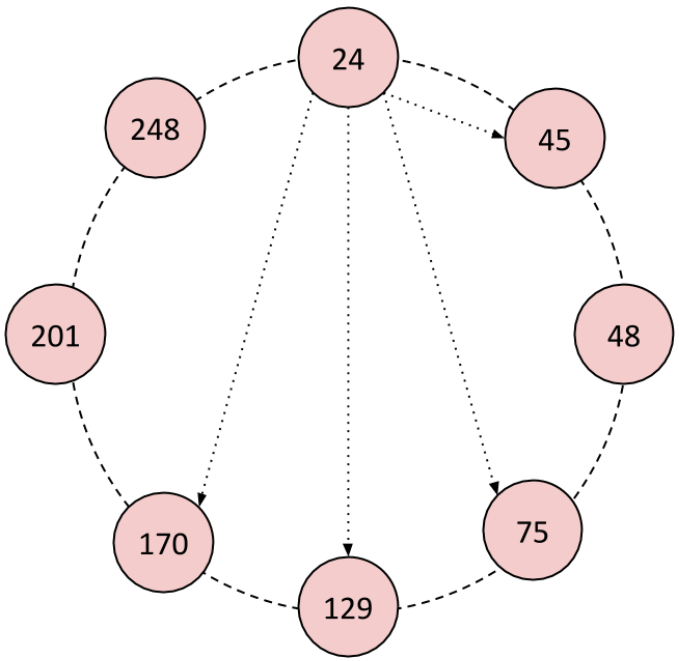
Components

- Domain Name Blockchain
- Distributed Hash Table (UrDHT)
- DNS server frontend (PowerDNS)



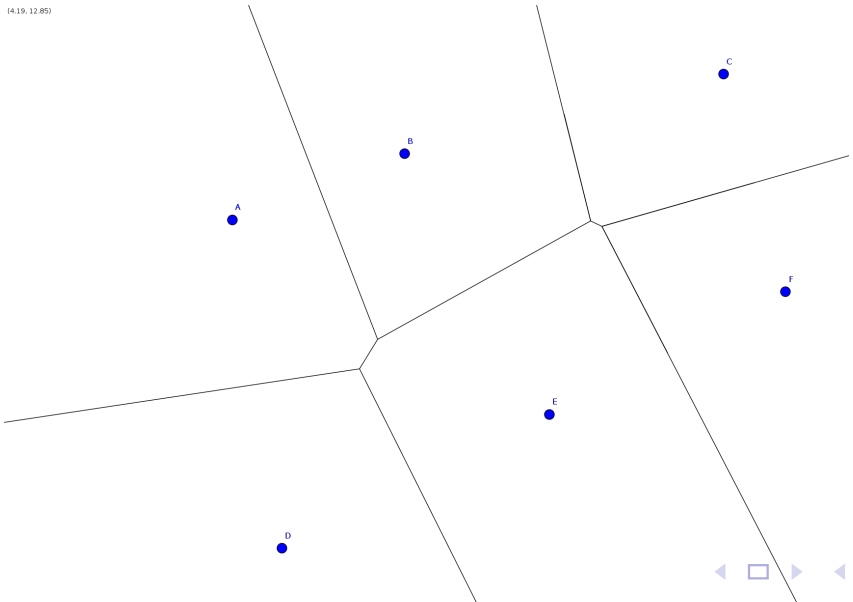
Distributed Hash Tables

- Means of organizing communication and responsibility in a P2P network
- Each peer is responsible for a verifiable span of hash values
- Facilitates one-to-one communication and one-to-many communication



UrDHT

- Uses Voronoi regions on an n-dimensional torus to assign responsibility
- Can define how to compute the regions to emulate almost any DHT topology
- Node responsibility:
  - Node is responsible for its space, defined by its neighbors
  - If a node leaves/fails, each neighbors assumes that it is responsible until corrected by maintenance



Fault Tolerance

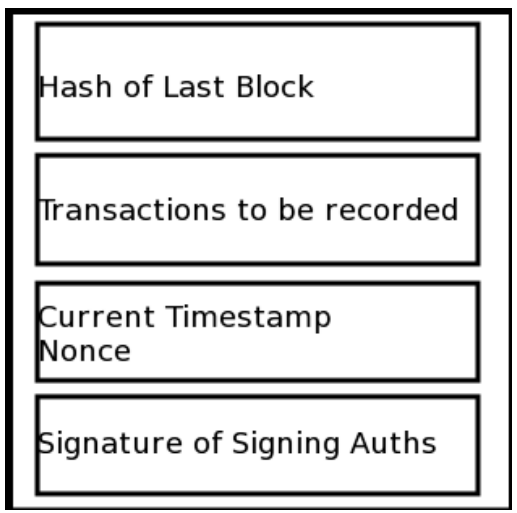
- Churn creates a period where I/O can fail
- With UrDHT:
  - Reads of backed up data are successful
  - Writes to the region are successful
  - Reads of **new** data are vulnerable until it is backed up
  - This means a much smaller window of vulnerability. Writes never fail.

Cool Things UrDHT Can Do

- Embed problem spaces into DHT topology
- Minimal latency based routing
- Basically turns routing into best-search first

DNS Blockchain

- Using a technique similar to Bitcoin, we can assign domain names as reward for mining new blocks and transfer domains between owners
- An 'owner' in this context is a public key
- These public keys can be used to verify stored DNS records by their signature records



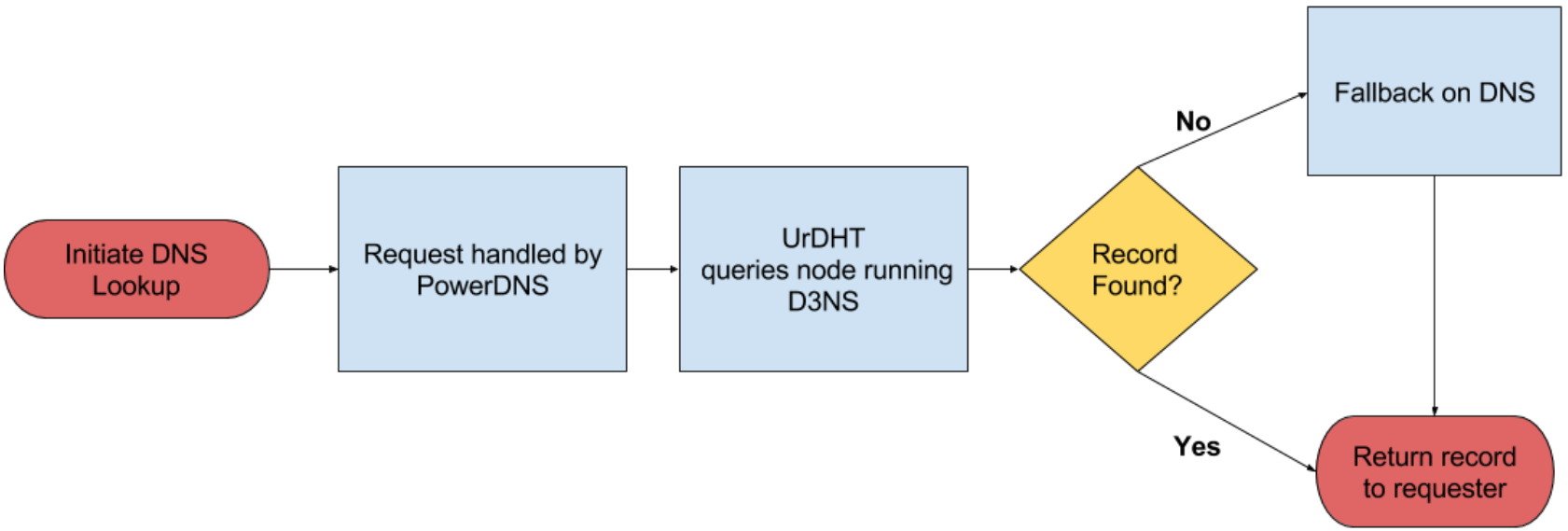
Example Record

```
DATA→CNAME.CHRONUS.TLD→IN→A→3600→-1→131.96.49.89
DATA→CHRONUS.TLD→IN→A→3600→-1→131.96.49.89
DATA→CHRONUS.TLD→IN→TXT→3600→-1→Hi mom!
DATA→CHRONUS.TLD→IN→CNAME→3600→-1→CNAME.CHRONUS.TLD
DATA→CHRONUS.TLD→IN→NS→3600→-1→CNAME.CHRONUS.TLD
DATA→CHRONUS.TLD→IN→SOA→3600→-1→CNAME.CHRONUS.TLD
ADMIN.CHRONUS.TLD 2013111900 172800 900 1209600 3600

APC5¨RSEOT?ò¯BELDC4~9SS2ßreNÛ×çi*
®ûUNBHÊ¿L/þACK¡ÜYDC1UýCãÏU@úÉPLUNBHáM§SGCI
úDC1àeÖãSPAêõNAKT
```

PowerDNS

- Well established authoritative DNS server software.
- Provides easy interface for custom applications.
- Serves the DNS requests for DHT client.



DNS Registration

- New domain names can be:
  - Awarded as part of mining process
  - Bought as a voucher
- Domain names can be transferred between owners by creating a new record
- These transactions are recorded in the blockchain

Ramifications

- More resilient against DDOS attacks
  - No top of hierarchy to attack
  - Attacker needs to target large number of servers
- Decentralization of authentication
  - Authentication is baked into replication
  - Changes to a record must be signed by the owner

Conclusions

- Deployable prototype of a decentralized and distributed top-level DNS
- Stronger robustness
- Fully reverse compatible
- Offers decentralized authentication
- Improvements to latency
- Any organization can create their own secure verification server

Future Work

- Optimize caching
- Handle backup comorbidity
- Further security measures
- Larger scale implementation