



# What is D<sup>3</sup>NS

## Distributed Decentralized Domain Name Service

- The goal is to create a secure distributed DNS
- Requirements:
  - Decentralization
  - Authentication
  - Backwards Compatible

# Motivation

- Recent events have demonstrated that centralized authorities are not as secure as 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.*<sup>1</sup> developed DDNS:

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

---

<sup>1</sup>Cox *et al.*, “Serving DNS using a Peer-to-Peer Lookup Service” in *Peer-to-Peer Systems*, pp. 155-165, Springer, 2002

# 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



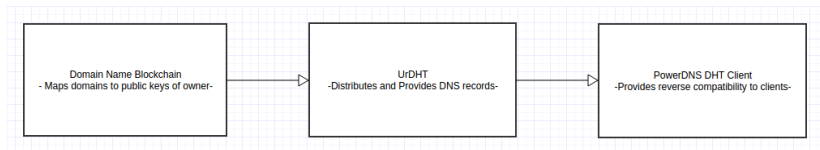
Figure: DNS operation. Source: ICANN

# 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

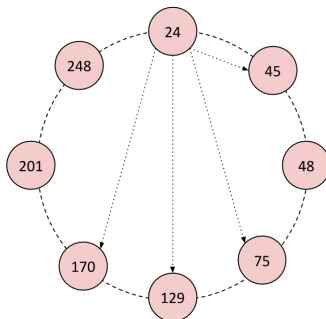
# Components

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



# Distributed Hash Tables

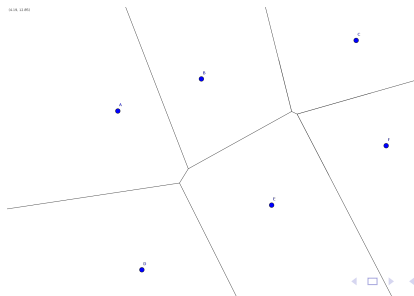
- Means of organizing communication and responsibility in a P2P network
- Each peer is responsible for a verifiable span of hash values
- Facilitates one-to-one communication and one-to-many communication





# 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

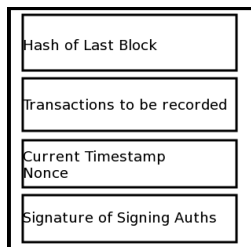
- Churn creates a period where i/o can fail.
- With UrDHT:
  - Reads of backed up data are successful.
  - Writes to the region are successful.
  - Reads of **new** data are vulnerable until it is backed up.
  - This means a much smaller window of vulnerability. Writes never fail.

# Cool Thing 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"RS[EOT?ò~BELDC4~9SS2BrenÜxçi*
@ûUNBHÊ¿L/pACKjÛYDC1UýcãIU@úÉPLUNBHámS SGC1
úDC1ãeöãSPAêöNAKT
```

# PowerDNS

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

# Flowchart

Insert flowchart here or as own slide.

# Man in the Middle In a DHT

- Need to have a distributed, reliable way to authenticate
- Given: an existing network where nodes have exchanged keys securely
- Given: a new peer who wishes to join the network and share their public key



# Prevention

- At least 2 members of the network interrogate the new peer for its public key
- Those interrogators compare their results
- If those results match:
  - The new peer creates an authentication record
  - The interrogators sign that record
  - The new record is distributed across the network
- If the results do not match
  - An attack is detected and reported to the new peer by all authenticating servers.
  - A member of the network may make a ban of the compromised peer
  - Otherwise the joining process can be repeated.

# Conclusions

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

# Future Work

- Optimize caching
- Backup Comorbidity
- Further security measures