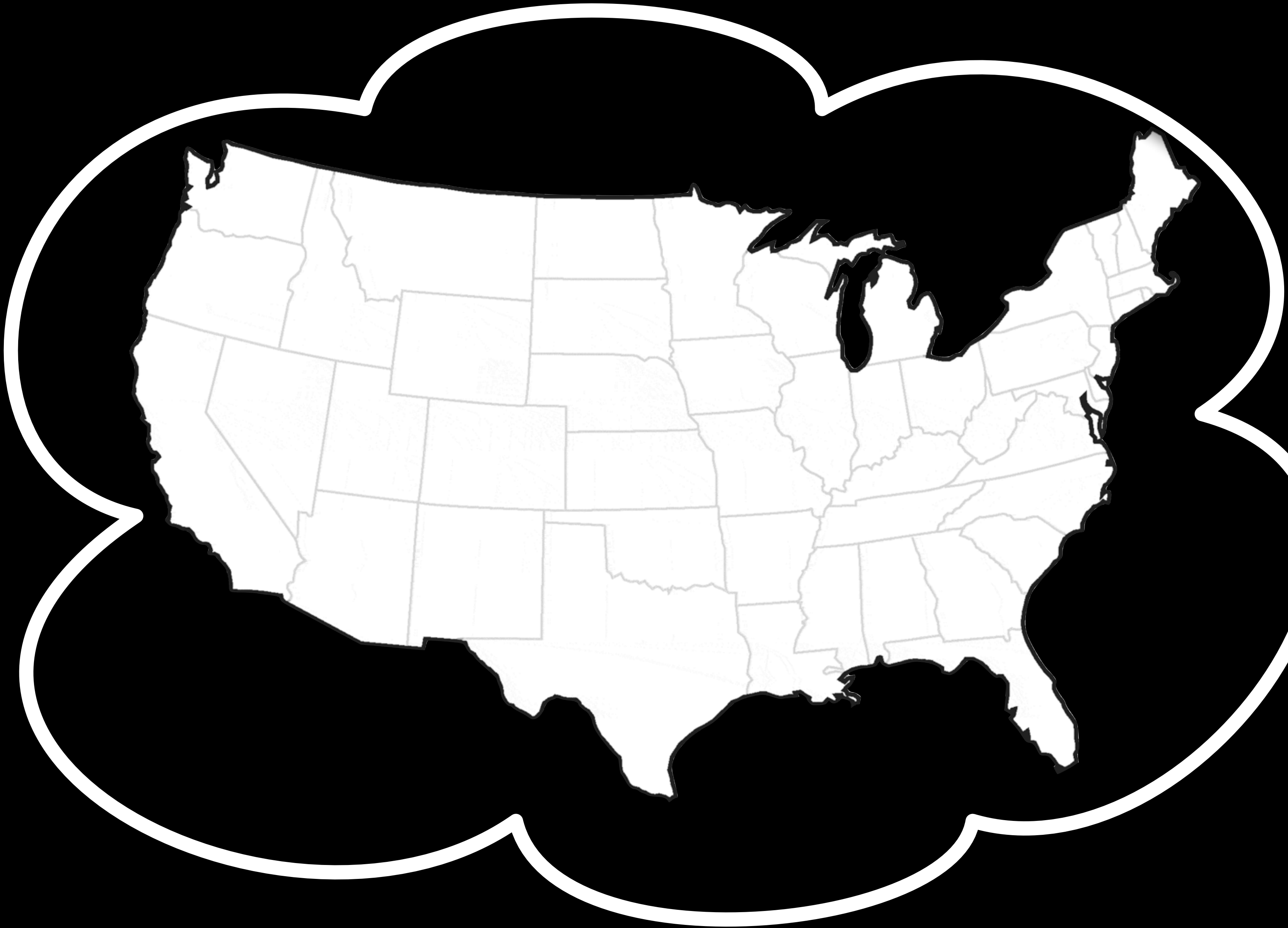
HotCloud 2011

Data Sovereignty

The importance of geolocating data in the cloud

Zachary N J Peterson
Mark Gondree
Rob Beverly

Your Data is Here



But, maybe it should be here

Or Here?

Breaking the Abstraction

- Is data within some **political** boundary
    - Privacy protections
    - Intellectual property protections
    - Regulatory compliance
- Has data been **replicated**

Existing Notions of Location in the Cloud

Regions of service

Content-distribution networks

Location guaranteed only by **service-level agreements** and **quality of service** metrics

No **interfaces** or external **techniques** for establishing the location of remote data

Data Sovereignty

Data Sovereignty

Protocols for establishing the **location** and **authenticity** of **data** in the cloud

In scope: Efficiently positioning **some copy** of data within some geopolitical boundary

Not in scope: the location of **any copy** of data

State of the Art

Geolocation

Geolocation of **hosts** (NICs)

Evidence gathering (whois, extrinsic evidence)

Delay-based measurements

Wang *et al.* NSDI '10: Street-level geolocation

Possession of Data

Provable Data Possession (PDP) &
Proofs of Retrievability (POR)

Probabilistic challenge & response protocols

Designed to **minimize** storage, computation, communication complexity

Techniques: Homomorphic signatures, PRFs, BLS signatures, MACs

Naïve Composition

Naïvely composing geolocation & PDP (e.g. serially) provides **limited assurance**

Data exists **somewhere**, and the **responder** is within some physical bound

(Not: the **data** exists within some physical bound)

Adversaries

DS considers a more **powerful** adversary

- One who may actively fool the challenger
- e.g. act as **proxy** for remote storage,
  cache subsets of data,
  manipulate delay measurements

Adding delay **increases** perceived distance

An Initial Approach

An Initial Approach

Leverage MAC-PDP:

Tag: t_i = $\mathrm{HMAC}_k(D_i)$

Store: $<D_i, t_i>$

Challenge: $<D_c, t_c>$ for c indices

Verify: $\mathrm{HMAC}_k(D_c) =? t_c$

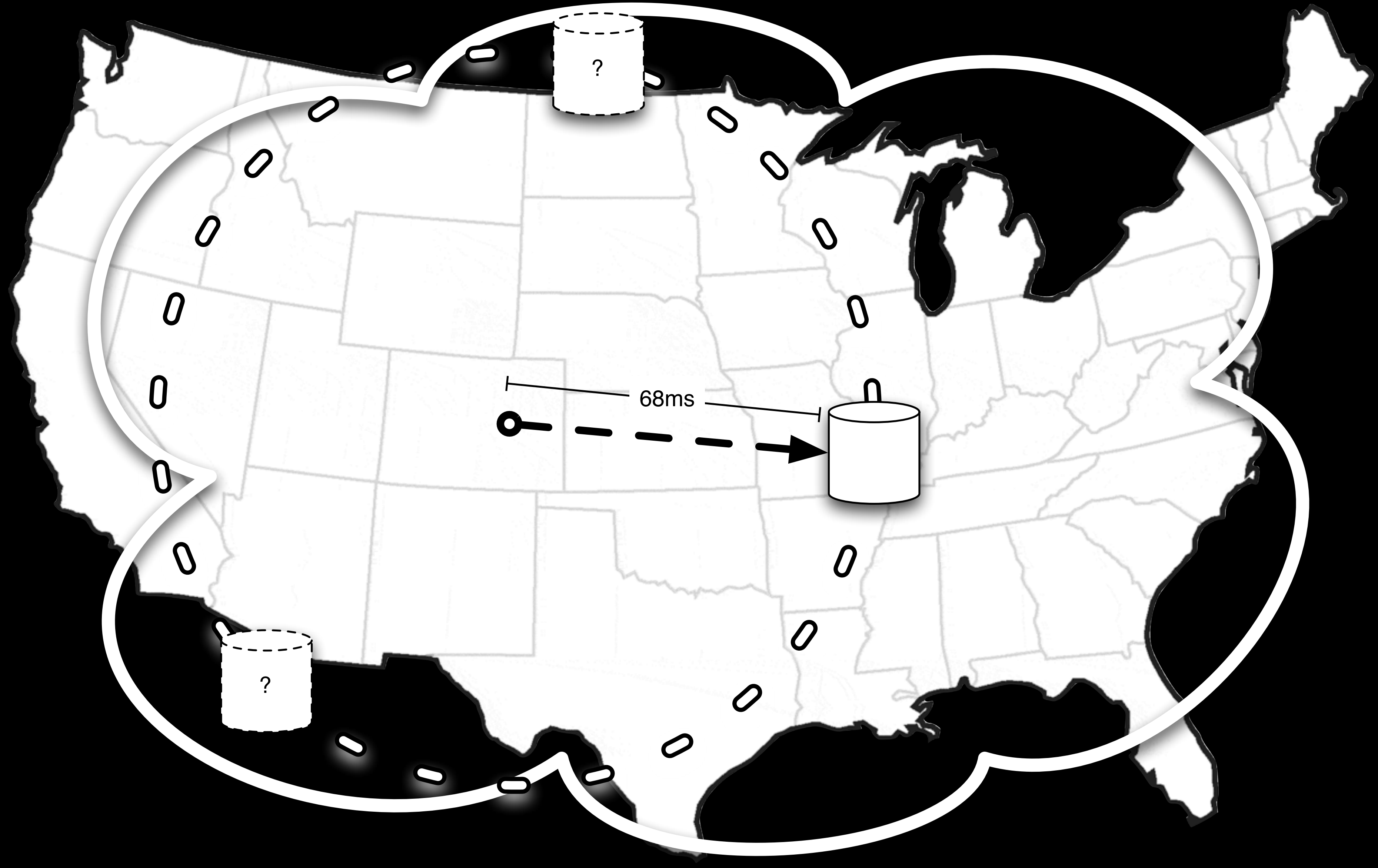
An Initial Approach

Augment MAC-PDP with network delay measurements

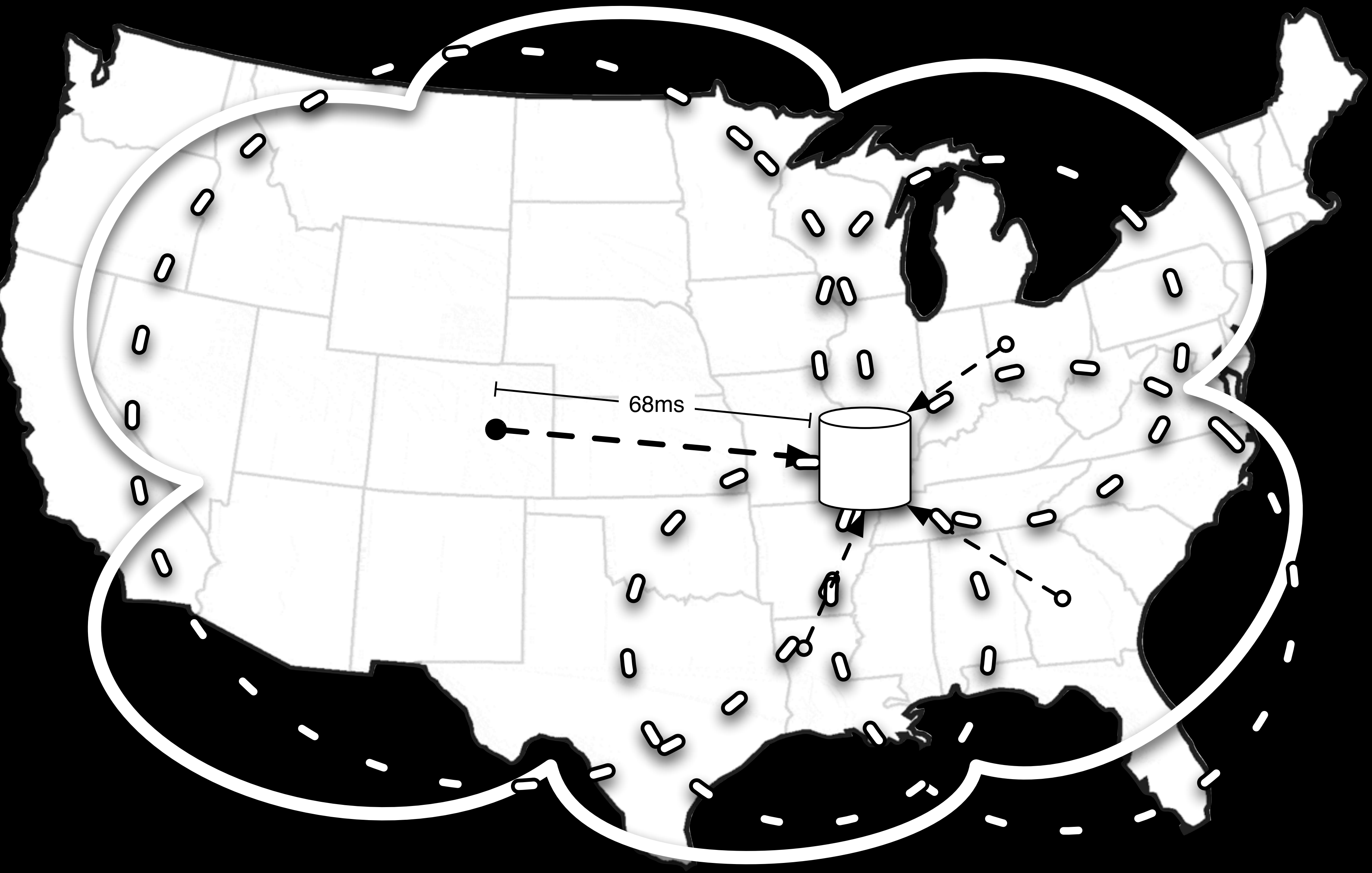
Query blocks one at a time, randomly

Measure the response time

Single response verifies data authenticity and calculates distance



Single Challenger



Multiple Challengers

An Initial Approach

Requires no server-side computation

Can be implemented on existing infrastructure, as part of an SLA compliance tool

But, at a high communication cost

And, susceptible to honest, variable overheads

Future Directions

Evaluation of our initial idea

Landmark placement and operation

More efficient and less adversarial DS schemes

- Given **existing** infrastructure
- Given some **future** infrastructure

Ways to bind **computation** to a location