LOTS OF COPIES KEEP STUFF SAFE

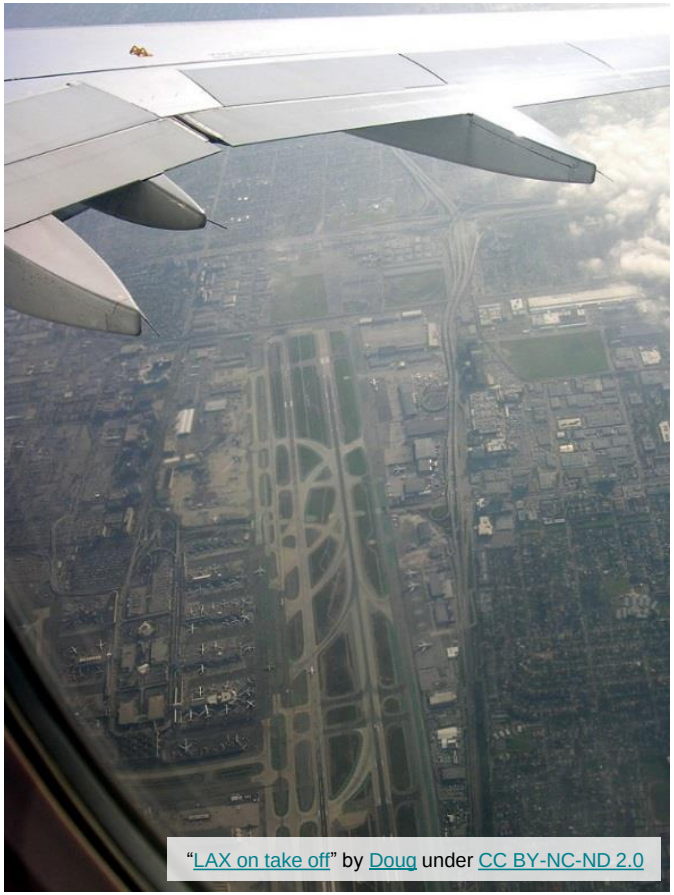
Lots More LOCKSS for Web Archiving: Boons from the LOCKSS Software Re-Architecture

Nicholas Taylor (@nullhandle)
Program Manager, LOCKSS and Web Archiving
Stanford University Libraries

Web Archiving Week
15 June 2017

overview

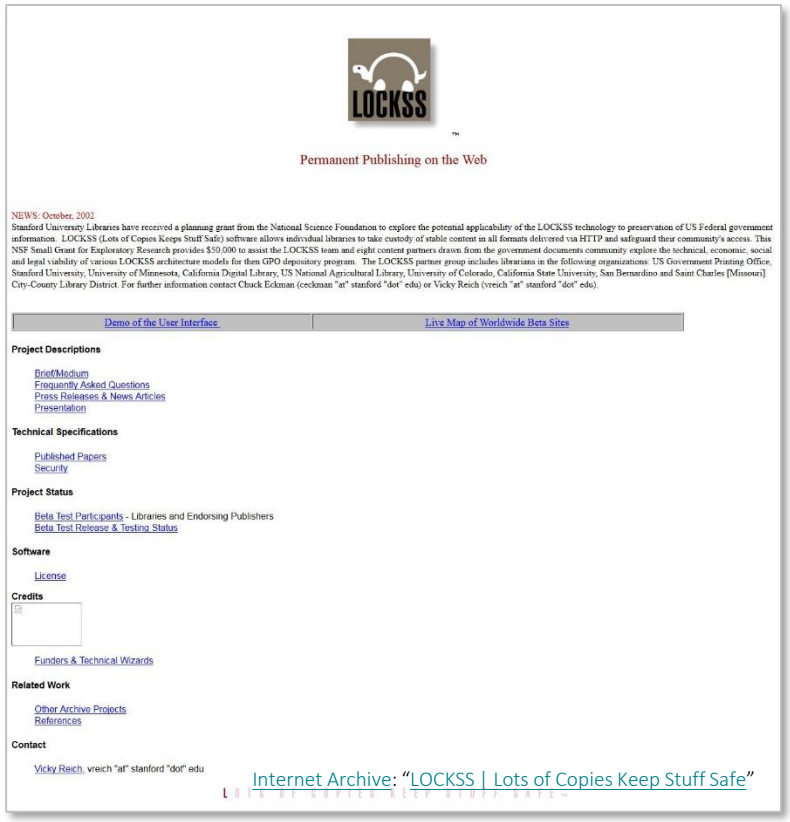
- LOCKSS background
- software re-architecture
- software components
- roadmap



LOCKSS Background

beginnings

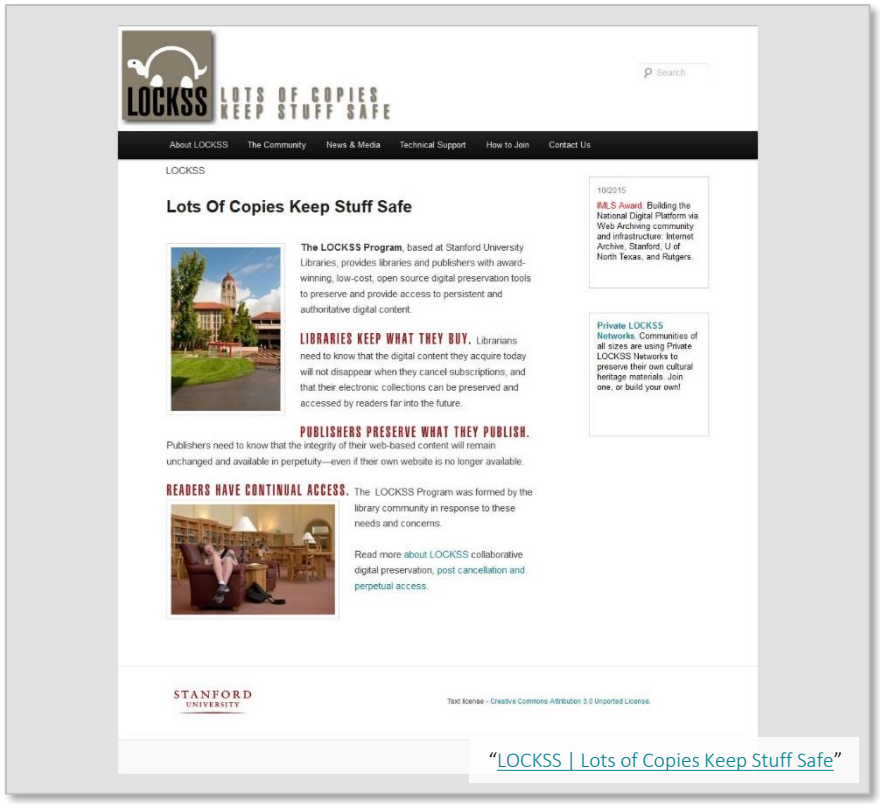
- a serials librarian + a computer scientist
- print journals → Web
- conserve library's role as preserver
  - collect from publishers' websites
  - preserve w/ cheap, distributed, library-managed hardware
  - disseminate when unavailable from publisher



Internet Archive: "LOCKSS | Lots of Copies Keep Stuff Safe"

present day

- financially self-sustaining
- tens of networks
- hundreds of institutions
- all types of content

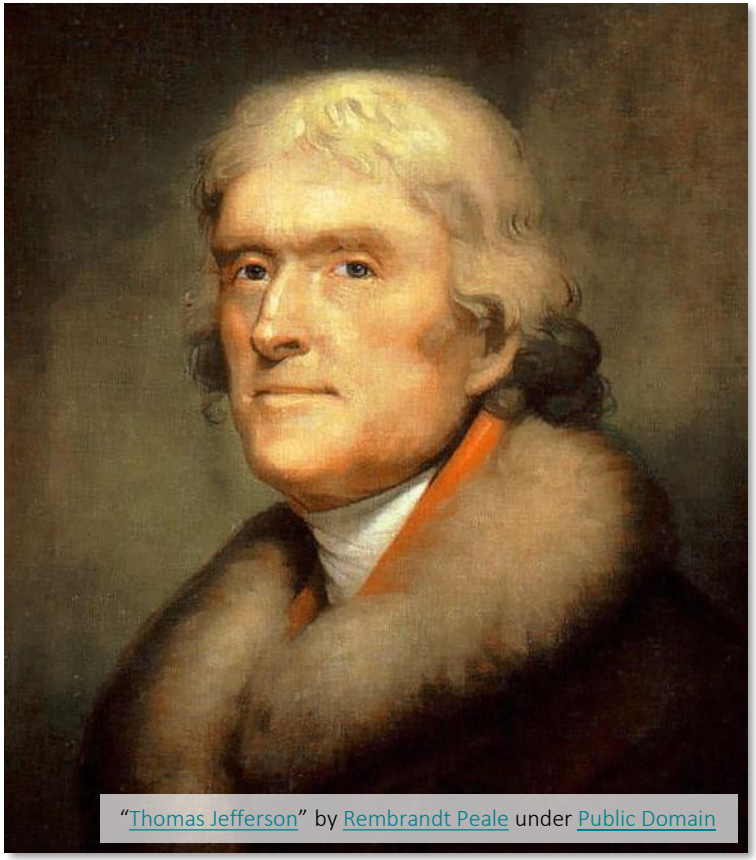


"LOCKSS | Lots of Copies Keep Stuff Safe"

lots of copies

"The lost cannot be recovered; but let us save what remains: not by vaults and locks which fence them from the public eye and use, in consigning them to the waste of time, but by such a multiplication of copies, as shall place them beyond the reach of accident."

National Archives: "from Thomas Jefferson to Ebenezer Hazard, 18 February 1791"



"Thomas Jefferson" by Rembrandt Peale under Public Domain

LOCKSS

decentralized copies

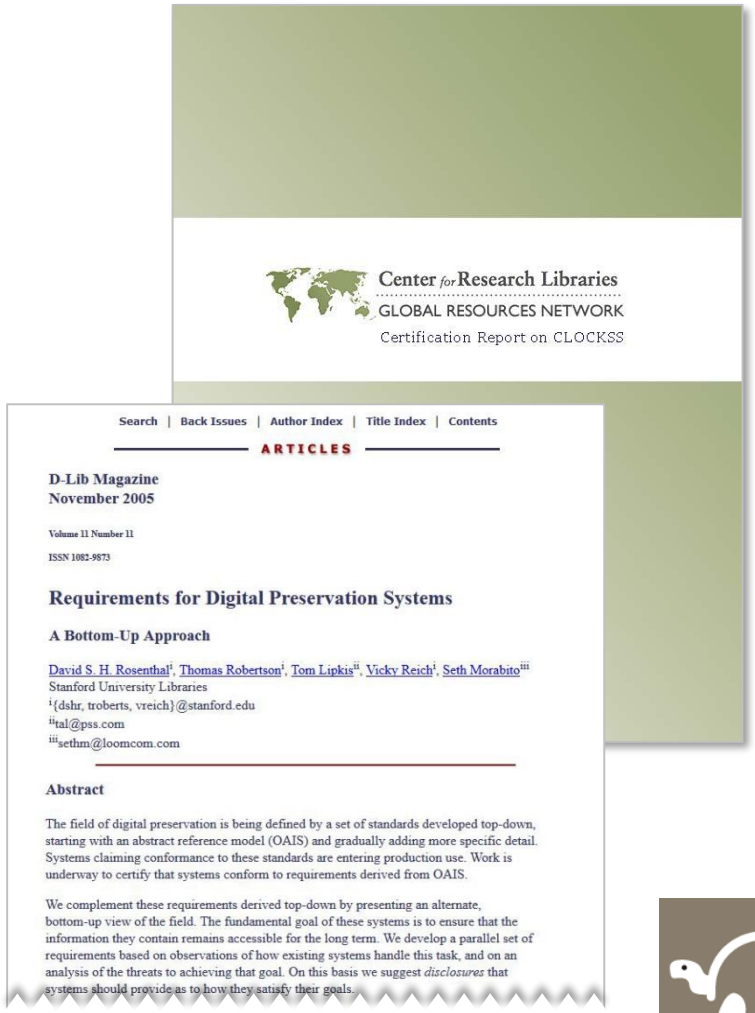
- no monopoly on copy-making
- independent, de-correlated copies
- no central point of failure or vulnerability
- local custody, self-determination

articulated threat model

- long-term **bit integrity** is a hard problem
- more (correlated) copies **doesn't necessarily** keep stuff safe
- don't underestimate:
  - people making **mistakes**
  - **attacks** on information
  - organizational **failure**

community-validated

- built upon peer-reviewed research
- successfully operating for almost 20 years
- CRL TRAC assessment of CLOCKSS
  - overall score matching previous best
  - only perfect technology score awarded to date

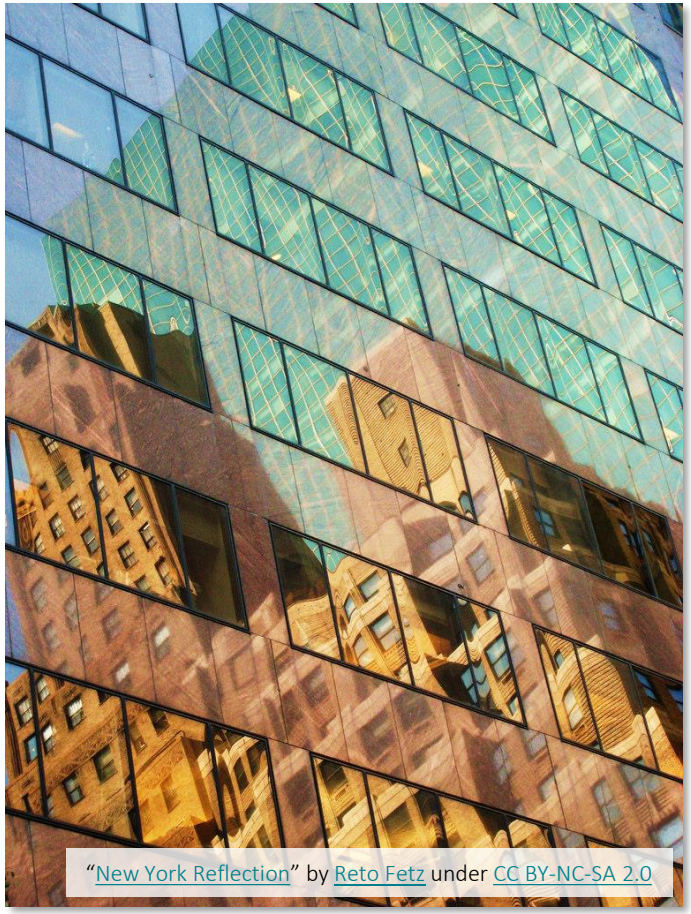


LOCKSS

Software Re-Architecture

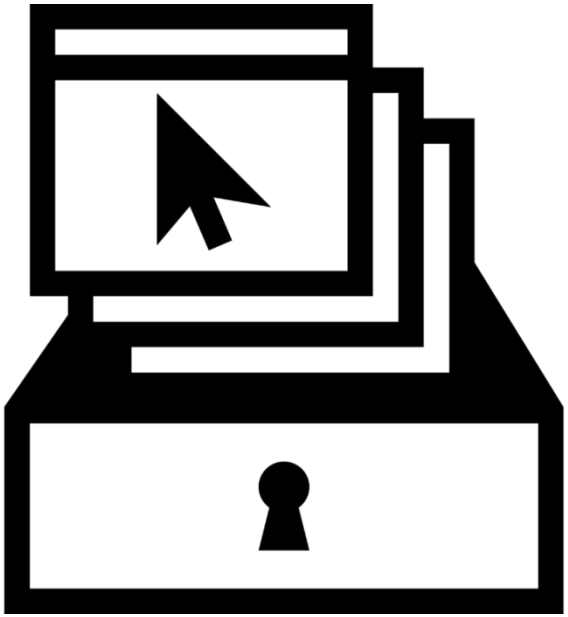
why re-architect LOCKSS?

- reduce support + operations costs
- de-silo components + enable external integration
- prepare to evolve w/ the Web



aligning with web archiving

Web ARChive (WARC) format



compatible technologies

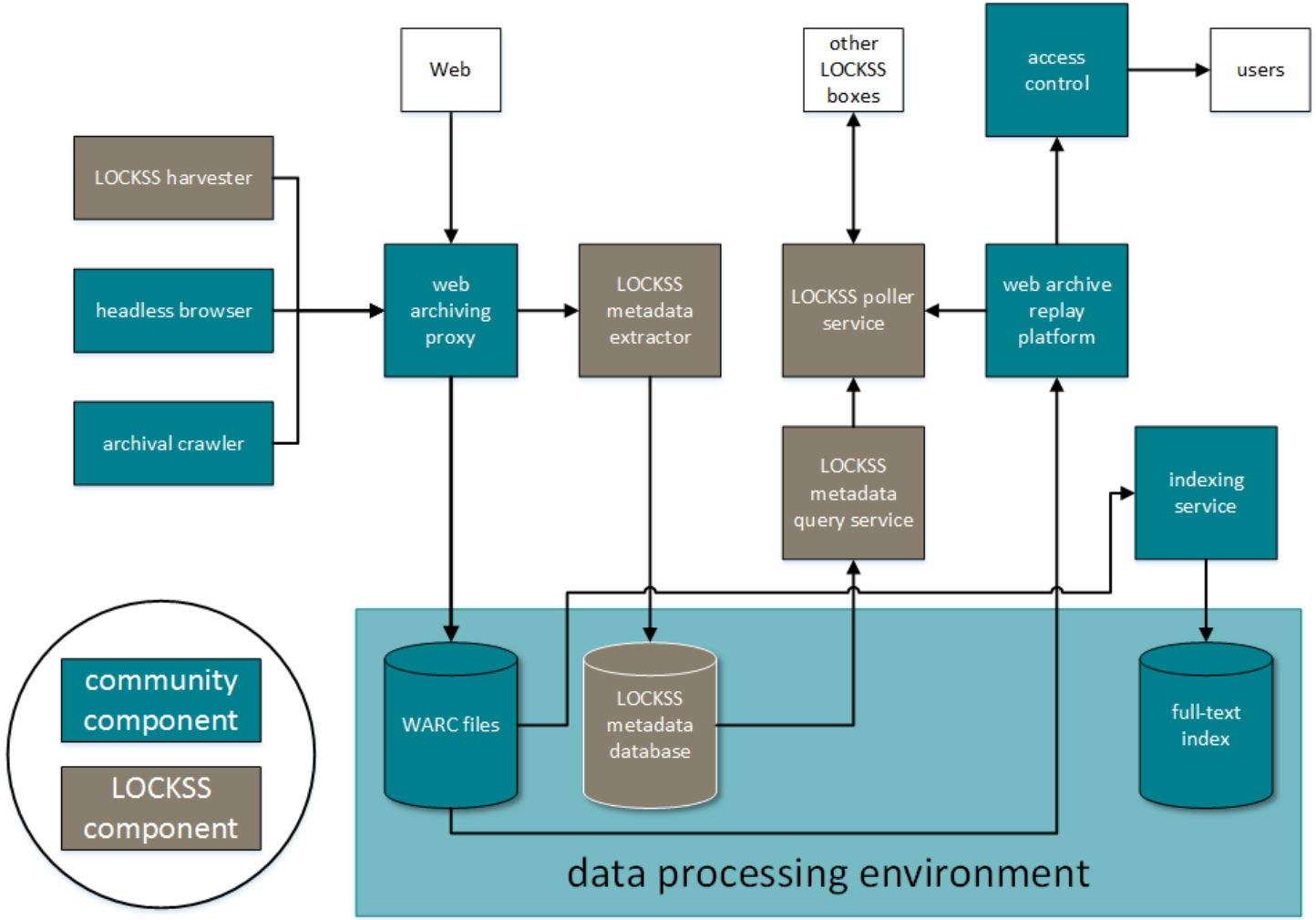
- Heritrix
- OpenWayback
- WarcBase
- Web Archiving Proxy

web archiving system APIs (WASAPI)

National Digital Platform Projects funded in August 2015

Systems Interoperability and Collaborative Development for Web Archiving (LG-71-15-0174-15): The Internet Archive, working with partner organizations University of North Texas, Rutgers University, and Stanford University Library will undertake a two-year research project to explore techniques that can expand national web archiving capacity in several areas.

leveraging community components



Software Components

bibliographic metadata extraction

functionality

- for web harvest + file transfer content
- map values in DOM tree to metadata fields
- retrieve downloadable metadata from expected URL patterns
- parse RIS + XML by schema

fields

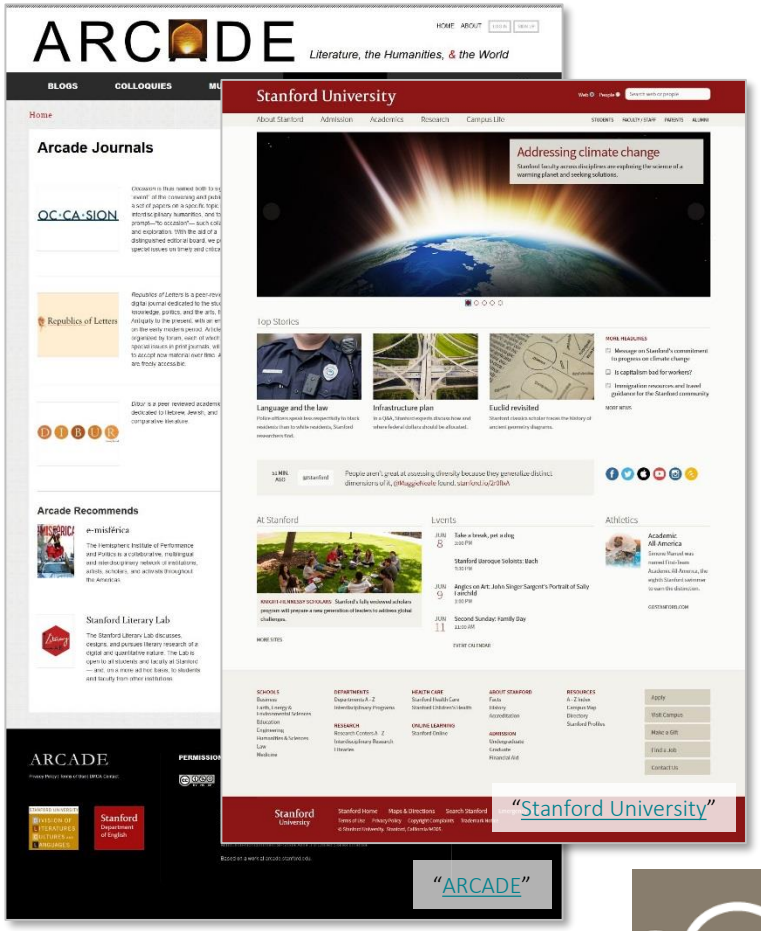
- creator
- title
- published year
- volume
- issue
- article name

publisher/platform heuristics

- plugins making bibliographic objects + their metadata on many publishing platforms machine-intelligible
- a framework for creating such plugins
- useful in absence of standard conventions, e.g. Signposting

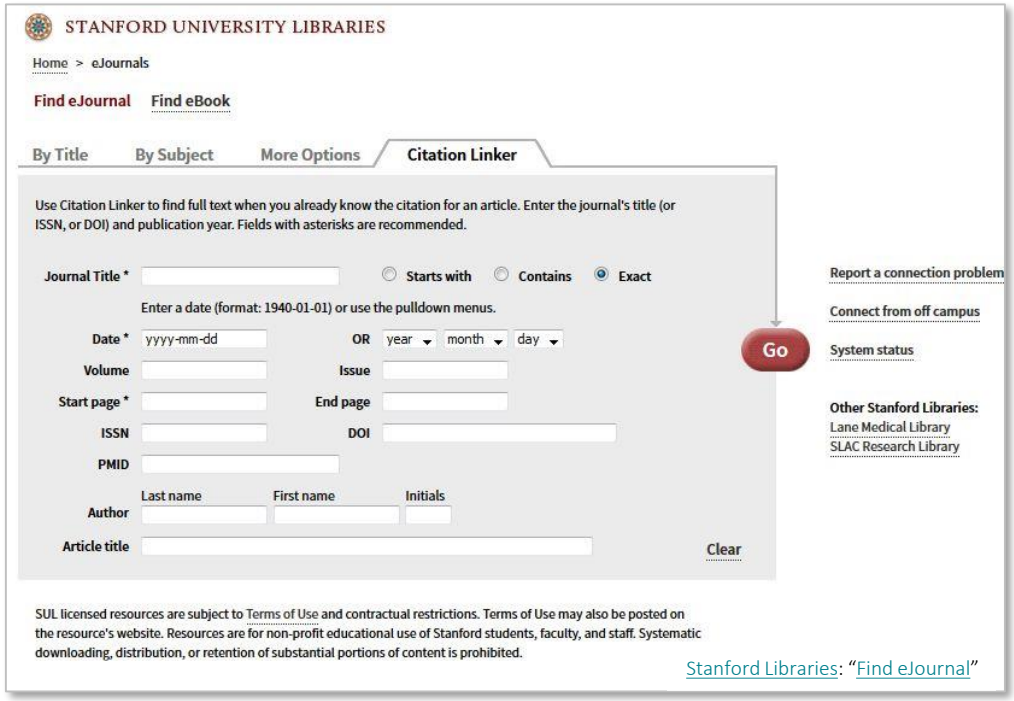
use cases for metadata extraction

- apply to consistent subsets of content in larger corpora
- curate OA materials within broader crawls
- retrieve faculty publications posted online, license allowing
- describe sub-sites collected while self-archiving from a single institutional CMS



discovery via bibliographic metadata

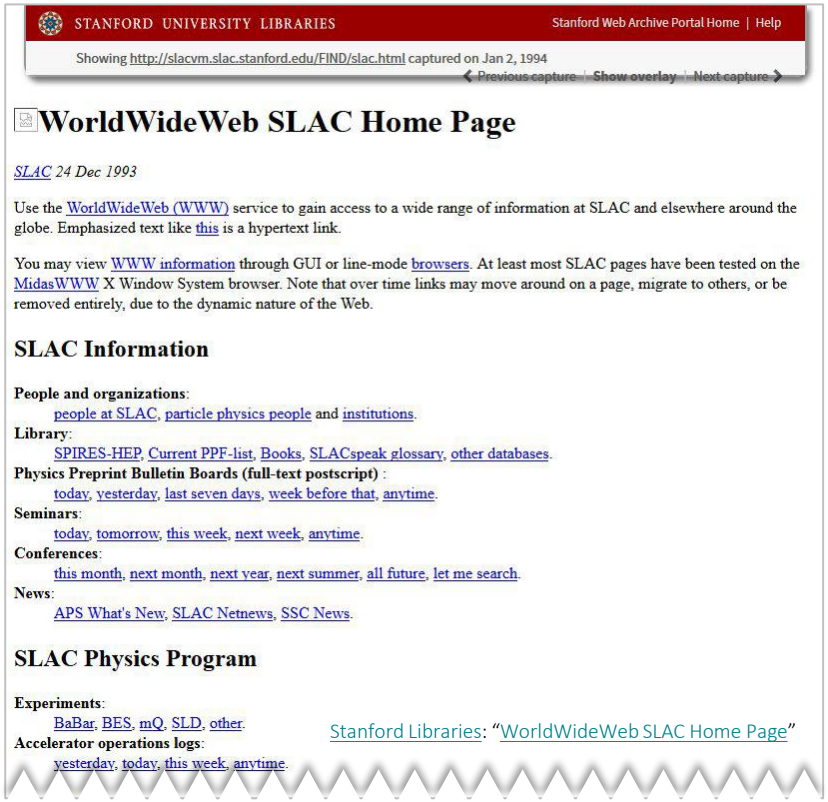
- submit DOI or OpenURL query
- get OpenWayback access URLs
- integrate w/ OpenURL resolver



Stanford Libraries: "Find eJournal"

on-access format migration

- as described in 2005 D-Lib paper by DSHR et al
- low obsolescence risk suggested by research from Holden, Jackson
- implement upstream in OpenWayback
- example: X-BitMap → GIF migration



Stanford Libraries: "WorldWideWeb SLAC Home Page"

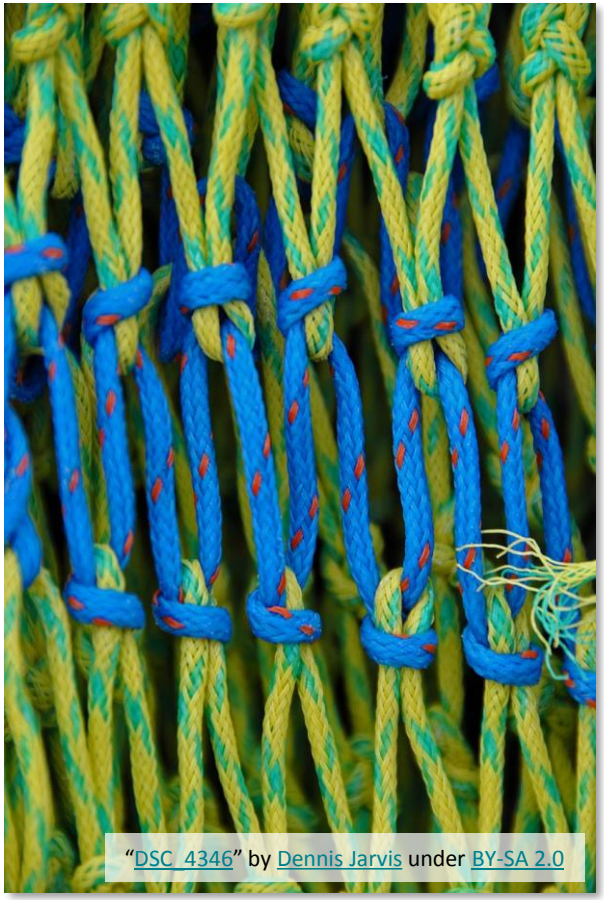
on-access format migration

- as described in 2005 D-Lib paper by DSHR et al
- low obsolescence risk suggested by research from Holden, Jackson
- implement upstream in OpenWayback
- example: X-BitMap → GIF migration

Stanford Libraries: "WorldWideWeb SLAC Home Page"

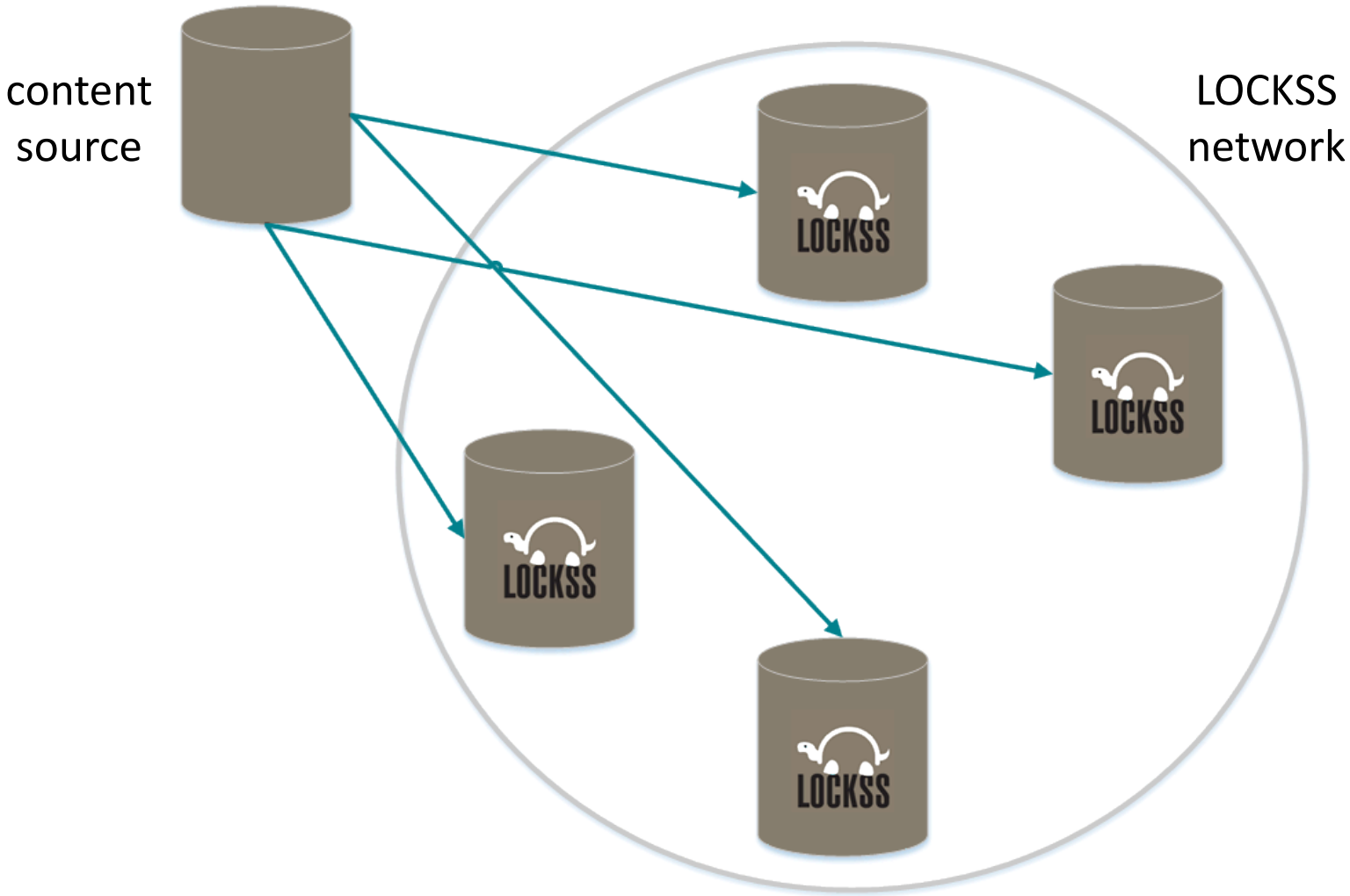
audit + repair protocol

- core preservation capability
- network nodes conduct polls to validate integrity of distributed copies of data chunks
- more nodes = more security
  - more nodes can be down
  - more copies can be corrupted
  - …and polls will still conclude
- nonces force re-hashing
- peers are untrusted
- polls are slow, to allow damage detection

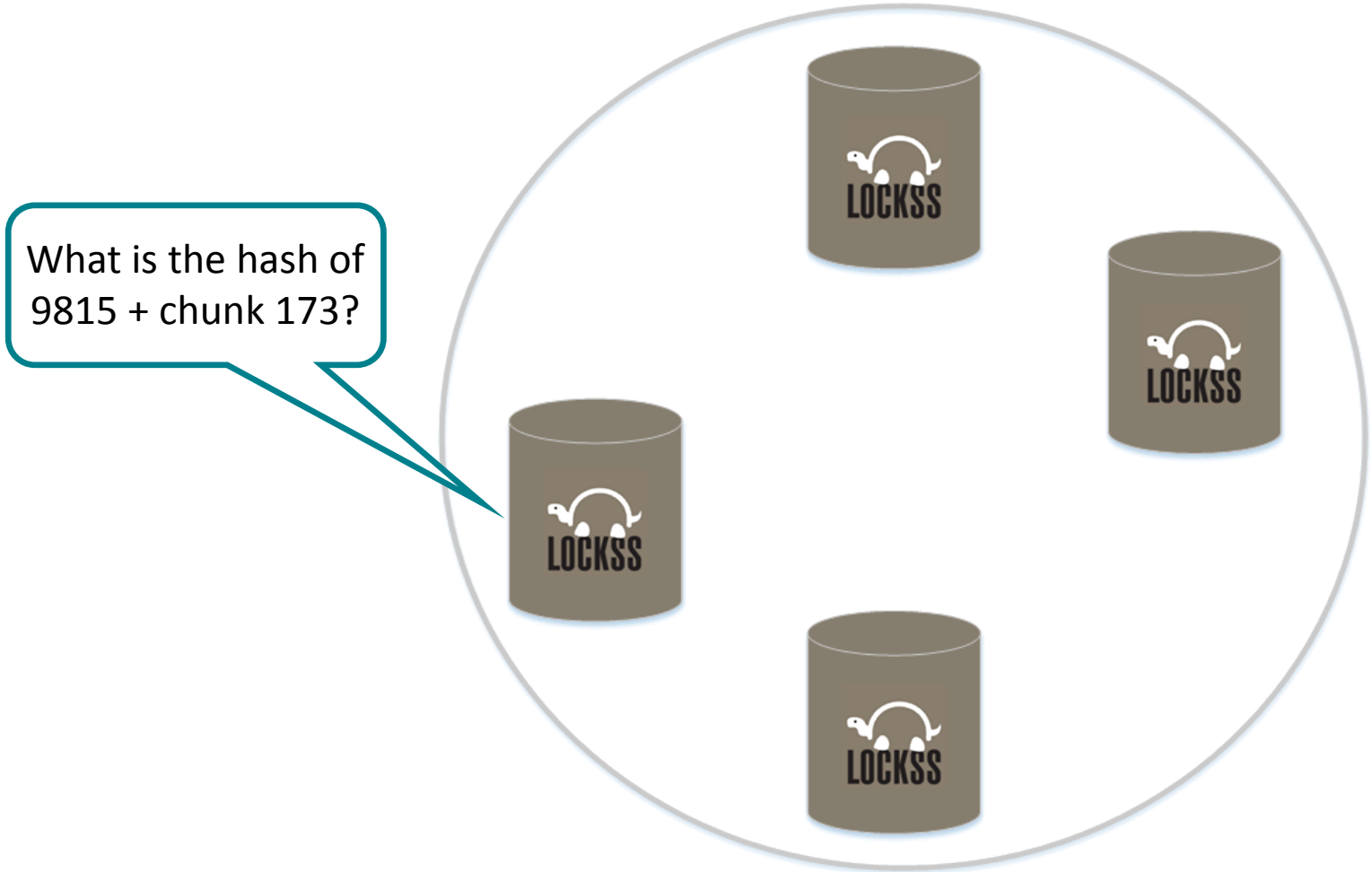


LOCKSS

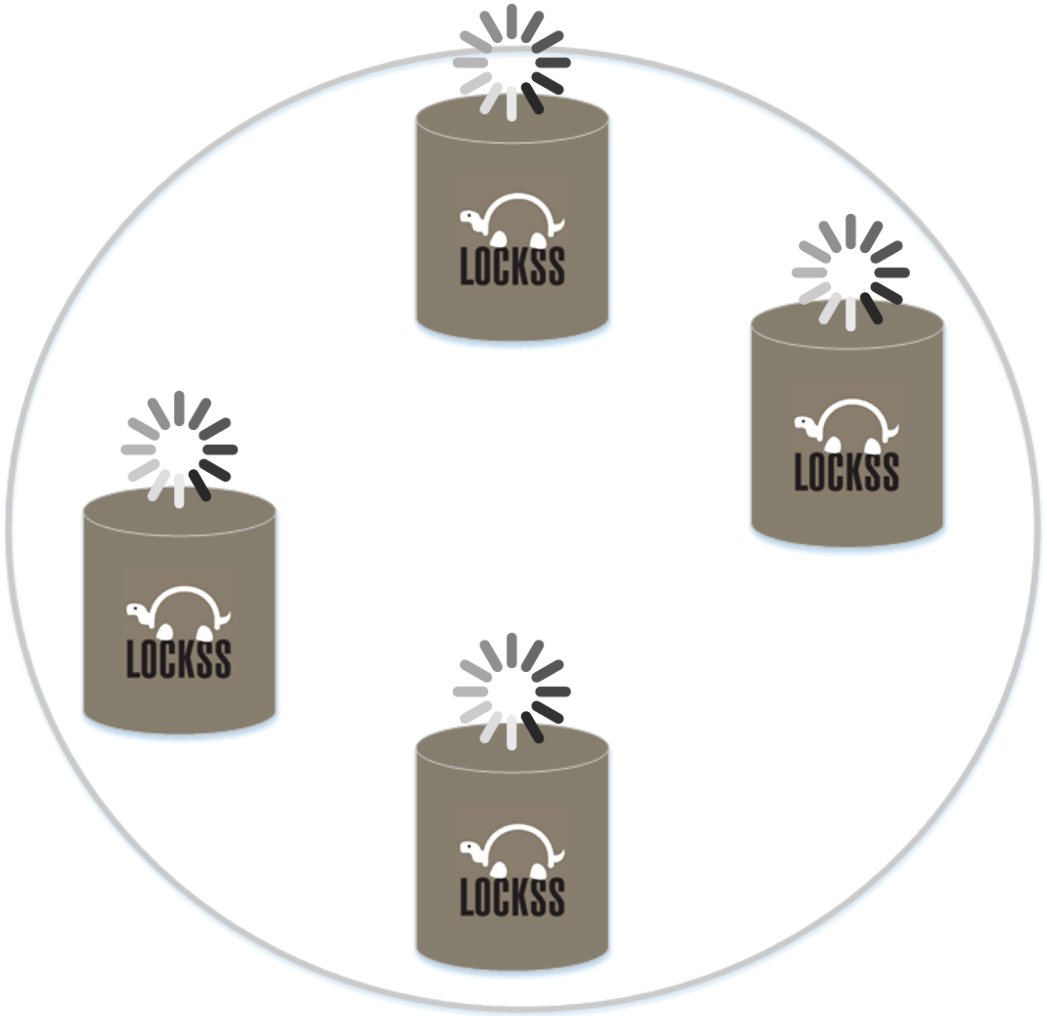
harvest from content source



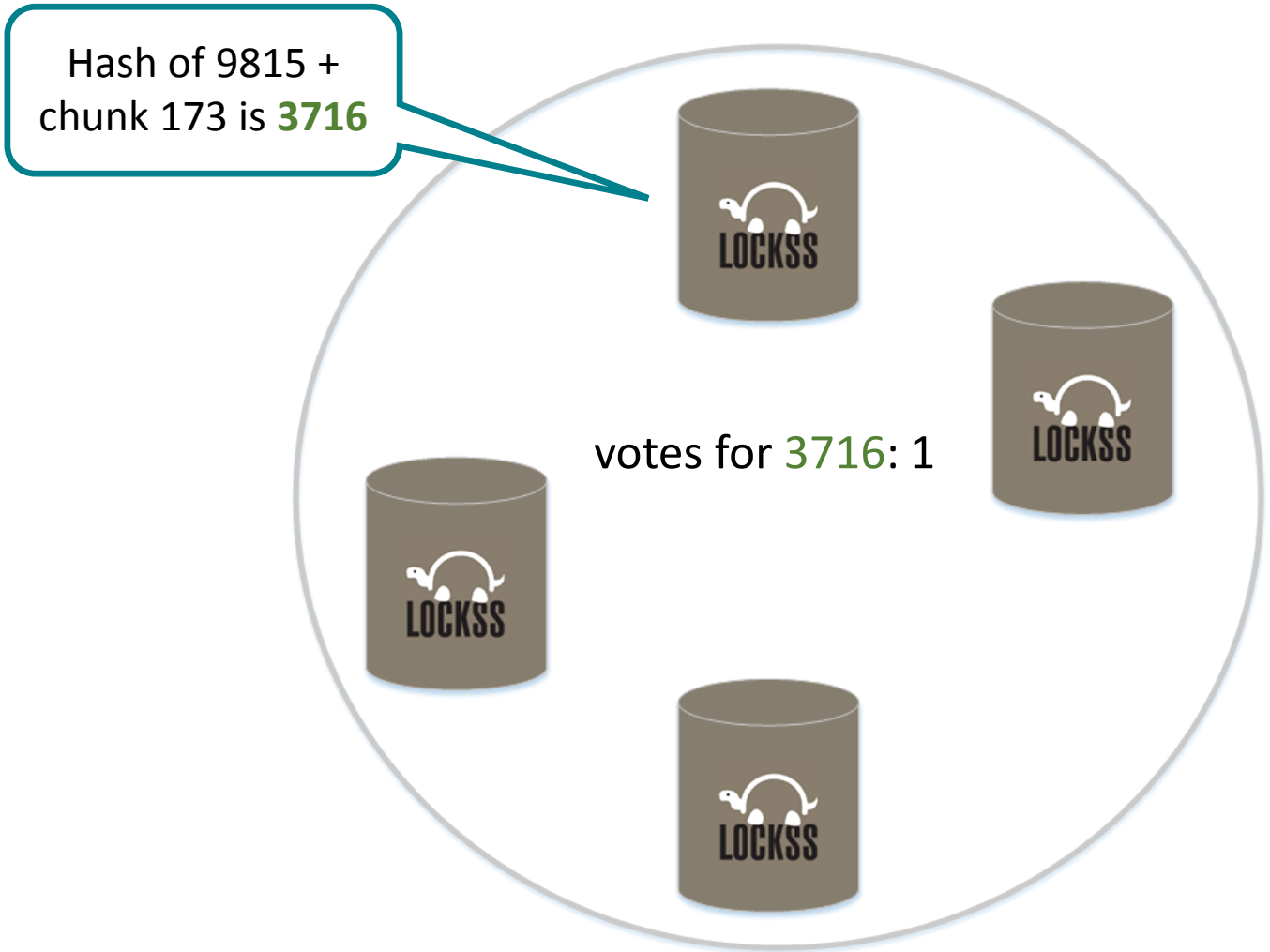
initiate poll w/ nonce



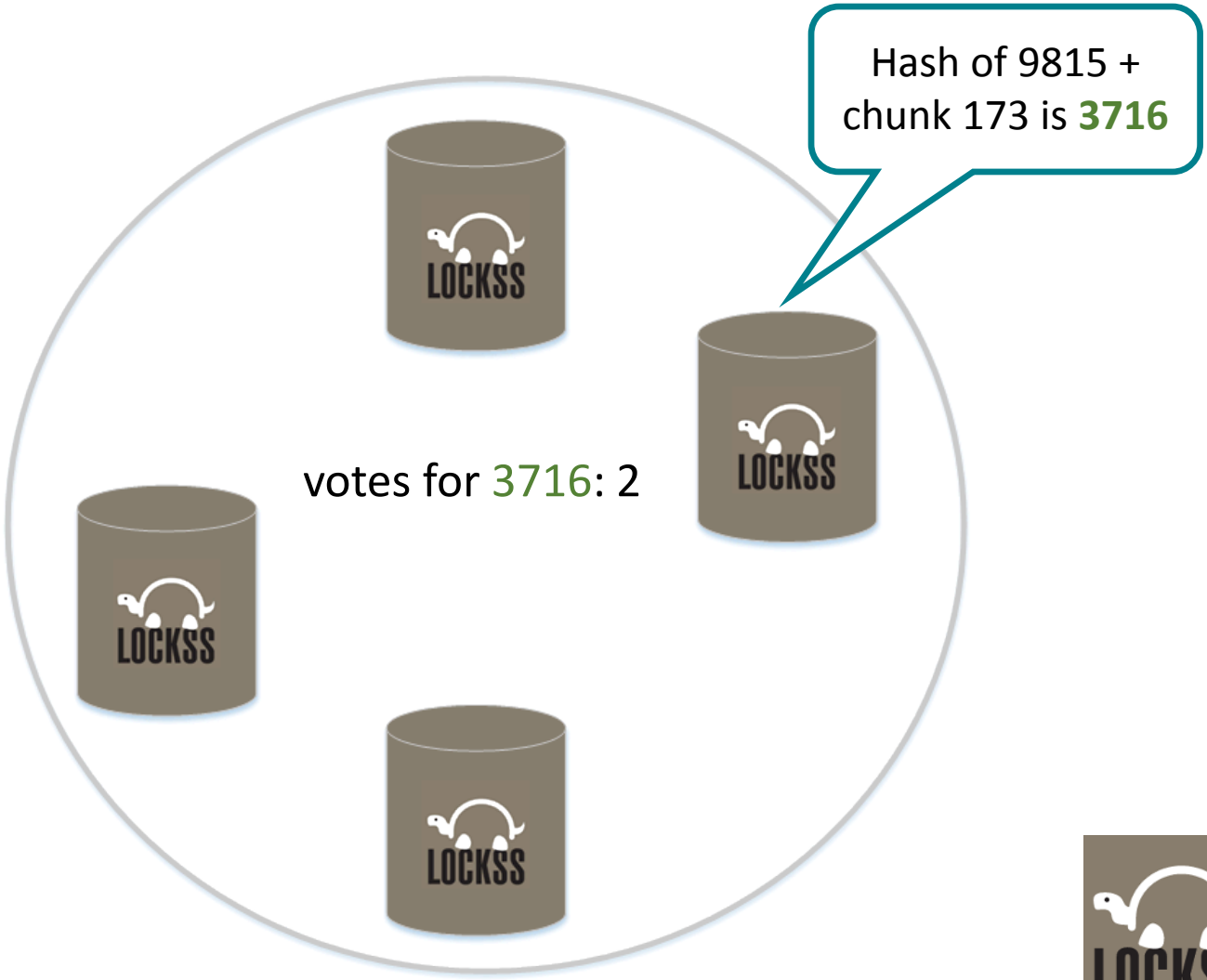
hashing



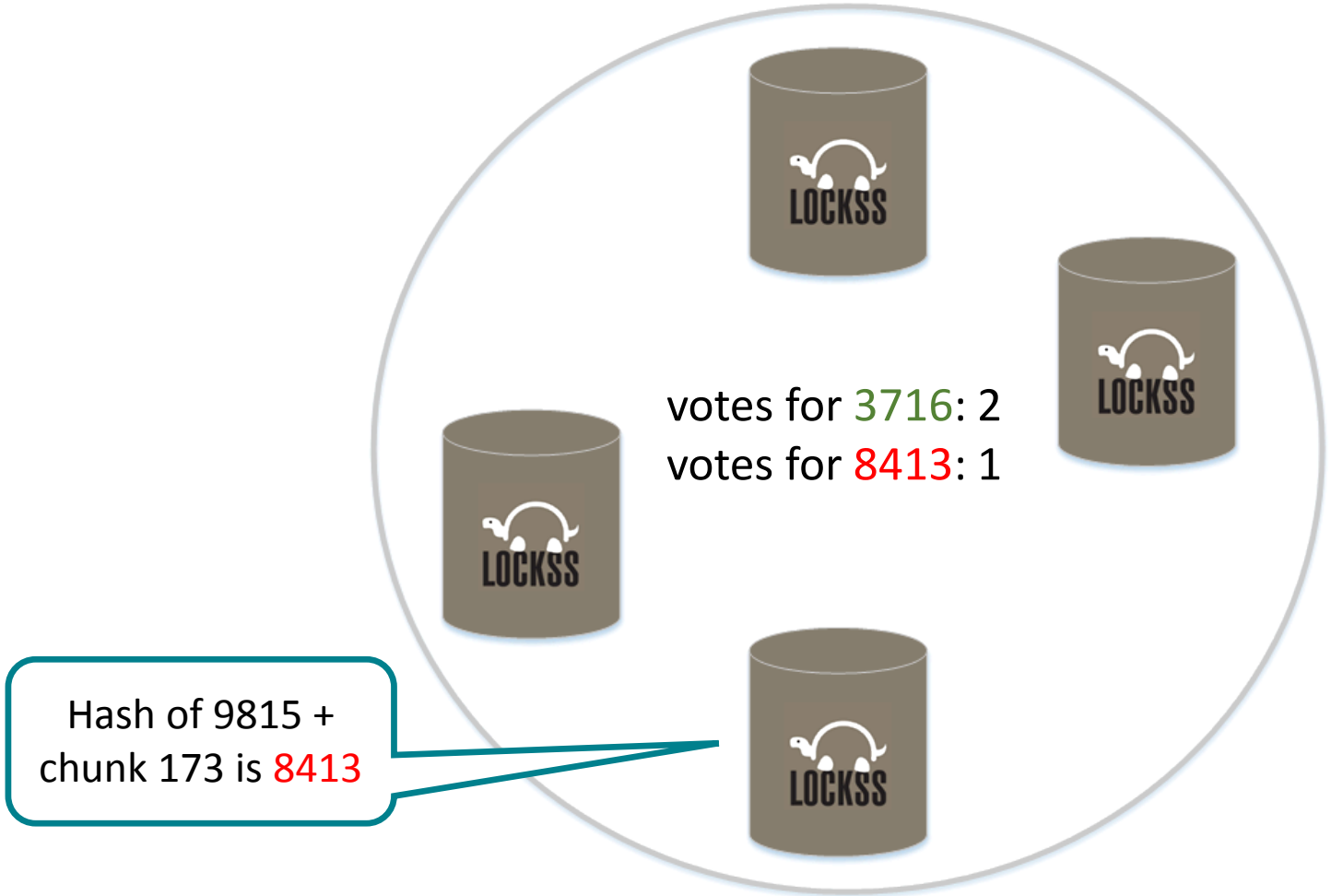
nodes respond to poll



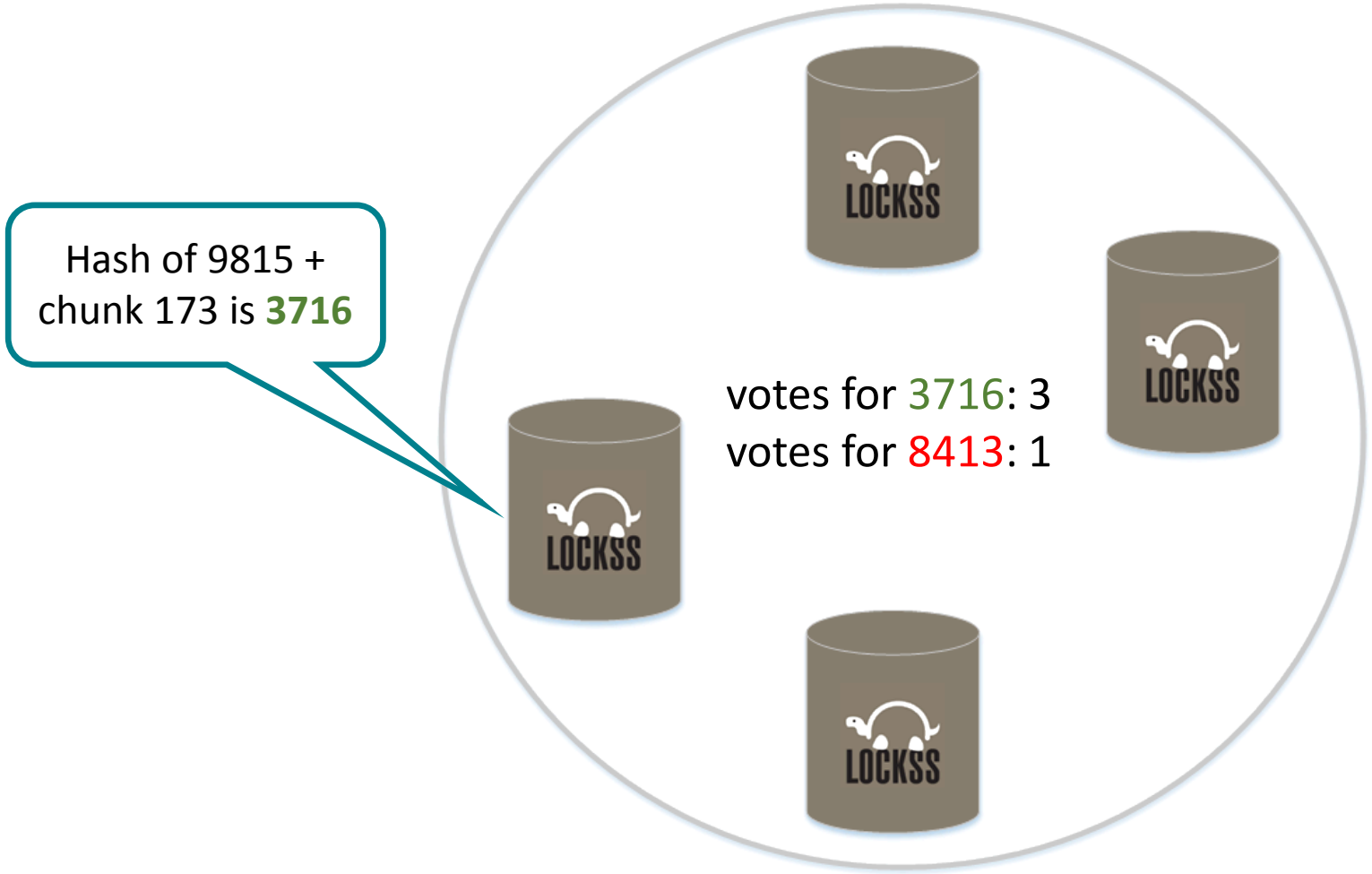
nodes respond to poll



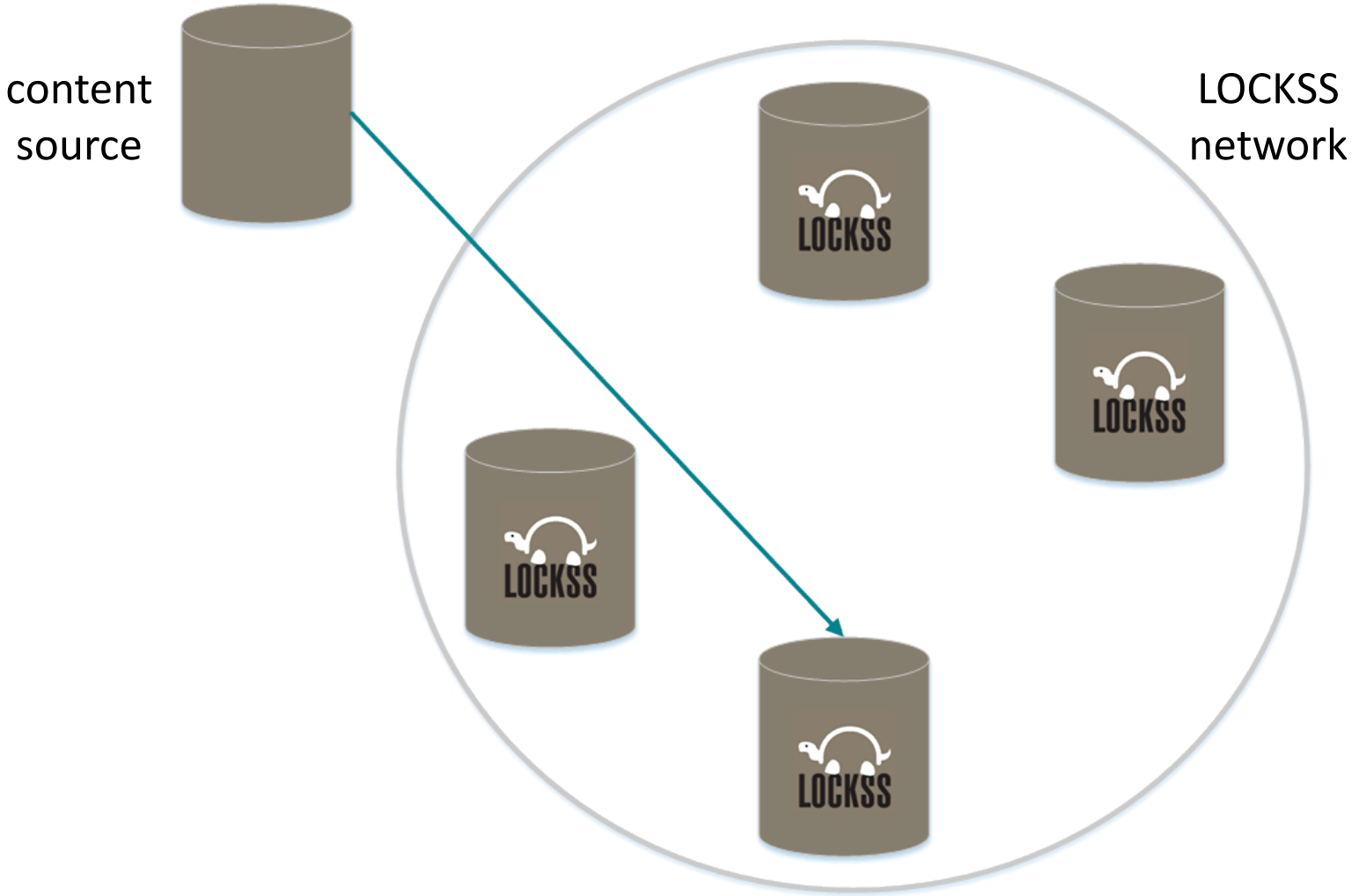
nodes respond to poll



initiate poll w/ nonce



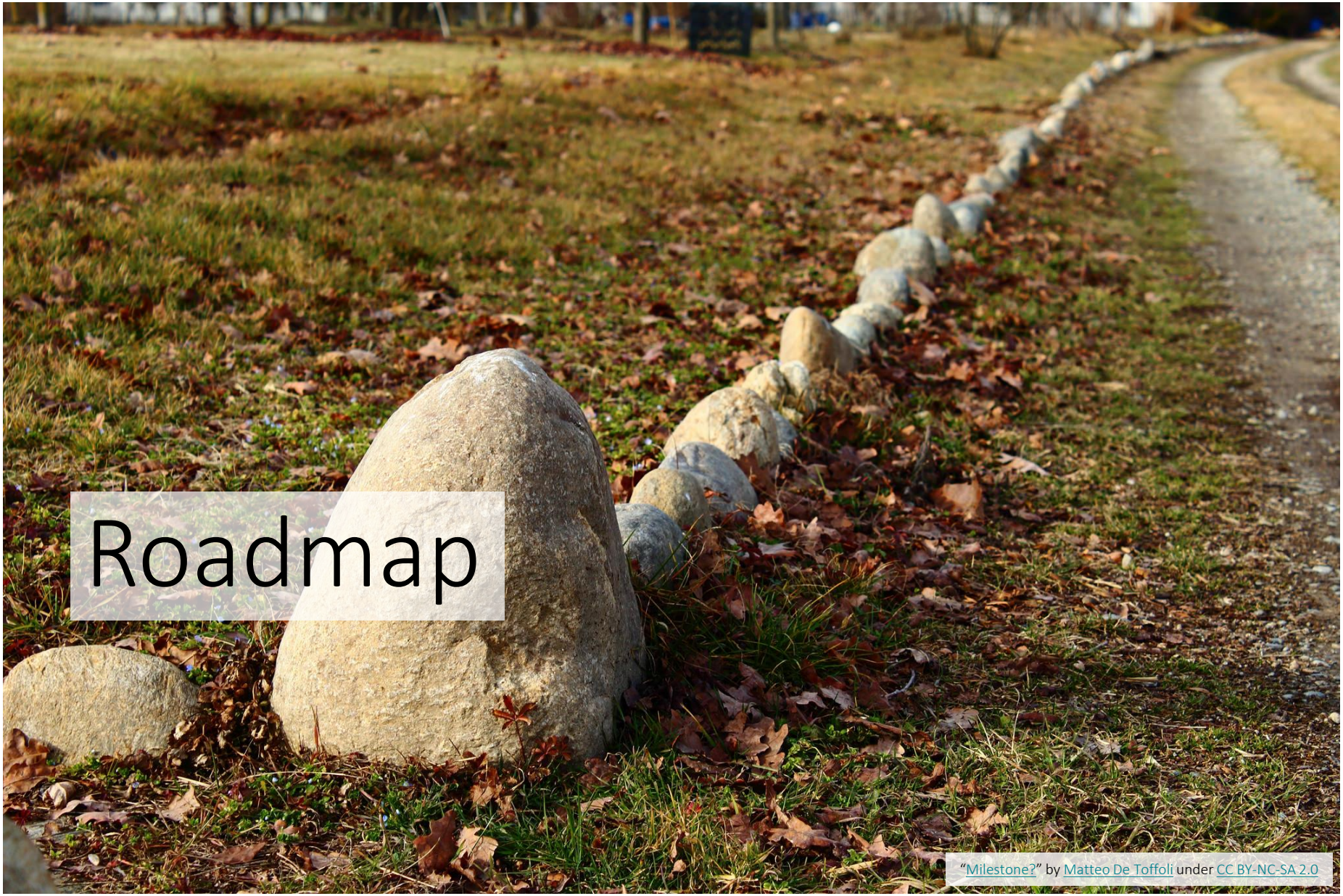
repair from content source



use cases for audit + repair

- other distributed digital preservation networks
- repository storage replication layers
- would like to support varied back-ends: tape, cloud, etc.

LOCKSS



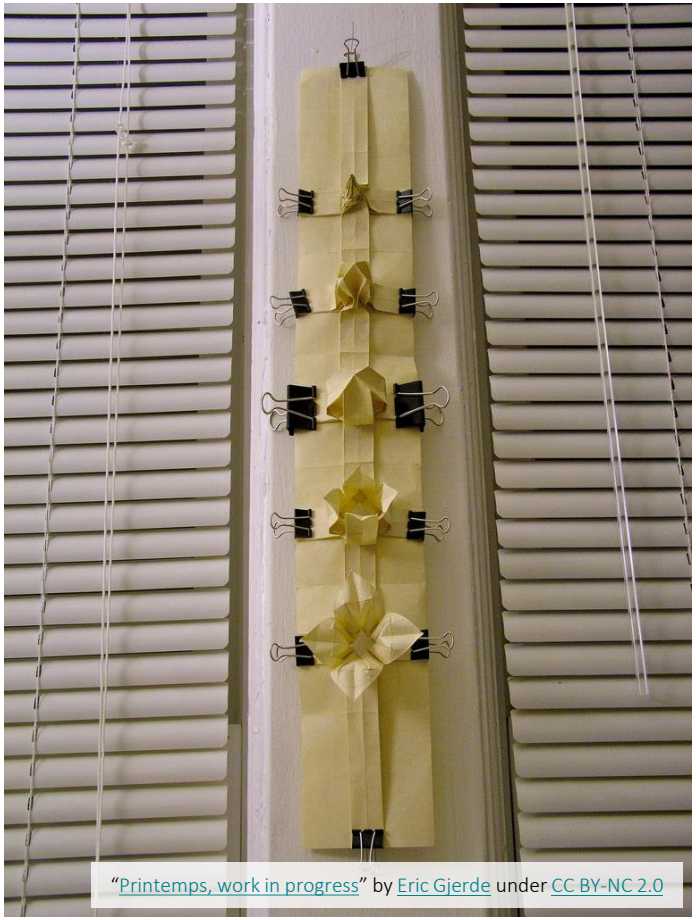
development progress

- access WARC-stored content via:
  - DOI
  - OpenURL
  - Memento (URL)
  - Solr full-text search
- web services:
  - metadata extraction
  - metadata query

LOCKSS

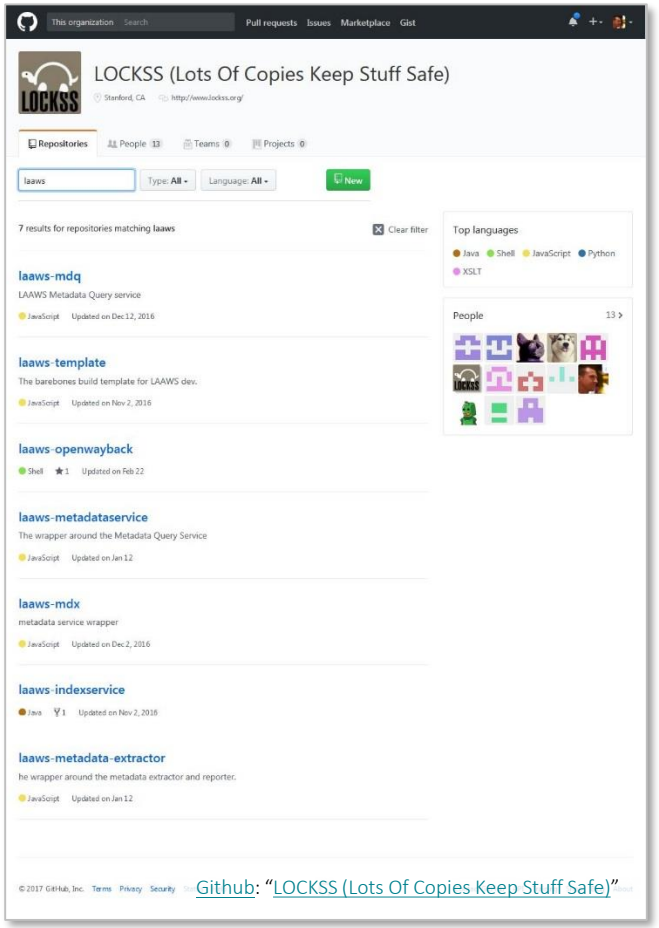
looking ahead

- by end of 2017
  - Docker-ize components
  - web harvest framework
  - polling + repair web service
- by end of 2018
  - IP address + Shibboleth access via OpenWayback
  - OpenWayback on-access format migration
  - full-text search web service



follow + plug in

- development (periodically) being pushed to Github
- moving toward more community-oriented software development
  - announcement of work cycles
  - sprint closeout reports + demos
  - community engagement



Github: "LOCKSS (Lots Of Copies Keep Stuff Safe)"

questions for you

- **what potential do you see** for LOCKSS technologies for web archiving, other use cases?
- what **standards or technologies could we use** that we maybe haven't considered?
- how could we help you to **use LOCKSS technologies**?
- how would you like to **see LOCKSS plug in more** to the web archiving community?