MetaArchive Architecture

Monika Mevenkamp
Emory University

10/29/08

Tasks in Preservation Systems

Ingest get content

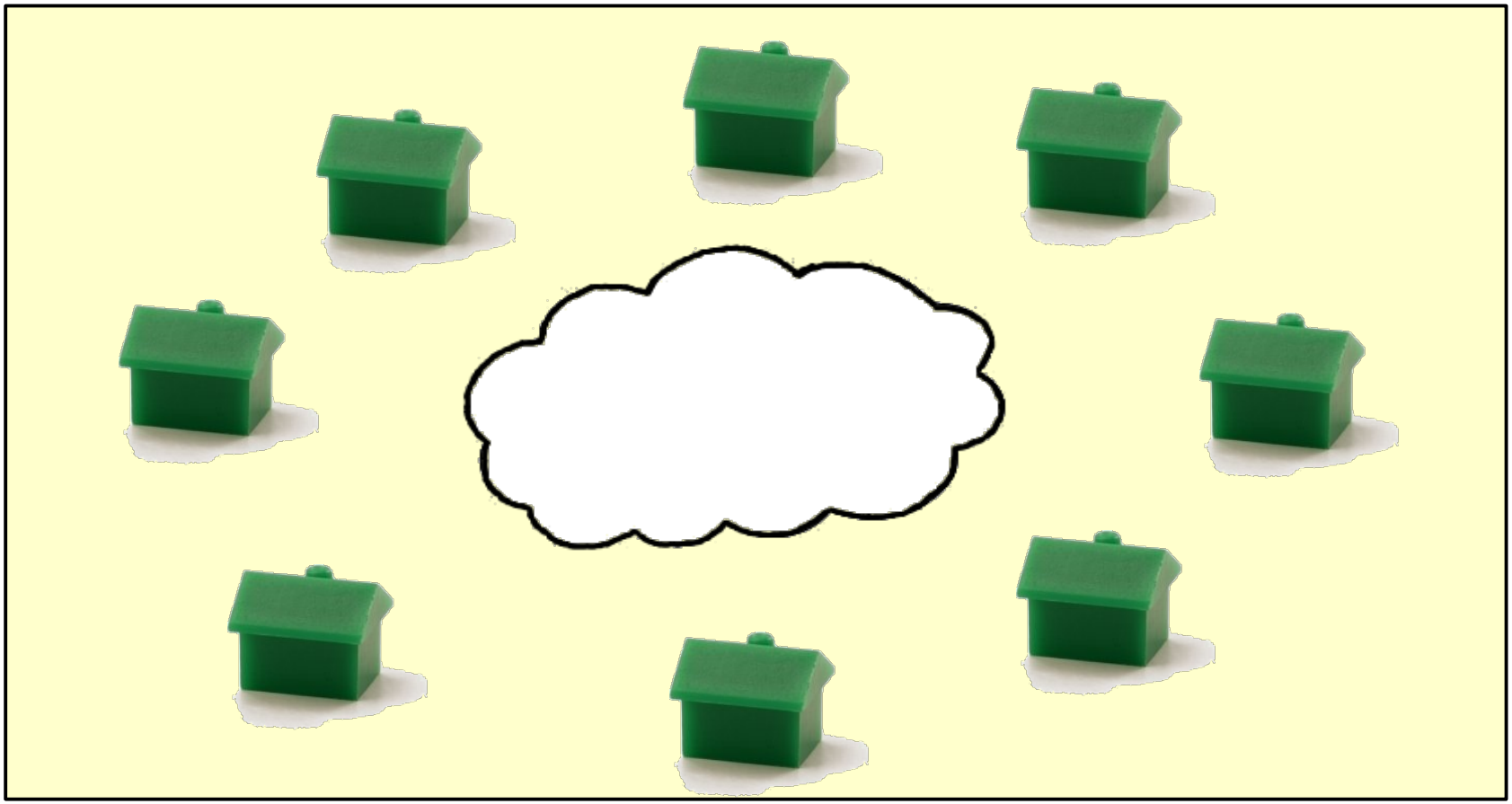
Preserve keep it safe

Update keep it up to date

ongoing

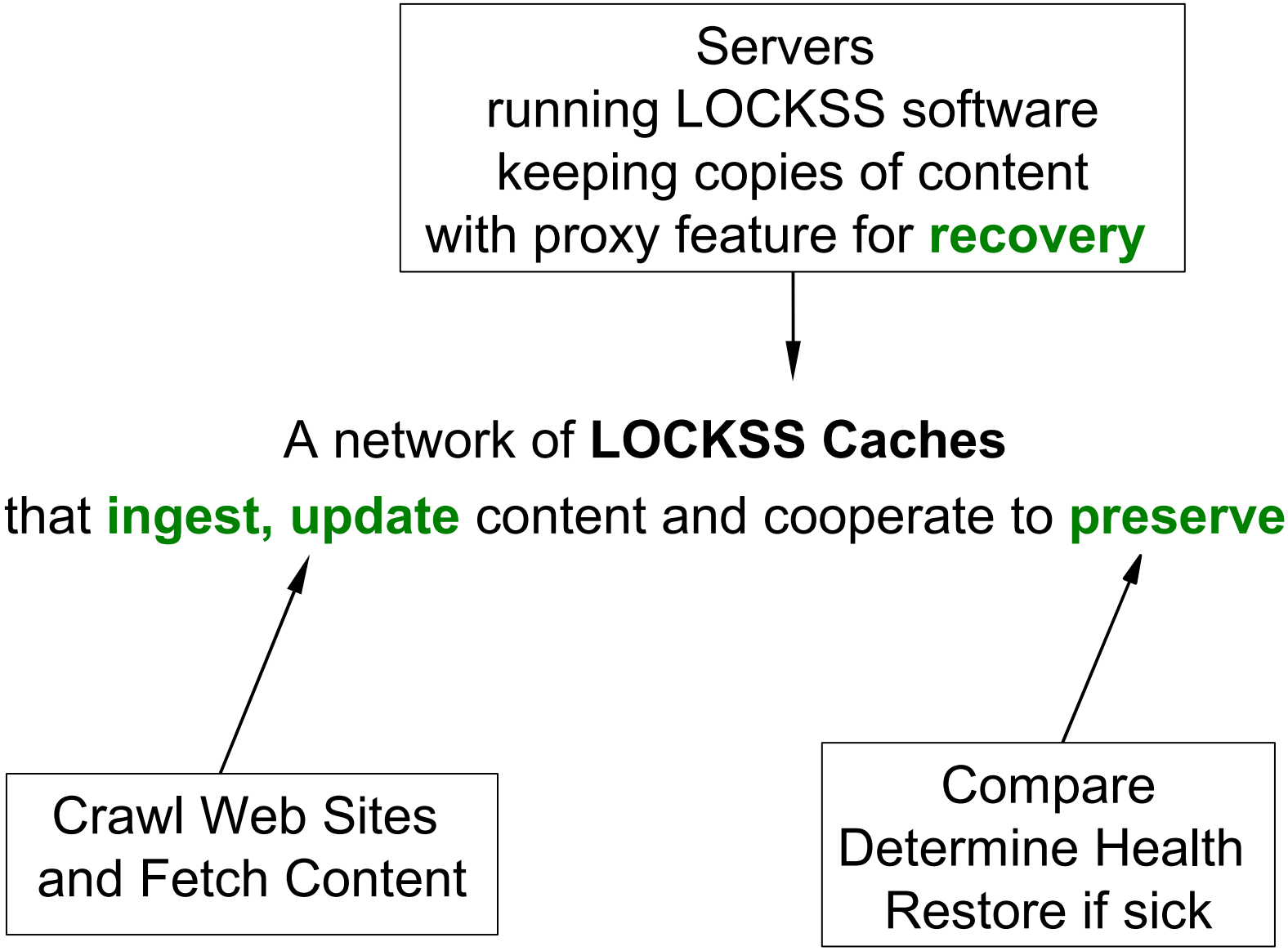
Recover when the data disaster hits

MetaArchive Private LOCKSS Network



A network of **LOCKSS Caches**

that **ingest, update** content and cooperate to **preserve**.


Servers
running LOCKSS software
keeping copies of content
with proxy feature for recovery
A network of LOCKSS Caches
that ingest, update content and cooperate to preserve.
Crawl Web Sites
and Fetch Content
Compare
Determine Health
Restore if sick

MetaArchive Private LOCKSS Network

LOCKSS daemon on each cache

Java software – could be anywhere

we run on security enhanced UNIX servers

need enough disk space to store content

ingest/update content through crawling web sites

easy to make content available on web site

replication by activating preservation of specific content

on individual caches (we do 6)

content recovery through proxy, copy from disk

communicate with each other through the Internet

we do encrypted messages using trusted certificates ==> simple to add caches

MetaArchive Network Overview

Conspectus Tool

Collection Definition
- Harvesting Info
  - Plugin Names &
  - Base URLs &
  - more? Param Values
- Meta Data
  - Publisher
  - Description

Used By
Content Providers

PLN Parameters

Title Database
Plugin Repository URLs
KeyStore for Plugins

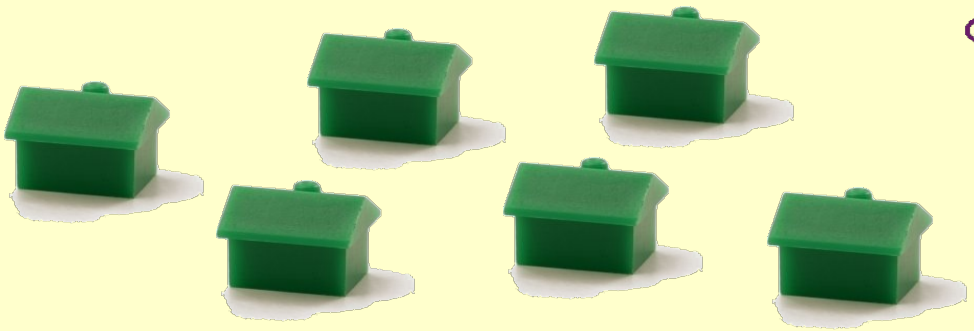
Maintained By
MetaArchive Staff

Vt Plugins Signed Jar Fles

Emory Plugins Signed Jar Fles

GaTech Plugins Signed Jar Fles

Maintained By
Plugin Developers



MetaArchive Caches Running LOCKSS Daemons

Provider site
Provider site
Provider site
Provider site
Provider site
Provider site

MetaArchive/LOCKSS Cache

Computer with big Disk

Running
    LOCKSS daemon
On
    Security Enhanced LINUX
....

MetaArchive's Conspectus Tool

Web Based Tool
Editor for Collection Data
    Title, Description, Publisher, Institution,
...
    Southern Digital Culture Archive
    ETD Archive
    Risk Rank
    Plugin Name
    Base_URL
    optional extra parameters
Generates archival unit definitions for
LOCKSS Title Database

php based -> ruby based

Title Database

Central XML Parameter File

Defines

- where to find plugins, keystore
- archival units
- trusted cache IPs
- LOCKSS UI users
- ...

Archival Unit

An Archival Unit is defined by
- Its plugin
- Its base_url
- The values of optional additional parameters

Each Archival Unit is
- maintained as a unit (voted, crawled, restored)
- saved in own directory on a LOCKSS cache's disk

After ingestion
- Definition can not change but Contents can

After getting to know an archival unit
LOCKSS daemons will forget

Plugin

XML File
defines Filtering Rules
used by Web Crawler component of LOCKSS daemons

defines which parts of web sites are fetched

what your plugin fetches is what you preserve

keystore

Only plugins from jar files that are signed with a certificate from the keystore are trusted.

Only trusted plugins are used.

Provider Site

The website where content is available .

LOCKSS daemons

periodically crawl these sites to fetch content

A plugin's recrawl interval

determines the frequency of visits.

Sites have to be accessible to all daemons.

Open firewalls !

MetaArchive Network Overview

Conspectus Tool

Collection Definition
- Harvesting Info
  - Plugin Names &
  - Base URLs &
  - more? Param Values
- Meta Data
  - Publisher
  - Description

Used By
Content Owners

PLN Parameters

Title Database
Plugin Repository URLs
KeyStore for Plugins

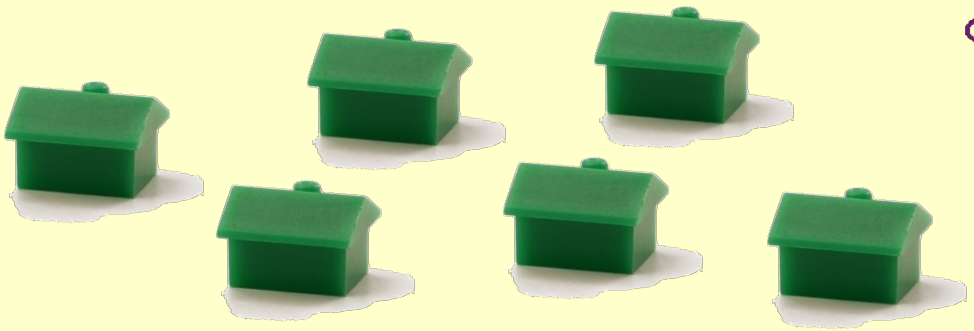
Maintained By
MetaArchive Staff

Vt Plugins Signed Jar Fles

Emory Plugins Signed Jar Fles

GaTech Plugins Signed Jar Fles

Maintained By
Plugin Developers



MetaArchive Caches Running LOCKSS Daemons

Provider site
Provider site
Provider site
Provider site
Provider site
Provider site

MetaArchive/LOCKSS Daemon

Title Database
plugin repository URLs
archival units
Keystore for Plugins

Plugin Repositories

Signed Jar Files

Signed Jar Files

provider site

provider site

provider site

provider site

initialize

do forever
crawl provider sites

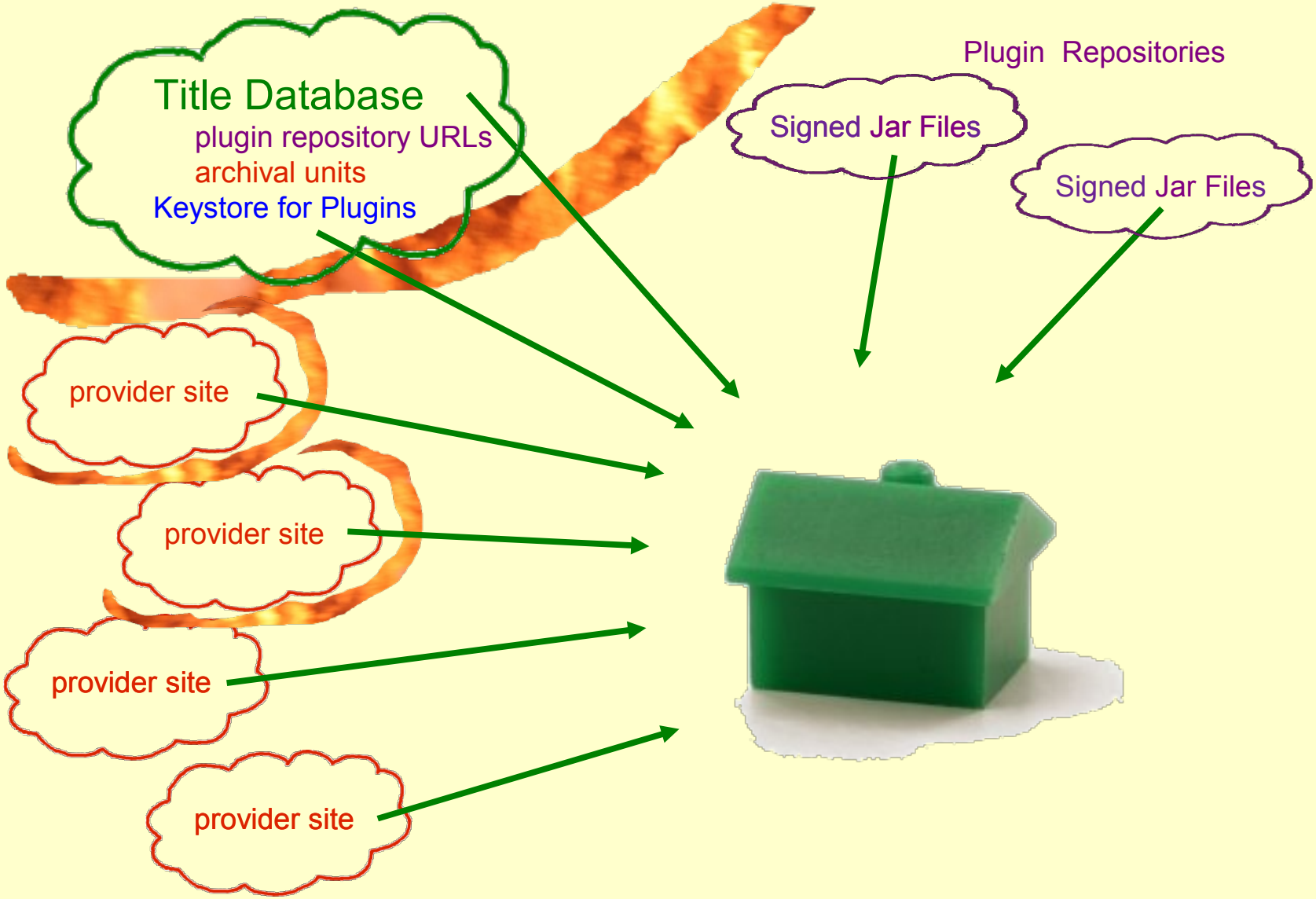
participate in votes about content state
initiate votes

repair broken content

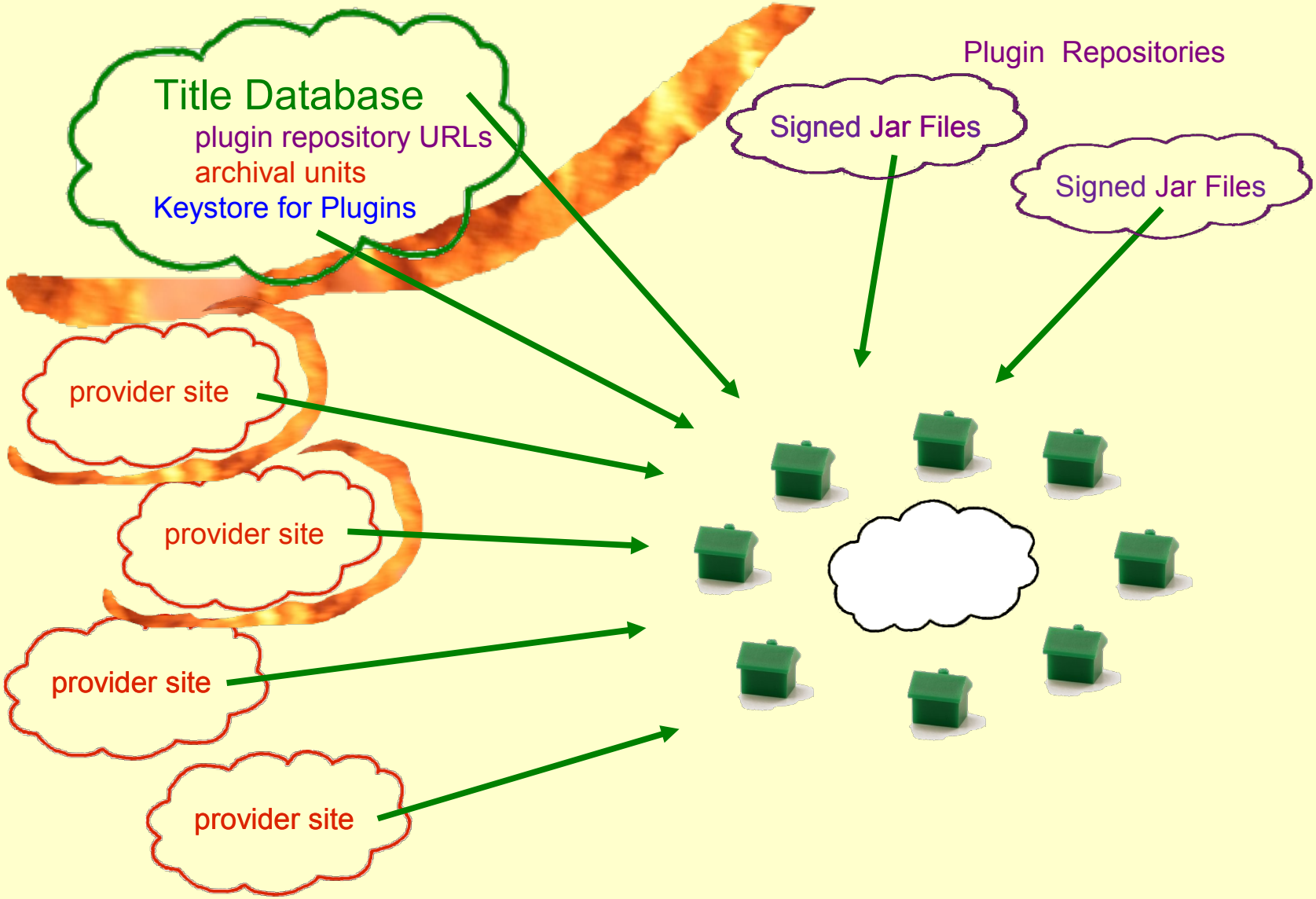
by re-crawling provider sites
or
by restoring from peer caches

reinitialize

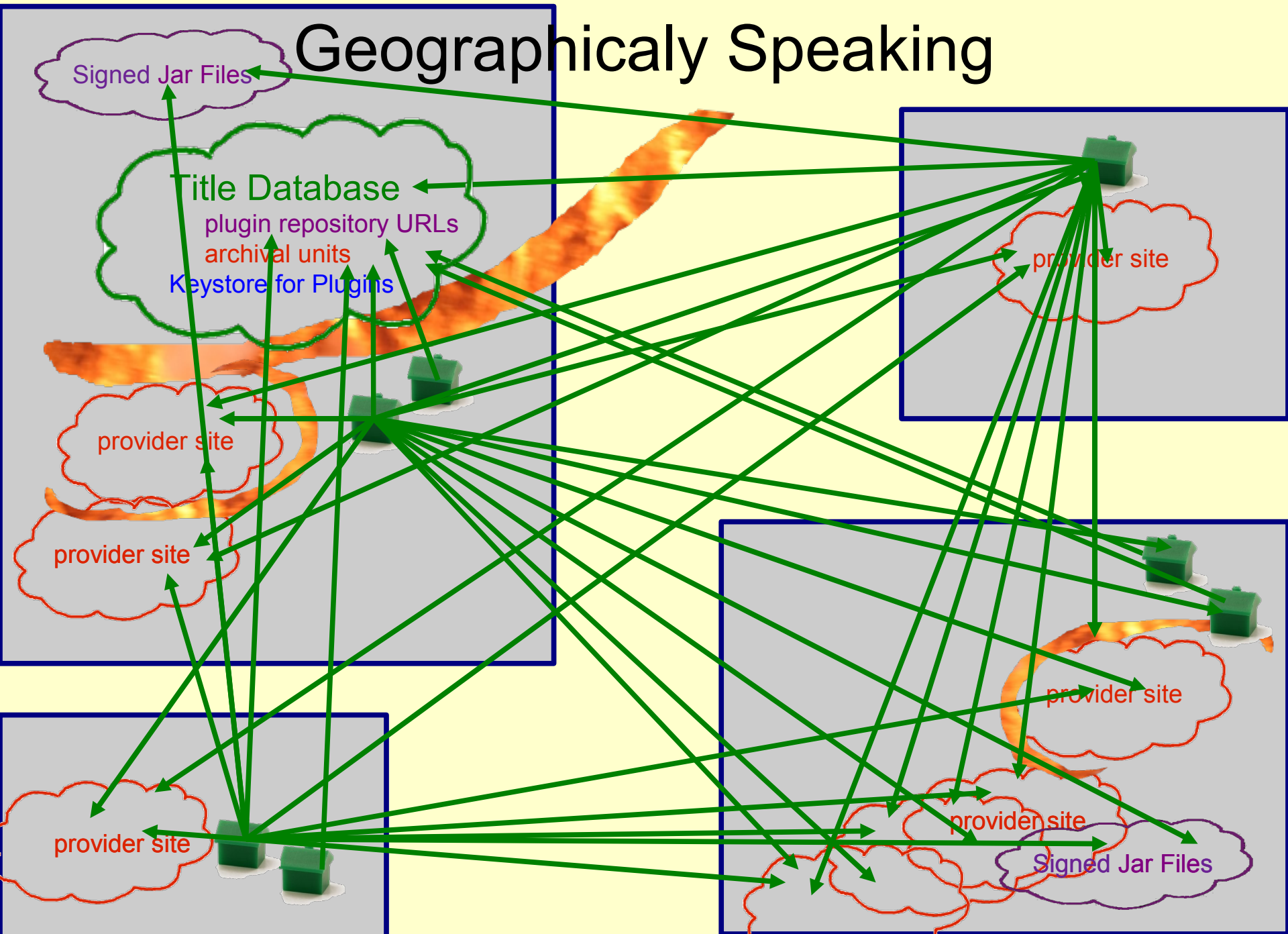
MetaArchive/LOCKSS Cache



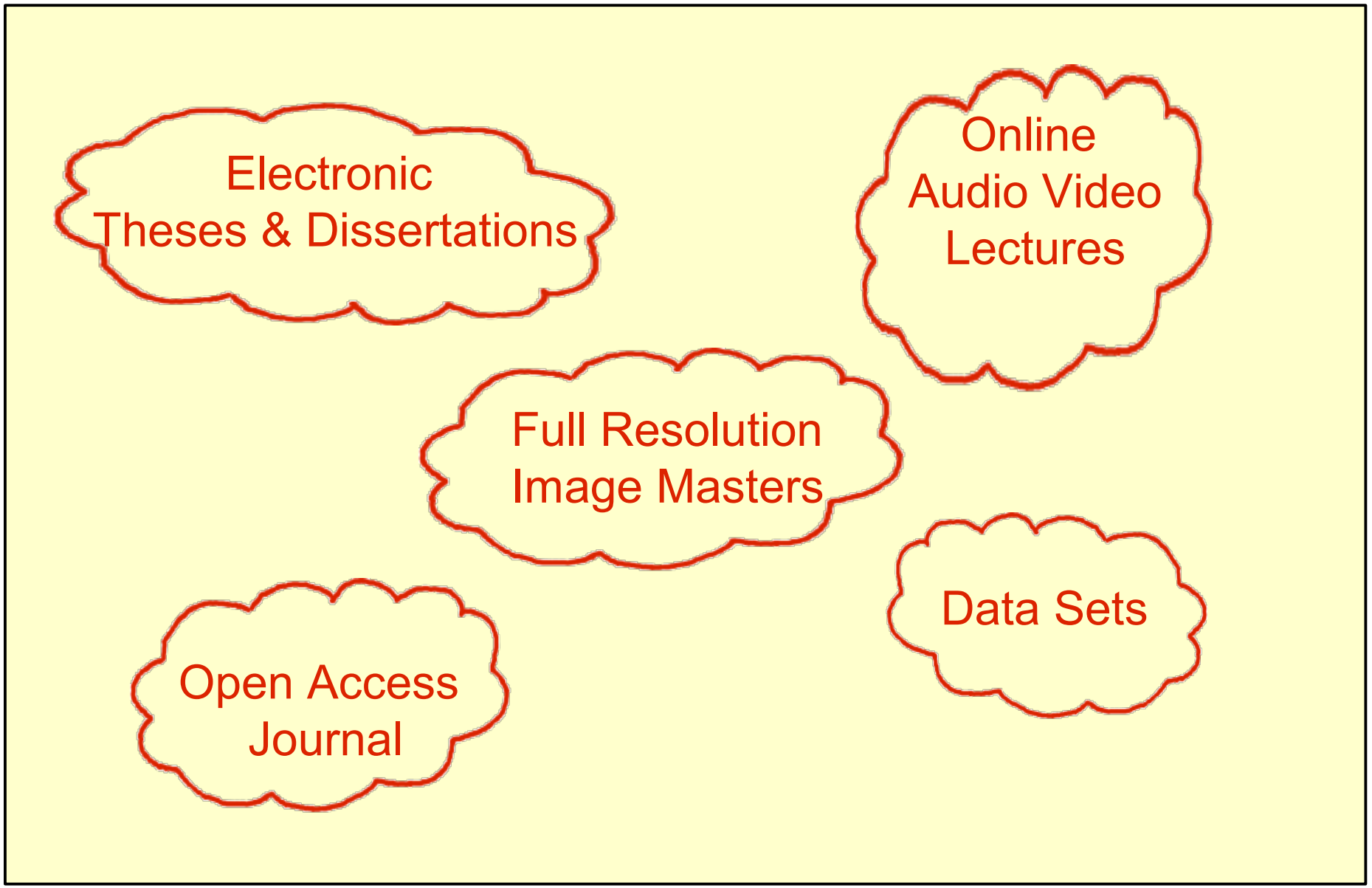
MetaArchive/LOCKSS Network



Geographicaly Speaking



Content → MetaArchive



Content → MetaArchive

http://somewhere.edu/someStuff

web locations with references to
content files
metadata about content
both in common formats

Small enough to fit in one archival unit

1GB <= Sum File Sizes <= 10GB

MetaArchive Network Overview

Conspectus Tool

Collection Definition
- Harvesting Info
  - Plugin Names &
  - Base URLs &
  - more? Param Values
- Meta Data
  - Publisher
  - Description

Used By
Content Providers

PLN Parameters

Title Database
Plugin Repository URLs
KeyStore for Plugins

Maintained By
MetaArchive Staff

Vt Plugins Signed Jar Fles

Emory Plugins Signed Jar Fles

GaTech Plugins Signed Jar Fles

Maintained By
Plugin Developers

MetaArchive Caches Running LOCKSS Daemons

Define Collection in Conspectus

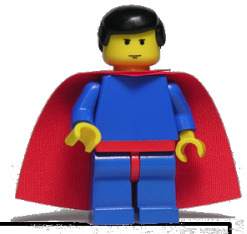
Title:	Talks By The Famous Guy
Description:	Lectures Given Since ....
Creator:	Famous Guy, Famous Co-Worker
Publisher:	Some Where University
Rights:	Famous Institute, Some Where University
Access Rights:	Unrestricted
Cataloged Status:	Cataloged (xml metadata file included)
URL available via:	http://somewhere.edu/someStuff
Format:	jpeg, wav, text, ...
Accrual:	adding every 3 month
Extent:	1500000

Harvesting Info /WebCrawl

Plugin:	edu.somewhere.allContent
Base Url:	http://somewhere.edu/someStuff

Create Plugin

edu.somewhere.allContent



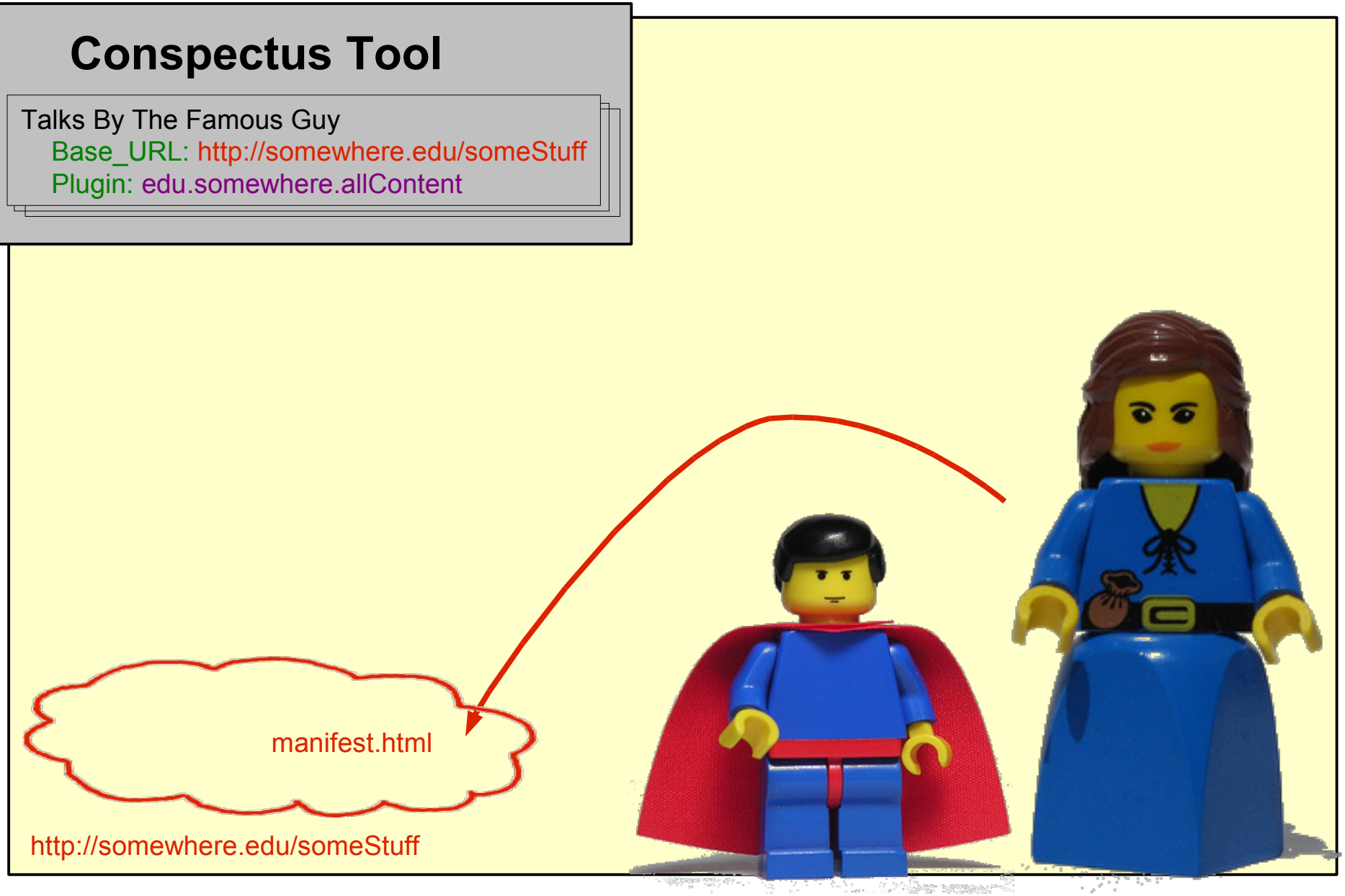
identifier/name	edu.someWhere.allContent
good to have Notes	This plugin fetches all content it encounters whose urls start with Base_URL. It is useful for small sites that are to be harvested completely as they are delivered from web servers. Database/software driven sites may want to include database dumps and software archives and link to them from the manifest page
must have Configuration Params	Base_URL
Start URL Template	Base_URL/manifest.html
Crawl Rules	Exclude No Match: "^Base_URL" Include: "^Base_URL/manifest.html$" Include: "^Base_URL$" Include: "^Base_URL/"

Based on metasource: org.metaarchive.example.allContent.xml

Create And Post Manifest Page

Conspectus Tool

Talks By The Famous Guy
Base_URL: http://somewhere.edu/someStuff
Plugin: edu.somewhere.allContent



Create Manifest Page

http://somewhere.edu/someStuff/manifest.html

good to have
link to conspectus entry

mail-contact

description

formats
which part
metadata

must have
crawl start-url

permission stmt

Talks By The Famous Guy LOCKSS Manifest Page

Collection Info:
* Conspectus Collection(s): Talks By The Famous Guy
* Institution: Famous Institute, Some Where University
* Contact Info: Lisa Krueger

This collections contains transcripts of lectures given by the famous guy starting with his phD defense in 1856 given at the Current Institute .... It contains plain text, scanned images, and pdf files.
The whole site is preserved.
... links to dublin core XML files are part of each lecture page.

Links for LOCKSS to start its crawl:
* index.html - the home page of the Famous Guy Web Site

LOCKSS system has permission to collect, preserve, and serve this Archival Unit.

Based on metasource: manifest_template from metasource

Tell the Network

Web Site	http://somewhere.edu/someStuff
Conspectus Entry	Talks By Famous Guy (Base_URL=http://somewhere.edu/someStuff)
Manifest Page	http://somewhere.edu/someStuff / manifest.html
Plugin	edu.someWhere.allContent

Sign and and Jar plugin and make it web accessible
  http:// emPluginServer /plugins/edu_somewhere_allcontent.jar
  http:// emPluginServer /plugins == plugin registries defined in the title database

Flag Talks By Famous Guy as ready for Harvest in the Conspectus tool

The title database is regenerated every 15 minutes from Conspectus Data
LOCKSS daemons reread the title database every 15 minutes
LOCKSS daemons reread plugin registries every six hours
==> All LOCKSS daemons get to know the new Talks By Famous Guy archival unit
    after at most 30min.

Visit the LOCKSS user interface of caches and add Talks By Famous Guy archival unit to its Configuration.

LOCKSS daemons/caches where Talks By Famous Guy archival unit was added start preserving it.

Deploy Plugin

Conspectus Tool

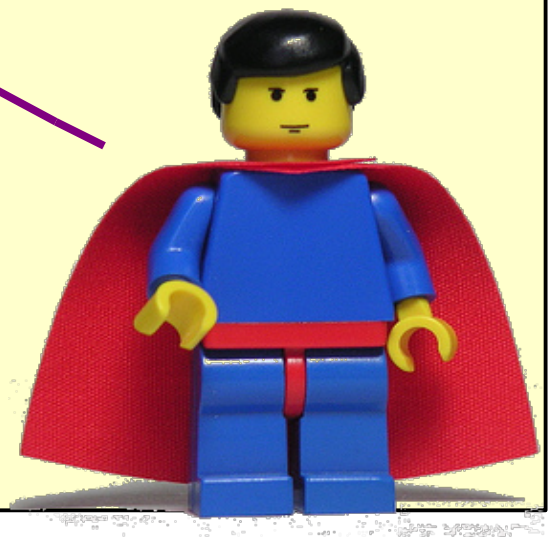
Talks By The Famous Guy
Base_URL: http://somewhere.edu/someStuff
Plugin: edu.somewhere.allContent

Plugin Repositories

Signed Jar Files

edu_somewhere_allcontent.jar
edu_xyz_other.jar
....

http://somewhere.edu/someStuff
manifest.html



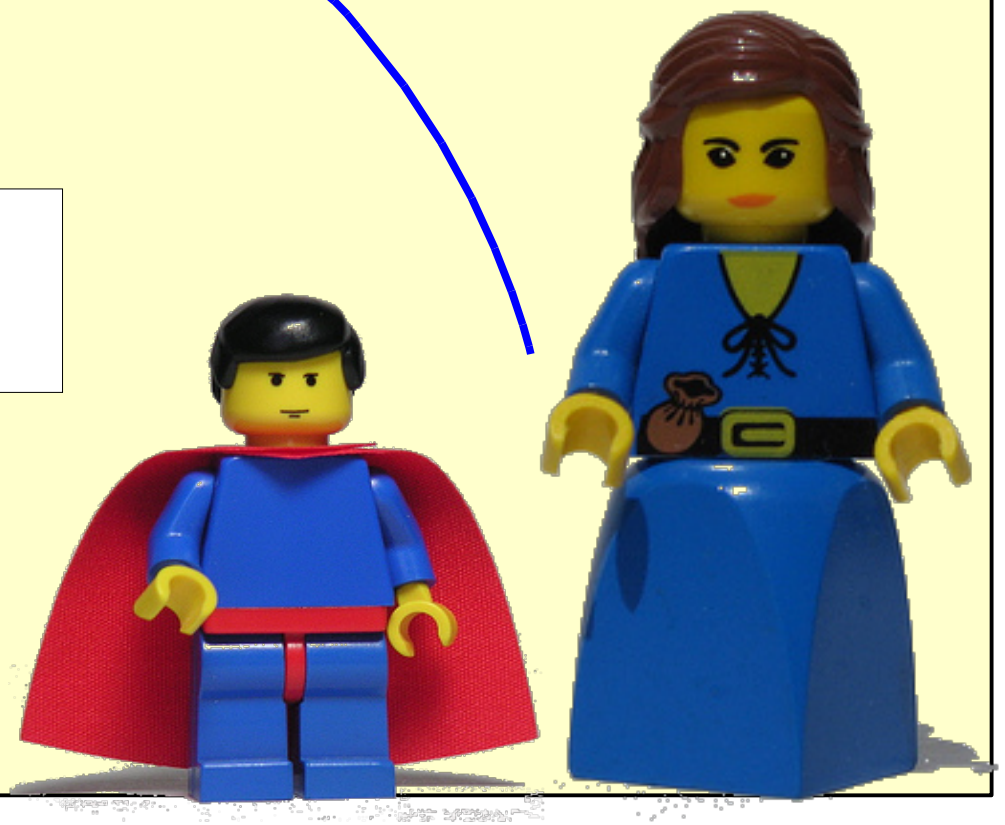
Collection is READY for Harvest

Conspectus Tool

Talks By The Famous Guy (READY)
Base_URL: http://somewhere.edu/someStuff
Plugin: edu.somewhere.allContent

web site is ready
plugin signed, jared, and deployed
READY for Preservation

http://somewhere.edu/someStuff
manifest.html



READY Go !

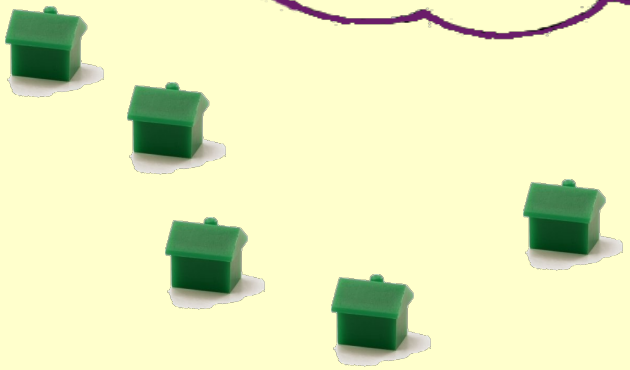
Conspectus Tool

Talks By The Famous Guy (READY)
Base_URL: http://somewhere.edu/someStuff
Plugin: edu.somewhere.allContent

Plugin Repositories

Signed Jar Files

edu_somewhere_allcontent.jar
edu_xyz_other.jar
....



Title Database: lockss.xml
Keystore for Plugins

http://somewhere.edu/someStuff
manifest.html

Caches Reload Plugins

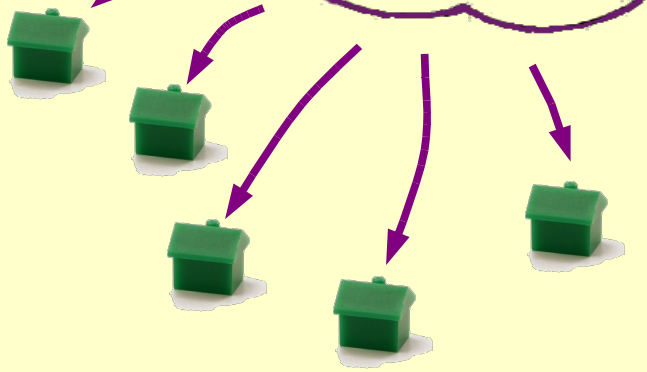
Conspectus Tool

Talks By The Famous Guy (READY)
Base_URL: http://somewhere.edu/someStuff
Plugin: edu.somewhere.allContent

Plugin Repositories

Signed Jar Files

edu_somewhere_allcontent.jar
edu_xyz_other.jar
....

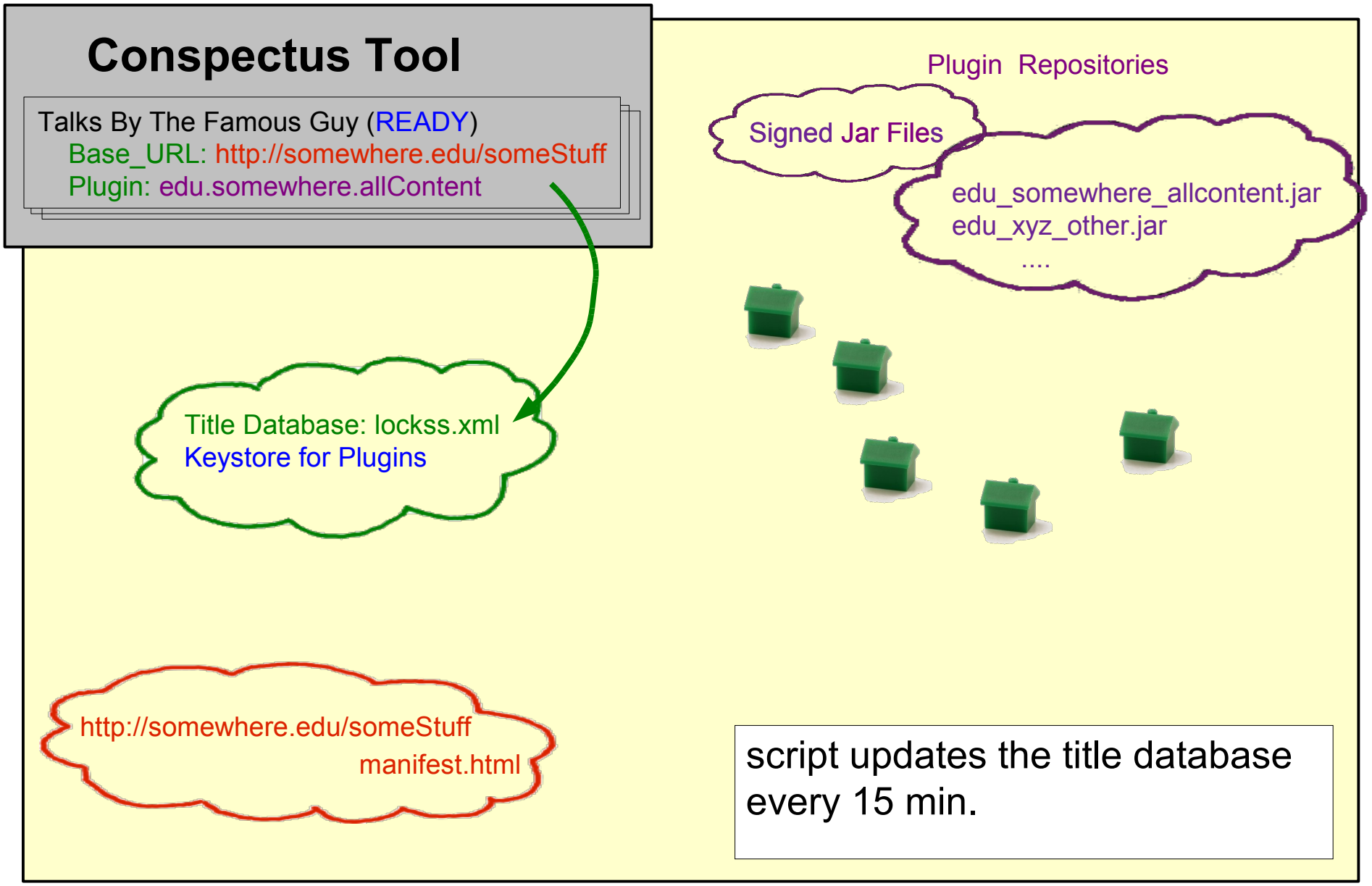


Title Database: lockss.xml
Keystore for Plugins

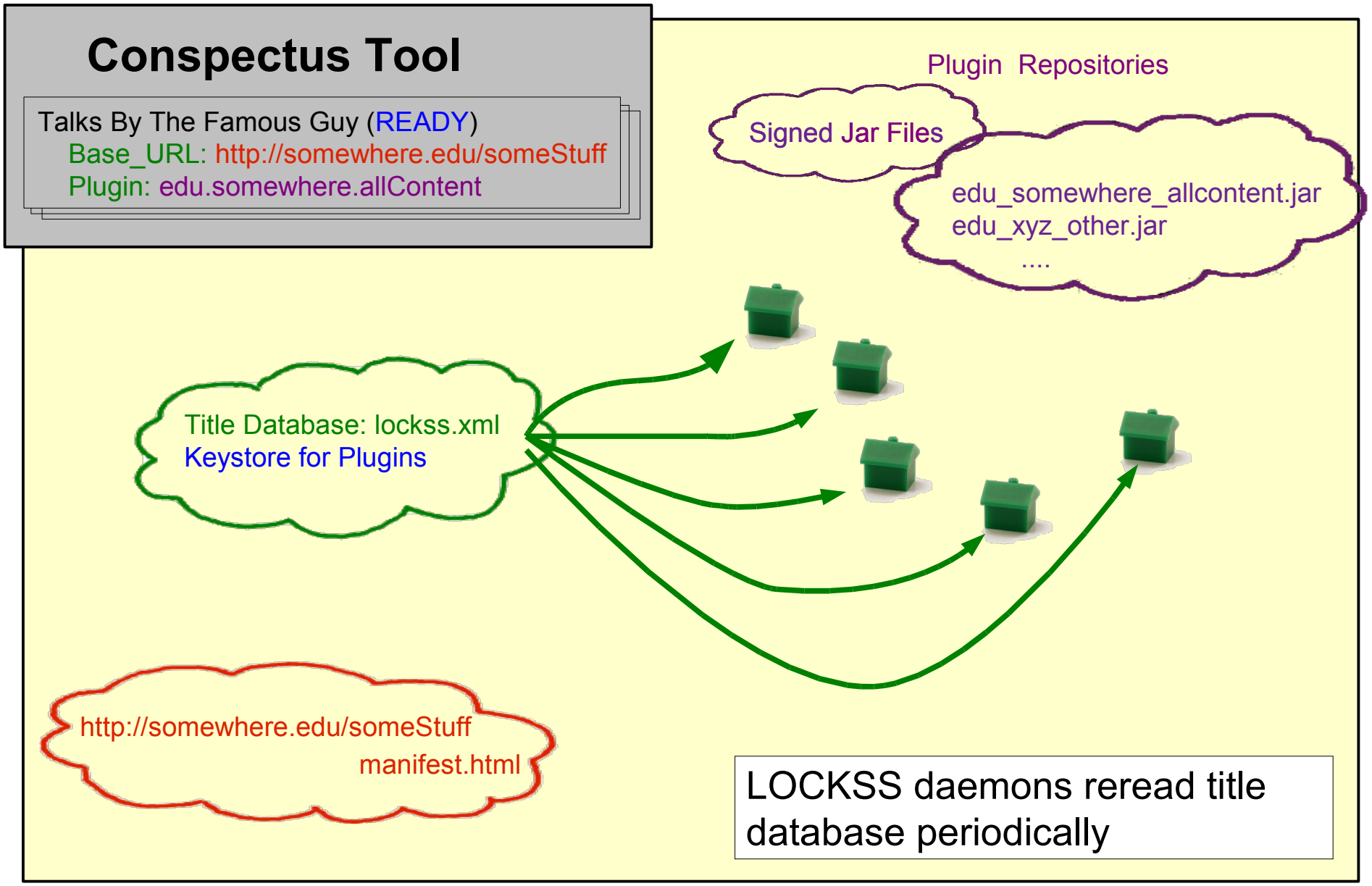
http://somewhere.edu/someStuff
manifest.html

LOCKSS daemons reread plugins every 6 hours

Update Lockss.xml



LOCKSS Daemons Reread lockss.xml



Human Adds To Daemon Configuration

Conspectus Tool

Talks By The Famous Guy (READY)
Base_URL: http://somewhere.edu/someStuff
Plugin: edu.somewhere.allContent

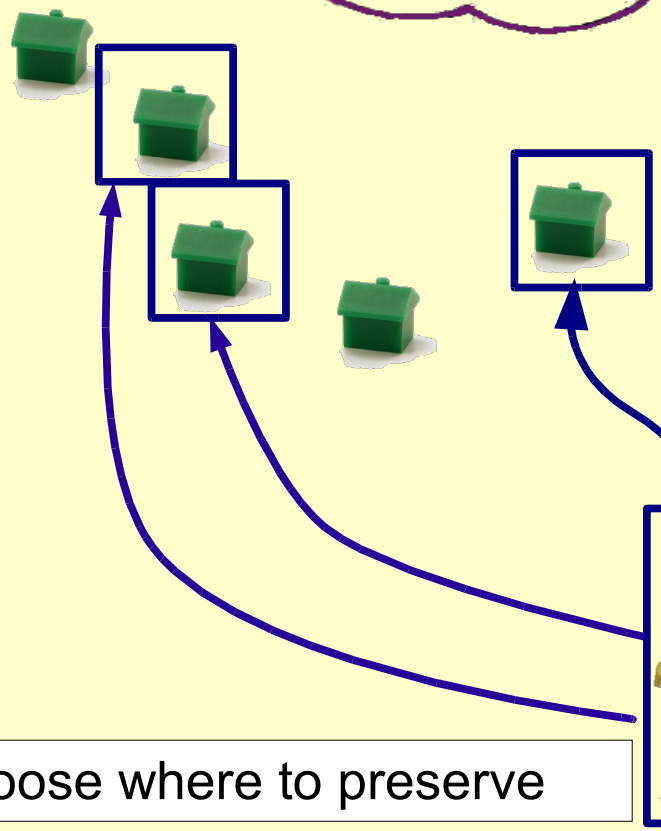
Plugin Repositories

Signed Jar Files

edu_somewhere_allcontent.jar
edu_xyz_other.jar
....

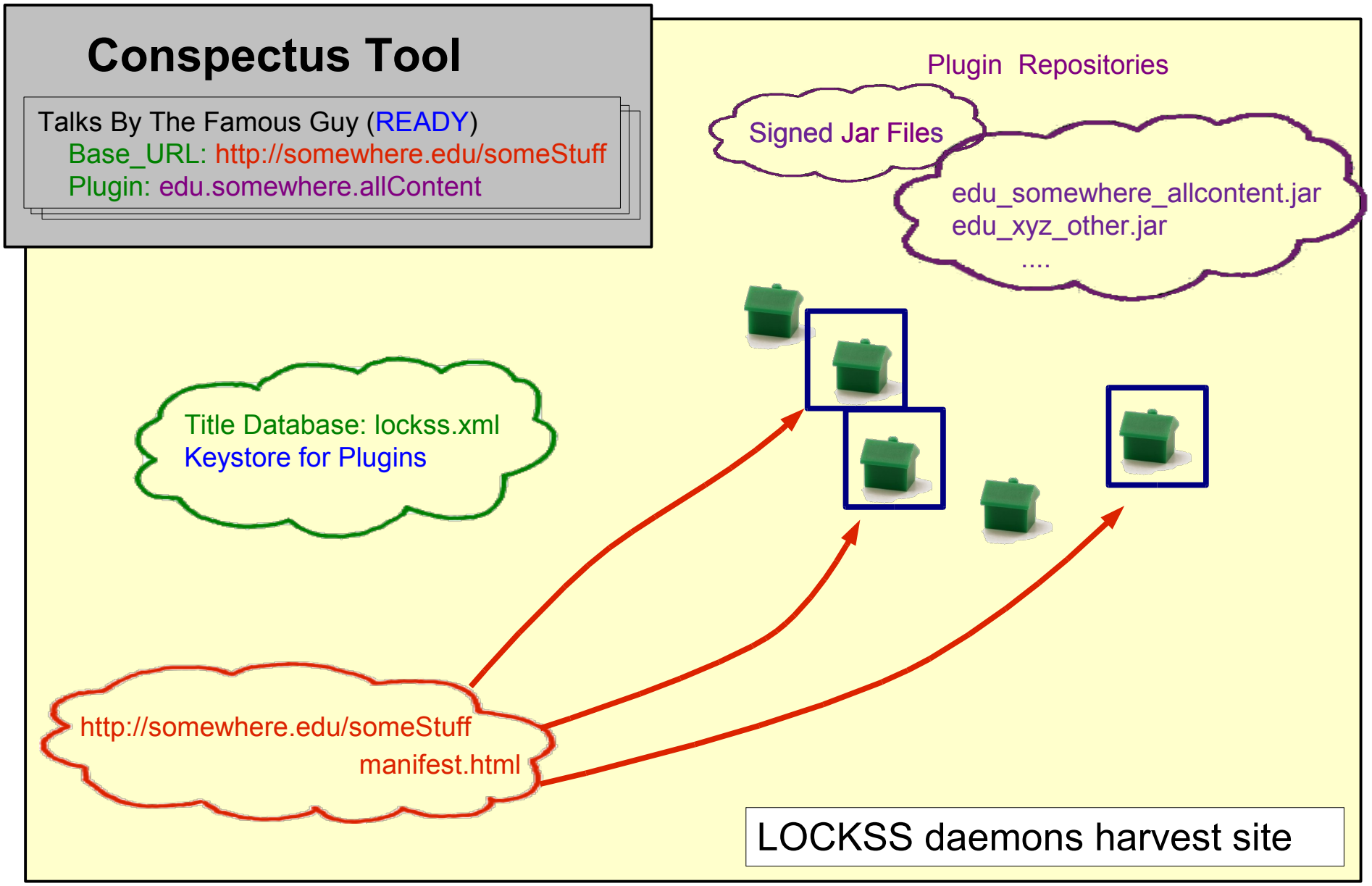
Title Database: lockss.xml
Keystore for Plugins

http://somewhere.edu/someStuff
manifest.html



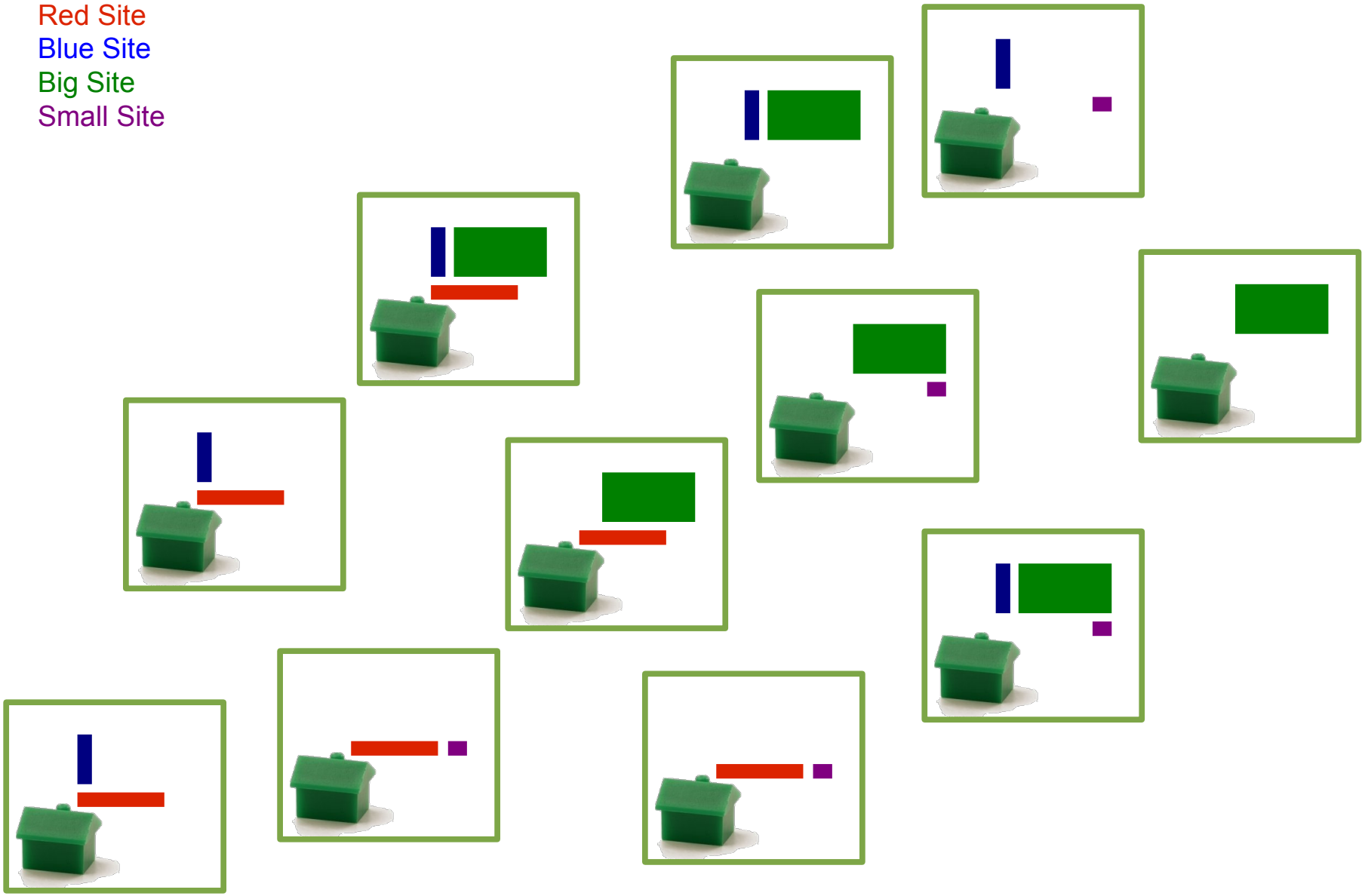
choose where to preserve

Daemons Harvest Content



6 Replications

Red Site
Blue Site
Big Site
Small Site



Plugins == Preservation Filters

LOCKSS daemons crawl by
- starting at manifest page
- parsing html pages to collect links
- fetching from filtered URLs

What You Fetch is What You Preserve

What You See is What You Preserve

Plain html based web site

web hosted directory structure

but web server can get into the way

Without extra effort:
you may end up with html only

What You See Is What You Preserve

SERVER has		CLIENT/CACHE sees
html files / docs	→	html files / docs
css files	→	css files
server side includes	→	expanded files
cgi processing	→	whatever cgi-scripts generate
database + code	→	whatever code generates
xml + xsl	→	xml
	→	transformed xml

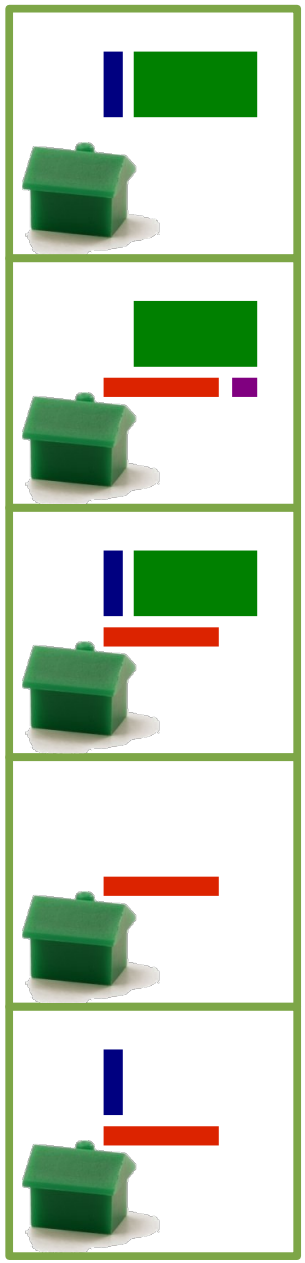
Does The Plugin Behave Correctly ?

Dry Run the Recovery

Use LOCKSS daemon's Audit Proxy

Check content in daemon.

If Content Site Changes Revisit Plugin



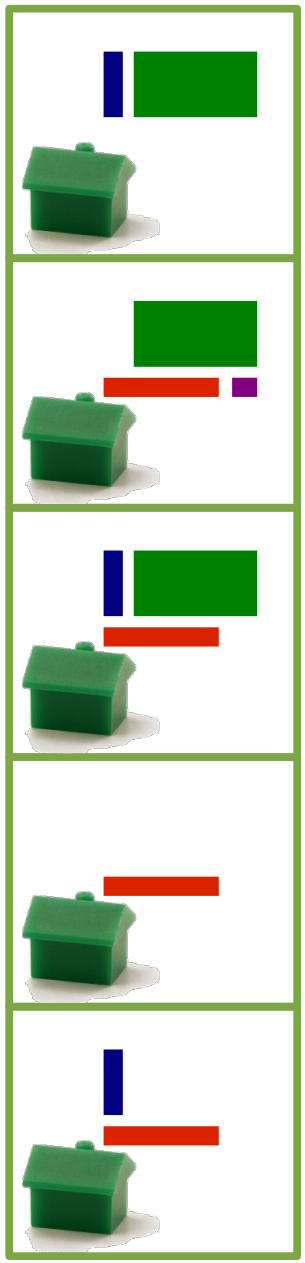
Network Monitoring

LOCKSS user interface

View status of particular cache in detail

Cache Manager

Look across network



Cache Manager

web based tool (Ruby)
Co-development with LOCKSS team

queries LOCKSS daemons on caches
stores info in database

produces lists of
- where content is replicated
- content size
- disk usage

tool flags
- troubled crawls
- archival units with problems

How safe is it ?

6 Copies

extremely unlikely that all are lost

geographic distribution of caches

total loss is even less likely

Constant Integrity Checking by caches

LOCKSS daemons do the right thing
as long as plugins behave and
provider sites are accessible.

LOCKSS daemons do the right thing

They communicate only with daemons on known caches
List of cache IPs managed by MetaArchive staff.

They encrypt their communication
They use the same technology as banking web sites do.
Certificates are stored locally on disk.
Certificates are safely transferred to members.

Configuration files on web hiding behind firewall.

User interface access is password and IP address protected.

Daemons refuse to use uncertified plugins.
Certification keys are stored with configuration files behind firewall.

Daemons are very conservative: Never delete content.

LOCKSS is award winning software, runs on 100s of caches.

Member Responsibilities

Take care of your Content

- Open Website Access to network caches
- Test/Audit Content
- Watch Status: Replication across Caches
  Archival Unit Status across Caches
- Plugin development/maintenance

Run a Cache

- Keep Daemon Software up to date
- Open LOCKSS UI access to cache manager
- Add to Content Configuration
- ... sys admin ...

Pay your dues: $$$

Credits

LOCKSS team
Stanford University Libraries

support
advice
cache manager codevelopment