MetaArchive Architecture

Monika Mevenkamp
MetaArchive Annual Membership Meeting
Houston, Texas
Friday October 23, 2009

Overall Architecture

Content -> MetaArchive

It is Safe

Tasks in Preservation Systems

Ingest get content

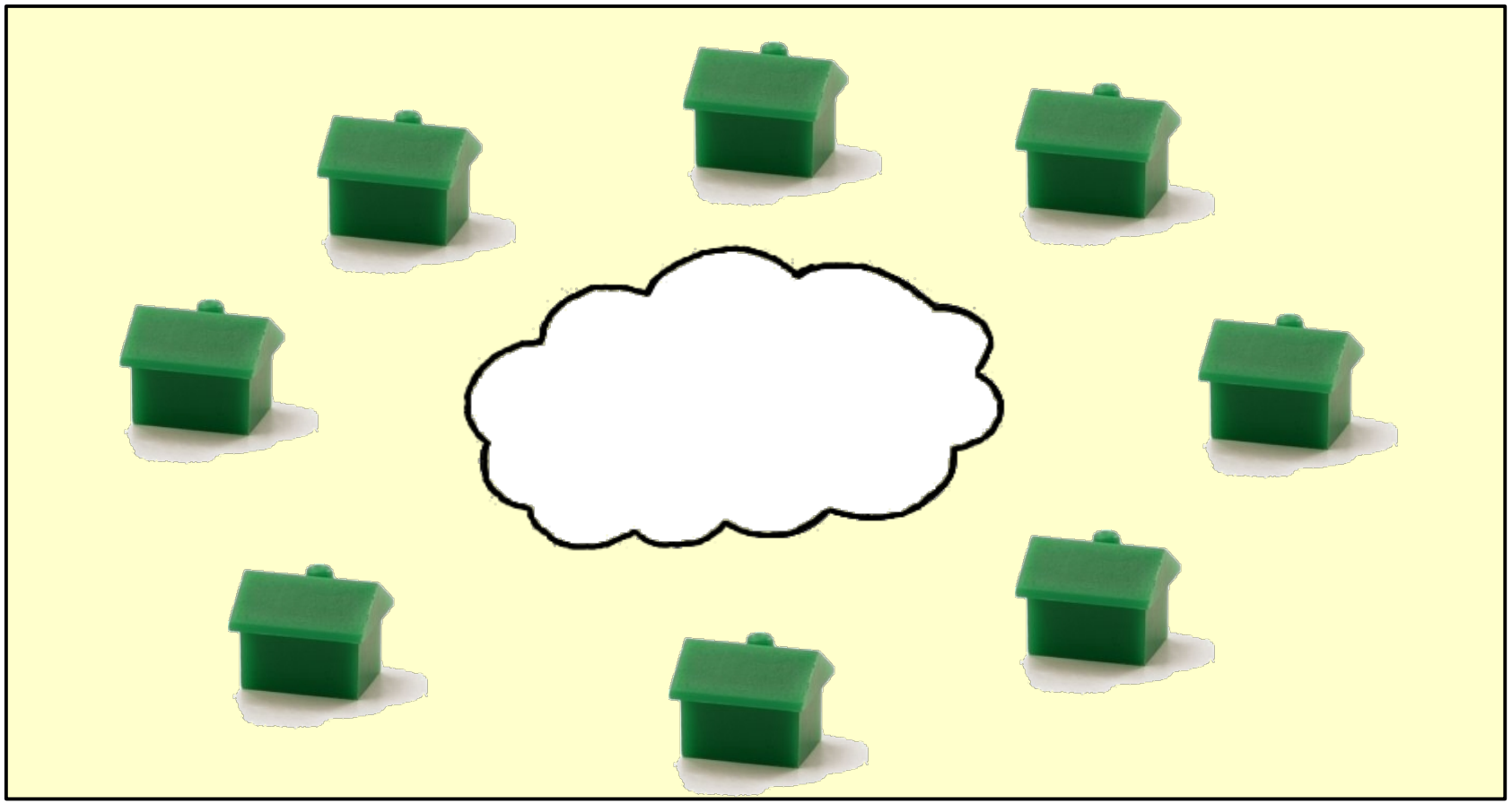
Preserve keep it safe

Update keep it up to date

ongoing

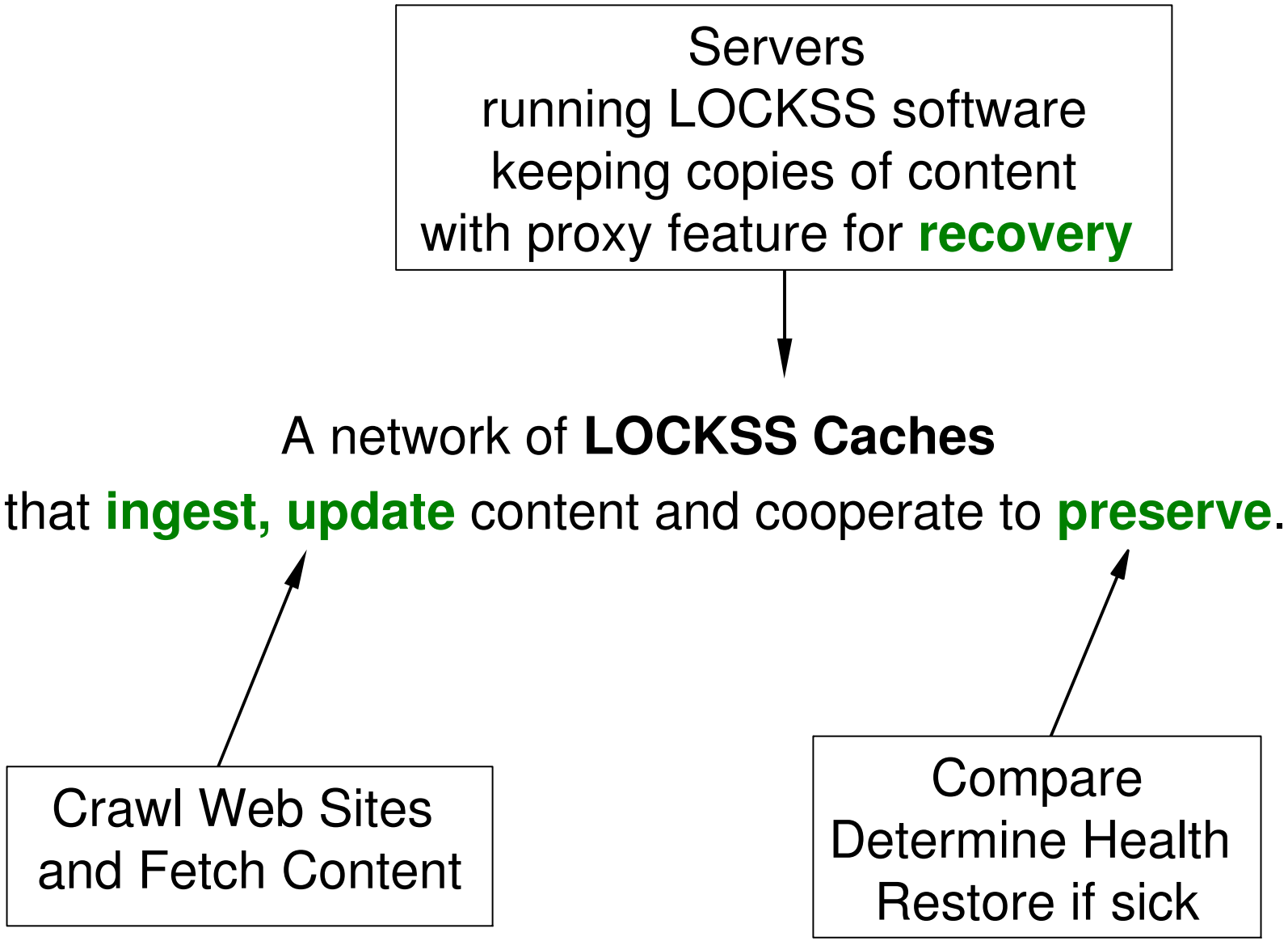
Recover when the data disaster hits

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 URL
KeyStore for Plugins

Maintained By MetaArchive Staff

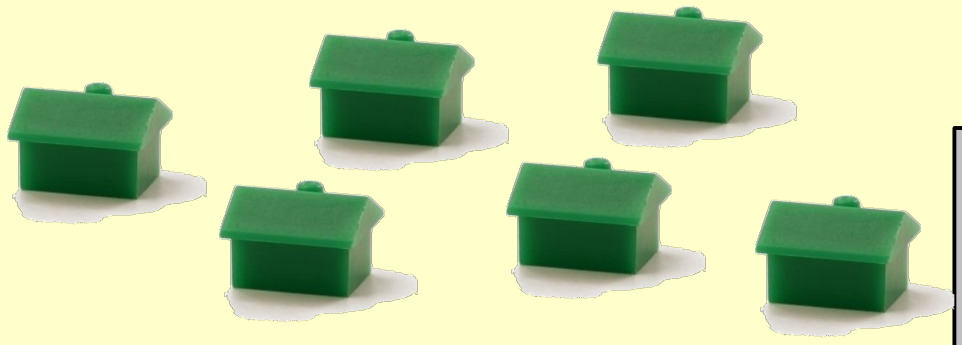
Plugin Repository

Signed Jar Fles

Subversion

Plugin XML ...

Used By Plugin Developers



MetaArchive Caches Running LOCKSS Daemons

Provider Sites

MetaArchive/LOCKSS Cache

Computer with big Disk

Running
- LOCKSS daemon

On
- Security Enhanced LINUX

....

Kickstarted

MetaArchive's Conspectus Tool

Web Based Tool
Editor for Collection Data
- Title, Description, Publisher, Institution, ...
- Risk Rank
- Plugin Name, Base_URL,
- optional extra parameters

Collections organized by Archives
- Southern Digital Culture Archive
- ETD Archive

Generates archival unit definitions for
LOCKSS Title Database

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 as a unit 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 never 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

created by Plugin developers a member sites

maintained in subversion

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 Providers

PLN Parameters

Title Database
Plugin Repository URL
KeyStore for Plugins

Maintained By MetaArchive Staff

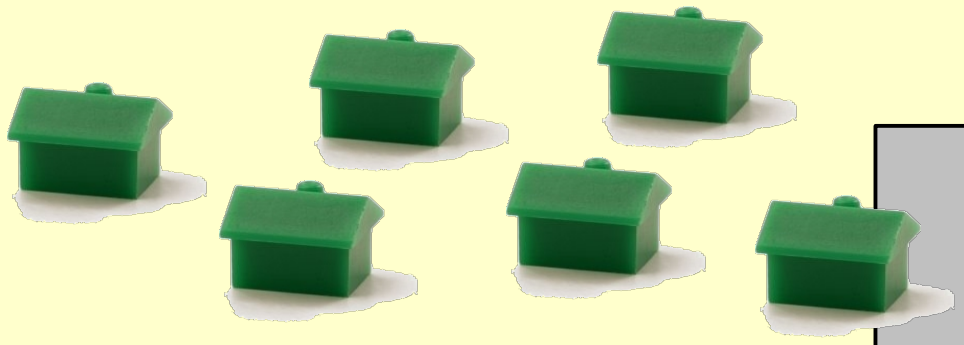
Plugin Repository

Signed Jar Fles

Subversion

Plugin XML ...

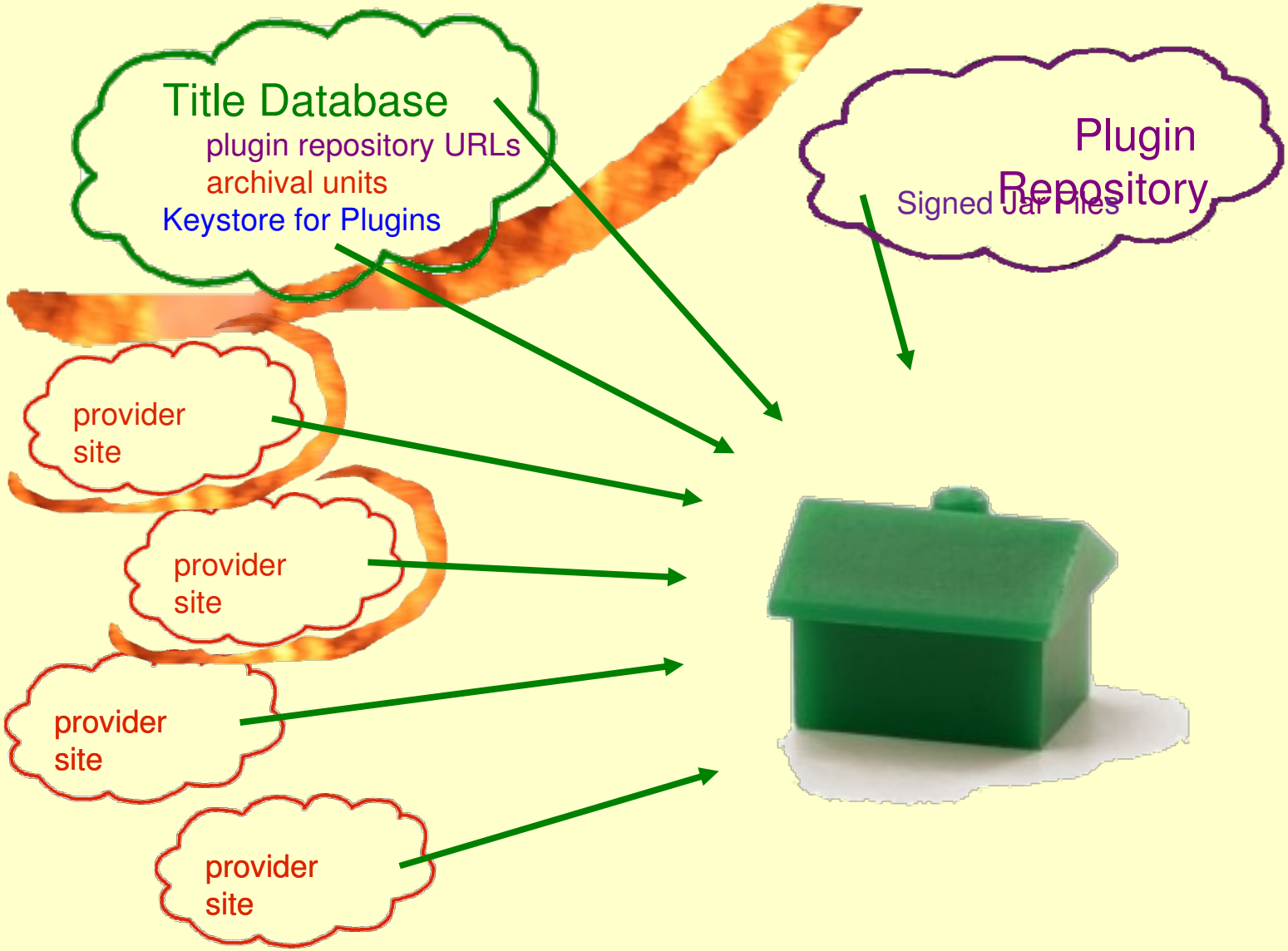
Used By Plugin Developers



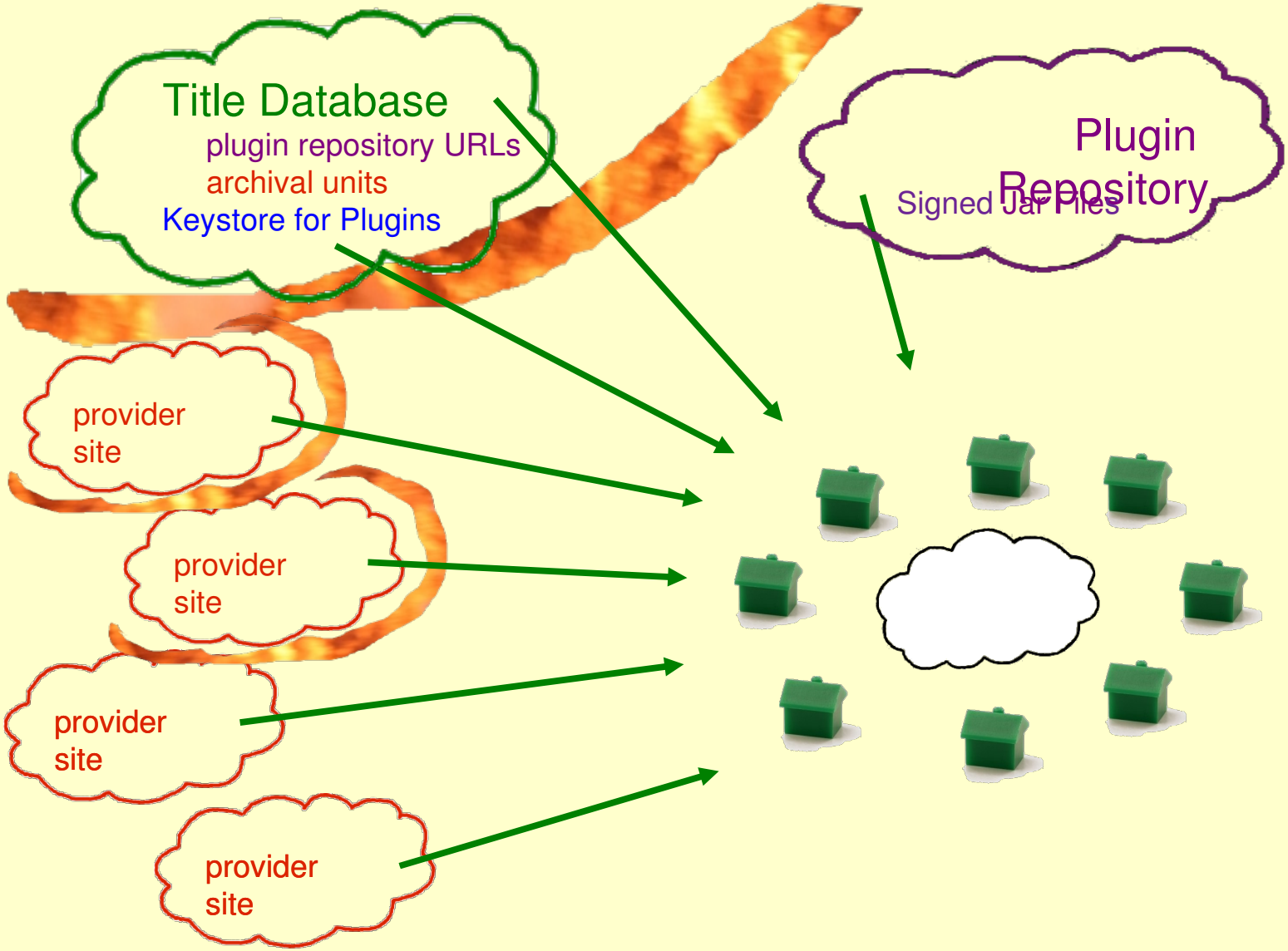
MetaArchive Caches Running LOCKSS Daemons

Provider Sites

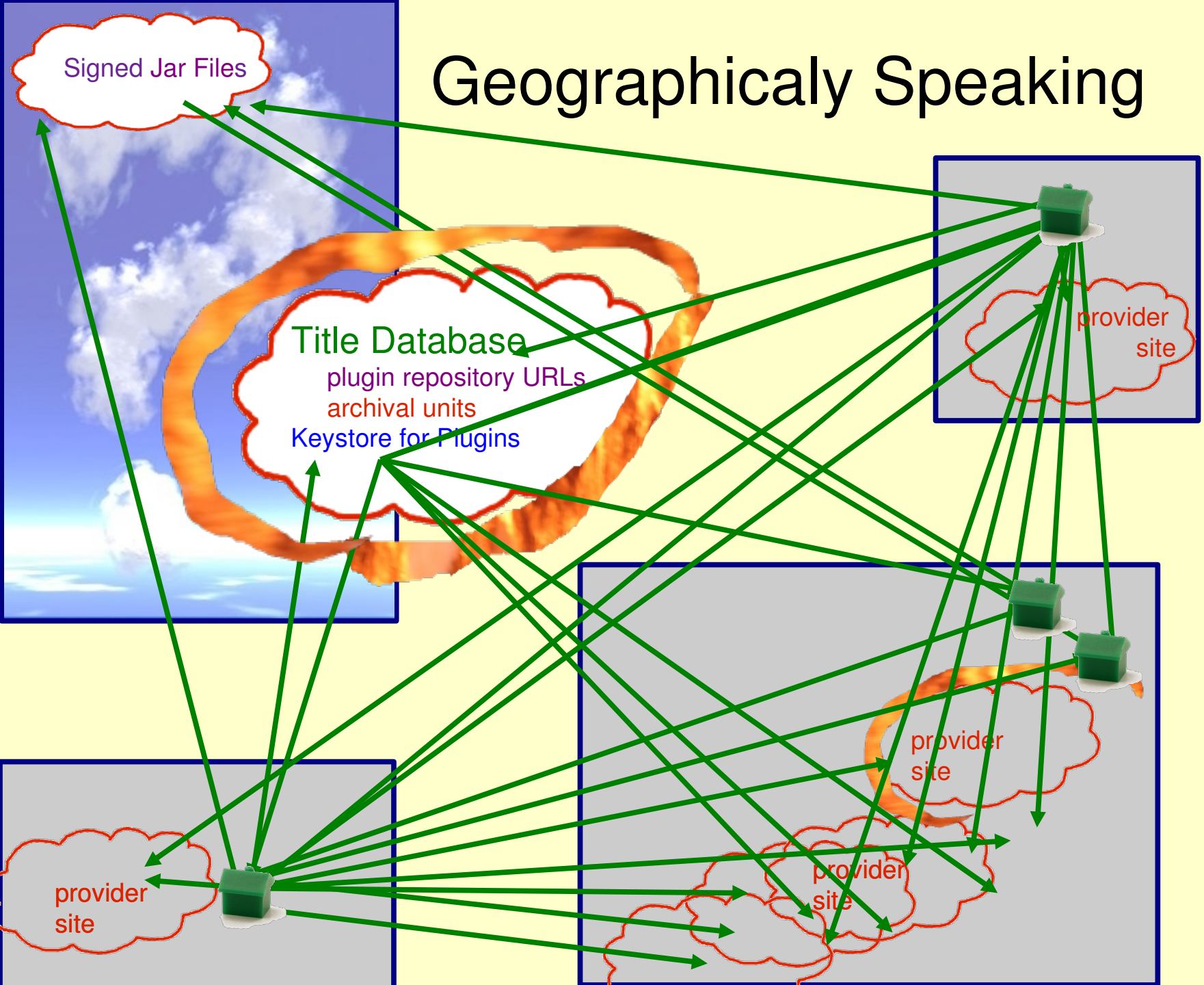
MetaArchive/LOCKSS Cache



MetaArchive/LOCKSS Cache



Geographicaly Speaking



MetaArchive/LOCKSS Daemon

Title Database
plugin repository URLs
archival units
Keystore for Plugins

Plugin Repository
Signed Jar Files

provider site

provider site

provider site

provider site

initialize

do forever
crawl provider sites

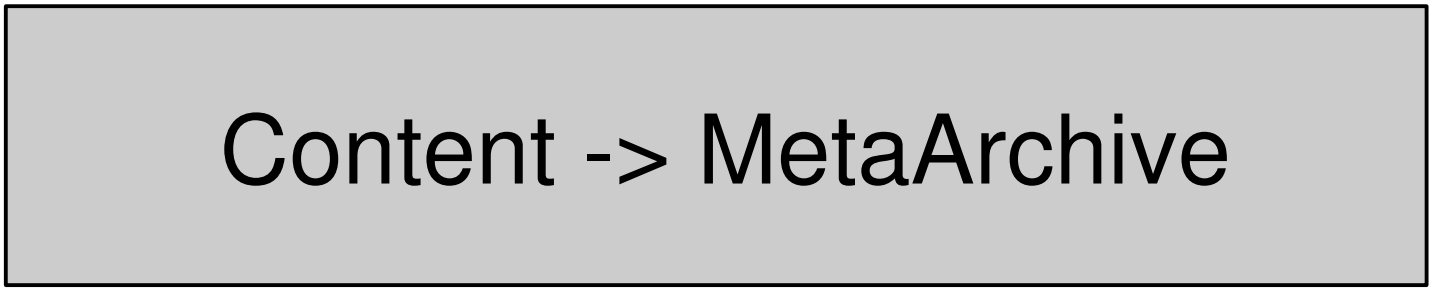
participate in votes about content state
initiate votes

repair broken content

reinitialize

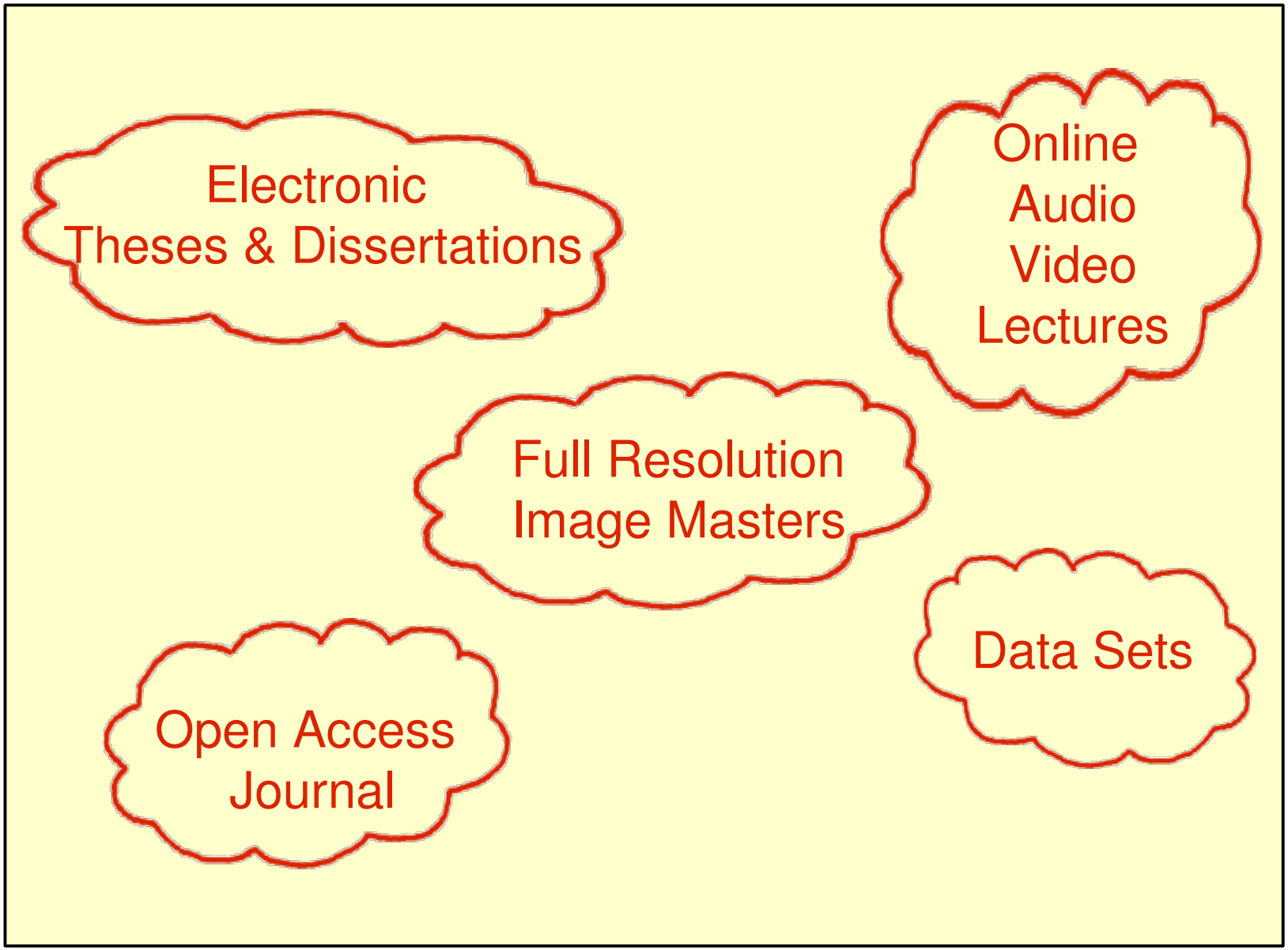
by re-crawling provider sites
or
by restoring from peer caches

Overall Architecture



It is Safe

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 URL
KeyStore for Plugins

Maintained By MetaArchive Staff

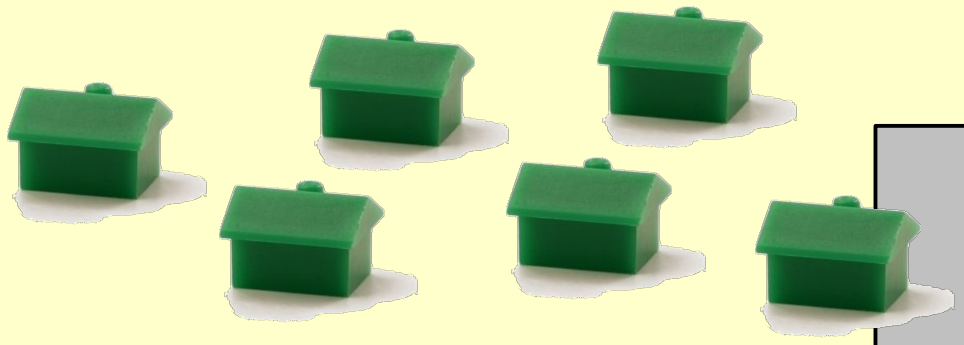
Plugin Repository

Signed Jar Fles

Subversion

Plugin XML ...

Used By Plugin Developers



MetaArchive Caches Running LOCKSS Daemons

Provider Sites

Define Collection in Conspectus

Title: Talks By The Famous Guy
Description: Lectures Given Since ....
ContentProvider: Some Where University

LOCKSS

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

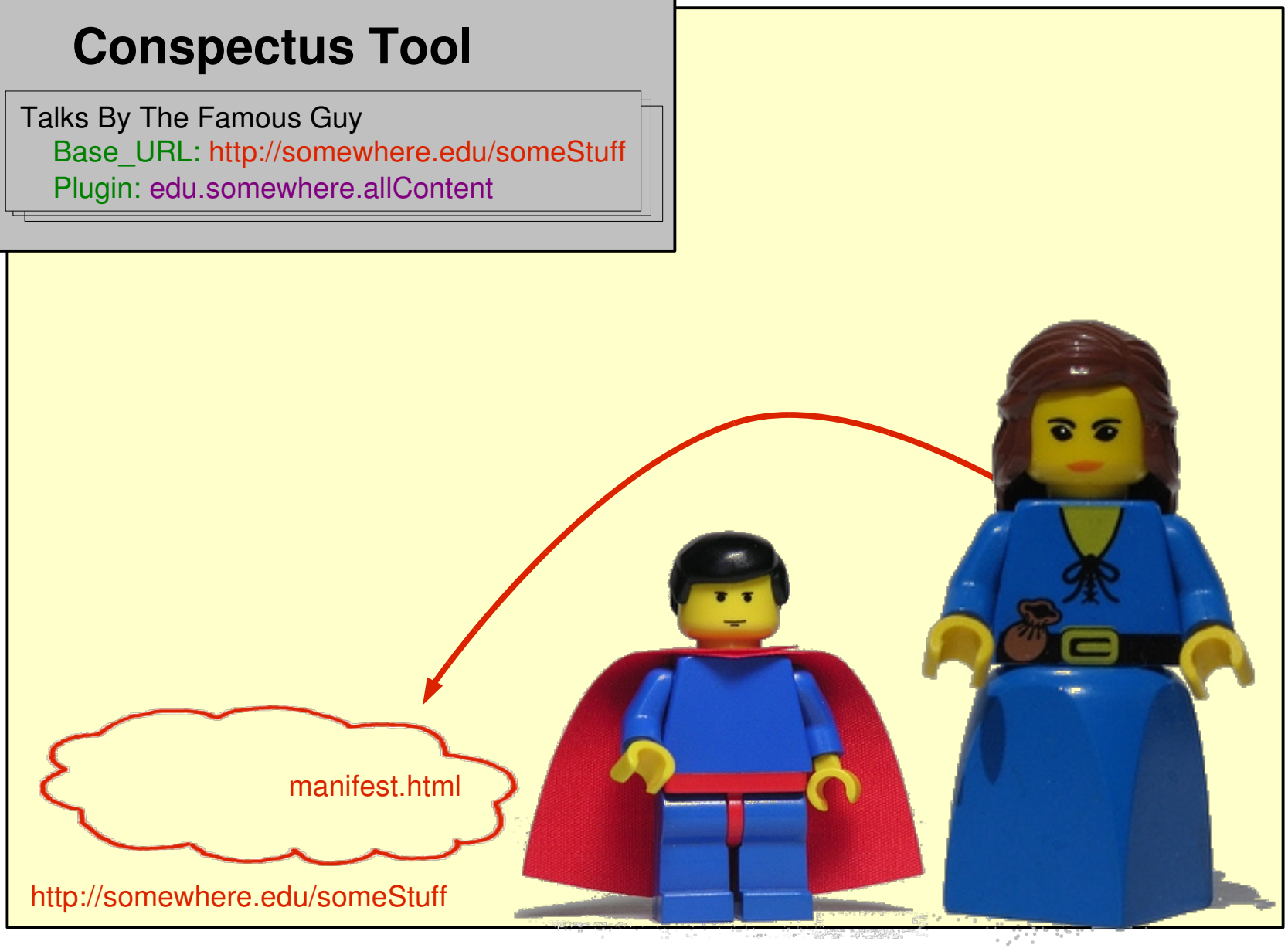
METADATA

Creator: Famous Guy, Famous Co-Worker
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

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

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 / Test Plugin

edu.somewhere.allContent

start with similar plugin (subversion/metawiki)
create/edit with plugintool
test with plugintool
test with run_one_daemon
mark aus in conspectus 'test'
test with metaarchive test cache
test for correct ingest
    does it get all desired content
test for correct update / reharvest
    does it pick up changes on web site ?
When ready:
    Metaarchive staff chooses Preservation Caches
After ingest
    check on status
    cachemanager & Cache UI

check plugintool trace
review content
    copied by run_one_daemon
use audit proxy
    of run_one_daemon/test cache)

Ingest into the Network

Conspectus Entry

Talks By Famous Guy

Web Site /BaseURL

http://somewhere.edu/someStuff

Manifest Page

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

Plugin

branches/test/edu.somewhere.allContent

Deploy Plugin – To SVN

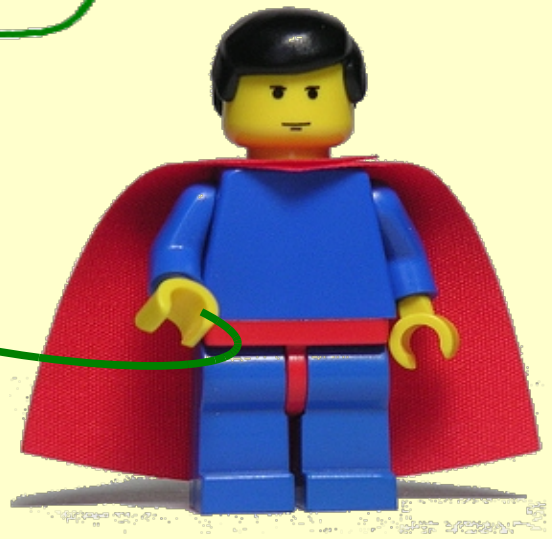
Conspectus Tool

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

Plugin Repository

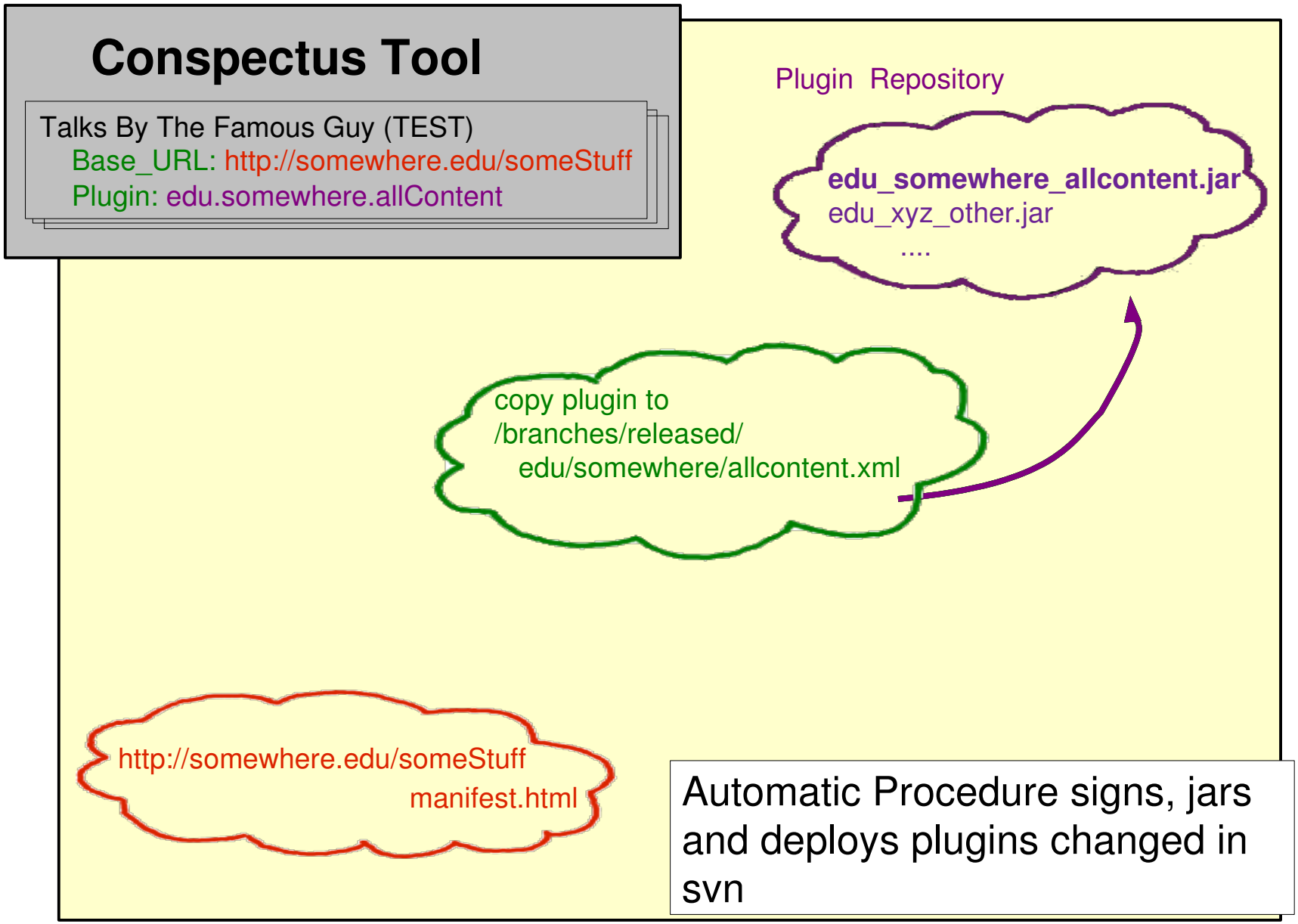
edu_xyz_other.jar
....

copy plugin to
/branches/released/
edu/somewhere/allcontent.xml

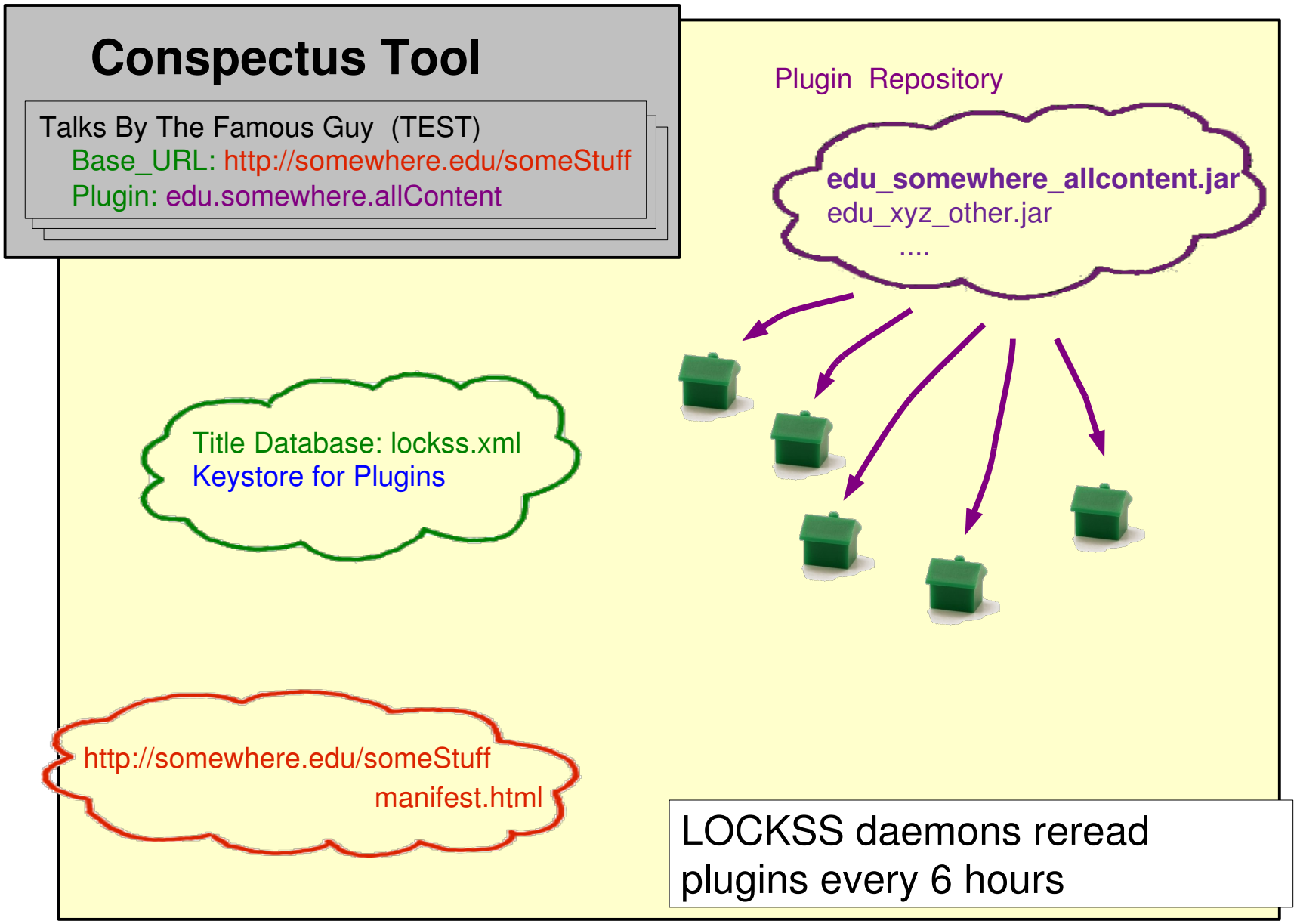


http://somewhere.edu/someStuff
manifest.html

Deploy Plugin – To Registry



Caches Reload Plugins



Collection Ready for INGEST

Conspectus Tool

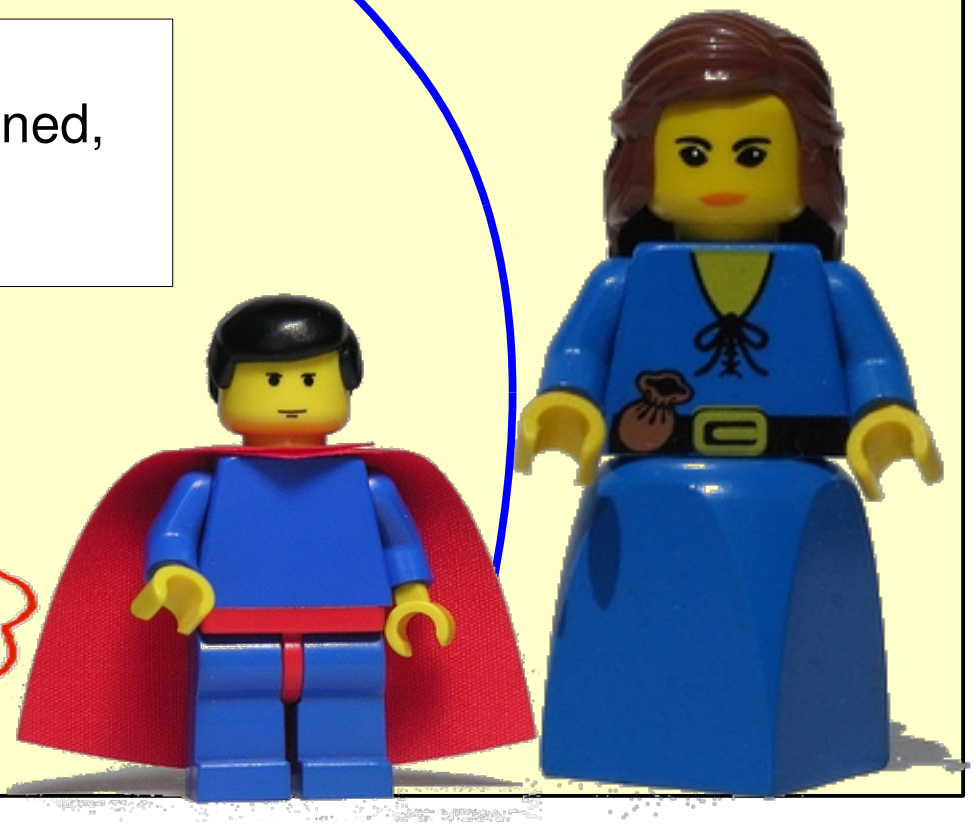
Talks By The Famous Guy (**INGEST**)
Base_URL: http://somewhere.edu/someStuff
Plugin: edu.somewhere.allContent

Plugin Repository

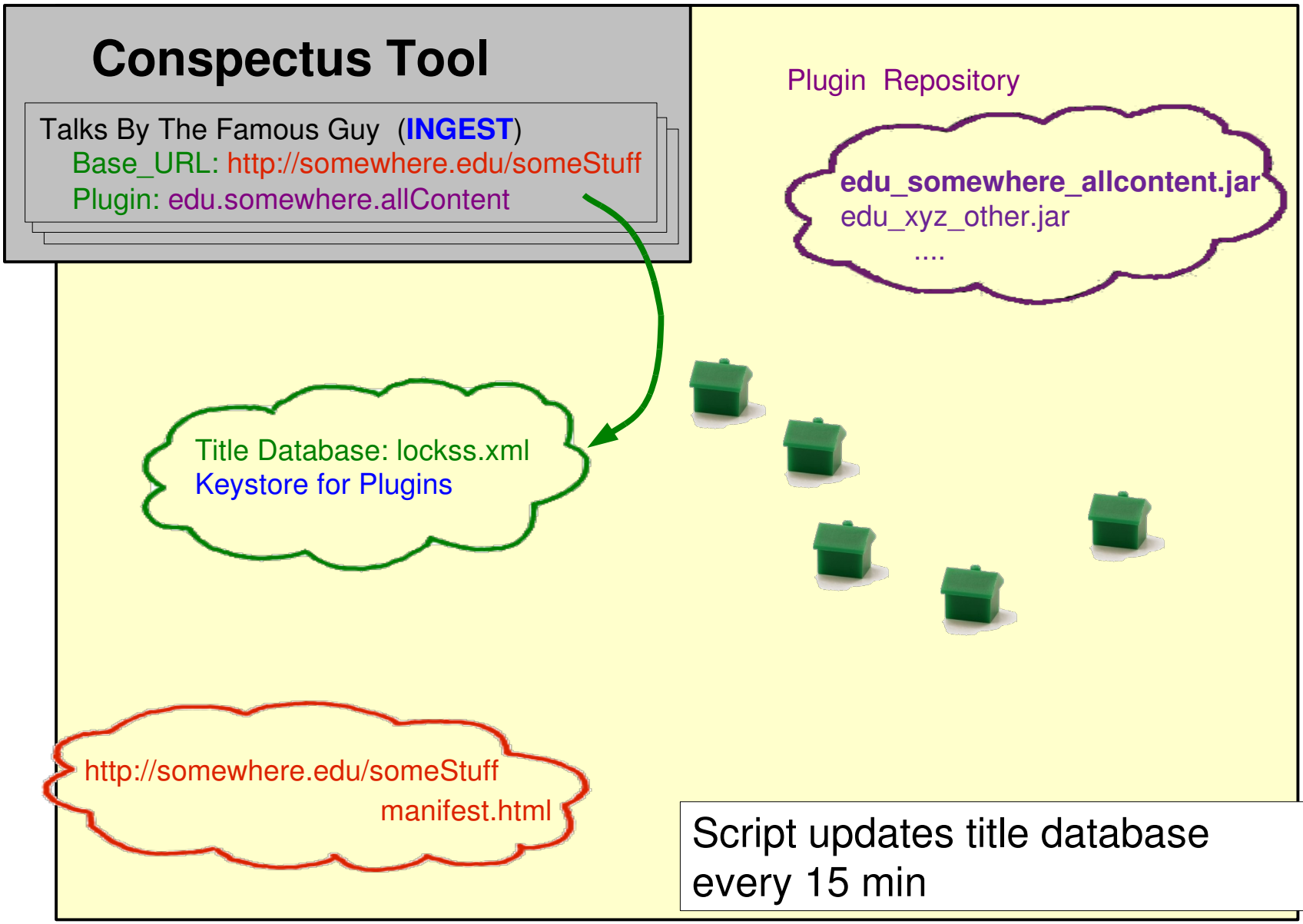
edu_somewhere_allcontent.jar
edu_xyz_other.jar
....

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

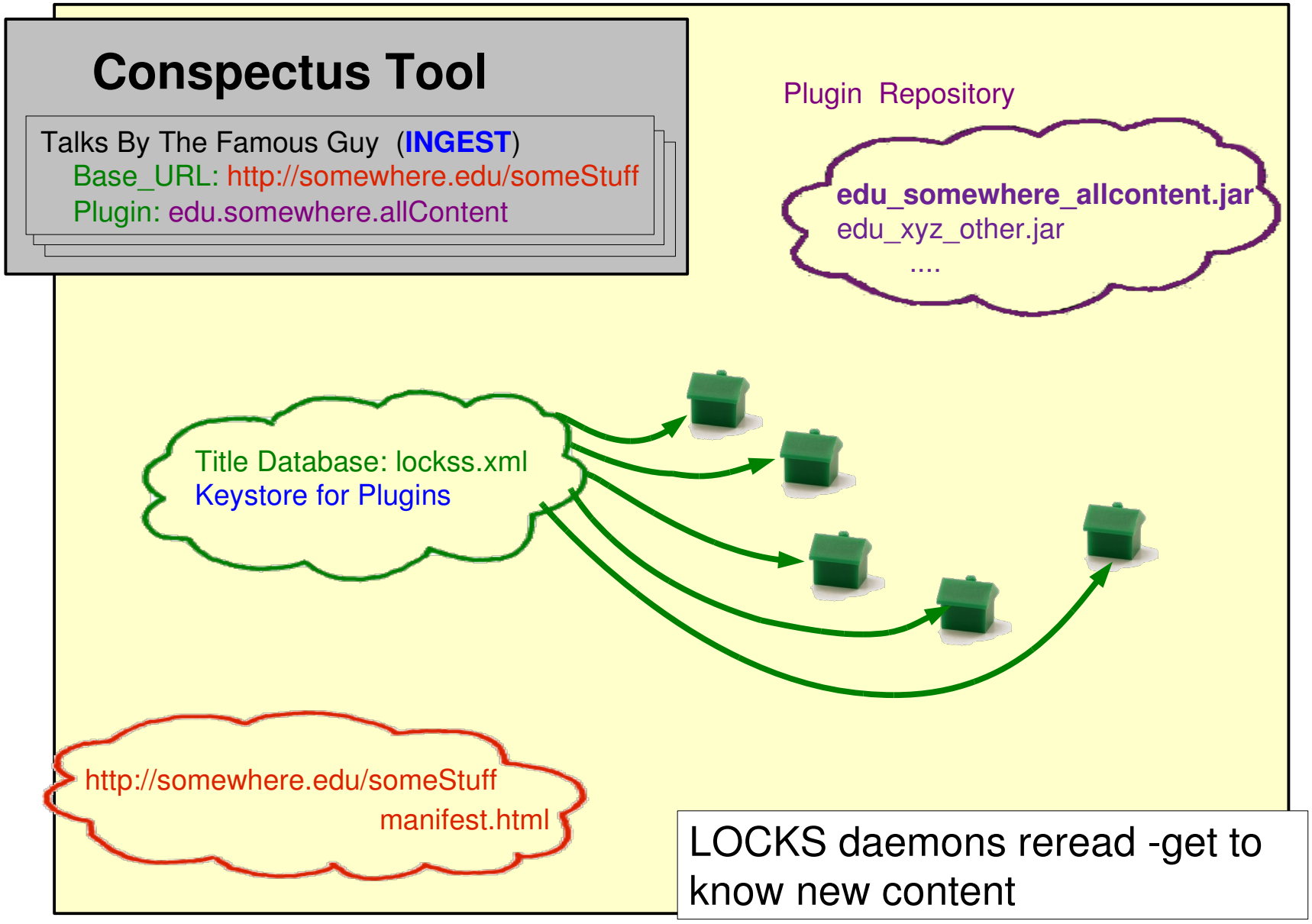
http://somewhere.edu/someStuff
manifest.html



Conspectus Updates Title DB



LOCKSS daemons read Title DB



Humans Activate Content

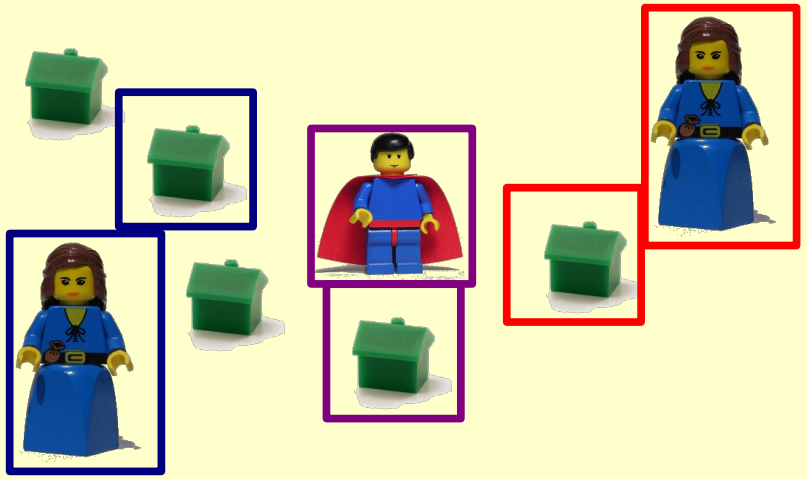
Conspectus Tool

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

Plugin Repository

edu_somewhere_allcontent.jar
edu_xyz_other.jar
....

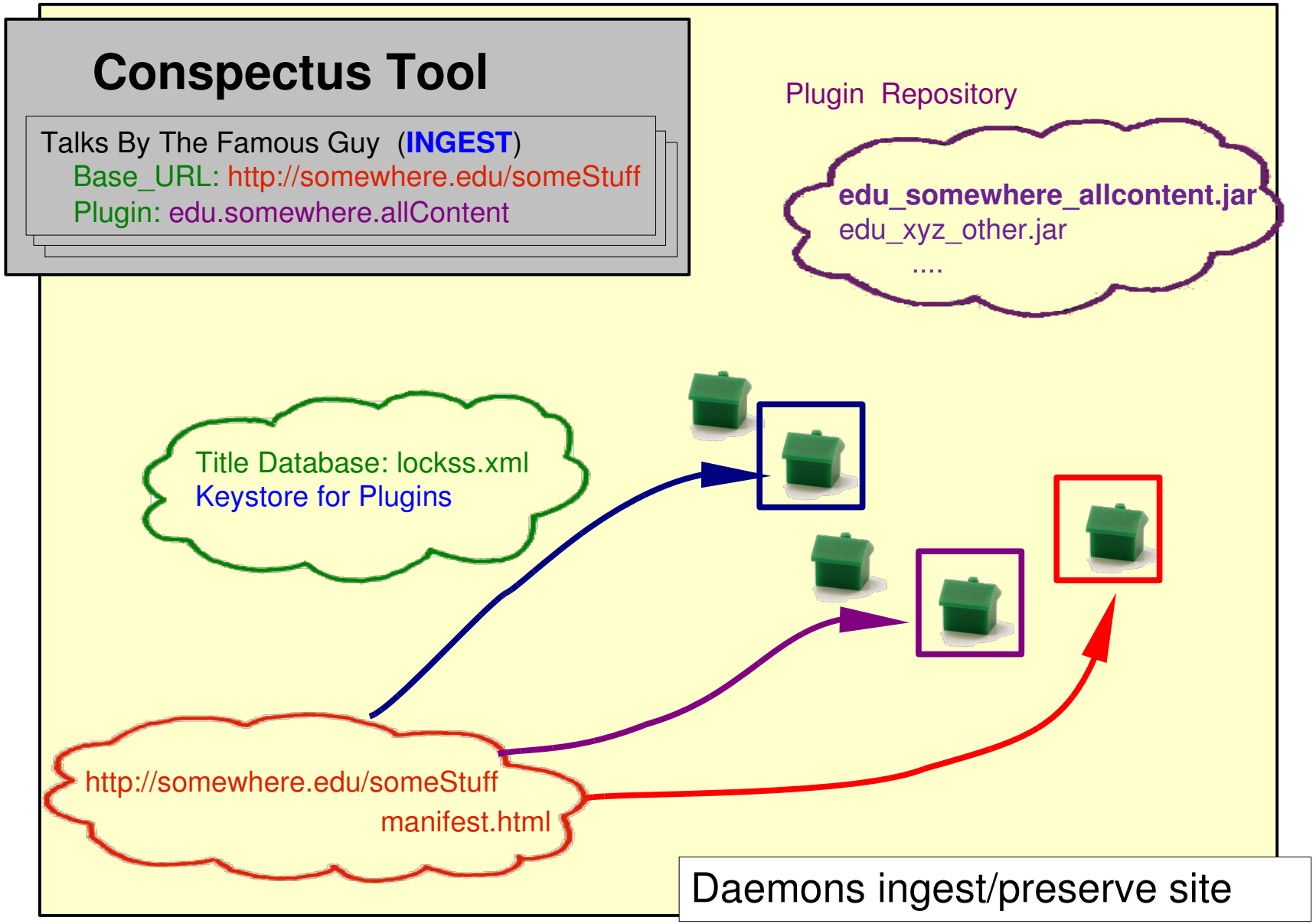
Title Database: lockss.xml
Keystore for Plugins



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

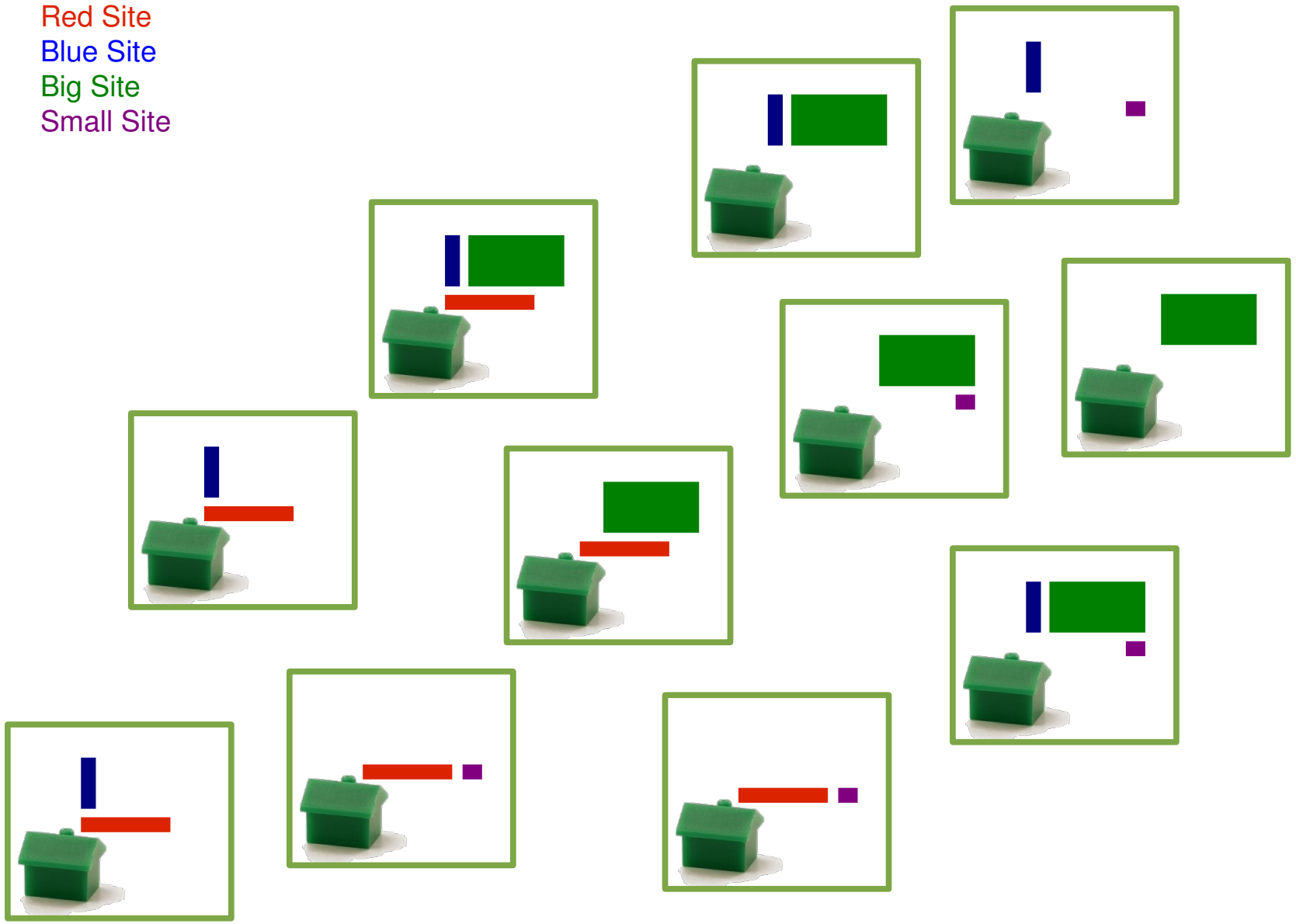
Choose where to preserve

Daemons harvest Content



6 Replications

Red Site
Blue Site
Big Site
Small Site



What They 'Fetch' is What You Preserve

LOCKSS daemons
- start at manifest page, collect and filter links
- use http – get to access content

Plain html based web site

web hosted directory structure

dynamically generated sites
without extra efffort – html only

What They Fetch is What You Preserve

What You Preserve is What You Restore

Plugins decide: Fetch or Not

Get the Plugin Right

Ensure that the Plugin does the Right Thing

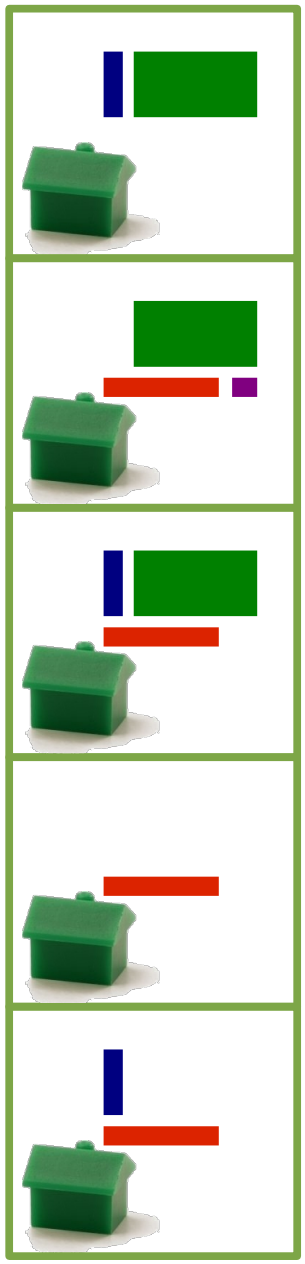
Don't Skimp in Plugin Development Stage

Check content in [test] daemon user interface.

Use LOCKSS daemon's Audit Proxy

Dry Run the Recovery

If Content Site Changes Revisit Plugin



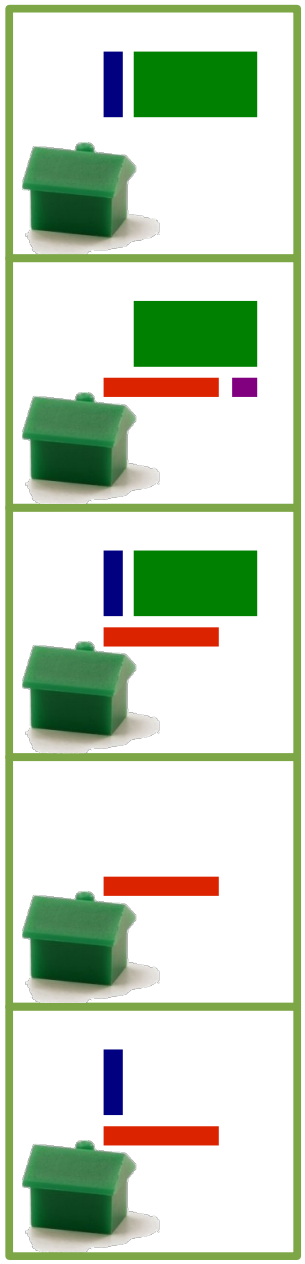
Network Monitoring

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
- Cache down

Overall Architecture

Content -> MetaArchive

It is Safe

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
and replication is maintained

LOCKSS daemons do the right thing

Configuration files on web hiding behind firewall

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

Communication is encrypted
They use the same technology as banking web sites do.
Certificates kept safe on local disk
Certificates tranfered safely to new 'caches'

Daemons refuse to use uncertified plugins.
Certificates behind firewall

LOCKSS cache UI access
with password from restructed list of trusted IP addresses

Daemons NEVER DELETE content.

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

The Human Factor

MetaArchive Staff goes postal
- access Caches in network only through web UI
- content deletion impossible

Cache Administrator zaps a cache
- 5 more copies

Somebody Hacks Web Site - even if unnoticed
- LOCKSS caches keep file revisions

Rogue cache tries to join network
- must get access to SSL key
- must get its IP address listed in title database
- its just one cache – can't tip voting balance
- takes too much resources

Wrap Up

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

Credits

LOCKSS team
Stanford University Libraries

support
advice
cache manager codevelopment