The *Depot* :
A Framework for Sharing Software Installation Across Organizational and Platform Boundaries

Kenneth Manheimer (klm@cme.nist.gov) - NIST

Barry A. W arsaw (warsaw@warsaw .nlm.nih.gov) - Century Computing

Stephen N. Clark (clark@cme.nist.gov) - NIST

Walter Rowe (rowe@cme.nist.gov) - NIST

National Institute of Standards and Technology

DRAFT

Why the Depot?

Need to maintain software packages that aren't part of the OS distribution.

- Commercial Third Party Packages
- Publicly Available packages (Gnu, X, sundry FTP and Usenet-posted stuff...)
- Locally developed software

Installing and managing big applications is not trivial. It takes precious time and expertise.

It's painful (read expensive) to duplicate these efforts.

Instead, we wanted a comprehensive sharing scheme that would allow different organizations to share the *products* of the experts efforts - to share the installations themselves.

Obstacles to Sharing

Chief one is the need to accommodate diversity, we need to accommodate:

Need to accommodate diversity -

- Of Host Environment
  Hardware, OS, configuration
  customization... across hosts used by different organizations
- Of Application Packages - come in many forms
- Of Affected Clientele

Not only
+ Application end users, but also different concerns of
+ Application managers , and
+ Host system managers are 'users' of the depot that must be carefully provided for

And Scalability - need to accommodate Incremental Growth

Additional interdependencies implicit in sharing require addt'l measures to assure reliability.

Overall we need something comprehensive but not unnecessarily complicated

What Does the Depot Do?

The depot provides a framework for distributing application installations across diverse host environments.

It provides for (all stuff mentioned in previous slide):

- Multiple Host Platforms,
- Diverse System Configurations,
- Diverse Application Packages,
- Normal Product Maintenance
- Gradual Additions of Hosts And Applications
- Redundant Fallbacks For and Staged Release of Applications

It strenuously avoids introducing unnecessary interdependencies

It employs conventional, contemporary OS mechanisms in reliable ways.

And in particular does all this while strenuously avoiding unnecessary complexity.

What Doesn't the Depot Do?

- Not for gradual project development
- Doesn't replace simpler */usr/local* types of arrangements, rather it complements them.

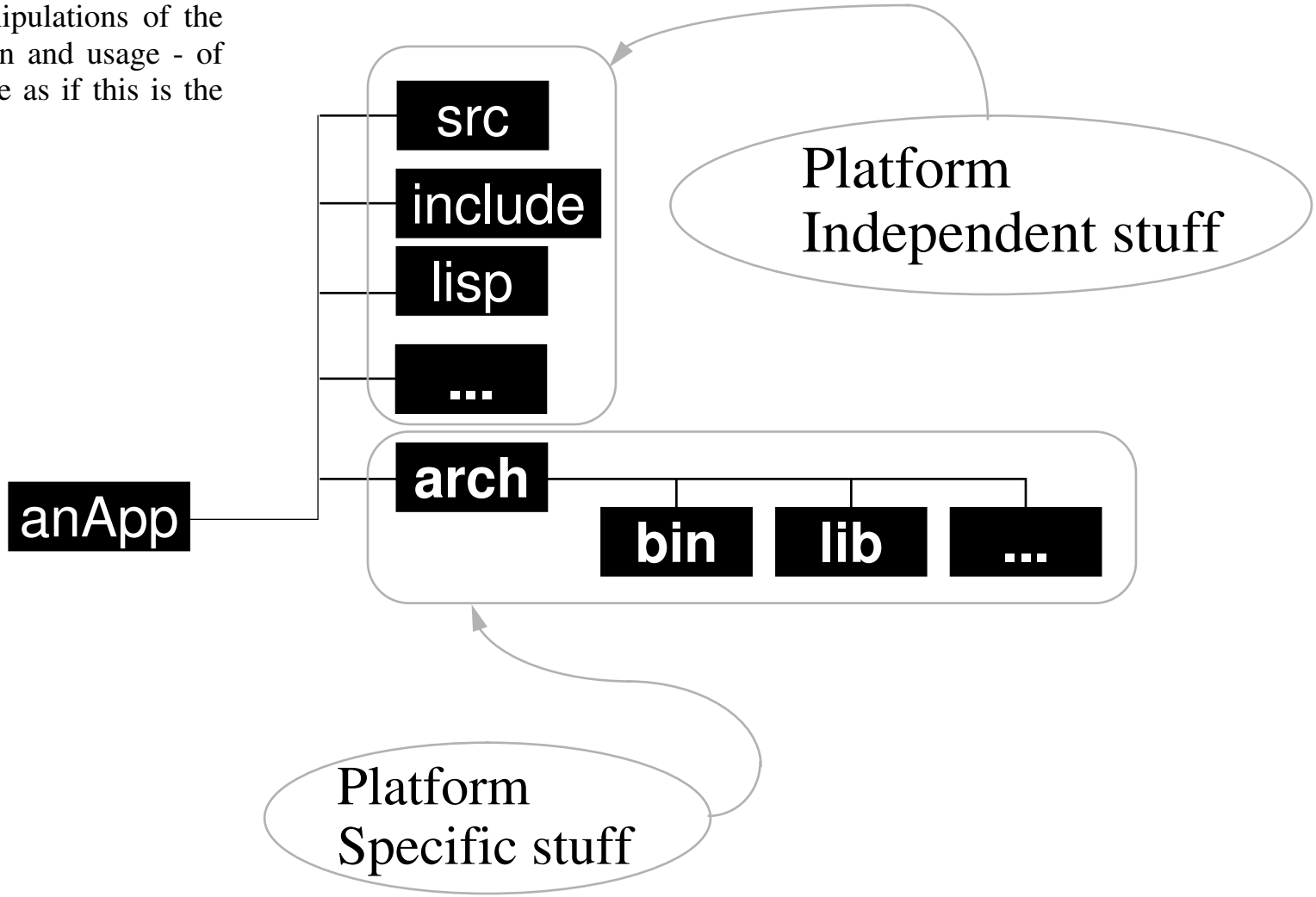
Consider the depot as a sort of globally scoped version of /usr/local, leaving /usr/local/ for use by individual cluster managers.

Design

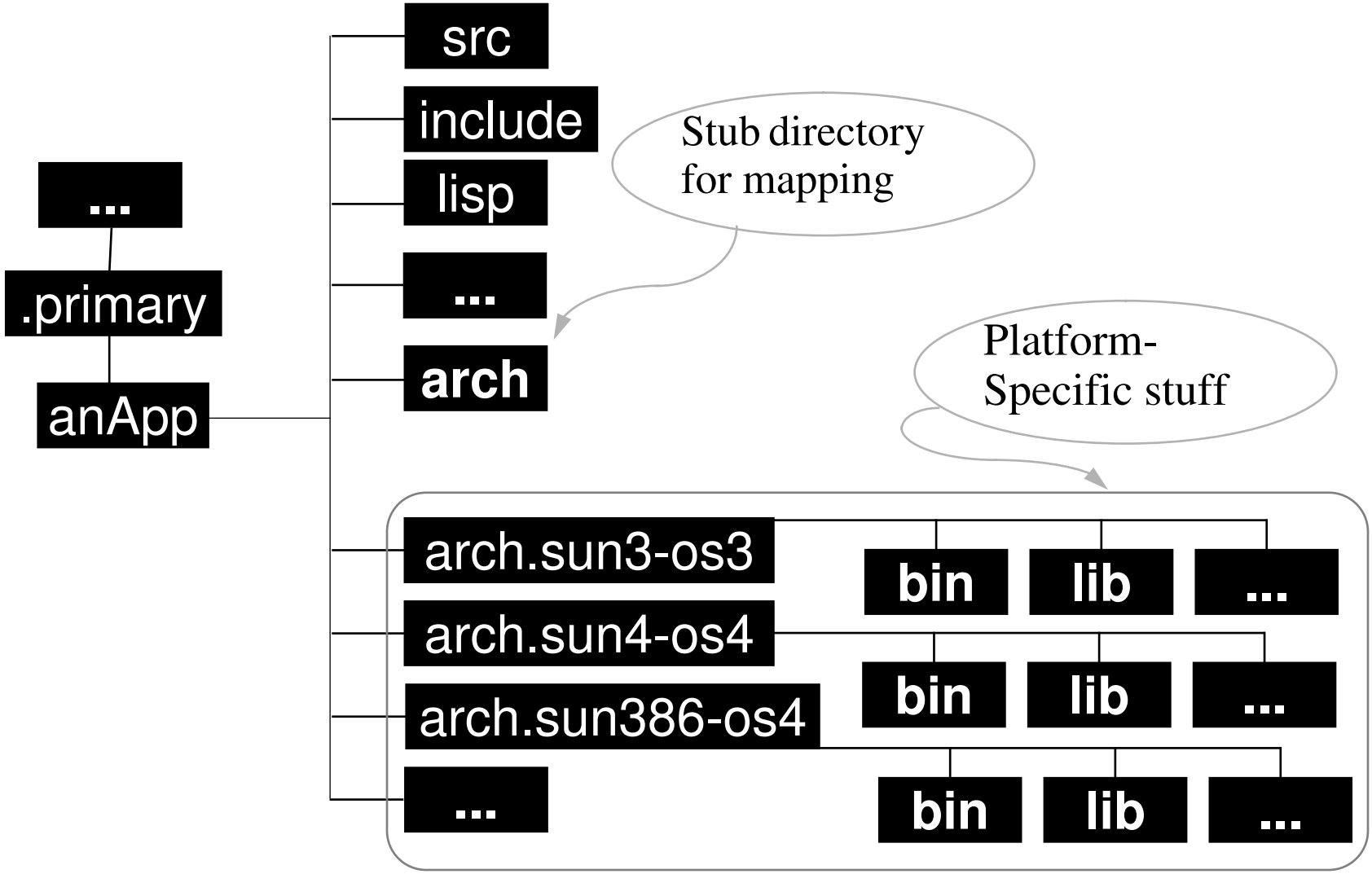
- Essentially, two file system structures and a mapping between them.
- First structure is actual residence of app distribution as well as places for platform-specific components for each supported platform. This is used as origin for...
- Second structure has suitable platform-specific components in a slot which is same location across platforms.
- Second, platform-specific view is operational view. Installation and access to applications is done with respect to the platform-specific client view, not the “origin” view.
- Host system administrators arrange for hooks to the client-specific arrangement however they want - hooks from */usr/local/bin* , user load paths that include the depot directory or whatever.

Abstract Application Layout

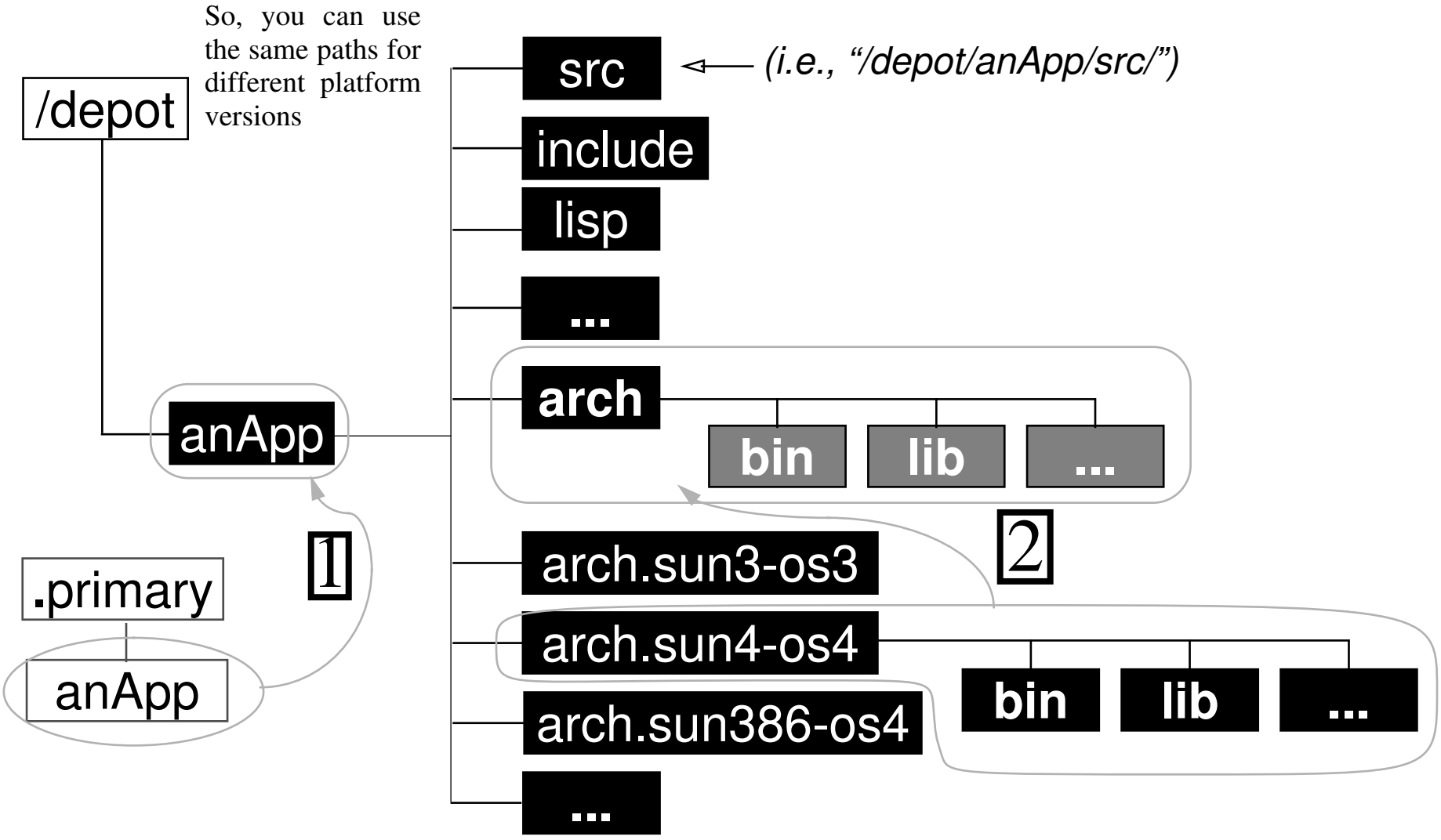
Ideally, all manipulations of the app - installation and usage - of the app are done as if this is the structure.



Application Origin Layout



Origin - Client Mapping



Automount is Very Helpful

Allows you to systematically administer depot mappings

Automount (and probably *amd*, which is available for many platforms) will properly provide for all clients (including clients that are also origins), as well as:

- Providing framework for network administration of distributed depot clients as coherent collections
- Overlay accomplished using 'hierarchical' mounts
- Automatic, proper selection among alternative mounts
- Alternatives accomplished using 'alternative' mounts

(Syntax for combined hierarchical and alternate mounts is tricky - example in paper.)

Results

- Been refining and using it ~ 1 year.
- Its use spans two major organizational divisions, will soon include a third.
- Serves numerous applications (all the hard ones that we use)
- To over 100 workstations.
- More uptime.
- We've used our additional space with redundant *.primary* and staging *.develop* dirs, but as scope grows so would savings
- With less duplication of effort we've achieved better and more up-to-date maintenance of applications.

Future Issues

- Have to investigate how we might be able to benefit from incorporating “transparent” (“translucent”, etc) file systems to our scheme
- Need tools to assist and possibly systematize configuration process
- Need to expand out of Sun platforms, if only to say we’ve done it. *Automount* available on at least Dec, IBM, Silicon graphics, etc...