



Sun Microsystems Laboratories

---

**SDS-99-0411**

*Global Names: Support for Managing  
Software in a World of Virtual Organizations*

**Michael L. Van De Vanter**

The Forest Project, Sun Microsystems Laboratories  
<http://www.sunlabs.com/research/forest/>

**Tobias Murer**

ETH Zürich  
TARSEC AG, <http://www.tarsec.com/>

---

*Ninth International Symposium on System Configuration Management  
Toulouse, France, 6 September 1999*

---

## *Outline*

---

- **Background & Goals**
- **Key Issue: CM + Cross-Organizational Collaboration**
- **Naming in JP**
- **Naming for the Application Web**
- **Related work**

## *Background & Goals*

---

- **JP Development Environment - Sun Labs**
  - reliable, repeatable, scalable build strategy (replace *make*)
  - version/CM repository of orthogonally persistent objects (replace files)
  - enhance code reuse by JP-compliant environments
  - addresses large and *compositionally complex* systems
- **Virtual Software House/GIPSY - ETH Zürich**
  - service architecture for the life cycle of component-based software
  - addresses *organizationally complex* business models
- **The Application Web - joint work**
  - addresses both kinds of complexity
  - fills gap between groupware and manual copying
  - key strategy: replace copying with connections
  - embed software life cycle support in a web of connections

## *Key Issue: CM + Cross-Organizational Collaboration*

---

- **JP strategy relies on closely coupled tools**
  - immutable *package versions* contain sources and a build script
  - *build scripts* are closed functional (style) programs
  - *configurations* are package versions that import other package versions
  - tools communicate via *language interfaces*, even for federated building
- **Virtual organizations rely on managed information sharing**
  - organizational *autonomy* is paramount
  - closeness of collaboration varies
  - cannot rely on common tools
  - manual copying (caching) is the typical last resort, leading to *information loss*
- **Approach: global naming of shared information**
  - hardly revolutionary (WWW)
  - how to integrate with JP development strategy?

## *JP Naming: Versioned Packages and Naming Systems*

---

- **Current tools require management of too many names**
  - language entities
  - files and directories
  - build modules
  - versions and configurations
- **JP developers see only:**
  - `com.sun.labs.forest.jp.util`
  - `com.sun.labs.forest.jp.util.7`
  - `com.sun.labs.forest.jp.util.7.MyStack`
- **JP names are presumed to be:**
  - globally unique
  - immutably bound and never reused

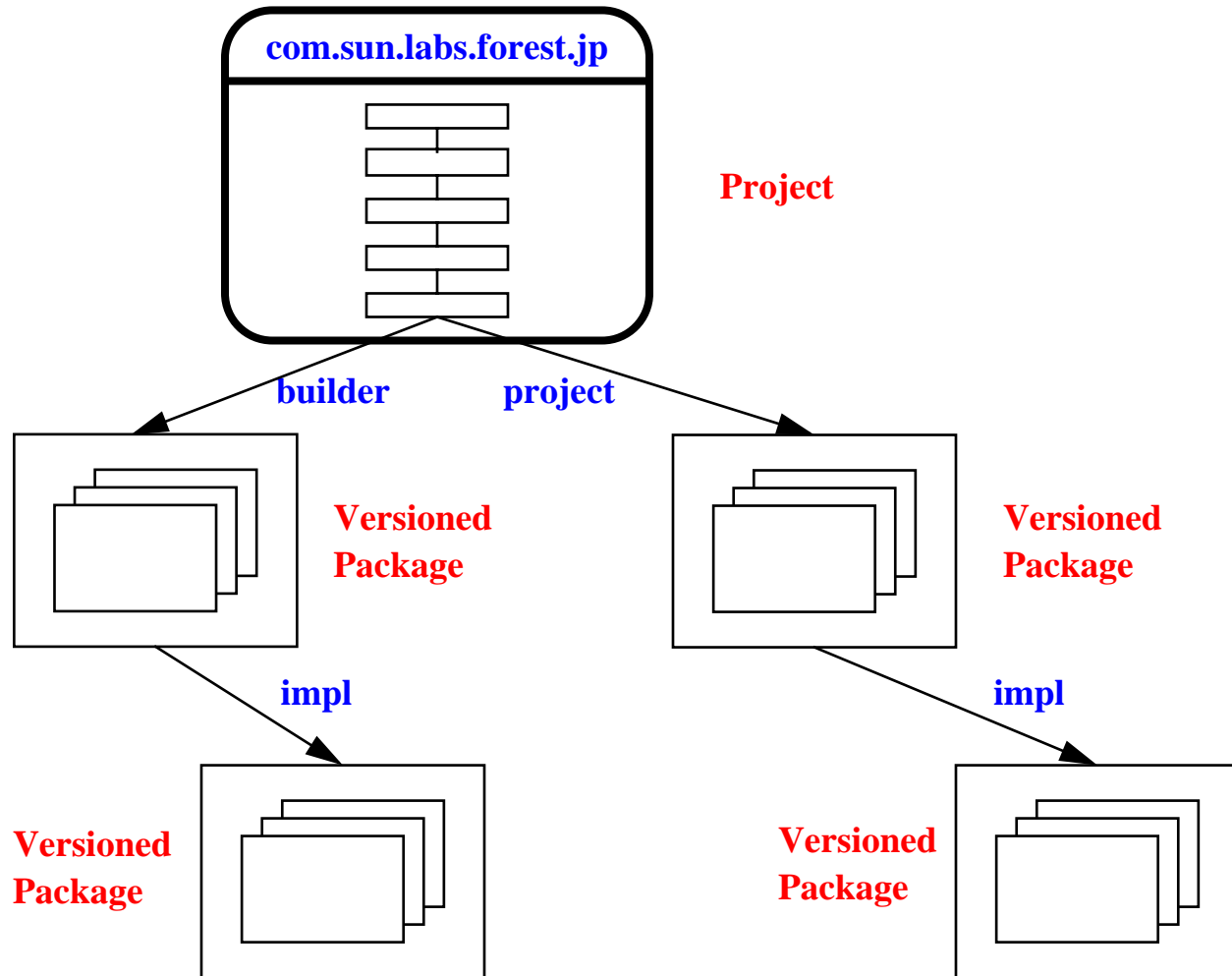
## *JP Naming: Not Used By Build System*

---

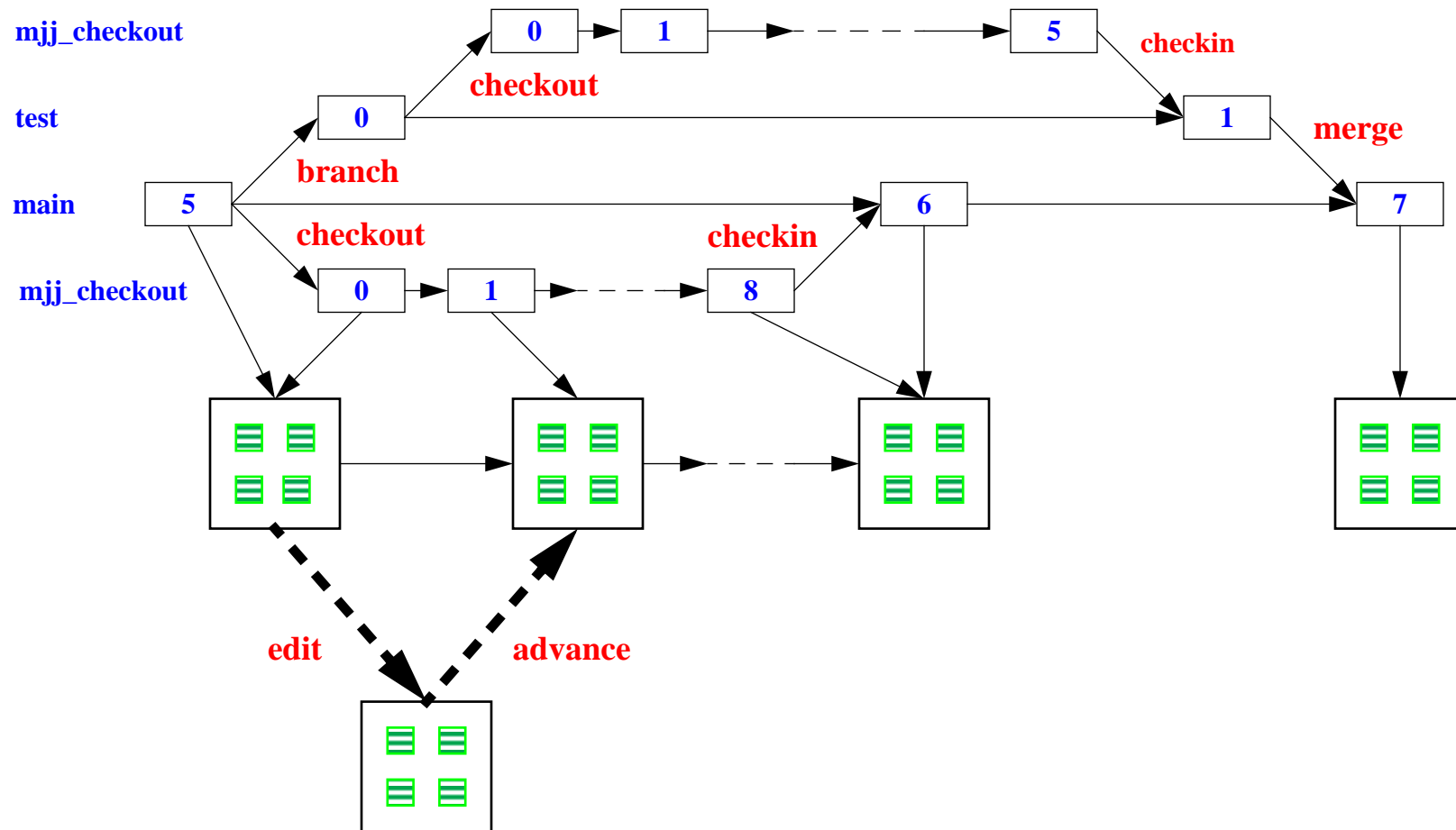
- **JP repository contains orthogonally persistent objects**
- **JP builder computes over repository objects**
- **Federated building uses remote object references**
- **These properties help JP achieve:**
  - **simplicity of implementation**
  - **orthogonality of subsystems**
  - **reliability & scalability**

# JP Repository: Packages

---

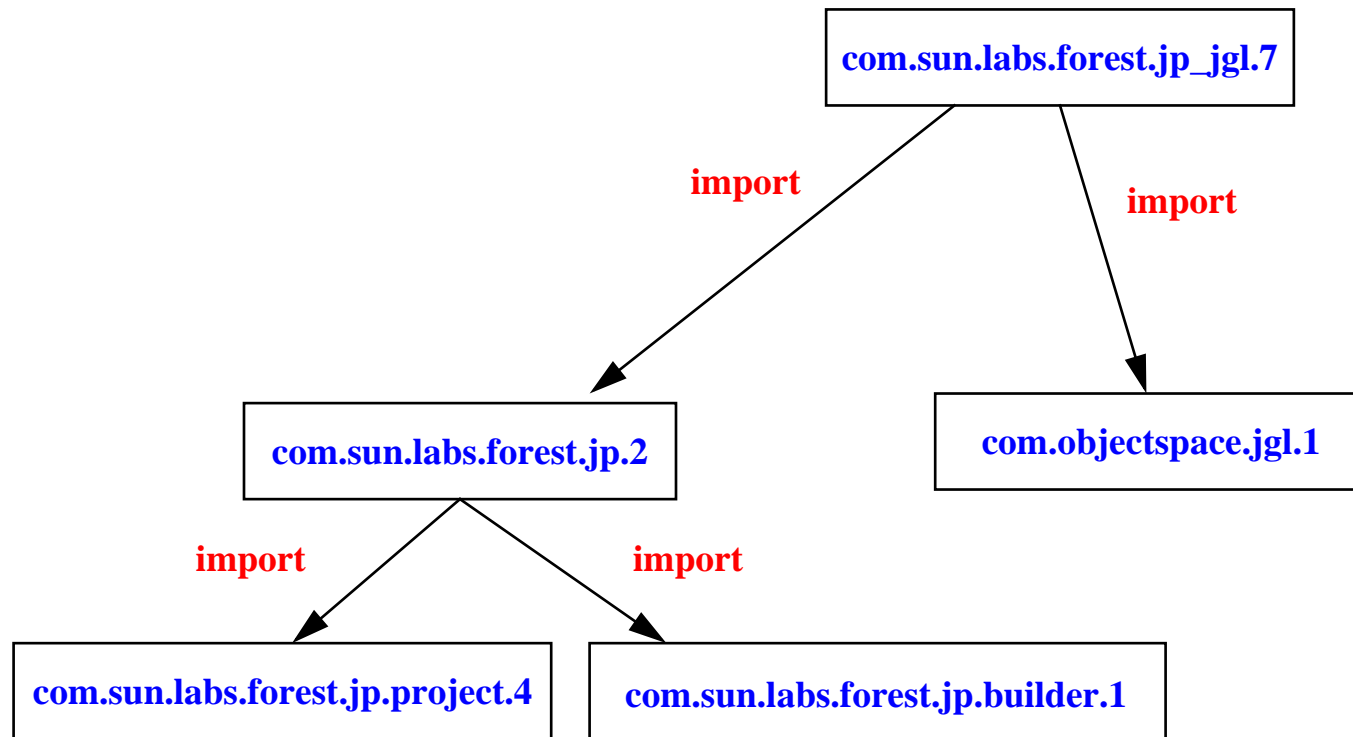


# JP Repository: Package Versions

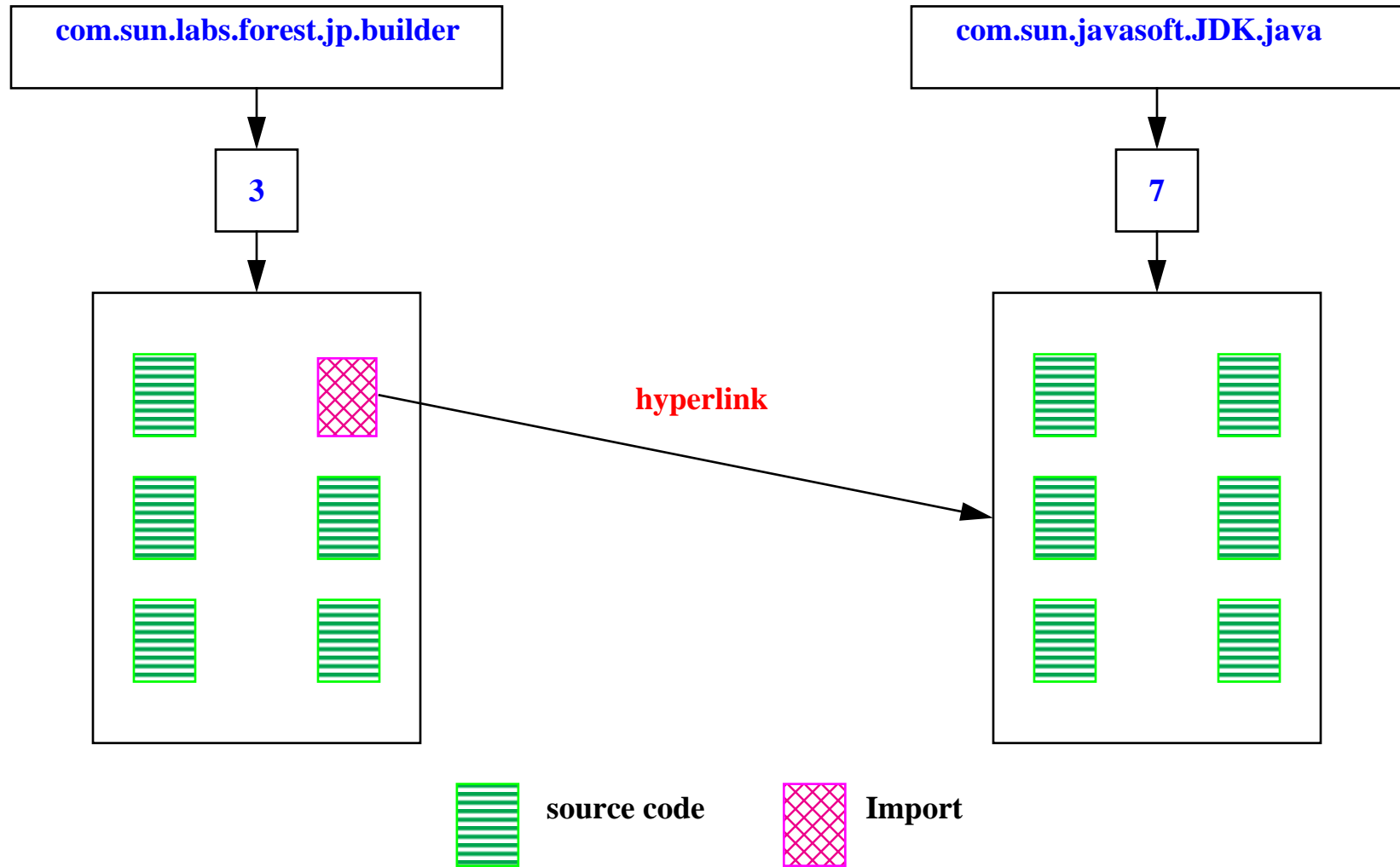


# *JP Repository: Configurations Assembled by Importing*

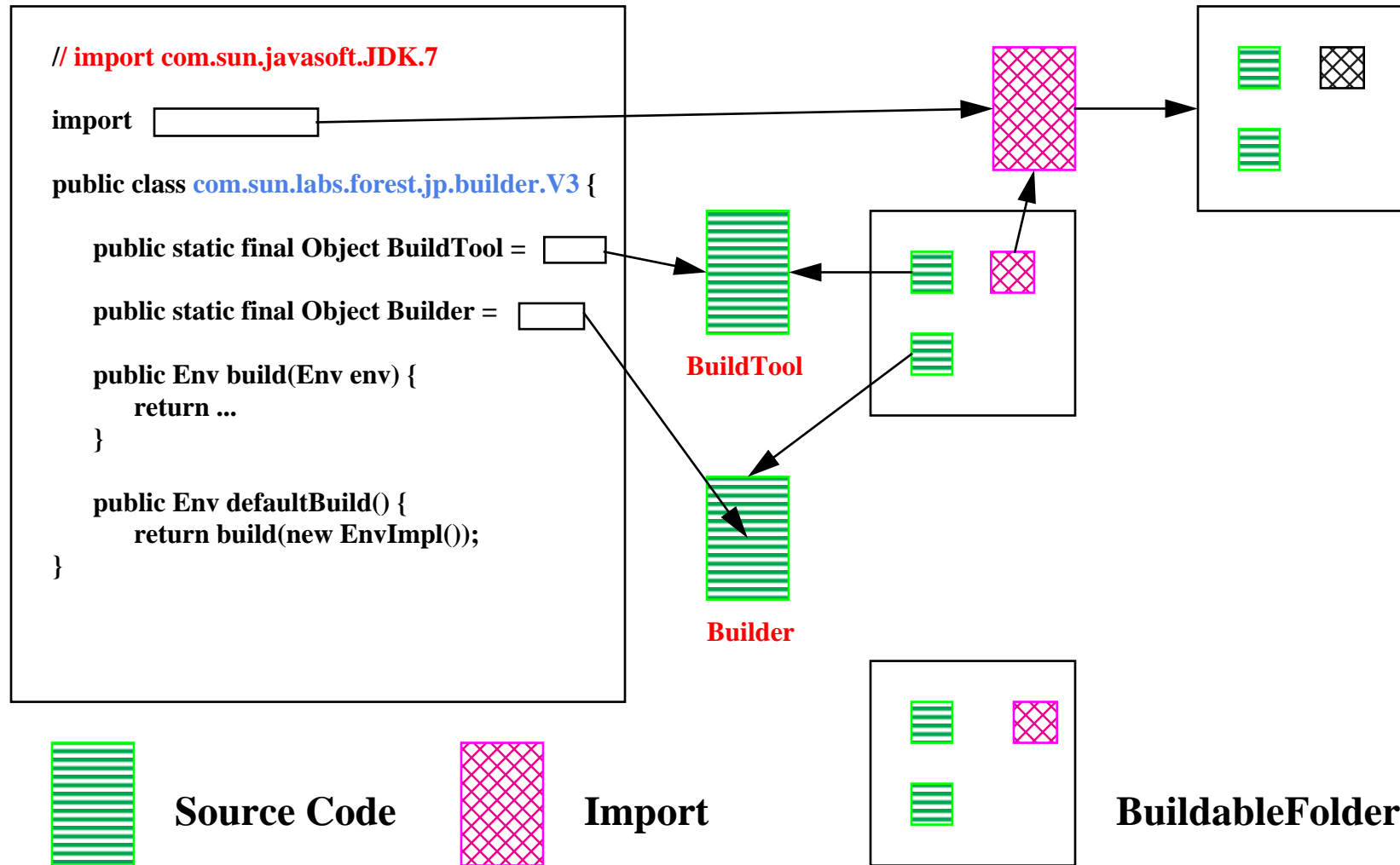
---



# JP Repository: Package Imports are Object References



# JP Repository: Build Scripts are Hyper-Programs



## *JP Repository: What Role Do Names Play?*

---

- **Invisible to JP builder**
- **Essential to JP developer**
  - aligned with language package names
  - visible in all tools
  - only important objects (to developers) are named
  - example: build results are *not* named
- **Essential for language tools**

## *Application Web*

---

- **Names *are* fundamental**
  - key to large scale management of information sharing
  - permit interoperation of many tools
  - align with WWW experience and expectations
- **In other words, make it more like the web**
  - but without loss in reliability for JP
- **Approach: revise and extend JP naming**
- **Ignore for now: authentication, access control, etc.**

## *Application Web Naming: Referential Integrity*

---

- **How to be sure that names are reliable?**
- **Can't guarantee remote information will be available, but...**
- **JP Versioned Packages are *fingerprinted***
  - cryptographic hash of entire object graph
  - enables end-to-end checking when names dereferenced

## *Application Web Naming: Syntax*

---

- **Eliminate possible confusion: packages vs. branches vs. parts**
  - `com.sun.labs.forest.jp.util/7#myStack`
- **Ensure that version naming is flexible**

## *Application Web Naming: Configuration-Relative Names*

---

- **Expose internal structure of JP configurations**
  - a configuration: **CH.ethz.ee.tik.vsh/8**
  - it imports **com.sun.labs.forest.jp/4**
  - canonical part name: **com.sun.labs.forest.jp/4#Main**
  - its embedded name: **CH.ethz.ee.tik.vsh/8/com.sun.labs.forest.jp.#Main**
  - its contextual name: **com.sun.labs.forest.jp#Main**
  - its language name: **com.sun.labs.forest.jp.Main**

## *Application Web Naming: Name More Objects*

---

- **Build results must now be partially named**
  - derived part name: `util.Connection`
  - its contextual name: `CH.ethz.ee.tik.vsh/8/compile#util.Connection`

## *Application Web Naming: Life Cycle Issues*

---

- **Creation**
- **Uniqueness**
- **Persistent Binding**
- **Unavailability**
- **Eternal Names**
- **Binding Expiration**

## *Related Work: WebDAV & Delta-V*

---

- **WebDAV: overlapping goals**
  - distributed authoring
  - HTTP-based
- **Delta-V protocol:**
  - similar models conceptually, different contexts
  - content neutral repository
  - versioned resource (versioned package)
  - editor/client interoperability via protocol (reported in SCM8)
- **Differences:**
  - language objects vs. files
  - Integrated build/CM
  - immutable configurations only

## Summary

---

- **Reliable foundations**
  - replace file system
  - replace *make*
- **Global namespace of versions and configurations**
  - support sharing
  - enable location independent information

## *Future Directions*

---

- **Infrastructure**
  - In particular, compatibility with WebDAV/Delta-V
- **Services**
  - Advanced management of federations of devices, hosts, software
  - Assumption: Software supplier participate on connection service
- **“Virtual Software House” - Software services on the Web**
  - Consistent, up-to-date, connected software catalogue provided by suppliers
  - Reliable software bundling and deployment
  - on-line consulting
  - charging, pay per use
  - component matching, component interoperability checking

## *More Information*

---

**<http://www.sunlabs.com/research/forest/>**

- **JP: Scalable Software Development**
  - papers
- **Orthogonal Persistence for the Java™ platform**
  - papers
  - download PJama (experimental implementation for the Solaris™ operating system)

## *Storage: Persistent Objects implemented by PJama*

---

- **An experimental virtual machine**
  - implements *orthogonal persistence* for Java
- **Collaborative research**
  - Persistence and Distribution Group at University of Glasgow, Scotland
- **A practical technology**
  - almost no API
  - robust – atomic transactions
  - scalable – store sizes up to 2Gb now and larger on the way
  - incremental – operates with an object cache as small as 6Mb
- **The next step on the journey**
  - extend the object model to persistent storage
  - makes objects *real*