

# The Harmony Framework

Software Engineering @LaBRI

[xavier.blanc@labri.fr](mailto:xavier.blanc@labri.fr)



# Empirical Software Engineering

- Define
  - Goal
- Design & Plan
  - Hypothesis, Variables, Instrument
- Operation
  - Perform, Collect Data
- Analysis & Interpretation

# OSS Repository

- Pros
  - Lots of data
  - Large software projects
  - Free
- Cons
  - Too many data
  - Toy software projects (noise)
  - Heterogeneous (GitHub, GoogleCode, ...)

# How to fetch the data

- Push model: The repository sends data to the database
  - The conceptual schema has to be defined
  - Scalable infrastructure has to be deployed
  - Fast queries but limited to the conceptual schema
- Browse model: Visit the repository to get the required data
  - Flexible
  - Thin infrastructure
  - Low queries but very expressive

# Browse Model without Harmony



Are library migrations frequent ?



Repository 1



Repository 2



Repository 3



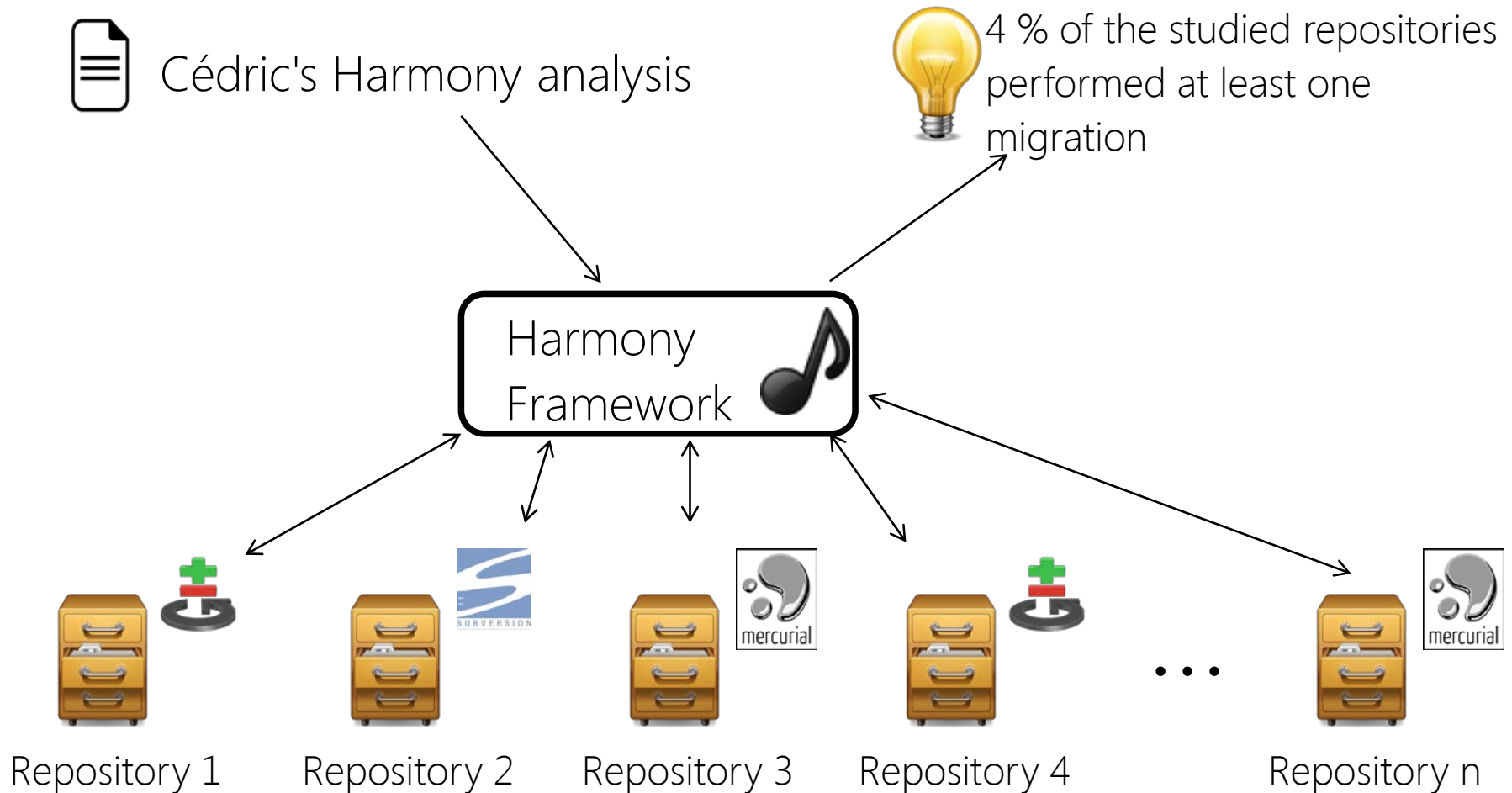
Repository 4

...



Repository n

# Browse model with Harmony

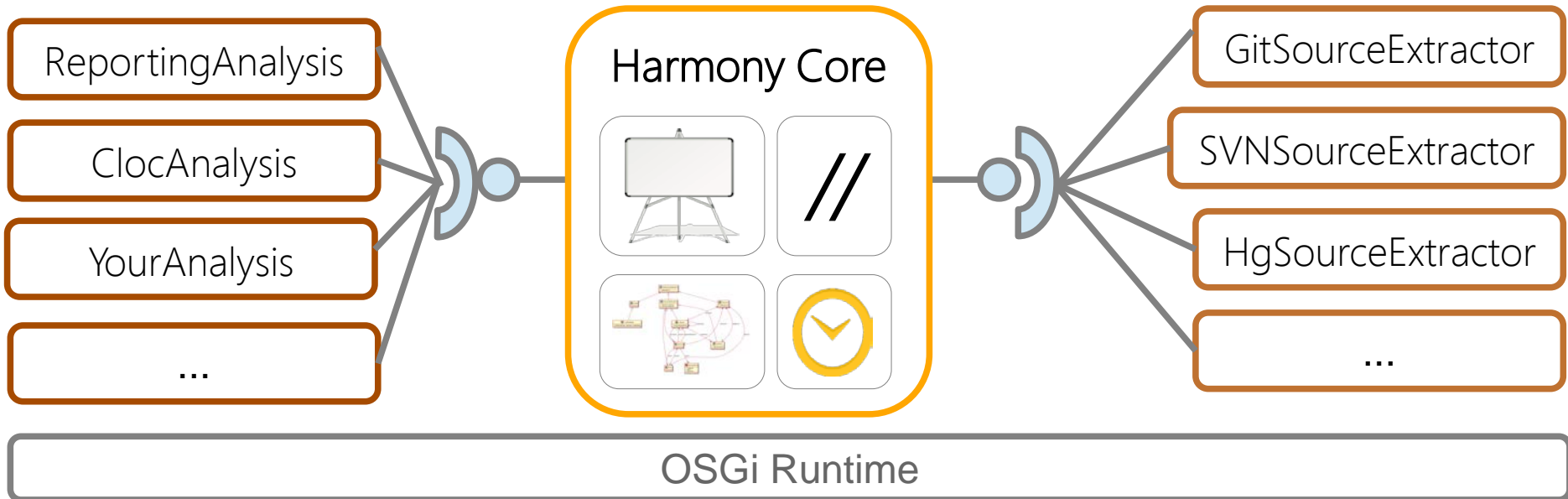


# Harmony Main Features

Ease the writing of browse model analysis

- Light abstract model
  - Source, Item, Event, Action, Authors
- Hook to repository
  - File f <- workspace.update(evt)
- Connectors to well-known repositories
  - Git, Mercurial, SVN, BugZilla
- Plugin Based (OSGi)

# Harmony Architecture





# Harmony @GoogleCode

Open and Free

- <http://harmony.googlecode.com>
- Equinox respository
  - OSGi version
- Doc repository
  - User Manual

Try and join !!!

# Demo

1. Run the dummy analysis
2. Change it to print names of authors
3. Change it to print names of authors of Java files
4. *Change it to print names of authors of Java files that import junit !*
5. Look at the workload analysis and run it
6. Use R to see if there is a correlation between Churn and touches

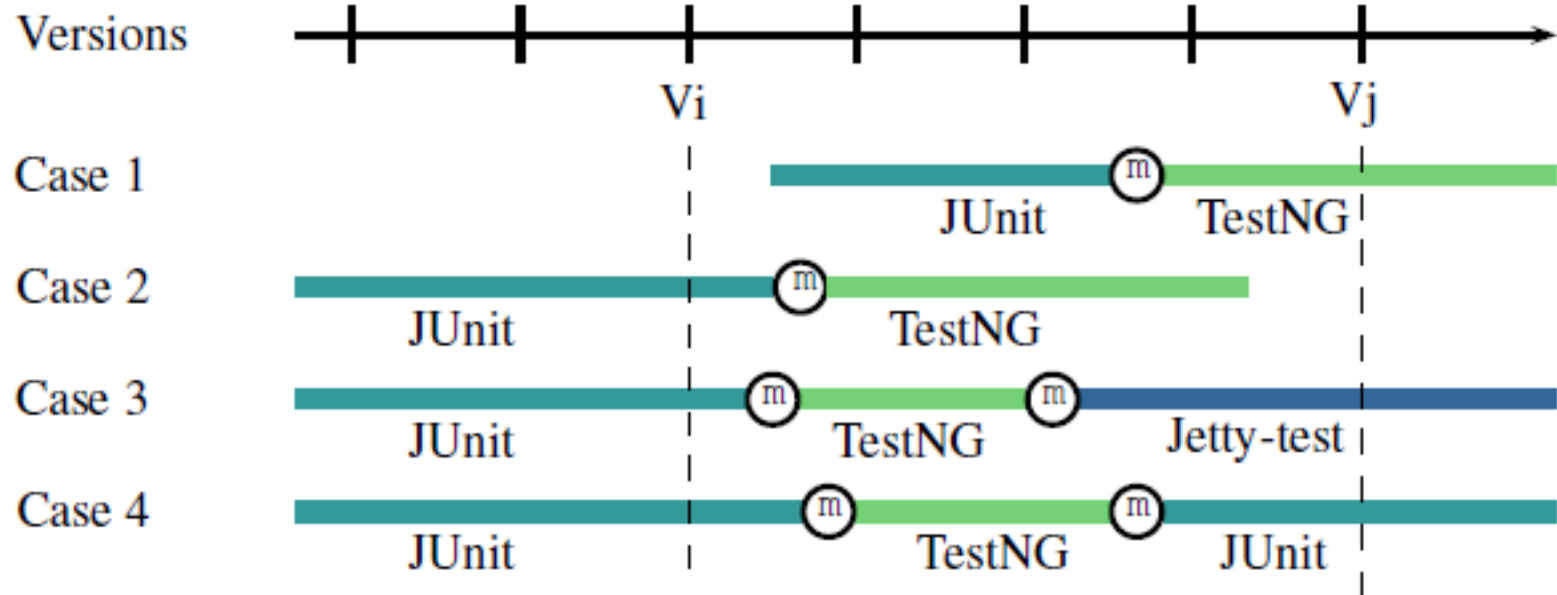
# Library Migration

- Library
  - An Id (Junit)
  - A set of exported symbols (assertTrue, assertEquals, ...)
- Program
  - An Id (MyApp)
  - A sequence of versions
  - A set of dependent libraries at each version
- Library migration
  - LibA was used by P at version i and was replaced by LibB afterward

# Research Questions

- Which software perform library migrations and how much of them?
- Which libraries are migrated and how many times?
- When migrations are performed?
- Why migrations are performed?
- How migrations are performed?

# Migration detection

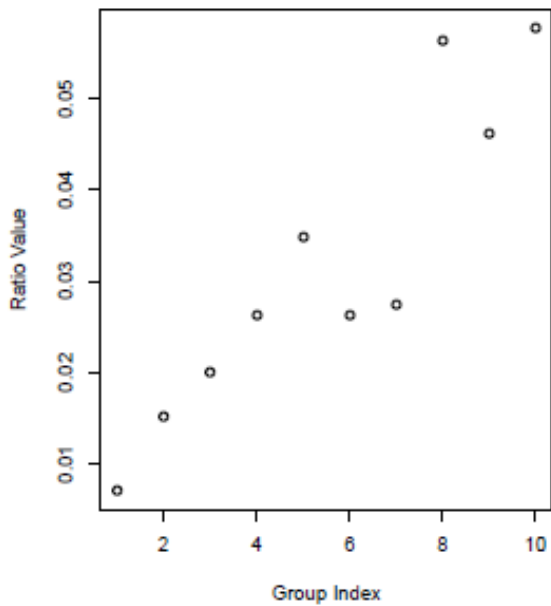


# Corpus

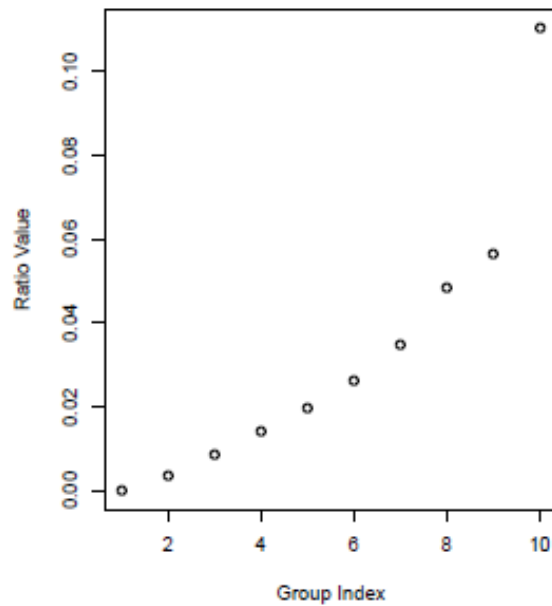
- 14,028 software projects
  - Random (GitHub, GoogleCode, SourceForge)
- 8,795 libraries
  - Maven

# Few migration (around 4%)

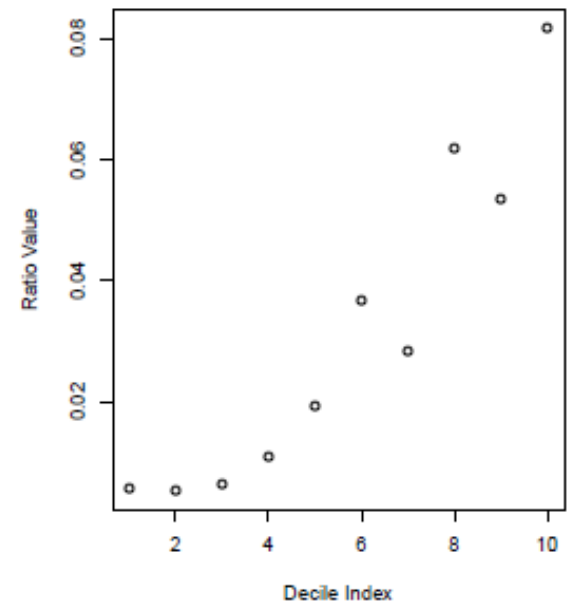
Ratio of Migrations for CLOC metric



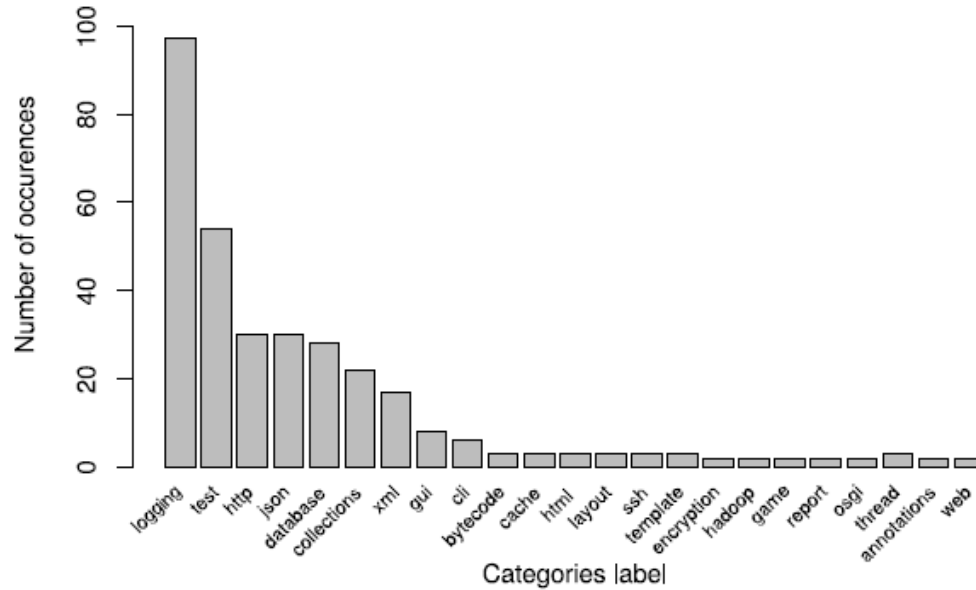
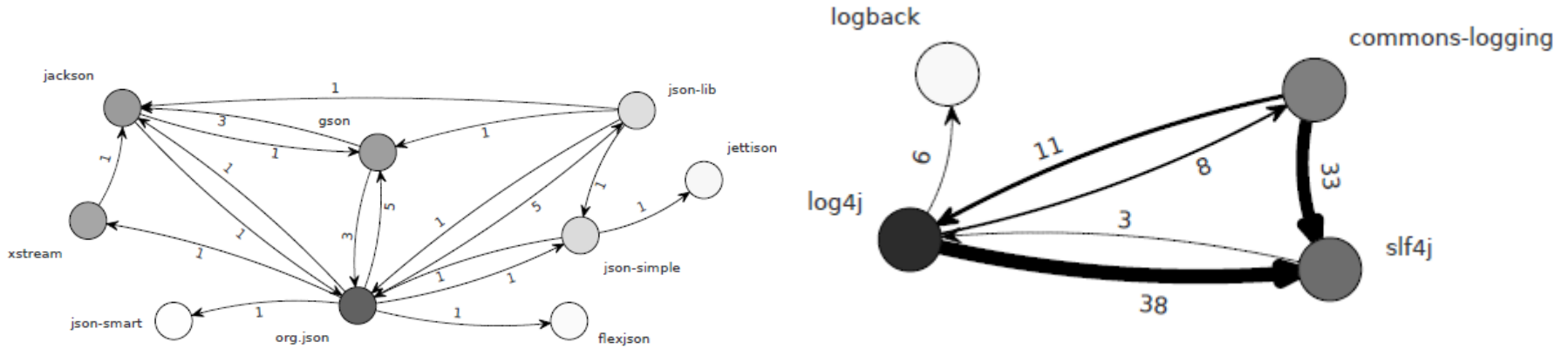
Ratio of Migrations for Commit metric



Ratio of Migrations for Duration metric

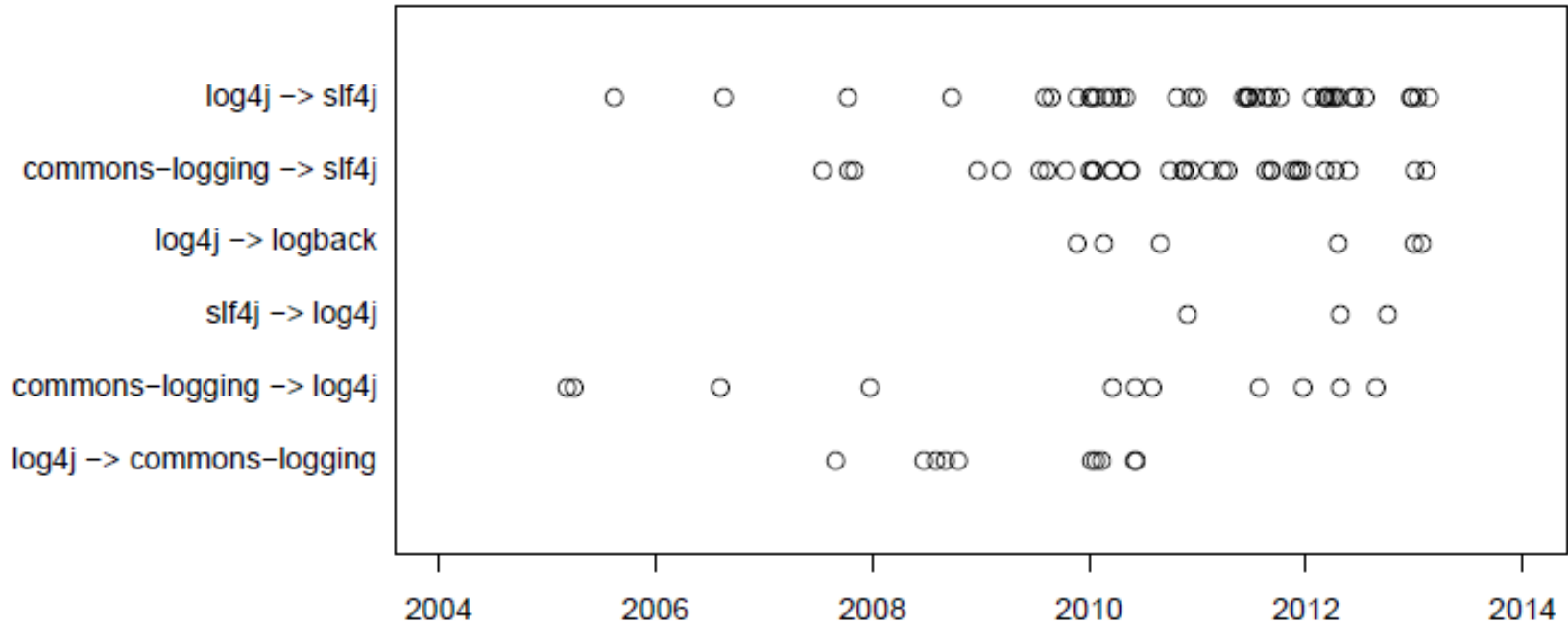


# Library Categories





# No tendency



# No generic reasons identified

Migration	Message	Reason
<i>commons-logging → slf4j</i>	<i>"replacing commons-logging with slf4j to help with osgi compliance per springsource recommendation."</i>	Configuration
<i>junit → testng</i>	<i>"use testng instead of junit which is a lot more configurable in test selection (and allows us to do a much better job a leaving the tree green even while developing tests that are known to fail)"</i>	Feature
<i>junit → testng</i>	<i>"convert tests to testng because we have groups here"</i>	Feature
<i>log4j → commons-logging</i>	<i>"switched from a log4j based logging api usage to commons-logging in order to allow enhanced logging techniques to be optional"</i>	Feature
<i>log4j → logback</i>	<i>"simple migration to logback since log4j is old"</i>	Outdated
<i>log4j → logback</i>	<i>"changed to logback to allow change of log levels"</i>	Feature
<i>org.json → gson</i>	<i>"replaced json.org package with gson for license compatibility"</i>	License
<i>json-lib → jackson</i>	<i>"use jackson json library (reduce dependencies)"</i>	Configuration
<i>json-lib → org.json</i>	<i>"demonstrate bug in json-lib"</i>	Bug
<i>org.json → json-smart</i>	<i>"change org.json library (don't be evil license) by json smart (apache license 2.0)"</i>	License
<i>commons-collections → lambdaj</i>	<i>"replaced commons-collections with lambdaj since it has a more modern search syntax"</i>	Feature
<i>gson → org.json</i>	<i>"attempting to get into maven central repo. requires moving from gson which isn't in central (except buggy version)"</i>	Configuration

Table VI. Results of the commit logs mining to identify developers motivations to migration

# Conclusion

Use repositories to perform large survey

- Heterogeneity issue
- Efficiency issue
- Storage issue

⇒ Harmony

Try it, adopt it, diffuse it!