REPERTOIRE
A Cross-System Porting Analysis Tool for Forked Software Projects

Baishakhi Ray, Christopher Wiley, Miryung Kim
The University of Texas at Austin

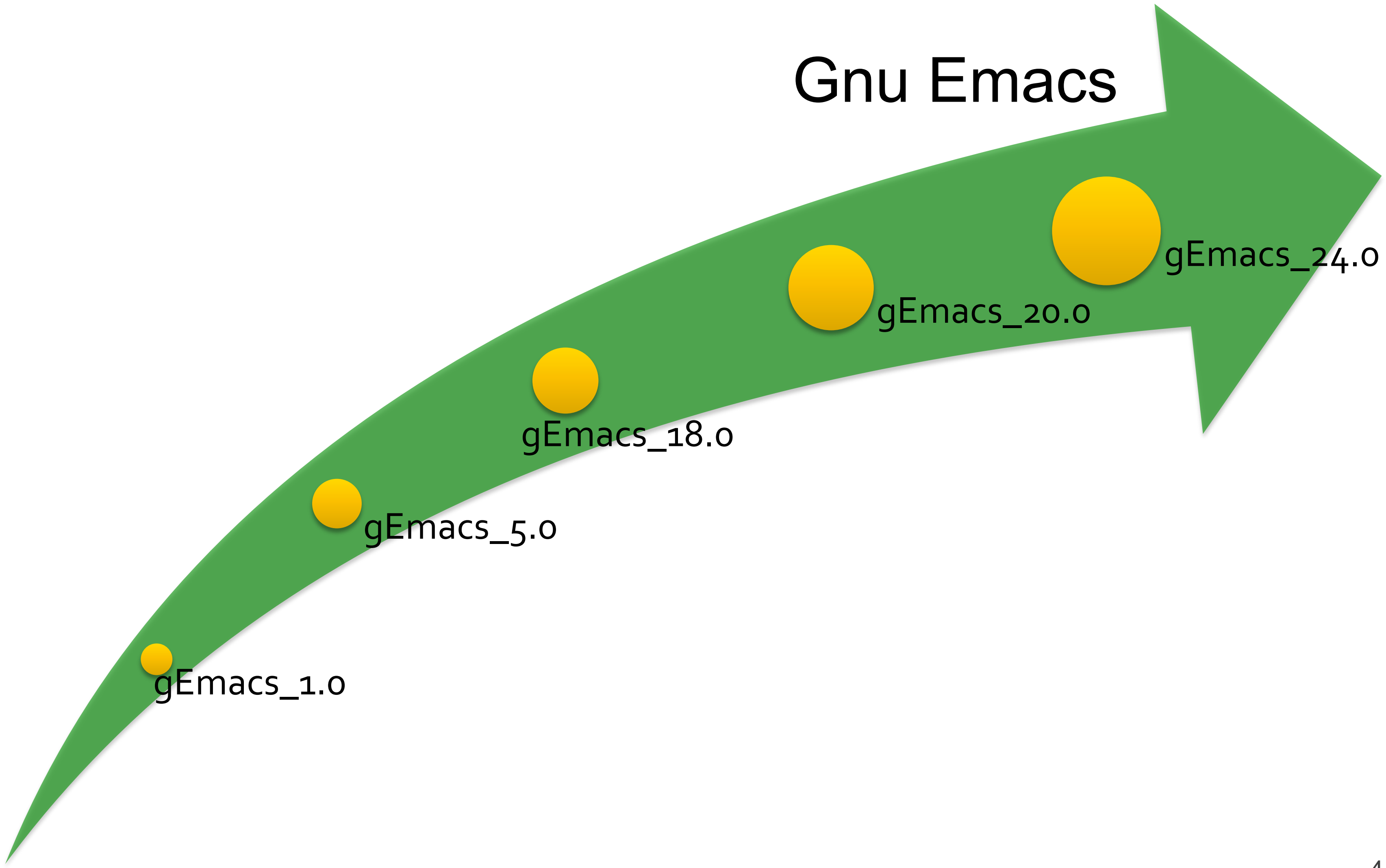
Motivation

- Software forking has become popular.
- Open source forked projects:
  - OpenBSD from NetBSD
  - XEmacs from GNU Emacs
- Proprietary forked projects:
  - Mac OS X from FreeBSD
  - EnterpriseDB from PostgreSQL

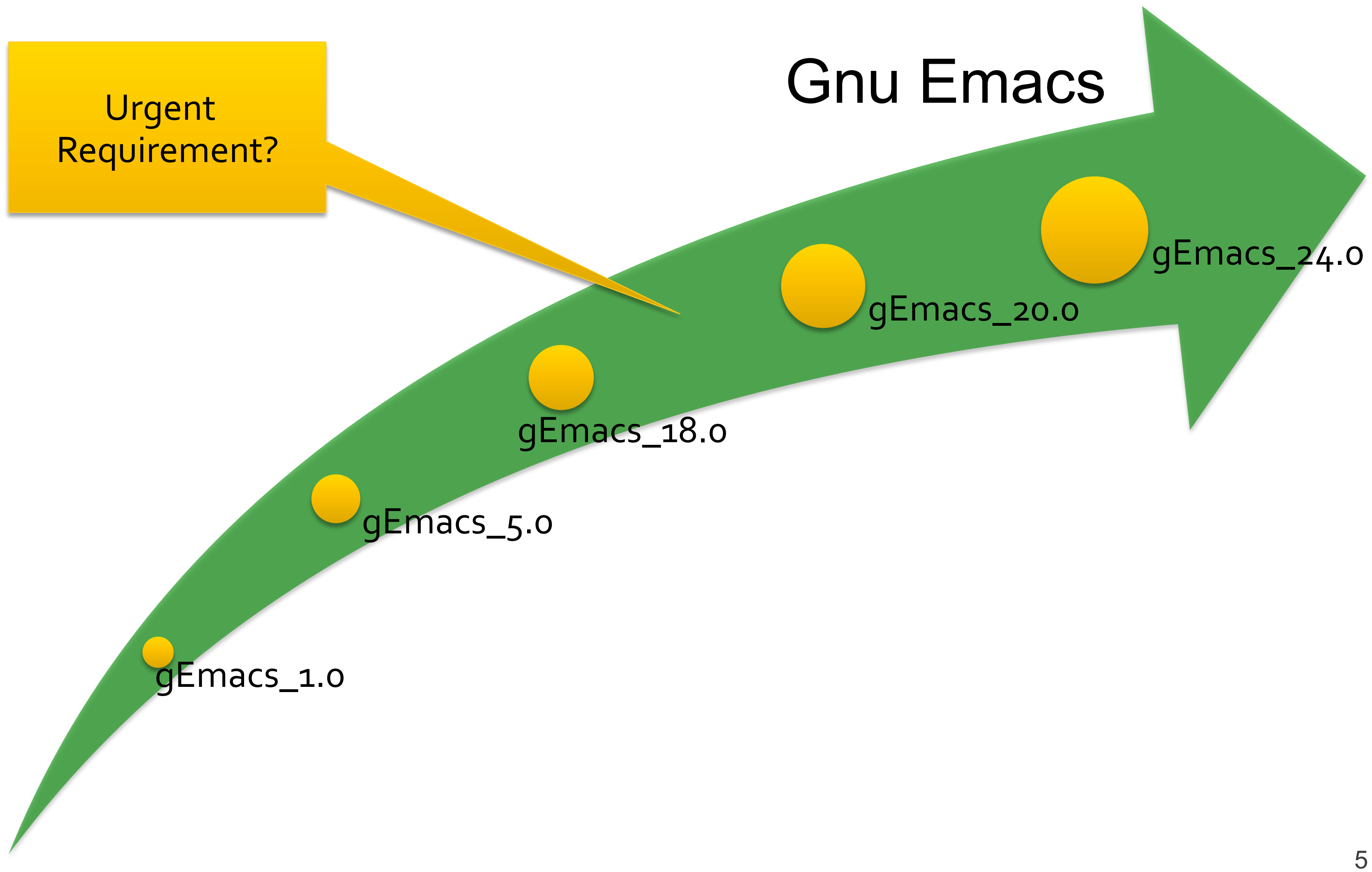
Motivation

- Developers port similar feature additions and bug-fixes across the projects.
- Repertoire analyzes the extent and characteristics of cross-system porting across forked projects.

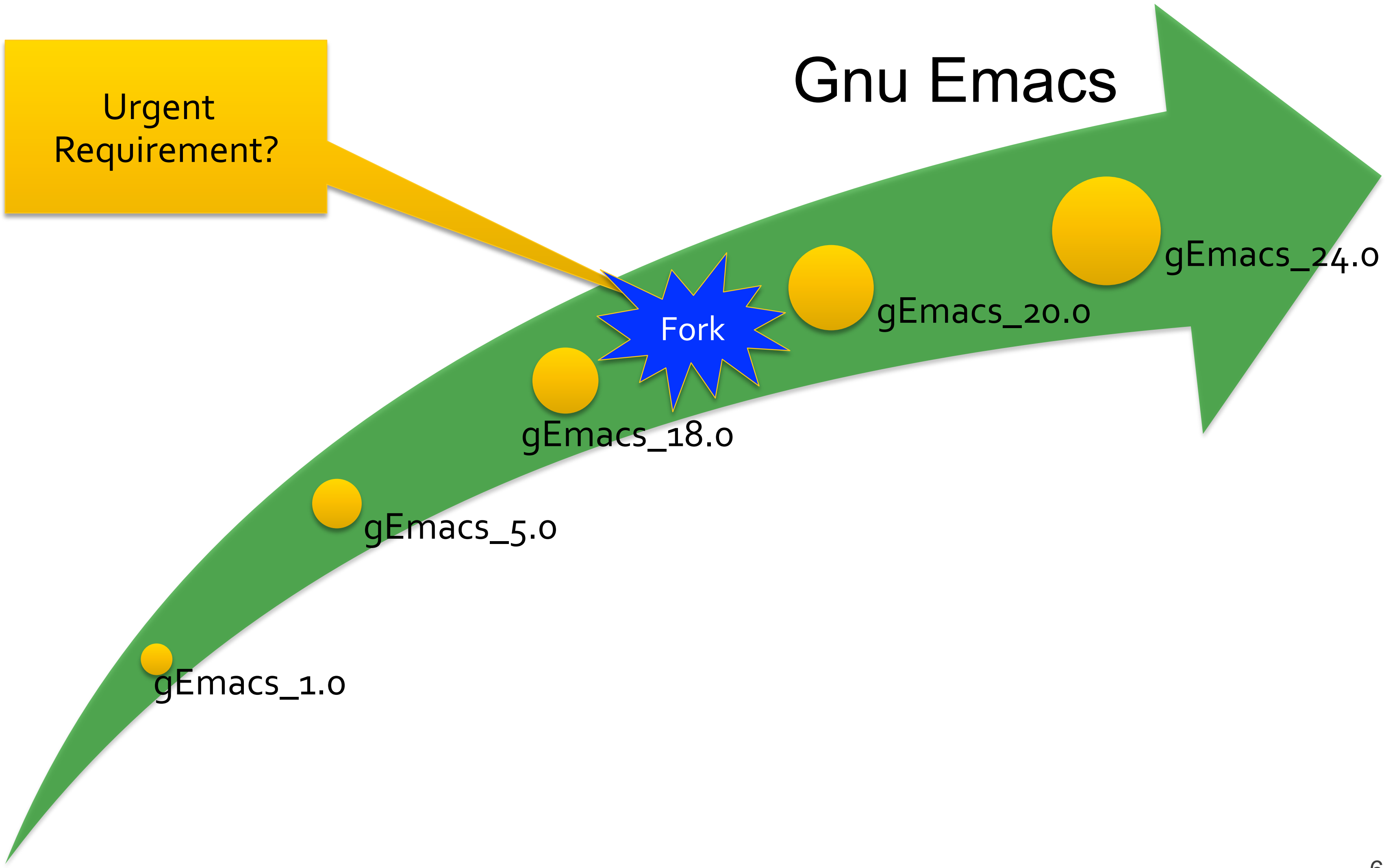
Example scenario: Gnu Emacs and XEmacs forking



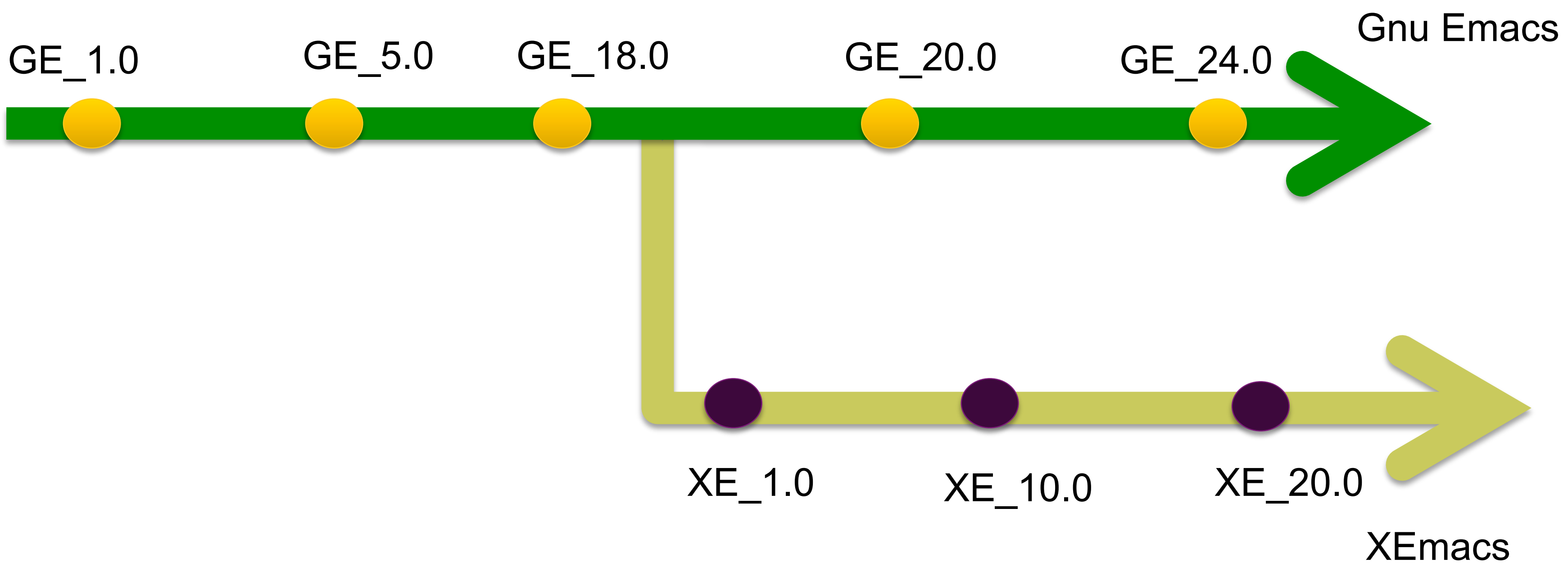
Example scenario: Gnu Emacs and XEmacs forking



Example scenario: Gnu Emacs and XEmacs forking

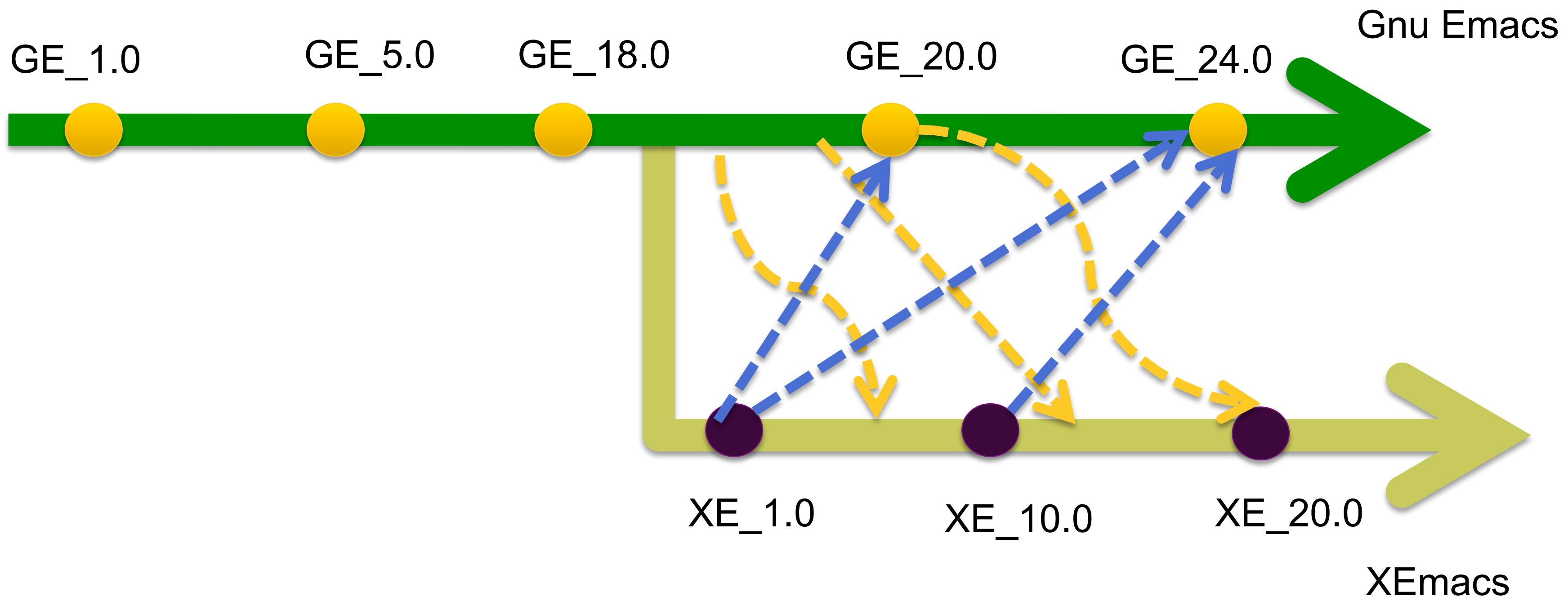


Example scenario: Gnu Emacs and XEmacs forking



Example scenario: Gnu Emacs and XEmacs forking

- Involves repetitive work to port bug-fixes and new feature.



Example scenario: Gnu Emacs and XEmacs forking

Gnu Emacs Commit Messages showing evidence of cross-system porting

```
Author: Stefan Monnier <monnier@iro.umontreal.ca>
Date:   Tue Jan 11 00:07:32 2011 -0500

   * lisp/progmodes/prolog.el: Fix up coding conventions and such.
   ...
   (prolog-emacs): Remove.  Use (featurep 'xemacs) instead.
  ...
```

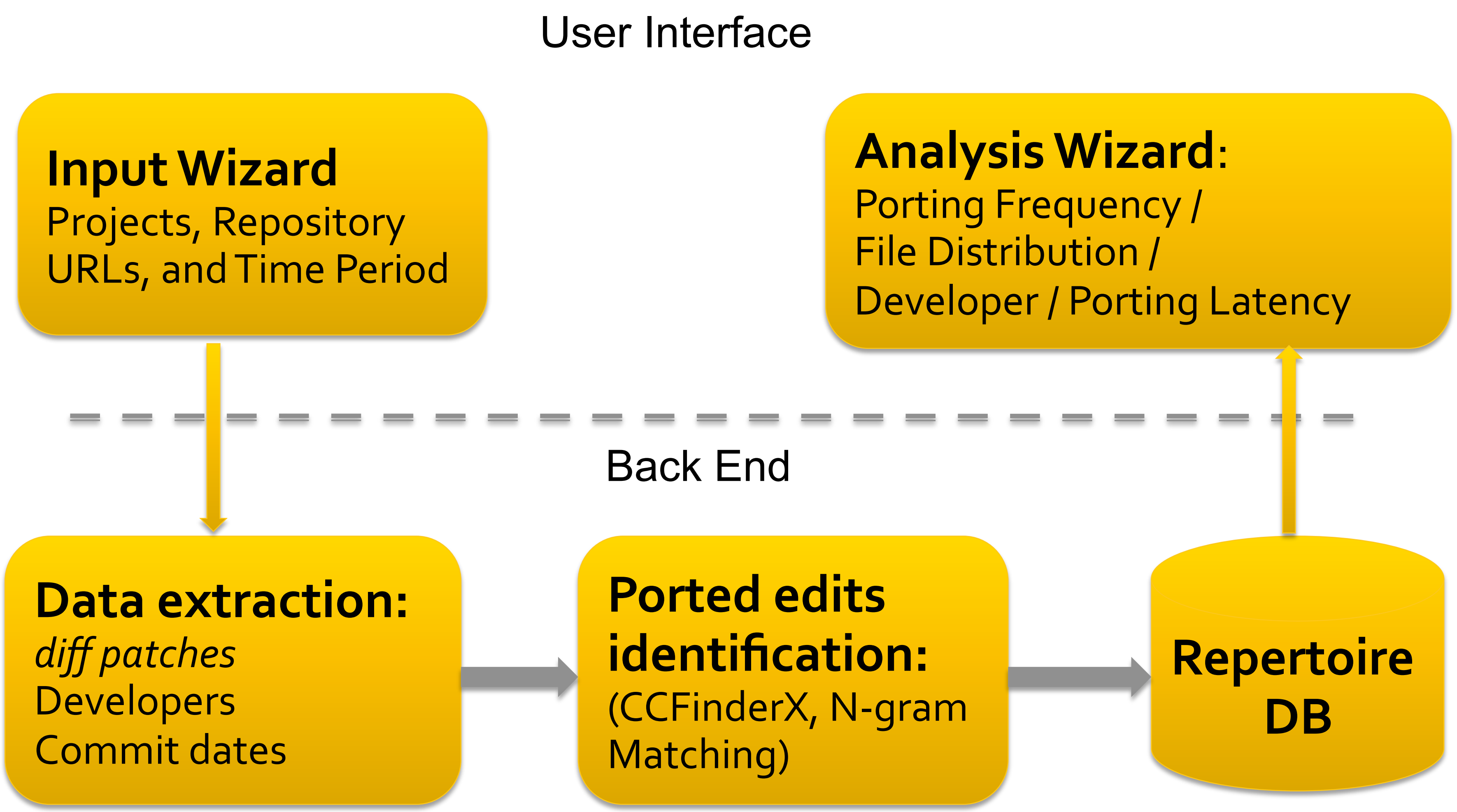
```
Author: Richard M. Stallman <rms@gnu.org>
Date:   Sun Jan 22 02:21:32 1995 +0000

   (term-if-emacs19, term-if-xemacs, term-ifnot-xemacs):  New macros
   to conditionalize at compile-time for different emacs versions.
```

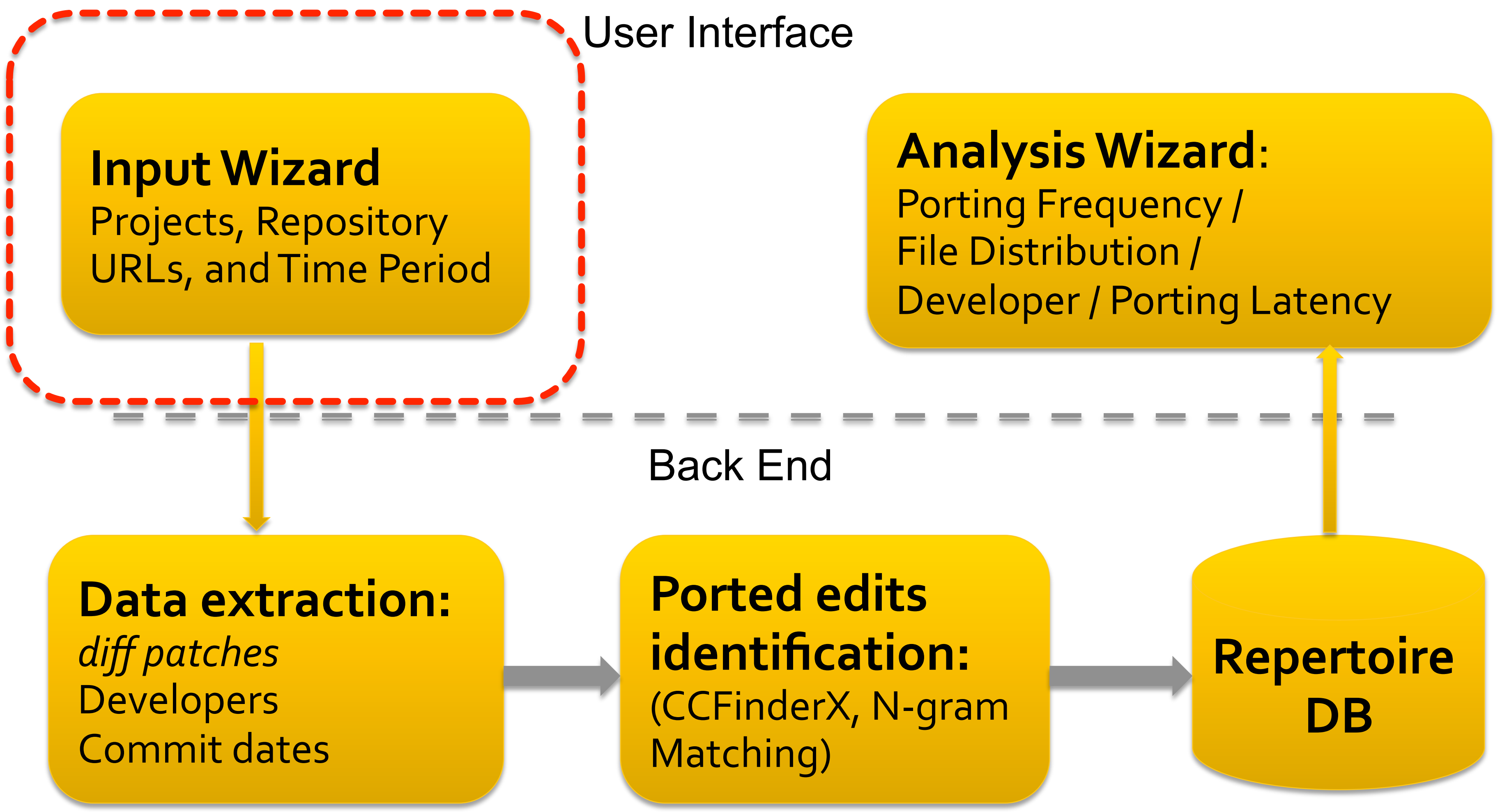
Gnu Emacs and Xemacs evolution from Jan, 2010 to Jan, 2012

	C/ Header files	KLOC	Authors	Number of Commits
Gnu Emacs	372	246	266	10525
XEmacs	496	282	11	754

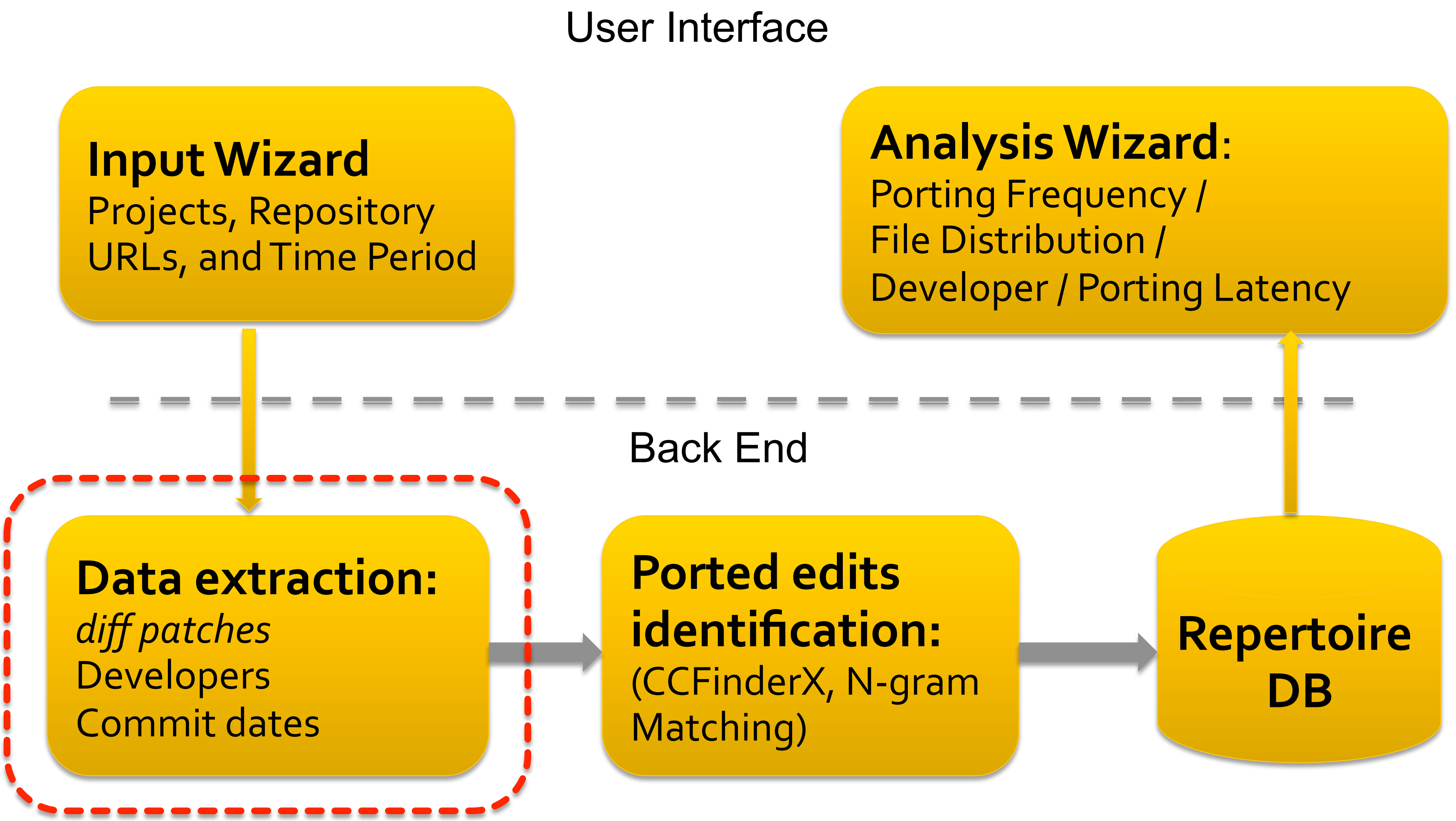
Repertoire Design



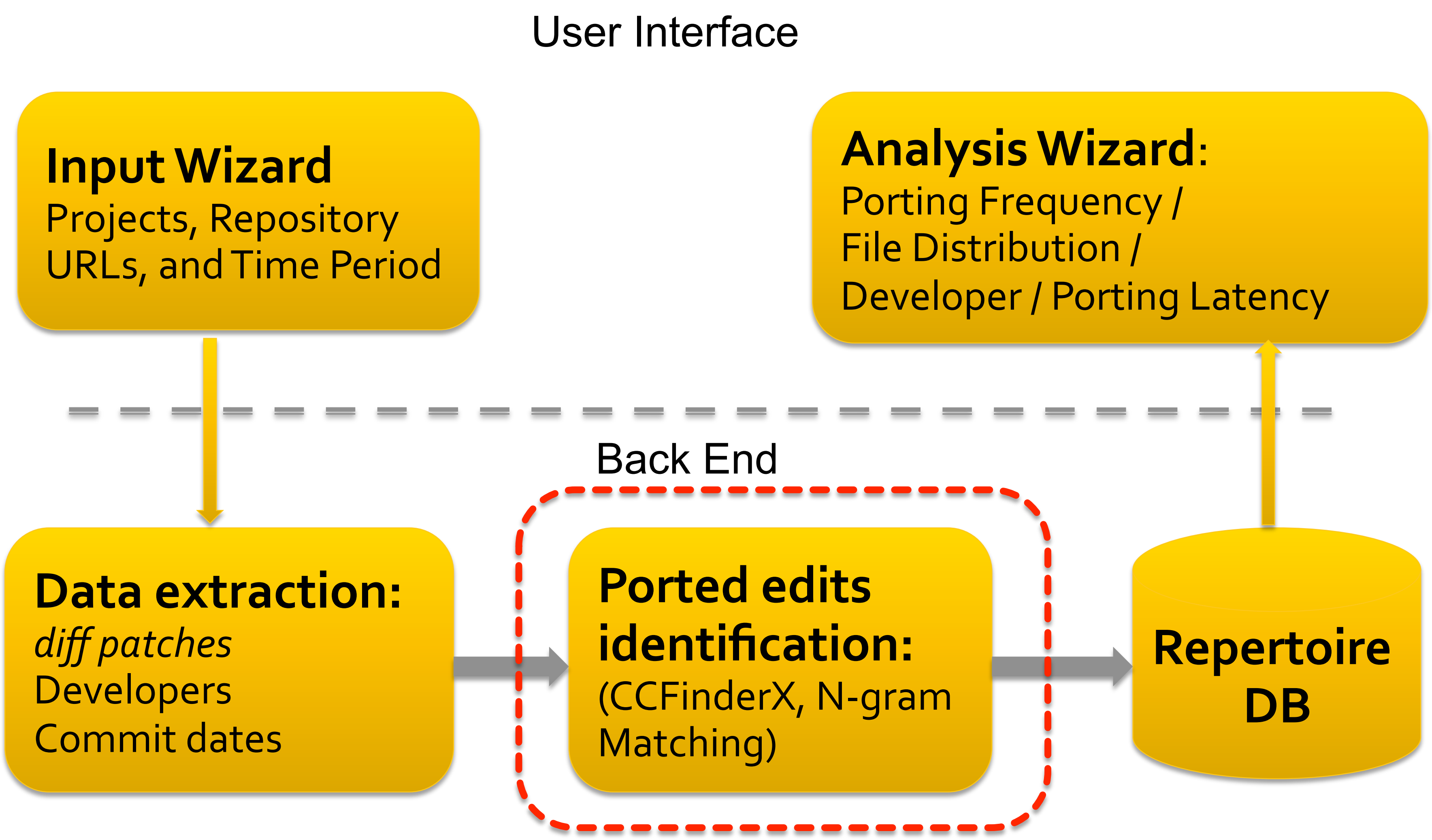
Repertoire Design



Repertoire Design


User Interface
Input Wizard
Projects, Repository URLs, and Time Period
Analysis Wizard:
Porting Frequency / File Distribution / Developer / Porting Latency
Back End
Data extraction:
diff patches
Developers
Commit dates
Ported edits identification:
(CCFinderX, N-gram Matching)
Repertoire DB

Repertoire Design


User Interface
Input Wizard
Projects, Repository URLs, and Time Period
Analysis Wizard:
Porting Frequency / File Distribution / Developer / Porting Latency
Back End
Data extraction:
diff patches
Developers
Commit dates
Ported edits identification:
(CCFinderX, N-gram Matching)
Repertoire DB

Repertoire Approach

- Input: two set of diff based program patches from the two input projects.
- Output: ported edits among the patches.
- Repertoire compares patches to identify contents and edit operations' similarity.

Step 1: Identify cloned regions using CCFinderX [Kamiya et al.]

Patch1 (Jan ’10)	Patch2 (Mar ‘10)
**** Old ****	**** Old ****
X1 for(i=0;i<MAX;i++){	Y1 for(j=0;j<MAX;j++) {
X2 - x = array[i]+x;	Y2 q = p + q;
X3 - y = foo(x);	Y3 - q = array[j]+p;
X4 - x = x-y;	Y4 - p = foo1(q);
X5 }	Y5 }
**** New ****	**** New ****
X6 for(i=0;i<MAX;i++) {	Y6 for(j=0;j<MAX;j++) {
X7 + y = x+y;	Y7 q = p + q;
X8 + x = array[i]+x;	Y8 + q = array[j] + q;
X9 + y = foo(x,y);	Y9 + p = foo1(p,q);
X10 }	Y10 }

Step 2: Match edit operations of cloned regions

Patch1 (Jan '10)	Patch2 (Mar '10)
**** Old ****	**** Old ****
X1 for(i=0;i<MAX;i++){	Y1 for(j=0;j<MAX;j++) {
X2 - x = array[i]+x;	Y2 q = p + q;
X3 - y = foo(x);	Y3 - q = array[j]+p;
X4 - x = x-y;	Y4 - p = foo1(q);
X5 }	Y5 }
**** New ****	**** New ****
X6 for(i=0;i<MAX;i++) {	Y6 for(j=0;j<MAX;j++) {
X7 + y = x+y;	Y7 q = p + q;
X8 + x = array[i]+x;	Y8 + q = array[j] + q;
X9 + y = foo(x,y);	Y9 + p = foo1(p,q);
X10 }	Y10 }

Step 2: Match edit operations of cloned regions

Patch1 (Jan ’10)	Patch2 (Mar ‘10)
**** Old ****	**** Old ****
X1 for(i=0;i<MAX;i++){	Y1 for(j=0;j<MAX;j++) {
X2 - x = array[i]+x;	Y2 q = p + q;
X3 - y = foo(x);	Y3 - q = array[j]+p;
X4 - x = x-y;	Y4 - p = foo1(q);
X5 }	Y5 }
**** New ****	**** New ****
X6 for(i=0;i<MAX;i++) {	Y6 for(j=0;j<MAX;j++) {
X7 + y = x+y;	Y7 q = p + q;
X8 + x = array[i]+x;	Y8 + q = array[j] + q;
X9 + y = foo(x,y);	Y9 + p = foo1(p,q);
X10 }	Y10 }

Ported edits

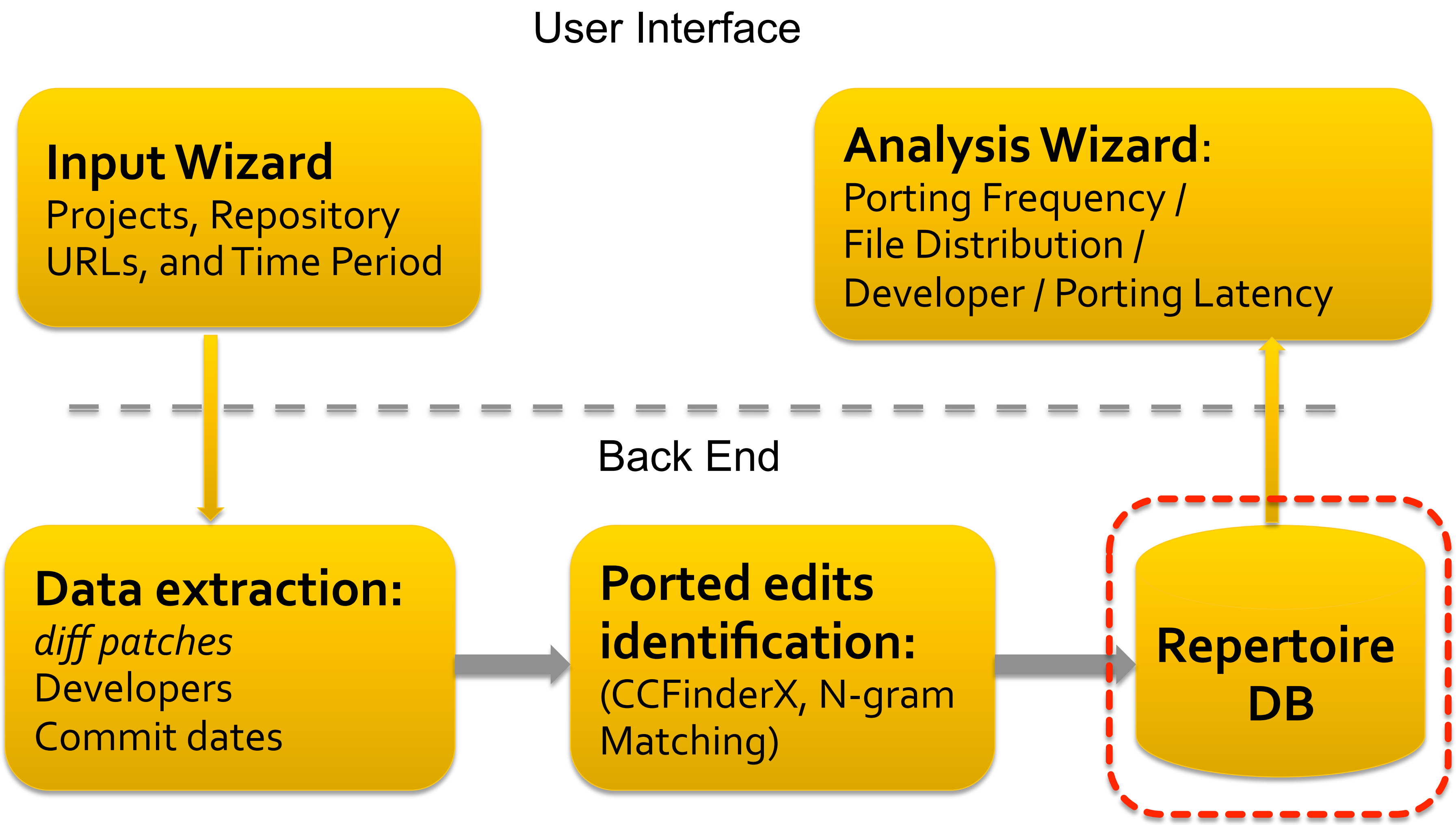
Step 3: Disambiguate source as destination of ported edit

Patch1 (Jan '10)	Patch2 (Mar '10)
**** Old ****	**** Old ****
X1 for(i=0;i<MAX;i++){	Y1 for(j=0;j<MAX;j++) {
X2 - x = array[i]+x;	Y2 q = p + q;
X3 - y = foo(x);	Y3 - q = array[j]+p;
X4 - x = x-y;	Y4 - p = foo1(q);
X5 }	Y5 }
**** New ****	**** New ****
X6 for(i=0;i<MAX;i++) {	Y6 for(j=0;j<MAX;j++) {
X7 + y = x+y;	Y7 q = p + q;
X8 + x = array[i]+x;	Y8 + q = array[j] + q;
X9 + y = foo(x,y);	Y9 + p = foo1(p,q);
X10 }	Y10 }

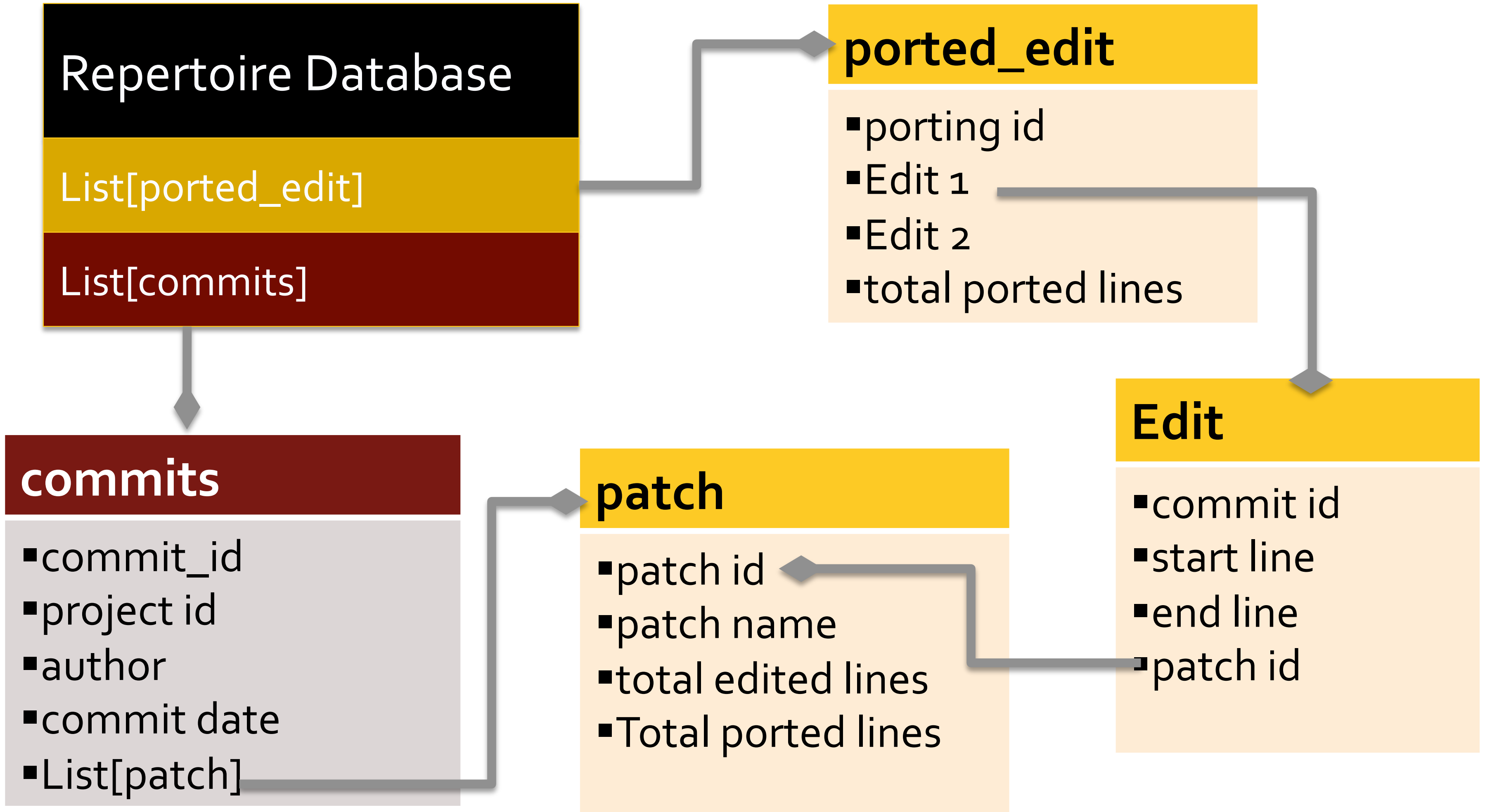
Accuracy Measurement

- From our empirical study of cross-system porting in the BSD product family, we find Repertoire's Precision: 94%, Recall: 84%.

Repertoire Design


User Interface
Input Wizard
Projects, Repository URLs, and Time Period
Analysis Wizard:
Porting Frequency / File Distribution / Developer / Porting Latency
Back End
Data extraction:
diff patches
Developers
Commit dates
Ported edits identification:
(CCFinderX, N-gram Matching)
Repertoire DB

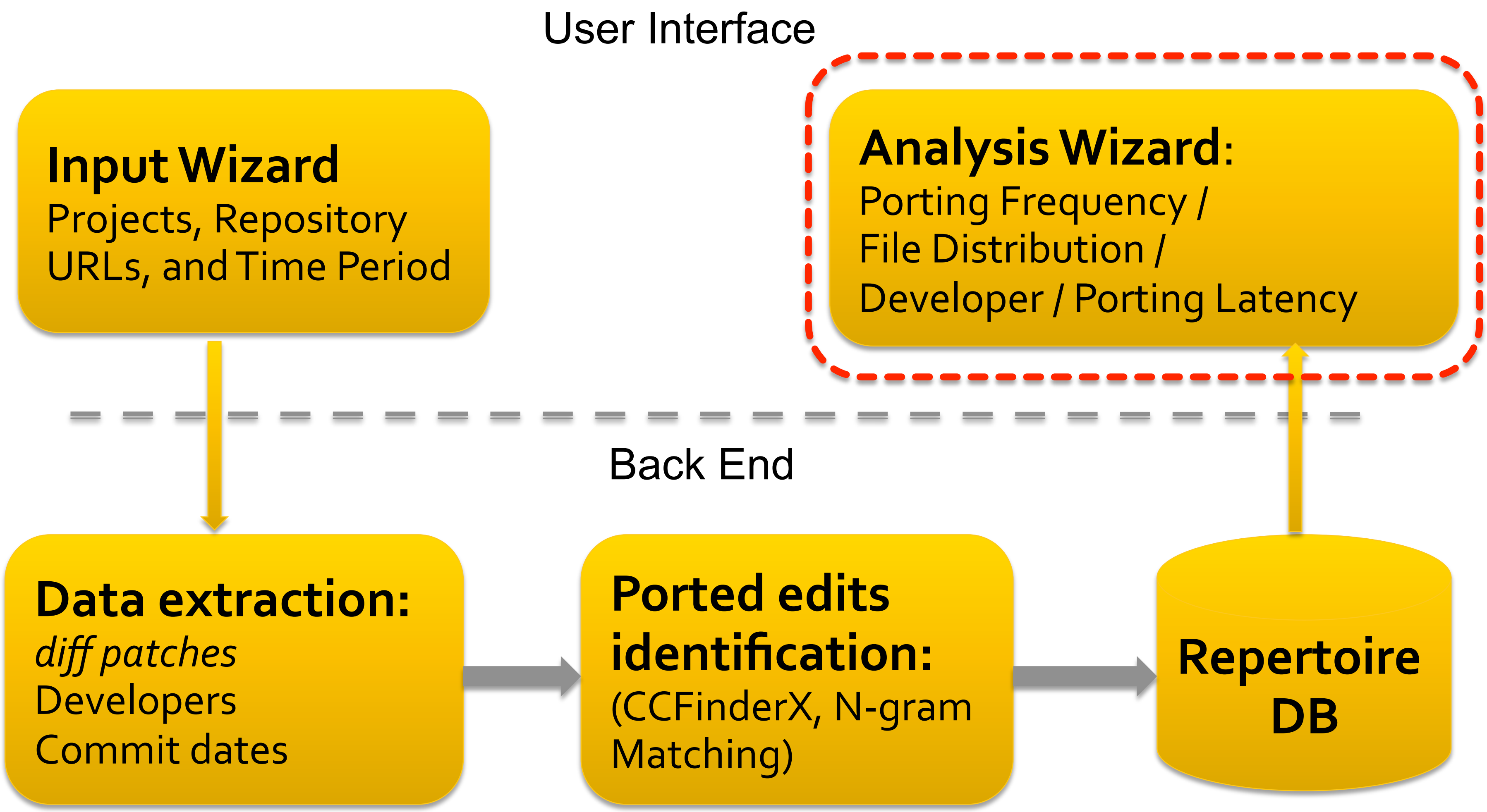
Repertoire Database



Populated by "Data Extraction" Module

Populated by "Ported edits identification" Module

Repertoire Design


User Interface
Input Wizard
Projects, Repository URLs, and Time Period
Analysis Wizard:
Porting Frequency / File Distribution / Developer / Porting Latency
Back End
Data extraction:
diff patches
Developers
Commit dates
Ported edits identification:
(CCFinderX, N-gram Matching)
Repertoire DB

Analysis Wizard

- Porting Frequency View
  - How much duplicate work is taking place to maintain the forked project?
- File Distribution View
  - Is porting mostly concentrated to certain files?
- Developer Distribution View
  - Which developers primarily port edits from one project to another?
- Porting Latency View
  - How long it takes a patch to port from one project to another?

Summary

REPERTOIRE helps to monitor cross system porting.

Managers and product architects can make informed decision about how to manage a product family.

Summary

- Repertoire can be downloaded from :

  http://dolphin.ece.utexas.edu/Repertoire.html

- A Case Study of Cross-System Porting in Forked Software Projects, Baishakhi Ray, Miryung Kim, **FSE** '12
  - Presentation: 8:30 am on Thursday

Acknowledgment

- This work was in part supported by National Science Foundation under the grants CCF-1117902, CCF-1149391, and CCF-1043810 and by Microsoft SEIF award.

REPERTOIRE
A Cross-System Porting Analysis Tool for Forked Software Projects

Baishakhi Ray, Christopher Wiley, Miryung Kim
The University of Texas at Austin