

OPAM

A Package Manager for OCaml

Frédéric Tuong
Fabrice Le Fessant
Thomas Gazagnaire

OCaml Users and Developers workshop

14/09/2012 Copenhagen, Denmark.

OPAM

- Funded by Jane-Street and EU grants
- Project started in April 2012
- ~10k loc, ~1k commits, 9 contributors
- GPL: <https://github.com/OCamlPro/opam>
- packages: ~200 packages, 10 contributors
 - <http://opam.ocamlpro.com/>
 - <https://github.com/OCamlPro/opam-repository>
- version 1.0 soon (before 2013)

Context

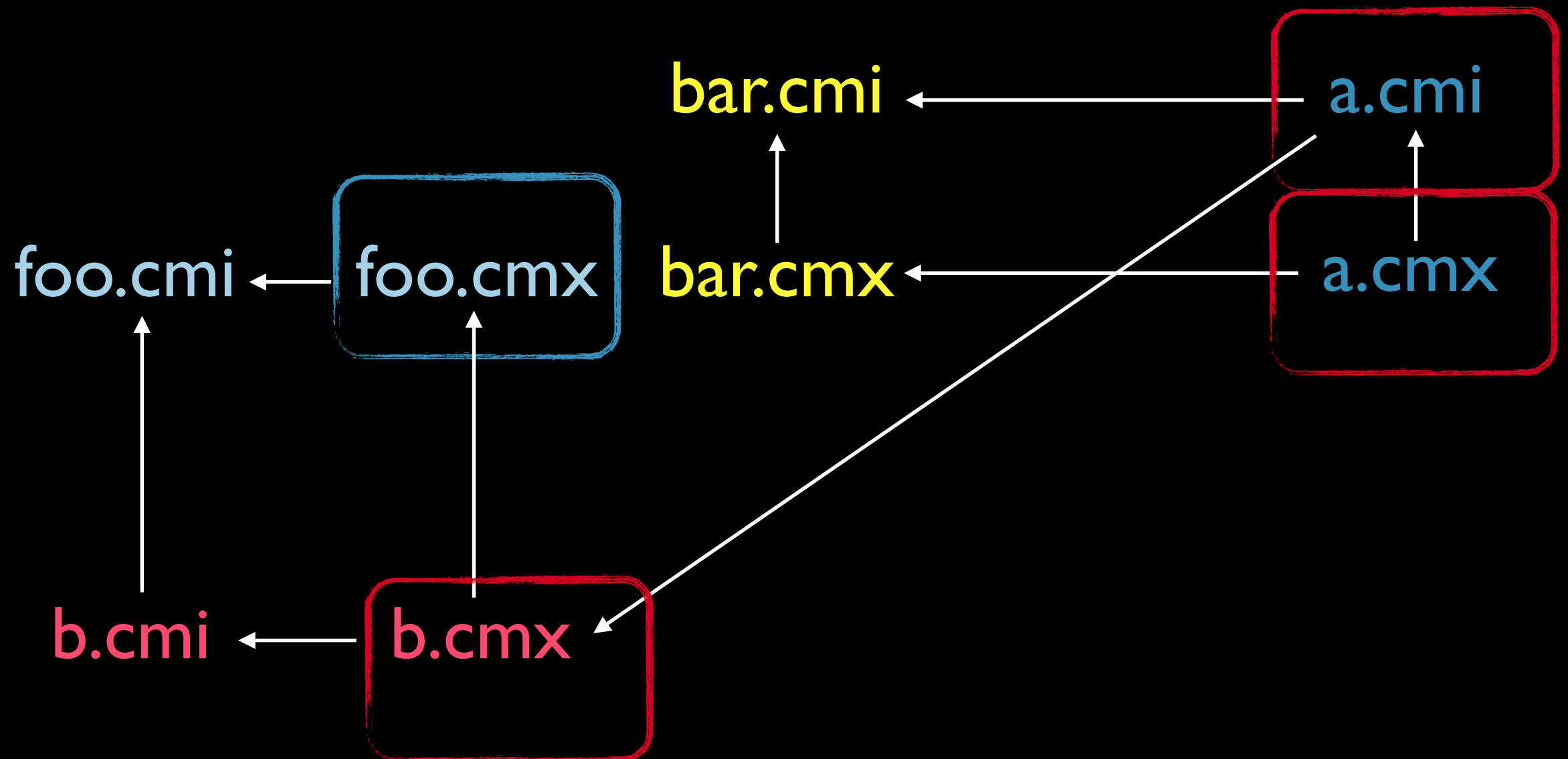
Compilation Units

- Tied to a compiler version
 - magic number
- Strong dependency relation
 - checksums computed at compile time
 - checked at link-time

➔ More complex than most of the other languages!

Context

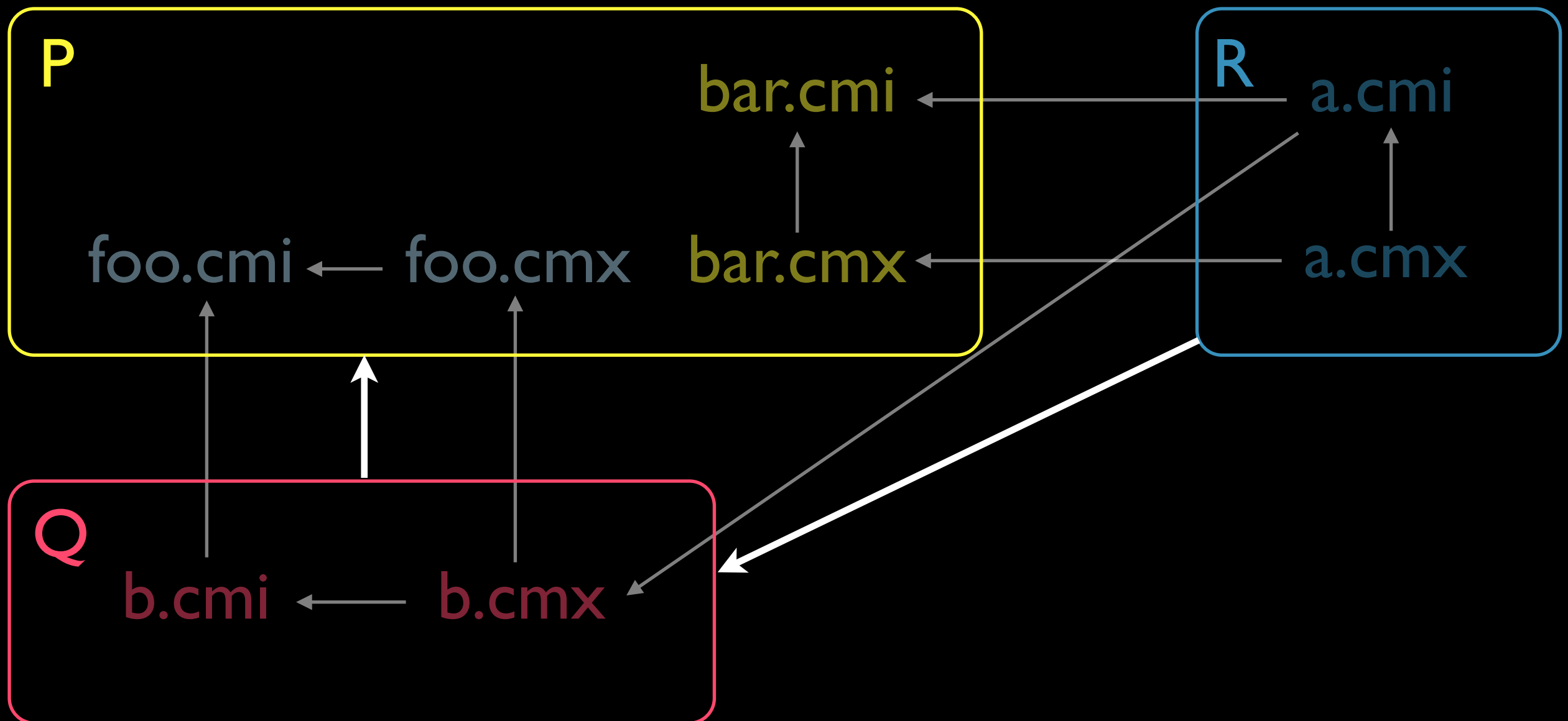
Compilation Units



Context

Packages

over-approximation of the dependency relation
tied to a given compiler version
no automatic checks at this level => easy to break



Summary

1. Manage Dependencies

2. Manage Repositories

3. Manage Compilers

4. Develop with OPAM

Manage Dependencies

Dependencies

▶ How ?

- **GODI**: godiva(outside the packages)
- **OASIS**: _oasis (in the packages)
- **OPAM**: opam (outside the packages)

▶ What ?

- **GODI**: limited version constraints
- **OASIS**: only positive constraints
- **OPAM**: support of CNF formulas (conflicts, alternatives)


```
opam-version: "1"
maintainer: "anil@recoil.org"
build: [
  [ "./configure"
    "--%{async:enabled}%-async"
    "--%{lwt:enabled}%-lwt" ]
  ["make" "all"]
  ["make" "install"]
]
remove: [
  ["ocamlfind" "remove" "cohttp"]
]
depends: [
  ("lwt" | "async")
  "ocamlfind" "re" "uri" "ounit" ]
depopts: [ "mirage-net" ]
```

```
$ opam install cohttp
```

Optional dependencies

```
$ opam install lwt cohttp  
$ opam instal [...]  
$ opam install async
```

1. uninstall cohttp
2. install async
3. install cohttp **with the right configure options**
4. **recompile** any packages depending on lwt

Manage Repositories

Repositories

▶ How does it work ?

- **GODI**: one central repository
- **OASIS/ODB**: one central repository + flexible package files
- **OPAM**: flexible remote repositories

▶ How to contribute ?

- **GODI**: commit changes in the svn
- **OASIS**: upload new packages on web-form
- **OPAM**: create pull requests on github

Repositories

OPAM: 3 kinds of repositories: HTTP, local and git

http://server

/local/path

git://repo.git

A.1 B.1

A.2

C.1

```
$ opam remote -add foo http://server/
```

```
$ opam remote -add bar /local/path
```

```
$ opam remote -add gno git://repo.git
```

```
$ opam list
```

```
A -- description of A
```

```
B -- description of B
```

```
C -- description of C
```

Repositories

- Setting-up internal repositories for organizations
- Caching archives to not depend on network access
- Testing new packages
- ...

Manage Compilers

Managing Compilers

► Why ?

- new compiler => recompile the packages
- support of patched version of ocaml

► How ?

- **GODI**: full control on the compiler + rely on upstream
- **OASIS**: use pre-installed compiler
- **barbra**: one tree per-install
- **OPAM**: use pre-installed compiler + compiler descriptions in remote repositories

Manage Compilers

```
opam-version: "1"  
name: "4.00.0"  
src: "http://caml.inria.fr/pub/distrib/ocaml-4.00/ocaml-4.00.0.tar.gz"  
make: [ "world" "opt" "opt.opt" ]  
packages : [ "base-unix" "base-bigarray" "base-threads" ]  
env: [  
  [ CAML_LD_LIBRARY_PATH += "%{lib}%/stublibs" ]  
]
```

```
$ opam switch 4.00.0
```

```
$ opam switch -list
```

Develop with OPAM

Version pinning

```
$ opam pin react /local/path  
$ opam upgrade react  
  [do some changes]  
$ opam update  
$ opam upgrade react
```

The following actions will be performed:

- recompile react.0.9.4
- recompile lwt.2.4.1
- recompile zed.1.2
- recompile lambda-term.1.2
- recompile utop.1.2.1

NEXT ?

- whole-tree compiler flags (#5723)
- cross-compiling
 - windows
 - arm
 - ios

Questions ?

```
$ opam install opam
```