

# OCaml: State of the Platform 2014

**Anil Madhavapeddy (speaker)**

Amir Chaudhry, Jeremie Diminio, Thomas Gazagnaire,  
Louis Gesbert, Thomas Leonard, David Sheets, Mark  
Shinwell, Leo White, Jeremy Yallop

with a vast amount of help from INRIA, OCamlPro, Jane  
Street, Citrix, and the wider OCaml community

# Recap: a Platform

- **Tooling** that works together beyond just a language, into the full dev lifecycle.
- **Quantitative** metrics to judge if we are healthy or not.
- **Agility** to judge the impact of language changes quickly to keep moving.

**Ultimate Goal: grow a sustainable  
open-source community**

# OPAM Releases: 2013

- OPAM 1.0 released in March 2013
- OPAM 1.1 final released October 2013
  - Solid bug fixing and improvement released based on *lots* of feedback.
  - Over 100 contributors, 500+ packages, 1500+ unique versions.
  - Migrating to [opam.ocaml.org](http://opam.ocaml.org) (CC0) as a community-maintained effort.

# OPAM Releases: 2014

- **Feb 2014: OPAM 1.1.1** (*RM: Thomas Gazagnaire*)

Bugfixes, solver stability and better interface to superior external solvers from Mancoosi, library interfaces for repository tools.

- **July 2014: OPAM 1.1.2** (*RM: Louis Gesbert*)

Switch to Makefiles for easier integration into binary packages for OS distros.

- **August 2014: OPAM 1.2.0** (*RM: Louis Gesbert*)

The "Platform Release"

Since	1.1.0 (2013-11-07):	318 PR merged,	331 issues closed,	367 new issues
	1.0.0 (2013-03-14):	133 PR merged,	273 issues closed,	305 new issues
Before	1.0.0 (2012-02-17):	116 PR merged,	357 issues closed,	402 new issues

# OPAM 1.2 : Simple Workflow

- Solver errors are explained in plain English rather than boolean formulae.

```
$ opam install mirage-www=0.3.0
```

```
The following dependencies couldn't be met:
```

```
- mirage-www -> cstruct < 0.6.0
```

```
- mirage-www -> mirage-fs >= 0.4.0 -> cstruct >= 0.6.0
```

```
Your request can't be satisfied:
```

```
- Conflicting version constraints for cstruct
```

# OPAM 1.2 : Query Interface

- More expressive queries (reverse dependencies and recursive).

```
$ opam list --depends-on cow --rec
```

```
# Available packages recursively depending on cow.0.10.0 for 4.01.0:  
cowabloga      0.0.7   Simple static blogging support.  
iocaml         0.4.4   A webserver for iocaml-kernel and iocamljs-kernel.  
mirage-www     1.2.0   Mirage website (written in Mirage)  
opam2web       1.3.1 (pinned) A tool to generate a website from an OPAM  
repository  
opium          0.9.1   Sinatra like web toolkit based on Async + Cohttp  
stone          0.3.2   Simple static website generator, useful for a  
portfolio or documentation pages
```

# OPAM 1.2 : Features

- Clone the source code and repo file for any OPAM package.

```
$ opam source cow
```

```
Downloading archive of cow.0.10.0...
```

```
[...]
```

```
$ cd cow.0.10.0
```

```
$ make
```

```
$ opam show cow --raw
```

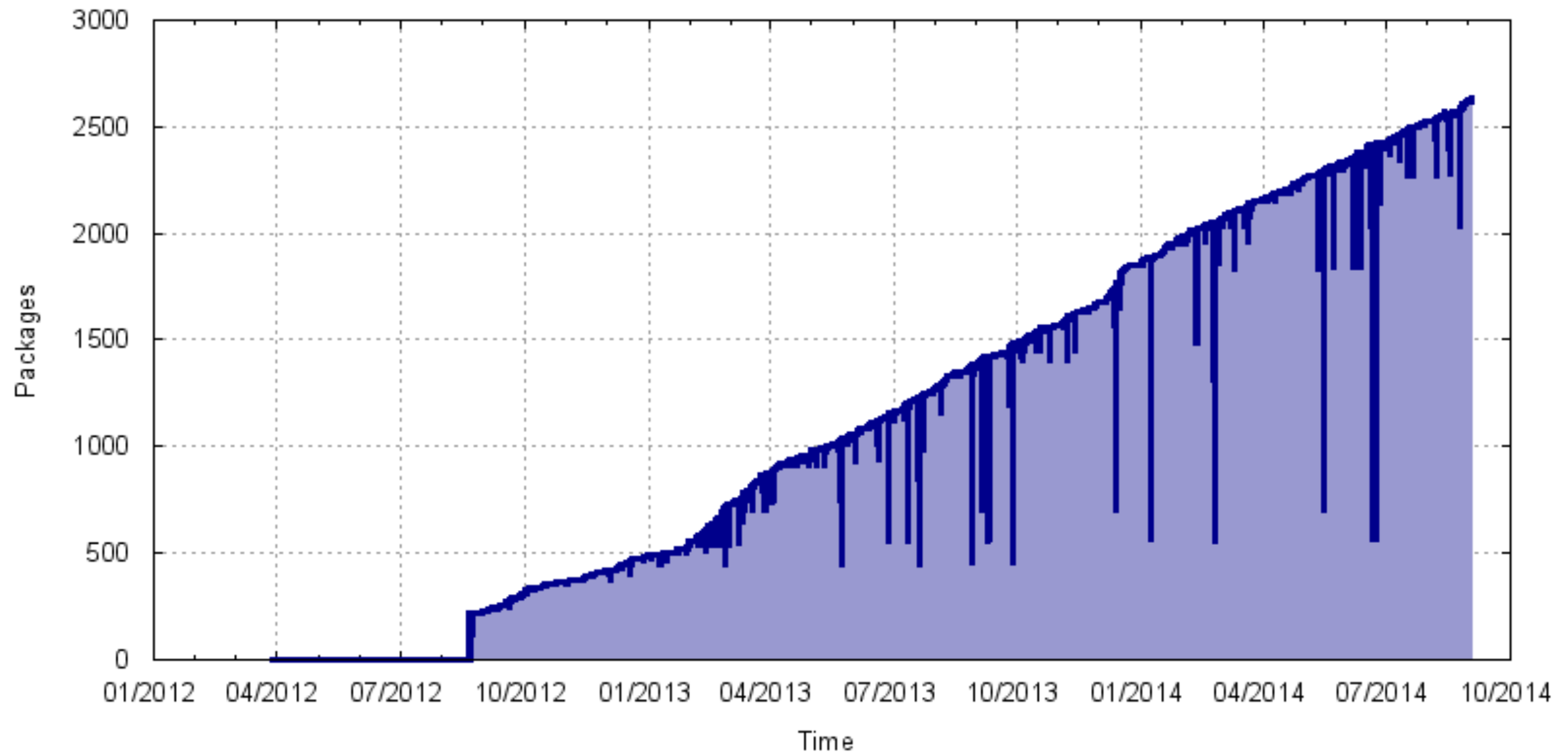
```
opam-version: "1"
```

```
name: "cow"
```

```
version: "0.10.0"
```

```
[...]
```

# OPAM 1.2 : Total Packages



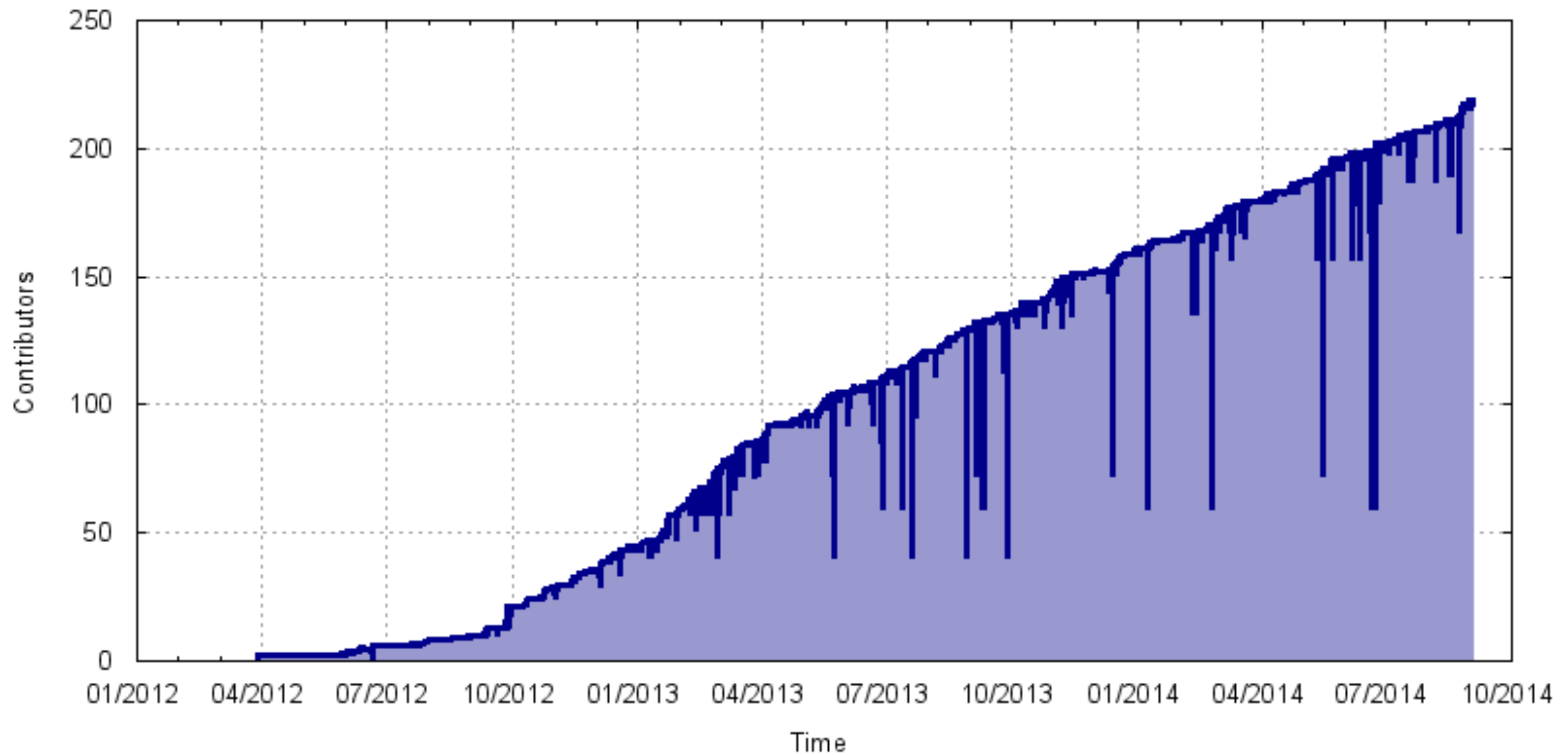
Since 1.1.0 (2013-11-07): 1178 merges, 2981 total

1.0.0 (2013-03-14): 660 merges, 1775 total

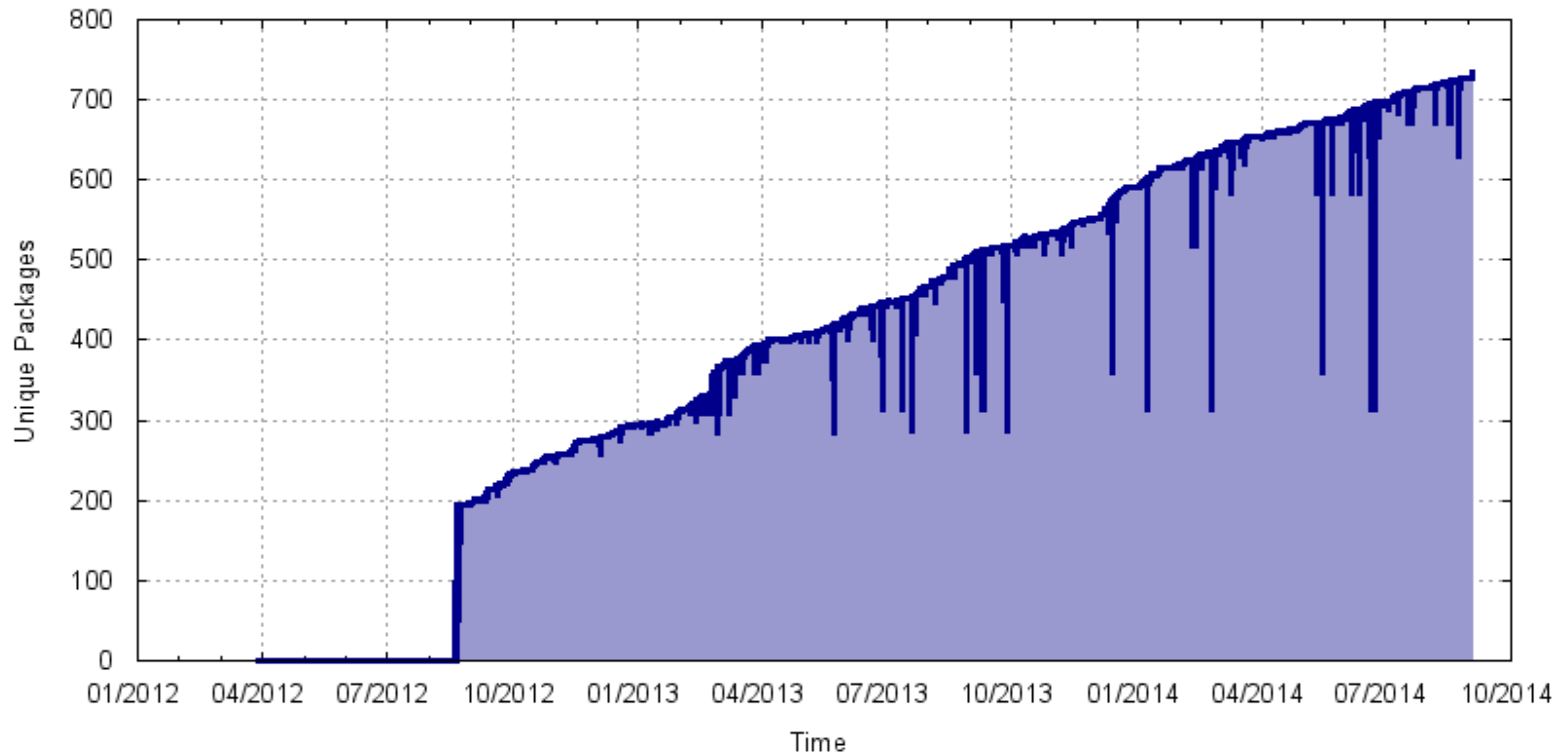
Before 1.0.0 (2012-02-17): 547 merges, 1762 total



# OPAM 1.2 : Contributors



# OPAM 1.2 : Unique Packages



# OPAM 1.2 : New Workflow

- Let's build a new package for a brand new ocp-reloc package.

```
$ cd ocp-reloc  
$ opam pin add ocp-reloc .
```

<http://opam.ocaml.org/blog/opam-1-2-pin/>

# OPAM 1.2 : New Workflow

- You will be prompted to create a new package from scratch with it pinned.

```
$ cd ocp-reloc  
$ opam pin add ocp-reloc .
```

```
Package ocp-reloc does not exist, create as a  
NEW package ? [Y/n] y  
  
ocp-reloc is now path-pinned to ~/src/ocp-reloc
```

<http://opam.ocaml.org/blog/opam-1-2-pin/>

# OPAM 1.2 : New Package

- An editor is brought up with a sensible base template.

```
opam-version: "1.2"
name: "ocp-reloc"
version: "0.1"
maintainer: "Louis Gesbert <lg@ocaml.org>"
authors: "Louis Gesbert <lg@ocaml.org>"
homepage: ""
bug-reports: ""
license: ""
build: [
  ["./configure" "--prefix=%{prefix}"
  [make]
]
install: [make "install"]
remove: ["ocamlfind" "remove" "ocp-reloc"]
depends: "ocamlfind" {build}
```

New metadata  
fields now available  
for future

Build and test only  
dependency  
predicates

<http://opam.ocaml.org/blog/opam-1-2-pin/>

# OPAM 1.2 : New Workflow

- Like other Unix tools, you get interactive error checking.

```
[ERROR] File "/home/lg/.opam/4.01.0/overlay/ocp-reloc/  
opam", line 13, character 35-36: '.' is not a valid token.  
Errors in /home/lg/.opam/4.01.0/overlay/ocp-reloc/opam,  
retry editing ? [Y/n]
```

<http://opam.ocaml.org/blog/opam-1-2-pin/>

# OPAM 1.2 : New Workflow

- And any dependent packages automatically get upgraded.

```
ocp-reloc needs to be installed.  
The following actions will be performed:  
- install cmdliner.0.9.5  
[required by ocp-reloc]  
- install ocp-reloc.0.1*  
=== 1 to install ===  
Do you want to continue ? [Y/n]
```

```
opam upgrade ocp-reloc
```

<http://opam.ocaml.org/blog/opam-1-2-pin/>

# OPAM 1.2 : New Workflow

- Metadata maintained and detected from source repository directly.

```
cd ocp-reloc  
git add opam  
git commit -m 'Add OPAM metadata'  
git push
```

- Other developers can pick up your development branches easily.

```
git clone git://github.com/OCamlPro/ocp-reloc.git  
opam pin add ocp-reloc/
```

<http://opam.ocaml.org/blog/opam-1-2-pin/>



# OPAM 1.2 : New Workflow

- Directly clone and pin development versions of existing packages.

```
$ opam source omd --pin  
$ cd omd.0.9.7  
...patch...  
$ opam upgrade omd
```

- New dev-repo metadata lets you grab bleeding edge source.

```
$ opam source --dev-repo --pin
```

<http://opam.ocaml.org/blog/opam-1-2-pin/>

# OCaml Platform

# ~~OCaml~~ OPAM Platform

Tools built around OPAM that provide a modular workflow for developing, publishing and maintaining OCaml source code, both online and offline.

Now with a blog at  
<https://opam.ocaml.org>  
We want *your* articles!

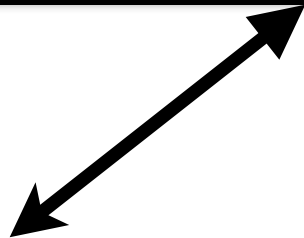
# OPAM 1.2: restructured

Data Libs Tools

opam-repository

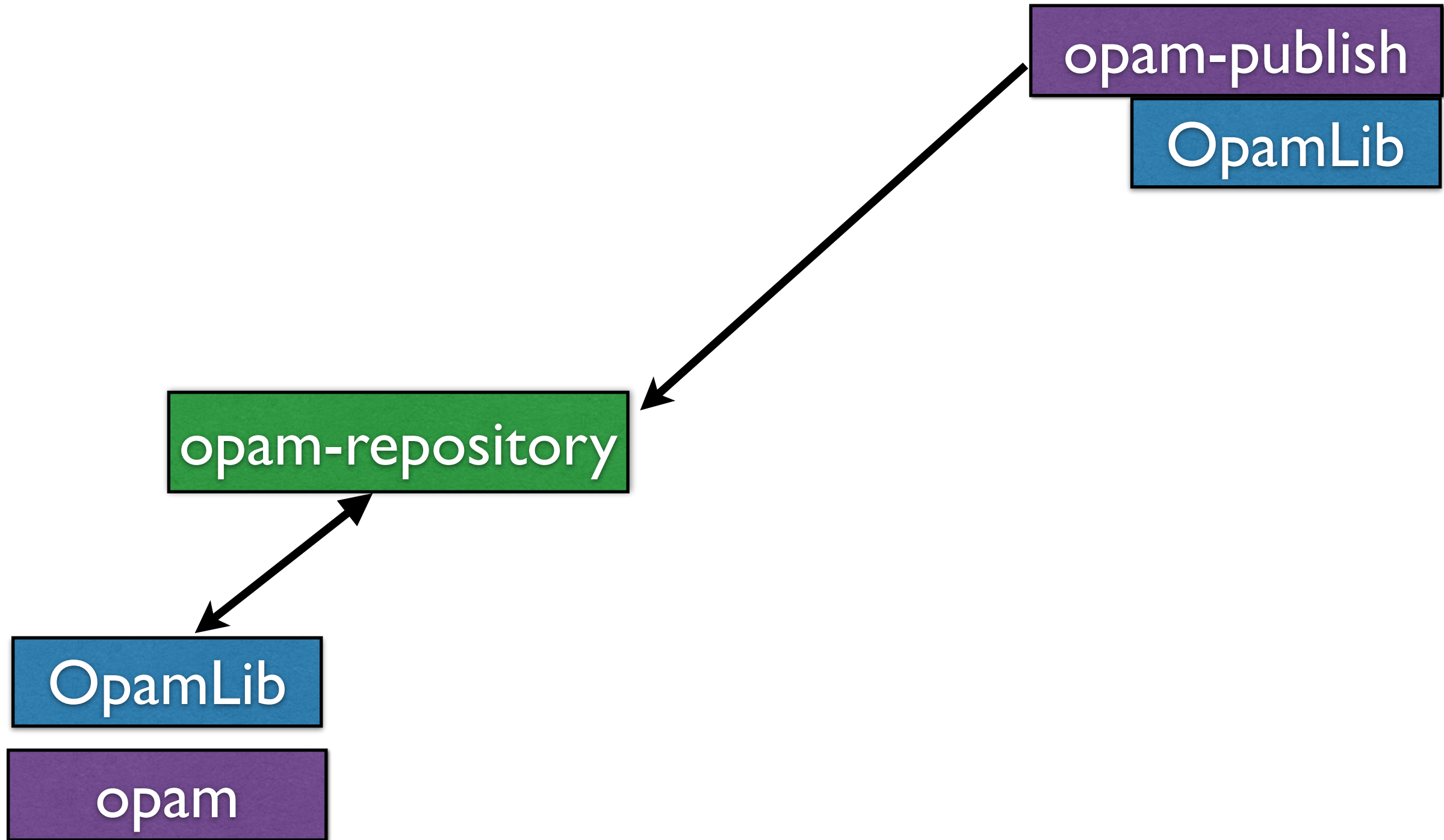
OpamLib

opam



# OPAM 1.2: restructured

Data   Libs   Tools



# OPAM Publish

- Single step publication of new packages
- 1) opam-publish prepare

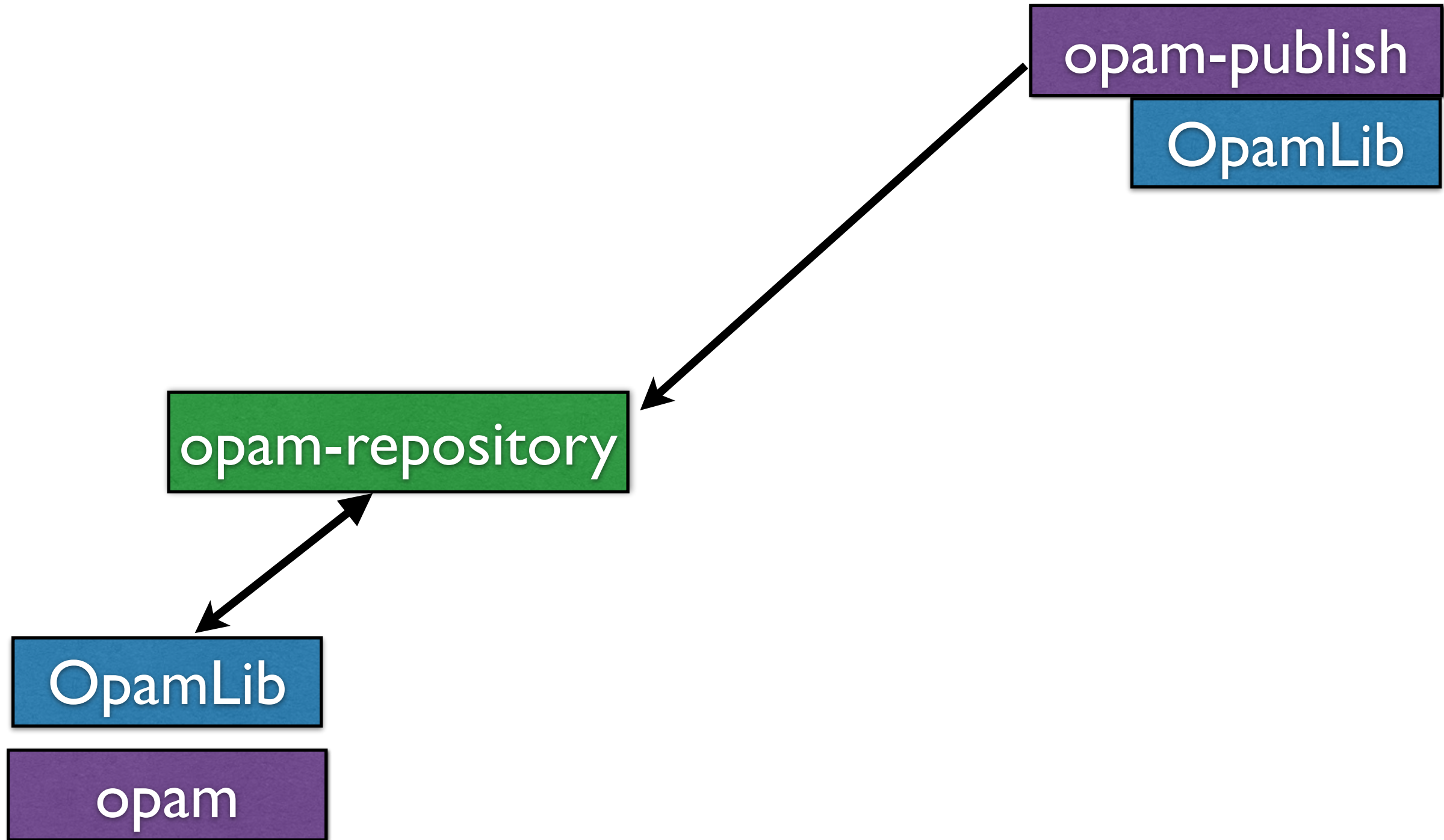
```
thomas@host-78-65-172-114:~/git/opam-repository$ opam-publish prepare uucp.0.9.0 http://erratique.ch/software/uucp/releases/uucp-0.9.0.tbz
[uucp-0.9.0.tbz] Downloading http://erratique.ch/software/uucp/releases/uucp-0.9.0.tbz
Template metadata generated in uucp.0.9.0/.
* Check the 'opam' file
* Fill in or check the description of your package in 'descr'
* Check that there are no unneeded files under 'files/'
* Run 'opam publish submit ./uucp.0.9.0' to submit your package
```

- 2): opam-publish submit

```
thomas@host-78-65-172-114:~/git/opam-repository$ opam-publish submit uucp.0.9.0/
[uucp-0.9.0.tbz] Downloading http://erratique.ch/software/uucp/releases/uucp-0.9.0.tbz
Please enter your github name: samoht
Enter host password for user 'samoht':
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 15751  100 15751    0     0  18948      0  --:--:-- --:--:-- --:--:-- 18931
```

# OPAM 1.2: restructured

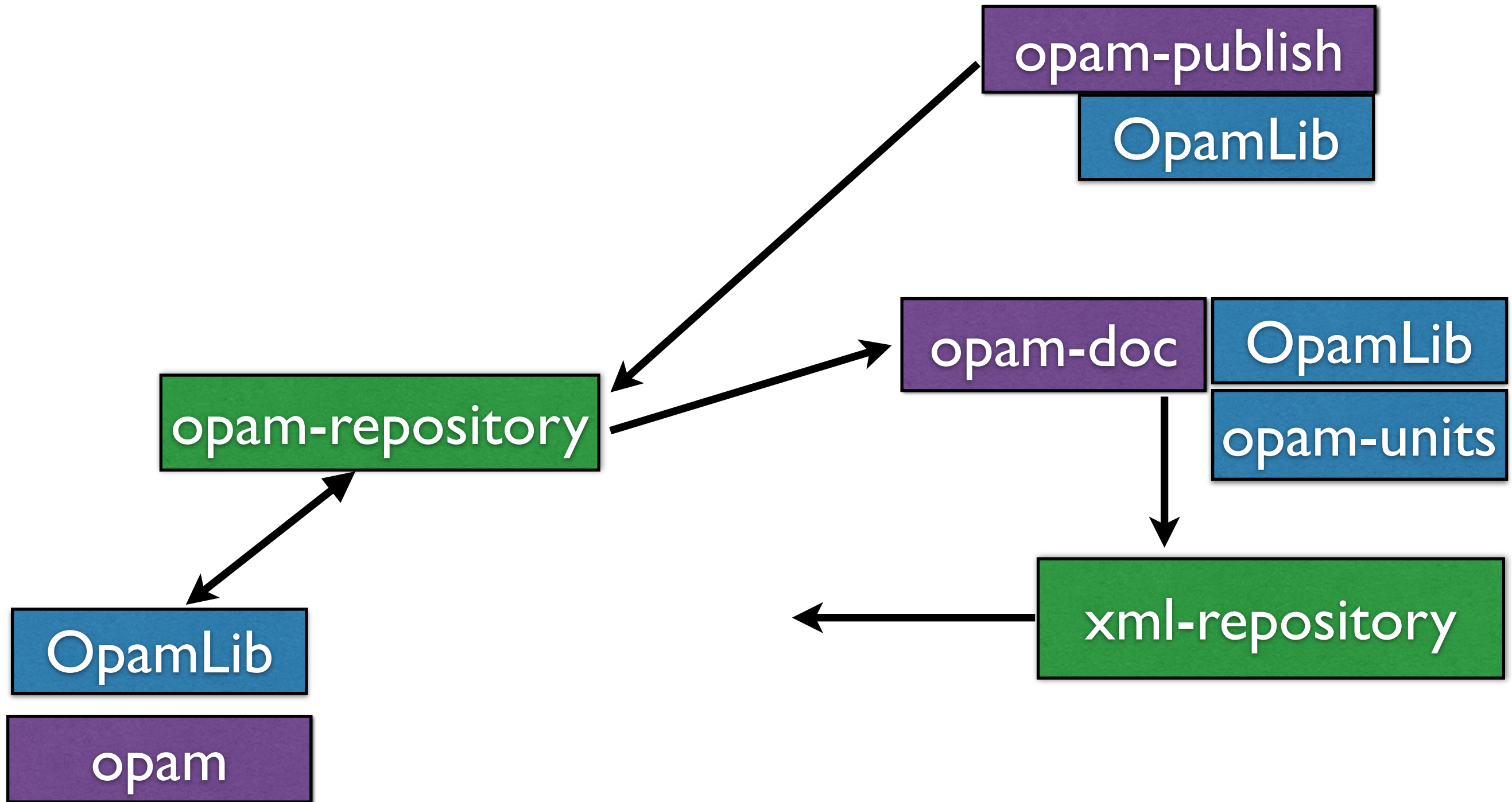
Data   Libs   Tools





# OPAM 1.2: restructured

Data   Libs   Tools





# OPAM Documentation

- **Goal:** documentation unified *across* packages, that handles cross-referencing and module inclusion well.
- **Why it's hard:**
  - not all packages can be installed simultaneously (solved via OPAM)
  - Resolving module inclusion statically leads to combinatorial explosion.
  - Need to integrate with the whole toolchain

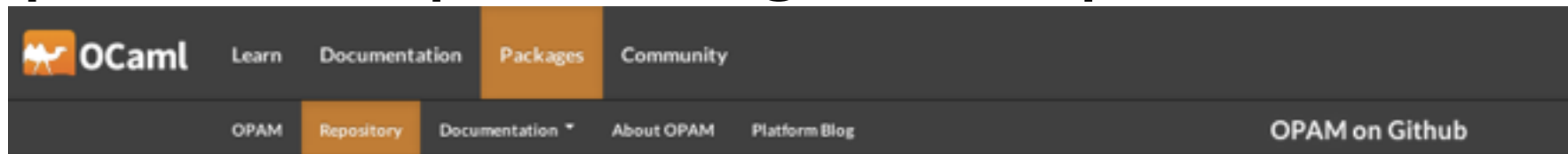
# OPAM Documentation

`opam remote add platform git://github.com/ocaml/platform-dev`

- **Use only the Typed AST (cmt, cmti)**
- Comments are transformed into attributes in the typed AST.
- Attributes are used by external tools (to generate XML / HTML docs).
- Comment attributes can be generated by ppx pre-processors.
- Need to resolve OCaml names to linkable URIs (across universe of packages)

# OPAM Documentation

preview: <http://ocaml.github.io/platform-dev>



xmlm 1.2.0

[ocaml.org](http://ocaml.org) integration

	1.1.1	version 1.2.0
	Daniel Bünzli <daniel.buenzli@erratique.ch>	
	BSD3	
Homepage	<a href="http://erratique.ch/software/xmlm">http://erratique.ch/software/xmlm</a>	
Maintainer	Daniel Bünzli <daniel.buenzli@erratique.ch>	
Tags	xml and codec	
Dependencies	ocamlfind	
OCaml	>= 4.00.0	
Published	Oct 11, 2013	
Source [http]	<a href="http://erratique.ch/software/xmlm/releases/xmlm-1.2.0.tbz">http://erratique.ch/software/xmlm/releases/xmlm-1.2.0.tbz</a> 7e6d3363c2395d84274f1b480e4b6003	
Edit	<a href="https://github.com/ocaml/opam-repository/tree/master/packages/xmlm/xmlm.1.2.0/opam">https://github.com/ocaml/opam-repository/tree/master/packages/xmlm/xmlm.1.2.0/opam</a>	

## Streaming XML codec for OCaml

Xmlm is a streaming codec to decode and encode the XML data format. It can process XML documents without a complete in-memory representation of the data.

Xmlm is made of a single independent module and distributed under the BSD3 license.

findlib packages

xmlm (findlib)

• [Xmlm](#)

module docs

# OPAM Documentation

preview: <https://ocaml.github.io/platform-dev>

```
type solution = (Cudf.package, ActionGraph.t) OpamTypes.gen_solution
```

```
type conflict
```

Abstract type that may be returned in case of conflicts

```
val dependencies : Cudf.universe -> Cudf.package list -> Cudf.package list
```

Return the transitive closure of dependencies of `set`, sorted in topological order

```
val reverse_dependencies : Cudf.universe -> Cudf.package list -> Cudf.package list
```

Return the transitive closure of dependencies of `set`, sorted in topological order

```
val check_request : ?explain:bool -> version_map:int OpamPackage.Map.t -> Cudf.universe -> Cudf_types.vpkg OpamTypes.request -> (Cudf.universe, conflict) OpamTypes.result
```

Check if a request is satisfiable and return the reasons why not unless `explain` is set to `false`

```
val get_final_universe : version_map:int OpamPackage.Map.t -> Cudf.universe -> Cudf_types.vpkg OpamTypes.request -> (Cudf.universe, conflict) OpamTypes.result
```

Compute the final universe state using the external solver.

```
val actions_of_diff : Diff.universe -> Cudf.package OpamTypes.action list
```

Compute the list of actions to match the difference between two universe. Remark: the result order is unspecified, ie. need to use `solution_of_actions` to get a solution which respects the topological order induced by dependencies.

```
exception Cyclic_actions of Cudf.package OpamTypes.action list list
```

```
val solution_of_actions : simple_universe:Cudf.universe -> complete_universe:Cudf.universe -> requested:OpamPackage.Name.Set.t -> Cudf.package OpamTypes.action list -> solution
```

Computes the actions to process from a solution, from the actions obtained by a simple universe diff. The 'simple' universe should not contain build dependencies and will be used for resolution ; `complete_universe` should include build-deps, it's used for ordering of actions and, together with the `requested` set of package names, for computing the reasons of the actions.  
May raise `Cyclic_actions`.

cross-refs  
between  
packages

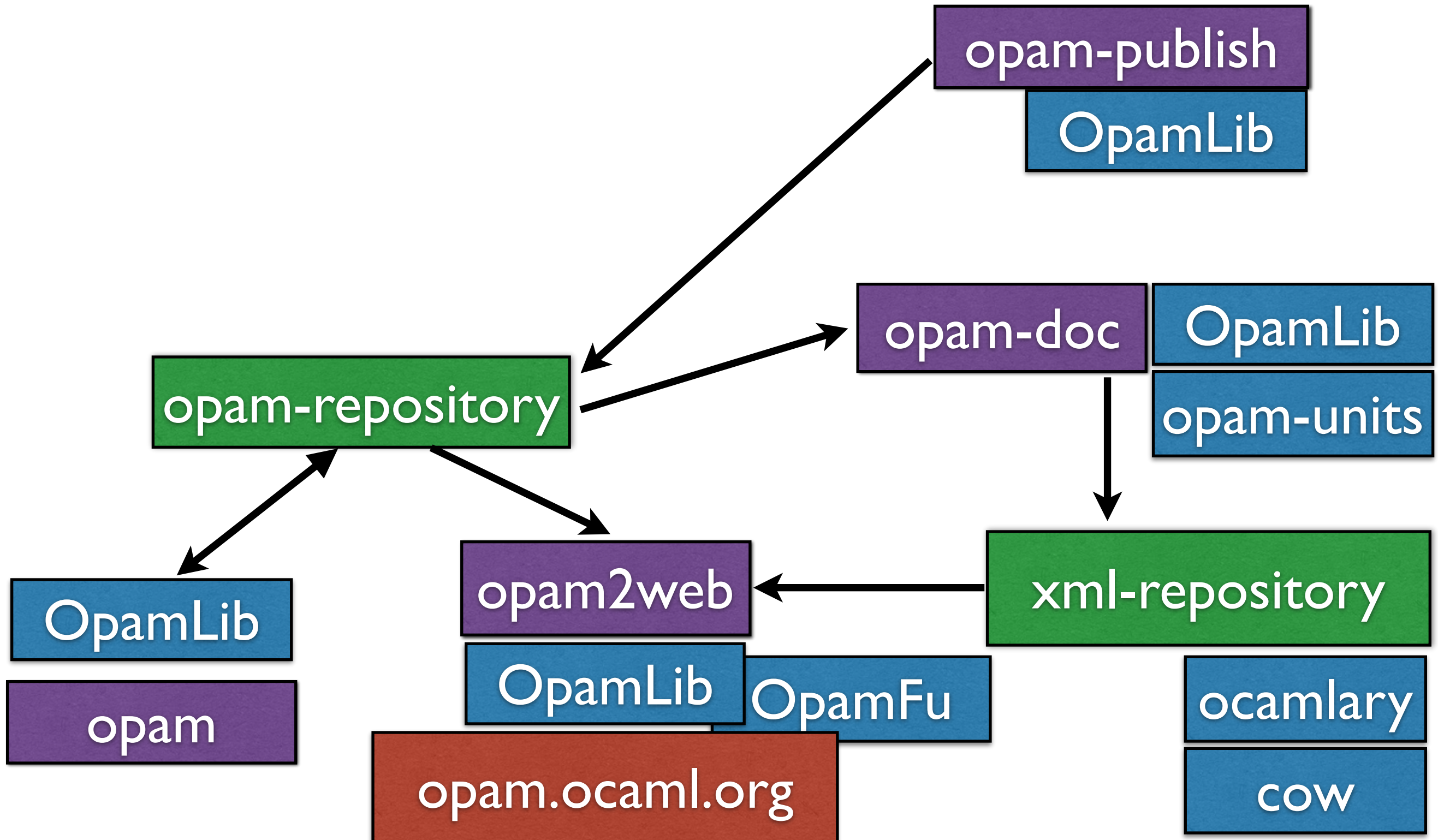
# OPAM Documentation

```
opam remote add platform git://github.com/ocaml/platform-dev
```

- **Current status:** working prototype. Need to improve the tooling, the style and polishing the integration
- **Timeline:**
  - **September:** online release, automatic builds triggered from GitHub pull requests.
  - **November:** use it locally in an OPAM switch (needs a patched OCaml compiler).
  - **December:** Build custom website for other repositories (Jane Street, Citrix, Mirage, my personal homepage, ...)

# OPAM 1.2: restructured

Data   Libs   Tools



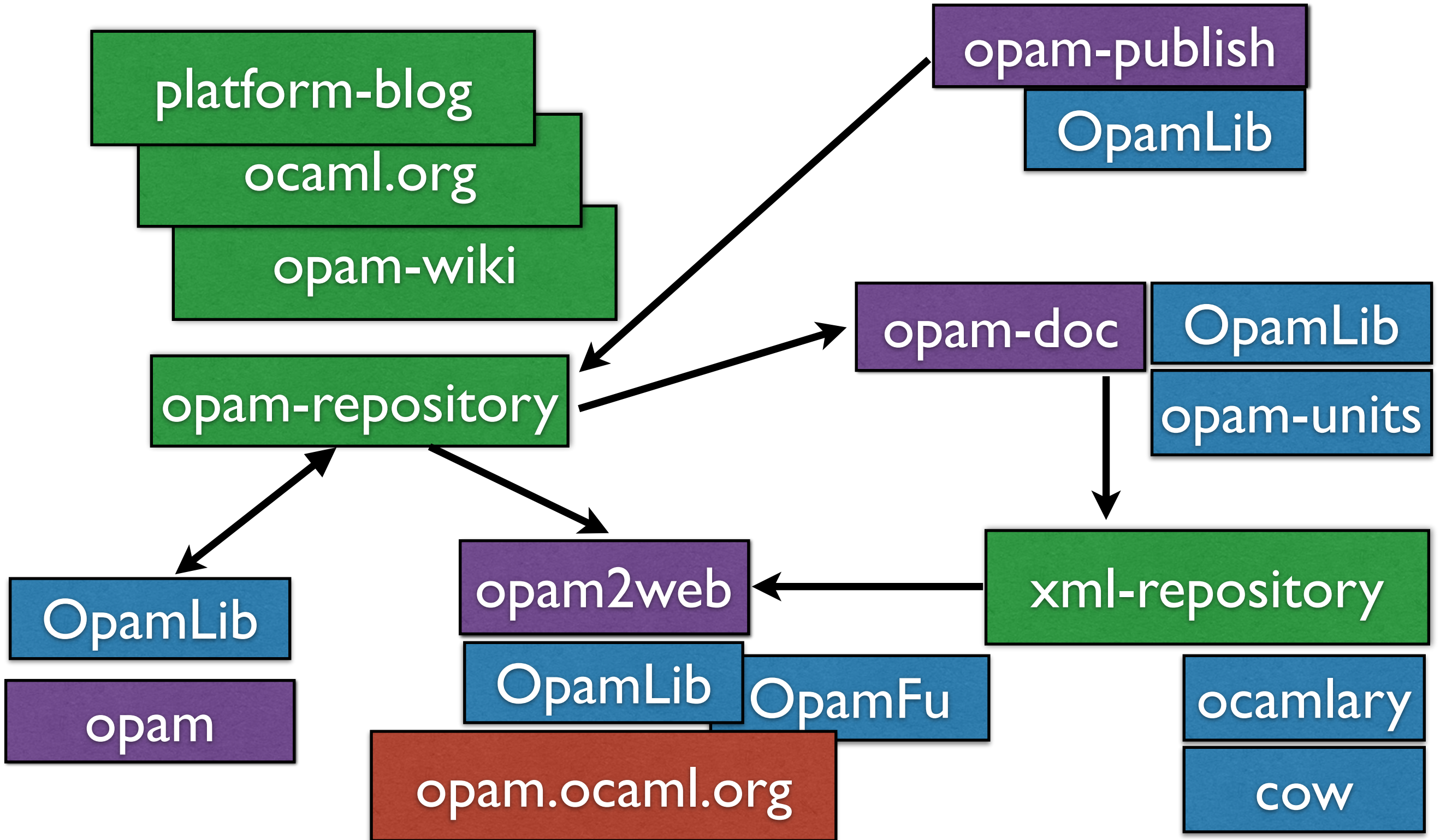


# OPAM 1.2: restructured

# Data

# Libs

# Tools



# Tooling: OCamlJS

- Now supports complete compiler REPL in JavaScript with a IPython toplevel (*source:Andrew Ray, uJamJar*)

```
$ opam remote add andy-opam-dev git://github.com/andrewray/opam-dev.git
$ opam update
$ opam switch install 4.02.0+improved-errors
$ eval `opam config env`
$ opam pin add iocamljs-kernel \
    git://github.com/andrewray/iocamljs.git#improved-error
$ opam install iocaml

$ iocaml -js min my_notebook.ipynb
```

<https://andrewray.github.io/iocamljs/>



# Tooling: GDB + types

```
$ gdb --args ocamlpt.opt -g -g-full -o test test.ml
GNU gdb (GDB) 7.6
Copyright (C) 2013 Free Software Foundation, Inc.
...
Reading symbols from ocamlpt.opt...done.
(gdb) break Selectgen.name_regs
Breakpoint 1 at 0x50dc40: file asmcomp/selectgen.ml, line 95.
(gdb) run
Starting program: ocamlpt.opt -g -g-full -o test test.ml

Breakpoint 1,
  Selectgen.name_regs (rv=[|...|] : Reg.t array, id={...} : Ident.t)
  at asmcomp/selectgen.ml:95
95      let name_regs id rv =
(gdb) print rv
$1 = [| { raw_name = 0;
      stamp = 29;
      typ = Addr;
      loc = Unknown;
      spill = false;
      part = None;
      is_parameter = false;
      interf = □;
      prefer = □;
      degree = 0;
      spill_cost = 0;
      visited = false;
    } : Reg.t |] : Reg.t array
(gdb)
```

# OPAM 1.2 : Polish

- **Easier to package and install.**

Seems minor, but is vital for upstream adoption in OS distros so OPAM is always available.

- **Binary releases now available on:**

Debian/Ubuntu, RHEL/CentOS/Fedora, Arch, OpenSUSE, FreeBSD, OpenBSD, 0install.

- **Documentation rewritten**

User-centric workflows instead of a flat feature list. More tips and tricks on new blog:

<http://opam.ocaml.org>

# Tooling: bulk builds

- **Docker + Xen**

<https://github.com/avsm/docker-opam>

Automate installing any OPAM package inside an isolated(ish) Linux container.

- **Parallel bulk builds of all packages**

- Outputs tracked in Git so developers can checkout logs to triage failures

- <https://github.com/ocaml/opam-bulk-logs>

- **Status:** adding auto-triage and keeping OCaml 4.03 running continuously instead of occasionally.

And just one more  
thing...



# Assemblage

## ALPHA

- eDSL to describe OCaml projects
- Declarative approach: a project is a set of libraries and binaries, which are composed of compilation units.
- Use OCaml as an host language (with Merlin auto-completion)
- Introspect the project description to generate build rules (Makefile, ...)



# Assemblage

## ALPHA

```
open Assemblage

(* OCamlfind packages *)
let lib_pkgs = [
  pkg "opam-lib";
  pkg "opam-lib.client";
  pkg "compiler-libs.common";
  pkg "compiler-libs.optcomp";
  pkg "ocamlgraph";
  pkg "findlib";
  pkg
]
1 pkg      ?available:Assemblage.Features.t -> ?flags:Assemblage.Flags.t -> ?opt:bool -> string -> [> `Pkg of Assemblage.pkg ] V
1 pkg_c    ?available:Assemblage.Features.t -> ?flags:Assemblage.Flags.t -> ?opt:bool -> string -> [> `Pkg of Assemblage.pkg ] V
  pkg_pp    ?available:Assemblage.Features.t -> ?flags:Assemblage.Flags.t -> ?opt:bool -> string -> [> `Pkg of Assemblage.pkg ] V

(* Compilation units *)
let opamUnitsConfig = unit "opamUnitsConfig" (`Path ["src"])
let opamLibrary = unit "opamLibrary" (`Path ["src"])
let opamUnit = unit "opamUnit" (`Path ["src"])
let opamUnitsState = unit "opamUnitsState" (`Path ["src"])
let main = unit "main" (`Path ["src"])

(* Binary and library *)
let l = lib ~deps:lib_pkgs "opam-units" (`Units [
  opamLibrary;
  opamUnit;
  opamUnitsConfig;
  opamUnitsState])

let b = bin ~deps:(l :: bin_pkgs) "opam-units" (`Units [main])

let () = assemble (project "opam-units" [b;l])
```

# Assemblage

## ALPHA

- Generate files to build, install, use a project

```
thomas@piana:~/git/opam-units$ assemblage setup
=> Loading

    opam-units b4b9c3 🍷
=> write Makefile
=> write META
=> write opam-units.install
=> write .merlin
```

- Timeline
  - **today:** alpha (use as your own risk)
  - **end of september:** beta-release
  - **1.0 release criteria:** the Buenzli test

# Conclusions

- Platform is an ongoing effort to build a modular set of tools, libraries and data.
- OPAM provides the workflow interface to the package universe (e.g. *Jun Furuse's Camlspotter can be integrated more easily!*)
- Contributors extremely welcome:
  - Mailing List: [platform@lists.ocaml.org](mailto:platform@lists.ocaml.org)
  - <http://lists.ocaml.org/listinfo/platform>