Alire: a library repository manager for the open source Ada ecosystem

Alejandro R. Mosteo

2018-jun-19

CUD

Centro Universitario de la Defensa Zaragoza

CONTENTS

Centro Universitario de la Defensa Zaragoza

MOTIVATION (personal)

- Experience with
  - Linux package managers:

```
sudo apt install libgtkada16.1.0-dev
```

  - Java (Android) gradle:

```
dependencies {
    compile 'com.example.android:lib-magic:1.3'
}
```

  - Python’s pip, javascript’s npm, …

MOTIVATION (general)

CODE REUSE

do not reinvent the wheel

SIMPLICITY

“it just works”

AVAILABILITY / PUBLICITY

reach your audience

REPRODUCIBILITY

tested configurations

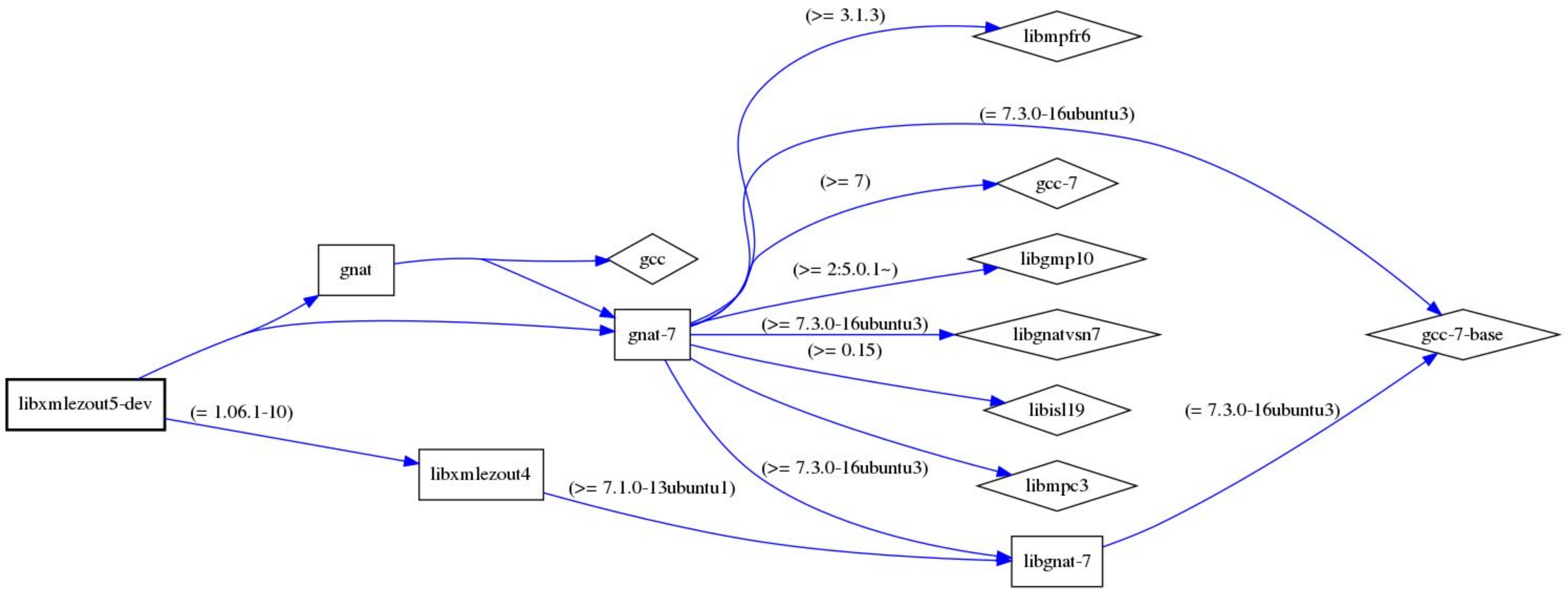
PORTABILITY

cross-platform packaging

SAFETY

bug & vulnerability fixes

DEPENDENCY RESOLUTION



- Initial deployment
  - Find valid combination
- Subsequent updates
  - Staying backward compatible

debtree libxmlezout5-dev --no-recommends --no-alternatives --condense --no-conflicts --max-depth=3 | dot -T png

Centro Universitario de la Defensa Zaragoza

SEMANTIC VERSIONING

version 1.2.3-prerelease+anything

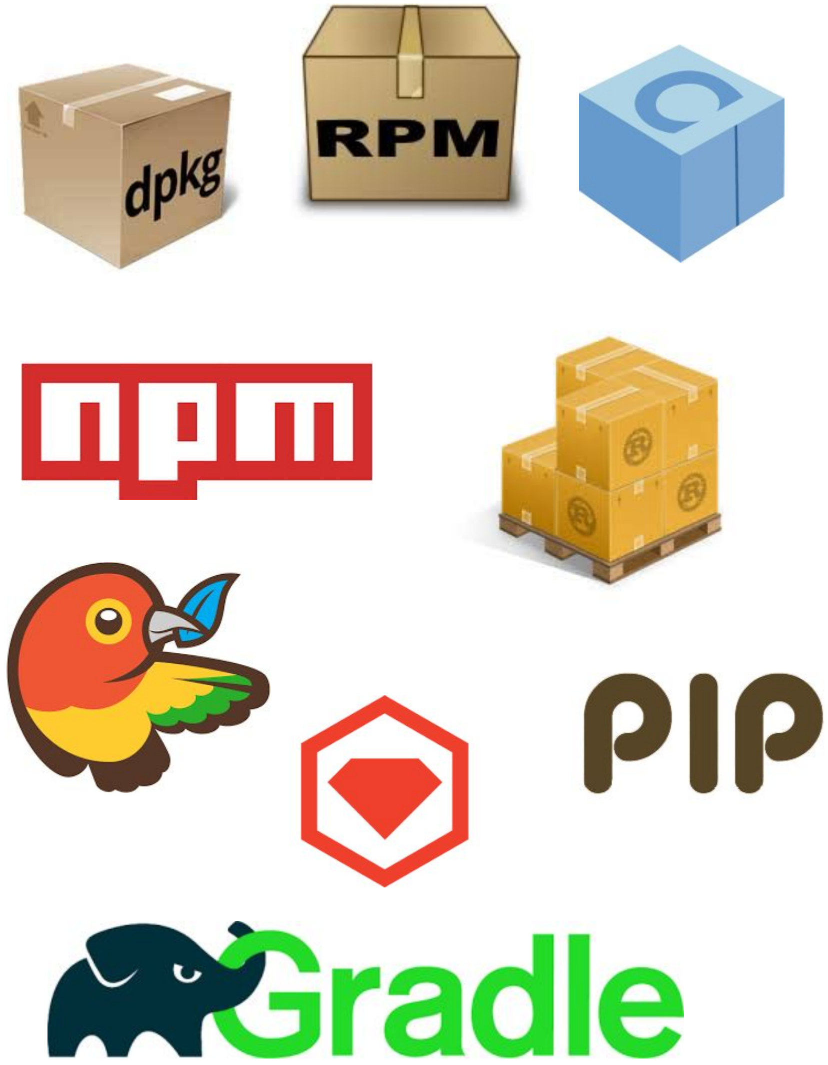
- **major . minor . patch**
  - Major changes break compatibility
  - Minor changes add functionality
  - Patch changes fix bugs
- Minor/Patch upgrades “should” be safe.

- Meaningful only when offering an API
- Can assimilate other versioning methods
  - Calendar versioning: 20180501.0.0

ALIRE MOTIVATION

Ada world

Out there



ALIRE MOTIVATION

Ada world

RAVENPORTS

ALR

Out there

dpkg

RPM

npm

PIP

Gradle

ALIRE MOTIVATION

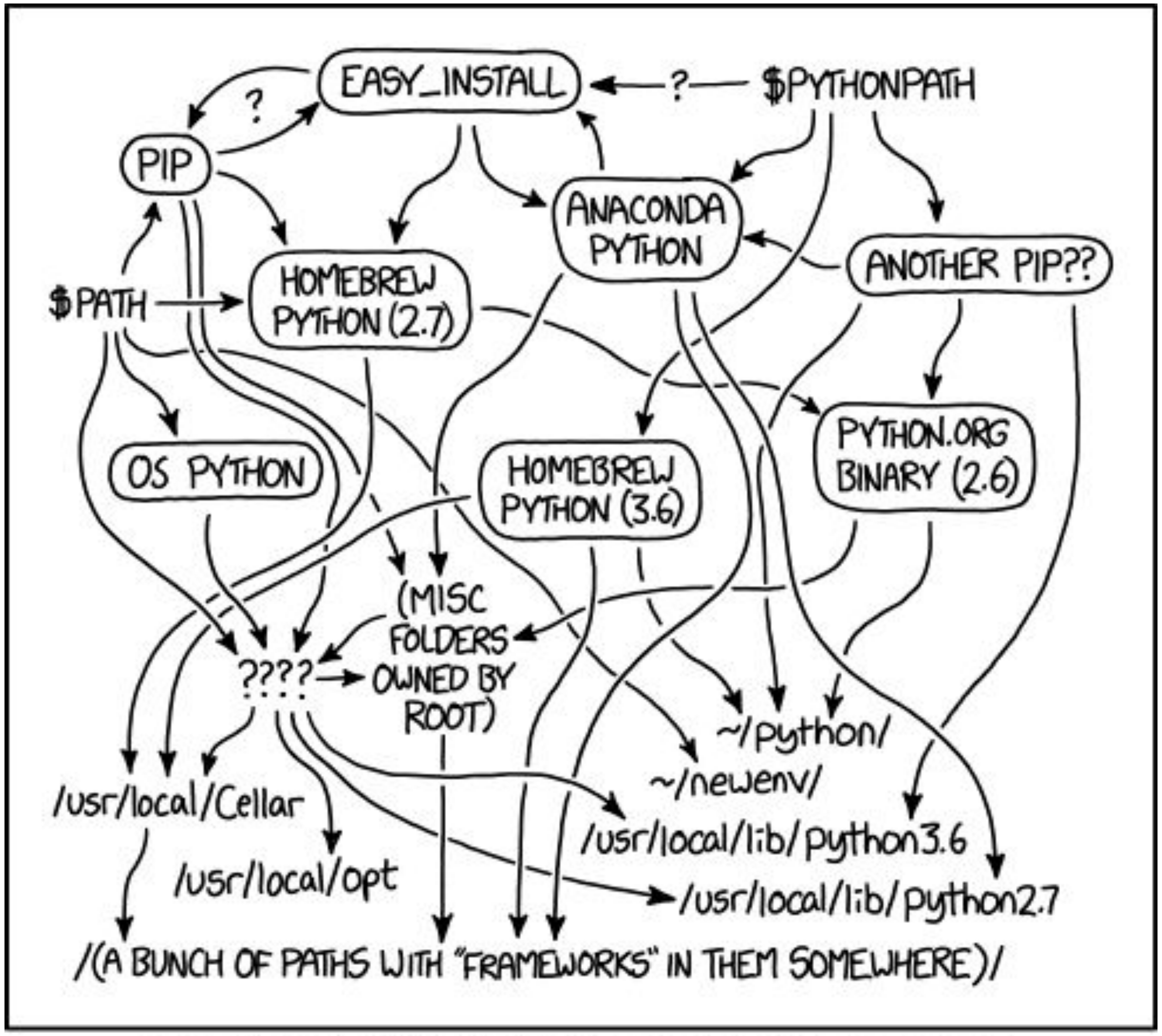
- So, why reinvent the wheel?

Stay within the Ada boundaries

Ada stability makes it an “easy” target

Because why not

THE WORST OF BOTH WORLDS (as of May 2018)



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

https://xkcd.com/1987/

EXAMPLES & KINDS

SYSTEM vs **SANDBOX**

PLATFORM vs **LANGUAGE**

BINARIES vs **SOURCES**

OFFICIAL vs **COMMUNITY**

CONTENTS

ALIRE vs alr

https://github.com/alire-project

Alire

- Database project
- "Passive" functionality

https://github.com/alire-project/alire

alr

- Command-line tool
- "Active" functionality
  - Dependency solver
  - Project fetching
  - Building process

https://github.com/alire-project/alr

Users just wanna have fun

```
$ alr get --compile eagle_lander
$ alr get --compile hangman

$ ls
hangman_1.0.0_a5790492

$ cd hangman_1.0.0_a5790492
$ alr run
***** W E L C O M E  T O  H A N G M A N  *****
 By: Jon Hollan, Mark Hoffman, & Brandon Ball

$ alr run --list
Project hangman builds these executables:
   hangmain (found at /tmp/demo/hangman_1.0.0_a5790492/bin/hangmain)
```

FINAL USERS

```
$ alr list
ada_lua          An Ada binding for Lua
adacurses        Wrapper on different packagings of NcursesAda
adayaml          Experimental YAML 1.3 implementation in Ada
adayaml.server   Experimental YAML 1.3 server component
agpl             Ada General Purpose Library with a robotics flavor
ajunitgen        Generator of JUnit-compatible XML reports
alire            Alire project catalog and support files
alr              Command-line tool from the Alire project
apq              APQ Ada95 Database Library (core)
aunit            Ada unit test framework

$ alr search x
NAME        VERSION  DESCRIPTION
rxada       0.1.0    RxAda port of the Rx framework
xml_ez_out  1.6.0    Creation of XML-formatted output from Ada programs
xstrings    1.0.0    Renaming of gnatcoll.strings w/o other dependencies
```

“with” a little help from my fellow developers

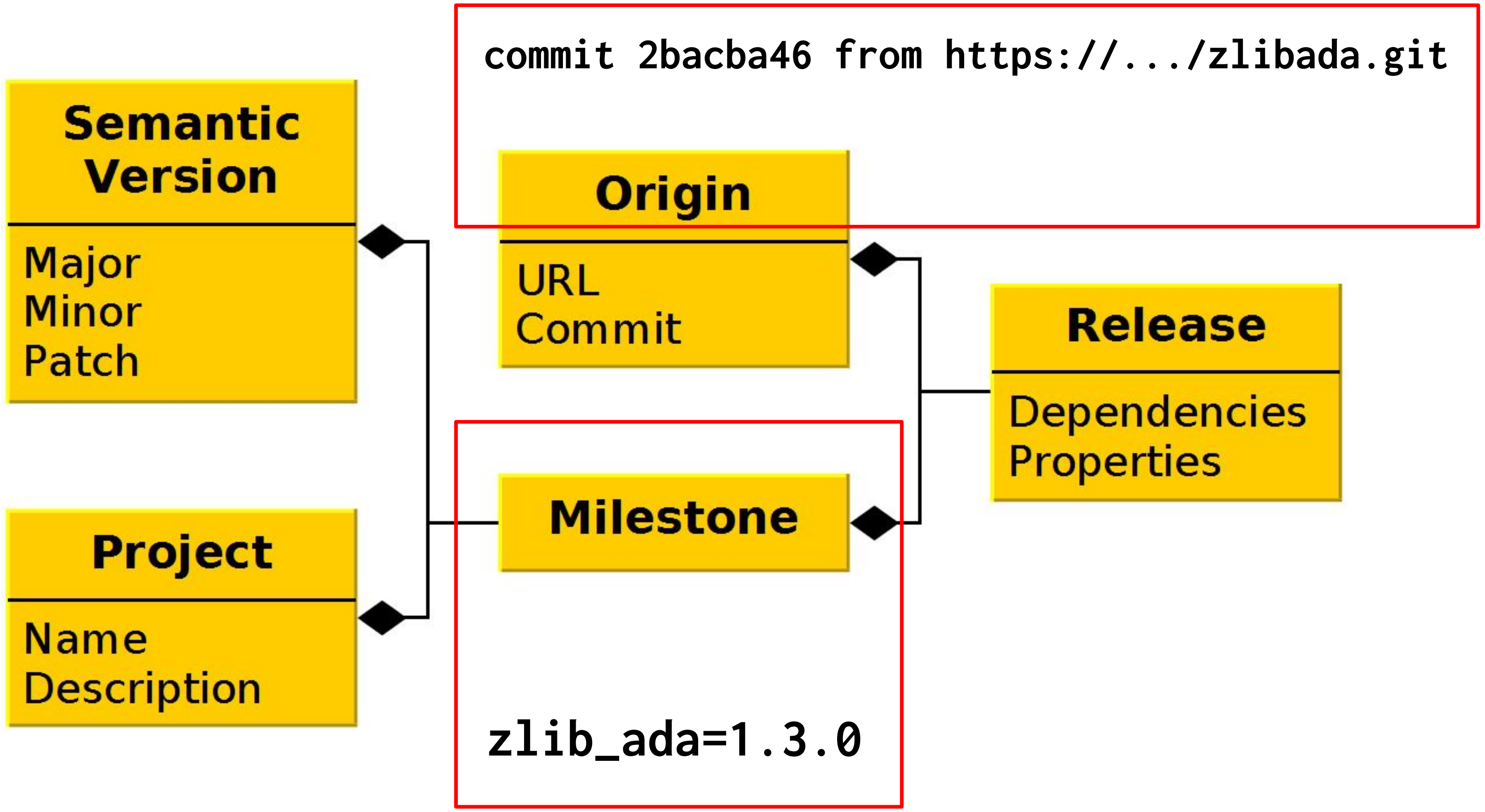
```
$ alr init --bin zzz
$ cd zzz
$ alr build ✓
$ alr run # null main
$ alr with xstrings
$ vi zzz.gpr
$ vi src/zzz.adb
$ alr run
Zzz...
```

```
$ alr init --bin zzz
$ cd zzz
$ alr build ✓
$ alr run # null main
$ vi zzz.gpr
   with “xstrings”;
   --  alr with xstrings
$ alr with --from zzz.gpr
$ vi src/zzz.adb
$ alr run
Zzz...
```

DEVELOPERS

```
$ alr show adayaml
adayaml=0.3.0: Experimental YAML 1.3 implementation in Ada
Origin: commit 2017a7c2523499c03b8d7fe06546a5a8bae6476d
        from   https://github.com/yaml/AdaYaml.git
Properties:
   Project_File: yaml.gpr
   Project_File: yaml-annotation_processor.gpr
   Project_File: yaml-utils.gpr
   GPR Scenario: Mode := debug | release
   Author: Felix Krause
   Website: https://ada.yaml.io/
   License: MIT
Dependencies (direct):
   aunit is At_Least (2017.0.0)
Dependencies (solution):
   aunit=2017.0.0
```

ALIRE DATA STRUCTURES



Index file in Alire

```
package Alire.Index.RxAda is

   function Project is new Catalogued_Project ("Rx in Ada");

   Repo : constant URL := "https://bitbucket.org/amosteo/rxada";

   V_0_1_0  : constant Release :=
                Project.Register
                  (V ("0.1.0"),
                   Hg (Repo, "361d4e2ab..."),
                   Properties =>
                     Executable ("rx-examples-basic") and

                     Author ("amosteo@unizar.es") and
                     License (LGPL_3_0));
```

Project name from GNAT.Source_Info.Enclosing_Entity

Semantic version

Commit

Optional properties

Specifiying dependencies

```
with Alire.Index.Alire;
with Alire.Index.AJunitGen;
with Alire.Index.XML_EZ_Out;

package Alire.Index.Alr is

   Base : constant Release := Project.Unreleased (...);
   --  All common properties declared here

   package V_0_4 is new Project_Release -- Version is “reflected”
     (Base.Replacing (Git (Repo, "721d1112..."))
          .Extending (Dependencies =>
                       AJunitGen.Project.Within_Major ("1.0") and
                             Alire.V_0_4.Within_Minor and
                        XML_EZ_Out.V_1_6.Within_Major));
```

CONTENTS

Centro Universitario de la Defensa Zaragoza

STAYING WITHIN Ada

- Because we can (we do what we must)
  - Done before:
    - AWS.Resources
    - Ada-Europe 2017, *Astronomical Ada*, Ahlan Marriott
- But `alr` needs to
  - Update the catalog
  - Parse working-project dependencies
- Solution: recompilation
  - With generated session files

alr's MULTIPLE PERSONALITIES

"stub" in (e.g.) `$HOME/bin/alr`

- Built at installation time, never recompiled
- Contains minimal index
- Generates full index and compiles:

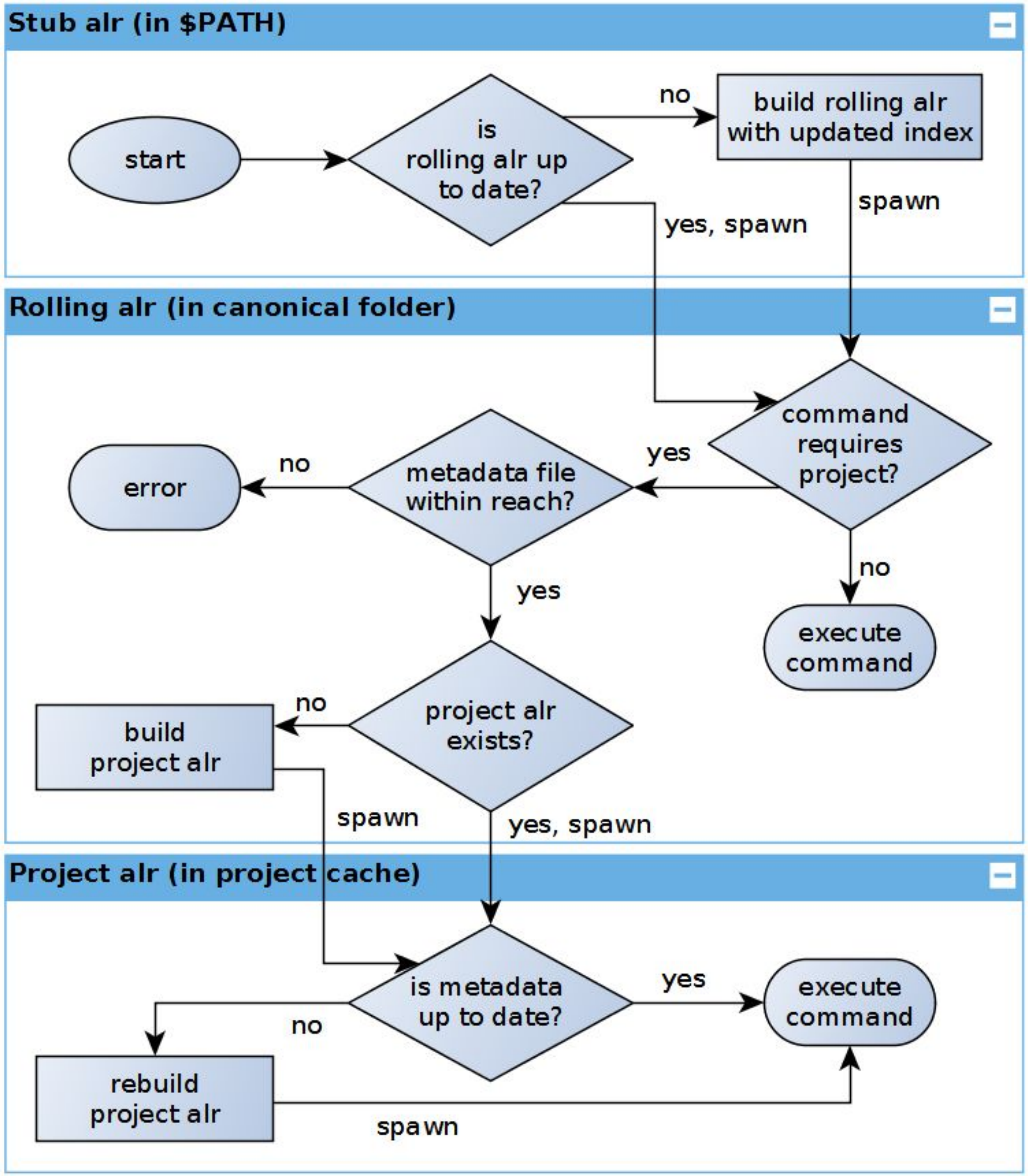
"rolling" in `$XDG_CONFIG_HOME/alire/…/alr`

- Built whenever catalog is updated
- Contains full index
- Generates project-specific Ada files and compiles:

"project" in `<working project>/alire/…/alr`

- Built whenever catalog or metadata changes
- Contains full index + project data

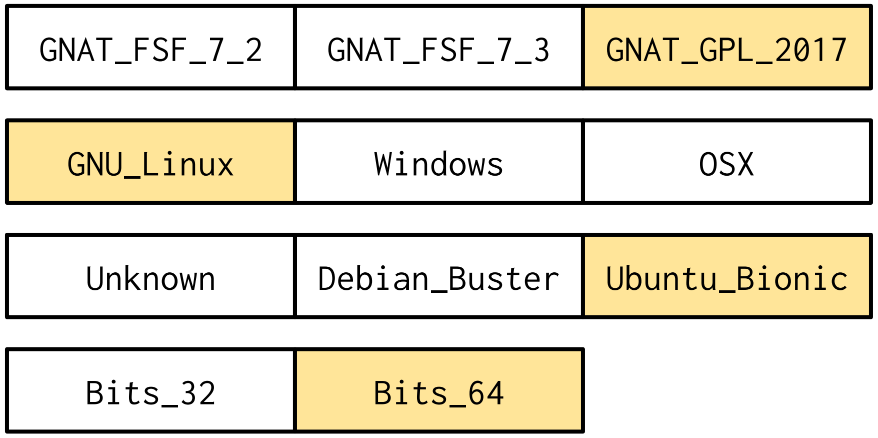
Centro Universitario de la Defensa Zaragoza


Stub alr (in $PATH)
start
is rolling alr up to date?
no
build rolling alr with updated index
spawn
yes, spawn
Rolling alr (in canonical folder)
command requires project?
yes
metadata file within reach?
no
error
no
execute command
yes
project alr exists?
no
build project alr
spawn
yes, spawn
Project alr (in project cache)
is metadata up to date?
yes
execute command
no
rebuild project alr
spawn

INDEX FORMAT: HIGHLIGHTS

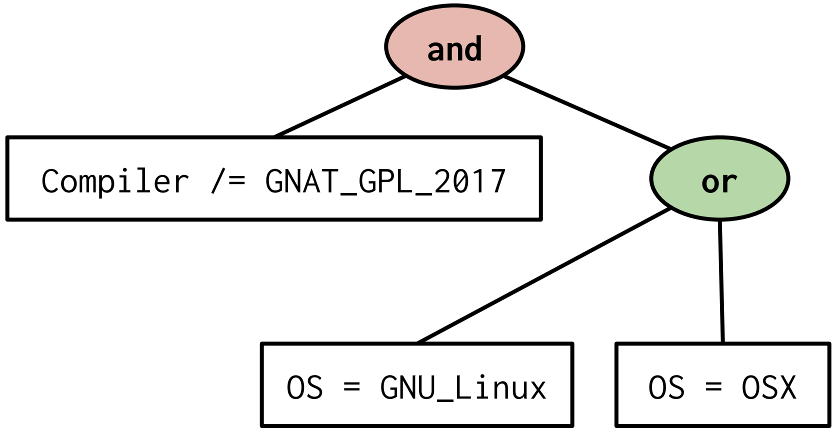
`Property'Class`

- Encapsulate some **environment** property
- Unknown until runtime
- Examples:
  - Compiler
  - Operating system
  - Architecture



`Requisite'Class`

- Parallel hierarchy
- Logical expressions on **matching** properties
- Evaluated at runtime against available properties



CONDITIONAL PROPERTIES

- All dependencies/properties are conditional
- A requisite tree has to be fulfilled
  - True if omitted

```
Project_File ("scenarios/catastrophical.gpr")        -- unconditional

On_Condition (Operating_System = Windows,            -- if-then-else
              Project_File ("project_win.gpr"))

Case_Operating_System_Is                             -- case is
   ((GNU_Linux => Comment ("Long life the penguin"),
     OSX       => Comment ("Oh shiny!"),
     others    => Comment ("Pick your poison")))
```

Centro Universitario de la Defensa Zaragoza

EXAMPLES: CONDITIONAL DEPENDENCIES

Three kinds of conditions:

```
Dependencies =>
   Half_Life.Project >= "3.0" and   -- Unconditional

   (On_Condition                    -- Conditional
      (Operating_System = Windows,
       When_True  => Star_Citizen.Project.Current,
       When_False => Nethack.Project /= "1.27")) and

   (GNATCOLL.Strings.Project or     -- One of several
    GNATCOLL.Slim.Project    or
    GNATCOLL.Project)
```

EXAMPLES: NATIVE PLATFORM PACKAGES

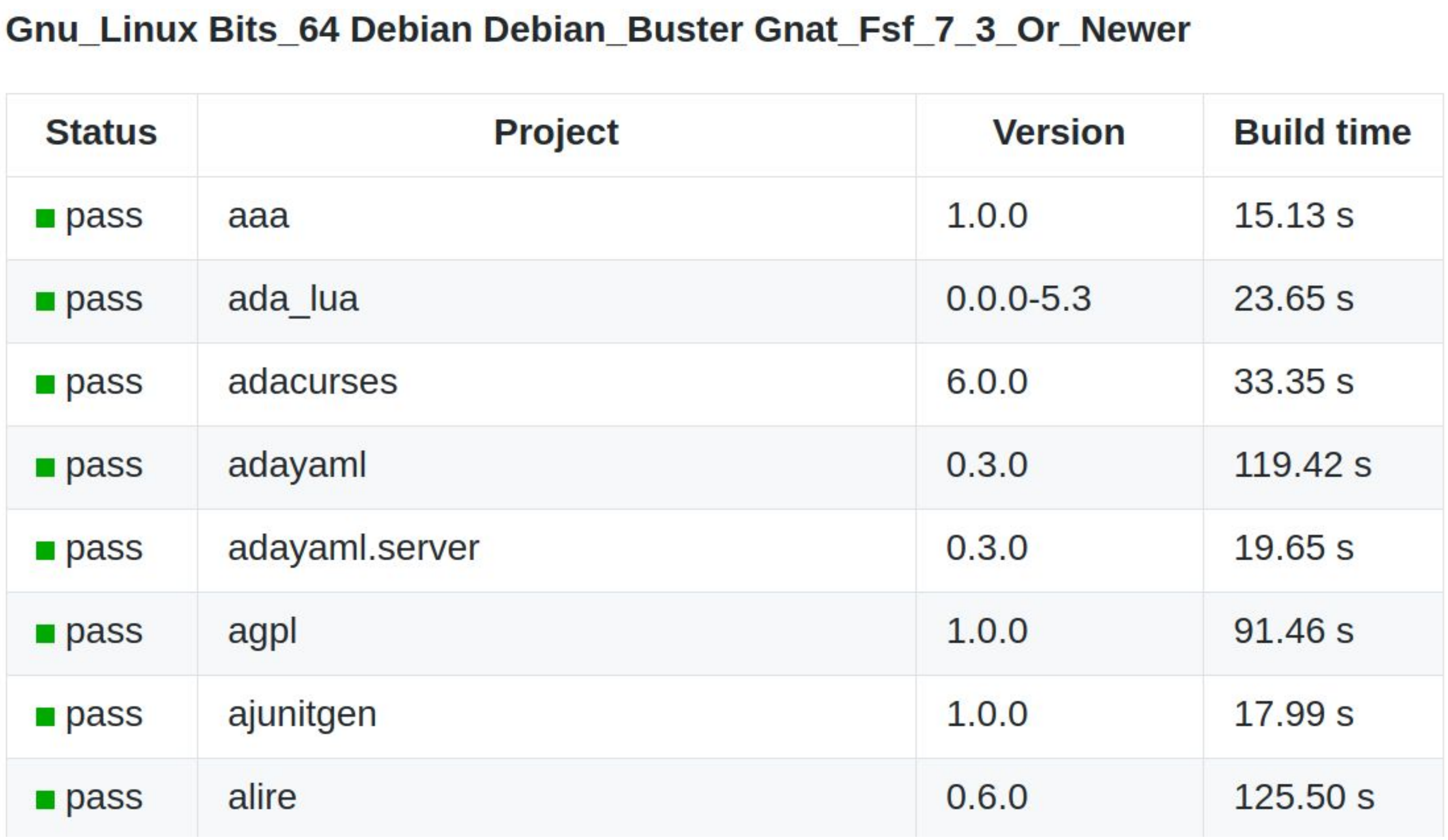
```
package Alire.Index.ZLib is

   function Project is new Catalogued_Project
     ("Library implementing the deflate method from gzip/PKZIP");

   package V_1_2 is new Project_Release
     (Base.Replacing
        (Origin =>
           Native ((Debian | Ubuntu => Packaged_As ("zlib1g-dev"),
                    others          => Unavailable))));

end Alire.Index.ZLib;
```

CONCLUSION

- Alire + `alr` exists already
  - Debian testing / Ubuntu 17.10, 18.04 LTS
  - GNAT FSF 7.x / GNAT GPL 2017 / ~~Community 2018~~
- Userspace-oriented
  - Does not manage a “global view” of installations
  - But can use available system packages
    - Eases initial packaging curve for complex dependencies
- It offers most expected capabilities
  - Flexible dependencies / properties
    - Conditional / Alternatives / Conflicts
  - Relying only on free / open source projects & services
    - Zero-cost at this time
  - Verified through Continuous Integration

Centro Universitario de la Defensa Zaragoza

CATALOG STATUS in master branch



Gnu_Linux Bits_64 Debian Debian_Buster Gnat_Fsf_7_3_Or_Newer

Status	Project	Version	Build time
🟩 pass	aaa	1.0.0	15.13 s
🟩 pass	ada_lua	0.0.0-5.3	23.65 s
🟩 pass	adacurses	6.0.0	33.35 s
🟩 pass	adayaml	0.3.0	119.42 s
🟩 pass	adayaml.server	0.3.0	19.65 s
🟩 pass	agpl	1.0.0	91.46 s
🟩 pass	ajunitgen	1.0.0	17.99 s
🟩 pass	alire	0.6.0	125.50 s

https://github.com/alire-project/alr/blob/master/status/gnat-fsf-7.3.md

FUTURE STEPS

- Cross-compilation
  - Ada is strong in the ~~Force~~ embedded world
- Support for more platforms
  - Windows
    - What to do about lack of native manager
- Promotion in the Ada community
  - Ada-Europe / Ada User Journal / `comp.lang.ada`
  - Await verdict of the masses
- Grow the catalog

THANKS FOR YOUR ATTENTION

- https://github.com/alire-project/
- amosteo@unizar.es
- @mosteobotic

?

Centro Universitario de la Defensa Zaragoza

cud.unizar.es

Academia General Militar · Ctra. Huesca s/n · 50090 Zaragoza · 976 739 500

My own questions to YOU ;-)

- `alr compile`:
  - aggregate project
  - builds ev. at once
  - using original GPRs

?

- `alr install` (tbd):
  - env. var., shared prefix
  - gprbuild + gprinstall
  - Stand-alone, safe order

- Consistence of the whole:
  - Libraries decide over:
    - shared / static / etc
  - Requires manual tinkering with most libs (!)
  - Global overriding of library kinds?
    - Or some standardized Externals?