The Alire package manager for Ada and SPARK

https://alire.ada.dev/

Alejandro R. Mosteo

Fabien Chouteau

June 14th, 2022

Centro Universitario de la Defensa Zaragoza

TOOLCHAIN INSTALLATION

- Alire indexes GNAT FSF for:
  - Windows
  - Linux
  - macOS
- Including cross-compilers for:
  - ARM Cortex-M
  - AVR
  - RISC-V
- Pre-built with public scripts at GitHub::
  - https://github.com/alire-project/GNAT-FSF-builds

USE CASE: TOOLCHAIN INSTALLATION

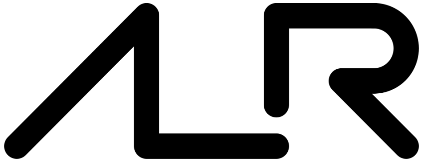
```
$ alr toolchain --select gnat gprbuild
$ alr toolchain --select
Please select the gnat version for use with this configuration
  1. gnat_native=11.2.4
  2. None
  3. gnat_external=9.4.0 [Detected at /usr/bin/gnat]
  4. gnat_arm_elf=11.2.4
  5. gnat_avr_elf=11.2.4
  6. gnat_riscv64_elf=11.2.4
  a. (See more choices...)
$ alr toolchain
CRATE         VERSION  STATUS    NOTES
gprbuild      22.0.1   Default
gprbuild      2019.0.0 Available Detected at /usr/bin/gprbuild
gnat_arm_elf  11.2.3   Available
gnat_native   10.3.2   Available
gnat_native   11.2.4   Default
gnat_external 9.4.0    Available Detected at /usr/bin/gnat
```

CONTENTS

- Introduction
- Use cases
- Ecosystem

PACKAGE MANAGERS

Ada world



AURA

TIMELINE

- → **Jun 2017: first repository & discussions** (with seeds at A-E 2017)
- → **Feb 2018: presentation at Ada-Europe 2018**
- → Apr 2019: AdaCore sponsorship
- → **Aug 2019: Website goes live**
- → Nov 2019: internal beta
- → **Feb 2021: v1.0**
- → Sep 2021: v1.1
- → May 2022: v1.2
- → Jun 2022: first tutorial

KINDS OF PMs

SYSTEM vs **USER**

PLATFORM vs **LANGUAGE**

BINARY vs **SOURCES**

OFFICIAL vs **COMMUNITY**

OVERVIEW

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

Alire

- Project as a whole
- Community index
  - Available packages
  - Packages ⇒ *“crates”*

`alr`

- Command-line tool
  - Dependency solver
  - Source downloading
  - Building

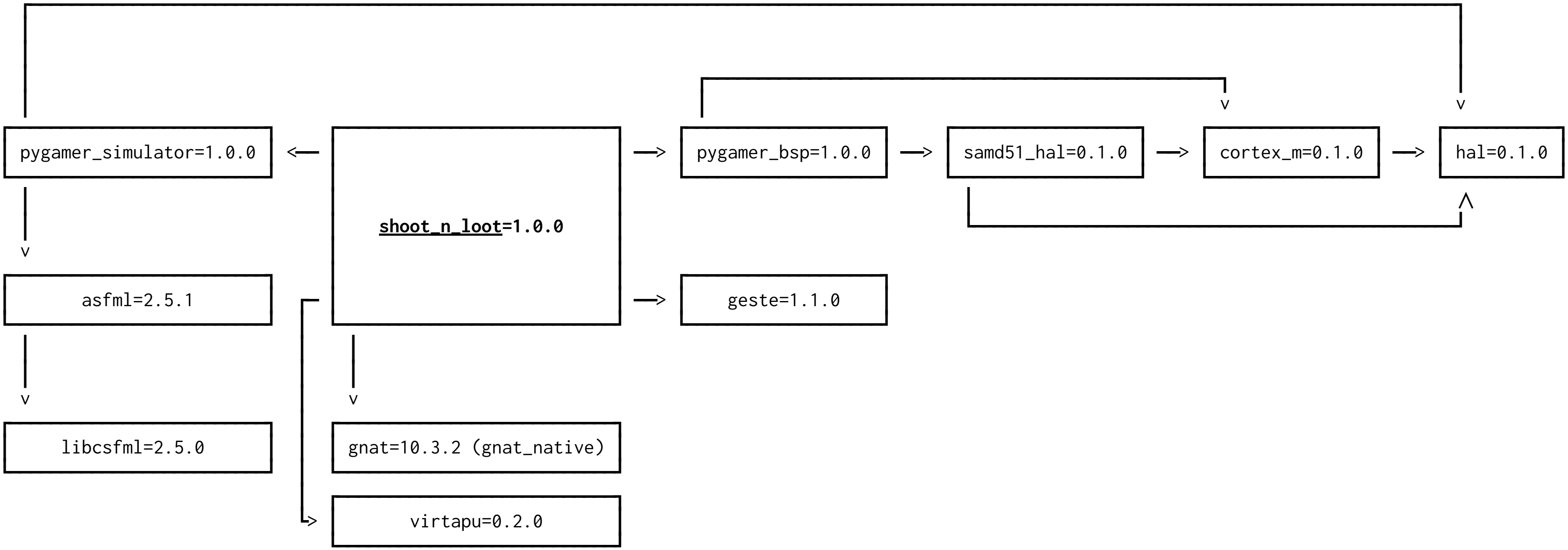
Ada **Li**brary **Re**pository

Centro Universitario de la Defensa Zaragoza

USE CASE: OBTAIN AND RUN AN EXECUTABLE

```
$ alr get 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 ./bin/hangmain)
```

USE CASES: LIBRARY REUSE / DEPENDENCIES



pygamer_simulator=1.0.0

shoot_n_loot=1.0.0

pygamer_bsp=1.0.0

samd51_hal=0.1.0

cortex_m=0.1.0

hal=0.1.0

asfml=2.5.1

geste=1.1.0

libcsfml=2.5.0

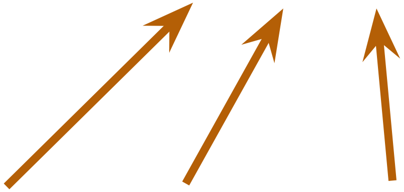
gnat=10.3.2 (gnat_native)

virtapu=0.2.0

- Dependency reuse
  - – Simply reuse code from other projects
    - Finding a valid dependency solution
- Updates

SEMANTIC VERSIONING

version 1.2.3-prerelease+anything



- **major . minor . patch**
  - **Major** changes that **break** compatibility
  - **Minor** changes that **add** functionality
  - **Patch** that fix bugs **preserving** the API w.o. changes
- Minor/patch updates are theoretically “safe”

- Special operators:
  - ^1.0 (>=1.0 & <2.0) -- Excludes e.g. 2.0-rc1
  - ~2.3 (>=2.3 & <2.4)

USE CASES: DEPENDENCIES, SEARCHING

```
$ alr search --crates
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 toml
NAME         STATUS  VERSION  DESCRIPTION
ada_toml             0.3.0    TOML parser for Ada
toml_slicer          0.1.0    Edit TOML files directly without parsing
```

USE CASES: BOOTSTRAP A PROJECT

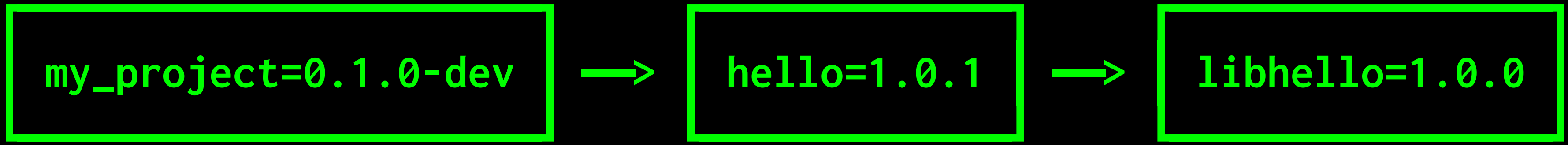
```
$ alr init --bin my_project
$ cd my_project

$ alr build
$ alr run
$ alr edit
```

USE CASES: DEPENDENCIES, USAGE

```
$ alr with hello
Changes to dependency solution:
   + hello    1.0.1 (new)
   + libhello 1.0.0 (new,indirect)
```

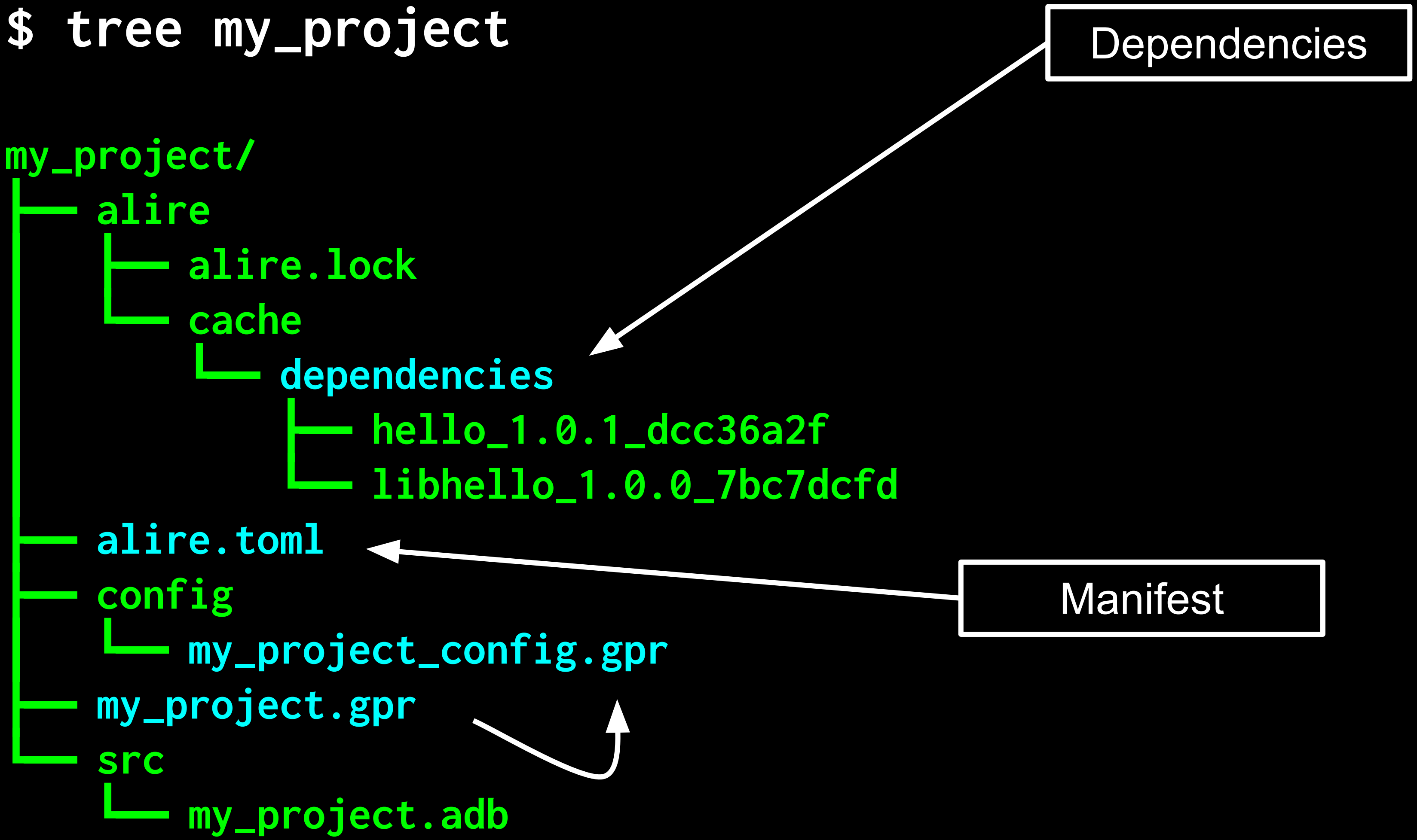
```
$ alr with --tree
my_project=0.1.0-dev
└── hello=1.0.1 (^1.0.1)
    └── libhello=1.0.0 (^1.0)
```

```
$ alr with --graph
```



my_project=0.1.0-dev → hello=1.0.1 → libhello=1.0.0

ON-DISK FILE TREE



```
$ tree my_project

my_project/
├── alire
│   ├── alire.lock
│   └── cache
│       └── dependencies
│           ├── hello_1.0.1_dcc36a2f
│           └── libhello_1.0.0_7bc7dcfd
├── alire.toml
├── config
│   └── my_project_config.gpr
├── my_project.gpr
└── src
    └── my_project.adb
```

Dependencies

Manifest

MANIFEST alire.toml

```
$ alr init --bin my_project && cd my_project && cat alire.toml

name = "my_project"
description = "Shiny new project"
version = "0.1.0-dev"

authors = ["Alejandro R. Mosteo"]
maintainers = ["Alejandro R. Mosteo <alejandro@mosteo.com>"]
maintainers-logins = ["mosteo"]

project-files = [“my_project.gpr”]
executables = ["my_project"]

[[depends-on]]
hello = "^1.0.1"
```

USE CASES: PUBLISHING

Publishing assistant: `alr publish`

- Local validation
  - Metadata
  - Compilation
- Manifest generation
  - Open *pull-request* on the community index
  - Remote validation based on *GitHub Actions*
- `alr index --update-all`
  - Private indexes can also be used
  - Several indexes can be used simultaneously

PUBLISHING EXAMPLE

```
 $ alr publish
 ✓ Local repository is clean
 ✓ Revision exists in local repository (4550aa3)
 Publishing assistant: step 1 of 6: Verify origin URL
 ✓ Origin is of supported kind: GIT
 ✓ Origin is hosted on trusted site: github.com
 Publishing assistant: step 2 of 6: Verify GitHub infrastructure
 ✓ User has a GitHub account: mosteo
 ✓ User has forked the community repository
 ✓ User's fork contains base branch: stable-1.1
 Publishing assistant: step 3 of 6: Deploy sources
 Publishing assistant: step 4 of 6: Build release
 ✓ Sources built successfull
 Publishing assistant: step 5 of 6: User review
      minirest=0.2.0-dev: Minimalist Ada REST client library
      Origin: commit 4550aa3 from https://github.com/mosteo/minirest.git
      Properties:
         Description: Minimalist Ada REST client library
         License: MIT
         Maintainer: alejandro@mosteo.com
         Name: minirest
         Version: 0.2-dev
 Publishing assistant: step 6 of 6: Generate index manifest
 ✓ Manifest generated at ./alire/releases/minirest-0.2.0-dev.toml
 ⓘ Please upload this file to
https://github.com/mosteo/alire-index/upload/stable-1.1/index/mi/minirest
```

USE CASE: STATIC CONFIGURATION

- Neither Ada nor GPRbuild have a pre-processor
  - Difficults non-trivial build processes
  - (But for good reasons)
- Alire has pre-build step
  - Exploited to generate a static file prior to compilation
  - Useable by all dependencies.
- Configuration section in the manifest

CONFIGURATIONS

```
# In crate “my_crate”
[configuration.variables]
Device_Name = {type = "String",  default = "no device name"}
Print_Debug = {type = "Boolean", default = false}
Debug_Level = {type = "Enum", values = ["Debug", "Warn", "Error"], default = "Warn"}
Buffer_Size = {type = "Integer", first = 0, last = 1024, default = 256}
Max_Power   = {type = "Real", first = 0.0, last = 100.0, default = 50.0}
```

```
# In another crate, depending on “my_crate”
[configuration.values]
my_crate.Device_Name = "Custom device name"
my_crate.Print_Debug = true
my_crate.Debug_Level = "Error"
my_crate.Buffer_Size = 42
my_crate.Max_Power   = 42.0
```

```
-- Generated by Alire (*.ads, *.h, *.gpr)
package my_crate_Config is
   Buffer_Size_First : constant := 0; Buffer_Size_Last : constant := 1024;
   Buffer_Size : constant := 42;

   type Debug_Level_Kind is (Debug, Warn, Error);
   Debug_Level : constant Debug_Level_Kind := Error;

   Print_Debug : constant Boolean := True;

   Max_Power_First : constant := 0.0; Max_Power_Last : constant := 100.0;
   Max_Power : constant := 42.0;

   Device_Name : constant String := "Custom device name";
end my_crate_Config;
```

USE CASES: MULTI-CRATE DEVELOPMENT

```
[[pins]]
foo = { version = "1.3.2+bugfix" }
# Specific version override

bar = { path = "../my/bar" }
# Local folder used to fulfill a dependency

baz = { url = "https://github.com/baz.git" }
# Use default branch, updated on `alr update`

wip = { url = "https://gitrepo.com/wip.git" branch="feature" }
# Use given branch, updated on `alr update`

gru = { url = "https://gitlab.com/gru.git" commit="..." }
# Use given commit, never updated
```

Pins always satisfy the corresponding dependency, no matter the version found at the pin location

USE CASES: TESTS, DEMOS

To submit to the community index

```
my_project
├── alire.toml
│
├── examples
│   └── alire.toml
│
├── src
│
└── tests
    └── alire.toml
```

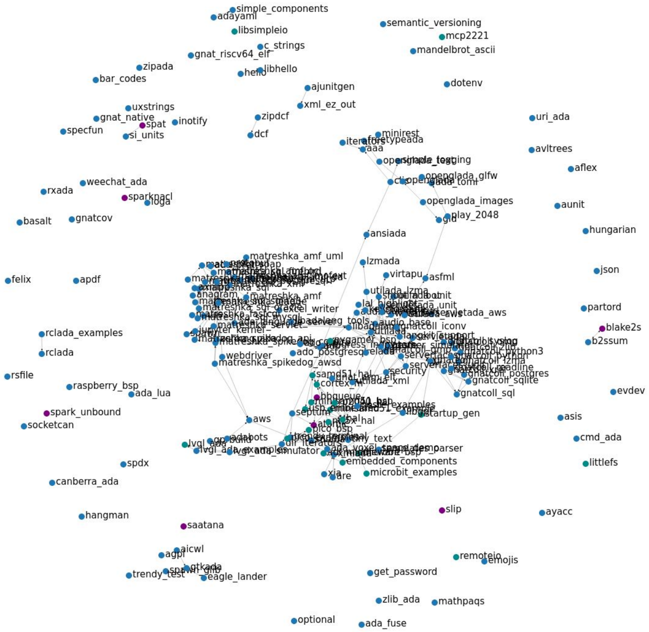
```
executables = [“demo1”, “demo2”]
# Demo binaries, normally not needed
# by dependencies so out of lib build.

[[pins]]
my_crate = { path=”..” }
```

```
[[pins]]
my_crate = { path=”..” }

[[depends-on]]
# Extra dependencies, for testing only
aunit = “*”
```

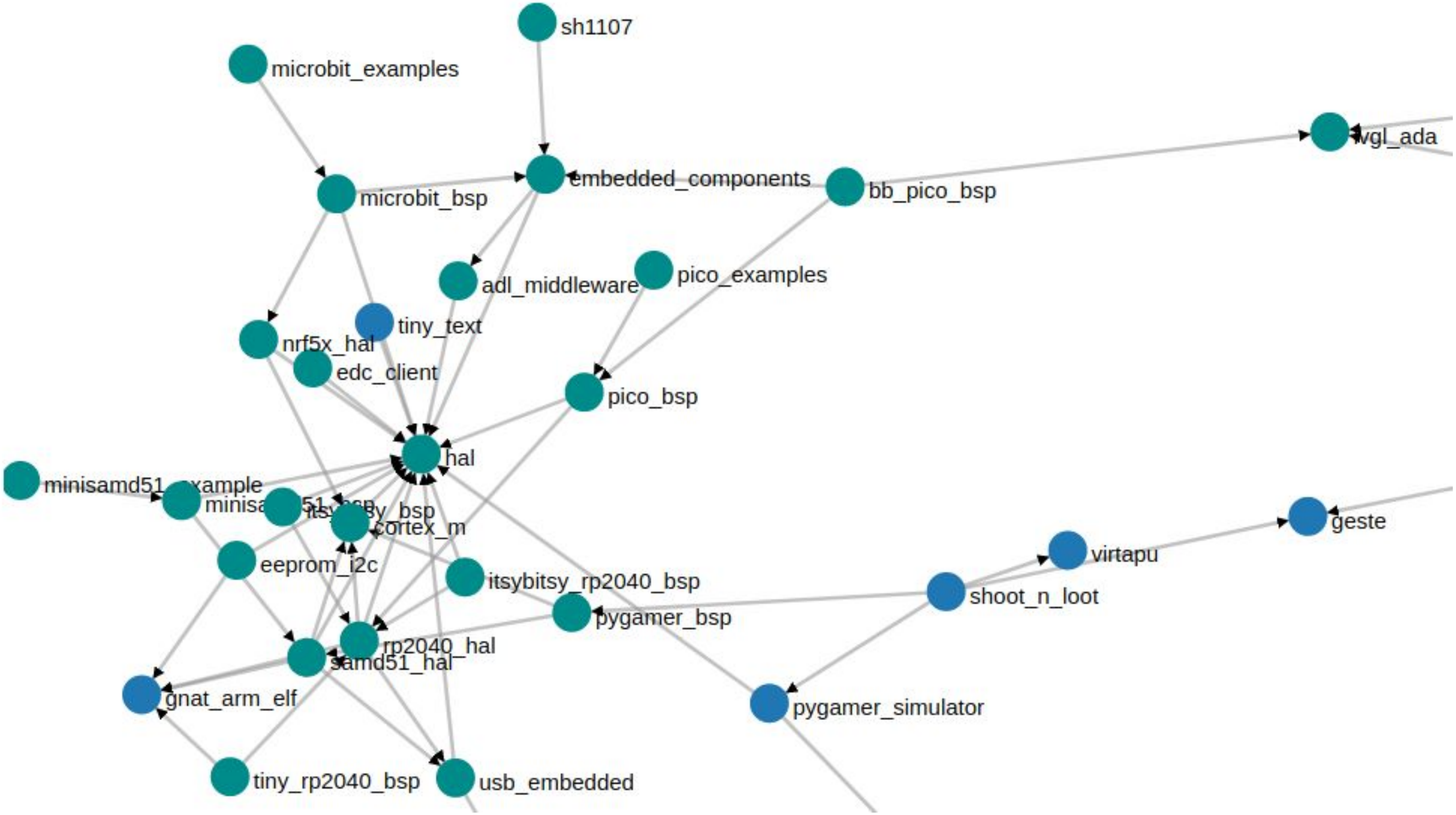
ECOSYSTEM

- 243 Crates
  - 35 Embedded
  - 14 SPARK
- *Top* 10 tags
  - embedded(32) nostd(26) gnatcoll(14) web(12) spark(10) database(10) rp2040(9) bindings(9) sql(8) game(8)
- 22500+ downloads of “`alr`”
  - https://hanadigital.github.io/grev/?user=alire-project&repo=alire

https://alire.ada.dev/network

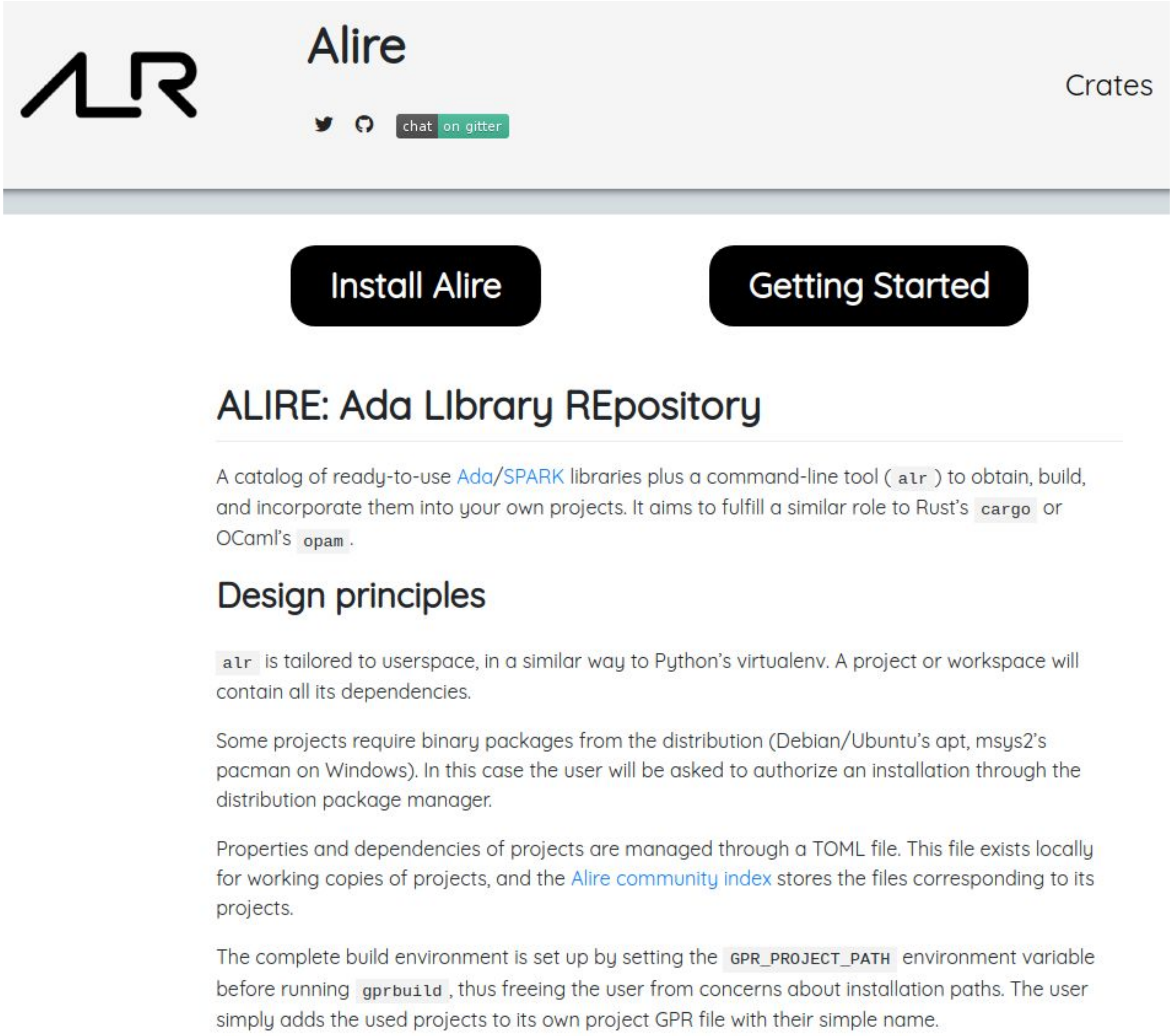


EMBEDDED



Centro Universitario de la Defensa Zaragoza

WEBSITE



ALIRE: Ada LIbrary REpository

A catalog of ready-to-use Ada/SPARK libraries plus a command-line tool (`alr`) to obtain, build, and incorporate them into your own projects. It aims to fulfill a similar role to Rust's `cargo` or OCaml's `opam` .

Design principles

`alr` is tailored to userspace, in a similar way to Python's virtualenv. A project or workspace will contain all its dependencies.

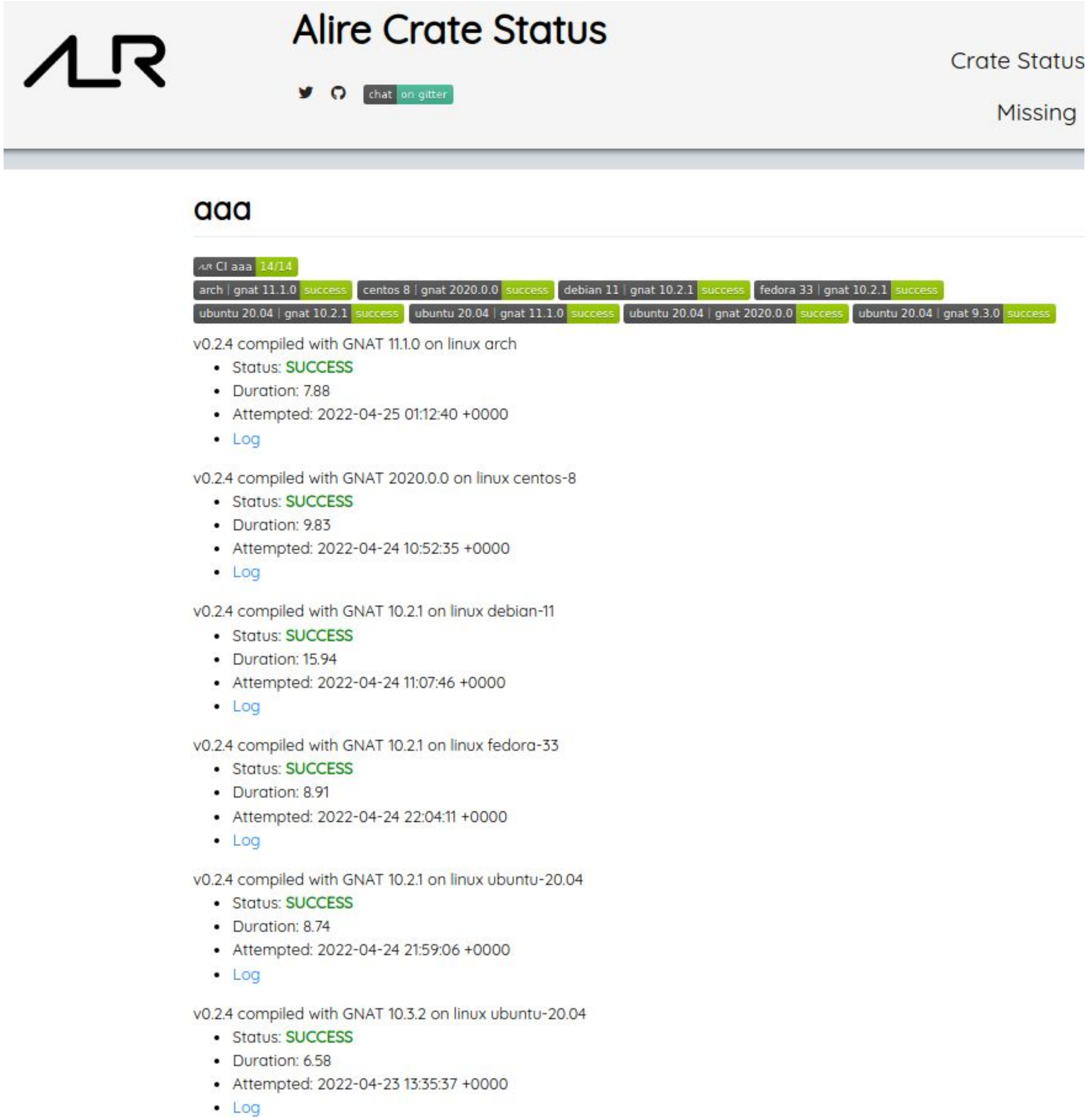
Some projects require binary packages from the distribution (Debian/Ubuntu's apt, msys2's pacman on Windows). In this case the user will be asked to authorize an installation through the distribution package manager.

Properties and dependencies of projects are managed through a TOML file. This file exists locally for working copies of projects, and the Alire community index stores the files corresponding to its projects.

The complete build environment is set up by setting the `GPR_PROJECT_PATH` environment variable before running `gprbuild` , thus freeing the user from concerns about installation paths. The user simply adds the used projects to its own project GPR file with their simple name.

https://alire.ada.dev

CRATE STATUS



aaa

CI aaa 14/14

arch | gnat 11.1.0 success · centos 8 | gnat 2020.0.0 success · debian 11 | gnat 10.2.1 success · fedora 33 | gnat 10.2.1 success

ubuntu 20.04 | gnat 10.2.1 success · ubuntu 20.04 | gnat 11.1.0 success · ubuntu 20.04 | gnat 2020.0.0 success · ubuntu 20.04 | gnat 9.3.0 success

v0.2.4 compiled with GNAT 11.1.0 on linux arch

- Status: SUCCESS
- Duration: 7.88
- Attempted: 2022-04-25 01:12:40 +0000
- Log

v0.2.4 compiled with GNAT 2020.0.0 on linux centos-8

- Status: SUCCESS
- Duration: 9.83
- Attempted: 2022-04-24 10:52:35 +0000
- Log

v0.2.4 compiled with GNAT 10.2.1 on linux debian-11

- Status: SUCCESS
- Duration: 15.94
- Attempted: 2022-04-24 11:07:46 +0000
- Log

v0.2.4 compiled with GNAT 10.2.1 on linux fedora-33

- Status: SUCCESS
- Duration: 8.91
- Attempted: 2022-04-24 22:04:11 +0000
- Log

v0.2.4 compiled with GNAT 10.2.1 on linux ubuntu-20.04

- Status: SUCCESS
- Duration: 8.74
- Attempted: 2022-04-24 21:59:06 +0000
- Log

v0.2.4 compiled with GNAT 10.3.2 on linux ubuntu-20.04

- Status: SUCCESS
- Duration: 6.58
- Attempted: 2022-04-23 13:35:37 +0000
- Log

https://alire-crates-ci.mosteo.com

CONCLUSIONS

- User-level package manager for Ada/SPARK
- Complete solver
  - Will find a solution if there is one
- Support for several use cases:
  - Toolchain installation
  - User interested only on final executables
  - Developer interested on library reuse
    - With support for multi-crate development
  - Embedded developer
    - Cross-compilers
    - Static configuration
  - Open Source contributor
    - Publishing assistant
- Rich and growing ecosystem
  - Public repository oriented
  - Chat at Gitter

THANKS FOR YOUR ATTENTION

🔗 https://github.com/alire-project

https://alire.ada.dev

https://gitter.im/ada-lang/Alire

https://www.reddit.com/r/ada/

✉ amosteo@unizar.es

🐦 @mosteobotic (Alejandro R. Mosteo)

@DesChips (Fabien Chouteau)

Centro Universitario
de la Defensa Zaragoza
cud.unizar.es