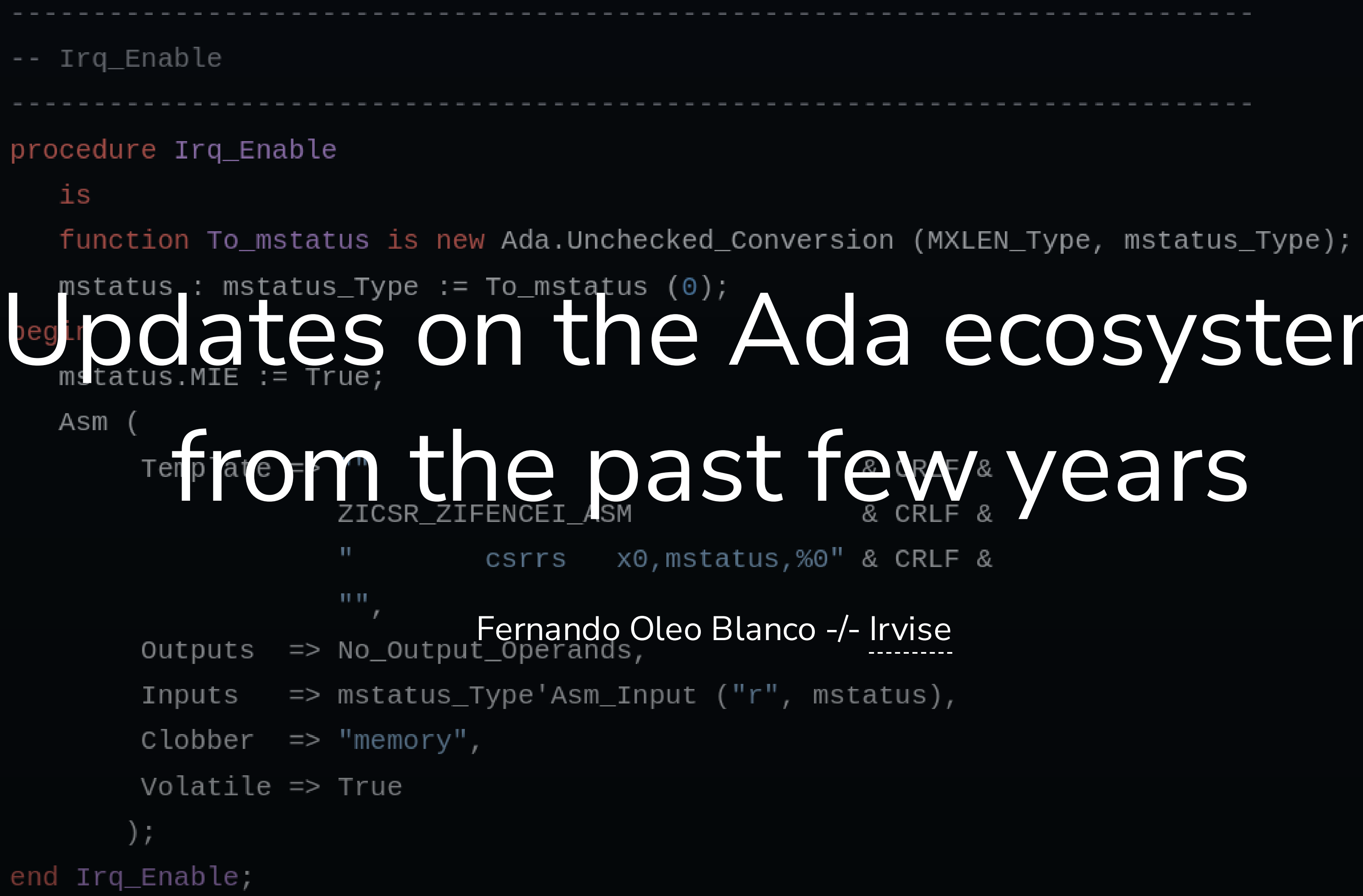
Updates on the Ada ecosystem from the past few years

Fernando Oleo Blanco -/- Irvise



```
------------------------------------------------------------------------------
-- Irq_Enable
------------------------------------------------------------------------------
procedure Irq_Enable
   is
   function To_mstatus is new Ada.Unchecked_Conversion (MXLEN_Type, mstatus_Type);
   mstatus : mstatus_Type := To_mstatus (0);
begin
   mstatus.MIE := True;
   Asm (
        Template => ""                               & CRLF &
                    ZICSR_ZIFENCEI_ASM               & CRLF &
                    "         csrrs   x0,mstatus,%0" & CRLF &
                    "",
        Outputs  => No_Output_Operands,
        Inputs   => mstatus_Type'Asm_Input ("r", mstatus),
        Clobber  => "memory",
        Volatile => True
       );
end Irq_Enable;
```

Table of contents

- Language
  - **Ada 2022**
  - GNAT-LLVM & WASM
  - SPARK improvements
- Tools
- Ada User
  - Ada-Lang, a new home
  - Ada User Society
  - NVIDIA & NSA
- Challenges & Future
- Learn.AdaCore

Ada 2022, new language constructs

```
-- Iterator filters (see the "when" guard)
for E of Some_Array when E /= 0 loop
   Put_Line (E'Image);
end loop;

-- Map/Reduce attribute!
function Factorial (N : Natural) return Natural is
([ for J in 1..N => J ]'Reduce("*", 1));

-- Declare expressions (ala anonymous functions)
Val : Integer := (declare X : constant Integer := Random_Value; begin X + X);

-- Static expression functions (easy comp-time evaluation)
function Half_Size (S : Integer) return Integer is (S / 2) with Static;
```

Ada 2022, new language constructs

```ada
-- Compiler generated 'Image for all datatypes!!! Put your data in ASCII easily
My_Subtype'Image (My_Object); -- It does not need to be a simple numeric value
-- Put_Image attribute for user-defined printing
My_Special_Type is ... with Put_Image => My_Custom_Printing_Function;

-- Delta aggregates
Record_Object := (Record_Object with delta Field1 => X, Field2 => Y);

-- Index parameters and array aggregates
subtype Index_Type is Positive range 1 .. 10;
type Array_Type is array (Index_Type) of Positive;
Squares_Array : Array_Type := [for I in Index_Type => I * I]; -- 1, 4, 9, 16...

-- Assignment target name
Some_Very_Long.And_Complex (Expression) := @ + 1;
```

Ada 2022, new libraries and extensions

- Support for Big_Nums `Ada.Numerics.Big_Numbers.Big_Integers` and `Ada.Numerics.Big_Numbers.Big_Reals`
- Packages for Atomic operations `System.Atomic_Operations.Exchange.{Exchange,Test_And_Set,Integer_Arithmetic,Modular_Arithmetic}`
- Simple support for Variadic C function import
- Jorvik profile
- `Nonblocking` aspect: `function Square (P : Positive) return Positive is (P**2) with Nonblocking;`
- First class `parallel` support (still unsuported by compilers)

Ada 2022, new libraries and extensions

- Support for Big_Nums `Ada.Numerics.Big_Numbers.Big_Integers` and `Ada.Numerics.Big_Numbers.Big_Reals`
- Packages for Atomic operations `System.Atomic_Operations.Exchange.{Exchange,Test_And_Set,Integer_Arithmetic,Modular_Arithmetic}`
- Simple support for Variadic C function import
- Jorvik profile
- `Nonblocking` aspect: `function Square (P : Positive) return Positive is (P**2) with Nonblocking;`
- First class `parallel` support (still unsuported by compilers)

Full list of changes in Ada-Auth’s Overview. GNAT(-LLVM) supports all of them except `parallel`

A friendly reminder that the Ada Reference Manual is open and available without cost!

And there are even more features already available! You can share your opinion too in the ARG

GNAT-LLVM

An LLVM backend with the GNAT frontend!

1. It is basically production ready
2. Work is ongoing to make it available in Alire
    1. Building it manually is a bit painful
3. Supports most of the LLVM features
4. It is mainly being used for WASM
    1. See AdaWebPack
5. CHERI/Morello is also being explored

WA

WEBASSEMBLY

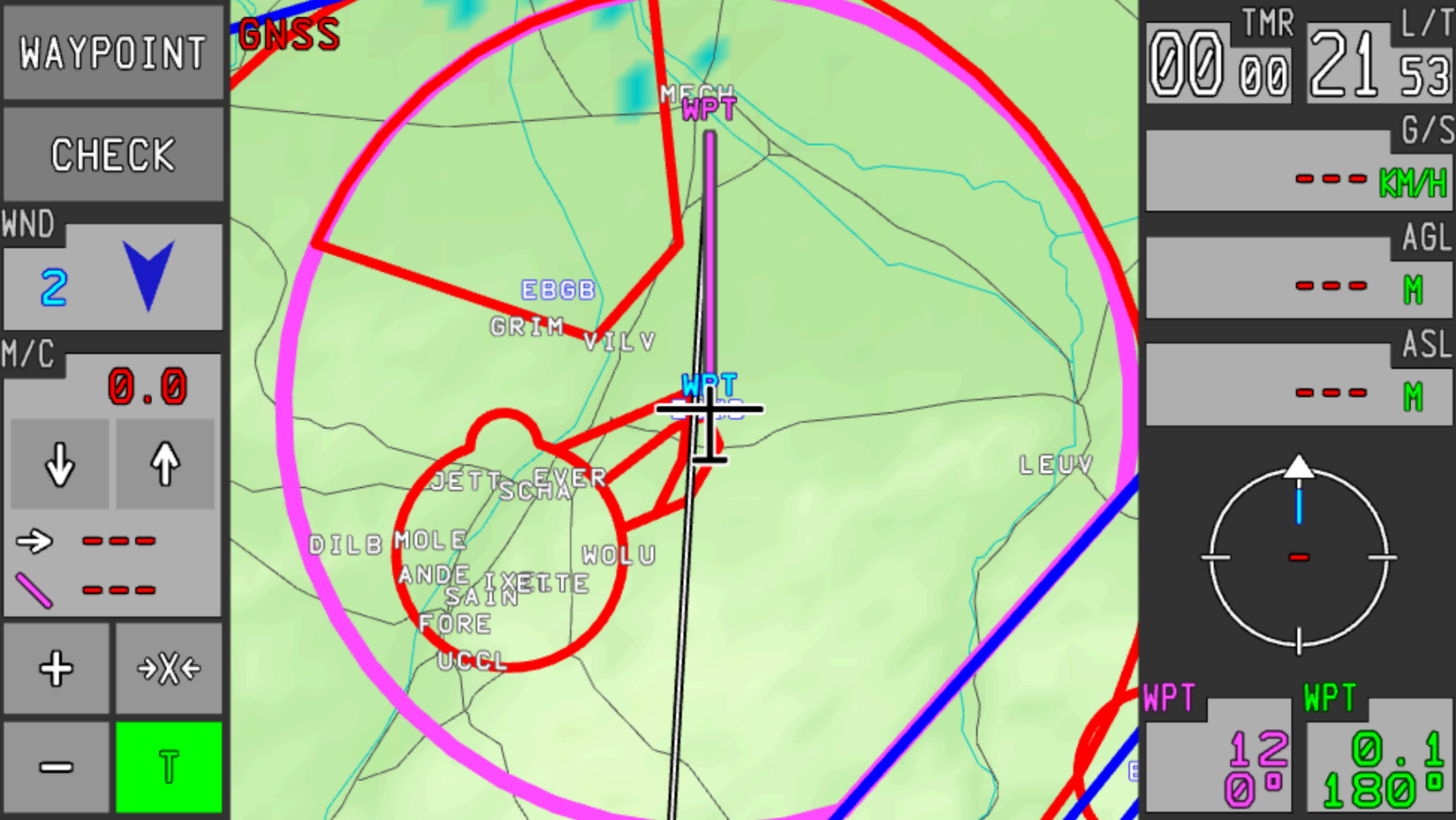
GNAT-LLVM

An LLVM backend with the GNAT frontend!

1. It is basically production ready
2. Work is ongoing to make it available in Alire
    1. Building it manually is a bit painful
3. Supports most of the LLVM features
4. It is mainly being used for WASM
    1. See AdaWebPack
5. CHERI/Morello is also being explored

A demo of G-NAV, an aviation software copiled to WASM, is provided in the next slide.

Watch a technical presentation on its architecture


WAYPOINT
CHECK
WND
2
M/C
0.0

+
→X←
−
T
GNSS
MECH
WPT
EBGB
GRIM
VILV
WPT
LEUV
JETT
SCHA
EVER
DILB
MOLE
WOLU
ANDE
IXEL
ETTE
SAIN
FORE
UCCL
TMR
00 00
L/T
21 53
G/S
--- KM/H
AGL
--- M
ASL
--- M
WPT
120°
WPT
0.1
180°

SPARK

We used to say… "SPARK is a formally verified subset of Ada…"

But SPARK is basically all of Ada nowadays

1. Since 2019, SPARK gained Rust’s borrow checker powers (memory ownership model)
    1. We can finally do dynamic memory management in SPARK!
    2. No need for lifetime annotations (inferred and proven)
2. Since 2024, we can use exceptions! See `Exceptional_Cases`
3. Improved termination analysis
4. Provers have become quite powerful
    1. Updated to `CVC5` and new versions of Z3 and Alt-Ergo
    2. Can theoretically use any Why3 supported prover

SPARK

We used to say... "SPARK is a formally verified subset of Ada..."

But SPARK is basically all of Ada nowadays

1. Since 2019, SPARK gained Rust’s borrow checker powers (memory ownership model)
   1. We can finally do dynamic memory management in SPARK!
   2. No need for lifetime annotations (inferred and proven)
2. Since 2024, we can use exceptions! See `Exceptional_Cases`
3. Improved termination analysis
4. Provers have become quite powerful
   1. Updated to `CVC5` and new versions of Z3 and Alt-Ergo
   2. Can theoretically use any Why3 supported prover

We now have SPARKlib, a large formally verified library

Vectors, Linked-lists, Sets, Maps and their *_Unbounded variants; BigNums. Also a large part of the standard library!

Tools - Alire

Ada/SPARK’s package manager, modelled after Cargo

In March 2024 Alire v2.0 was released

- Allows local/user installation of crates
  - It behaves similar to a system package manager
- Dependencies are now shared among projects
  - Reduces disk space and build times
- Improvements to `alr toolchain`
- Improvements for OSX ARM64
- QoL improvements
  - Pinning crates, remote crates,

Tools - Alire

Ada/SPARK’s package manager, modelled after Cargo

In March 2024 Alire v2.0 was released

- Allows local/user installation of crates
  - It behaves similar to a system package manager
- Dependencies are now shared among projects
  - Reduces disk space and build times
- Improvements to `alr toolchain`
- Improvements for OSX ARM64
- QoL improvements
  - Pinning crates, remote crates,

Things to come

- Native toolchains for ARM64 hosts (OSX and Linux)
  - Build system is dependent on Github’s CI availability
- GNAT-LLVM should arrive soonish

Feedback and PRs are welcomed!

Tools - Licensing change from AdaCore and ecosystem changes

Having the FSF build of GNAT (LGPL) and AdaCore's (GPL) was confusing

- AdaCore's distribution is now stopped, yay!
- The FSF build of GCC/GNAT has all the features, yay!
- All the packages are now available through Alire, yay!

Tools - Licensing change from AdaCore and ecosystem changes

Having the FSF build of GNAT (LGPL) and AdaCore's (GPL) was confusing

- AdaCore's distribution is now stopped, yay!
- The FSF build of GCC/GNAT has all the features, yay!
- All the packages are now available through Alire, yay!

AdaCore has changed its default license to Apache v2

New libraries will be released under Apache v2 (more permissive) instead of *GPL

Relicense some libraries to Apache v2: **SPARKlib**, VSS, spawn, libadalang, langkit, gpr, etc

Tools - Licensing change from AdaCore and ecosystem changes

Having the FSF build of GNAT (LGPL) and AdaCore’s (GPL) was confusing

- AdaCore’s distribution is now stopped, yay!
- The FSF build of GCC/GNAT has all the features, yay!
- All the packages are now available through Alire, yay!

AdaCore has changed its default license to Apache v2

New libraries will be released under Apache v2 (more permissive) instead of *GPL

Relicense some libraries to Apache v2: **SPARKlib**, VSS, spawn, libadalang, langkit, gpr, etc

Updates to GNATStudio, Ada_Language_Server and the VS Code extension!

Tools - Lots of updates!

SweetAda, an incredible embedded development framework supporting tons of architectures and boards. Also check out bb-runtimes!

New OS releases: Muen v1.1, HiRTOS v2 and FreeRTOS-Ada both with RISC-V enablement, Ironclad

New Simple Components updates

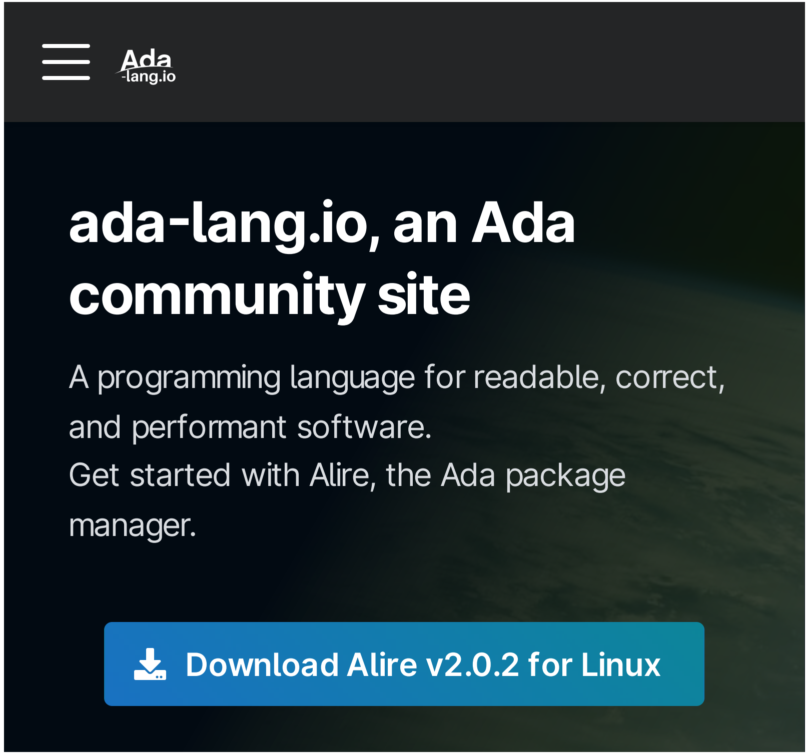
GNATColl, libadalang and other components had their v25 release

VulkanAda now supports Vulkan 1.3

GHDL v4.1, a VHDL compiler written in Ada

... and plenty more!

Ada-Lang



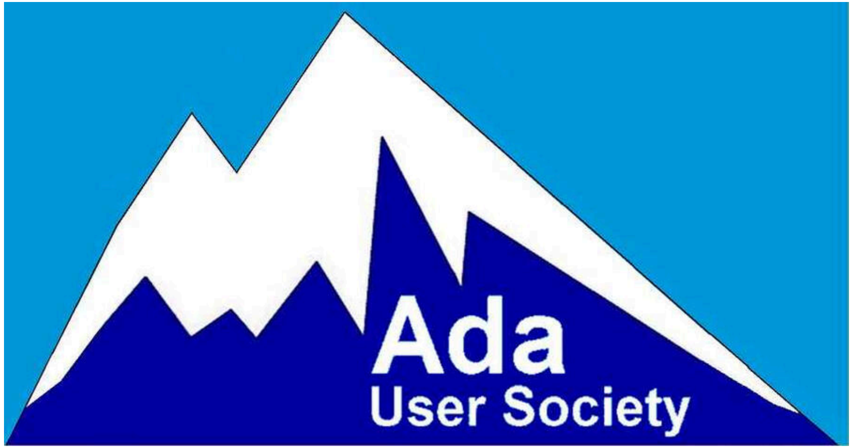
Ada -lang.io

ada-lang.io, an Ada community site

A programming language for readable, correct, and performant software.
Get started with Alire, the Ada package manager.

A community home for Ada: tutorials, get-started, forums, resources...

Ada User Society



New World-wide association for Ada users

Current primary task: to ensure continued evolution of Ada as an ISO standard

The NSA lists Ada as a recommended memory safe language

Is Ada really memory safe?

The NSA lists Ada as a recommended memory safe language

Is Ada really memory safe?

Short answer, **almost**, for 100% memory safety use Ada/SPARK instead, which has the borrow checker implemented

The NSA lists Ada as a recommended memory safe language

Is Ada really memory safe?

Short answer, **almost**, for 100% memory safety use Ada/SPARK instead, which has the borrow checker implemented

Long answer...

- Robust syntax and standard
  - Manual memory management is rarely needed
  - When needed, there is syntax to help you get it right
  - Memory/Storage pools
  - Runtime checks
- A basic standard library
  - Queues, Maps, Unbounded_Strings, etc. All memory safe

The NSA lists Ada as a recommended memory safe language

Is Ada really memory safe?

Short answer, **almost**, for 100% memory safety use Ada/SPARK instead, which has the borrow checker implemented

Long answer...

- Robust syntax and standard
  - Manual memory management is rarely needed
  - When needed, there is syntax to help you get it right
  - Memory/Storage pools
  - Runtime checks
- A basic standard library
  - Queues, Maps, Unbounded_Strings, etc. All memory safe

NVIDIA and Ada, what!?

NVIDIA — Securing the Future of ...
NVIDIA
SPARK / ADA JOURNEY
TO ADOPTION
Daniel Rohrer, NVIDIA - November 21, 2019

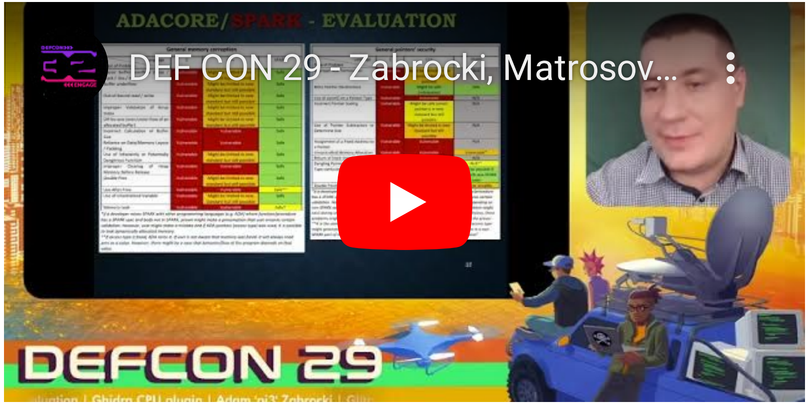
- Using it at least since 2019
- In high safety critical software
- Ada’s usage has grown since
- SPARK is heavily used
- Very good results...
- (± PROPIETARY) program CUDA from Ada
  - Here is a video showcase

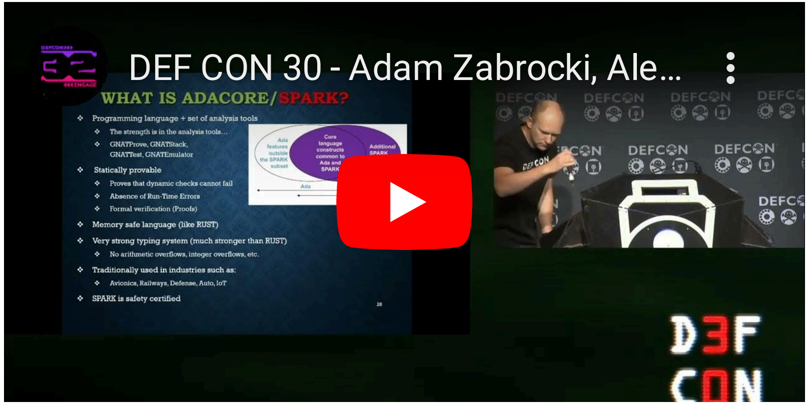
NVIDIA and Ada, what!?

NVIDIA — Securing the Future of ...
NVIDIA
SPARK / ADA JOURNEY TO ADOPTION
Daniel Rohrer, NVIDIA - November 21, 2019

- Using it at least since 2019
- In high safety critical software
- Ada's usage has grown since
- SPARK is heavily used
- Very good results...
- (± PROPIETARY) program CUDA from Ada
  - Here is a video showcase

NVIDIA's own safety team


DEF CON 29 - Zabrocki, Matrosov...
ADACORE/SPARK - EVALUATION
DEFCON 29



WHAT IS ADACORE/SPARK?

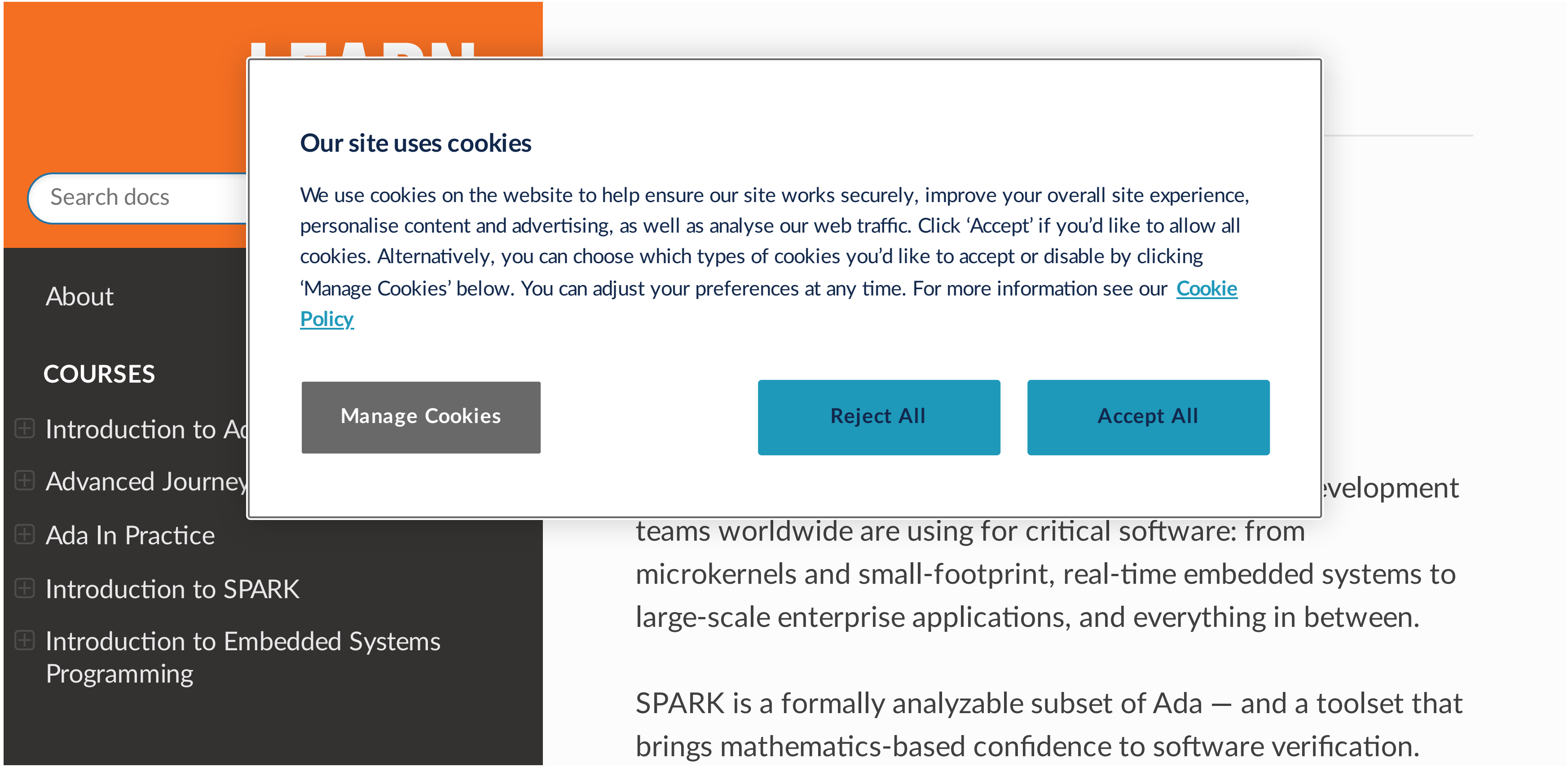
- Programming language + set of analysis tools
  - The strength is in the analysis tools...
  - GNATProve, GNATStack, GNATTest, GNATEmulator
- Statically provable
  - Proves that dynamic checks cannot fail
  - Absence of Run-Time Errors
  - Formal verification (Proofs)
- Memory safe language (like RUST)
- Very strong typing system (much stronger than RUST)
  - No arithmetic overflows, integer overflows, etc.
- Traditionally used in industries such as:
  - Avionics, Railways, Defense, Auto, IoT
- SPARK is safety certified

Ada features outside the SPARK subset
Core language constructs common to Ada and SPARK
Additional SPARK
Ada
26
DEFCON
D3F CON

Challenges ahead!

- Grow the community
  - And make it more pro-active!
- Grow the quality and accesibility of the documentation
- Improve other forms of media about Ada
- IMPROVE THE TOOLS
  - Some parts of the Ada experience are less than ideal
  - There are quite a few changes taking place
- Grow the libraries that are available
- Prepare the language for its next edition.
  - See what is being cooked (and participate)

Learn.AdaCore



Our site uses cookies

We use cookies on the website to help ensure our site works securely, improve your overall site experience, personalise content and advertising, as well as analyse our web traffic. Click 'Accept' if you'd like to allow all cookies. Alternatively, you can choose which types of cookies you'd like to accept or disable by clicking 'Manage Cookies' below. You can adjust your preferences at any time. For more information see our Cookie Policy

evelopment teams worldwide are using for critical software: from microkernels and small-footprint, real-time embedded systems to large-scale enterprise applications, and everything in between.

SPARK is a formally analyzable subset of Ada — and a toolset that brings mathematics-based confidence to software verification.

Thank you!

Questions?

Fernando Oleo Blanco -/- irvise@irvise.xyz

“In strong typing we trust.”

— Ada Users