Fast, Functional, Flexible Programming with OCaml

Gemma Gordon (speaker),
Anil Madhavapeddy (speaker), with
contributions from the OCaml Labs team of
David Allsopp, Stephen Dolan, Jeremy Yallop,
Thomas Gazagnaire, and KC Sivaramakrishnan

QCon London
March 2018

"Old Languages Made New"

- **Why ML is important** among programming languages
- What uses have driven **the recent resurgence**?
- What we are doing to **grow the community**!
- A look forward to the **exciting developments** coming.

A Brief Intro to ML

Background

OCaml is a industrial grade, mixed-style functional and imperative language.

From the ML heritage of programming languages *(Milner, Stanford/Edinburgh/Cambridge)*

Originally the metalanguage for LCF, a theorem prover developed in the 1980s.

Caml: *1987*, Caml Light: *1990*, OCaml: *1997*

https://dev.realworldocaml.org/00-prologue.html

OCaml: a quick primer

```
let x = 1
let y = "world"

let fn a =
  Printf.sprintf "%s %d %s" a x y

let _ = print_endline (fn "hello")
```

```
val x : int
val y : string
val fn : string -> string
```

OCaml: a quick primer

```
let x = 1
let y = "world"

let fn a =
  Printf.sprintf "%s %d %s" a x y

let _ = print_endline (fn "hello")
```

Variable names bound with “let”

fn takes a string argument and returns a string

Then we just print the result of calling fn

OCaml: a quick primer

```
let x = 1
let y = "world"

let fn a =
  Printf.sprintf "%s %d %s" a x y

let _ = print_endline (fn "hello")
```

Every value (functions or constants) has a static type

```
val x : int
val y : string
val fn : string -> string
```

Mixing types up results in a compile time error

```
# x + y;;
Error: This expression has type string but
an expression was expected of type int
```

OCaml: Features

- **Language**
- Pattern Matching
- Algebraic Data Types
- Type Inference
- First Class Functions
- Static type checking
- Parametric Polymorphism

```
type t =
     Apple
   | Orange
   | Pear

let is_apple fruit =
  match fruit with
   | Apple -> true
   | Orange -> false
```

```
Warning 8: this pattern-
matching is not exhaustive
Here is an example of a
case that is not matched:
Pear
```

OCaml: Features

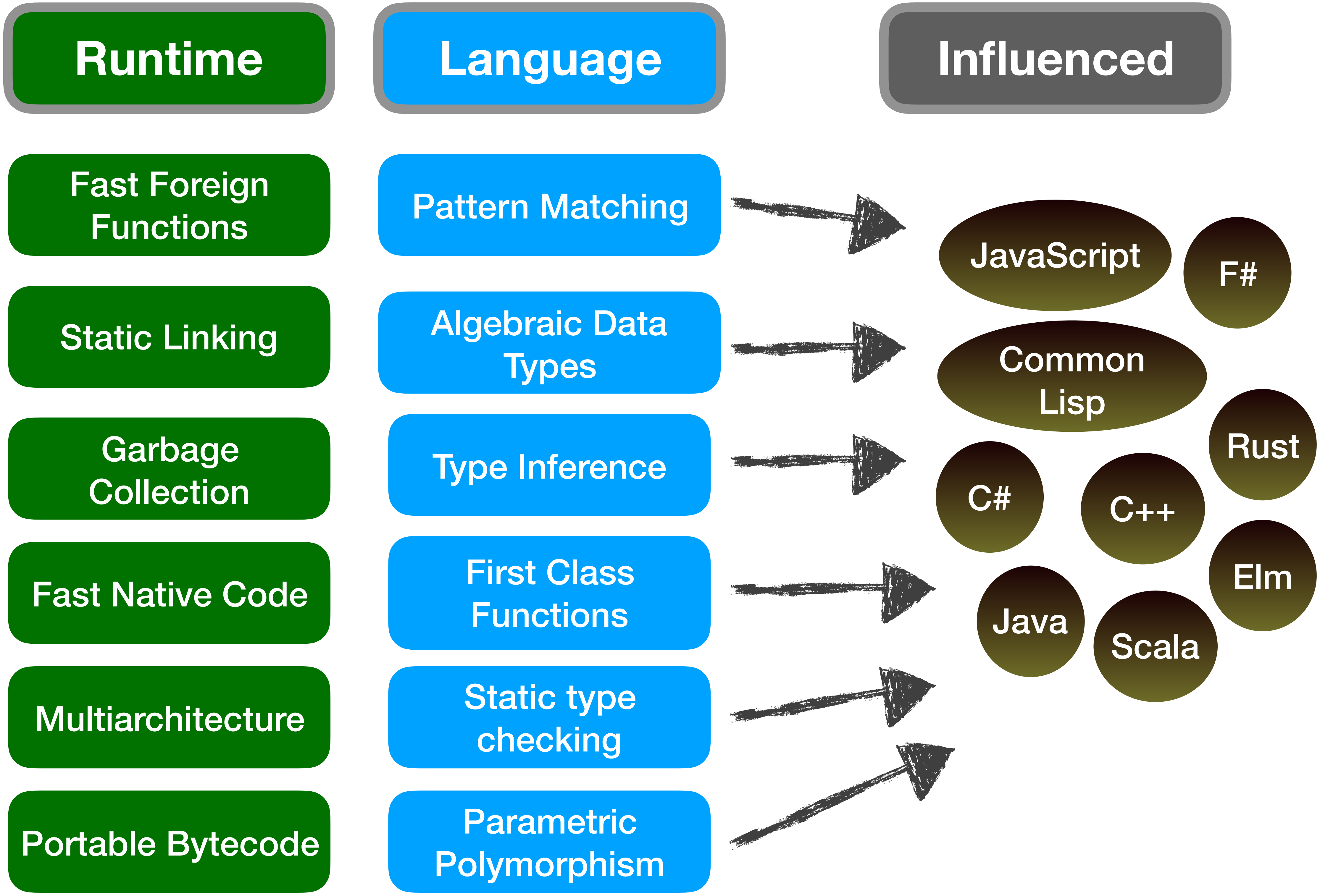
Runtime

- Fast Foreign Functions
- Static Linking
- Garbage Collection
- Fast Native Code
- Multiarchitecture
- Portable Bytecode

Language

- Pattern Matching
- Algebraic Data Types
- Type Inference
- First Class Functions
- Static type checking
- Parametric Polymorphism

OCaml: Features



Runtime

Fast Foreign Functions

Static Linking

Garbage Collection

Fast Native Code

Multiarchitecture

Portable Bytecode

Language

Pattern Matching

Algebraic Data Types

Type Inference

First Class Functions

Static type checking

Parametric Polymorphism

Influenced

JavaScript

F#

Common Lisp

Rust

C#

C++

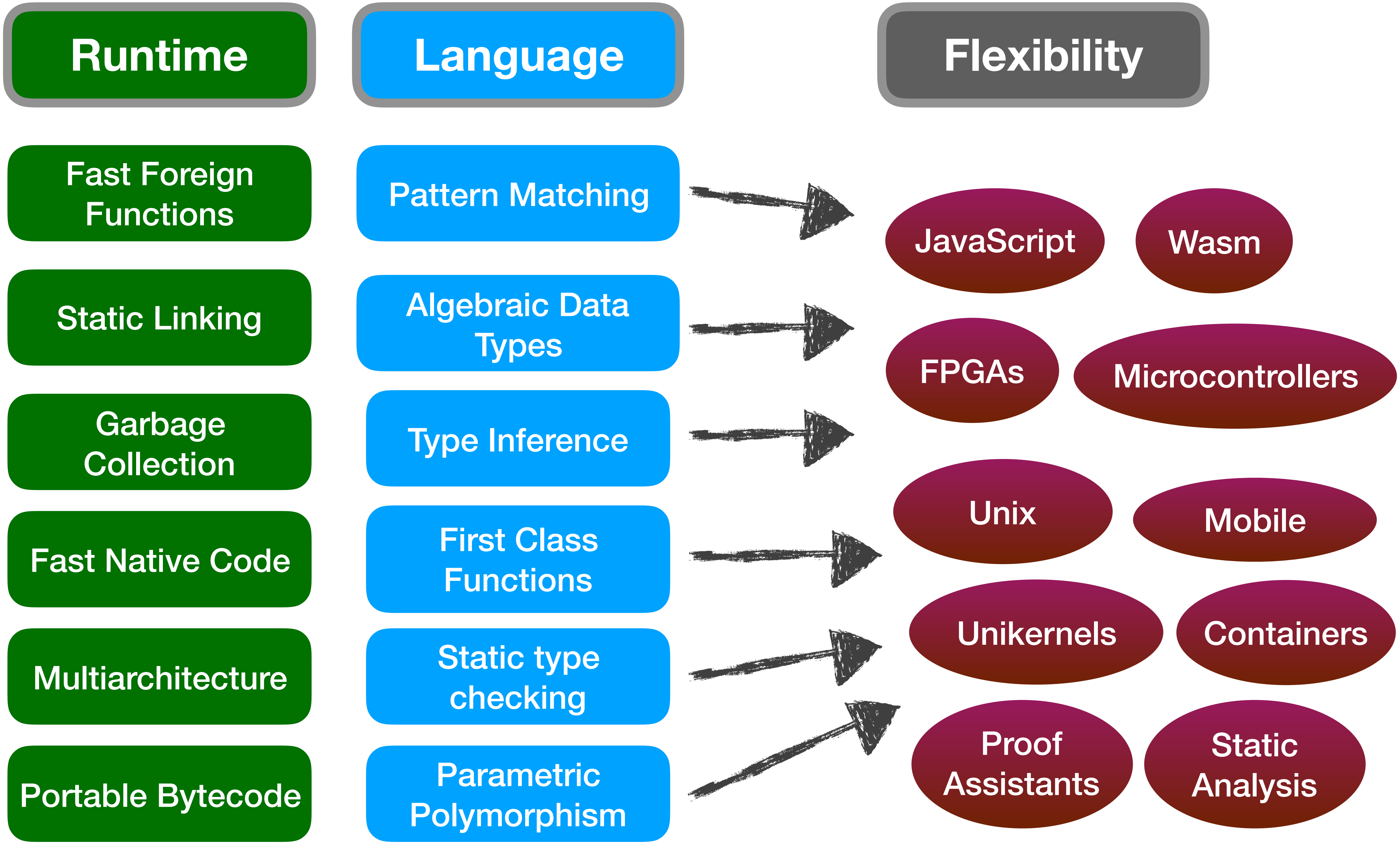
Elm

Java

Scala

What has driven the resurgence of ML?

OCaml: Features



Runtime

- Fast Foreign Functions
- Static Linking
- Garbage Collection
- Fast Native Code
- Multiarchitecture
- Portable Bytecode

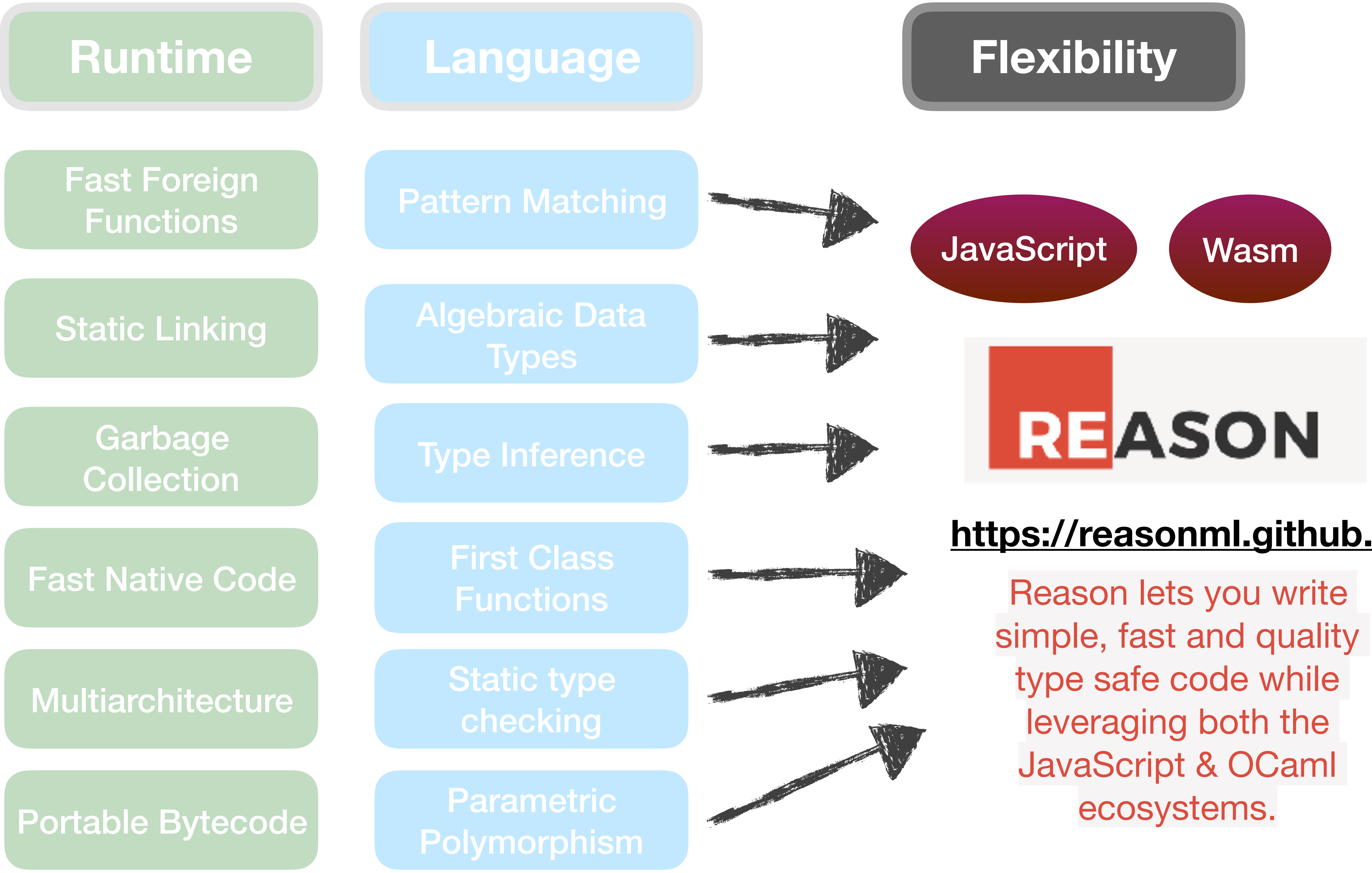
Language

- Pattern Matching
- Algebraic Data Types
- Type Inference
- First Class Functions
- Static type checking
- Parametric Polymorphism

Flexibility

- JavaScript
- Wasm
- FPGAs
- Microcontrollers
- Unix
- Mobile
- Unikernels
- Containers
- Proof Assistants
- Static Analysis

OCaml: Features



Runtime

Fast Foreign Functions

Static Linking

Garbage Collection

Fast Native Code

Multiarchitecture

Portable Bytecode

Language

Pattern Matching

Algebraic Data Types

Type Inference

First Class Functions

Static type checking

Parametric Polymorphism

Flexibility

JavaScript

Wasm

REASON

https://reasonml.github.io

Reason lets you write simple, fast and quality type safe code while leveraging both the JavaScript & OCaml ecosystems.

OCaml: Features

Runtime

Language

Flexibility

OCaPIC: PIC microcontrollers programmed in OCaml

Static Linking

Algebraic Data Types

FPGAs

Microcontrollers

Garbage Collect

Type Inference

Fast Native

Multiarchit

Portable By

Polymorphism

ORCONF2015

ORCONF 2015

Writing hardware in OCaml,
Running OCaml in hardware

Andrew Ray

HardCaml is a structural hardware design DSL embedded in Ocaml. The library can be used for front end design tasks up to the synthesis stage where a VHDL or Verilog netlist is generated. Libraries for fast simulation using LLVM, waveform viewing and co-simulation with Icarus Verilog are provided.

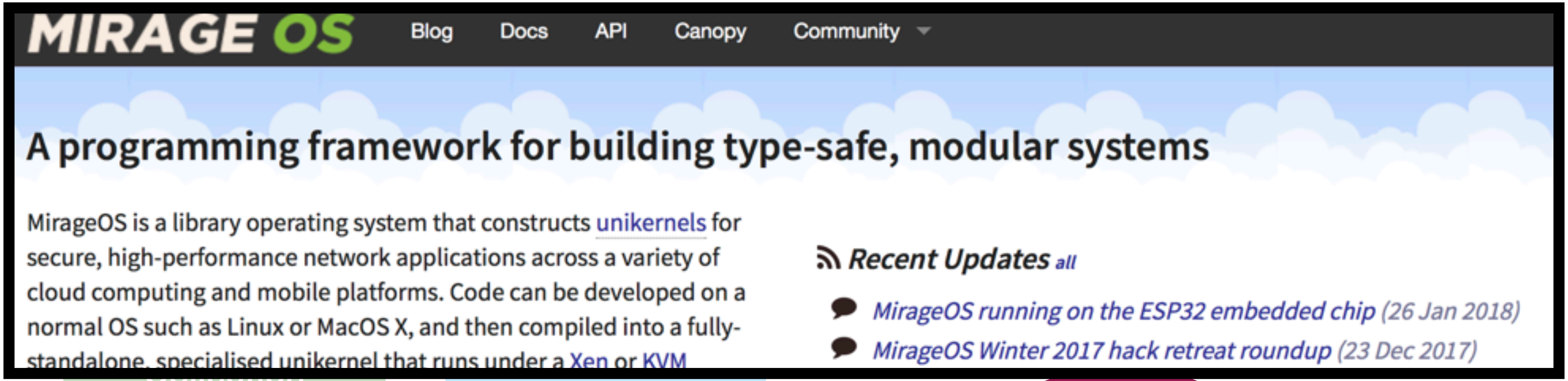
HardCaml-RiscV is a simple pipelined RV32I core, targetted towards a FPGA implementation and built with HardCaml.

OCaml: Features

Runtime

Language

Flexibility



A programming framework for building type-safe, modular systems

MirageOS is a library operating system that constructs unikernels for secure, high-performance network applications across a variety of cloud computing and mobile platforms. Code can be developed on a normal OS such as Linux or MacOS X, and then compiled into a fully-standalone, specialised unikernel that runs under a Xen or KVM

Recent Updates all

- *MirageOS running on the ESP32 embedded chip* (26 Jan 2018)
- *MirageOS Winter 2017 hack retreat roundup* (23 Dec 2017)

Collection

Fast Native Code

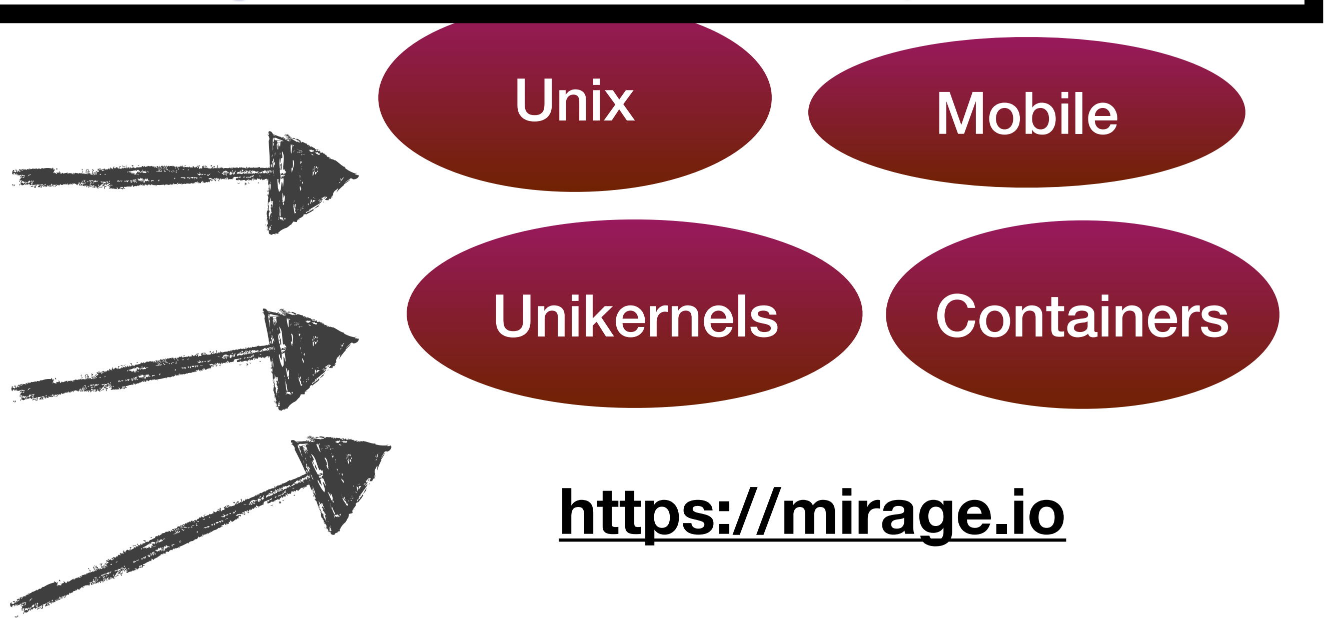
First Class Functions

Multiarchitecture

Static type checking

Portable Bytecode

Parametric Polymorphism



https://mirage.io

OCaml: Features

Runtime

Fast Foreign Functions

Static Linking

Garbage Collection

Fast Native Code

Multiarchitecture

Portable Bytecode

Language

Pattern Matching

Algeb
T

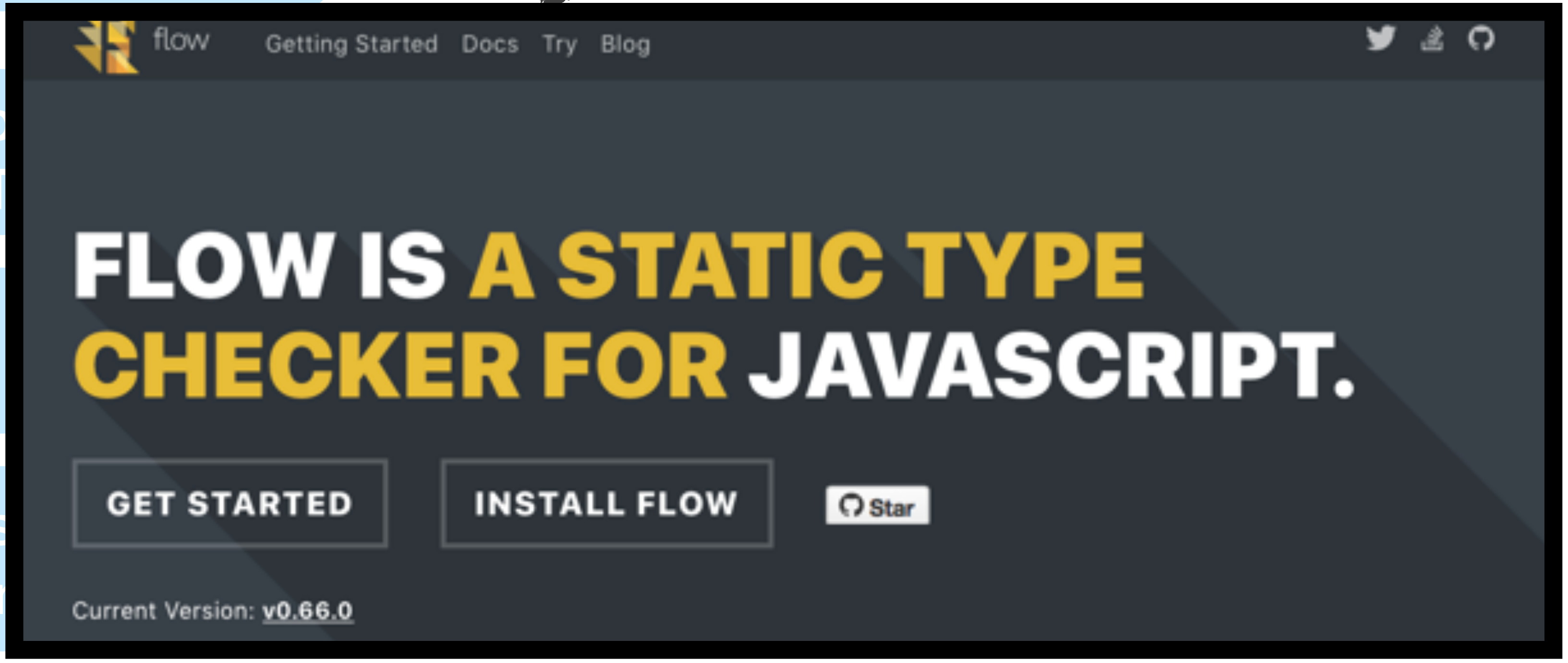
Type

Firs
Fu

Static type checking

Parametric Polymorphism

Flexibility



FLOW IS A STATIC TYPE CHECKER FOR JAVASCRIPT.

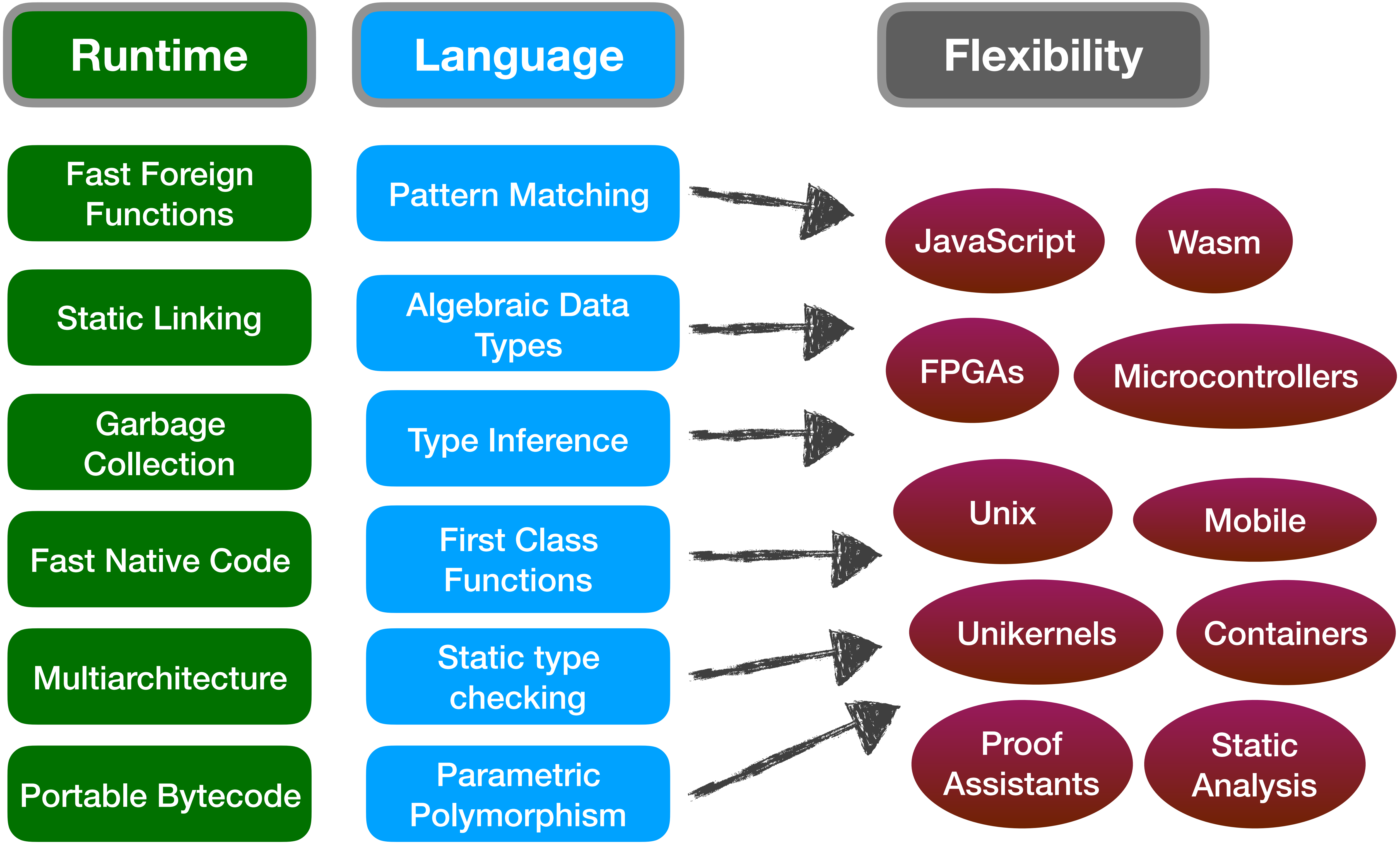
Current Version: v0.66.0

https://flow.org

Proof Assistants

Static Analysis

OCaml: Features



Runtime

- Fast Foreign Functions
- Static Linking
- Garbage Collection
- Fast Native Code
- Multiarchitecture
- Portable Bytecode

Language

- Pattern Matching
- Algebraic Data Types
- Type Inference
- First Class Functions
- Static type checking
- Parametric Polymorphism

Flexibility

- JavaScript
- Wasm
- FPGAs
- Microcontrollers
- Unix
- Mobile
- Unikernels
- Containers
- Proof Assistants
- Static Analysis

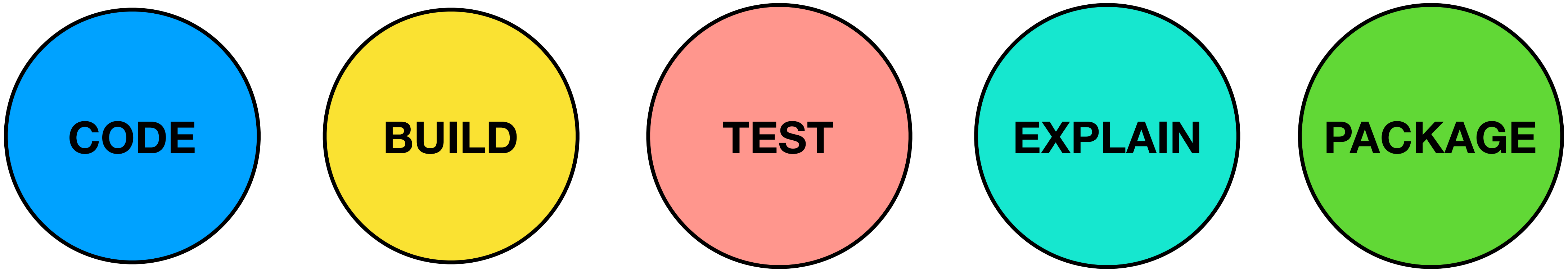
OCaml: Features

Runtime	Language
Fast Foreign Functions	Pattern Matching
Static Linking	Algebraic Data Types
Garbage Collection	Type Inference
Fast Native Code	First Class Functions
Multiarchitecture	Static type checking
Portable Bytecode	Parametric Polymorphism

The working programmer needs a lot more than just a nice language these days!

OCaml: Features

Runtime	Language	Ecosystem
Fast Foreign Functions	Pattern Matching	Libraries?
Static Linking	Algebraic Data Types	Packages?
Garbage Collection	Type Inference	Sharing?
Fast Native Code	First Class Functions	Editors?
Multiarchitecture	Static type checking	Documentation?
Portable Bytecode	Parametric Polymorphism	Tests?



CODE

BUILD

TEST

EXPLAIN

PACKAGE

What is the OCaml Platform?

The OCaml Platform combines the OCaml compiler toolchain with a coherent set of tools for build, documentation, testing and IDE integration.

The project is a collaborative effort across the OCaml community, tied together by the OCaml Labs group in Cambridge, and OCamlPro in Paris.

The requirements of the Platform are guided by large industrial users such as Jane Street, Citrix, Docker, Facebook, Microsoft and LexiFi, as well as accrued feedback from the opam project.

CODE BUILD TEST EXPLAIN PACKAGE

What is the OCaml Platform?

Users with production deployments driving growth

The OCaml Platform combines the OCaml compiler toolchain with a coherent set of tools for build, documentation, testing and IDE integration.

The project is a collaborative effort across the OCaml community, tied together by the OCaml Labs group in Cambridge, and OCamlPro in Paris.

The requirements of the Platform are guided by large industrial users such as Jane Street, Citrix, Docker, Facebook, Microsoft and LexiFi, as well as accrued feedback from the opam project.

CODE

BUILD

TEST

EXPLAIN

PACKAGE

The requirements of the Platform are guided by large industrial users such as **Jane Street**, Citrix, Docker, Microsoft, Facebook and LexiFi, as well as accrued feedback from the opam project.

Trading Platform
(trillions of dollars)

CODE

BUILD

TEST

EXPLAIN

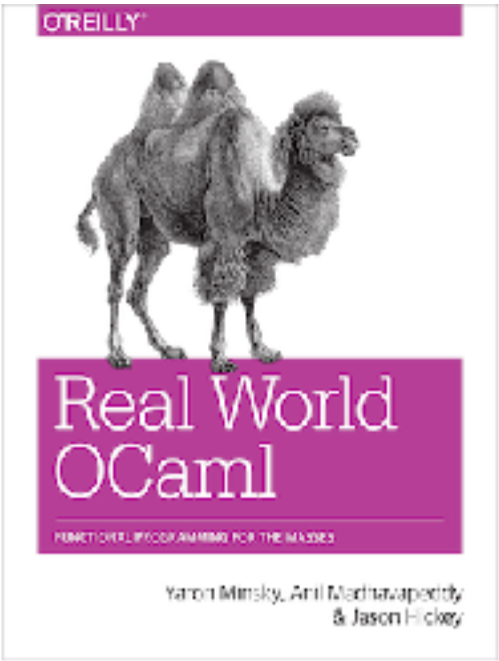
PACKAGE

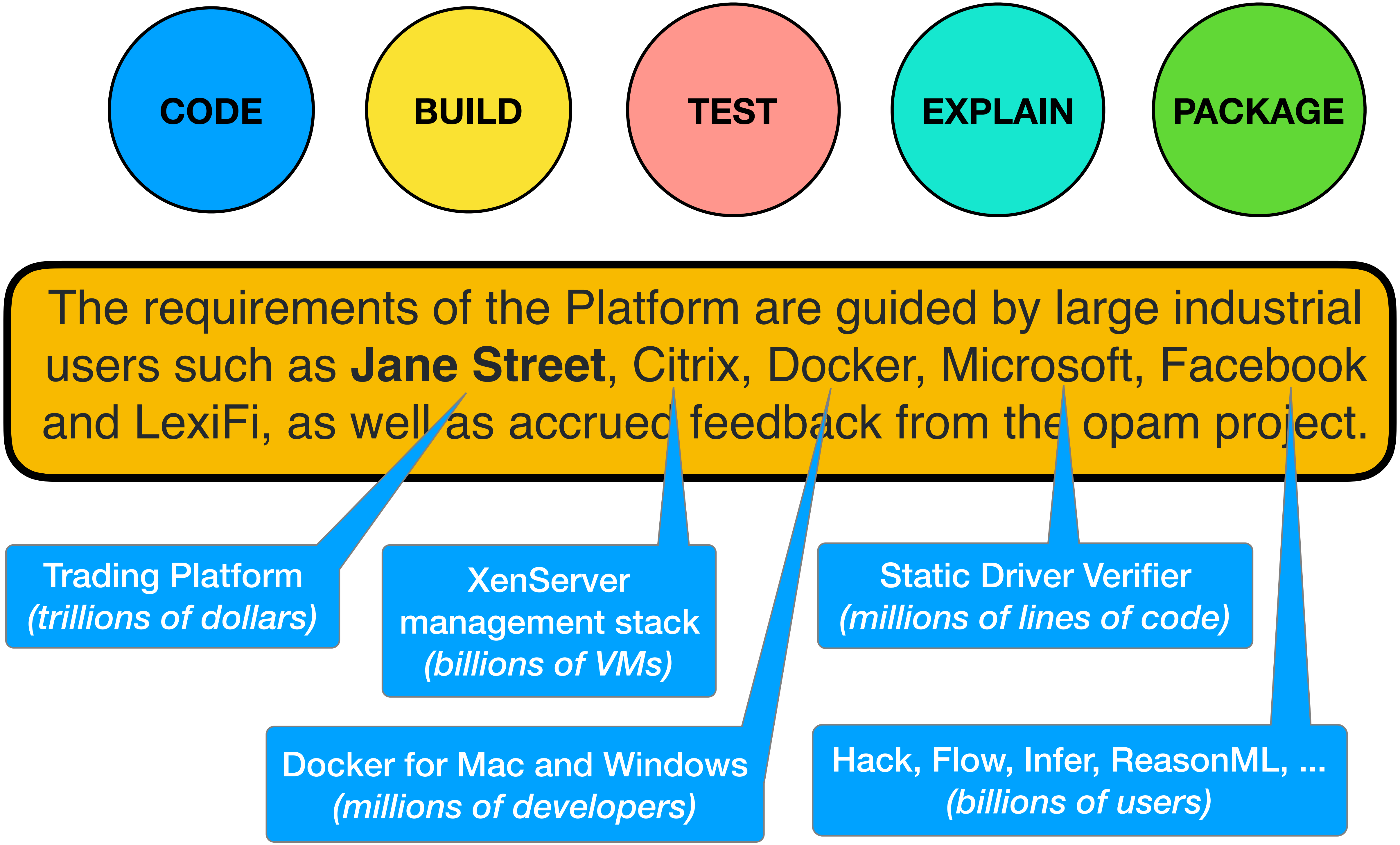
The requirements of the Platform are guided by large industrial users such as **Jane Street**, Citrix, Docker, Microsoft, Facebook and LexiFi, as well as accrued feedback from the opam project.

Trading Platform
(trillions of dollars)

Published millions of lines of production OCaml basic libraries as open source code

Real World OCaml
O'Reilly Associates
dev.realworldocaml.org


O'REILLY
Real World OCaml
FUNCTIONAL PROGRAMMING FOR THE MASSES
Yaron Minsky, Anil Madhavapeddy
& Jason Hickey



CODE

BUILD

TEST

EXPLAIN

PACKAGE

The requirements of the Platform are guided by large industrial users such as **Jane Street**, Citrix, Docker, Microsoft, Facebook and LexiFi, as well as accrued feedback from the opam project.

Trading Platform
(trillions of dollars)

XenServer management stack
(billions of VMs)

Static Driver Verifier
(millions of lines of code)

Docker for Mac and Windows
(millions of developers)

Hack, Flow, Infer, ReasonML, ...
(billions of users)

CODE BUILD TEST EXPLAIN PACKAGE

The requirements of the Platform are guided by large industrial users such as Jane Street, Citrix, Docker, Microsoft, Facebook and LexiFi, as well as accrued feedback from the opam project.

A relatively **small number of users** with **huge codebases** and **mission-critical uses**. An unusual combination!

	CODE	BUILD	TEST	EXPLAIN	PACKAGE
<2008	ocamlfind	omake		ocamldoc	godi
2009			ounit		odb
2010		oasis			
2011	mirage				
2012					

The wilderness years

Every project an island

3 hour "tutorial" at conference

	CODE	BUILD	TEST	EXPLAIN	PACKAGE
<2008	ocamlfind	omake		ocamldoc	godi
2009			ounit		
2010		sis			
2011					
2012					
2013	merlin		ocamlot	RWO codoc	opam 1.0
2014	irmin ctypes	assemblage		ocaml.org	
2015		ocamlbuild	docker	AFP course	
2016	git		datakit-ci	ocamllabs.io	topkg
2017		jbuilder	crowbar	odoc odig	opam 2.0

OCaml Labs founded

A surge of development

Shift to smaller libs

Disaggregation & coevolution

	CODE	BUILD	TEST	EXPLAIN	PACKAGE
<2008	ocamlfind	omak		amldoc	godi
2009					
2010		oasis			
2011					
2012					
2013	merlin		ocamlot	RWO codoc	opam 1.0
2014	irmin ctypes	assemblag		ocaml.org	
2015		ocamlbuild	docker	AFP course	
2016	git		datakit-ci	ocamllabs.io	topkg
2017		jbuilder	crowbar	odoc odig	opam 2.0

workflow support

break out components

distribute via opam

	CODE	BUILD	TEST	EXPLAIN	PACKAGE
<2008	ocamlfind	omake		ocamldoc	godi
2009			ounit		
2010		oasis			
2011	mirage				
2012					
2013	merlin		ocamlot	RWO codoc	opam 1.0
2014		ssemb		o	
2015		camlb		AF	
2016			akit-ci	ocamllabs.i	topkg
2017	**ppx**	**jbuilder**	**crowbar**	**odoc** **odig**	**opam 2.0**

"One in one out"

Sustainable ecosystem

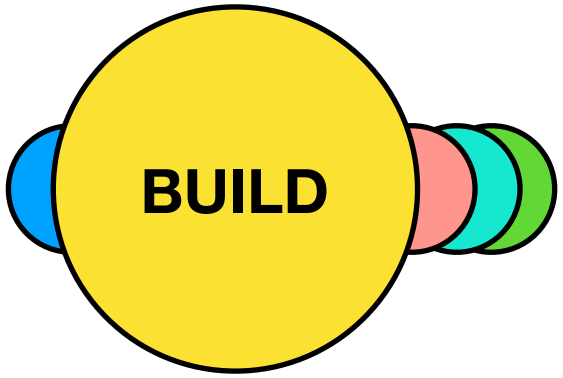
Consistent interfaces

CODE	BUILD	TEST	EXPLAIN	PACKAGE
merlin	dune	afl	odoc	opam
				a

Merlin 3.0

- **Major Merlin 3.0 developer tool release**
  - Scalable protocol to communicate with IDEs
  - Robust Windows support.
  - Now promoted to **https://github.com/ocaml/merlin**
- **Community now using it as a standard for IDEs**
  - Visual Studio Code, Atom, Sublime Text
  - Facebook Reason syntax support also.
  - More sophisticated short paths algorithm than upstream.

https://github.com/reasonml-editor/vscode-reasonml



https://github.com/ocaml/dune

Dune

A build system specialised to real world OCaml code. Provide a description of your project, and it will be built!

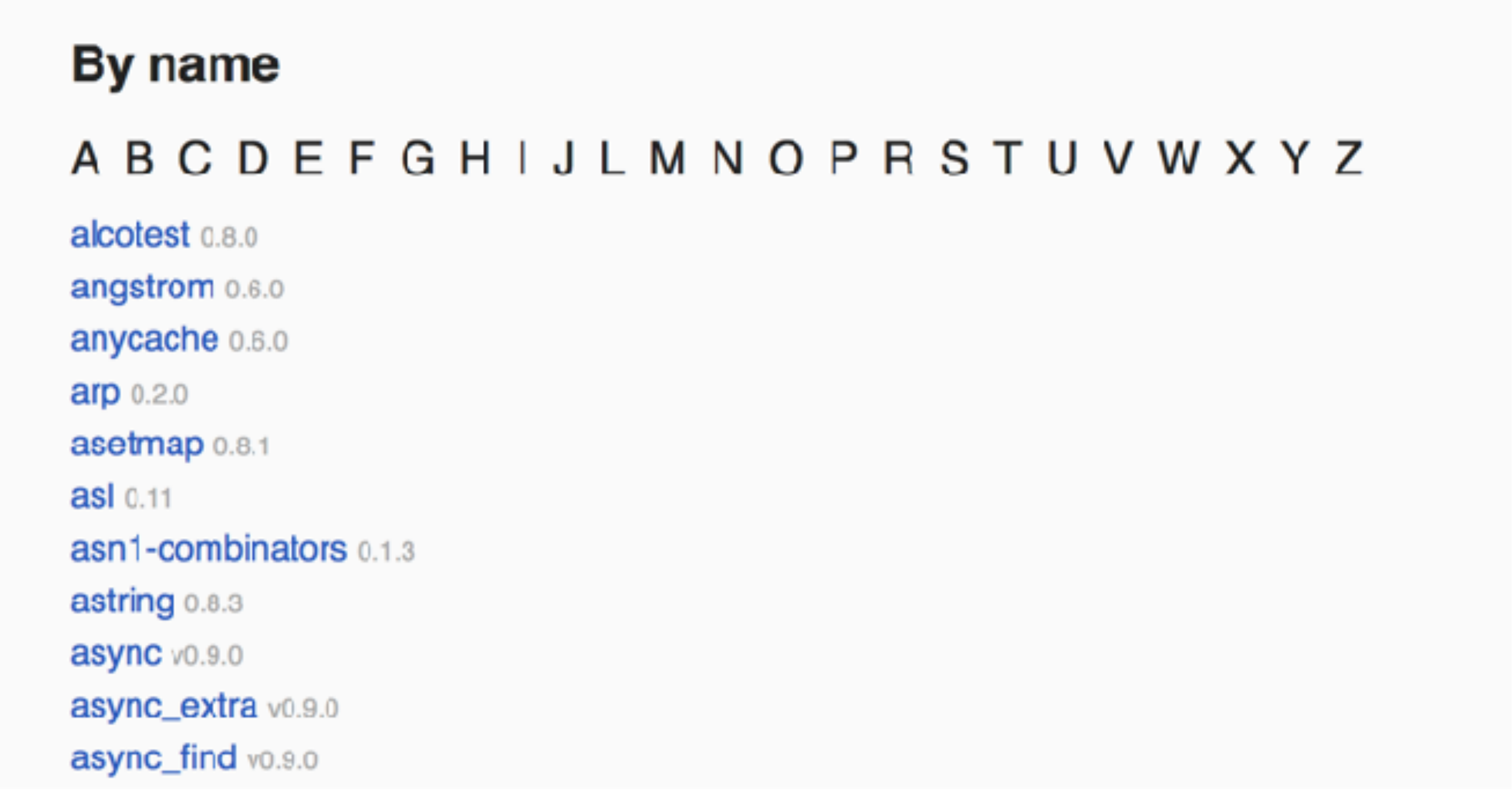
- **Compose multiple checkouts** in subdirs and it can be built in one pass
- **Multiple workspaces** to support different OCaml versions or build options (e.g. afl or flambda)
- **Declarative model** encourages portable build rules, so it "just works" on Windows.
- **Fast. Really fast.**

http://docs.mirage.io

EXPLAIN

Documentation

odoc: generate HTML for a group of libraries, with cross referencing.



By name

A B C D E F G H I J L M N O P R S T U V W X Y Z

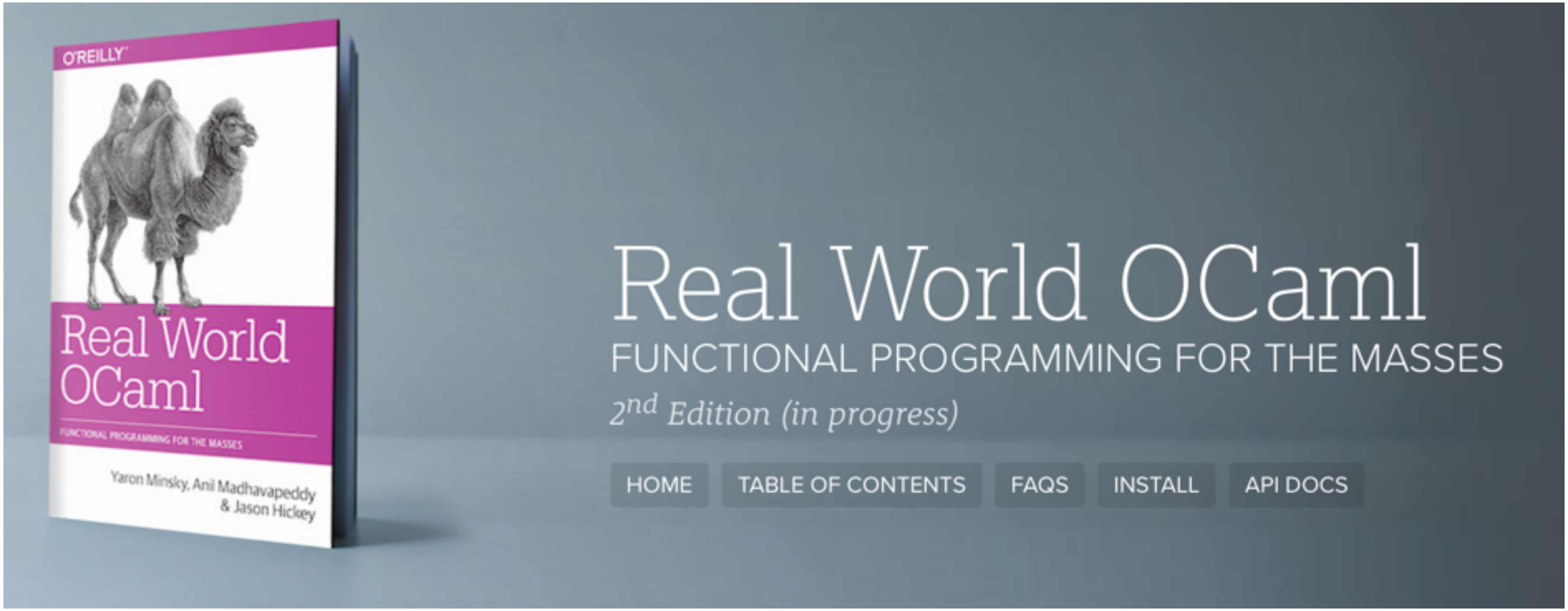
alcotest 0.8.0
angstrom 0.6.0
anycache 0.6.0
arp 0.2.0
asetmap 0.8.1
asl 0.11
asn1-combinators 0.1.3
astring 0.8.3
async v0.9.0
async_extra v0.9.0
async_find v0.9.0

```
$ opam install odoc odig
$ opam install mirage # and anything
$ odig odoc
```

http://dev.realworldocaml.org

EXPLAIN

Documentation



Real World OCaml

FUNCTIONAL PROGRAMMING FOR THE MASSES

2^{nd} *Edition (in progress)*

O'REILLY
Real World OCaml
FUNCTIONAL PROGRAMMING FOR THE MASSES
Yaron Minsky, Anil Madhavapeddy & Jason Hickey
HOME
TABLE OF CONTENTS
FAQS
INSTALL
API DOCS

Ongoing refresh at dev.realworldocaml.org

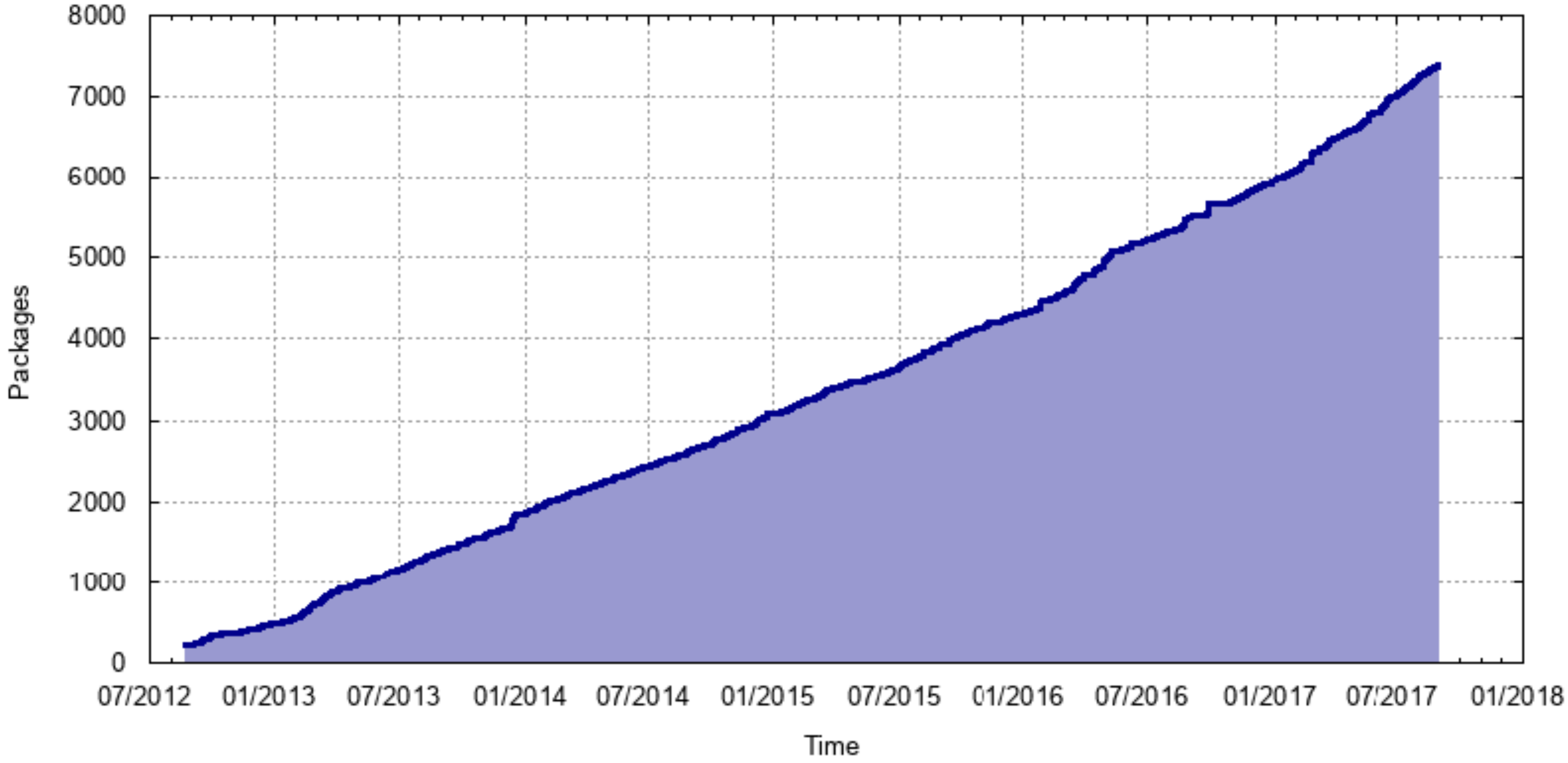
PACKAGE

github.com/ocaml/opam-repository

opam.ocaml.org

OPAM 2.0beta

- OPAM is the source-based package manager for publishing OCaml code, with package descriptions on GitHub.
- Focus this year has been on stabilising the upcoming 2.0 **Over 7000 packages now managed**

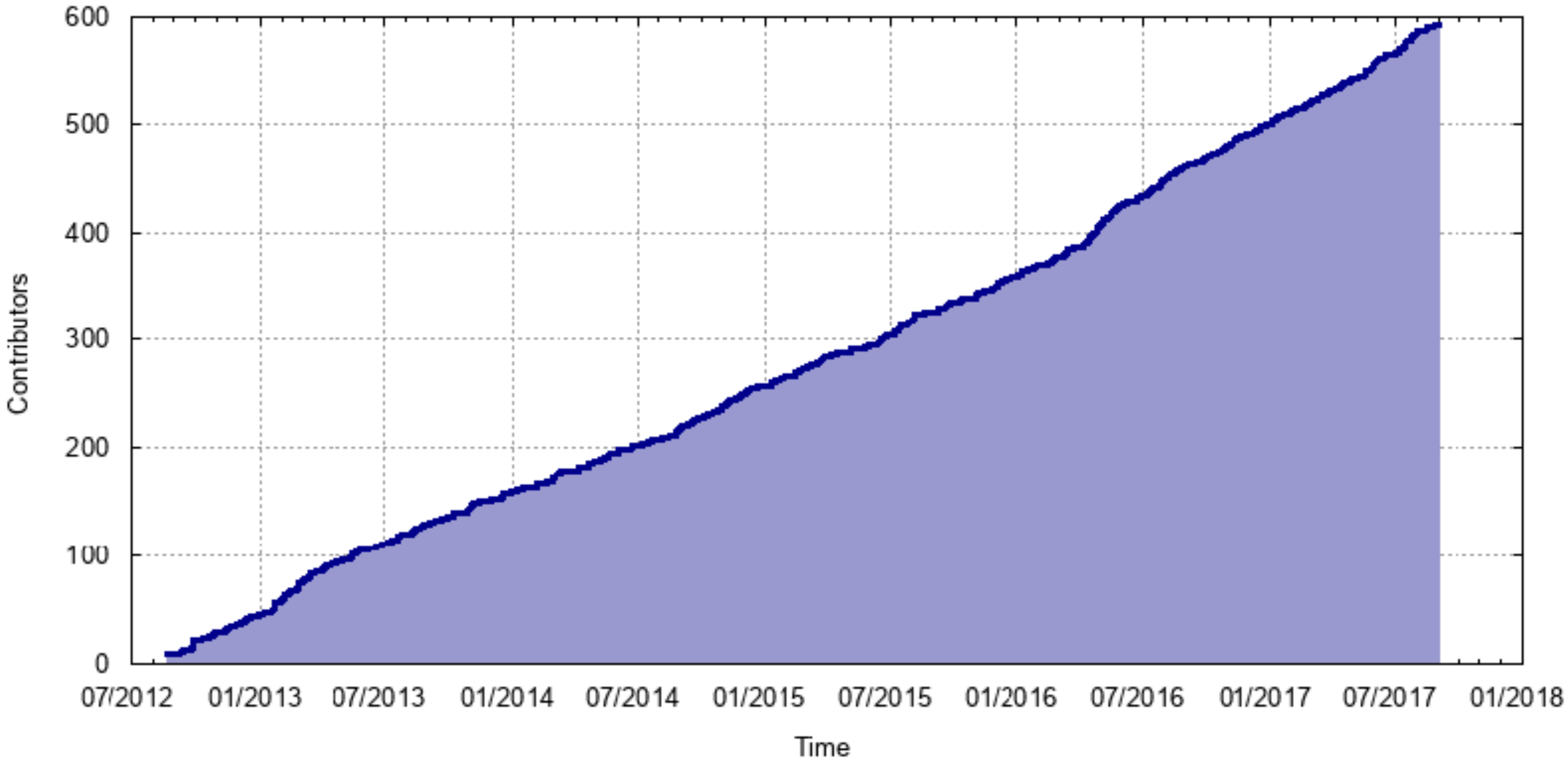


PACKAGE

github.com/ocaml/opam-repository opam.ocaml.org

OPAM 2.0beta

- OPAM is the source-based package manager for publishing OCaml code, with package descriptions on GitHub.
- Focus this year has been on stabilising the upcoming 2.0 **Over 600 individual contributors to the repository**



PACKAGE

github.com/ocaml/opam-repository

opam.ocaml.org

OPAM 2.0beta

- **New features in recent betas:**
  - more expressive package dependencies
  - computed versions to make multiple packages easier
  - local switches for use per-project
- **Solver integrated as a library, now a standalone binary!**
- **Windows support is being upstreamed!**

opam maintenance team has expanded to ~15 https://hub.docker.com/u/ocaml

TEST

Continuous Integration

- **Automated infrastructure** is very efficient *vs* engineering time, so scripting everything we can!
- Travis CI (Linux/macOS) and Appveyor (Windows) support works great with opam.
- **autoci** generates the right `.travis.yml` or `appveyor.yml` config from your project metadata
- Docker containers regularly rebuilt for many Linux distros and OCaml versions (Debian, Alpine, Ubuntu, RHEL, CentOS, Fedora, OpenSUSE, ...)

	CODE	BUILD	TEST	EXPLAIN	PACKAGE
<2008	ocamlfind	omake		ocamldoc	godi
2009			ounit		
2010		oasis			
2011	mirage				
2012					
2013	merlin		ocamlot	RWO codoc	opam 1.0
2014		ssemb		oc	
2015				AFF	
2016	g		akit-ci	ocamllabs.io	topkg
2017	**ppx**	**dune**	**autoci**	**odoc** **odig**	**opam 2.0**

"One in one out"

Sustainable ecosystem

Consistent interfaces

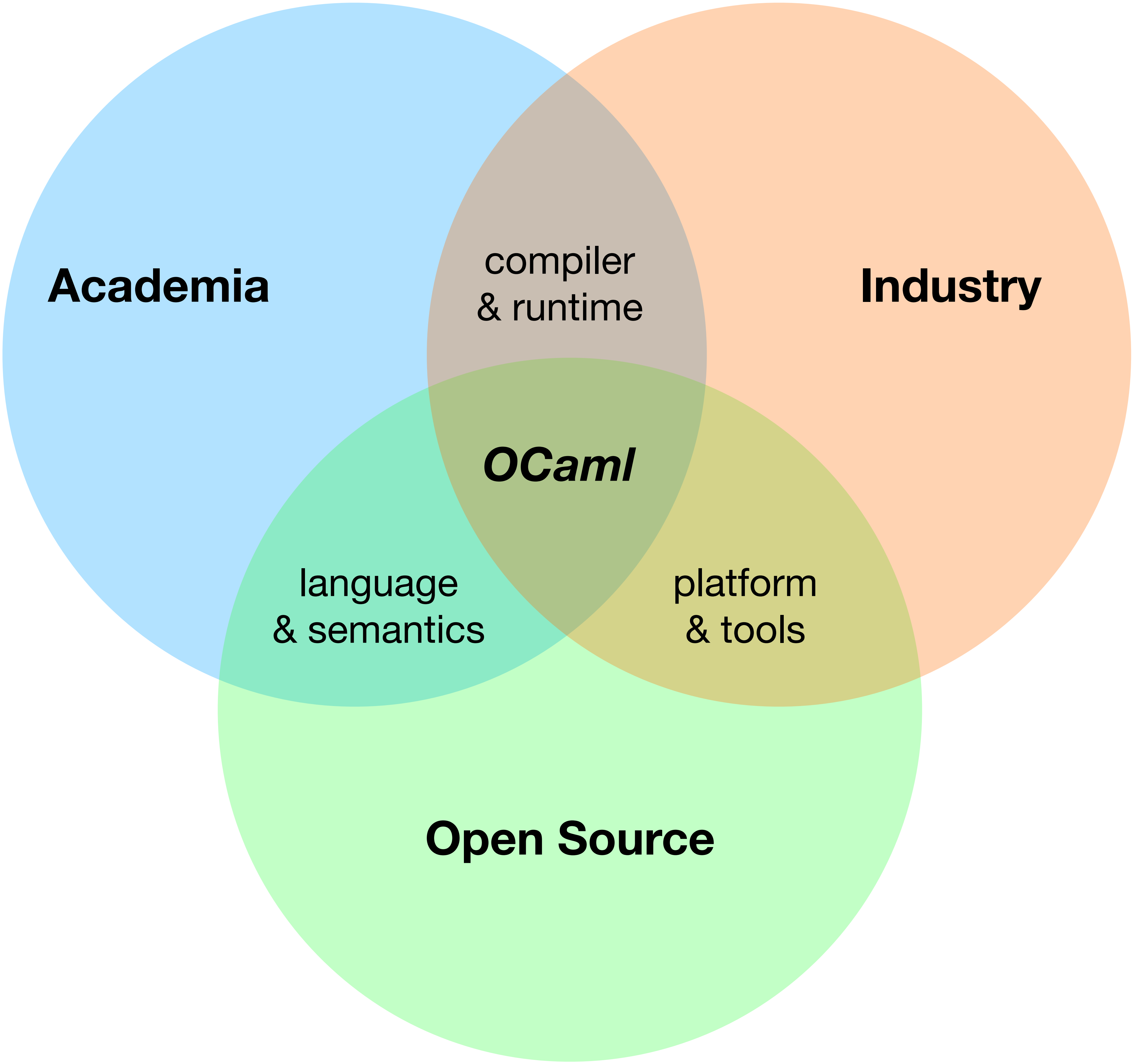
Towards a sustainable open source community

The OCaml Community

- OCaml, as an "old language" has a heady mix of academia, industry and open source enthusiasts.

- Most “new languages” have a large corporate founders (Go has Google, Rust has Mozilla, Swift has Apple, ...)

- **Problem #1:** academia and industry are both inflexible in their own ways for open-source development.



Academia

compiler
& runtime

Industry

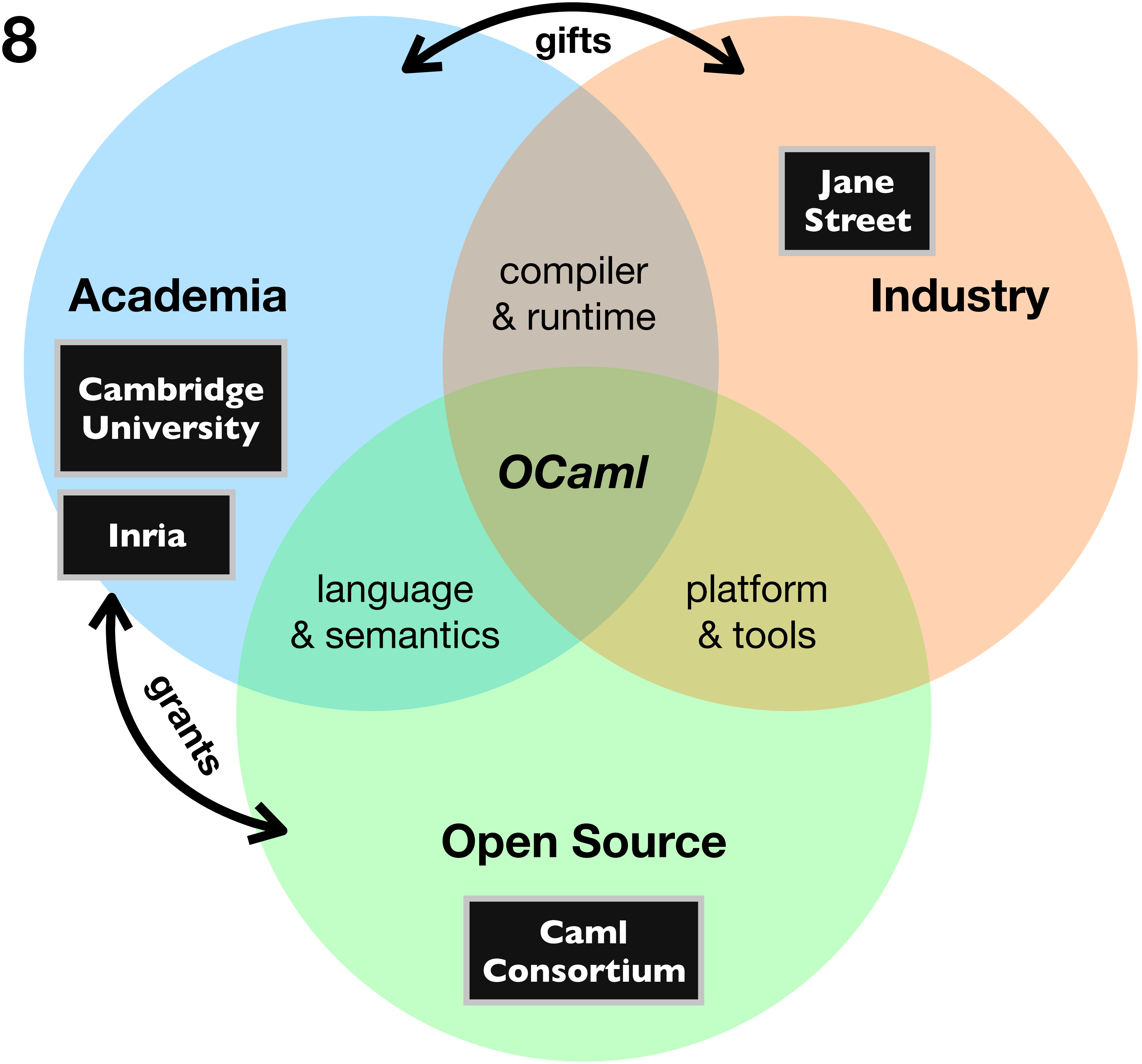
OCaml

language
& semantics

platform
& tools

Open Source

~2008



gifts

Academia

Cambridge University

Inria

Industry

Jane Street

Open Source

Caml Consortium

grants

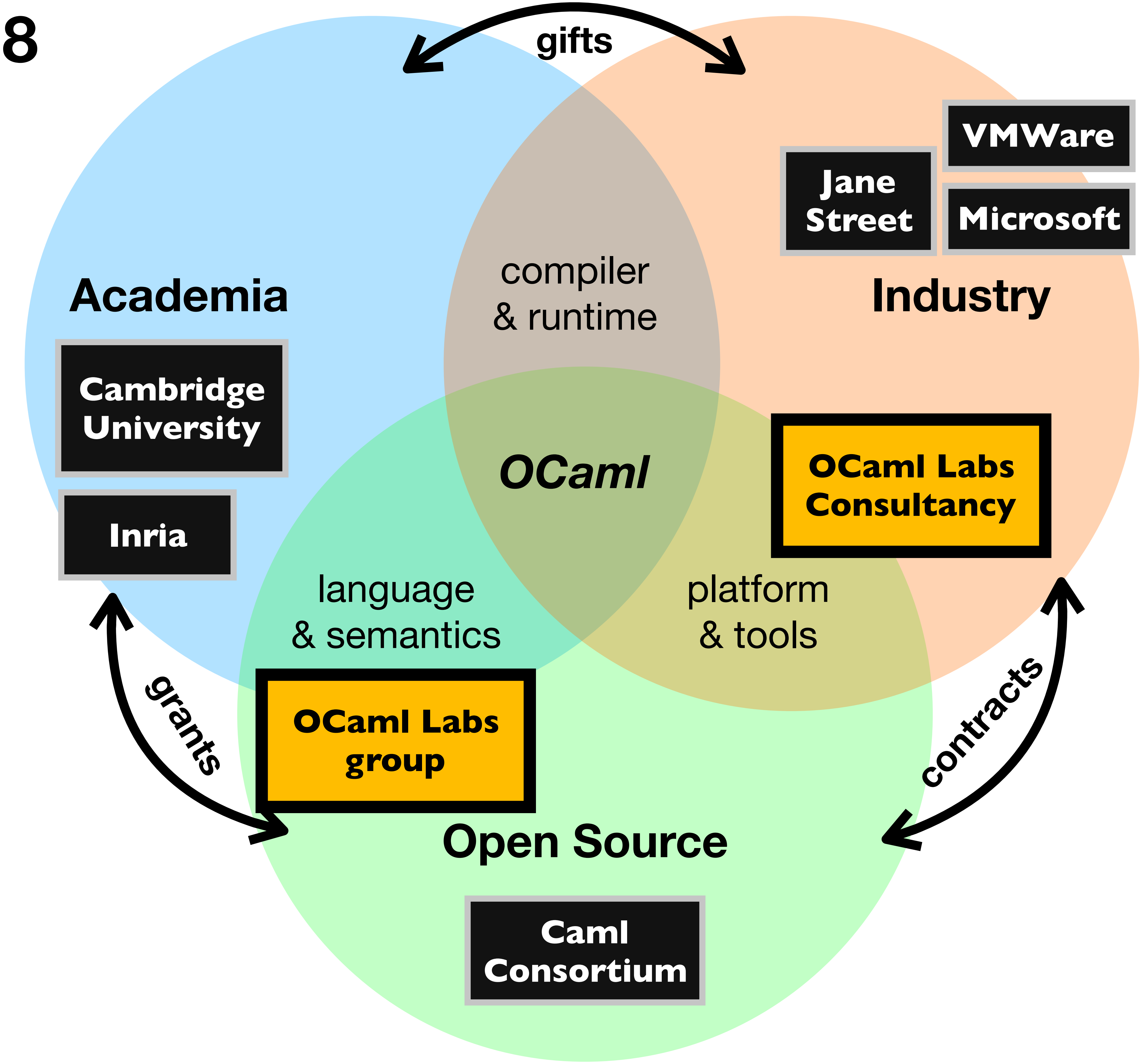
compiler & runtime

OCaml

language & semantics

platform & tools

~2018


gifts
Academia
Cambridge University
Inria
compiler & runtime
VMWare
Jane Street
Microsoft
Industry
OCaml
OCaml Labs Consultancy
language & semantics
platform & tools
grants
OCaml Labs group
contracts
Open Source
Caml Consortium

Who is OCaml Labs?

PI: Anil Madhavapeddy, anil.recoil.org

Technical Director:
KC Sivaramakrishnan, kcsrk.info

Operations Director:
Gemma Gordon, reynard.io

Postdocs:
Stephen Dolan, Daniel Buenzli, David Allsopp, Jeremy Yallop

Graduate Students:
Heidi Howard, David Kaloper-Meršinjak

Faculty:
Richard Mortier, Alan Mycroft, Ian Leslie, Jon Crowcroft

Fellows: Thomas Gazagnaire, Mark Shinwell, Leo White, Dave Scott, Hannes Mehnert

2012-2017: University of Cambridge

Part of the charity of the residential university. Difficult to operate outside of the UK.

2017-: Worldwide operations

Established "contract" division to make it easier to operate outside of Cambridge, alongside the University.

Romain Calascibetta *(git)*, Nicolas Assouad *(multicore)*, Frederic Bour *(Merlin)*, Gabriel de Perthuis *(storage)*, Mindy Preston *(fuzz)*, Anton Bachin *(odoc)*, Thomas Gazagnaire (*platform)*, Rudi Grinberg *(build)*

Co-working arrangement with University, but also remote work (Canada, USA, France, Netherlands, ...)

Participation

- **Problem #2: welcoming new users without leaving traditional users behind**
- discuss.ocaml.org has been a big success in 2017/2018!
  - Found the sweet spot between interactive chat and email
  - Has had rapid adoption in the community
  - Maintains the right atmosphere to encourage newcomers
- For the OCaml Platform:
  - Open up a semi-private area for senior maintainers to discuss the interlocking design decisions
  - Identify new maintainers from the community and empower them

Online Community

ocaml.org

opam.ocaml.org

realworldocaml.org

lists.ocaml.org

discuss.ocaml.org

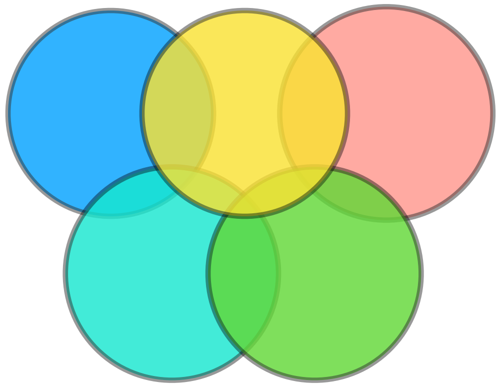
ci.ocaml.io

docs.ocaml.org

github.com/ocaml

Unifying the design of these sites, with ocaml.org as the main public site

Creating searchable API for all online OCaml resources



Community Events

Dev Meetings

ocamllabs.io

Hack Retreat

retreat.mirage.io

User Conference

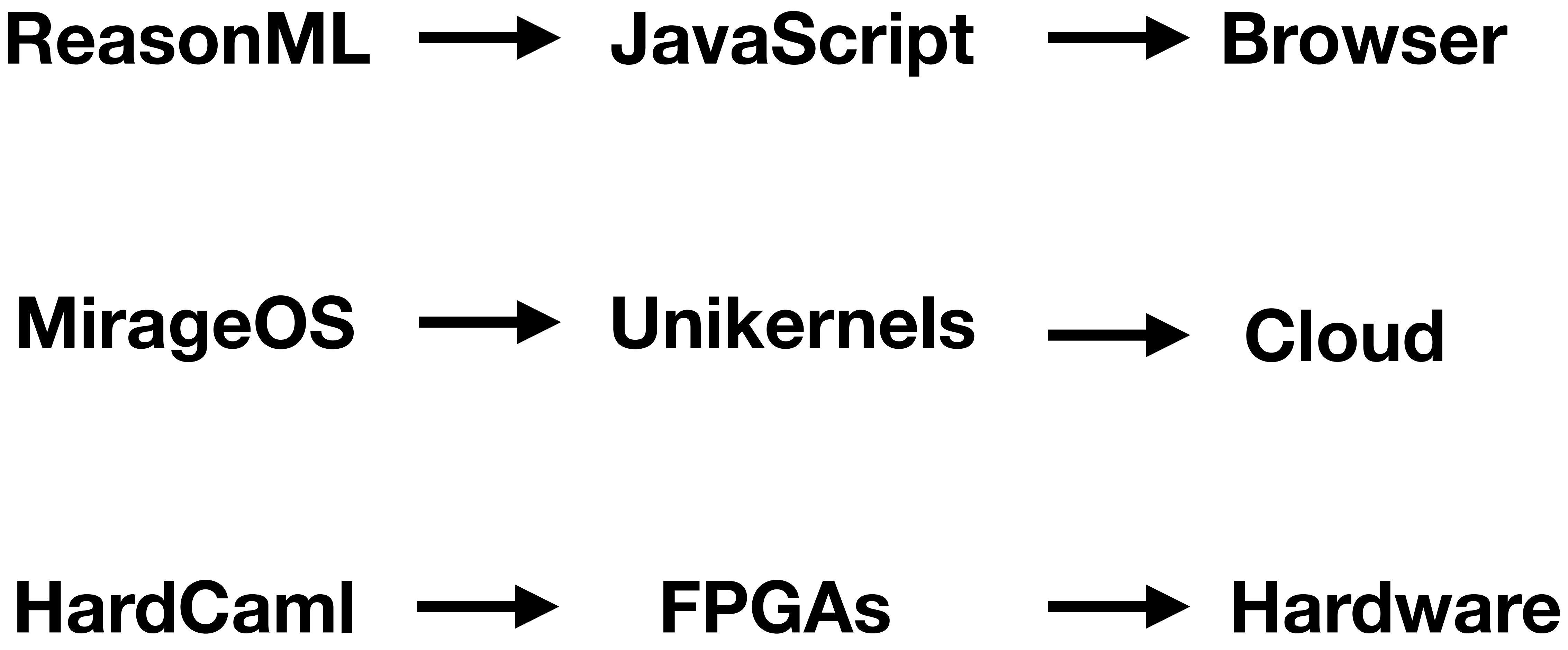
reason-conf.com

REASONCONF
World's first Reason conference for web-developers & OCaml enthusiasts

The exciting new developments

Beyond Von Neumann!

ML could be the universal language for heterogenous hardware



ReasonML → JavaScript → Browser

MirageOS → Unikernels → Cloud

HardCaml → FPGAs → Hardware

Further Resources

ocaml.org

opam.ocaml.org

discuss.ocaml.org

dev.realworldocaml.org

reasonml.github.io

ocamllabs.io

mirage.io

reynard.io

@avsm

anil.recoil.org