AFR-0

The Amlantis System Overview

Version 0.11

Markéta Lisová

June 26, 2020

Special Thanks To

Sabina Eva Lisová

Marie Strnadová

Ondřej Profant

Vojtěch Biberle

Contents

Part I

Introduction

A Brief Introduction to the Amlantis System

Amlantis System is a collection of specifications of programming and data definition languages and their related tools, runtimes and libraries.

Those specifications do not always require particular implementations, but attempt to avoid undefined behaviours as much as possible.

Amlantis also provides an open source reference implementation of those specifications, but it is indeed possible for other people to write their own implementations or forks[1] of this default implementation, probably focusing on optimising other aspects of the system, maybe exploring new options of future development to be pull-requested into the default implementation.

A Few Notes on the Name

Amlantis' name has quite some history. The project started being named *Coral*, but that collided with another language of a similar name, *CORAL 66*. Then it got renamed to *Gear*, but that again collided with another language of the same name, which seemed inactive at the time, *zippers/gear*. Than an idea was born and Aml was named *Amlantis*, which is whatever you want it to be. It could be a misspelling of *Atlantis*[2], it could be an acronym like *A ML Language*, or maybe even something like *A ML Language And Neat Technology Improvement System*, or maybe *Caml* without the *C*. For the meaning of the cryptic *ML* part, search for the *Standard ML* or *OCaml*.

[1] Forks and pull requests are preferred way of help!

[2] Intentionally – because otherwise, it would be named Atlantis, but there is already a city of that name.

The Amlantis Languages

The Amlantis System is a home to more than only one programming language – in fact, it can be home to any number of languages, ones that Amlantis users can either extend from existing languages create or even create new ones entirely. Every language build within the Amlantis System shares a common type system and value models, and thus are interoperable.

The minimal Amlantis System contains these languages:

- `Aml/Base`, a minimal Lisp-like[3] language that is easy to parse and limited in functionality.
- `Aml/Core`, an ML-like[4] language that is easier to read and harder to parse, fully featured in functionality.
- `Aml`, an ML-like[5] language that extends `Aml/Core`, is easier to read and the hardest one to parse, also fully featured in functionality, and introducing more grammar than its parent `Aml/Core`.

More languages may be added to this list over time.

[3] One might say that `Aml/Base` is a grandchild of Racket and Clojure.
[4] One might say that `Aml/Core` is a grandchild of OCaml, SML and F♯.
[5] One might say that `Aml` is a grandchild of `Aml/Core`, Haskell and Ada.