**RYBEON**

**WHAT IS RYBEON**

Before the advent of modern-day smartphones, existing phones looked perfect. Steve Jobs and his team then looked at phones from a perspective of what they should be like/from a perspective of the best way they should exit/from a perspective of perfection, and we were ushered into a new and astonishing era of smartphones. Perfection is beautiful and should be striven for.

Rybeon is a suite (of technologies) that gives software engineers **the capacity to carry out perfect software engineering**. Rybeon is an attempt to bring perfection to software engineering.

**Composition**

Rybeon is a collection of a general-purpose programming language (Rybe) and some brilliant supporting technologies like reapplicable (reusable) code infrastructure, language translation softwares, language migration technologies, etc.

**Audience**

Rybeon is meant for all types of entity that engineer softwares: governments, organizations, companies, startups, small businesses, groups, individuals, AIs, aliens, etc.

**Usage**

Rybe will see applications in system software engineering, application software engineering, cloud software engineering, game engineering, AI engineering, etc.

**GENESIS**

The creator of Rybeon (Brian Ibrahim Qamardeen) is a Software Engineer who had used a total of 12 programming languages (C, PHP, JavaScript, SQL, C++, Java, Python, Ruby, Lua, Kotlin, Golang, and Rust).

Often, when he picked up a new programming language, the language taught him new programming philosophies, some of which he sometimes found phenomenal; we will call this type of thing an epiphany.

After picking up Golang and Rust and experiencing epiphanies again, he started to wonder if there were more beautiful programming philosophies.

He felt there should be more of them out there.

As a perfectionist, Brian started to wonder how he could acquire all the beautiful programming philosophies that possibly exist.

He came to the realization that, if he wants to acquire them all, he has to start approaching programming according to how it should be done, and not how existing languages encourage it to be done.

When Brian started to approach programming according to how it should be done, he started to see many ways programming can be significantly better, compared to mainstream approaches.

Brian started trying to adopt his newly discovered programming approach to his day-to-day programming tasks. Because existing languages were not designed based on his newly discovered approach, it was very difficult to adopt the approach, using any of the existing languages.

Because Brian could not adopt his approach using any of the existing languages, he decided to create a language that enables him to do so. This decision gave birth to Rybeon in October 2019.

**However, The Initial Idea Has Grown**

Although Brian’s initial intention was just to create a programming language, he saw that it would be wise to morph the project into a bigger idea: a suite of programming technologies (a language and supporting technologies like infrastructure that facilitate code reapplication, technologies that can help interested entities convert their codebase from other languages to Rybe, etc).