Contents

Preface

The way we design and write software to do computation and AI sucks. It's a vat of boiling poop, mixed with pee, slowly swirling and bubbling toward that dehydrated semi-solid state of goo that serves to repel and repulse most normal people only attracting the few unfortunate-fortunate folks that happen to be obsessed with scat.

Hrm, too much? Probably. I guess you'd expect me to use concrete examples and cite evidence to make my points, me being a professor and all. I mean, I could write something like *"The fundamental imperative programming model utilized in near all of the production software produced in the last four decades has not changed since blah blah blah..."* to meet expections. I'd certainly sound more credible and perhaps super smart. Well, I'm not going to do that here. Let's have fun. Afterall, Jaseci has never been work for me, its play. Very ambitious play granted, but play at it's core.

Everything here is based on my opinion and intution. That suffices for me, and I hope it does for you. I have spent many decades coding and leading teams who code and my gut that tells me that we can do better. This is my attempt at better. I hope you find value in it. If you do, awesome! If you don't, also awesome.

Chapter 1

Introduction

Chapter 2

What and Why is Jaseci

2.1 Viewing the Problem Landscape Spacially

2.2 Compute via The Collective, The Worker Bee Model

Chapter 3

Abstrations of Jaseci

3.1 Graph

3.2 Walkers

3.3 Abilities

3.4 Other Abstractions Not Yet Actualized

Chapter 4

Architecture of Jaseci and Jac

4.1 Anatomy of a Jaseci Application

4.2 The Jaseci Machine

4.2.1 Machine Core

4.2.2 Jaseci Cloud Server

Chapter 5

Interfacing a Jaseci Machine

5.1 JSCTL: The Jaseci Command Line Interface

5.2 Jaseci Rest API

Chapter 6

The Jac Programming Language

Chapter 7

Architecting Jaseci Core

Chapter 8

Architecting Jaseci Cloud Serving

Epilogue