More Fine-grained and Distributed Separation of Responsibilities in Microservice Architecture: The Arrival of Femtoservices

Matthew Safar, Ryan Hornby

March 29, 2025

Abstract

1 Introduction

Microservices have been the gold standard in modularity, scalability, and distributed development for the past two decades. In recent years there have been incremental improvements in microservice architectures, but we propose a monumental leap in microservice architecture: femtoservices. These femtoservices will greatly further the goal of microservices, by simply shrinking the size of the services themselves.

2 Technical Details

The noble intent of microservices is to create separate, independent, and maintainable systems that work together in order to provide the full functionality of a much larger more complicated system. In principle, microservices serve the same purpose as functions in a program by encapsulating logic, and dividing responsibility so that the program as a whole is more understandable and easier to modify and maintain. Following this analogy leads us to a problem, however, a lot of microservices, unlike well designed functions, aren't that small.

To fix this issue we created a tool for the Go programming language, that can take a Go program and output a collection of new and improved femtoservices. This tool will read through existing Go code and automatically divides it into femtoservices along function boundaries, thus allowing for unprecedented modularity and scalability.

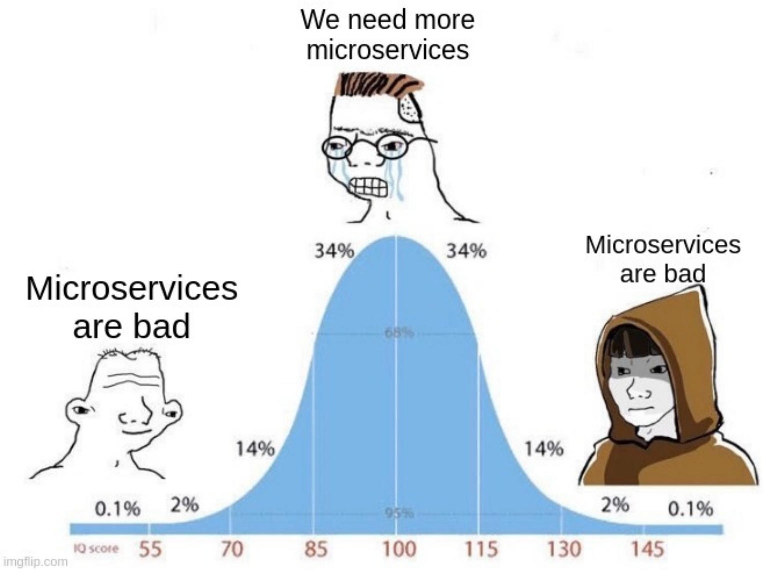
2.1 Launching and Running the Program

Using this tool is easy, simply run it to compile your code and then run the output script to initialize all the femtoservices.

2.2 Limitations

Despite our extensive testing, we have yet to find a single flaw with this new paradigm. This will truly revolutionize the software ecosystem, so don't get left behind!

Abstract



3 Actual Introduction

Okay are the managers gone?

Alright saddle up bucko, now that your organization has bought the enterprise version of this software, you gotta either figure out to use it or find a way to fake it when they ask.

3.1 Actual Technical Details

https://backendathome.github.io/404.html

3.2 Actually Launching and Running the Program

Currently, the options for running your program using femtoservices are pretty limited. The tool currently allows you to output a BASH script that will run your program by spinning up each of the microservices individually, or if you prefer containerization, we provide a basic Docker Compose setup.

3.3 Actual Limitations

As all good programmers know global variables are an absolute no no, and that is the case with femtoservices as well. They *will* break the compiler. Goto statements on the other hand are compiled into femtoservices just fine, so they must not be as bad as big programmer wants us to believe.

This project is a work in whenever we feel like it, so if you need the following features please wait or add them yourself:

- A program with multiple Go files
- Functions with multiple arguments
- Functions with multiple returns
- Updating APIs without re-running the whole thing
- Kubernetes integration
- More intelligent port assignments
- Calling functions from shared objects
- Copying relevant configuration files across microservices when containerizing

And many more!

4 Why?

I ask myself the same thing every day.

5 Source?

If for some reason you're committed to getting this working, you can find the source code here: https://github.com/RyanHornby/femtoservices.