CS340 Proposal

Jackson Mowry

August 22, 2024

1 FARPCs

(*F**ast *A synchronous **R** emote **P** rocedure **C** all **s** erver)

1.1 Summary

FARPCs is a proposed system where users can define both datatypes and procedures in one file, which can then be used to generate both client and server code to enable communication between the two. Remote procedure calls are an alternative to REST apis, SOAP services, or more modern GraphQL apis, which allow the programmer to simply call a procedure without having worry about the underlying network request taking place.

1.2 Problem Being Solved

Modern web services often communicate by encoding data in JSON, which not only encures a cost on the producer, but again on the consumer as the message is decoded back into types the host language can understand. Companies may choose to move some of their more critical APIs to a custom binary encoding, but doing this on a case by case basis leads to unmaintainable code, with multiple solutions being implemented simultanously across a single codebase.

farpss would allow for a single source of truth to be defined in a schema file, which is then used across all server/client languages to generate the appropriate code. if more lanugages need to be supported in the future they can easily be added though the client generation template files used to write language specifications.

On the server side FARPCs will generate a template server in C, declaring the appropriate prototypes for each procedure defined in the schema file. The server will be entirely asynchronous, implemented using an event look on top of `io_uring`, a new system in Linux for making system calls asynchronously. Server choice should also be able to expand, taking advantage of the same template system for client code.

1.3 Major Features

- Single source of truth for both type and procedure definitions
- Modular code generation system for clients, allowing clients to easily be added for any new languages
- An out of the box blazingly fast asynchronous web server written in C to serve remote procedure calls
  - With the ability to add more servers through code generation template files
- Type safe function calls across the network boundry
- Versioned schemas, allowing for iterative development, while maintaining backwards compatability

1.4 Languages

I am not set in stone on any particular language for developing this tool. I'd prefer to work in `v` or `go` as I am most familiar with those tools, but I am also not opposed to `python`, `typescript`, `java`, or `scala`.

We can start by designing a Javascript client generator, along with a C server generator, working knowledge of these two languages is preferable.

Schema files will look something like XML, so a knowledge of XML and XML parsing will be helpful.

```
<schema version = "1.0.0">
  <messages>
    <message name="Reaction">
      <field type="string" required="true">Emoji</field>
      <field type="integer" default="0">Count</field>
    </message>
    <message type="chat_message">
      <field type="string" required="true">Author</field>
      <field type="timestamp" required="true">Message Sent</field>
      <field type="integer">Likes</field>
      <field type="string" required="true">Body</field>
      <field type="reaction[]">Reactions</field>
    </message>
  </messages>
  <procedures>
    <procedure name = "send_message">
      <description>
        Submits a new chat message along with the users authentication token.
      </description>
      <parameters>
        <parameter type="chat_message" required="true">Message</parameter>
        <parameter type="string" required="true">Auth Token</parameter>
      </parameters>
      <returns>
        <field type="boolean">Success</field>
      </returns>
    </procedure>
    <procedure name = "get_messages">
      <description>
        Get the latest ‘Count‘ messages from the server, requires a users authenticatio
        ‘Count‘ defaults to 10 messages.
      </description>
      <parameters>
        <parameter type="integer">Count</parameter>
        <parameter type="string" required="true">Auth Token</parameter>
      </parameters>
      <returns>
        <field type="chat_message[]">Messages</field>
      </returns>
    </procedure>
  </procedures>
</schema>
```

1.5 Who Would Use This?

Any company/project that is currently dealing with server/client performance issues due to JSON encoding/decoding. This tool would also help a new project get off the ground faster, by facilitating code generation across many different languages from one central schema file.