Protobufs

for Beginners

by a beginner

What?

a language-neutral, platform-neutral, extensible way of serializing structured data.
— Google

What? v2.0.0

A standardized way of structuring information across different languages.

— Mat

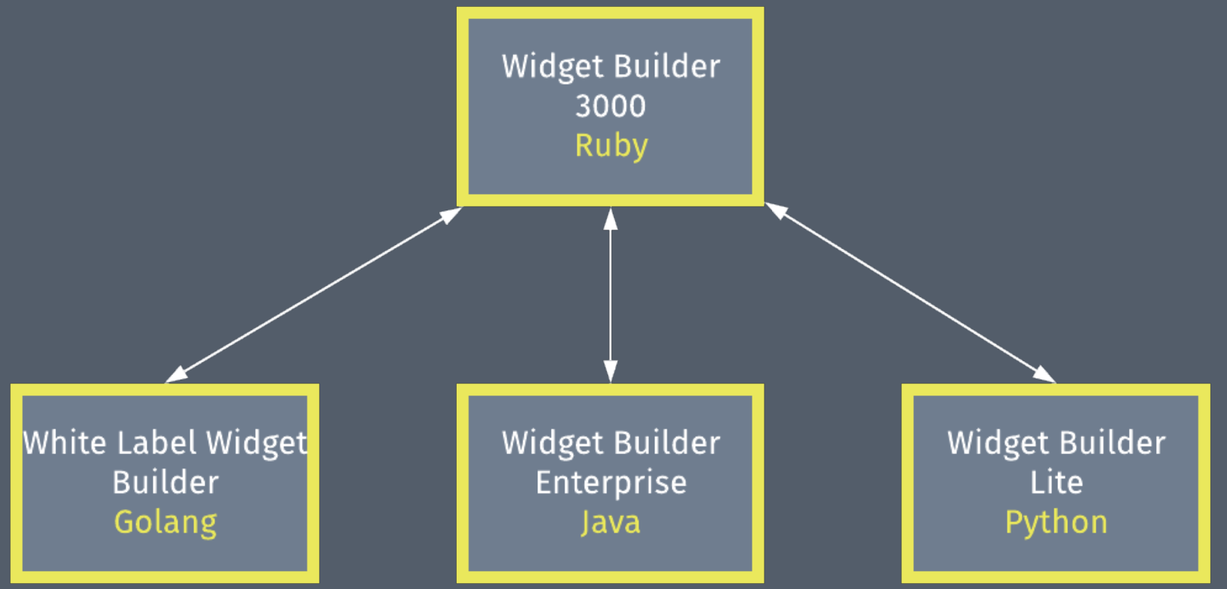
Setting the stage

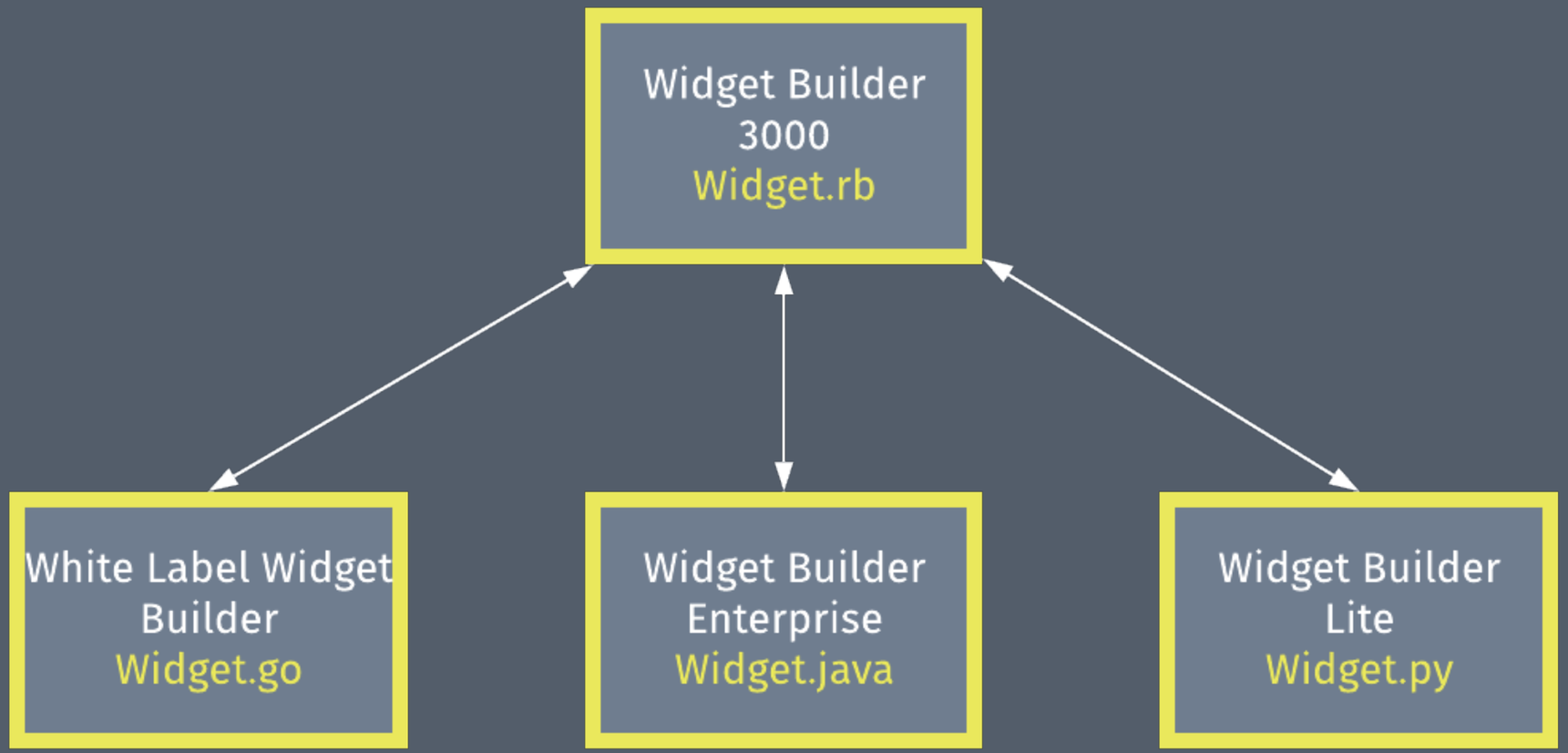
Company: Global Widgets Incorporated

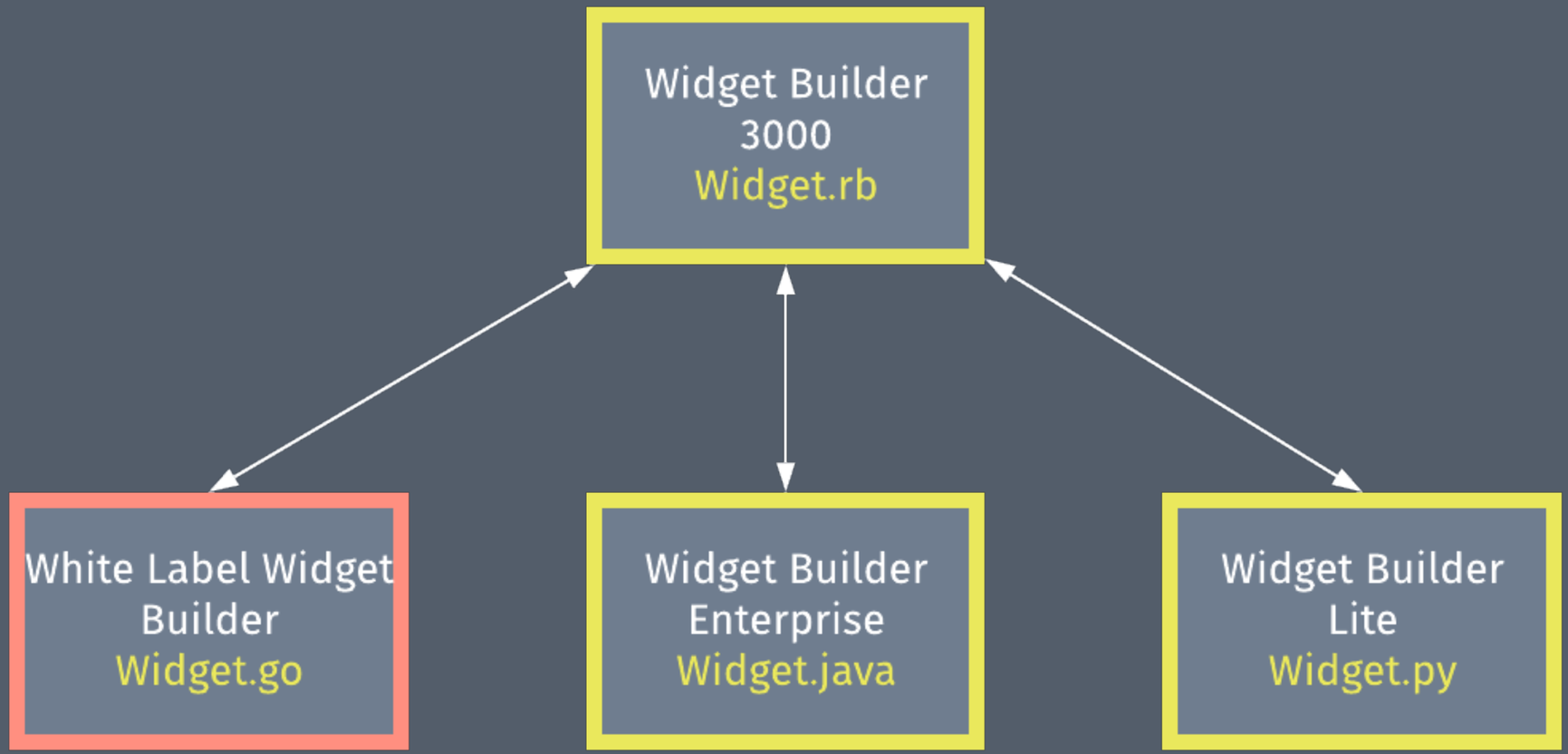
Developer: Wally

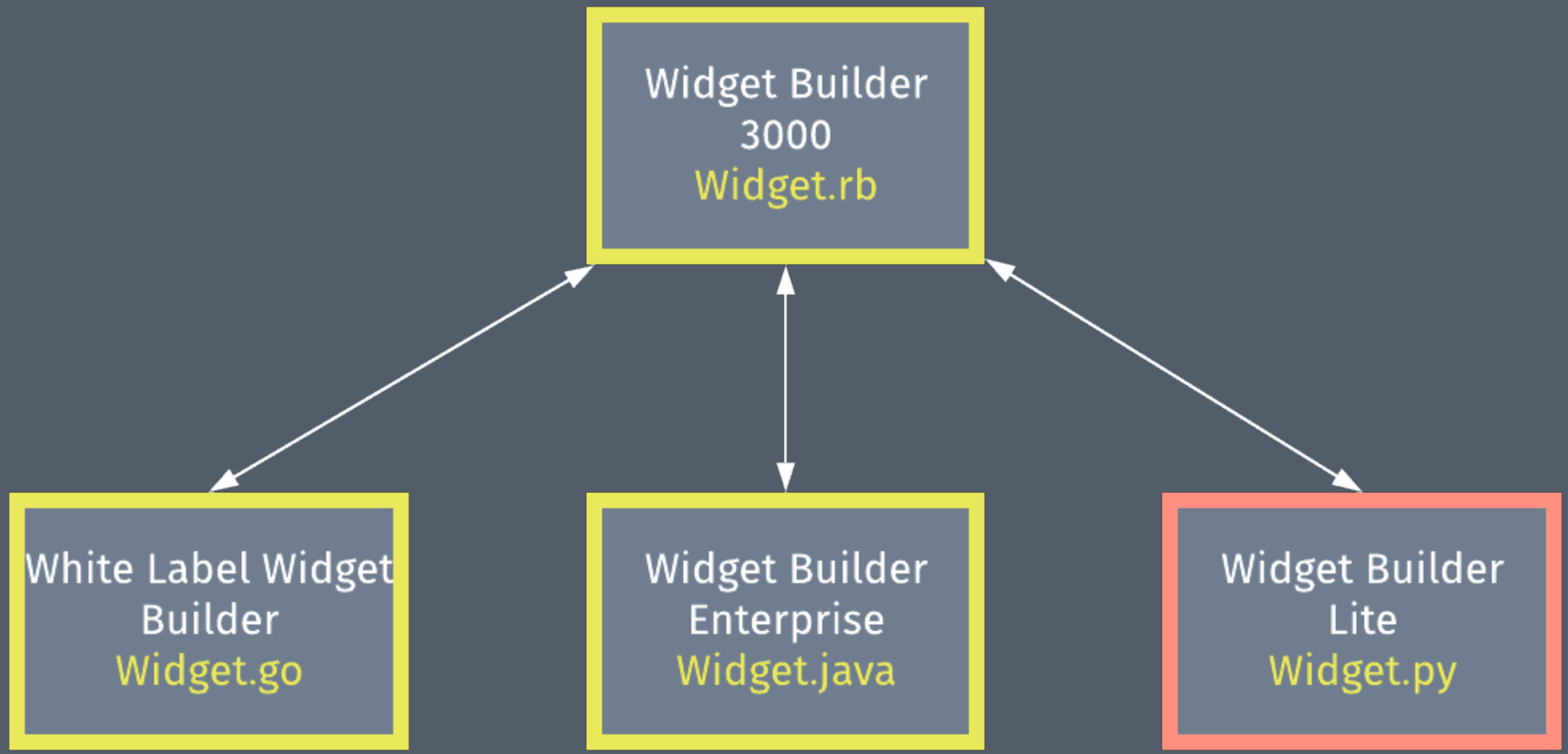
Problem(s)*: Several languages across several services

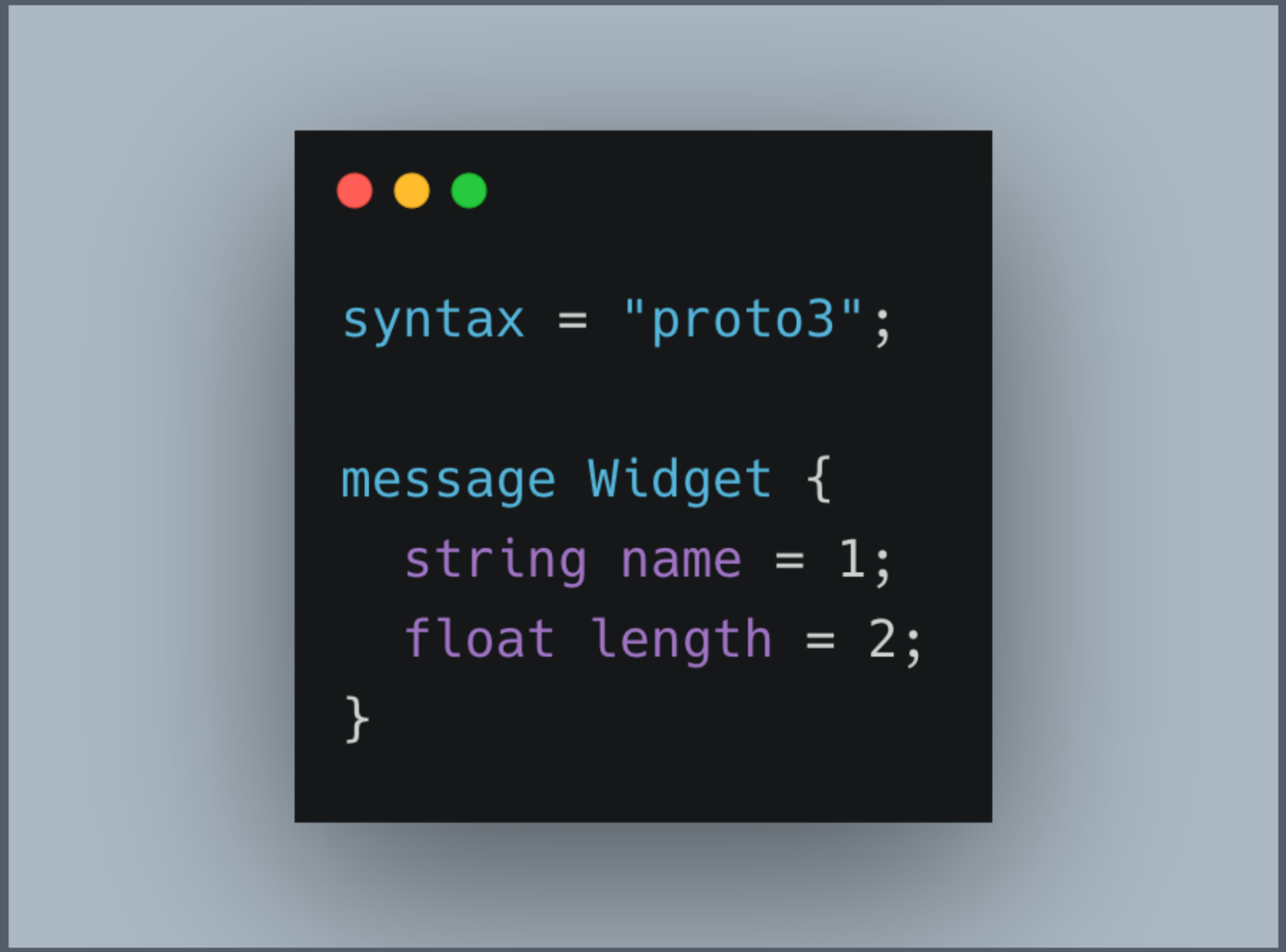
*Of which there are a few




Widget Builder
3000
Widget.rb
White Label Widget
Builder
Widget.go
Widget Builder
Enterprise
Widget.java
Widget Builder
Lite
Widget.py


Widget Builder
3000
Widget.rb
White Label Widget
Builder
Widget.go
Widget Builder
Enterprise
Widget.java
Widget Builder
Lite
Widget.py


Widget Builder
3000
Widget.rb
White Label Widget
Builder
Widget.go
Widget Builder
Enterprise
Widget.java
Widget Builder
Lite
Widget.py



widget.proto

```ruby
# Generated by the protocol buffer compiler.  DO NOT EDIT!
# source: widget.proto

require 'google/protobuf'

Google::Protobuf::DescriptorPool.generated_pool.build do
  add_file("widget.proto", :syntax => :proto3) do
    add_message "Widget" do
      optional :name, :string, 1
      optional :length, :float, 2
    end
  end
end

Widget = ::Google::Protobuf::DescriptorPool.generated_pool.lookup("Widget").msgclass
```

```
protoc --ruby_out=./ widget.proto
CreatePresentationRequest.new(title: "Protobufs For Experts", speaker_name: ["Mat"], length_of_talk: 60)
```

But why, you ask?

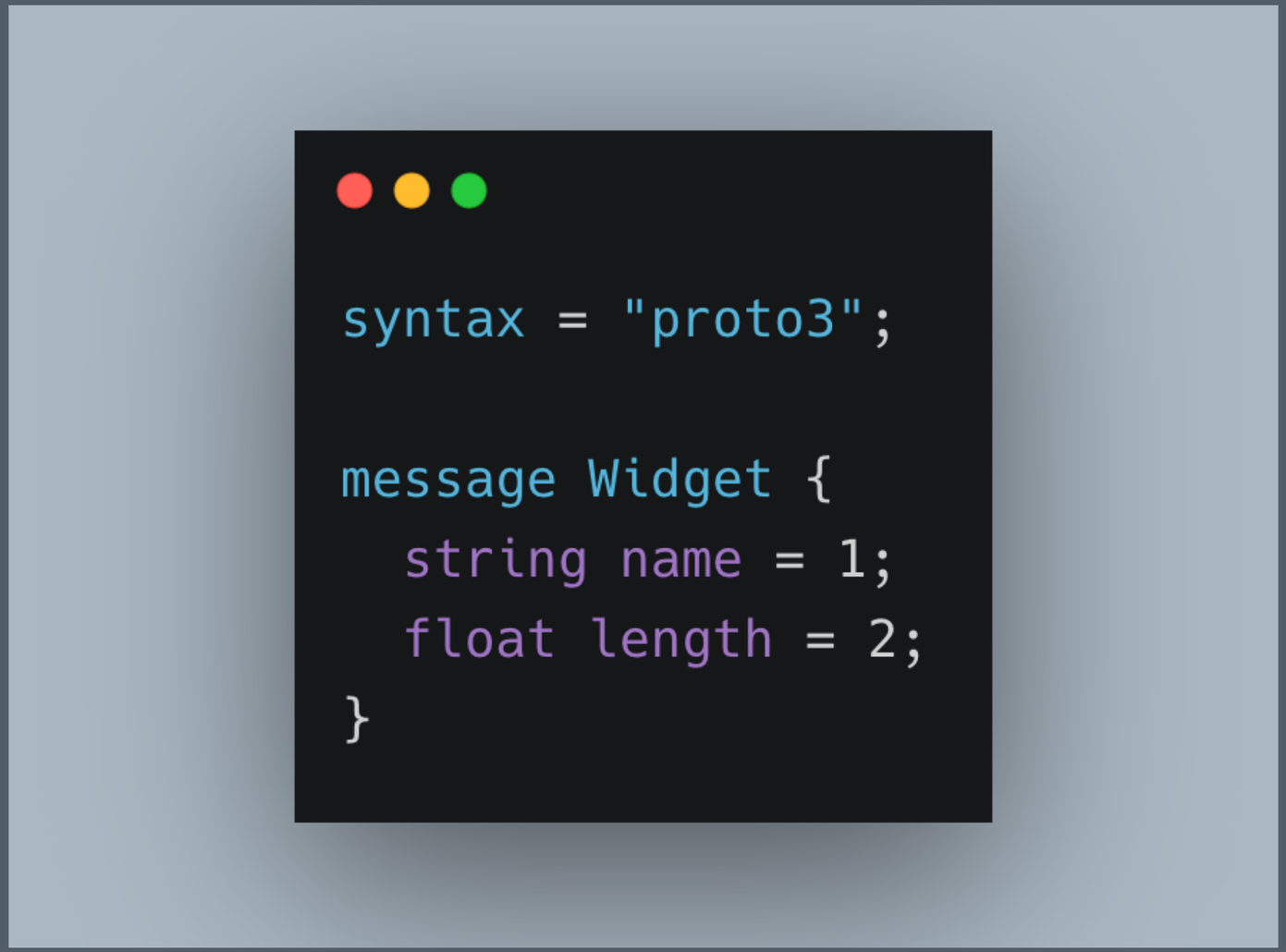
```
protoc --ruby_out=./ widget.proto

protoc --java_out=./ widget.proto

protoc --csharp_out=./ widget.proto

protoc --python_out=./ widget.proto

protoc --ruby_out=./ widget.proto --java_out=./ widget.proto --csharp_out=./ widget.proto --python_out=./ widget.proto
```



widget.proto

We got types!

.proto Type	Notes	C++ Type	Java Type	Python Type[2]	Go Type	Ruby Type	C# Type	PHP Type
double		double	double	float	float64	Float	double	float
float		float	float	float	float32	Float	float	float
int32	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint32 instead.	int32	int	int	int32	Fixnum or Bignum (as required)	int	integer

Bits and Pieces

More info for protocol buffers
https://developers.google.com/protocol-buffers/

Languages supported by Google
C++, C#, Dart, Go, Java, Python

Plus some community plugins to support
JS, TS and I imagine a bunch more.

Thanks all!