WIZELINE ACADEMY

UPCOMING COURSES

NODE.JS WORKSHOP
September 21 @CDMX, Livestreamed

TECH TALK SERIES: AN INTRO TO VAULT
September 27 @CDMX, Livestreamed

MICROSERVICES WITH GOLANG WORKSHOP
September 28 @GDL, Livestreamed

PROBLEM SOLVING & ALGORITHMS CRASH COURSE
Sept. 30-Oct. 1 @GDL

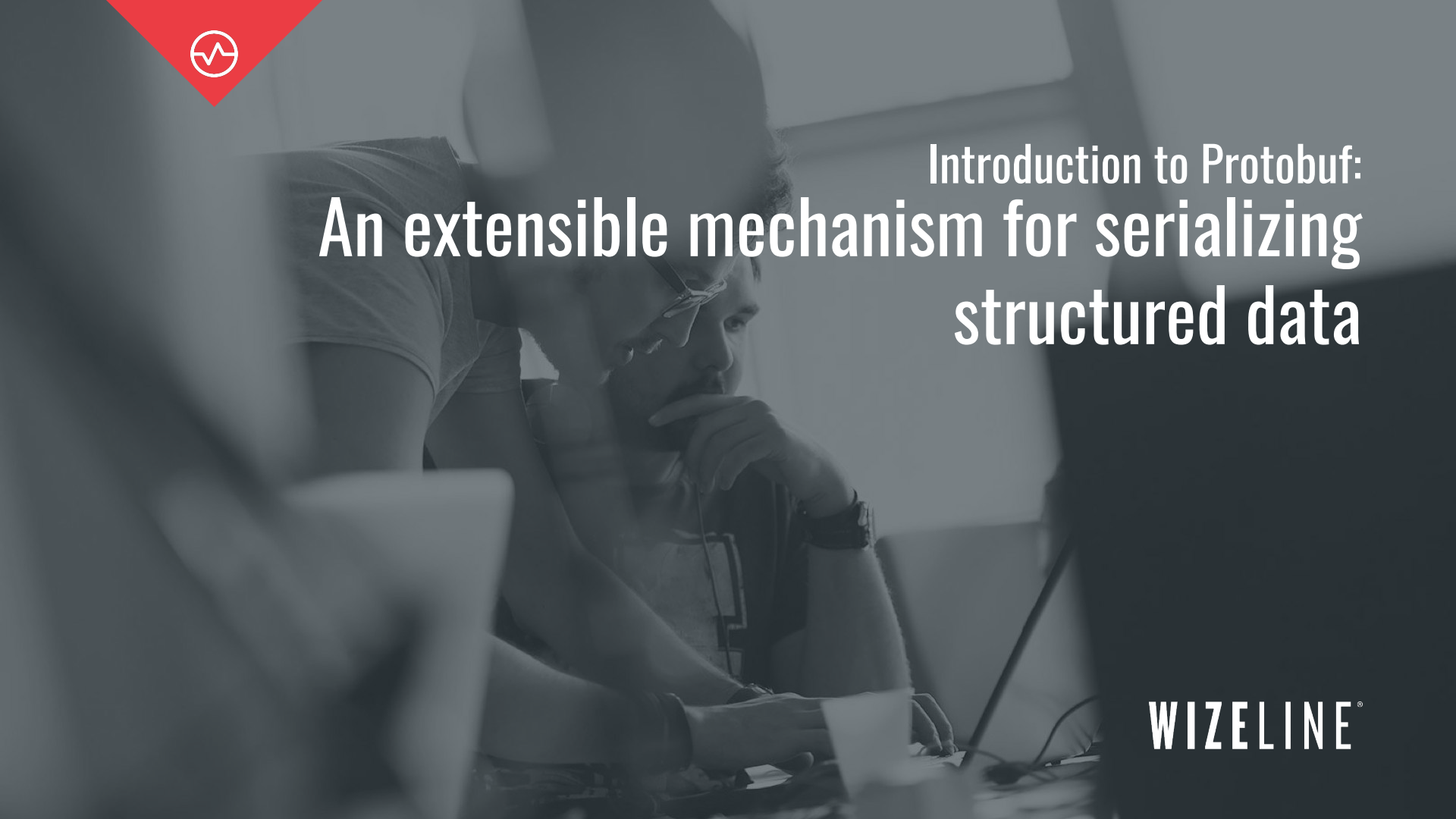
VUE.JS WORKSHOP
October 7 @GDL, Livestreamed

@WizelineAcademy

/WizelineAcademy

academy.wizeline.com

Get notified about courses:
tinyurl.com/WL-academy



Introduction to Protobuf:
An extensible mechanism for serializing structured data

WIZELINE®

Agenda

- Context
- What is Protobuf
- Companies using it
- How it works
- Installation
- Swift Protobuf
- Demo

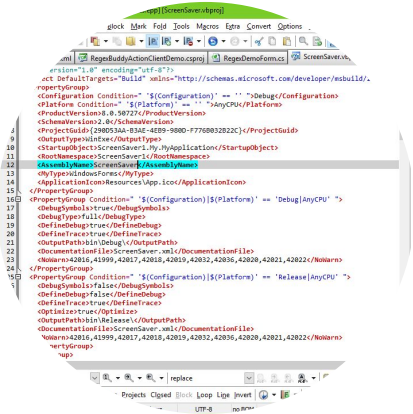

Context

Web Service

- Service offered by an electronic device to another electronic device
- Communicate via World Wide Web
- Originally designed for human-to-machine communication
- Transfer machine readable file formats such as XML and JSON

XML

- XML stands for eXtensible Markup Language.
- Defines a set of rules for encoding documents
- <key tags>
- 1996



JSON

- JavaScript Object Notation
- Lightweight data-interchange format
- Easy for humans to read and write
- ECMA-262 3rd Edition - December 1999
- Language independent

JSON

Pro:

- Simple syntax, which results in less "markup" overhead compared to XML.
- Easy to use with JavaScript as the markup is a subset of JS object literal notation and has the same basic data types as JavaScript.
- JSON Schema for description and datatype and structure validation
- JsonPath for extracting information in deeply nested structures

Con:

- Simple syntax, only a handful of different data types are supported.

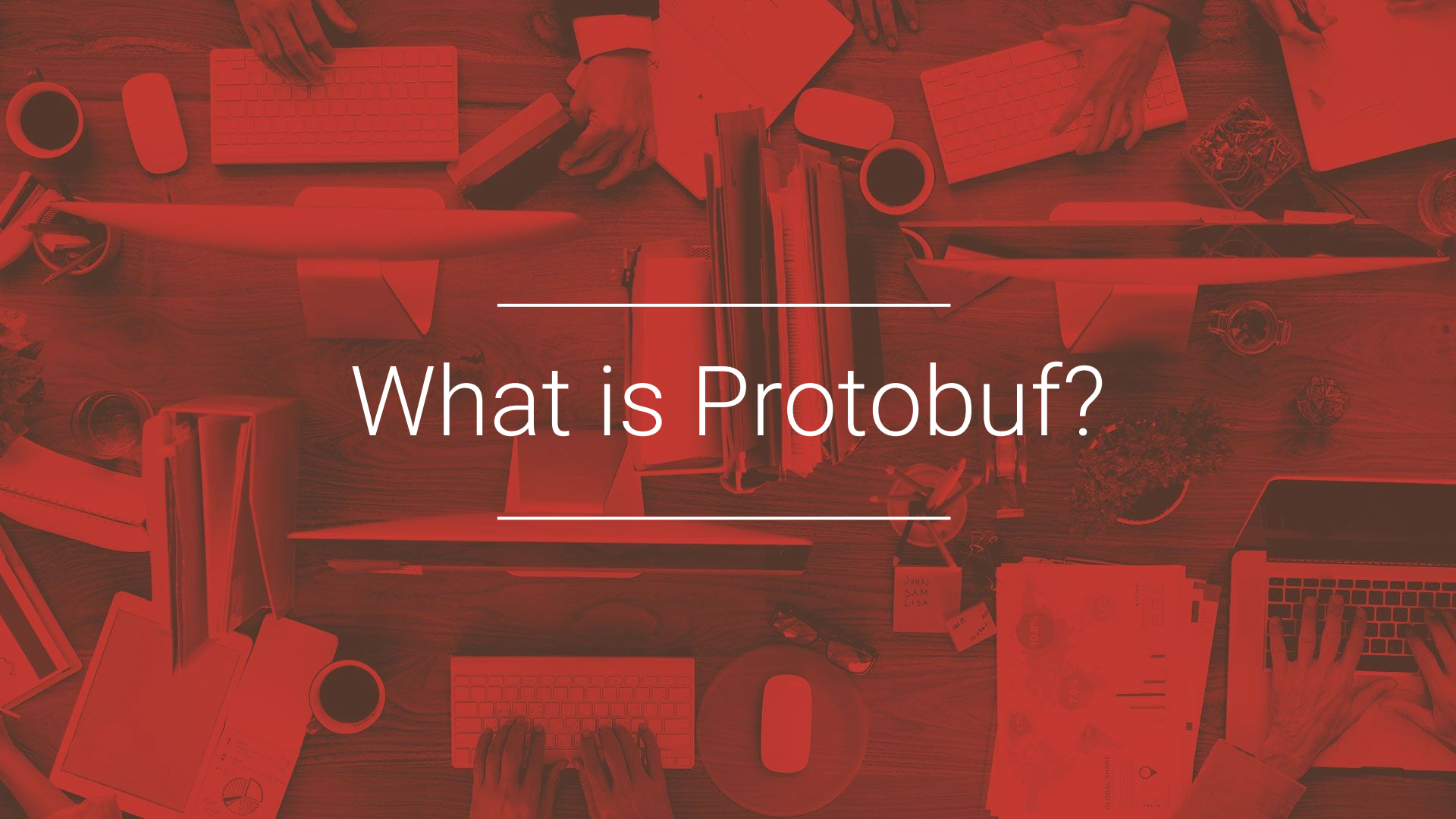
XML

Pro:

- Generalized markup; it is possible to create "dialects" for any kind of purpose
- XML Schema for datatype, structure validation. Makes it also possible to create new datatypes
- XSLT for transformation into different output formats
- XPath/XQuery for extracting information in deeply nested structures
- built in support for namespaces

Con:

- Relatively wordy compared to JSON (results in more data for the same amount of information).


What is Protobuf?

Protocol Buffer

- Method of serializing structured data
- Stream of bytes that represents the structured data
- 2001 Google internal - 2008 Public version
- Protocol buffers are a flexible, efficient, automated mechanism for serializing structured data
- You define how you want your data to be structured once (Proto)

0a 0d 08 f9 27 12 02 4f 4b 18 8a 8c 06 20 4e

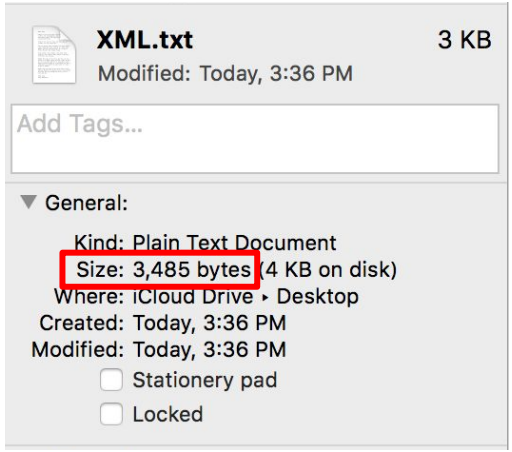
https://developers.google.com/protocol-buffers/docs/encoding

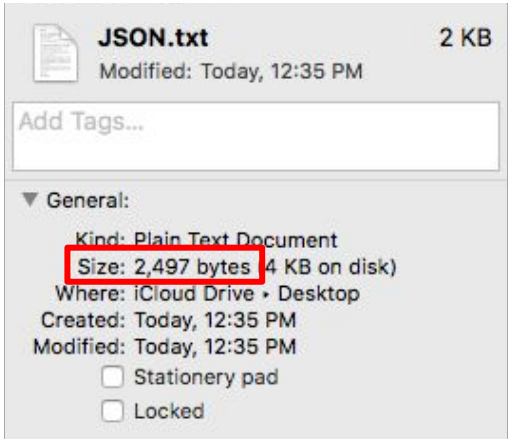
Companies Using it

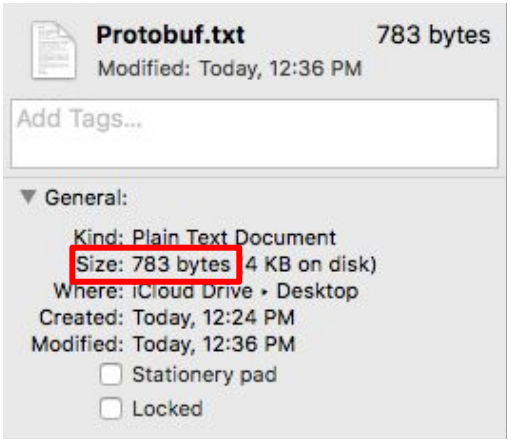
Protocol buffers are now Google's *lingua franca* for data – at time of writing, there are 48,162 different message types defined in the Google code tree across 12,183 `.proto` files. They're used both in RPC systems and for persistent storage of data in a variety of storage systems.

https://developers.google.com/protocol-buffers/docs/overview

- Google
- Twitter
- Apache Mesos
- Netty
- Open StreetMap
- UbuntuOne
- Instagram before React








```
<0>
   <first_name>Darrell</first_name>
   <last_name>Kruse</last_name>
   <address></address>
   <phone_numbers>
      <type>LANDLINE</type>

<number>(540)3542911</number>
   </phone_numbers>
</0>
```

```
{"first_name": "Darrell", "last_name":
"Kruse", "address": {"address_lines":
[]}, "phone_numbers": [{"type":
"LANDLINE", "number":
"(540)354-2911"}]}
```

```
25 0d 0a 44 61 72 72 65 6c 6c 4b 72
75 73 65 22 0d 0a 28 35 34 30 29 33
35 34 2d 32 39 31 31 0d 0a 47 69 75
73 65 70 70 65 48 75 74 63 68 69 6e
73 0d 0a 30
```


How does it work?

.Proto file

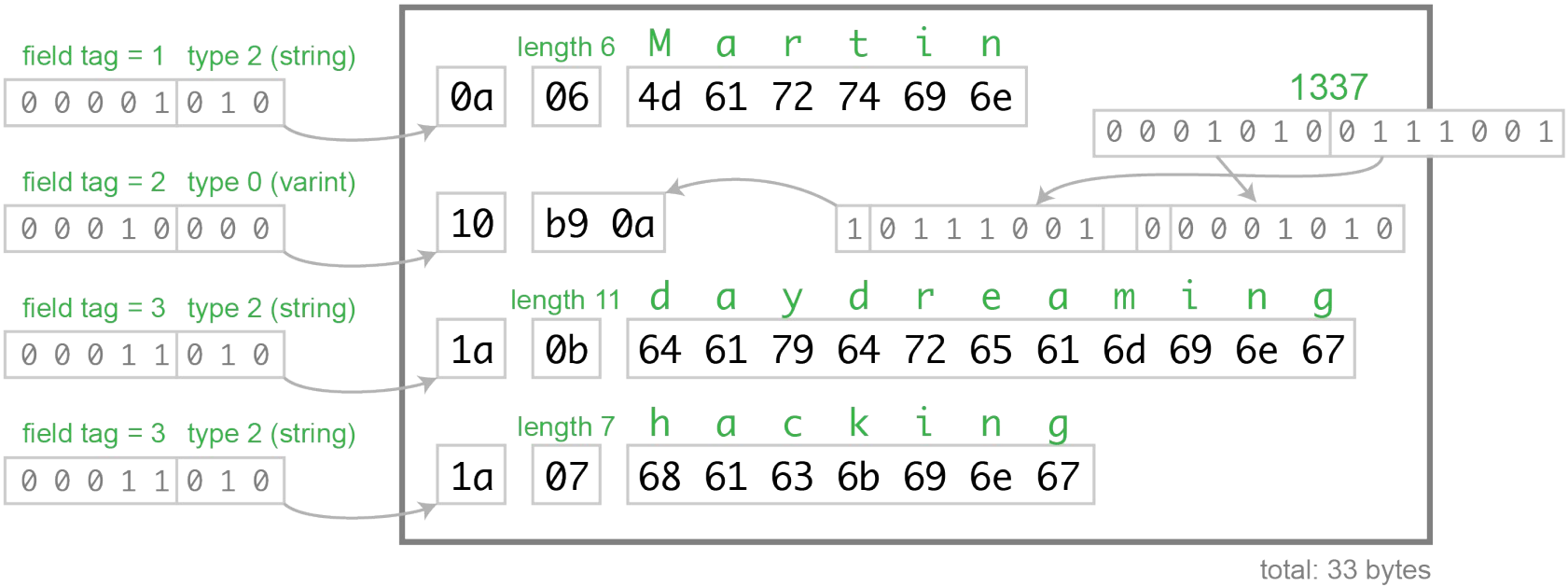
```
syntax = "proto2";

message AddressBook {
  repeated Contact contacts = 1;
};

message Address {
  repeated string address_lines = 1;
  optional string postcode = 2;
};

message Contact {
  required string first_name = 1;
  required string last_name = 2;
  optional Address address = 3;
  repeated Phone phone_numbers = 4;
};
```

Protocol Buffers



total: 33 bytes

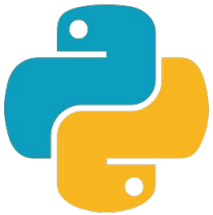

Setup

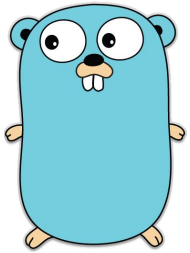
Requirements

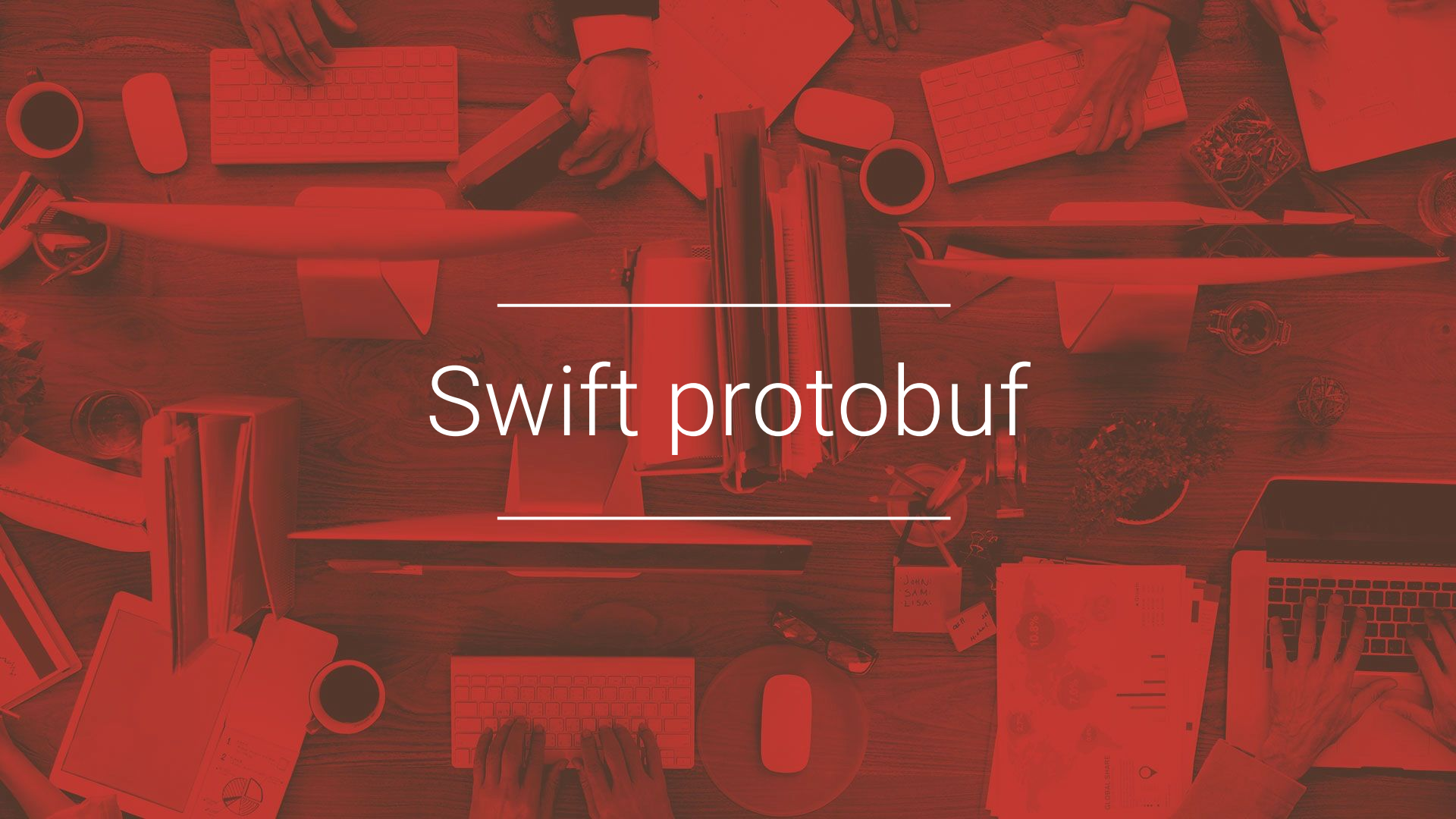
- autoconf
- automake
- libtool
- curl (used to download gmock)
- make
- g++
- Unzip
- xcode-select

https://github.com/google/protobuf/blob/master/src/README.md

Language support



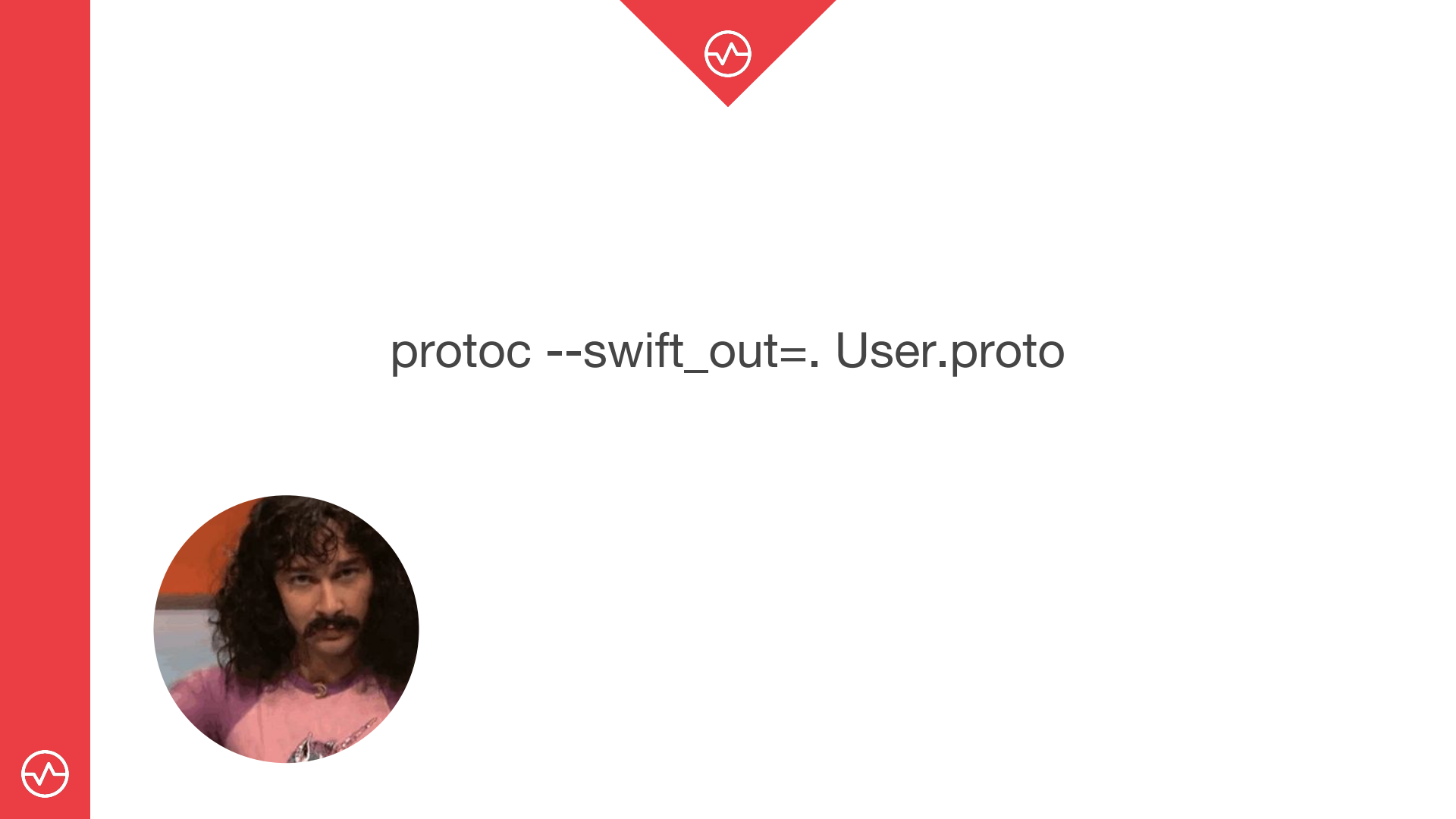



Swift protobuf

Swift Protobuf

https://github.com/apple/swift-protobuf

This project provides both the command-line program that adds Swift code generation to Google's `protoc` and the runtime library that is necessary for using the generated code. After using the protoc plugin to generate Swift code from your .proto files, you will need to add this library to your project.



protoc --swift_out=. User.proto


Demo Time


https://goo.gl/W6PZS7

More resources

http://bit.ly/protoRef

https://github.com/google/protobuf

http://thrift.apache.org

https://grpc.io

https://github.com/apple/swift-protobuf


THANK
YOU
WIZELINE®

WIZELINE ACADEMY

UPCOMING COURSES

NODE.JS WORKSHOP
September 21 @CDMX, Livestreamed

TECH TALK SERIES: AN INTRO TO VAULT
September 27 @CDMX, Livestreamed

MICROSERVICES WITH GOLANG WORKSHOP
September 28 @GDL, Livestreamed

PROBLEM SOLVING & ALGORITHMS CRASH COURSE
Sept. 30-Oct. 1 @GDL

VUE.JS WORKSHOP
October 7 @GDL, Livestreamed

@WizelineAcademy

/WizelineAcademy

academy.wizeline.com

Get notified about courses:
tinyurl.com/WL-academy