Google Protocol Buffers

```
message Presentation {
    required string name      = 1;
    optional string date      = 2;
    optional string presenter = 3 [default = “Warwick”];
    repeated string attendee  = 4;
}
```

Warwick Hunter

w.hunter@computer.org

What are Protobufs?

Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data – think XML, but smaller, faster, and simpler.

You define how you want your data to be structured once, then you can use special generated source code to easily write and read your structured data to and from a variety of data streams and using a variety of languages – Java, C++, Python, Swift.

Why bother?

- Never write serialisation/deserialisation/parsing code again and forget about data packing and endianisms.
- Cross language: Java, Python, C++, Swift, Objective-C...
- Easy to change the protocol/structure and maintain backward compatibility.
- Easy to debug and build protocol tracing tools with generated toString() capabilities.
- Compact serialised representation.

Example

- Mythical photo sharing application
- Code and presentation available on GitHub
  https://github.com/warwickhunter/play/tree/master/protobuf_preso

Under the Covers

- Messages are encoded using the field number you assign and a type indicator. The type indicator tells the decoder how long a serialised value is so it can skip values it doesn't understand
- Numbers, including field numbers are encoded in a 7 bit variant scheme. Top bit set on a byte means there is more to come.
- Details: https://developers.google.com/protocol-buffers/docs/encoding

Lessons Learned

- Very important to use the version of the protobuf compiler that matches the jars/libraries you ship with your application
- An element marked “required” is forever, otherwise you break backward compatibility
- Adding new elements to the protocol/structure never breaks compatibility as long as you maintain the field number ordering. Don't renumber the fields.

Developing

Tools

- Protobuf compiler available in most Linux repos and from Google for Mac and Windows
- Protobuf jar or libraries distributed with your application

Complimentary implementations

- https://code.google.com/p/protobuf-embedded-c/
- https://github.com/booyah/protobuf-objc
- https://code.google.com/p/protobuf/wiki/ThirdPartyAddOns

Questions

Thank You