



CLOUD COMPUTING APPLICATIONS

Protocol Buffers and Thrift

Prof. Roy Campbell

Protocol Buffers

- Invented by Google
- Build on RPCs
- Language-neutral, platform-neutral
- Extensible mechanism
- Serializing structured data
- For distributed services

Example Schema

```
1. Message Person{  
2.     required int32 id = 1;  
3.     required string name = 2;  
4.     optional string email = 3;  
5. }
```

Example Code (Ruby for example)

```
1.  #!/usr/bin/env ruby
2.  # Generated by the protocol buffer compiler. DO NOT EDIT!
3.  require "protocol_buffers"

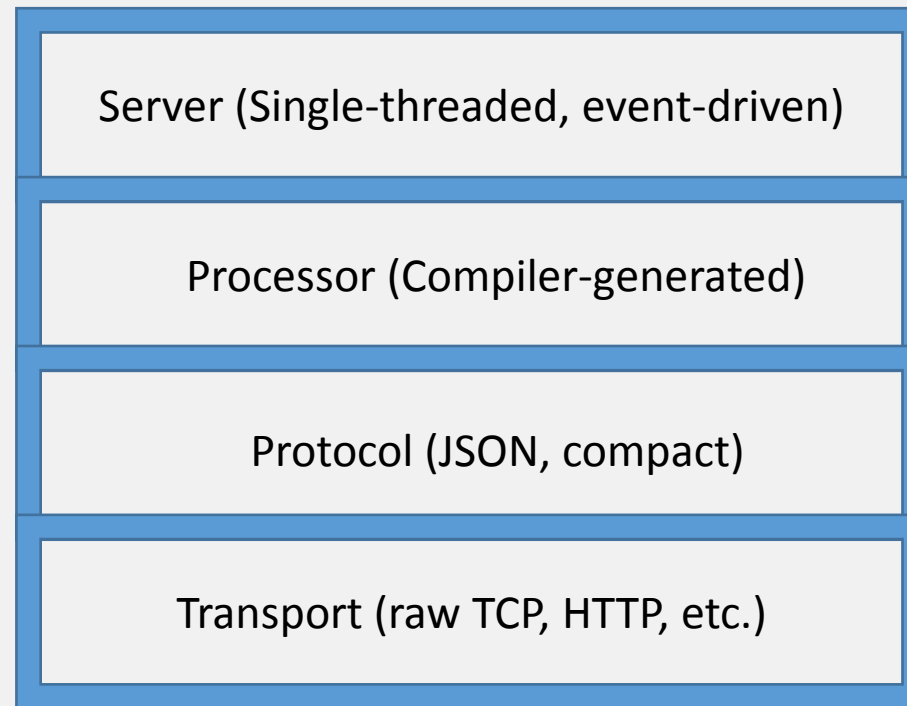
4.  # forward declarations
5.  class Person < ::ProtocolBuffers::Message; end

6.  class Person < ::ProtocolBuffers::Message set_fully_qualified_name "Person"

7.  required :int32, :id, 1
8.  required :string, :name, 2
9.  optional :string, :email, 3
10. end
```

Thrift Network Stack

Interface Definition Language. Creates files for clients and servers from needs to serialize structured data (Facebook)



<https://thrift.apache.org/static/files/thrift-20070401.pdf>

Transport Methods

- Transport
 - open
 - close
 - read
 - write
 - flush
- Server transport also has open, listen, accept, and close allowing
 - **Read / write to / from a file on disk**
 - **http**

Example File Transports

- TFileTransport – This transport writes to a file
- TFramedTransport – Transport for non-blocking server
- TMemoryTransport – Uses memory for I/O
- TSocket – Uses blocking socket I/O for transport
- TZlibTransport – Performs compression using [zlib](#)

Example server codes

- TNonblockingServer – A multi-threaded server using [non-blocking I/O](#)
- TSimpleServer – A single-threaded server using standard blocking I/O. Useful for testing.
- TThreadPoolServer – A multi-threaded server using standard blocking I/O.

Example Schema

```
1. struct Person{  
2.     int32 id = 0,  
3.     string name,  
4.     optional string email  
5. }
```