


Protocol Buffers

c1/4h von sECuRE

NoName e.V., 2012-03-22

powered by 

Protocol Buffers

- „Protocol Buffers are a serialization format with an interface description language developed by Google.“
- FOSS, BSD-Lizenz
- ...wie XML, aber ganz anders
- Datenspeicherung, Datenaustausch (RPCs)
- Auf- und Abwärtskompatibel (think JSON)

Sprachen

- Referenzimplementation: C++, Java, Python
- Beschreibungsformat, was dann kompiliert wird

Beispiel: stats.proto

```
message SystemStats {  
  required string kernel_version = 1;  
  optional int64 uptime = 2;  
  repeated Process processes = 3;  
}
```

```
message Process {  
  required int32 pid = 1;  
  
  enum ProcessState {  
    RUNNING = 0;  
    DEAD = 1;  
  };  
  
  optional ProcessState state = 3 [default = RUNNING];  
}
```

Beispiel: main.cc (C++)

```
#include <iostream>
#include <fstream>
#include "stats.pb.h"

using namespace std;

int main() {
    SystemStats stats;
    stats.set_kernel_version("3.2");
    Process *process1 = stats.add_processes();
    process1->set_pid(32768);

    fstream output("/tmp/output", ios::out | ios::binary);
    stats.SerializeToOstream(&output);
}
```

Kompilieren

```
# apt-get install protobuf-compiler libprotobuf-dev
$ protoc --cpp_out=. stats.proto
$ g++ -o main main.cc stats.pb.cc \
    $(pkg-config --cflags --libs protobuf)
$ ./main
$ hd /tmp/output
00000000  0a 03 33 2e 32 1a 04 08  80 80 02 |..3.2.....|
0000000b
```

Vergleich zu XML

- Beispiel aus der Doku:
 - Platz: 28 Bytes (PB) gegenüber 69 Bytes (XML)
 - Parsen: 100-200 ns (PB) gegenüber 5000-10000 ns (XML)
- (relativ) einfacher, generierter Code
- Allerdings nicht human-readable (XML ein bisschen)

Links

- <http://code.google.com/apis/protocolbuffers/>
- http://en.wikipedia.org/wiki/Protocol_Buffers