Introduction

• The Health Level Seven (HL7)-Protocol provides the most relevant standard for data transfer and communication within Hospital Information Systems (HIS) [1].
• The upcoming HL7 Fast Healthcare Interoperability Resources (FHIR) utilizes, among others, XML and Representational State Transfer (REST) [2].
• FHIR resources have to be serialized, transmitted and deserialized during each data exchange.
• With vast amounts of medical (especially mobile) devices waiting to be integrated in hospital information systems in the future, this faces challenges concerning computational performance and bandwidth load.
• **Objective:** Exploration of an alternative binary serialization approach using Google Protocol Buffers (Protobuf) to decrease (de-)serialization computation time while also decreasing needed bandwidth.

Materials and Methods

• **Protobuf:** Binary format to (de-)serialize objects using three steps; a) Protobuf definition file describing the structure (.proto), b) automated translation of the .proto file into a computer language (e.g. Java) and c) inclusion of generated source in project.
• **FHIR:** Set of XML1 schema files describing one resource (e.g. patient) in a clinical context. Resources are posted to specific REST endpoints, executing one operation (e.g. admission).
• **Transformation:** The public available FHIR-XML schema are converted into .proto definitions, while conserving type safety and cardinalities.
• **Evaluation:** A patient’s XML schema is transformed into a .proto definition. Sample data containing 29 attributes are repeatedly (de-)serialized on a 800 MHz system; computation time and transferred data size is measured.

Results

• **Results for 10,000 samples with pre-initialized (de-)serializers**
  • Decreased time needed for serialization: 2,764 ms for XML vs. 984 ms for Protobuf (35% of time compared to XML).
  • Decreased time needed for deserialization: 446 ms for XML vs. 7.461 ms for Protobuf (71% of time compared to XML).
  • Decreased size of one serialized resource: 1,052 bytes for XML vs. 315 bytes for Protobuf (30% of size compared to XML).
• If the (de-)serializer is created on-demand, using 1,000 samples, to decrease the overall memory usage, the Protobuf performance will be even better compared to XML, as the Protobuf (de-)serializer is instantiated much faster.

Discussion

• **Improved performance of communication in terms of computation time and bandwidth.**
• Transformation of FHIR-XML schema to Protobuf definition could also be done automatically. Currently, the applied approach results in too many intermediate types (decreasing performance) [9].
• Protobuf is available for many languages, so highly portable.
• By replacing HTTP with Protobuf envelope performance will increased even more (but at the price of giving up REST-conformity).
• **Main drawback:** Communication not human readable anymore

Contact
Dr. rer. nat. Dipl.-Phys. Cord Spreckelsen
cspreckelsen@mi.rwth-aachen.de

---

1) Also available as JSON descriptions.