Google's Protocol Buffers (Protobuf)

Particle

+ pdg()
+ vertex()
+ p()
+ ...

Generated protobuf code
(native C++, Python, Go, Java, etc.)

0100101011100001010...

protoc

protobuf wire format

protobuf "language" describes simple
data types from unsigned ints to maps
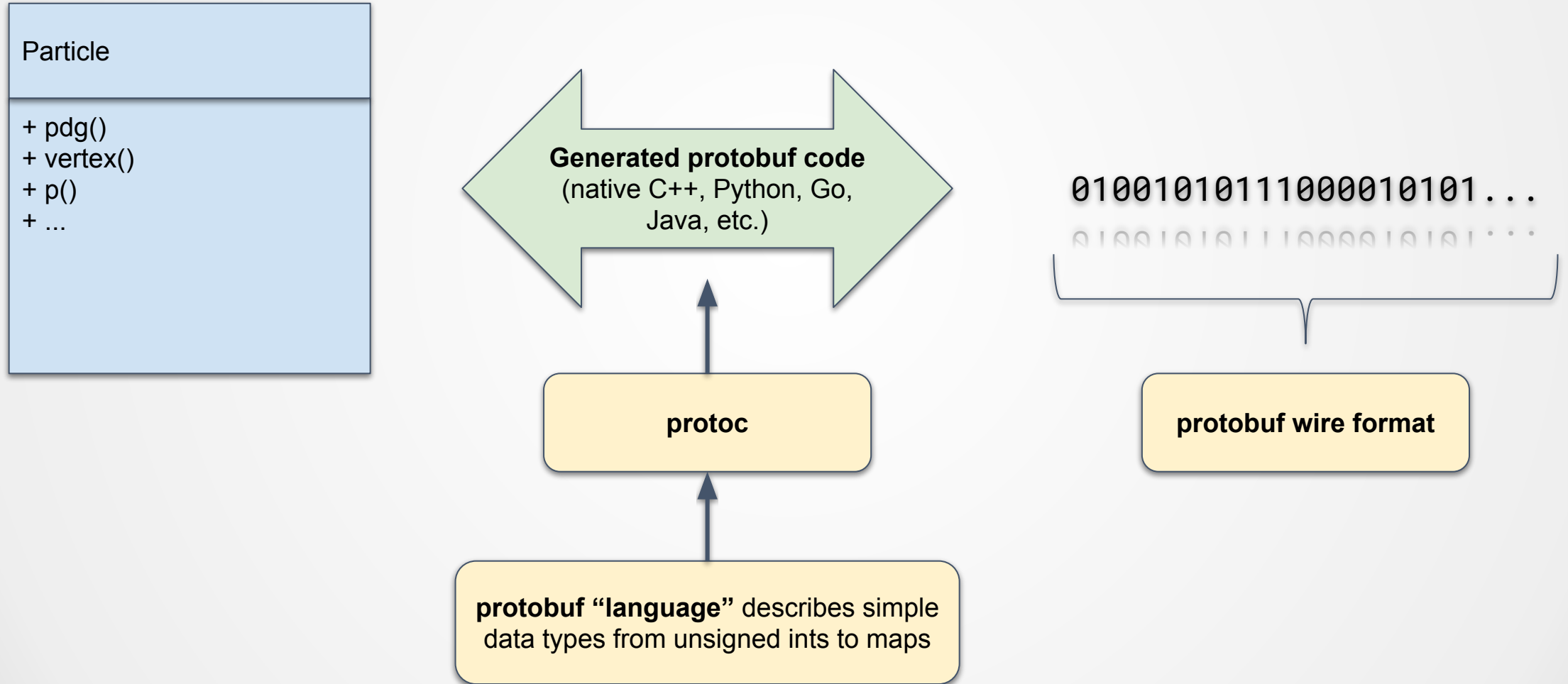
Notable Features

Pros

- Language neutrality
- Varint compression and natural compression of sparse data
- Forward/backward compatibility
  - Fields (members) can be safely added and removed over time
- Developed and maintained by IT industry

Cons

- Low-level: *HEP-oriented features have to be added on top*
  - E.g. additional compression, lazy decoding, random access
- Field identifiers add size overhead for dense data

ProIO

- Project for utilizing protobuf for HE(N)P in a language-neutral way
  - C++, Python, Go, and Java native libraries already implemented*
- Supported by ANL LDRD and eRD20 (multi-lab EIC Software Consortium)
- Based on pioneering work by Sergei Chekanov (ANL) and Alexander Kiselev (BNL)
- Currently migrating from https://github.com/decibelcooper/proio to https://github.com/proio-org
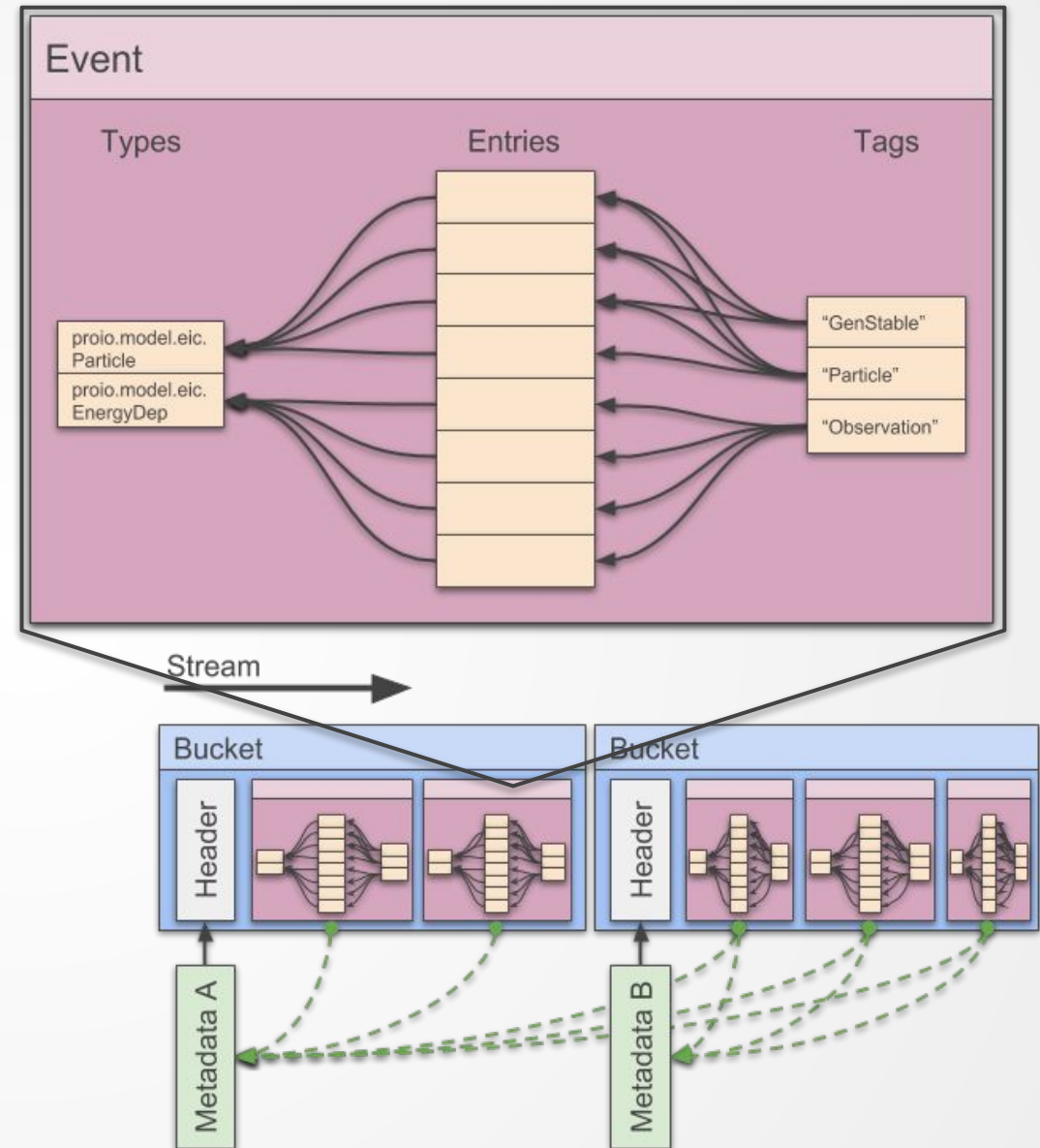
*Java implementation is currently incomplete, but read functionality is there

Data Size and Varint (fixed-point) Compression

Data size LHCb OpenData

pp Events from Pythia8

J. Blomer, A quantitative review of data formats for HEP analyses ACAT 2017

http://atlaswww.hep.anl.gov/hepsim/info.php?item=326