ASTM INTERNATIONAL
Helping our world work better

Open Vector Format for laser-based computer aided manufacturing

Sebastian Dirks, M.Sc.
19.11.2020

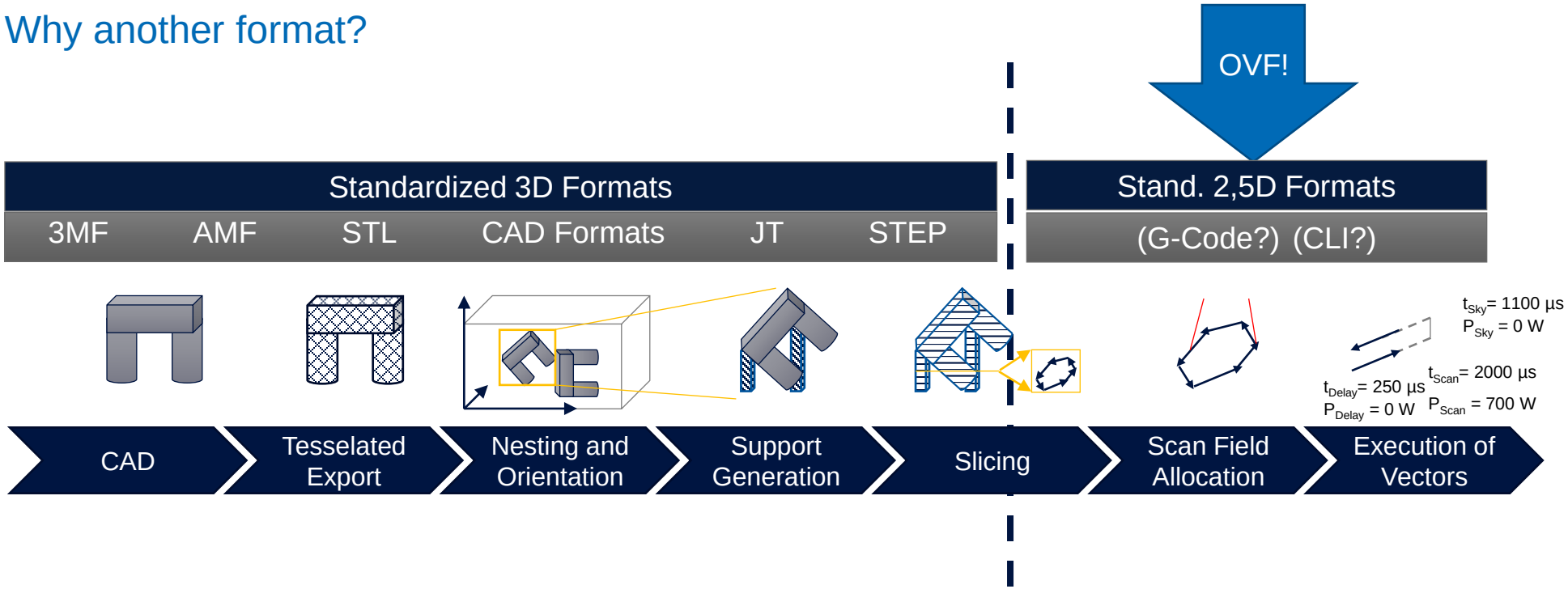
www.amcoe.org

Sebastian Dirks

- From Aachen, Germany
- Studied mechanical engineering and automation technology at RWTH Aachen University
- Student assistant at Fraunhofer Institute for Laser Technology ILT
- Research assistant at RWTH Aachen University – Chair of Digital Additive Production (DAP) since 2018
- Main topics:
  - LPBF production data chains
  - Build time and economic analysis
  - Production data networking and data based optimization

Open Vector Format for laser-based computer aided manufacturing

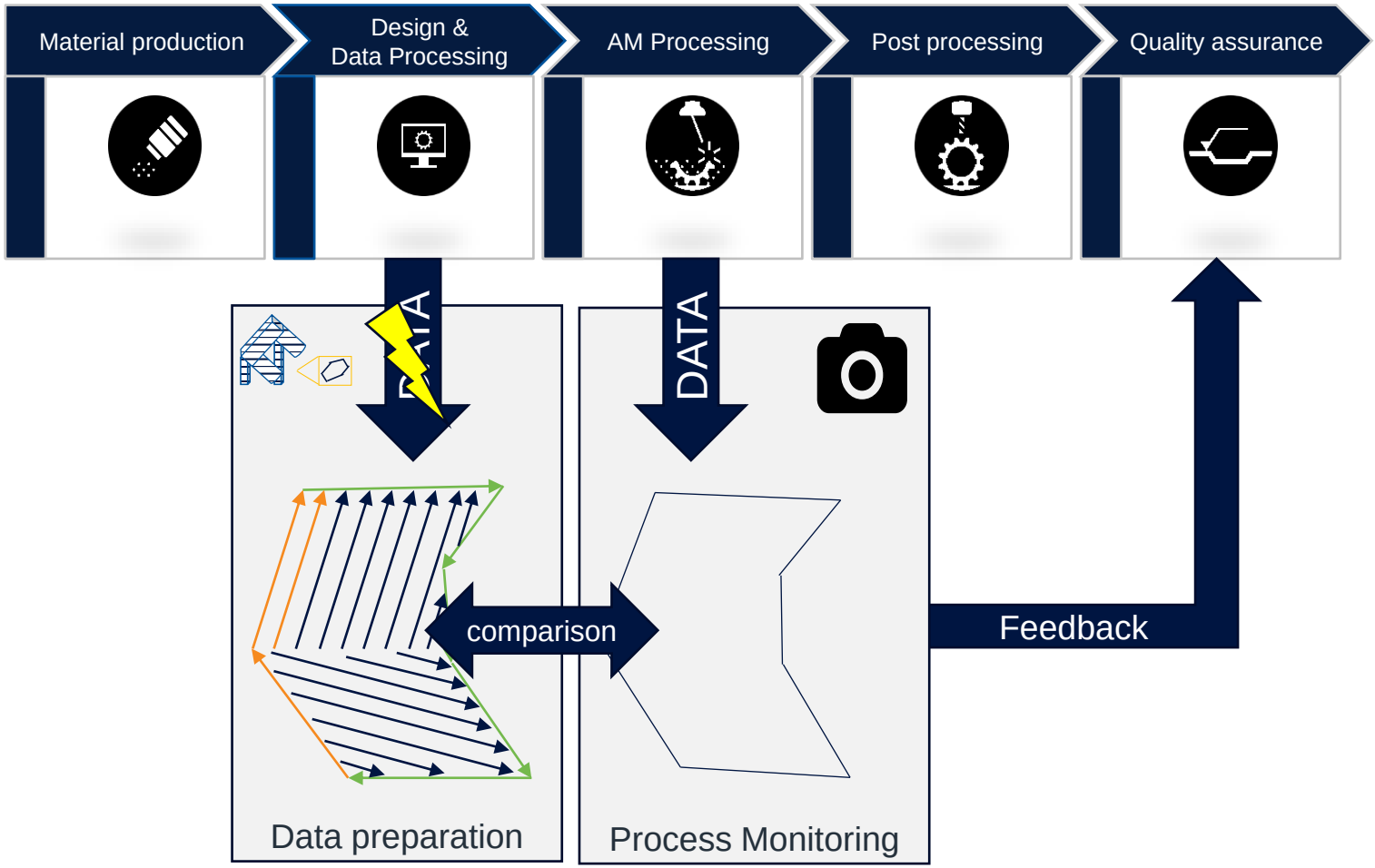
Why another format?



- Vector based – aims at machine commands AFTER slicing
- No currently standardized format does the job for laser-based AM

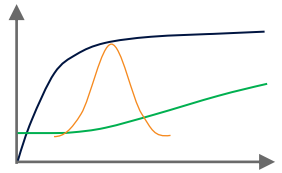
Open Vector Format for laser-based computer aided manufacturing

Data along the AM process chain



More examples

Production KPI calculations

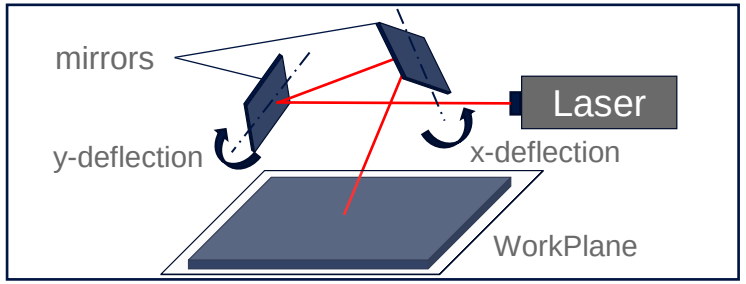


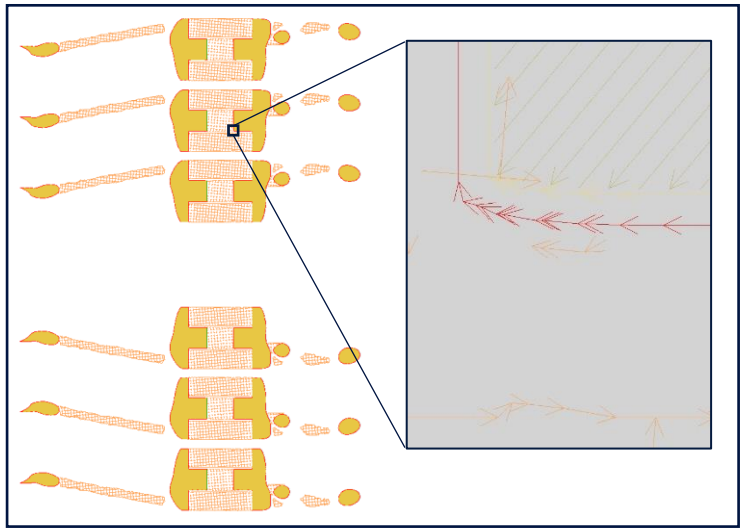
Machine transferability, performance and benchmarking

The requirements

Why not just use G-Code?

- Should abstract hardware to a certain extent
- Requirements are mainly driven by the galvanometer scanners used to deflect laser beams programmatically
- Handling of several gigabytes of vector coordinates (thousands of layers per build)
- Inclusion of process parameters and indirect hardware commands (e.g. laser delays, process states)





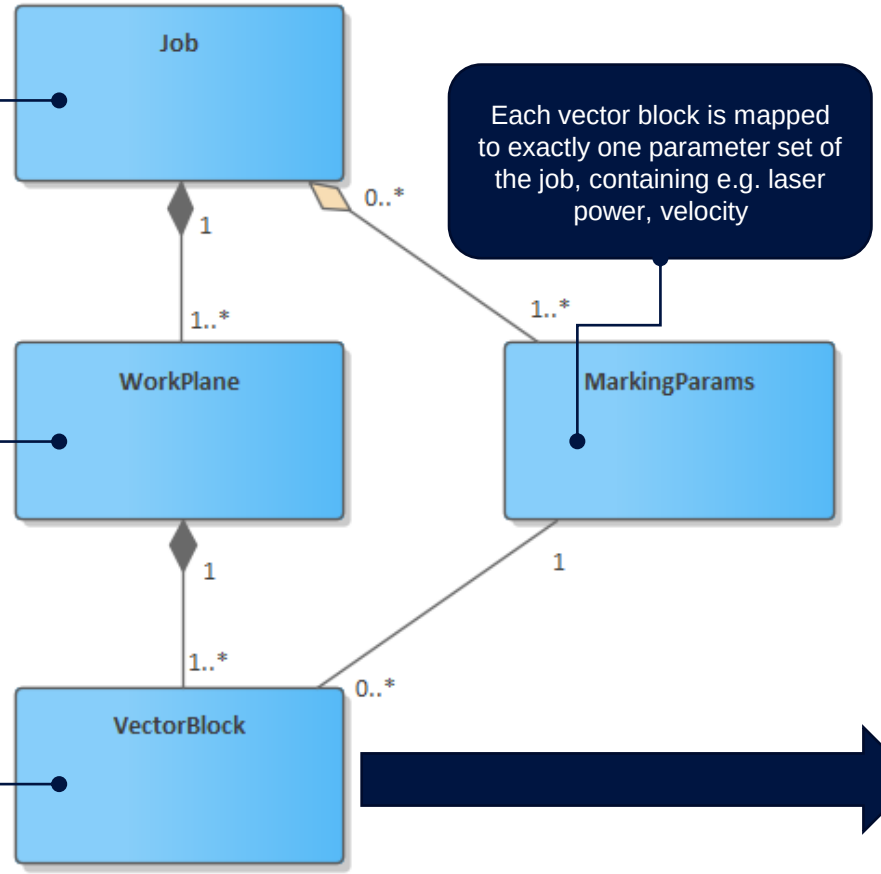
- ✓ Hardware abstraction
- ✗ Gigabytes of vector data
- ✗ Fast data handling
- ✗ Process parameters

Basic object model

Basic object is a Job

A job consists of work planes: 2D drawing plane located in a 3D space

Each work plane consists of vector blocks that can be one of various types, executed in order



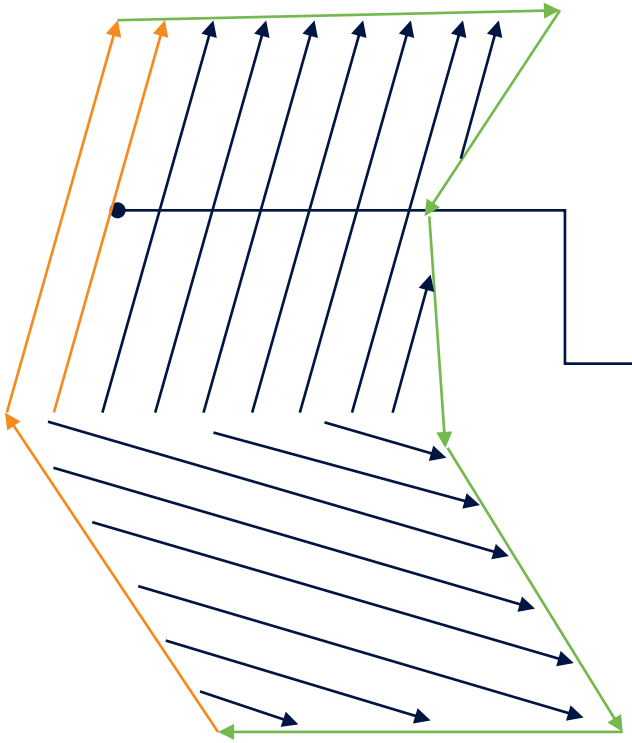
Special vector block types cover edge cases like dynamic parameter adaption per vector that require additional parameters per vector

Data model is streamlined at all applications that use motion systems to reposition a part relative to a 2D deflection system, e.g. LPBF

The most important Vector Blocks:

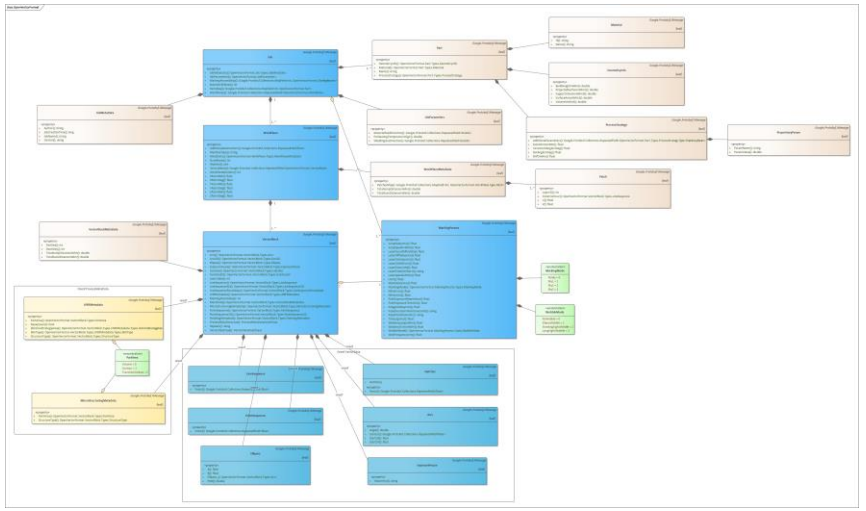
Polygones	Hatches
Arcs	Ellipses
Points	Exposure pauses

Meta Data

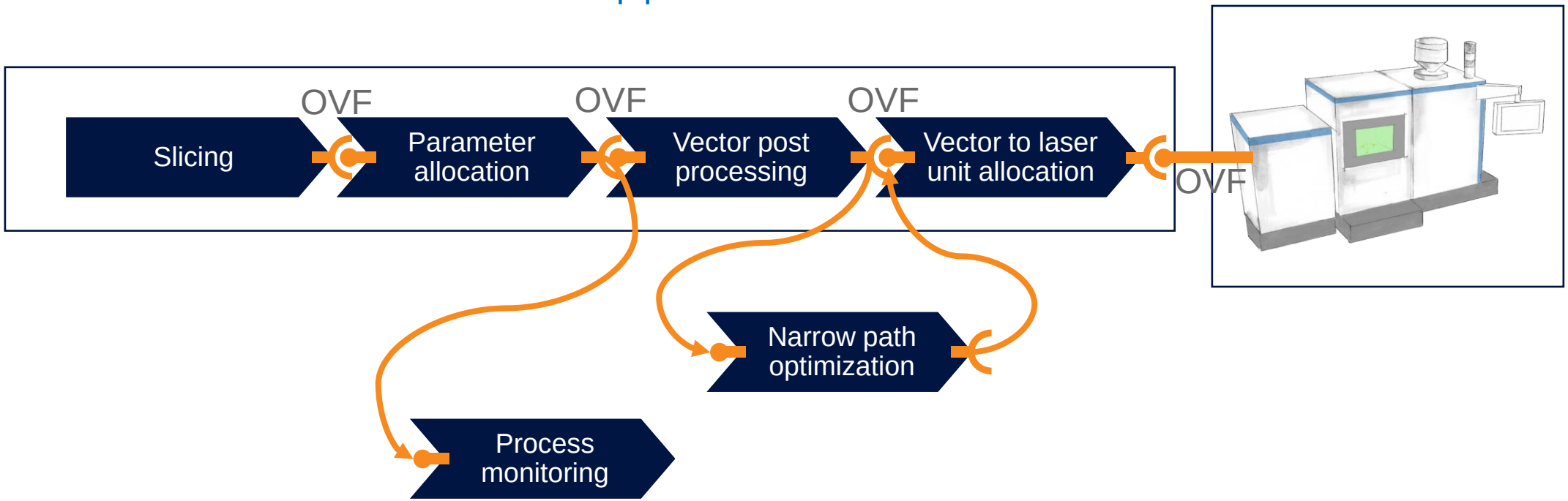


This vector:

- Is located in vectorBlock 3 in layer 25
- Belongs to “part1” with a volume of 216 mm³
- StructureType is **SUPPORT**
- **b**elongs to the filling **VOLUME**
- Is not a **REEXPOSURE**
- Belongs to the **DOWNSKIN**
- Has been created using the “**checkerboard**” strategy with a hatch distance of 50 µm



Native Performance enables continues pipeline

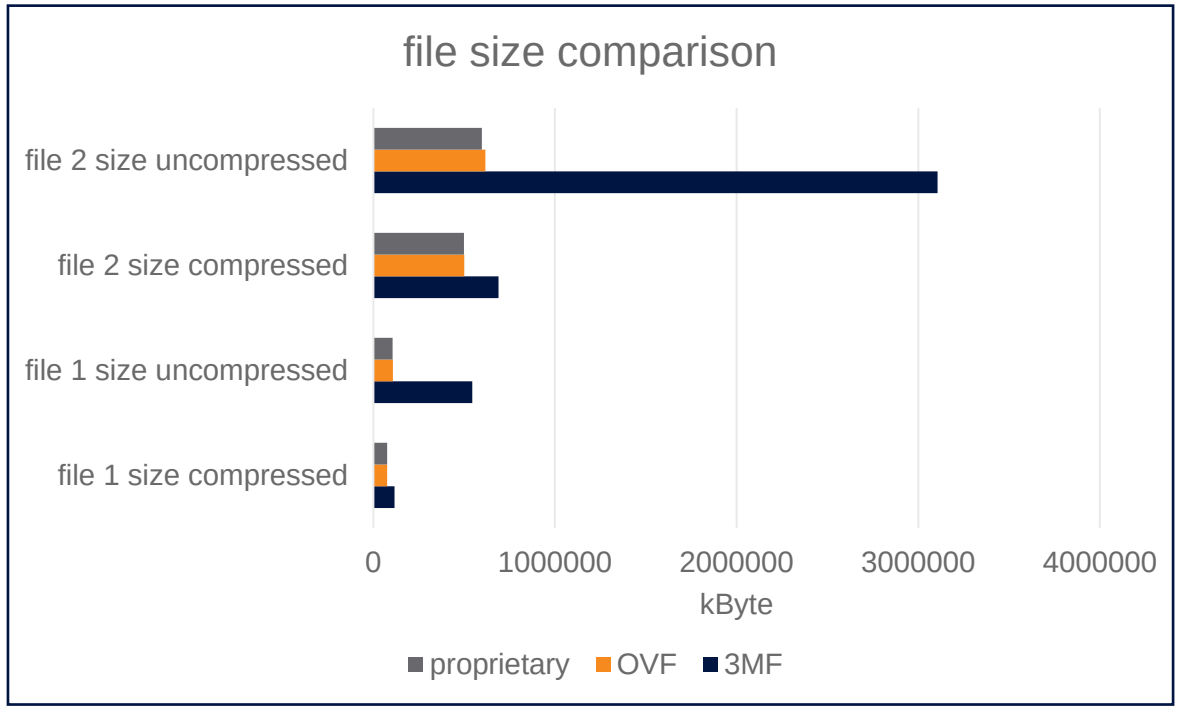


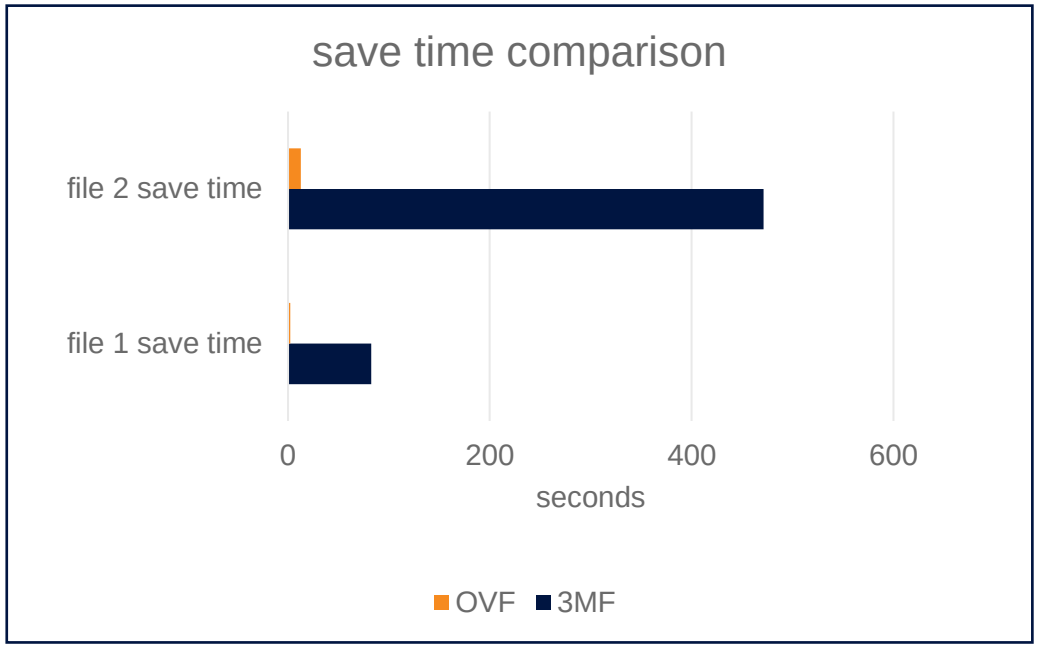
- Addon pipeline steps from third parties can easily be added
- A continues vector based pipeline is only possible if import/export are fast and preserve metadata

The secret sauce: Protobuf

- Open source technology **Protocol Buffers** created and used by Google for structured data
- Widely used, e.g. in TensorFlow for machine learning
- Includes an Interface Description Language (**IDL**)
- Advantages of generic **structured data** approaches like XML or JSON combined with **faster, near native performance**
- **Easier to use** than XML in multiple programming languages because of **source code generation**
  - No string matching
- Streamable
  - Enables modern micro service architecture
- Programming **language independent**
- Binary forwards-/backwards compatible
- **Downside**: **not human readable**
  - geometry coordinates can’t be edited with a text editor, needs Protobuf editor

Performance comparison





- saving call only was measured in debug mode
- For 3MF official C# 3MFLib.dll was used
- Hardware: AMD Ryzen PRO 2500u Win10
- Toshiba KXG50ZNV 512GB PCIE SSD

– Vector data of two jobs is written in a proprietary format, OVF and 3MF (slice extension, XML based)

– Size: **5 times** as big uncompressed, 3MF compressed still bigger than uncompressed OVF

– Serialization performance: saving time **37 times as high**

Open Source and meant to contribute

- Used in labs of RWTH Aachen and Fraunhofer ILT
- Components already available
  - Conversion framework for legacy formats, e.g. CLI
- Meant as a blueprint for standardization needs
  - Shows which meta data is needed for what use cases
  - Showcases data structure implementation to efficiently store the meta data

ASTM INTERNATIONAL
Helping our world work better

Thank you.

Sebastian Dirks
19.11.2020

www.amcoe.org