welding data exchange
open science initiative

BAM
Bundesanstalt für Materialforschung und -prüfung

Sicherheit in Technik und Chemie

17.05.2022

WELDX – A FILE FORMAT FOR PROCESSING AND ARCHIVING WELDING RESEARCH DATA

C. Fabry, A. Pittner, M. Rethmeier, Berlin

Group of Welding Technology, Department of Component Safety
Bundesanstalt für Materialforschung und -prüfung

www.bam.de

contents

WelDX
welding data exchange format

- introduction
- gas metal arc welding
- motivation
- WelDX
  - core features
  - file format
  - Python API
  - examples
  - ecosystem
- outlook

introduction
contact

www.bam.de/weldx

https://github.com/BAMWelDX/weldx

M.Sc. Cagtay Fabry
cagtay.fabry@bam.de
+49 30 8104-4675

Dr.-Ing. Andreas Pittner
andreas.pittner@bam.de
+49 30 8104-3696

Dipl.-Ing. Volker Hirthammer
volker.hirthammer@bam.de
+49 30 8104-4597

introduction
contact

www.bam.de/weldx

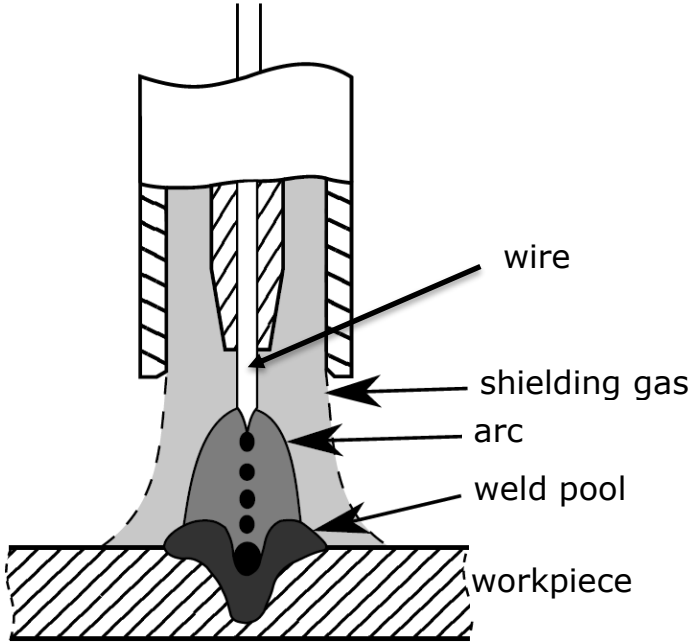
https://github.com/BAMWelDX/weldx

M.Sc. Cagtay Fabry
cagtay.fabry@bam.de
+49 30 8104-4675

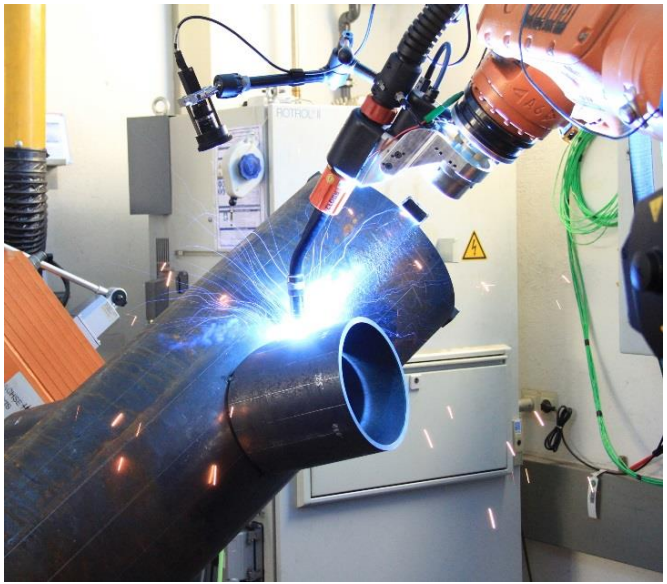
- BAM research associate since 2014
- **background in welding applications and control**
- **“I like doing RDM because I need it.”**

➡ **Build the tools we want to use !**

introduction
Gas Metal Arc Welding (GMAW)

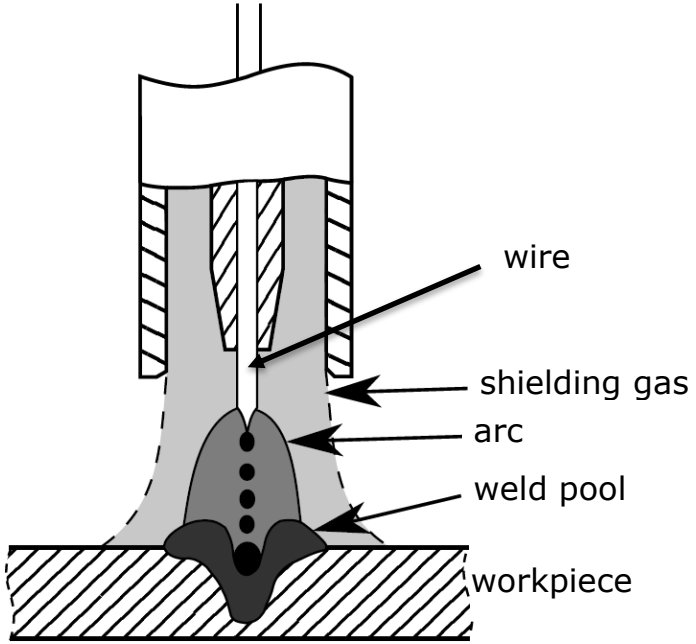


GMAW principle

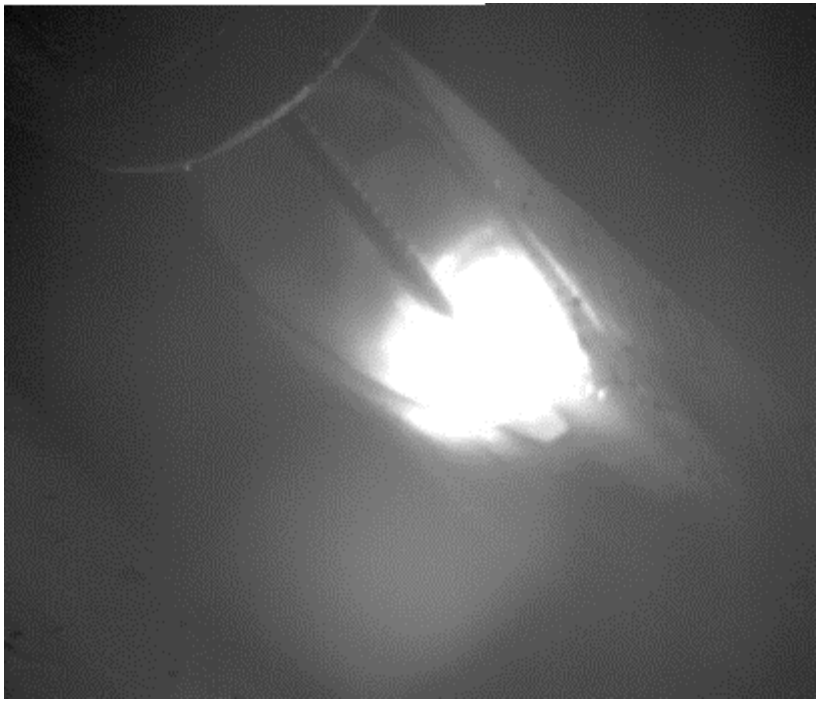


automated GMAW

introduction
Gas Metal Arc Welding (GMAW)

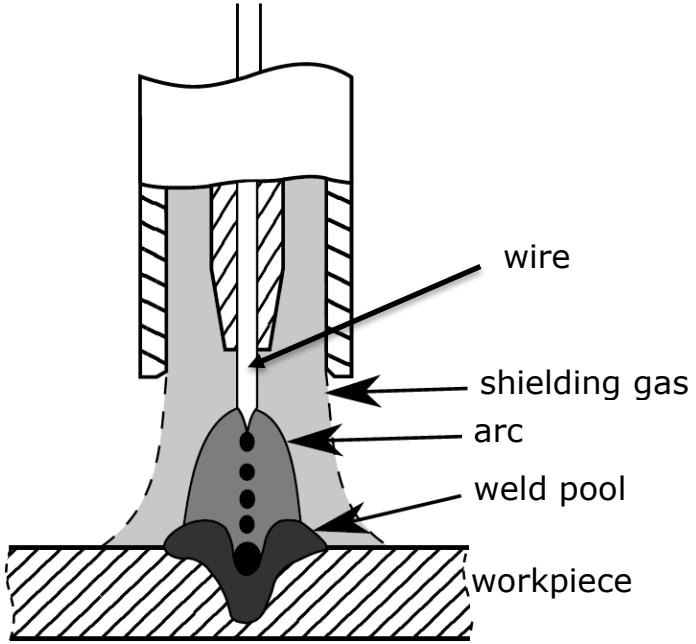


GMAW principle

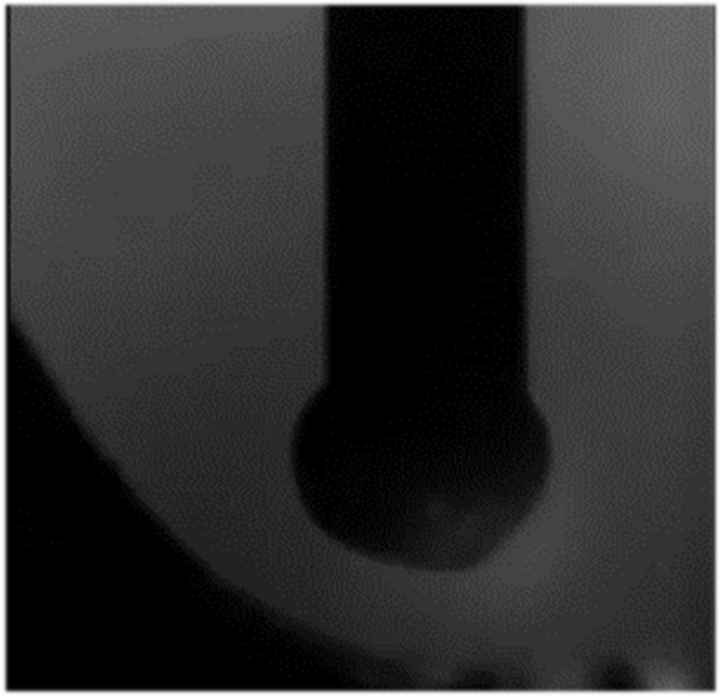


HDR recording

introduction
Gas Metal Arc Welding (GMAW)

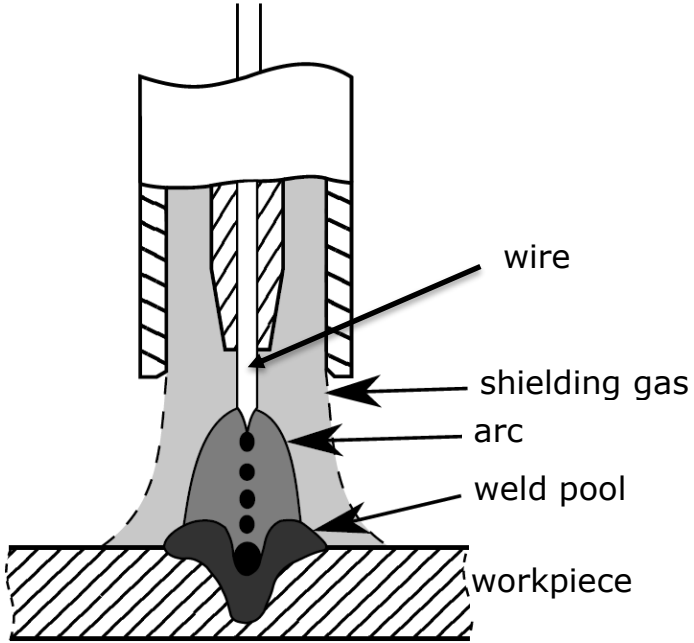


GMAW principle

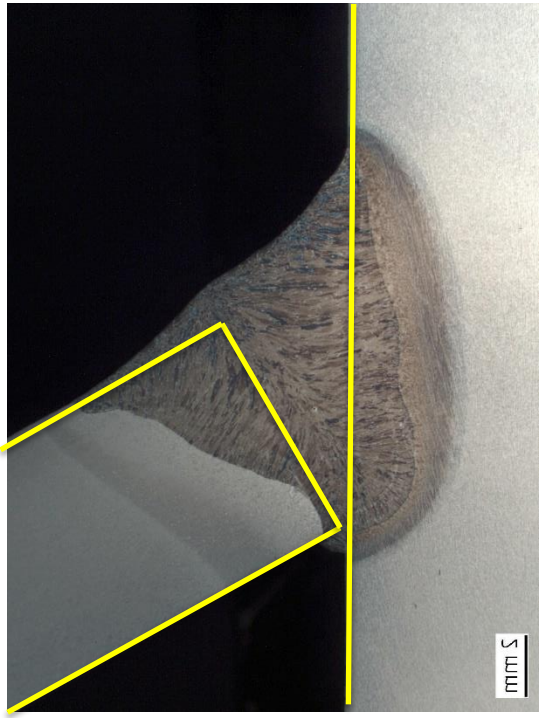


highspeed recording

introduction
Gas Metal Arc Welding (GMAW)

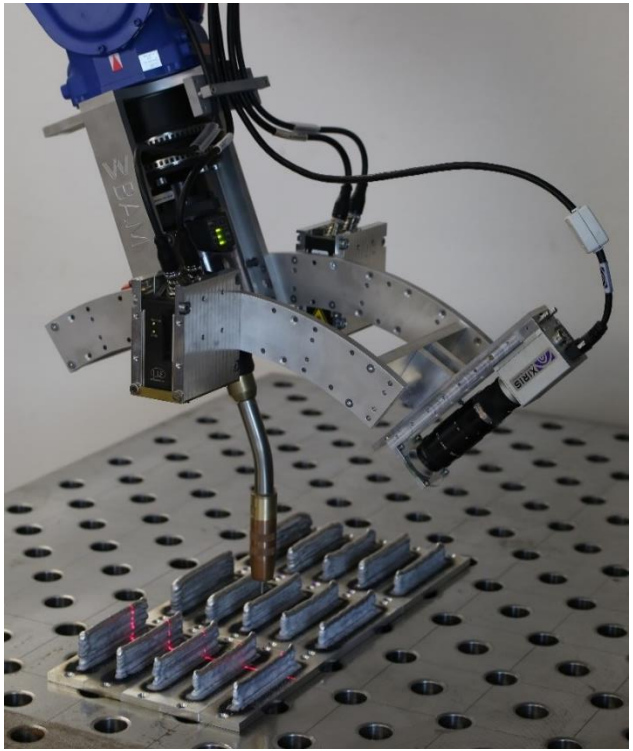


GMAW principle



weld seam cross section

introduction
Gas Metal Arc Welding (GMAW)



motivation

- promote use of Open Science and adoption of FAIR principles
  *FAIR - **F**indable, **A**ccessible, **I**nteroperable, **R**eusable*
- promote sharing and publication of welding knowledge as research data
- promote integration of existing community standards into research work

➡ **domain specific file format**

Recommendations for an Open Science approach to welding process research data.
Fabry, C. et al. Weld World (2021). https://doi.org/10.1007/s40194-021-01151-x

goals & aims

- store experimental welding data
- provide file format and open source software
- **suitable for daily research work and archival use**
- develop consistent analysis and visualization tools
- focus on implementation of arc and laser beam welding

➡ **collaborate as a welding research data community**

Recommendations for an Open Science approach to welding process research data.
Fabry, C. et al. Weld World (2021). https://doi.org/10.1007/s40194-021-01151-x

weldx core elements
“what is weldx ?”

1. weldx format
2. weldx standard
3. weldx Python API for welding + tools

weldx core elements
key features

- consistent unit support

- support for time varying data
  - as discrete data
  - as analytical expressions

- complex spatiotemporal data
  - sensors and measurements moving though space and time

- describing measurement chains

welding data exchange format
file format – basic

- The weldx format is an extension to the **Advanced Scientific Data Format** (ASDF)
  - https://github.com/asdf-format/asdf
  - https://doi.org/10.1016/j.ascom.2015.06.004

- combination of **YAML header + binary blocks** in one file

- ASDF is an extended YAML implementation for **JSON schema**

- additional features, including:
  - schema versioning
  - (circular) references
  - custom validators

welding data exchange format file format

weldx file

- store experiment information
- metadata
- experimental setup
- (raw) measurement data
- hybrid text / binary format

➡ **single file**
➡ **storage and research use**
➡ **data exchange**

schema definition files

- multiple different files
- simple text format (YAML)
- describe structure and contents
  - individual elements
  - whole weldx file
- used for validation of weldx files
- curated by individuals or community

➡ **ensure file integrity and structure**
➡ **define community standards**

welding data exchange format file format – layout

File-Header

```
#ASDF 1.0.0
#ASDF_STANDARD 1.5.0
%YAML 1.1
%TAG ! tag:stsci.edu:asdf/
%TAG !weldx! asdf://weldx.bam.de/weldx/tags/
--- !core/asdf-1.1.0
```

File-Metadata

```
asdf_library: !core/software-1.0.0 {author: The ASDF Developers,
  homepage: 'http://github.com/asdf-format/asdf', name: asdf, version: 2.8.3}
history:
  extensions:
  - !core/extension_metadata-1.0.0
    extension_class: asdf.extension.BuiltinExtension
    software: !core/software-1.0.0 {name: asdf, version: 2.8.3}
  - !core/extension_metadata-1.0.0
    extension_class: weldx.asdf.extension.WeldxExtension
    extension_uri: asdf://weldx.bam.de/weldx/extensions/weldx-0.1.1
    software: !core/software-1.0.0 {name: weldx, version: 0.6.0}
```

File-Contents

```
current: !weldx!units/quantity-0.1.0
  value: !core/ndarray-1.0.0
    source: 0
    datatype: int64
    byteorder: little
    shape: [2000]
  units: !weldx!units/units-0.1.0 milliampere
```

welding data exchange format file format – example

weldx file

```
gas_component_1: !weldx!aws/process/gas_component-0.1.0
  gas_chemical_name: argon
  gas_percentage: !weldx!units/quantity-0.1.0
    value: 82
    unit: !weldx!units/units-0.1.0 percent
```

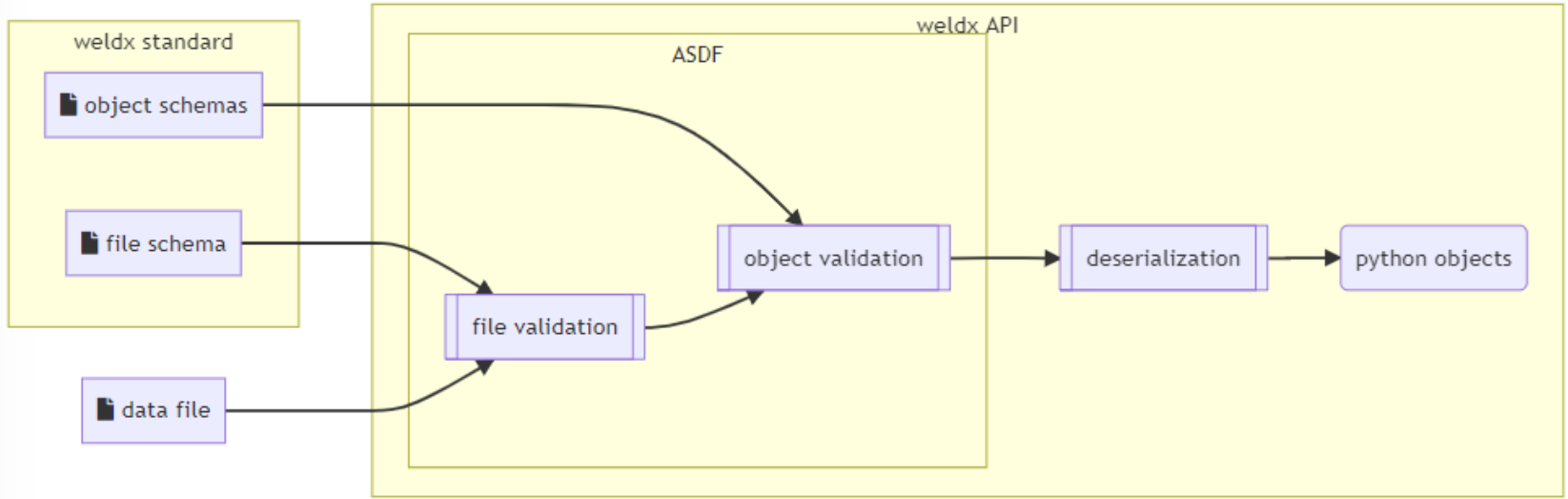
schema definition file

```
id: "weldx/schemas/aws/process/gas_component-0.1.0"
type: object
properties:
  gas_chemical_name:
    type: string
    enum:
      - argon
      - carbon dioxide
      - helium
      - hydrogen
      - oxygen
  gas_percentage:
    description: |
      Percentage by weight this gas occupies
      of the total gas mixture.
    tag: !weldx!units/quantity-0.1.0
    wx_shape: [1]
    wx_unit: percent
required: [gas_chemical_name, gas_percentage]
```

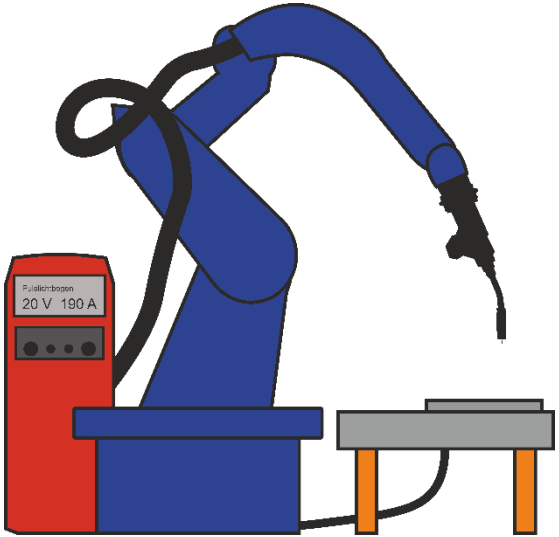
welding data exchange format file format – advanced schema example

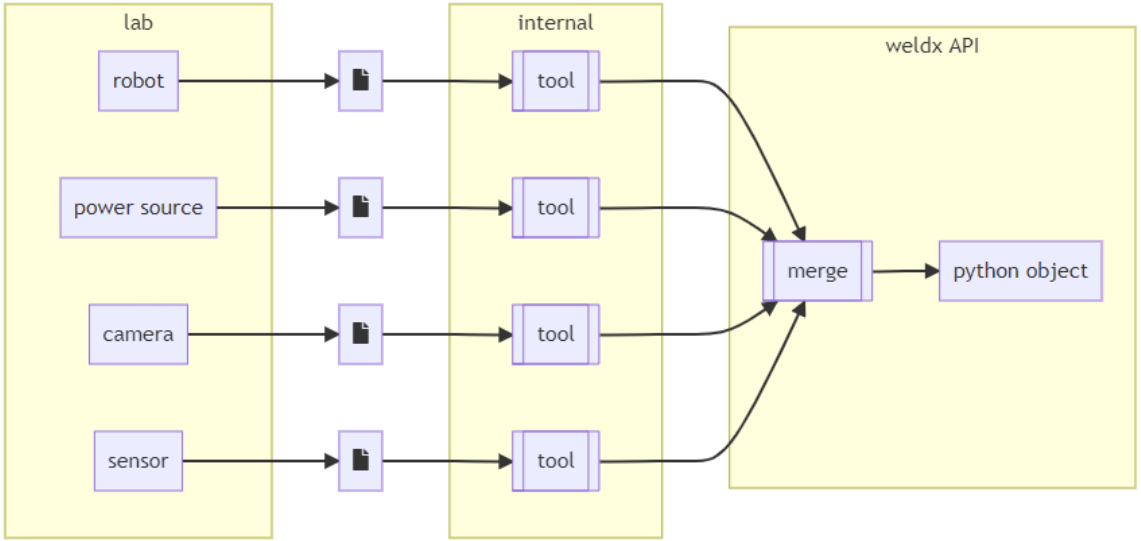
```yaml
# coordinate transformation
type: object
properties:
  coordinates:
    oneOf:
      - tag: "asdf://weldx.bam.de/weldx/tags/core/variable-0.1.*"
      - tag: "asdf://weldx.bam.de/weldx/tags/core/time_series-0.1.*"
    wx_unit: "m"
  rotation:
    tag: "asdf://weldx.bam.de/weldx/tags/core/variable-0.1.*“
  time:
    tag: "asdf://weldx.bam.de/weldx/tags/time/timedeltaindex-0.1.*"
wx_shape:
  coordinates: [..., (t), 3]
  rotation: [..., (t), 3, 3]
  time: [(t)]
```

workflows
opening and validating a weldx file

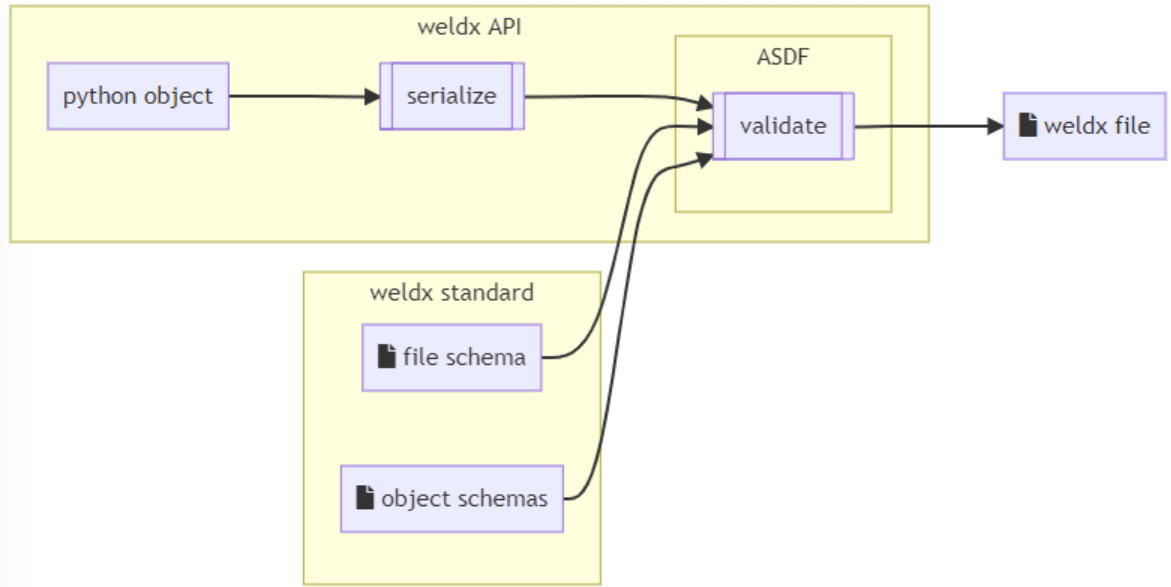


workflows
internal data processing

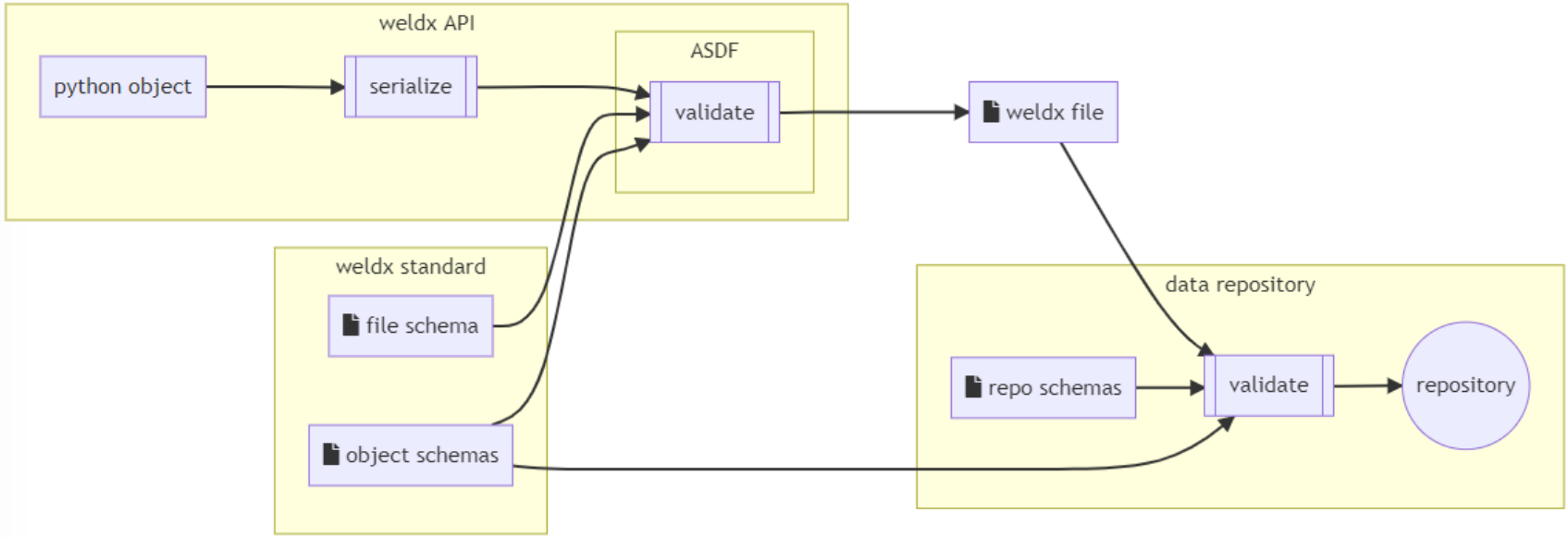




workflows
saving a weldx file



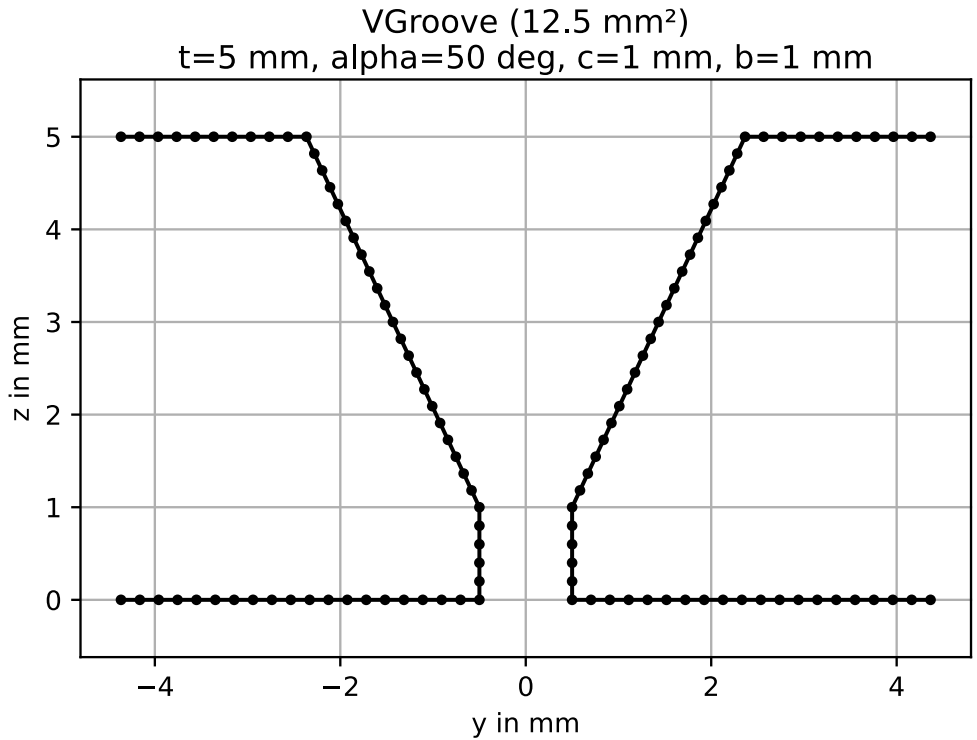
workflows
publishing a weldx file



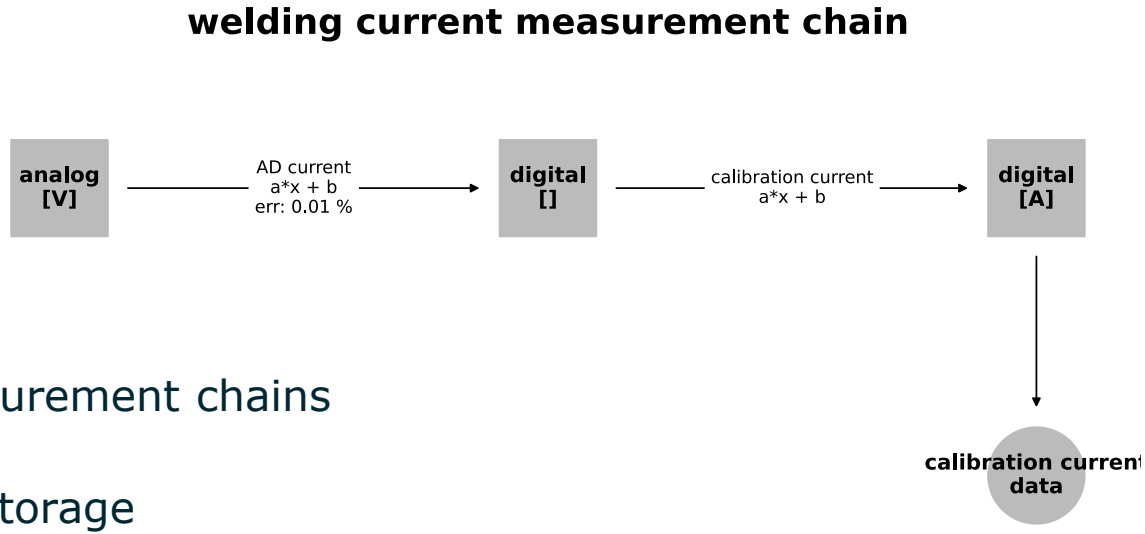
weldx API

weldx API
weldment description

- describe welding applications
- groove types
- materials & workpieces
- power source settings
- shielding gas composition



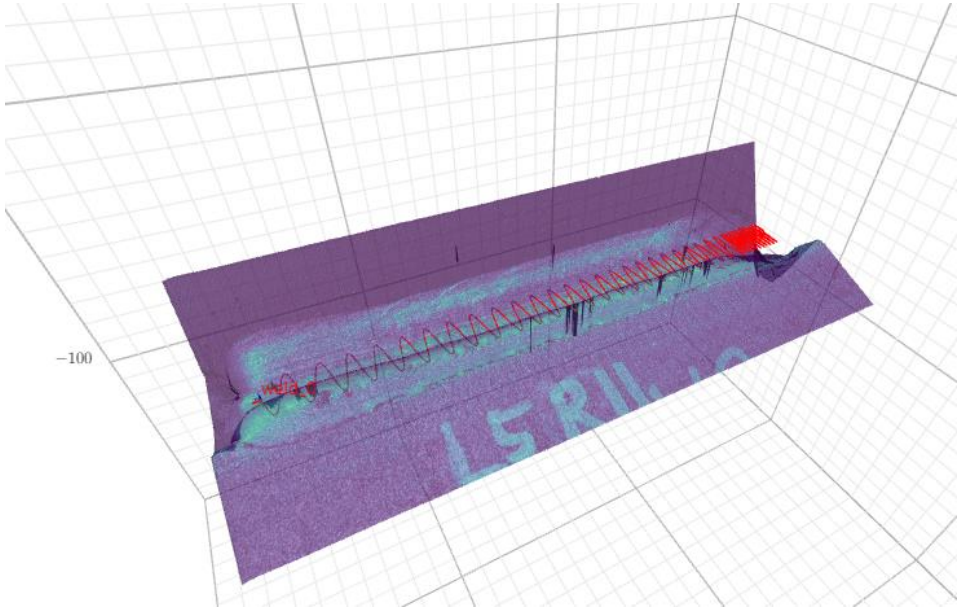
weldx API
measurement chains



- store and describe measurement chains
- signal conversions and storage
- additional metadata
  (e.g. precision, calibration certificates)

weldx API
spatiotemporal transformations

- complex welding traces
- time dependencies
- point and mesh data
- data transformation
- data interpolation



weldx API
adding external data

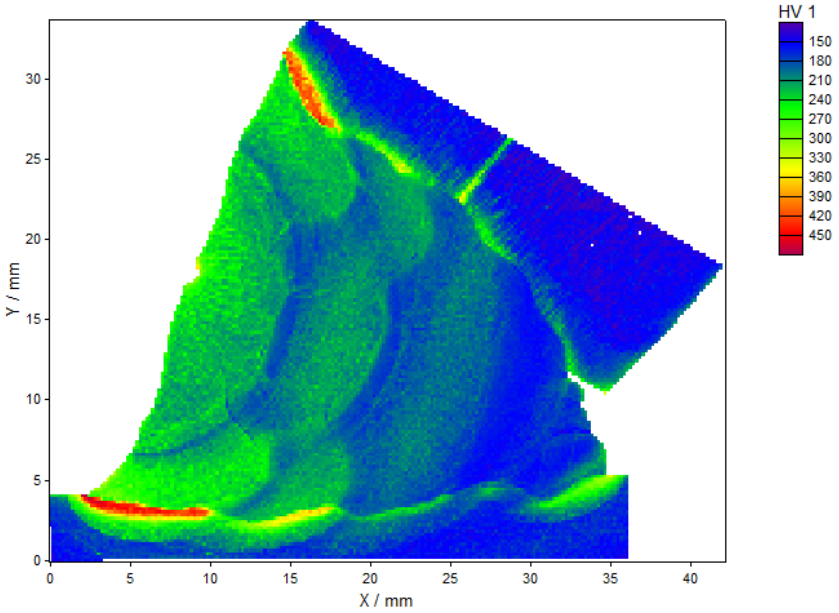
- integrate external data

- example:
  hardness measurement

weldx API
adding external data

- integrate external data

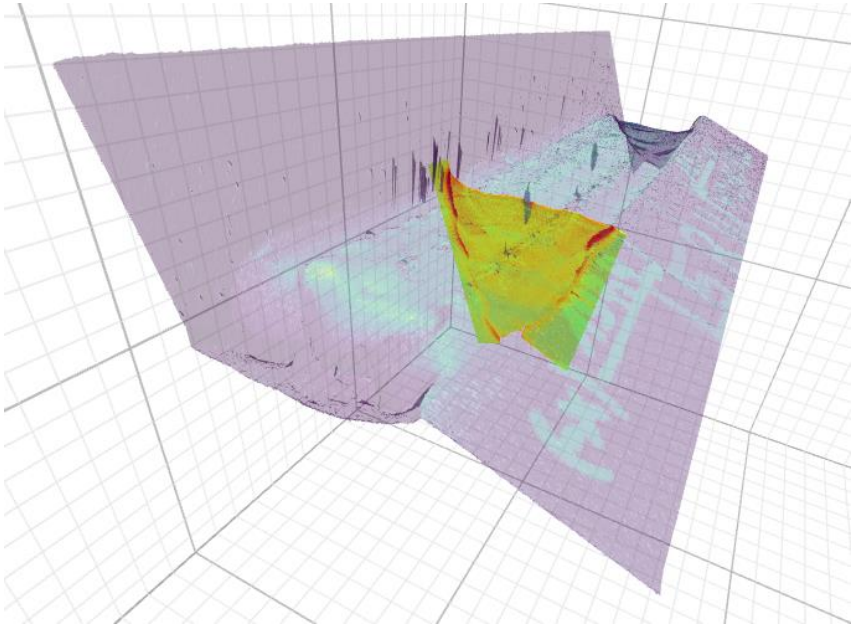
- example:
  hardness measurement



weldx API
adding external data

- add data from external sources
- example:
  hardness measurement
- read data with custom tools
- add and process data with weldx tools

➡ one file for the experiment



weldx quality standards

- weldx standard defines base elements
- user can define their own additional requirements that override the default standard
- weldx can also be extended via ASDF extensions

- Examples:
  - define a strict experimental subset (e.g. only static power source settings)
  - enforce data structures (e.g. specific shape restrictions on data or units)
  - add requirements to specific objects (e.g. every sensor must have a PDF calibration certificate)

acknowledgement

Thank you for your attention

This research was supported by the Federal Ministry of Education and Research of Germany in the framework of WelDX under project number 16QK12

SPONSORED BY THE

Federal Ministry of Education and Research

Thank you for your attention.

welding data exchange
open science initiative

Contact
M.Sc. Cagtay Fabry
cagtay.fabry@bam.de
+49 30 8104-4675

Dr. Andreas Pittner
andreas.pittner@bam.de
+49 30 8104-3696

www.bam.de

Questions

- **What kind of data/information is retrieved from the robot in the lab? Log files?**

- robot positioning data is collected during welding via a real time interface (EtherCAT / TwinCAT)

Questions

- **Do the files need the explicit schema markup, or can it be inferred from the key names?**

- The implementation currently uses only explicit markup.
- Adding (simple) inference should be easy (e.g. DOI)
- Where (digital) standards/onthologies etc. already exist those could be automatically translated or dynamically parsed into weldx

Questions

- **(Not coming from welding background) how WelDX is different than any other format like simple JSON?**

- Some key features compared to “simple” JSON:
  - YAML features (e.g. Anchors, Aliases)
  - binary data support (compression, slicing etc.)
  - tags to bridge into Languages/APIs
  - Schema versioning

welding data exchange format file format – advanced schema example

```
# coordinate transformation
type: object
properties:
  coordinates:
    oneOf:
      - tag: "asdf://weldx.bam.de/weldx/tags/core/variable-0.1.*"
      - tag: "asdf://weldx.bam.de/weldx/tags/core/time_series-0.1.*"
    wx_unit: "m"
  rotation:
    tag: "asdf://weldx.bam.de/weldx/tags/core/variable-0.1.*“
  time:
    tag: "asdf://weldx.bam.de/weldx/tags/time/timedeltaindex-0.1.*"
wx_shape:
  coordinates: [..., (t), 3]
  rotation: [..., (t), 3, 3]
  time: [(t)]
```

Questions

- **It looks like you're using xarray for the Python backend, is that the case?**

- Yes!
- Core numeric python stack:
  - numpy
  - xarray
  - pint + pint-xarray
  - Dask (evaluation)

- others
  - networkx
  - scipy / sympy

Questions

- **Is the usability of weldx files without API/Python an issue? -> Python is not for everybody …**

- Maybe.
- Our current idea is to have the initial (fast) core development in Python.
  Also provide faster initial benefit.
- Implementations in other languages are possible, personally I think writing “well defined” exporters to other formats is probably easier.

Questions

- **Is the weldx format already in use at other institutions?**

- Not yet, we are currently focussing on finishing up the first stable release and publishing more examples, tutorials etc.
- The idea is to start an initial series of workshops (online).