



CPSLint: A DSL ENABLING CPS DATASET VALIDATION AND SANITISATION

URAZ ODYURT
ÖMER SAYILIR
VADIM ZAYTSEV

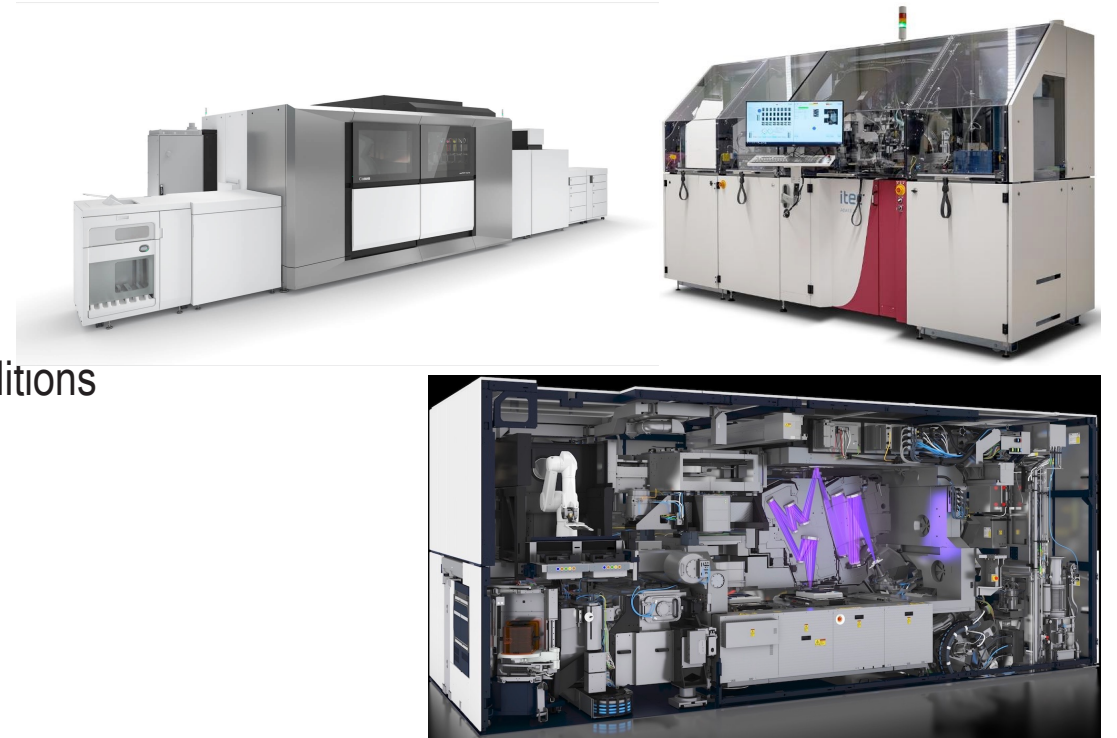
LANGDEV 2025 – 17 OCTOBER 2025



WHAT ARE CPS?

- Industrial Cyber-Physical Systems (CPS)
 - Primarily utilised in high-tech manufacturing industry
 - Often too complex to model (motivation for data-centric)
 - Limited domain of tasks and activities, **purpose-built**
 - Designed for batch processing, **highly repetitive**
 - Operational deviations: Varying input and environmental conditions
 - **Nondeterministic behaviour** (but contained)
 - Full of **sensors**, physical and virtual

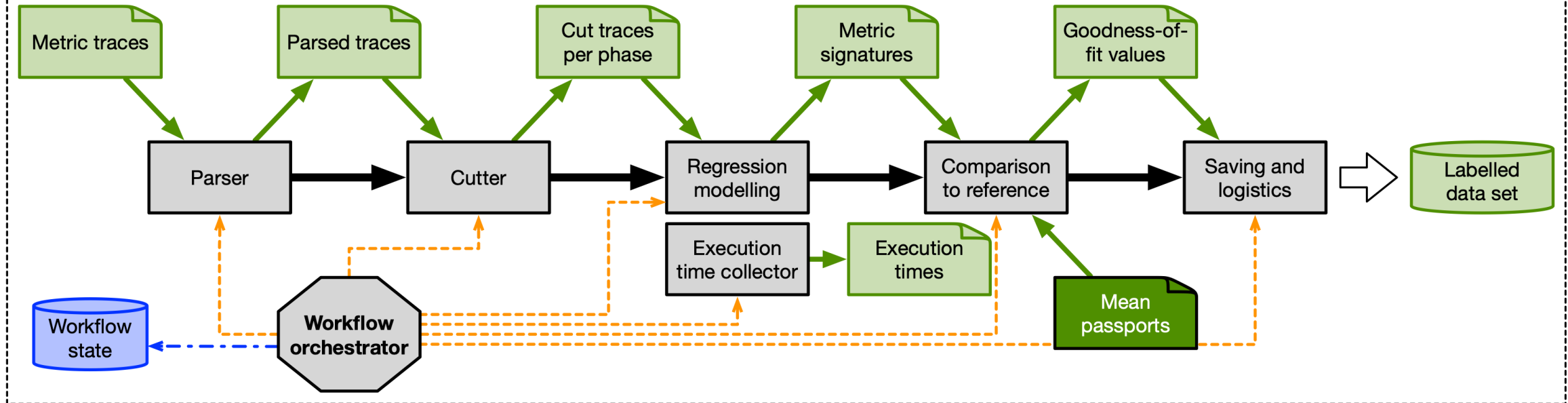
- **Constant** stream of **data** is being logged
 - **Corruptions do occur**
 - Necessitating **sanitisation before use**



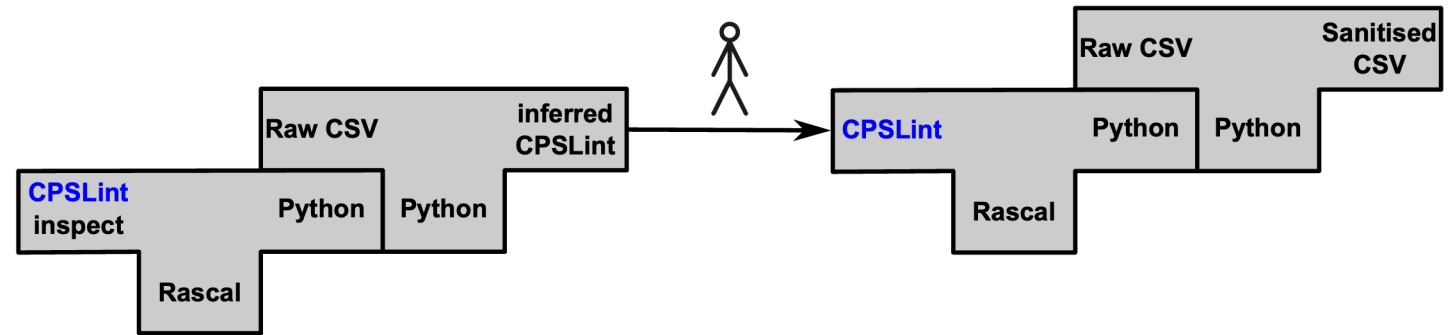
TRADITIONAL DATA PREPROCESSING

Traditional ML: Dataset generation flow

Under anomalous machine behaviour



OUR SOLUTION



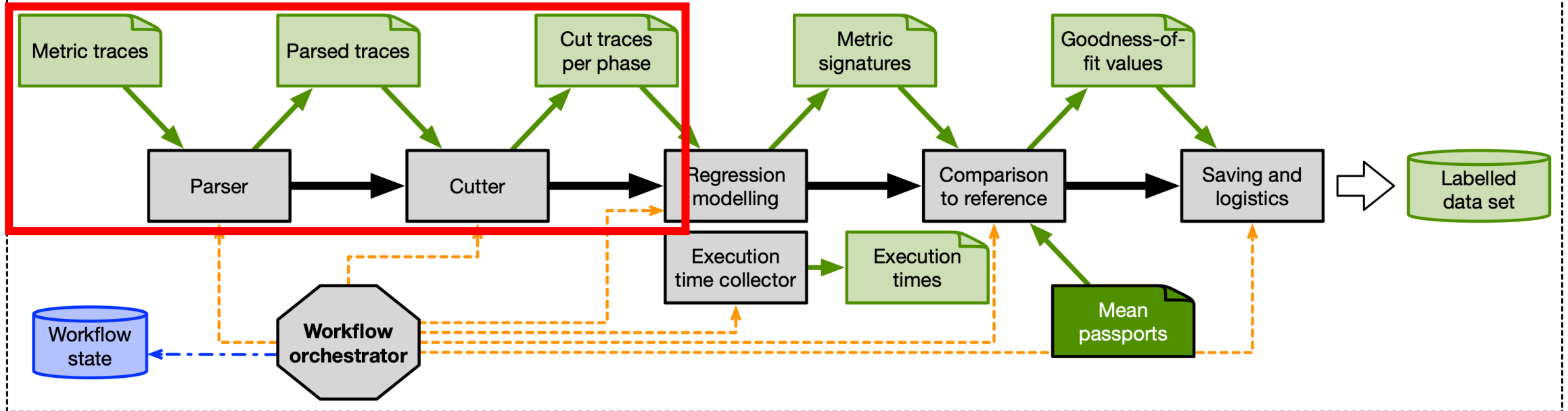
- Rascal as language workbench
- Python as target language
 - Using **common data science libraries** (pandas, numpy, scipy)
- First run:
 - Infer **baseline CPSLint specification** from csv
- **Expert refines specification**
- Second run:
 - Data is **sanitised** based on **specification**
- Compile to annotated **human-readable Python** code
 - **Expert can modify** compiled Python when needed



WHERE DOES OUR WORK FIT?

Traditional ML: Dataset generation flow

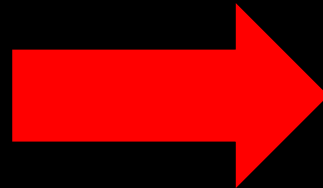
Under anomalous machine behaviour



DEMO

Let's fix some corrupt data!

```
1996 0.49$85,0.60383$3,###,$1.511853,$
1997 0.49@875,###0.6038125,@,1.51@261,abc
1998 0.4abc99,@0.603792,abc5.018,1.513367###,###
1999 0.4992abc5,0.6abc038125,###,1.51###4125,###
2000 0.4995###,0.603@833,###,1.5$14882,abc
2001 0.49$975,0$.6038535,@,1abc.51564,$
2002 0.@5,0.60abc3874,5.017$,1.51639$7,$
2003 0.500@25,0.603874$,###,1.5@17154,$
2004 0.5005###,abc0.603874,$,1.51791@2,abc
2005 0.500abc75,0.60###3874,$,1.abc518669,@
```



```
1996 0.4985,0.603833,,1.511853,
1997 0.49875,0.6038125,,1.51261,
1998 0.499,0.603792,5.018,1.513367,
1999 0.49925,0.6038125,,1.514125,
2000 0.4995,0.603833,,1.514882,
2001 0.49975,0.6038535,,1.51564,
2002 0.5,0.603874,5.017,1.516397,
2003 0.50025,0.603874,,1.517154,
2004 0.5005,0.603874,,1.517912,
2005 0.50075,0.603874,,1.518669,
```

CAPABILITIES OF CPSLINT

Data analysis:

- Column and type inference
- suggest basic operations

Cleaning up corrupt data:

- Enforce column data types
- Restrict out of bounds values
- Reorder out of order rows
- Impute missing fields
- Impute missing rows

Data compartmentalisation:

- Cutting datasets based on flags in the data

To be implemented:

- Data visualisation
- Statistical insights

To be implemented:

- Bespoke imputation strategies

To be implemented:

- Cutting datasets based on peak detection

TO CONCLUDE

- CPSLint helps
 - **Nonprogrammer domain experts** clean their data themselves
 - **Data scientists** save time
- Next steps
 - In-field usage with **data from industrial partners**
 - Mature the prototype further
- Try it yourself!
 - Github (<https://github.com/omersayilir75/CPSSLint> coming soon!)
 - Reach out (o.f.sayilir@utwente.nl, or during the coffee break)

