

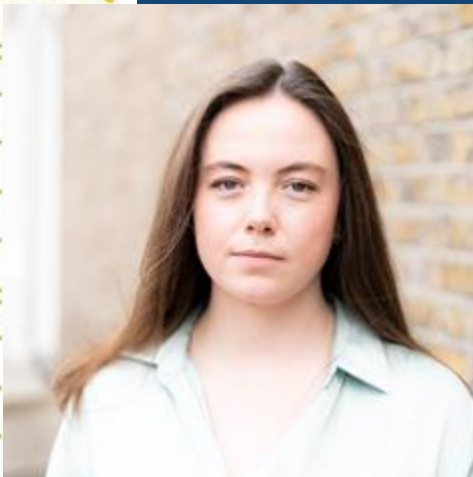


**2023**  
**EUROPE**  
**INTERCHANGE**  
COPENHAGEN | 26-27 APRIL



How we Achieved Real Time Validation by Integrating CORE into our  
Django (Python) Clinical Trial Platform

# Meet the Speakers



## Madeleine De Forest-Brown

**Title:** Software Engineer

**Organization:** Lindus Health

Madeleine De Forest-Brown is a full-stack software engineer with a mechanical engineering background. She was driven to work at Lindus Health as she is passionate about building products that bring the advantages of technology to healthcare.



## Amiel Kollek

**Title:** Senior Software Engineer

**Organization:** Lindus Health

Amiel Kollek, a full stack software engineer, joined Lindus Health after working at Meta. He is enthusiastic about being part of a fast-moving team aimed at enhancing clinical trials and benefiting patients.



# Disclaimer and Disclosures

- *The views and opinions expressed in this presentation are those of the author(s) and do not necessarily reflect the official policy or position of CDISC.*



# Before We Start: CDD

Conference Driven Development



# Conference Driven Development

## Definition: Conference Driven Development (CDD)

A method of software development whereby a talk is submitted for a yet-to-be-completed but exciting project. Acceptance of the talk acts as a deadline for feature completion.



## Agenda

1. Background & How We Learned About CORE
2. Motivations and Goals
3. What We Did (demo)
4. Key Challenges We Faced
5. Next Steps



# Our Background

And how we learned about CORE



# Our Backgrounds

- **Madeleine**
  - Mechanical Engineering
  - EdTech / Asset finance
- **Amiel**
  - eCommerce software
  - (non-clinical) B2B SaaS

No clinical data experience, until...





We're a next gen CRO  
delivering end-to-end clinical  
trials for health and biotech  
pioneers.





# What We Do Now

We are building an integrated clinical trial platform that handles

- ePRO
- eCRF
- Patient Communication
- Clinical workflows
- Monitoring
- SDTM dataset exports

# How We Discovered CORE

CDISC standards

# How We Discovered CORE

CDISC standards



Dataset-JSON hackathon

# How We Discovered CORE

CDISC standards



Dataset-JSON hackathon



CORE



## Goals & Motivations

Make aligning our data with SDTM standards easier

# Goal

**Our goal is to bring validation as close to collection as possible**

# How Do We Align Currently?

- Front load validation by design
- Staff oversight

**This wastes valuable engineer and trial staff resources!** 🤖





**So What Did We Do?**

# What We Did

```
def core_validate(self):  
    from cdisc_rules_engine import rules_engine
```

# What We Did

```
def core_validate(self):  
    from cdisc_rules_engine import rules_engine  
  
    from libs.sdtm_export.sdtm_exporter import SDTMExporterBase  
  
    exporter = SDTMExporterBase(self.participant.study)  
    data = exporter.export_to_csv_for_validation(self)
```

# What We Did

```
def core_validate(self):
    from cdisc_rules_engine import rules_engine

    from libs.sdtm_export.sdtm_exporter import SDTMExporterBase

    import pandas as pd

    exporter = SDTMExporterBase(self.participant.study)
    data = exporter.export_to_csv_for_validation(self)

    rules = load_rules()
    df: pd.DataFrame = run_rules_engine(data, rules)
```

# What We Did

```
def core_validate(self):
    from cdisc_rules_engine import rules_engine

    from libs.sdtm_export.sdtm_exporter import SDTMExporterBase

    import pandas as pd

    exporter = SDTMExporterBase(self.participant.study)
    data = exporter.export_to_csv_for_validation(self)

    rules = load_rules()
    df: pd.DataFrame = run_rules_engine(data, rules)

    return df[
        [
            "core_id", "ran_successfully", "description",
            "was_triggered", "included_domains", "excluded_domains",
            "error_class", "error_message", "subject_ids",
        ]
    ]
```



Demo

YEST Study | TEST003 | Participant Activated

### Adverse Event CORE Validation

**1 RULE FAILED**

Rules Checked: 30 | Rules Passed: 29 | Rules Failed: 1

Rule	Description	Passed
C000-000000	Raise an error when ACDSE is completed and value is not 'Y' or 'N'	Fail
C000-00001E	Raise an error when ACDCCYR exists in AE dataset	Pass
C000-00003H	Raise an error when --RSTNET is present	Pass
C000-00006T	Raise an error when ACDSE is completed and value is not 'Y' or 'N'	Pass

Close



**Did Anything Go Wrong?**





# Key Challenges We Faced

- Focusing on .xpt without needing it
- Confusion around integrating with the CDISC Library api (rules in different formats)
- Projects in their inception have less documentation (struggles of lazy engineers)
- Lack of dirty data
  - Excited to see the power of CORE in the future when we have significantly more SDTM data



## Next Steps

Where we want to take our implementation & how we want to contribute



# Evolution of Our Implementation

**Our goal is to bring validation as close to collection as possible**

- Prevent the creation of invalid records
- Integrate CORE into our CRFs
- In-product summary of all issues
  - Locating problematic records in one click

**Barriers:** Speed, Depth of Integration, User Experience



# How We Want to Contribute

- Help write rules!
- Documentation
- Investigate faster rule checking



# Thank You!

[madeleine@lindushealth.com](mailto:madeleine@lindushealth.com)

[amiel@lindushealth.com](mailto:amiel@lindushealth.com)

[www.lindushealth.com](http://www.lindushealth.com)

