Aalto University
School of Science

International Workshop on Requirements Engineering for Explainable Systems (RE4ES- IEEE RE 2021)

Holistic Explainability Requirements for End-to-end ML in IoT Cloud Systems

My-Linh Nguyen, Thao Phung, Duong-Hai Ly, Hong-Linh Truong
{linh.m.nguyen, thao.phungduc, duong.ly, linh.truong}@aalto.fi
Aalto University
https://rdsea.github.io

Motivating example: ML for Base Transceiver Stations (BTS)

- **ML solution in IoT Cloud systems for predicting behaviors of BTS equipment and infrastructures**
  - Dynamic inferences of near real-time IoT data
- **Challenges:**
  - **Multiple stakeholders, each stakeholder is related to only a part of the ML development.**
  - ***How to do we identify and capture the requirements for explainability***

Multiple relevant stakeholders in predictive maintenance

BTS owner (Telco)

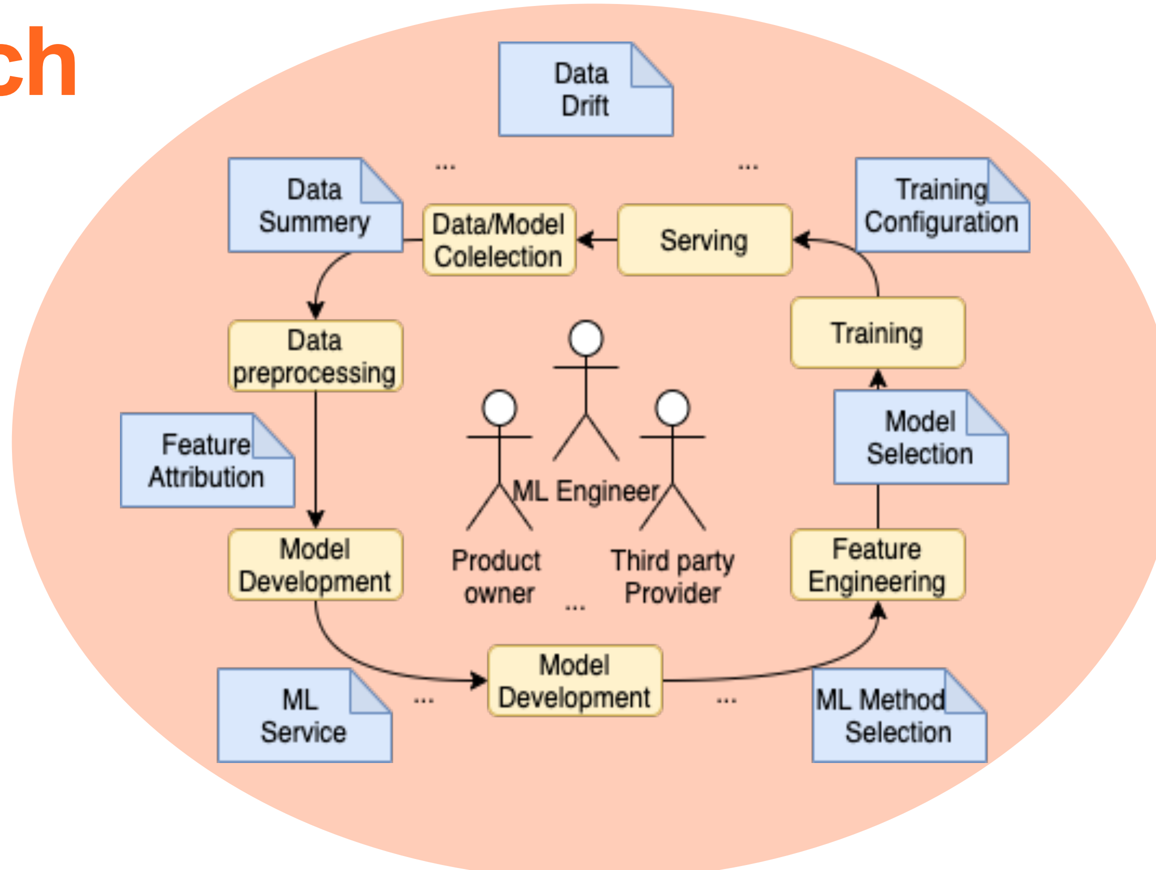
Third-party providers:
Equipment manufacturers, Electricity/power providers, Equipment suppliers, etc.

(Predictive) Maintenance Company

ML development team
(software and telco professionals)

Methodology - Holistic approach

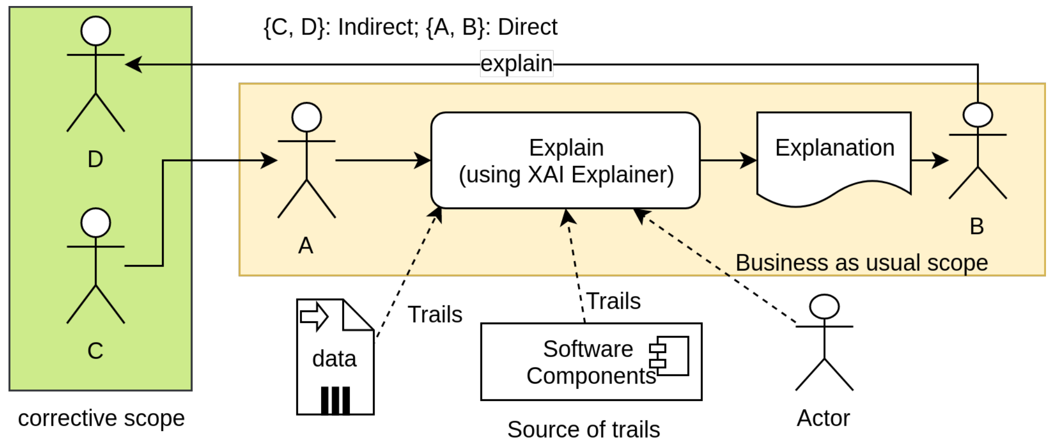
- Explainability requirements covering:
  - *multi-stakeholders*
  - *multiple explainability aspects*
  - *for* an *end-to-end ML system*



Scoping stakeholders

Goal:

- Who is the explaination for?
- What is their relationship?

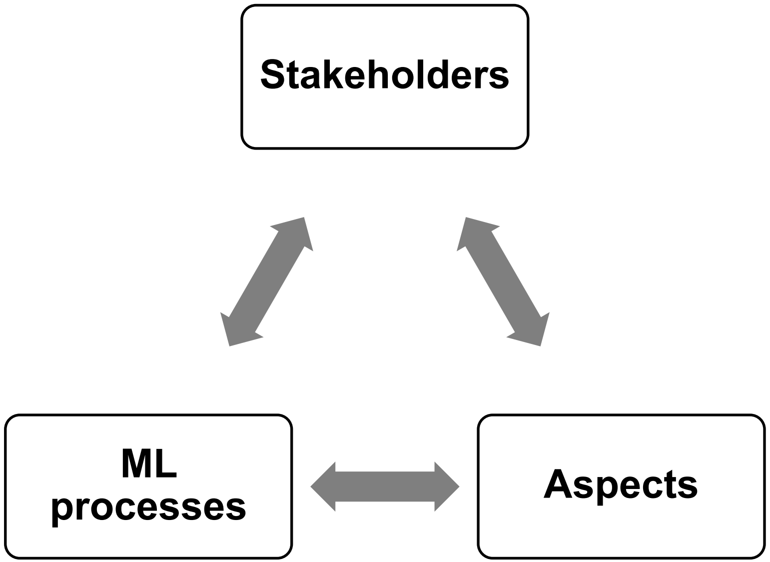


Direct: explanation triggered through "explain" task, e.g., between developers for feature engineering and model training

Indirect: following a chain of direct dependencies, e.g., BTS owner and MLOps team

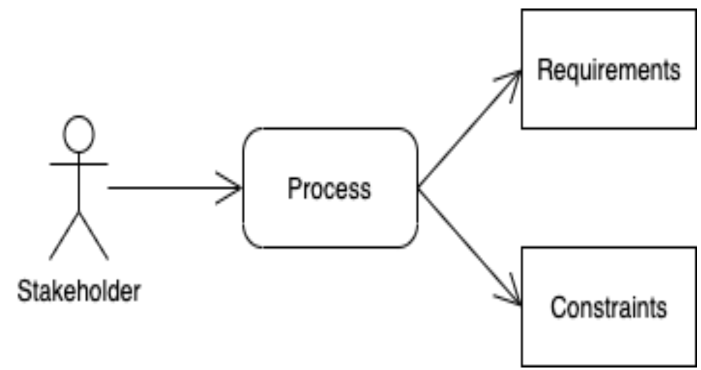
Scoping ML Processes

- Identify processes/ tasks stakeholders responsible for or interested in
- Map stakeholders to relevant phases, covering ML requirement elicitation, service design, and development.
- Explainability for end-to-end ML preserves informative connection

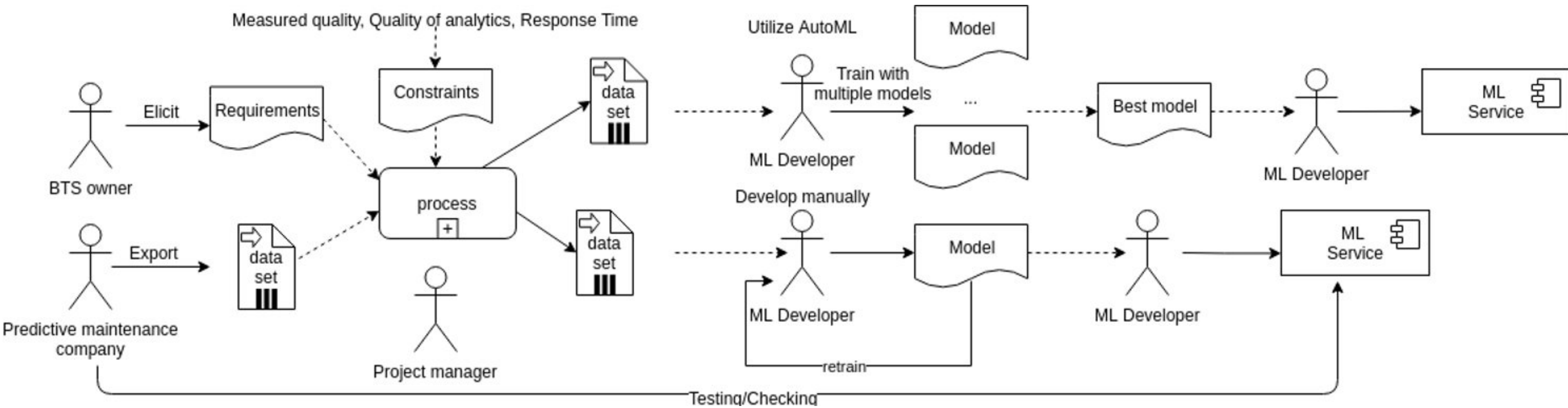


Scoping ML explainability aspects

- Each stakeholder works/ supervises directly specific entities (data, ML models, etc.).
- Overall, a wide range of entities, and associated constraints (metrics and aspects)
- Analyzing their role in each process/task => requirements and constraints



Example in the BTS case



Scope of BTS example

Stakeholders: BTS owner, predictive maintenance software company, technical project managers of the project, ML developers.

Explainability requirements in ML processes/tasks

Data & Model Collection:

- Who collects or provides data?
    - E.g.: BTS data is provided by BTS monitoring system/company.

Data Preprocessing:

- Who did the pre-processing?
- Why did the stakeholder choose to perform specific techniques?

Model Development:

- Who decides machine learning methods?

Feature Engineering (FE):

- Why certain feature engineering techniques (e.g.: feature split, scaling) are used?
    - In BTS, we may use some FE techniques: feature split, grouping operations, scaling.

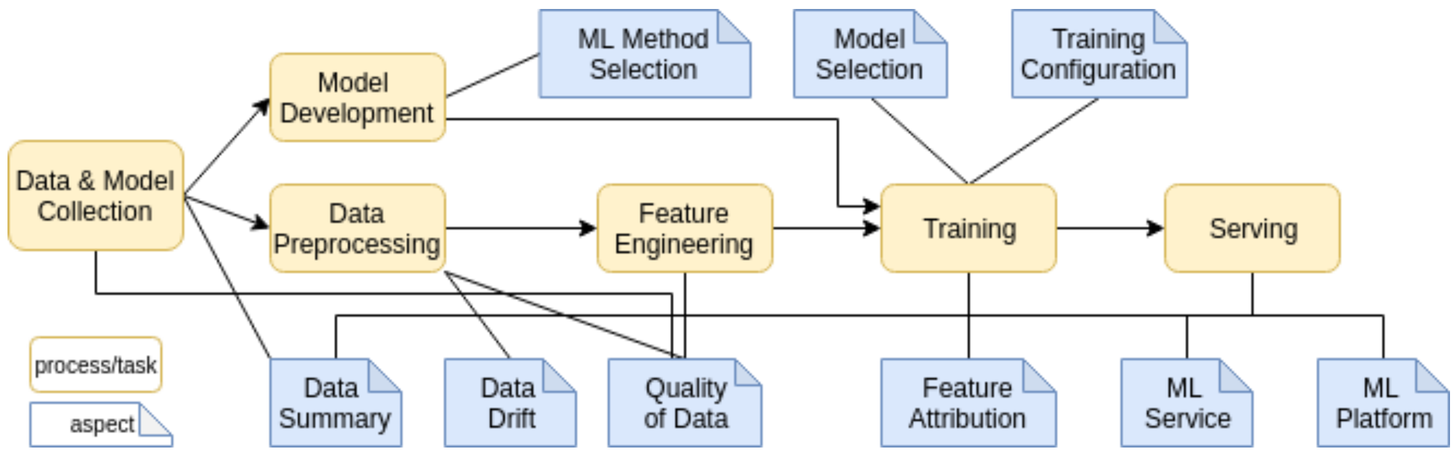
Training:

- Who did the training? (manually or by AutoML tools).

Serving:

- Which service provider is used?
- Where is the solution deployed?

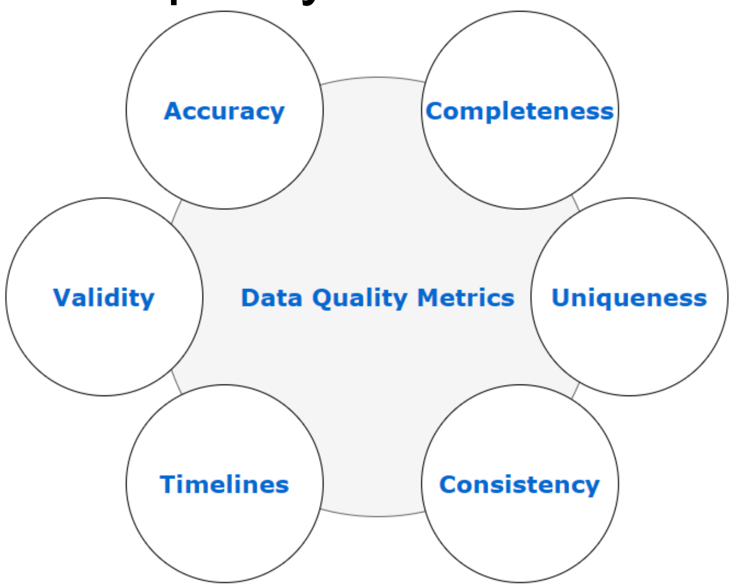
Requirements in explainability aspects



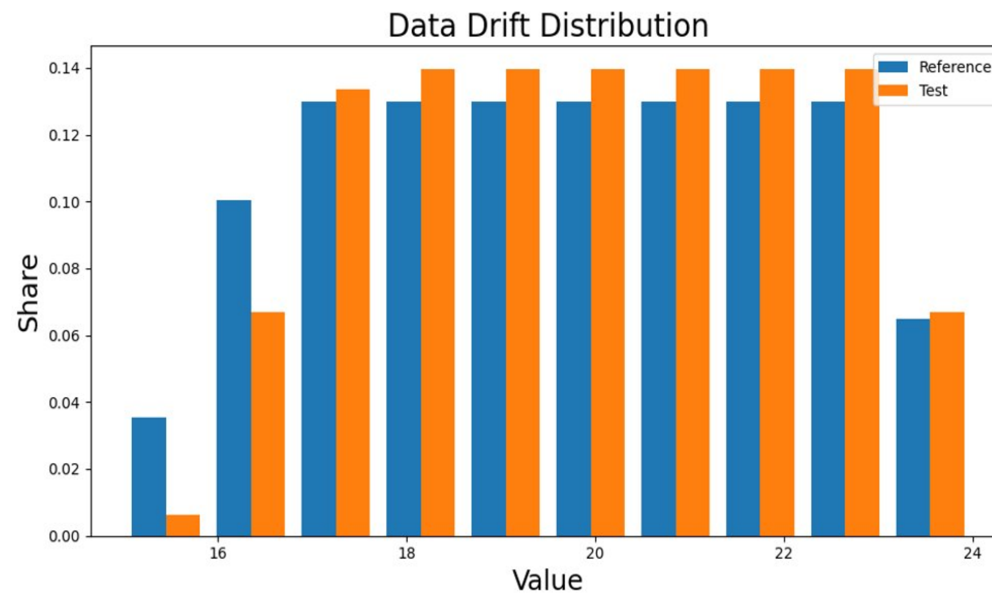
Data-related aspects are crucial in ML solutions in IoT Cloud Systems

Examples: capture data related requirements

IoT Data – important data quality metrics



Data Drift Impact

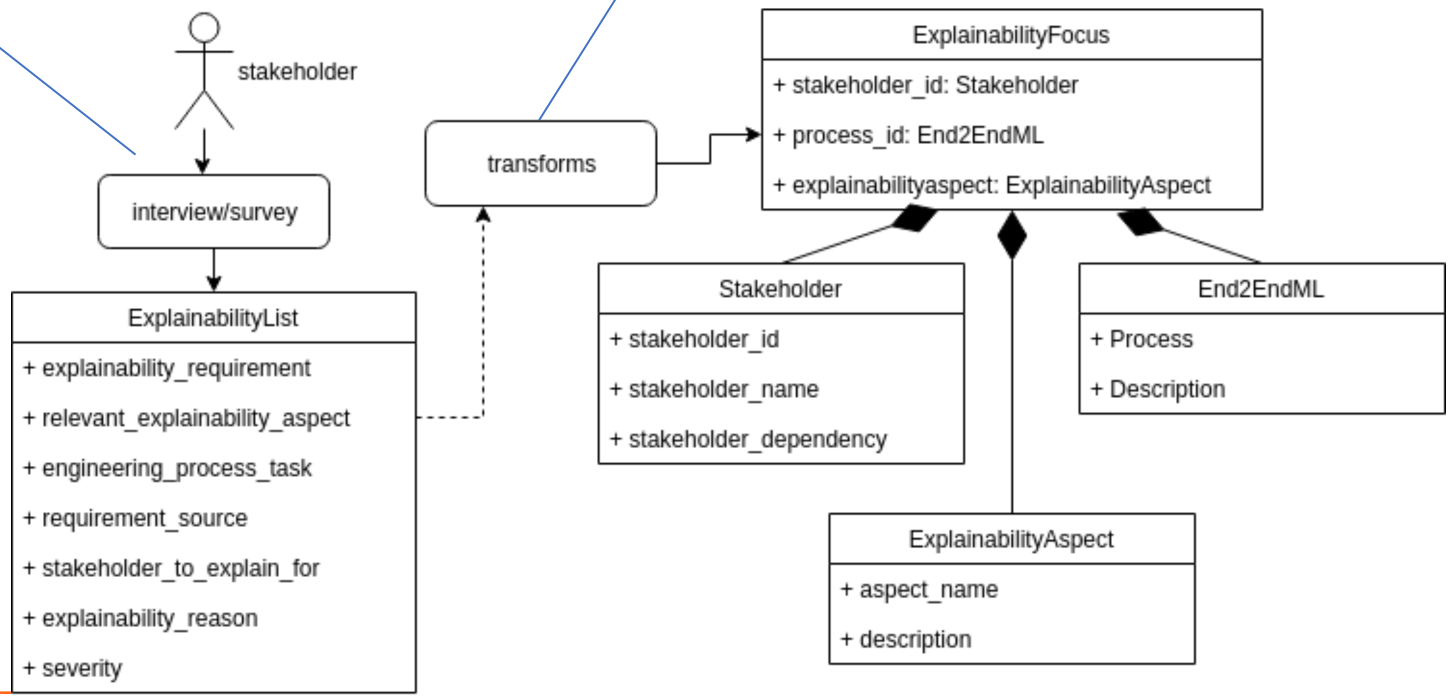


An example of comparison between BTS data distribution regarding feature sensor reading temperature of air-conditioner in one station within one hour tested using Kolmogorov Smirnov (KS) Test.

Concrete explainability requirements elicitation

Transform surveyed data and ingest the data into data services

Interview concrete stakeholders



Continuous requirements update

BTS
Explainability Requirement Example

	Explainability Requirement	Explainability aspect	ML Processes/Tasks	Requirement sources	Target Stakeholders	Explainability Story/Reason	Severity
1	Dataset completeness quality must be at least 0.8	Quality of Data	Data and Model Collection	ML Developer	Predictive Maintenance Company	Dirty data cause model to perform poorly, and introduce bias to the model.	HIGH
2	Record of training configurations (model types, parameters, architectures, etc)	Model	Training	BTS owner, Predictive Maintenance Company	ML Developer	Explain outcome by inspecting model types (pre-trained, fine-tuned or self train) and different combination of input parameters.	HIGH
3	Priorities in using interpretable models over black box model	ML Method Selection	Model Development	BTS owner	ML Developer	Interpretable model helps produce explanations that are more faithful to what the model actually computes	MEDIUM
4	Record of feature importance values	Feature Attribution	Feature Engineering	BTS owner, Predictive Maintenance Company	ML Developer	Knowing feature importance could help increase the model accuracy, and reduce the computation resource	MEDIUM
5	Response time must be < 2000ms	Service quality	Serving	BTS owner, Cloud service that host data and model	ML Developer	To explain the delay in service, and the choice of model affecting response time	LOW

Example of requirements based on our proposed methods

How to utilize our result?

- Integrate explainability requirements into cloud-native DevOps ML in IoT cloud data

- Provide input for identifying and managing diverse types of trails for explanation tools
  – *Experiment data, metadata about data, metadata about models etc*

- Identify, recommend and configure explainer tools
  - Combine suitable explainer for a particular case

Conclusion and future works

Conclusion

- We identify and classify explainability requirements engineering through:
  - Involvement of relevant stakeholders
  - End-to-end data, model, and service engineering processes
  - Multiple explainability aspects.

Future work

- Tools and services for collecting different type data
- Composition of explainer for data drift in end-to-end

Aalto University
School of Science

[Paper]

[AaltoSEA research group]

Thank You!

Questions?