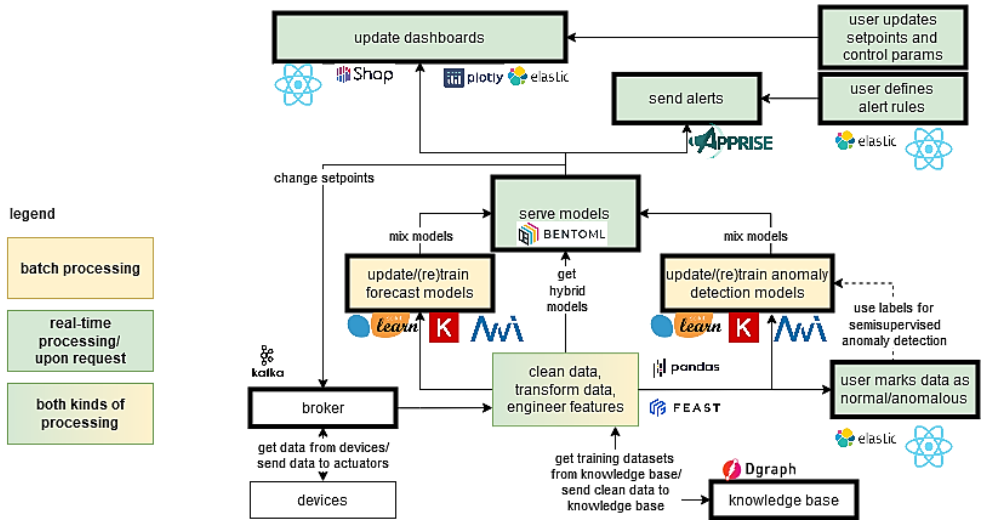
Technical Specification Double-side Page

1. **TECHNICAL SCOPE:** Summarize the mock-up devised during the EXPLORE phase: how have you addressed the challenge/Theme Challenges and tackled with its requirements and data. Include a diagram.

The mockup is a web app. After login authorized users can

1. see a 2D view of a monitored line with text boxes of real-time data; examine charts/stats of time series from that line; change setpoints/control inputs. We focus on energy-related variables matching Idea75's
2. create their own 2D view of a line by uploading a picture, creating text boxes of real-time data (with alerts) and dragging them over the picture
3. plot anomalies per asset, understand anomalies through interpretability plots and request semisupervised anomaly detection models for that asset by flagging normal/anomalous samples and assigning tags ('power surge', 'overheating', …) to anomalous samples
4. plot forecasts per asset and evaluate the difference between real data and forecasts
5. request forecast and unsupervised anomaly detection models to be trained on selected assets and datasets
6. optimize processes based on forecast models, objective functions (e.g., minimum energy consumption) and bounds on features; schedule jobs that can fine-tune setpoints in an automated way
7. manage groups. Each group can access different dashboards and models.

Only selected groups can exploit features 2, 3, 5, 6. The mockup covers blocks with a thicker line in the sketch below.



2. **ALGORITHMS, TOOLS AND CONCLUSIONS:** Detail the algorithms and tools identified to accomplish the challenge/Theme Challenges. Show clear understanding of the used REACH dataset/s and addressed challenge/Theme Challenges.

Idea75's dataset includes 4 features sampled every 60 s. As these multivariable time series change a lot with time, show some periodicity and are related to one another, we plan to perform tasks such as anomaly detection, forecast and control optimization using spot-on deterministic (e.g., changepoint detection, thresholds from standards) and ML (machine learning) models. To ensure accuracy and robustness we plan to mix deterministic and ML models: both regression/forecast and anomaly detection models will be hybrid. Energy consumption will be forecast by mixing time-dependent regression models (e.g., random forest, boosted trees, long short-term memory) to capture complex nonlinear process/asset dynamics. In this way we will solve data-driven optimization problems (e.g., minimum energy consumption in a time horizon) with bounds and constraints (e.g., mill yield within bounds) to find the best control inputs. We plan to mix ML models as per doi.org/10.1609/aaai.v34i09.7111; e.g., AOM (average of maximum): models output anomaly scores for each sample; first, scores are split into subsets; then, the highest score is taken as the subset score; finally, all subset scores are averaged. When the EXPERIMENT phase starts we plan to build 1 forecast model and 1 anomaly detection model from small data batches (from few periods of interest): these tasks need more time. Semisupervised anomaly detection will exploit deep neural classifiers with confidence scores as sample weights. Feature engineering (creation of ML features) will also need care. It will not be just data driven.
Concerning libraries, we plan to use Kafka for data streams; Pandas for data analysis; scikit-learn and Keras for ML; NNI for hyperparameter tuning; SHAP for XAI (explainable AI); React, Dash, Elastic UI for the web app.
All libraries are open-source. We will save IoT data/models to Dgraph, an open source GraphQL database.
We will define DTs (digital twins) via FIWARE's Smart Data Models or Azure DTDL (DT language). Both ontologies are based on JSON-LD. All models will be dockerized.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951981

3. **SCALABILITY AND FLEXIBILITY OF THE SOLUTION:** Discuss whether the solution can truly cope with humongous and increasing datasets and how flexible it is to adapt to other related domains

We plan to orchestrate services using Kubernetes.
MLOps will help cope with increasing datasets as it automates ML tasks. MLOps open-source tools will be used to cut ML maintenance cost; e.g, DVC (versioning of models/experiments); CML for CI/CD (retraining, testing, monitoring); Feast (feature store).
We plan to train ML models at scale using big open-source datasets (e.g., CREAM) and synthetic data. We will bring data from a cabling factory (23e6 anonymized samples of current, active energy, reactive energy, power factor). We are assessing whether a bearing factory EURIX is monitoring could provide energy-related big data. We aim to weigh up transfer learning by the end of the EXPERIMENT phase.
We will try distributed training on Spark. Our team is skilled in it. Spark-related data analysis tools such as Pandas API on Spark and ML libraries such as sk-dist and TensorFlowOnSpark will be assessed in the EXPERIMENT phase.
We plan to serve ML models at scale using BentoML: it produces API server docker images that are easy to scale with Kubernetes.
Dgraph ships with NoSQL-like scalability and provides SQL-like transactions at the same time.
We would like to use REACH's Big Data Infrastructure once we know how many extra resources we need.

4. **DATA GOVERNANCE AND LEGAL COMPLIANCE:** Describe the security level of the proposed solution, i.e. how authentication, authorization policies, encryption or other approaches are used to keep data secure. Explain how will be compliant with the current data legislations concerning security and privacy (e.g. GDPR).

Privacy: Tenants will be isolated and separated using API gateway LoopBack. API gateways can route all user requests to the backend APIs after validation. API gateways log all activity, gather API usage statistics and expose less technical, high-level APIs. API gateways allow to decouple the development of backend and frontend.
Users will be authenticated using JWT (JSON Web Token) checks.
We look forward to using FEDEHR Anonymizer as a privacy-enhancing tool from REACH Incubator. This tool could also help with synthetic data. We plan to aggregate data related to critical assets/processes whenever doable. We plan to set a minimum retention time; i.e., only retain data for the time needed for MITEE to work properly.
Security: We plan to secure communication on public endpoints using TLS protocol. Let's Encrypt will provide TLS certificates. In-transit data will be encrypted using TLS. A Docker registry will manage images securely. Docker Scan will report any vulnerabilities. We plan to save IoT data and asset/process models to Dgraph, which provides encryption at rest and encrypted backups.
EURIX's datacenter complies with ISO 27001.
Legal compliance: The GDPR states users can object to and opt out of automated decision-making. The GDPR also states the right to an explanation of such decisions. Users will be able to opt out of ML models. We plan to ensure the interpretability of ML models via 1) tables listing model features and type; 2) SHAP plots.

5. **QUALITY ASSURANCE AND RISK MANAGEMENT:** Describe the quality process planed for the final product. Technologically, which are the potential risks in all the phases of the project (design of the solution, development, testing, deployment…) and indicate mitigation plans to still fulfil the challenge/Theme Challenges and data provider requirements.

Quality assurance: EURIX has been ISO 9001 certified since 2006. We plan to deliver software in an incremental way according to the Agile methodology; i.e., short, time-boxed release cycles with incremental deployments based on high-priority requirements. Some of us are Scrum certified. We plan to coax Data Providers into providing continuous feedback. We plan to monitor some KPIs throughout the next phases: 1) good UI/UX KPI values for the web app; 2) high data quality; 3) industry-level accuracy metrics for forecast and anomaly detection models; 4) lower energy consumption of tunable processes/assets by over 15% by the end of the next phases; 5) low latency. As stated above, MLOps tools will ensure accuracy stays high during model serving.
Risk management: Across all phases, the main risks (level = medium) are 1) small training datasets before/after data cleaning; 2) unreliable data, which describes processes/assets inaccurately. Risk 1 can be reduced by picking models one can train on less data; upsampling; creating synthetic data; transferring learning from related domains. Risk 2 can be reduced by appointing data quality controllers; getting data from similar industrial processes; scaling goals down. Low UI/UX KPI values pose a noteworthy risk (level = low-medium) to design, development and testing. This risk can be reduced by agreeing with project stakeholders on usability/effectiveness and by adopting Design Thinking.
Low-level risks are listed below:

1. Inaccurate ML models; model drift. Mitigation: retrain the models after tweaking data preprocessing (cleaning, transformation, ...) and feature engineering; change models
2. Bad scalability of training with dataset size. Mitigation: improve communication between nodes; e.g., try synchronous/asynchronous updates
3. Few labels marking data as normal/anomalous. Mitigation: prove the relevance of semisupervised anomaly detection to project stakeholders
4. Costly maintenance of MITEE. Mitigation: scale goals down; rely more on MLOps

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951981