# UnRısk

UnRısk ALGORITHM SERVICE

Automated Execution of Algorithms Connected to Data

What is UnRɪsk ALGORITHM SERVICE?

UnRɪsk ALGORITHM SERVICE is a scalable microservice infrastructure, designed to seamlessly connect your data with in-house developed algorithms. It offers continuous and automatic import, identification, and assignment of incoming data, enabling the interactive linking and processing of diverse data types, including industrial IoT and financial data, through your algorithms. The service supports a wide range of technologies, such as native executables, scripts, Java, Scala, Python, Mathematica notebooks and MatLab Runtime. Moreover, it facilitates the dynamic linking of incoming data to specific algorithms based on your data's meta information tags.

Key Features

- OS and programming language agnostic algorithm support (executables, scripts, versions, configurations, etc.).
- Collection of data in any format via REST endpoint or Apache KAFKA.
- Smart meta information parsing of input data (JSON).
- Highly configurable data to algorithm routing.
- Smart retrieval of previous results as additional algorithm input.
- Processing options of input data (chronological, timeout, maximum instances).
- Full featured HATEOAS-compatible REST interface (including Swagger UI documentation).
- Web-based algorithm management interface.
- Graphical filtering/ordering of inputs and algorithm results.

- Email notifications on warnings/errors during algorithm execution.
- Simple deployment (Java Spring Boot Web as single jar).
- Support of multiple relational databases via JDBC (Postgres, MariaDB, Oracle, H2, Derby, MSSSQL, etc.).
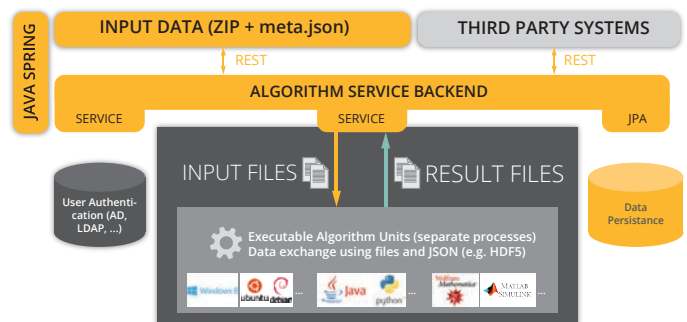
Key Benefits

- ✓ Save time with the automated import, including identification and classification of the algorithms. Focus on developing your algorithms instead.
- ✓ Version and configuration management for your algorithms.
- ✓ Customizable and secure web interface with advanced upload, download, management, and search capabilities.

- ✓ Interactive resend/recalculation features reapplying new algorithm versions to historic data.
- ✓ On-the-fly algorithm invocation via REST module execution endpoints.
- ✓ High scalability through containerized hosting (e.g. Docker, Kubernetes, etc.).