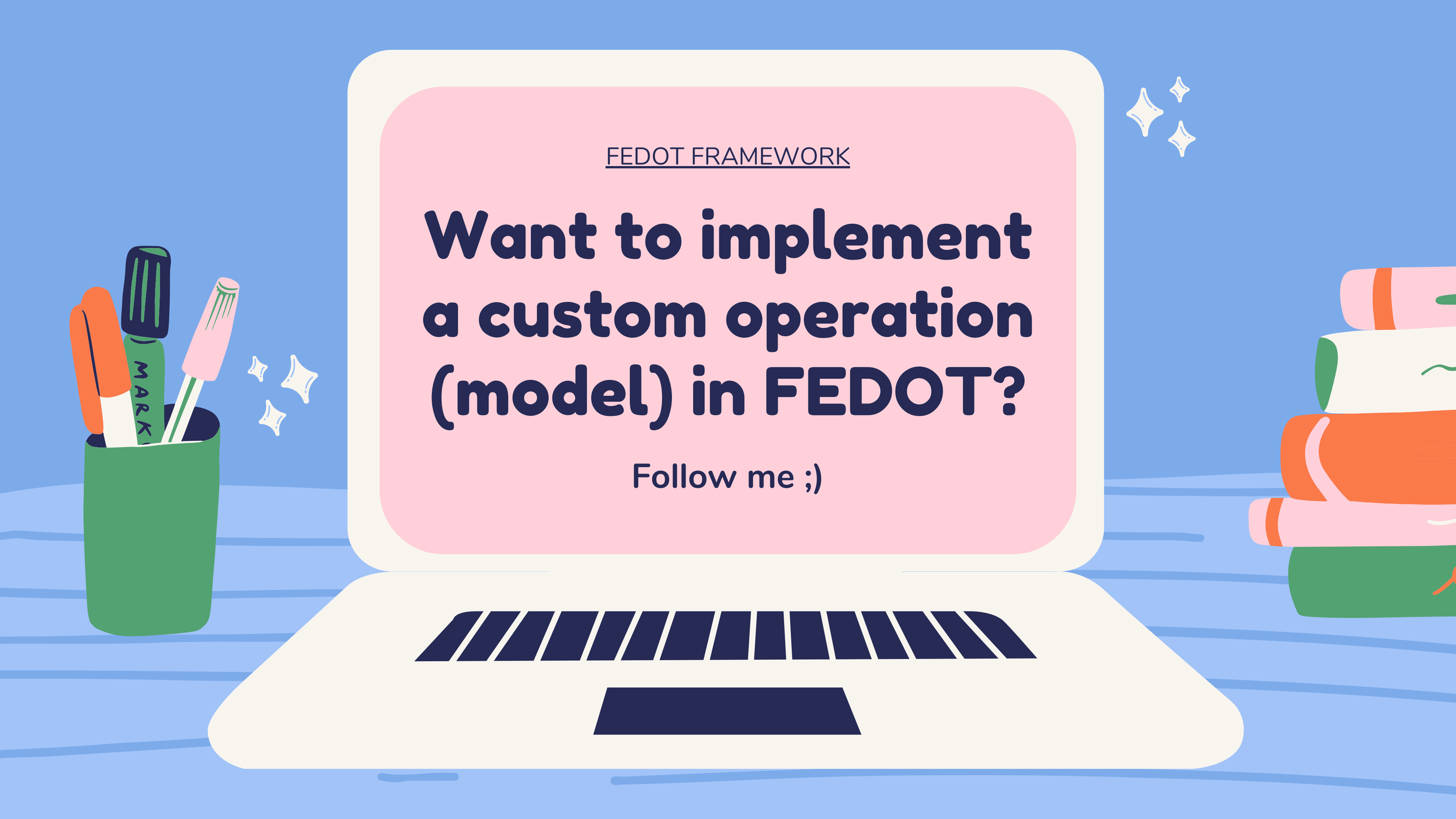

FEDOT FRAMEWORK
Want to implement a custom operation (model) in FEDOT?
Follow me ;)

FIRST, LET'S REMEMBER

All the contributions to the master branch can be made only via pull request

Therefore,
Create a new branch from the master branch and start working!

FEDOT user abstractions

PIPELINE

This is a set of several models or preprocessing methods that are contained in a single structure

NODE

A container in which the operation is placed. A single node can contain only one operation

OPERATION

A machine learning model or preprocessing operation or statistical models

Pipeline consists of nodes (one or more)

Nodes contains operations

The model you call e.g. from sklearn

FEDOT developer abstractions

FEDOT consists of several layers

Pipeline

Pipeline class

Node

PrimaryNode and SecondaryNode classes

Operation

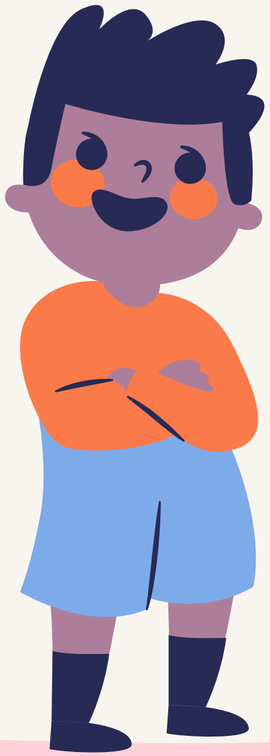
Model and Opertion classes

EvaluationStrategy

There are many of them. This allows you to use models from different libraries. Also, all strategies are divided into types of tasks to be solved, such as classification and regression

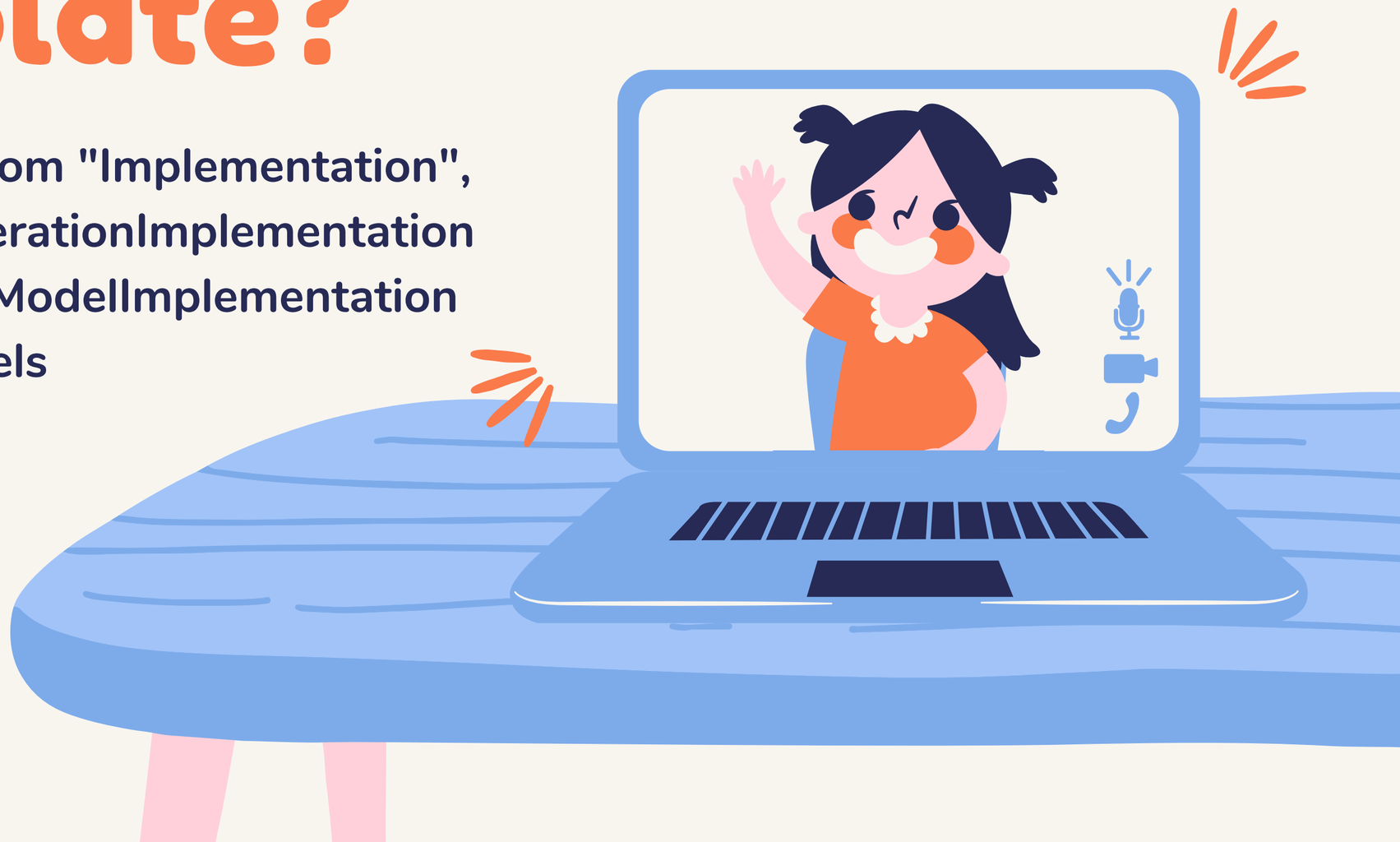
Implementation

"Lowest Asbstraction". These are our models and preprocessing realisations. If you want to implement your own, you're right here.



What I can use as a template?

Answer: To implement a custom "Implementation", use an abstract class DataOperationImplementation for data preprocessing and a ModelImplementation class to create your own models



Step 0

Implementing your custom model or data processing operation

What to do when the custom model has been implemented

That's great, you've done so much already!

Step 1

Choose an appropriate strategy to which your operation should correspond to

It could be, for example:
CustomClassificationStrategy
CustomClassificationPreprocessingStrategy
CustomRegressionPreprocessingStrategy
CustomRegressionStrategy
CustomTsForecastingStrategy
CustomTsTransformingStrategy

Step 2

The operation must be included in the repository

You need to think of a short name for the opertaion and put it into a json file with the repository

If you want your operation to tune well, don't forget to take care of the hyperparameters

Step 3

Write the default hyperparameters in the json file

Step 4

Enter hyperparameter intervals in the get_operation_parameter_range function
This is necessary so that the tuner can tune your operation

CONGRATULATIONS!

Your custom operation can now be used in FEDOT

Step 5

Don't forget to write tests for the new functionality!

If you have any questions, feel free to contact us
This presentation was prepared by Natural Systems Simulation Team