



ML Abstraciton for Keras to Serve Several Cases

Hendri Karisma



Actual Topic : Micro Kernel Architecture Implementation for Machine Learning

Hendri Karisma



Hendri Karisma

- Principal R&D and Data Lead at [Akar Inti Data](#)
- Working for advance data analytics platform, big data, geographic performance
- Working with Software Engineering team for Data marketplace (<https://nusadata.ai>)
- Data solution architect

Indonesia's First Multi-Industry Data Marketplace

Connecting data across many major industries in the private and public sectors through trusted world-class 21st century secure and private environments.

[WATCH DEMO](#)

Welcome to Nusa Data

Create your Nusadata Account

[SIGN UP](#)

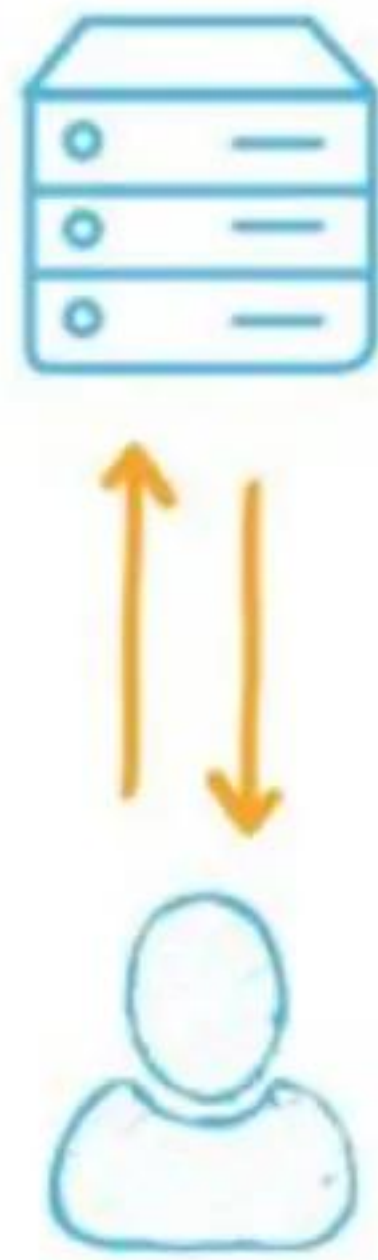
Already have an account? [Sign In](#)

[Our Partnership](#)

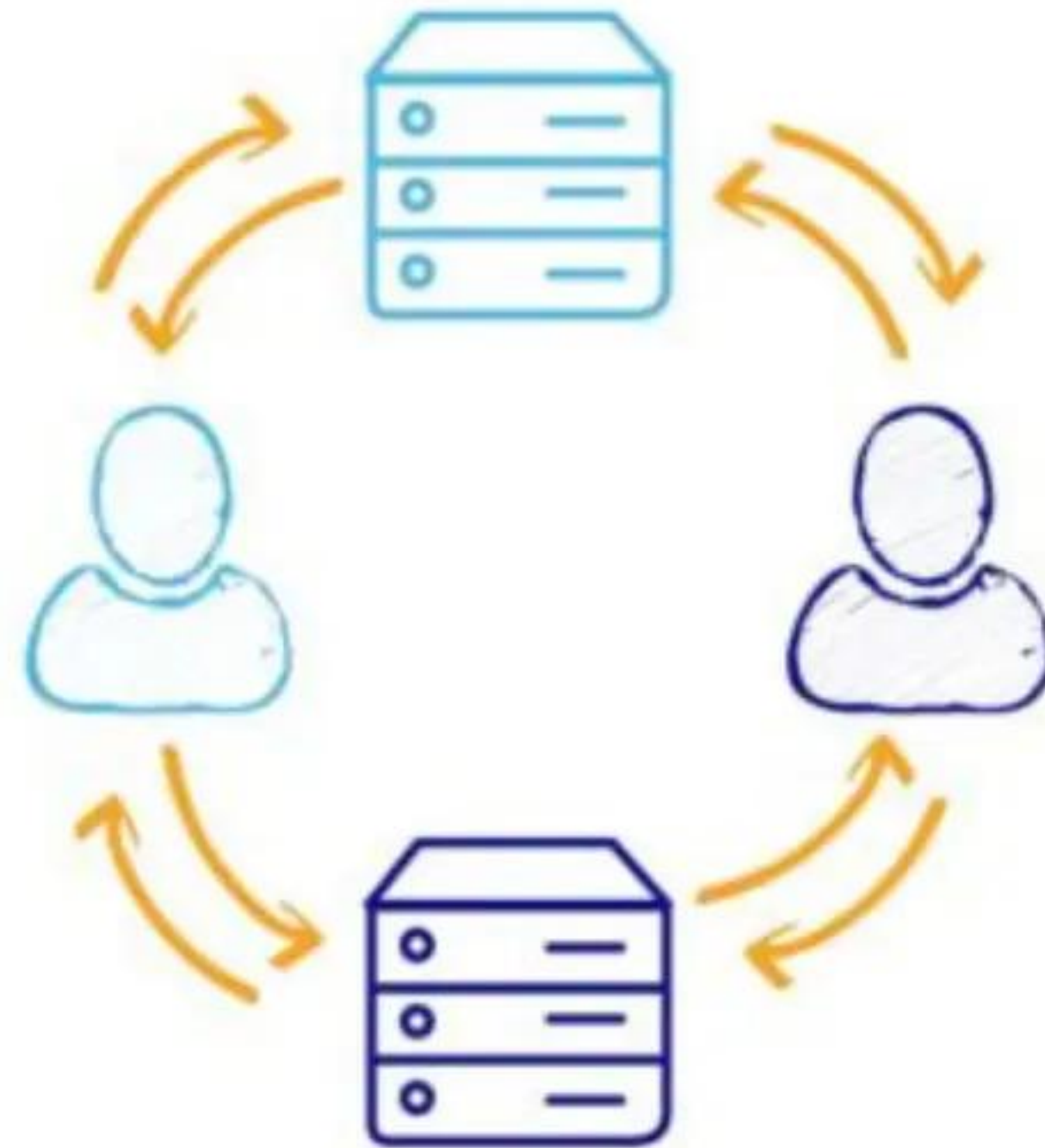
Trusted by



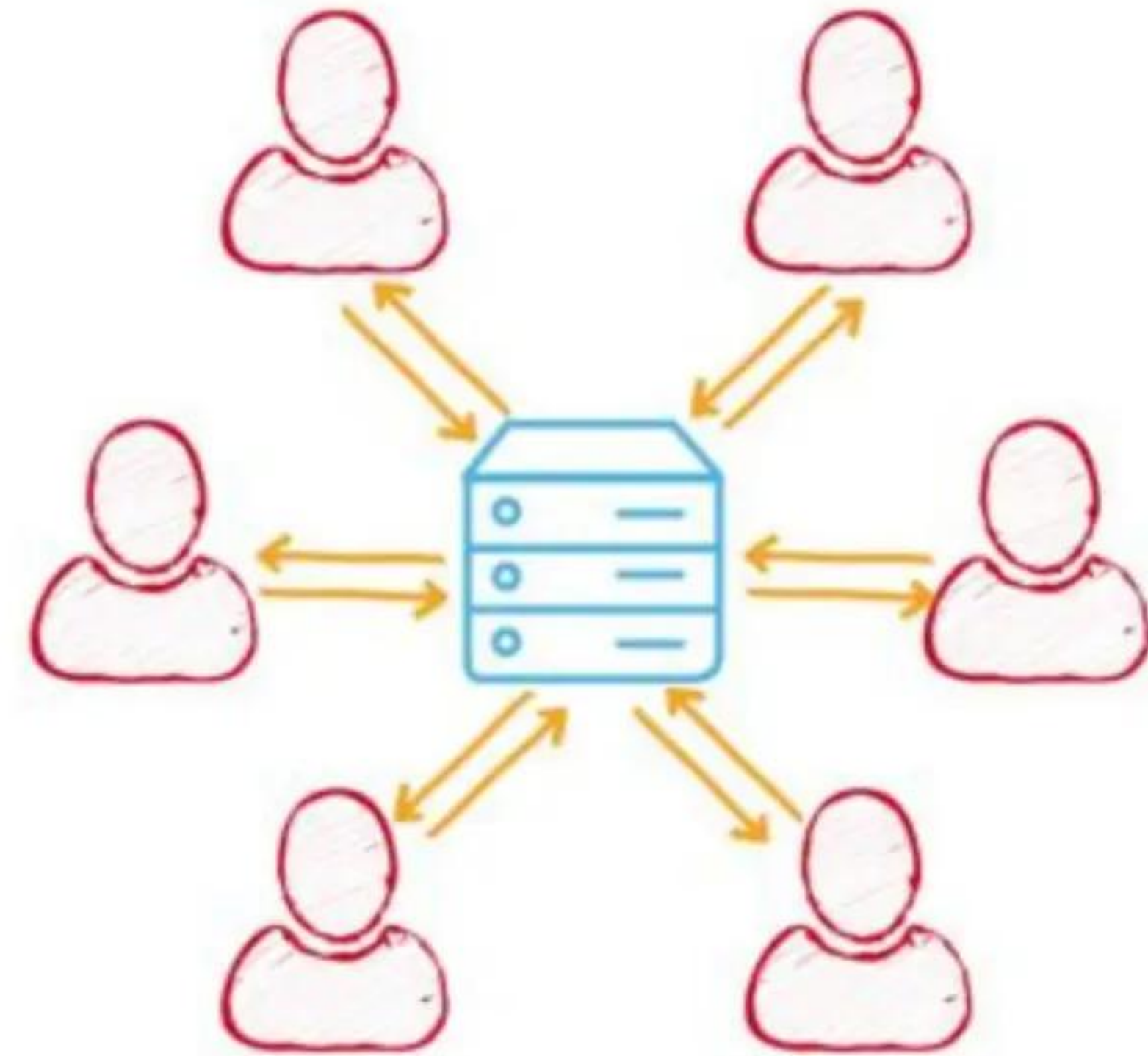
What is 3rd Party Data



first-party data



second-party data



third-party data

Data Marketplace





Why we need to abstract the solution?

- We facing different cases
 - Different solutions
 - Different approachments
 - Different methods
 - Different Technologies
-
- Different deadline schedule...
 - So we need to make the components reusable



Computational Thinking

DECOMPOSITION



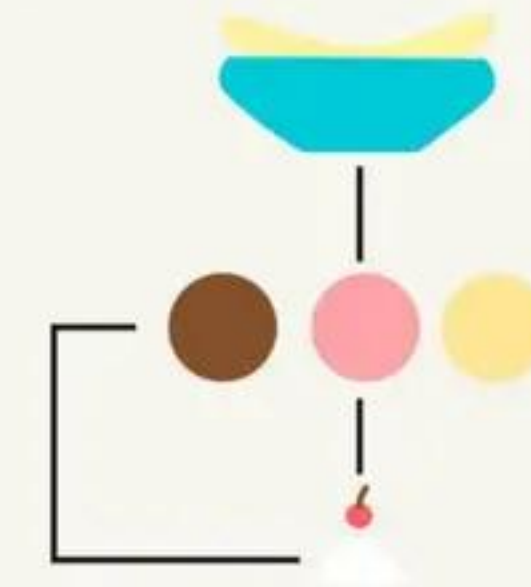
PATTERN RECOGNITION



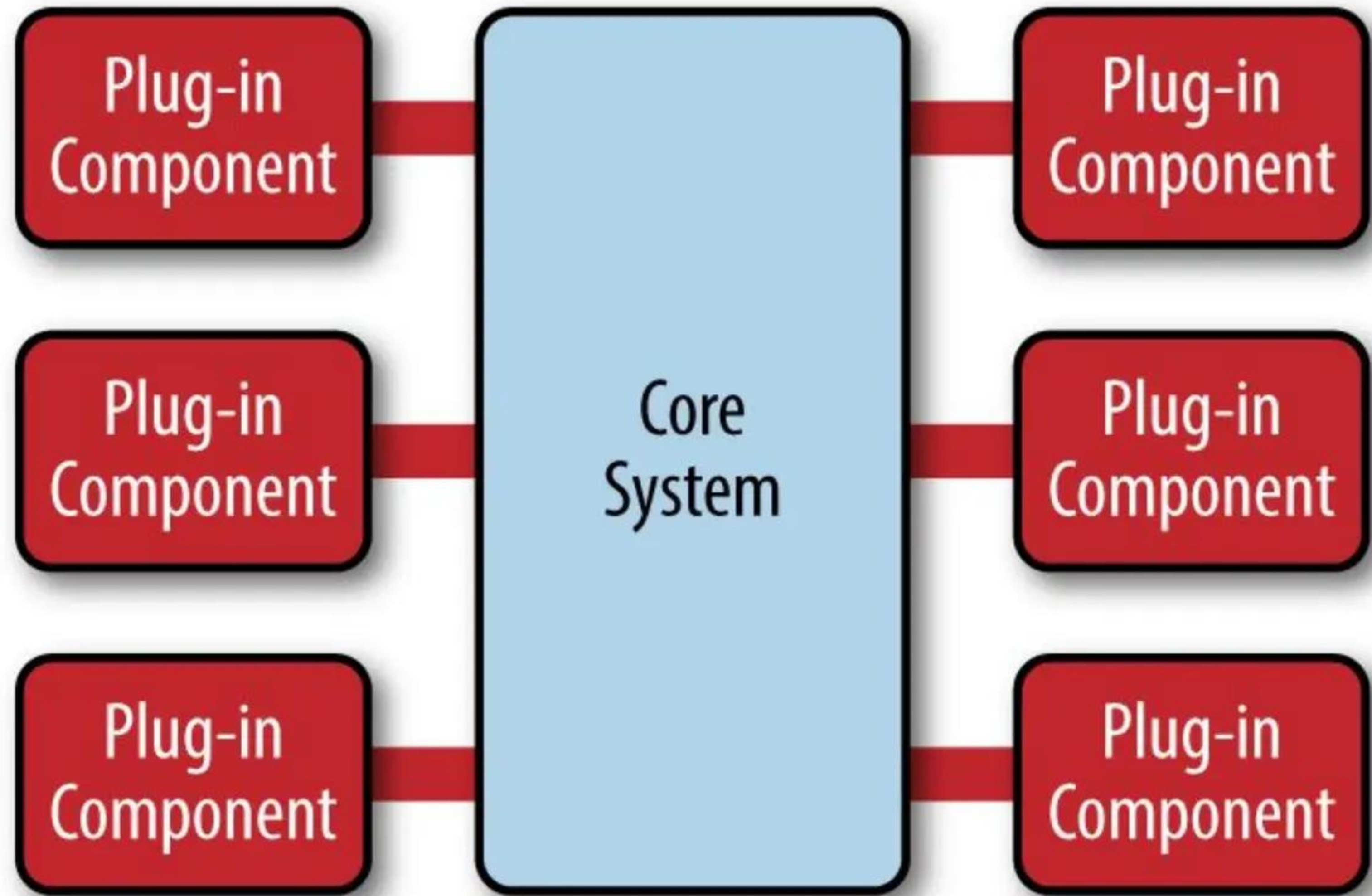
ABSTRACTION



ALGORITHM

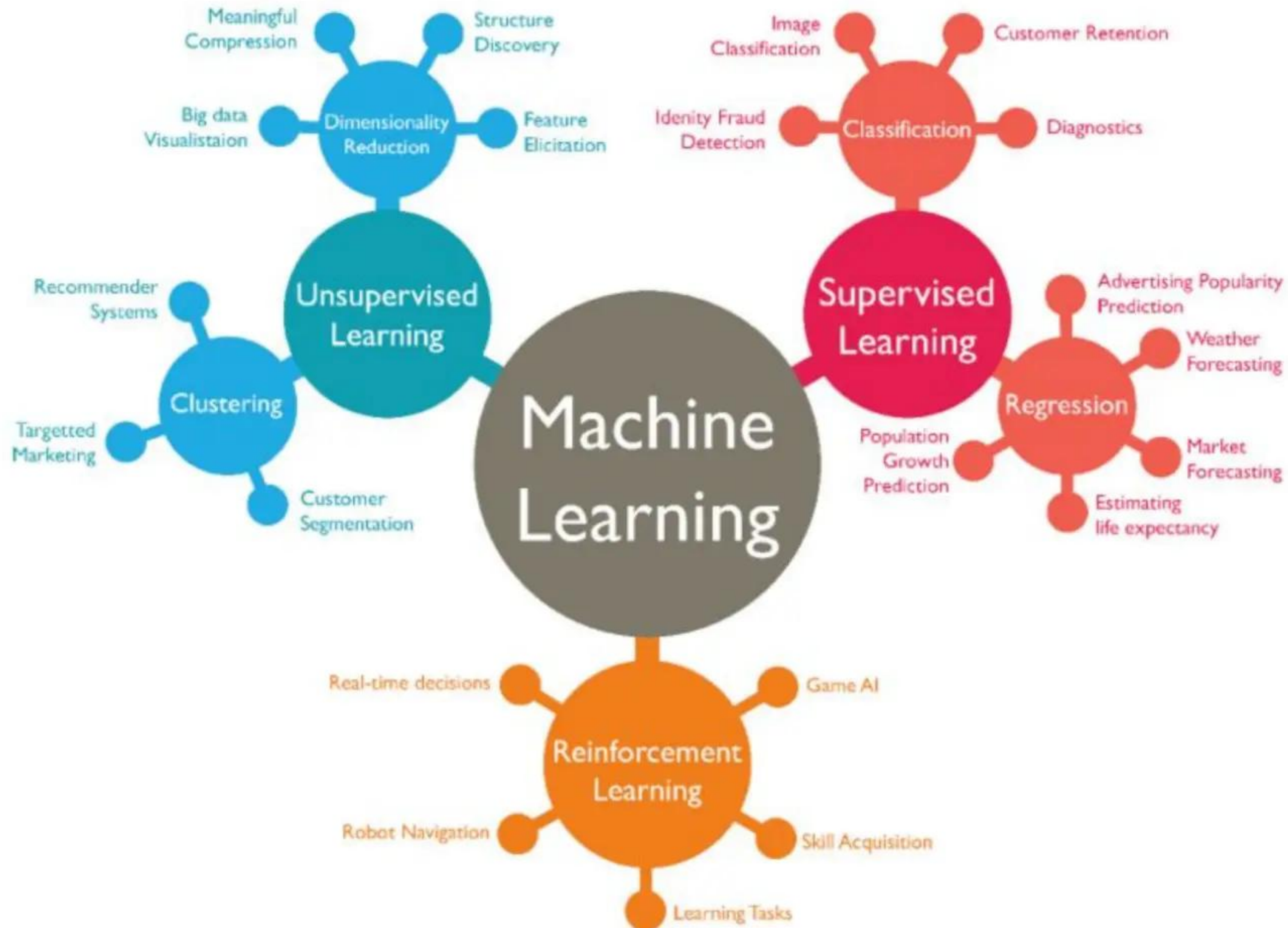


Micro Kernel Architecture





ML Taxonomies





Machine Learning basic things...

- Training
- Evaluating
- Predicting
- Data preprocessings



Engineering, why?

- SOLID Principal
- Reusability:
 - We use the algorithm in backend service and data pipeline
 - Difference technologies like programming language or platform
- Scalability
 - Could adopt several libraries or frameworks or our own implementation
 - Performance scale up and scale out
 - Run more efficient



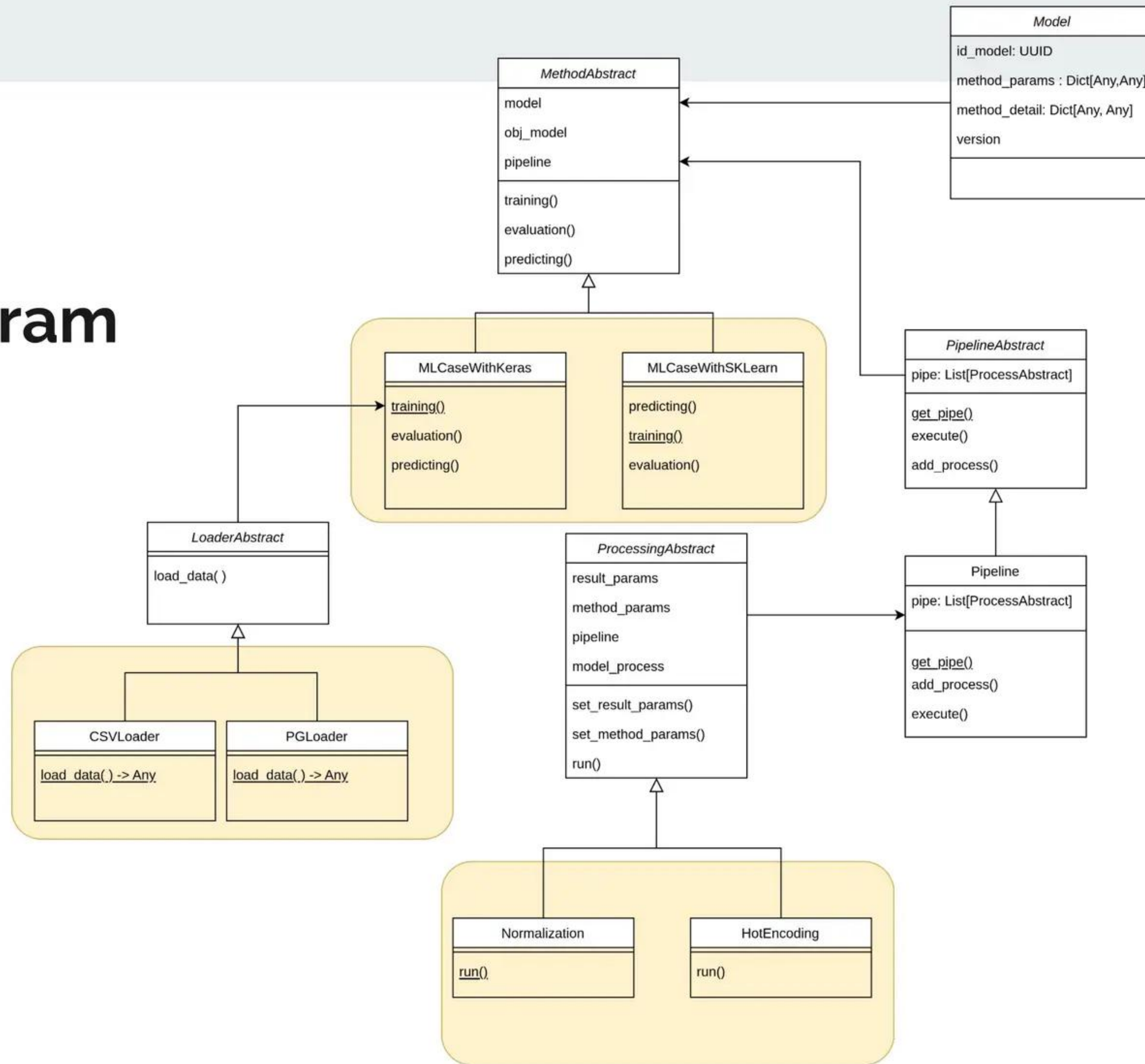
Sample Case

We have 2 cases that need to be handled:

- Keras
- SKLearn

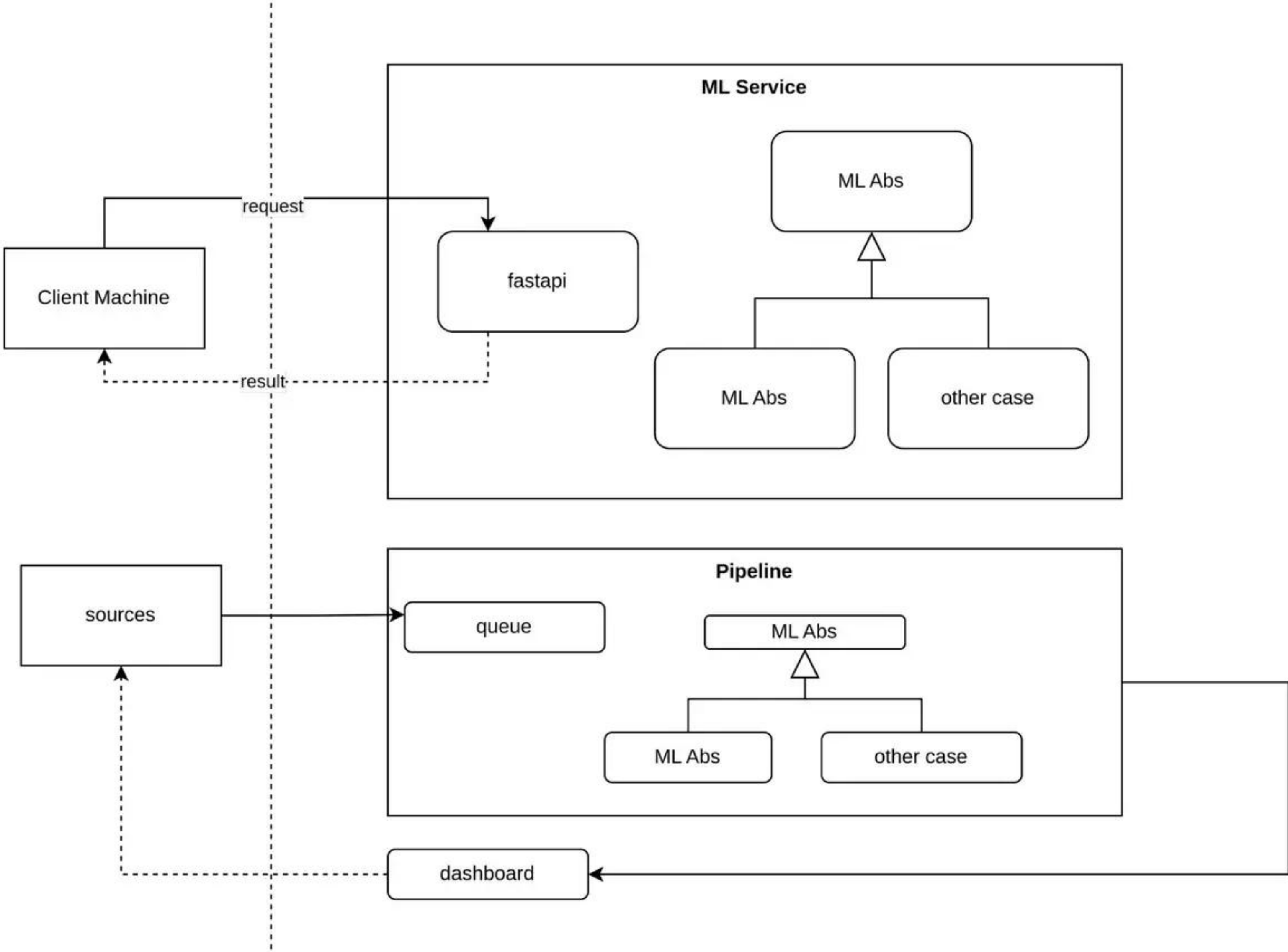
The possibilities are could have different data type or structures and also different model, and it could be different libraries.

Class Diagram





Architecture





Lets see the sample

```

>> [RUN unittest]
def test_ml_predict(self) -> None:
    project_info = ProjectInfo( : ProjectInfo
        model_name='testing_model',
        dataset_id= str(uuid.uuid4()),
        project_id= str(uuid.uuid4()),
        model_desc='SKBackproation testing',
        method_detail={
            'random_state' : 1,
            'max_iter' : 300,
            'solver' : 'sgd',
            'alpha' : 1e-5,
            'hidden_layer_sizes' : (10,)
        },
        version='0.0.1')

    pipeline = Pipeline() : Pipeline
    norm = Normalization('norm1') : Normalization
    norm.set_method_params({
        'field_list' : [
            '_source.sepal_length',
            '_source.sepal_width',
            '_source.petal_length',
            '_source.petal_width'
        ]
    })
    pipeline.add_new_task(norm)

    lmp = SKBackpropagation(dataset = self.df.copy(), label_field='_source.species', info = project_info, pipeline=pipeline) : SKBackpropagation

    model = lmp.train() : Model

    print(model)
    other = SKBackpropagation() : SKBackpropagation
    other.set_model(model)
    sample = { : dict[str, float]
        '_source.sepal_length' : 5.1,
        '_source.sepal_width' : 3.5,
        '_source.petal_length' : 1.4,
        '_source.petal_width' : 0.2
    }
    result = other.predict(sample) : List[float]
    print(result)
    assert other.model.project_id == UUID(project_info.project_id)
    assert model.project_id == UUID(project_info.project_id)
    assert type(model.model) == dict

```

Any Questions ?

Thank you

