RAY

Ray for Reinforcement Learning

A general-purpose system for parallel and distributed Python

https://github.com/ray-project/ray

Robert Nishihara

@robertnishihara

riselab

UC Berkeley

A Growing Number of Use Cases

蚂蚁金服 ANT FINANCIAL

Berkeley UNIVERSITY OF CALIFORNIA

BONSAI AI

ERICSSON

intel

Alibaba.com

facebook

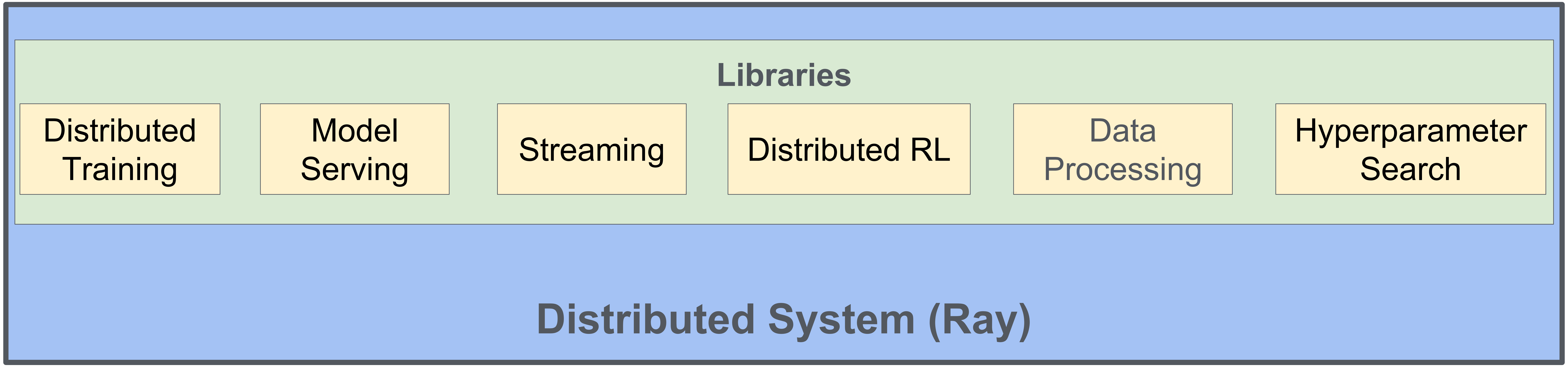
J.P.Morgan

Morgan Stanley

PRIMER

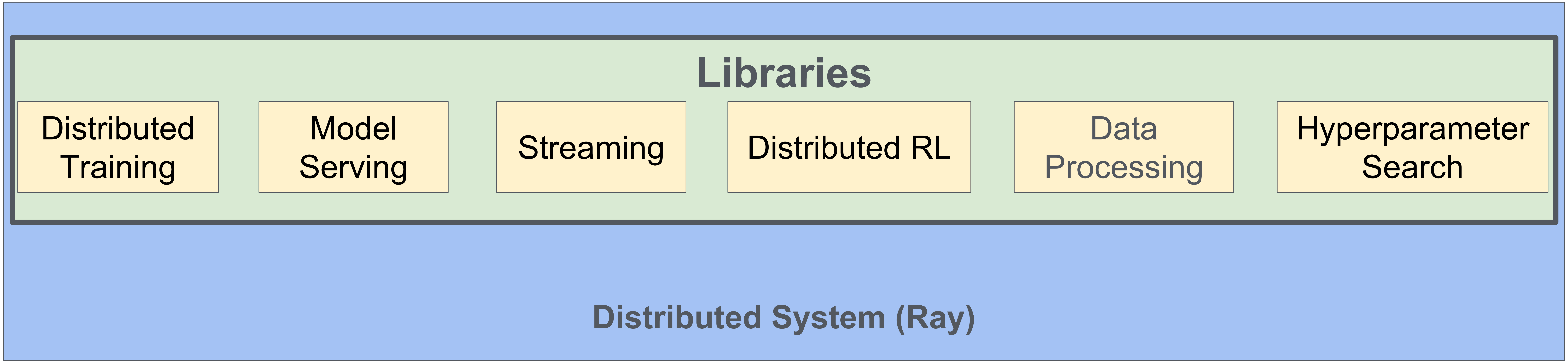
Microsoft

The Big Picture



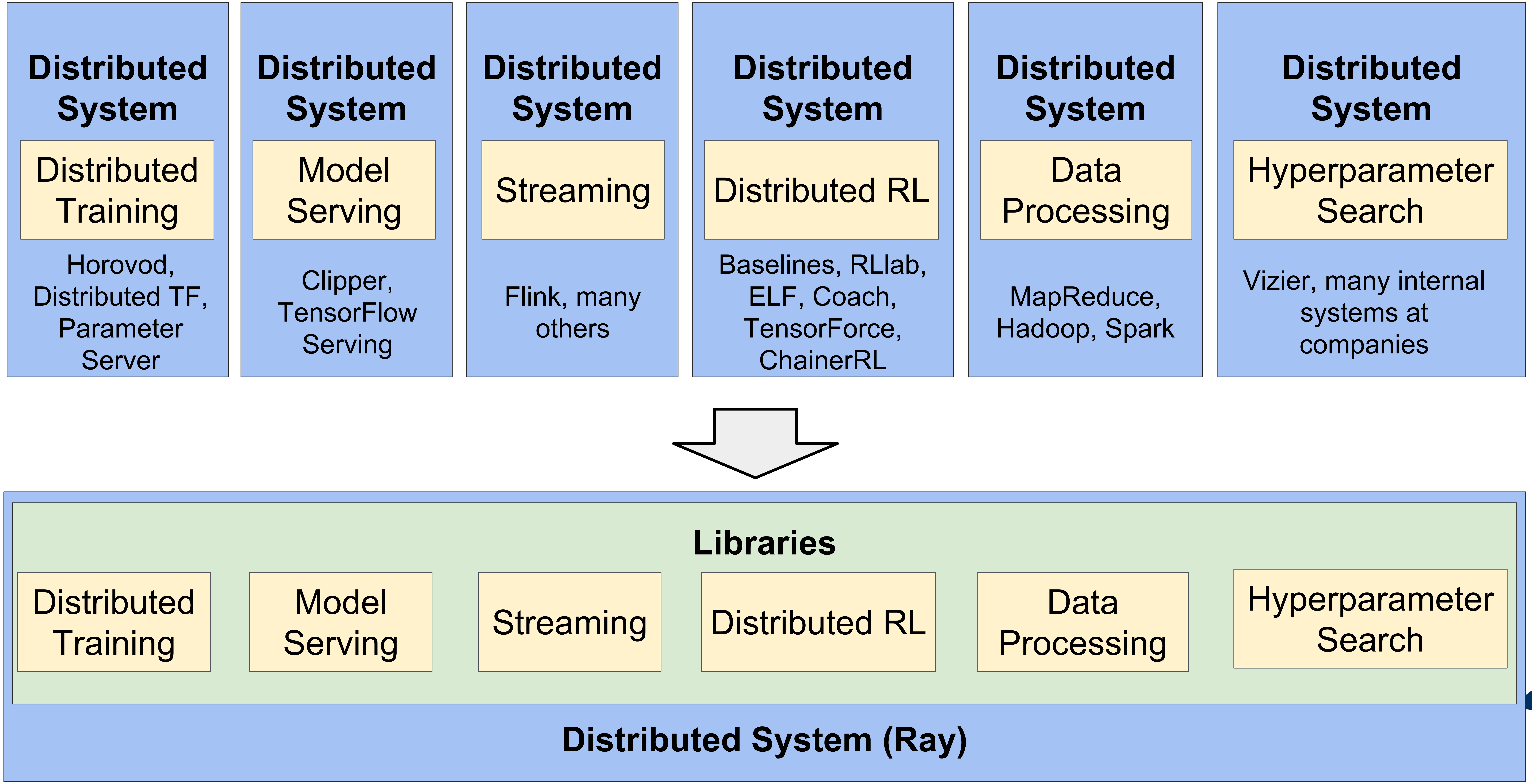
©2017 RISELab

The Big Picture

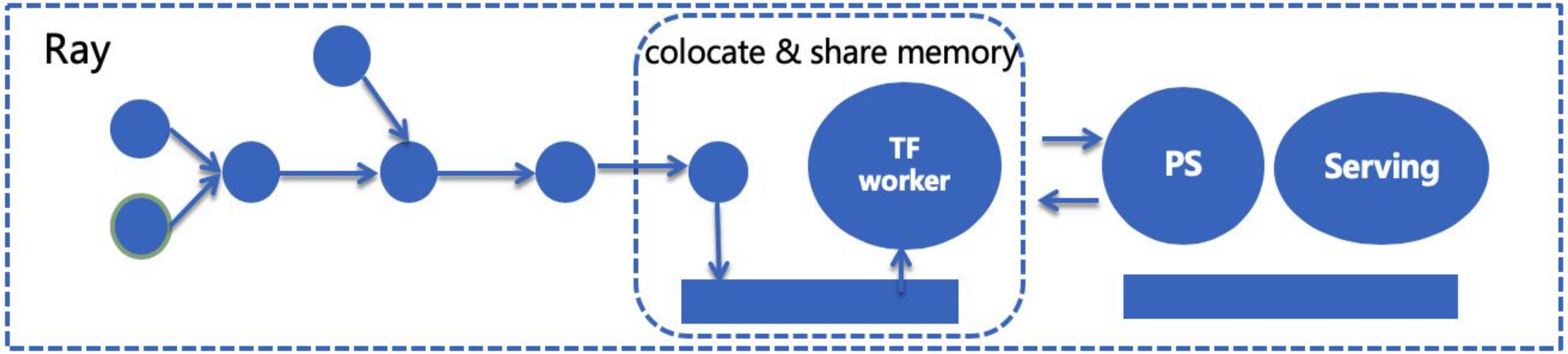


©2017 RISELab

The Big Picture



Use Case: Online Machine Learning



- 3 min, streaming + model training, from feature / label to model output
- 5 min, streaming + training + serving, from feature / label to model deploy
- 5% CTR improvement comparing to offline model; 1% CTR improvement comparing to blink solution

Ray API

Functions -> Tasks

```
def read_array(file):
    # read array “a” from “file”
    return a


def add(a, b):
    return np.add(a, b)
```


©2017 RISELab

Ray API

Functions -> Tasks

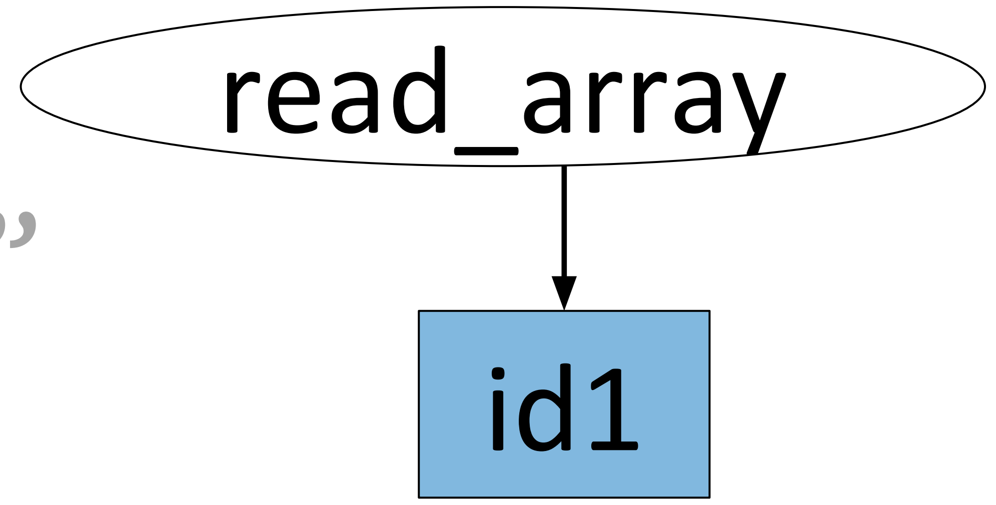
```
@ray.remote
def read_array(file):
    # read array “a” from “file”
    return a

@ray.remote
def add(a, b):
    return np.add(a, b)
```


©2017 RISELab

Ray API

Functions -> Tasks

```
@ray.remote
def read_array(file):
    # read array “a” from “file”
    return a

@ray.remote
def add(a, b):
    return np.add(a, b)

id1 = read_array.remote([5, 5])
```

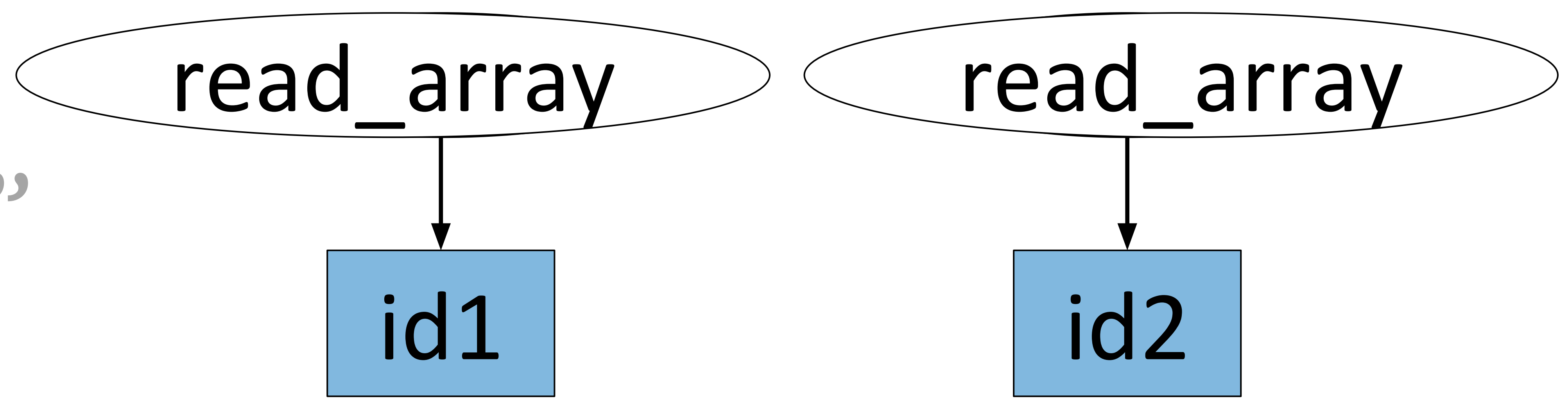


©2017 RISELab

Ray API

Functions -> Tasks

```
@ray.remote
def read_array(file):
    # read array “a” from “file”
    return a

@ray.remote
def add(a, b):
    return np.add(a, b)

id1 = read_array.remote([5, 5])
id2 = read_array.remote([5, 5])
```

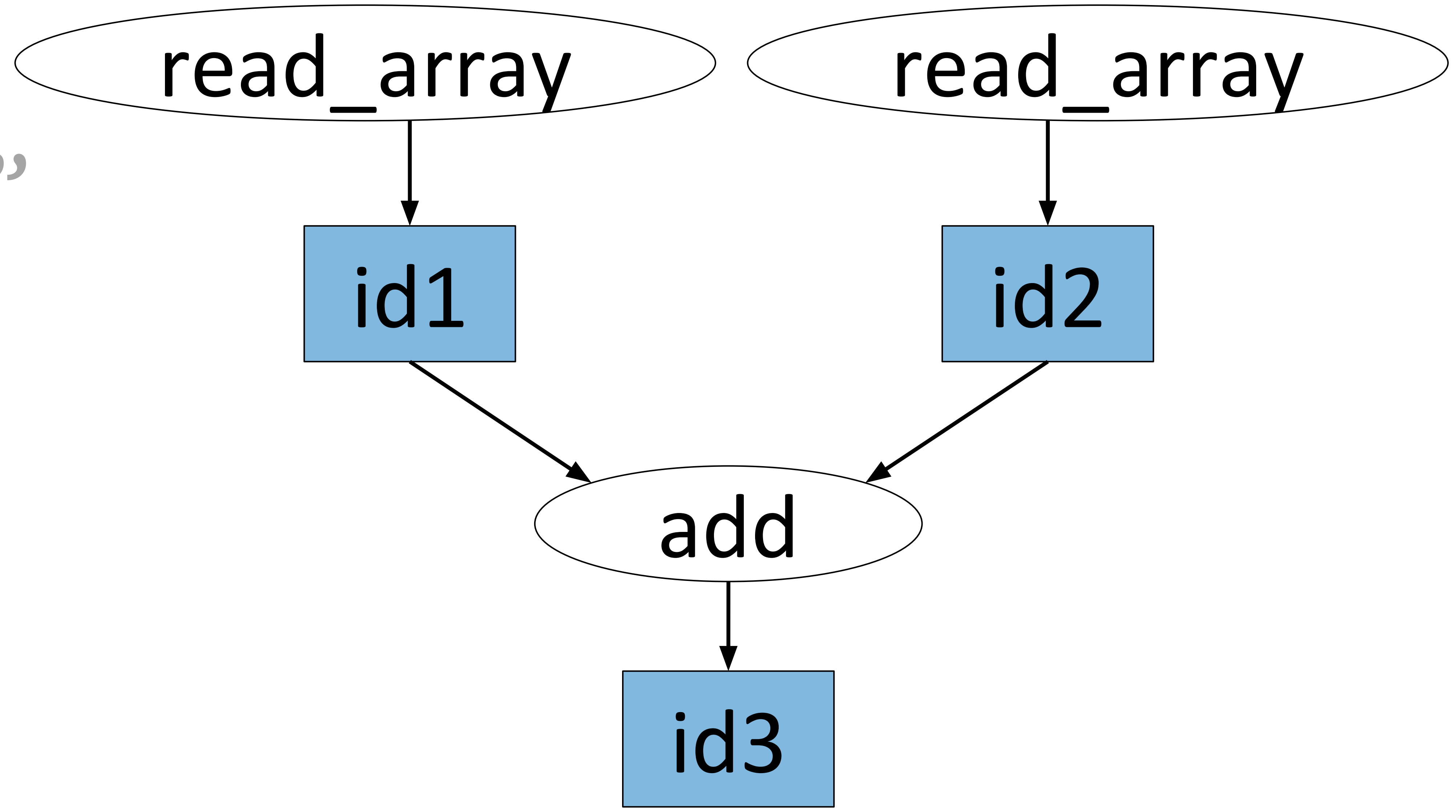


©2017 RISELab

Ray API

Functions -> Tasks

```
@ray.remote
def read_array(file):
    # read array "a" from "file"
    return a

@ray.remote
def add(a, b):
    return np.add(a, b)

id1 = read_array.remote([5, 5])
id2 = read_array.remote([5, 5])
id3 = add.remote(id1, id2)
```

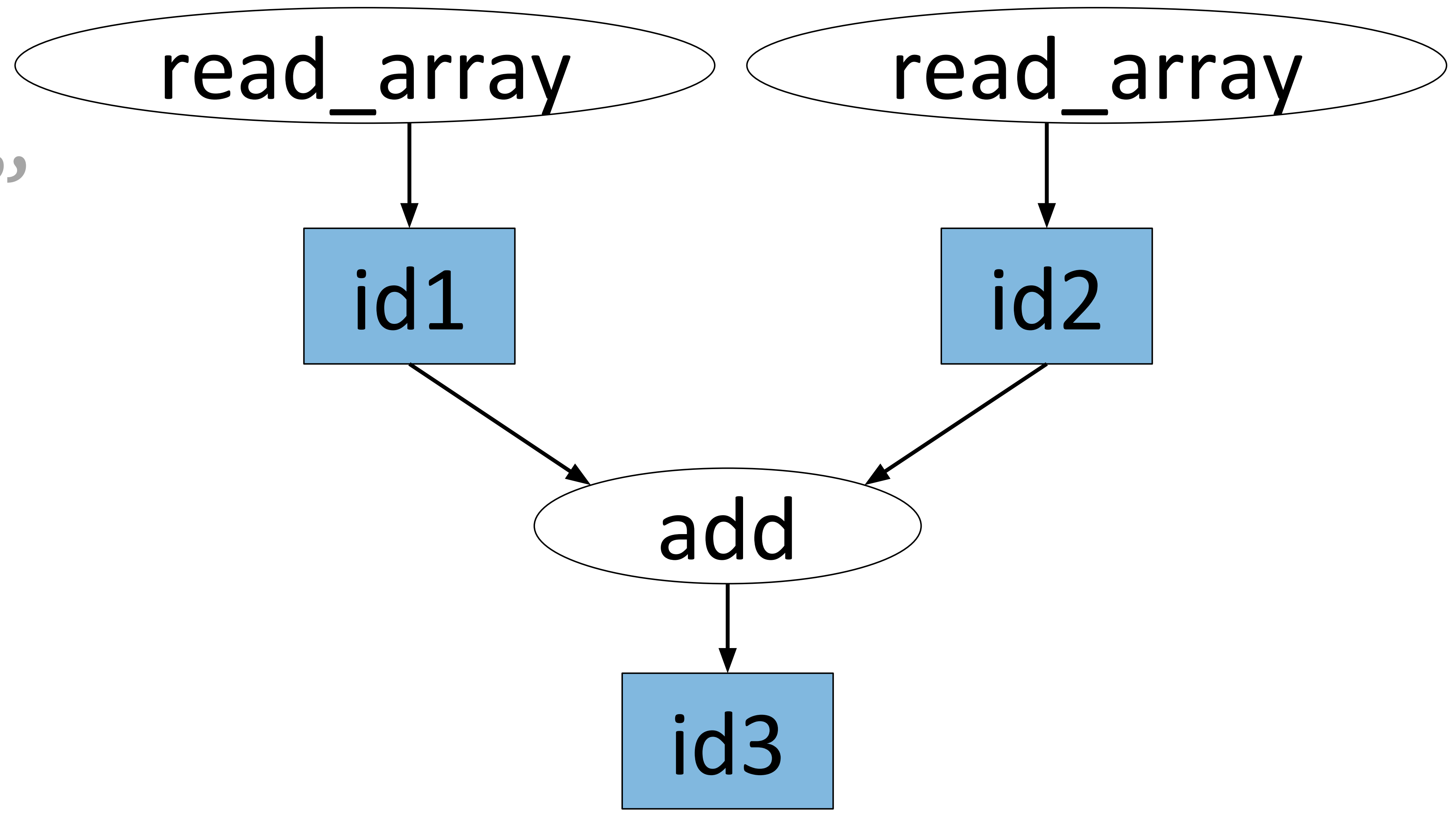


©2017 RISELab

Ray API

Functions -> Tasks

```
@ray.remote
def read_array(file):
    # read array “a” from “file”
    return a

@ray.remote
def add(a, b):
    return np.add(a, b)

id1 = read_array.remote([5, 5])
id2 = read_array.remote([5, 5])
id3 = add.remote(id1, id2)
ray.get(id3)
```



©2017 RISELab

Ray API

Functions -> Tasks

```
@ray.remote
def read_array(file):
    # read array “a” from “file”
    return a

@ray.remote
def add(a, b):
    return np.add(a, b)

id1 = read_array.remote([5, 5])
id2 = read_array.remote([5, 5])
id3 = add.remote(id1, id2)
ray.get(id3)
```

Classes -> Actors

©2017 RISELab

Ray API

Functions -> Tasks

```
@ray.remote
def read_array(file):
    # read array “a” from “file”
    return a

@ray.remote
def add(a, b):
    return np.add(a, b)

id1 = read_array.remote([5, 5])
id2 = read_array.remote([5, 5])
id3 = add.remote(id1, id2)
ray.get(id3)
```

Classes -> Actors

```
@ray.remote(num_gpus=1)
class Counter(object):
    def __init__(self):
        self.value = 0
    def inc(self):
        self.value += 1
        return self.value
```


©2017 RISELab

Ray API

Functions -> Tasks

```python
@ray.remote
def read_array(file):
    # read array “a” from “file”
    return a

@ray.remote
def add(a, b):
    return np.add(a, b)

id1 = read_array.remote([5, 5])
id2 = read_array.remote([5, 5])
id3 = add.remote(id1, id2)
ray.get(id3)
```

Classes -> Actors

```python
@ray.remote(num_gpus=1)
class Counter(object):
    def __init__(self):
        self.value = 0
    def inc(self):
        self.value += 1
        return self.value

c = Counter.remote()
id4 = c.inc.remote()
id5 = c.inc.remote()
ray.get([id4, id5])
```


©2017 RISELab

Actors: Parameter Server Example

```
@ray.remote
class ParameterServer(object):
    def __init__(self):
        self.params = np.zeros(10)
    def get_params(self):
        return self.params
    def update_params(self, grad):
        self.params -= grad
```


©2017 RISELab

Actors: Parameter Server Example

```python
@ray.remote
class ParameterServer(object):
    def __init__(self):
        self.params = np.zeros(10)
    def get_params(self):
        return self.params
    def update_params(self, grad):
        self.params -= grad
```

```python
@ray.remote(num_gpus=1)
def worker(ps):
    while True:
        params = ray.get(ps.get_params.remote())
        grad = ...   # Use TensorFlow
        ps.update_params.remote(grad)
```

Actors: Parameter Server Example

```python
@ray.remote
class ParameterServer(object):
    def __init__(self):
        self.params = np.zeros(10)
    def get_params(self):
        return self.params
    def update_params(self, grad):
        self.params -= grad
```

```python
@ray.remote(num_gpus=1)
def worker(ps):
    while True:
        ...
```

```python
@ray.remote(num_gpus=1)
def worker(ps):
    while True:
        ...
```

```python
@ray.remote(num_gpus=1)
def worker(ps):
    while True:
        params = ray.get(ps.get_params.remote())
        grad = ...  # Use TensorFlow
        ps.update_params.remote(grad)
```

Actors: Parameter Server Example

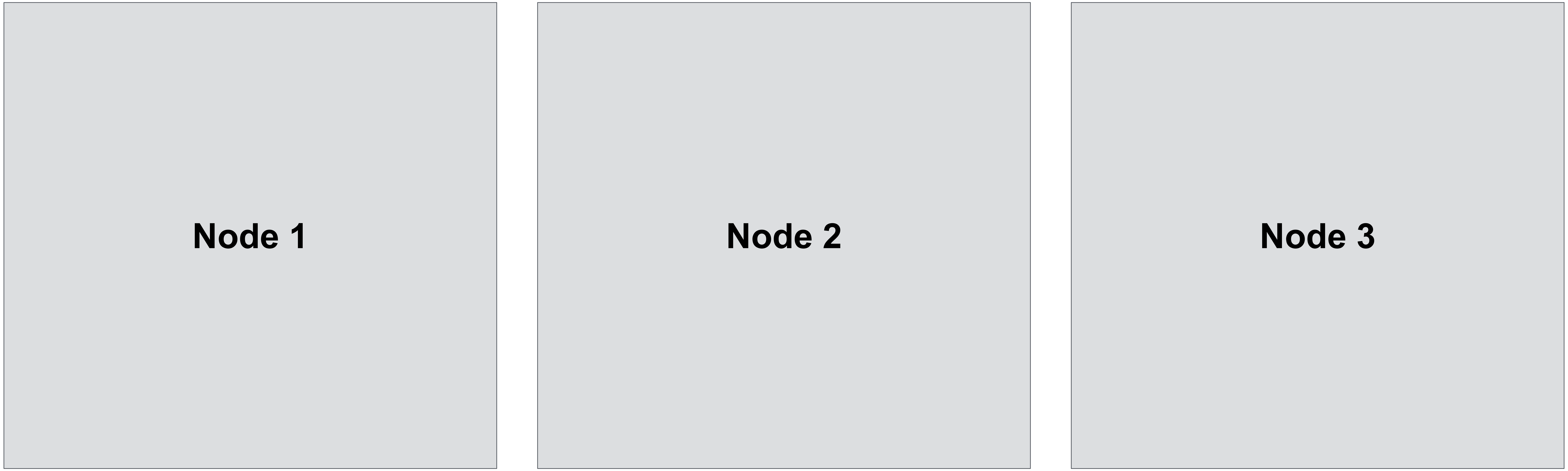
```python
@ray.remote
class ParameterServer(object):
    def __init__(self):
        self.params = np.zeros(10)
    def get_params(self):
        return self.params
    def update_params(self, grad):
        self.params -= grad
```

```python
@ray.remote(num_gpus=1)
def worker(ps):
    while True:
        ...
```

```python
@ray.remote(num_gpus=1)
def worker(ps):
    while True:
        ...
```

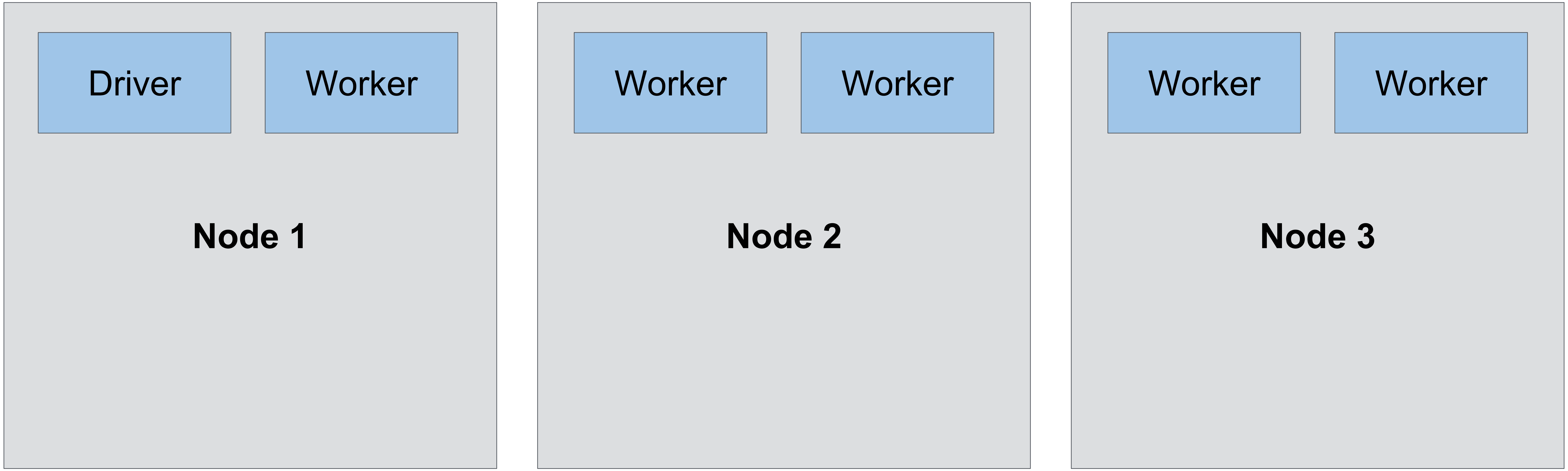
```python
@ray.remote(num_gpus=1)
def worker(ps):
    while True:
        params = ray.get(ps.get_params.remote())
        grad = ...  # Use TensorFlow
        ps.update_params.remote(grad)
```

Ray Architecture



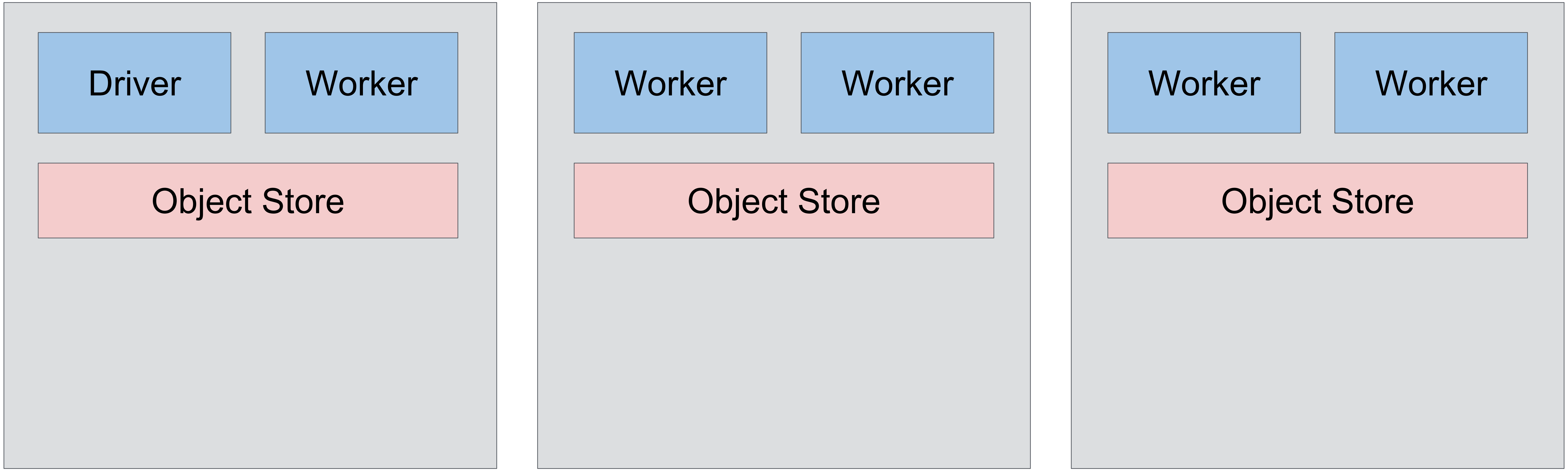
©2017 RISELab

Ray Architecture



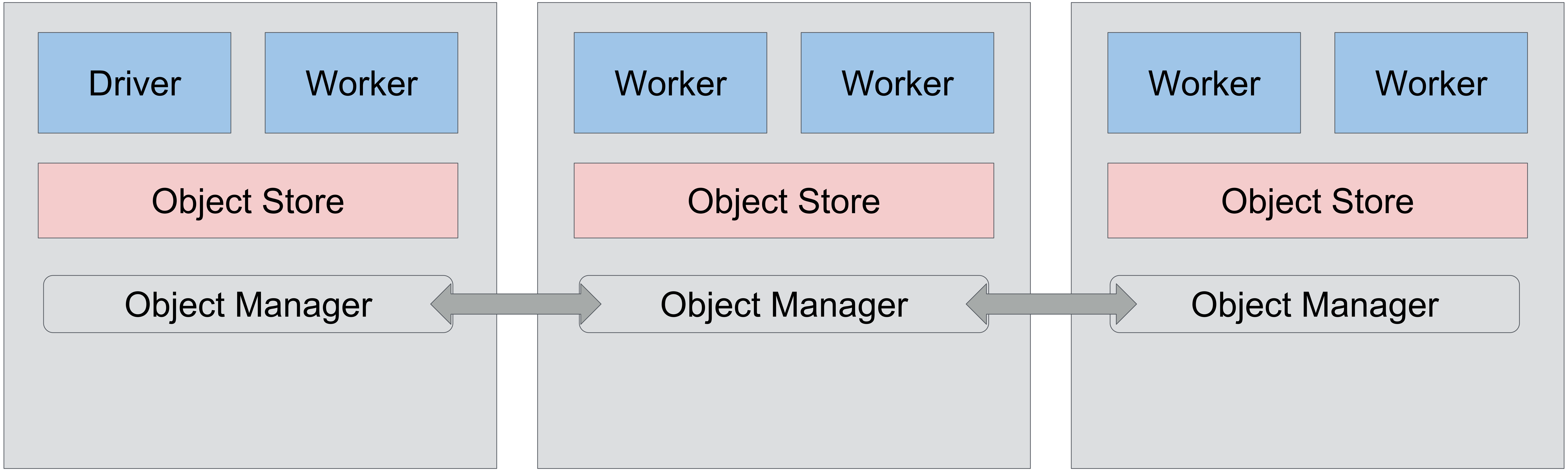
©2017 RISELab

Ray Architecture



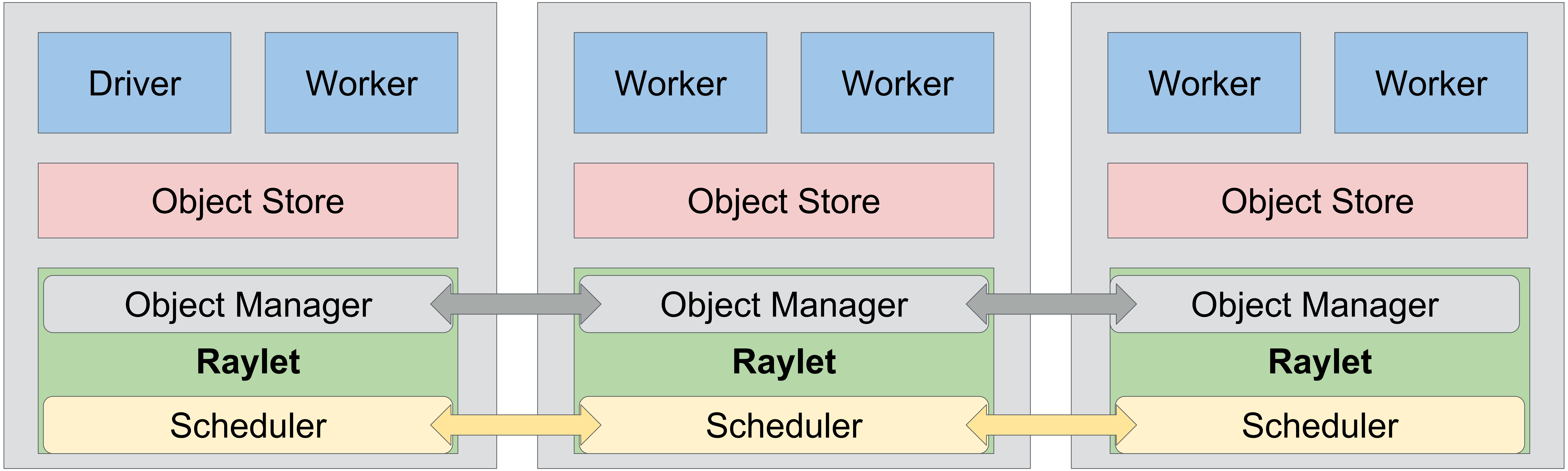
©2017 RISELab

Ray Architecture



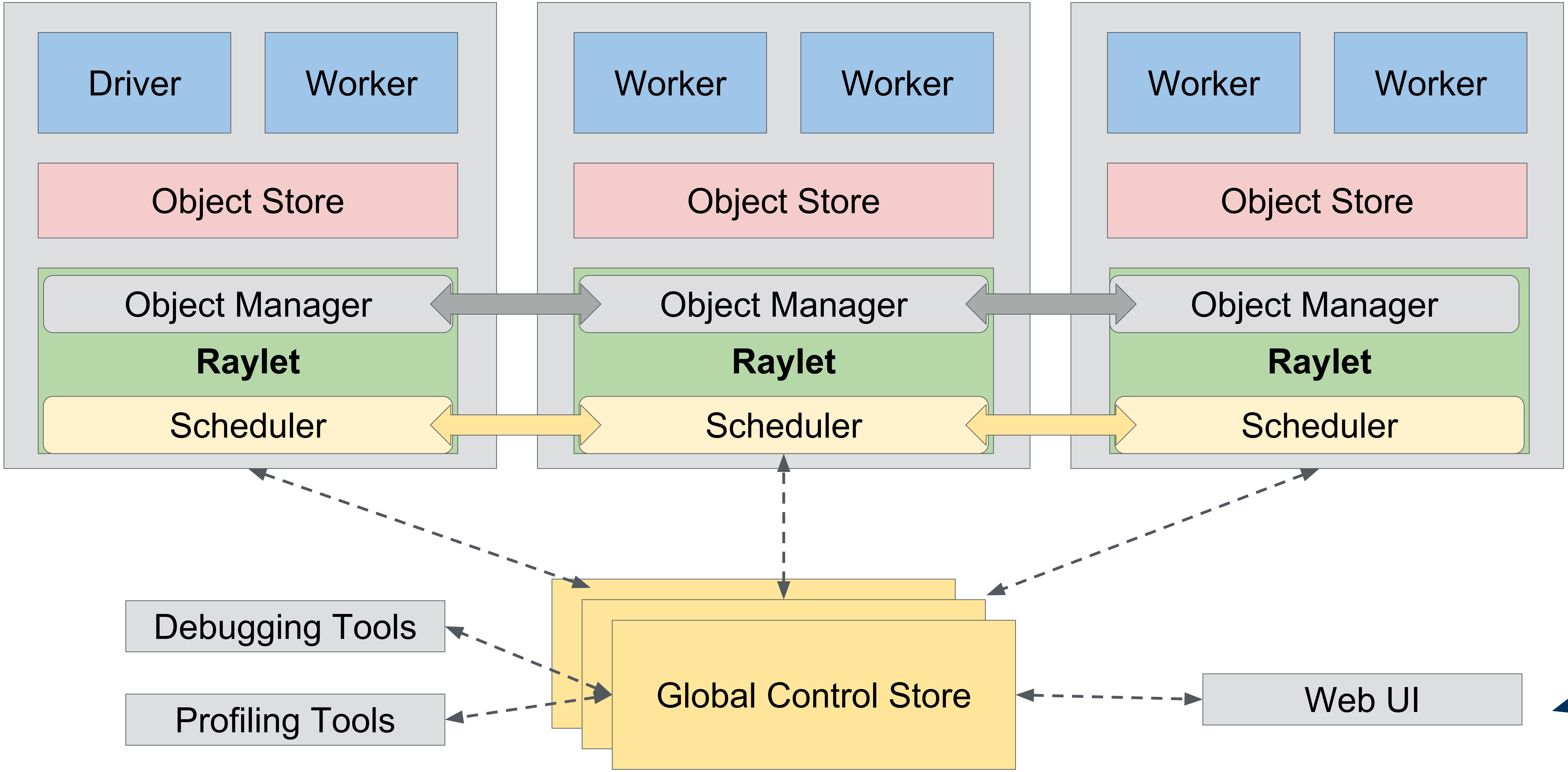
©2017 RISELab

Ray Architecture



©2017 RISELab

Ray Architecture

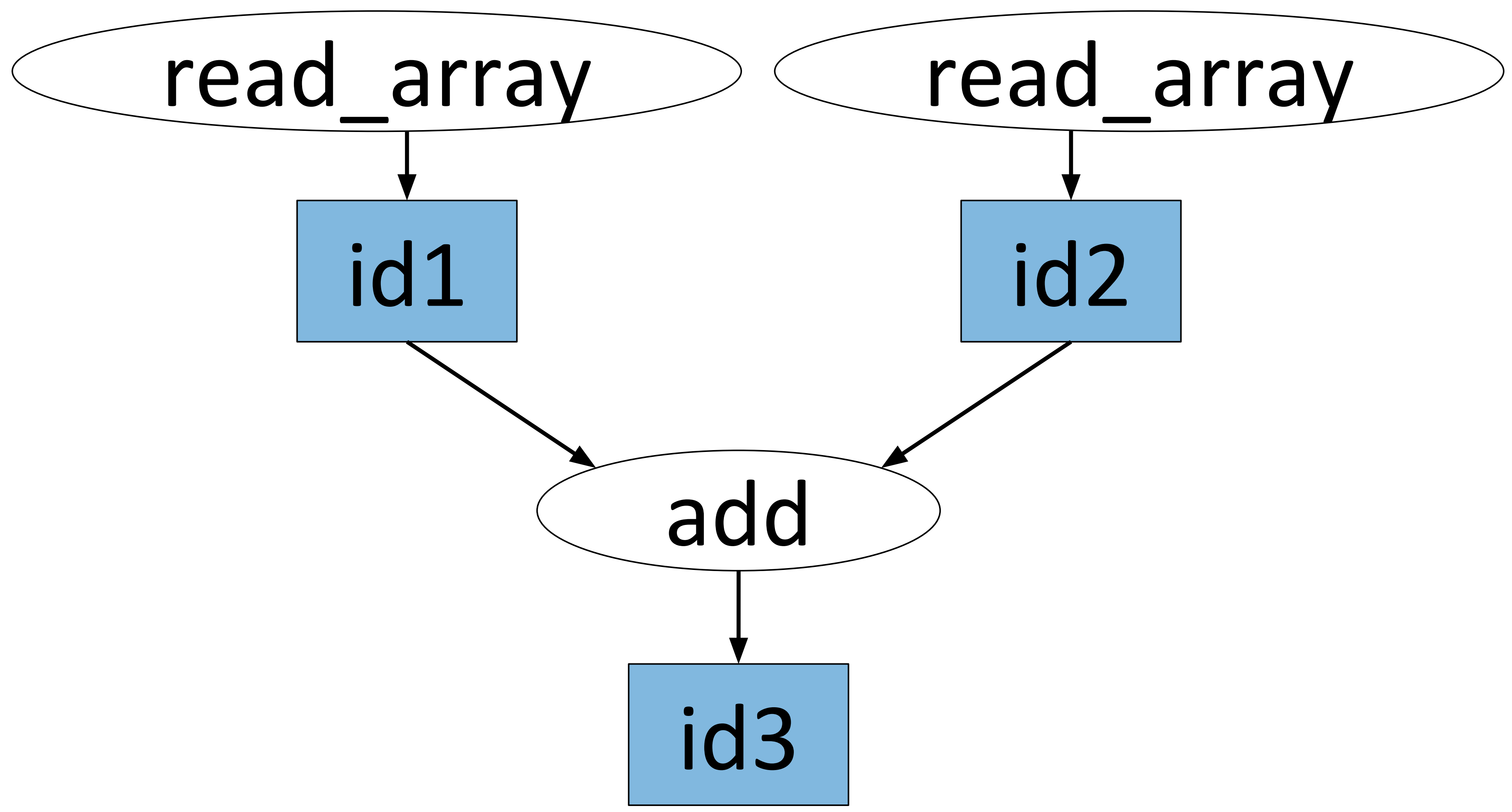


How does this work under the hood?

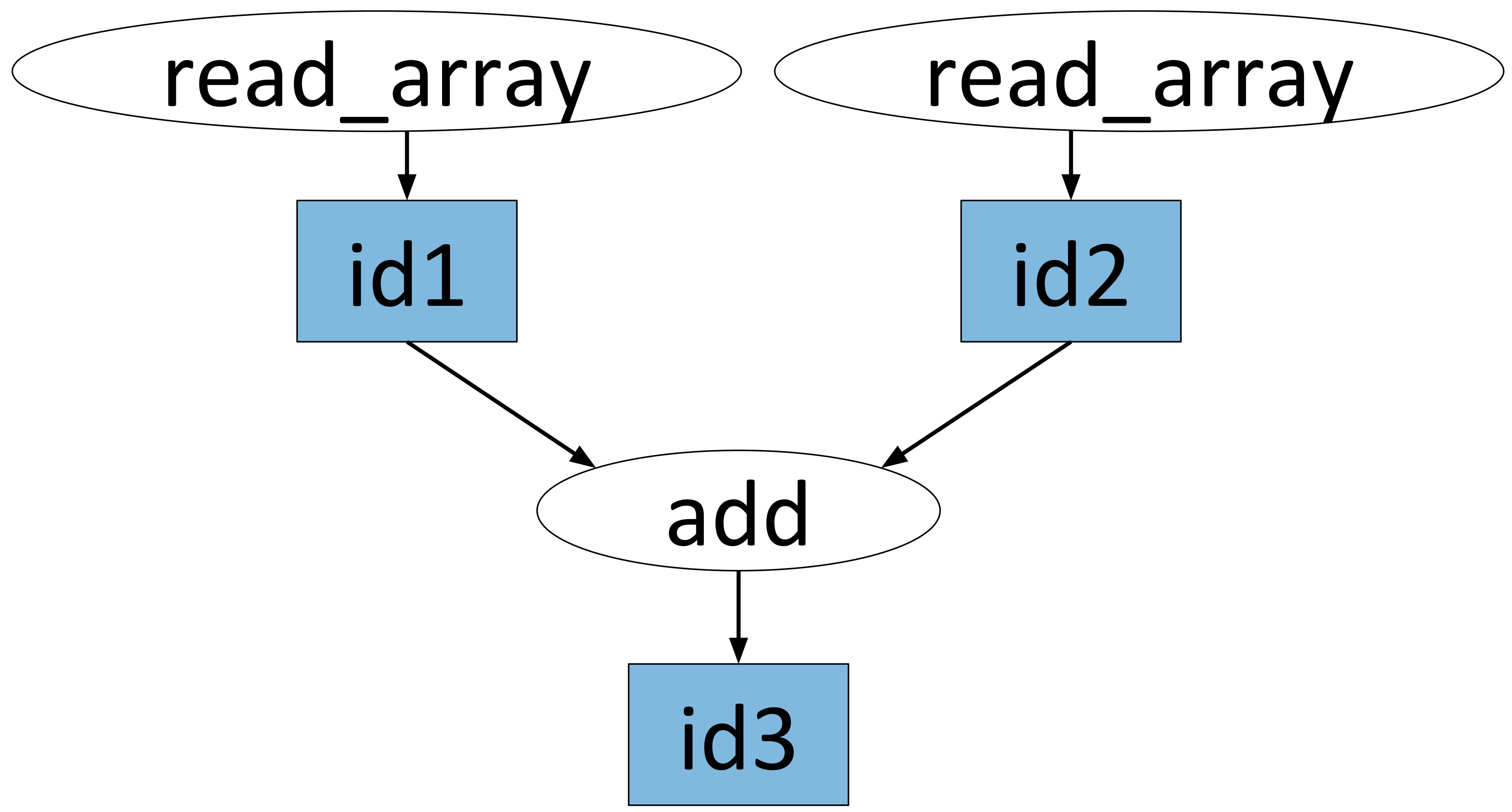
How does this work under the hood?

Tasks

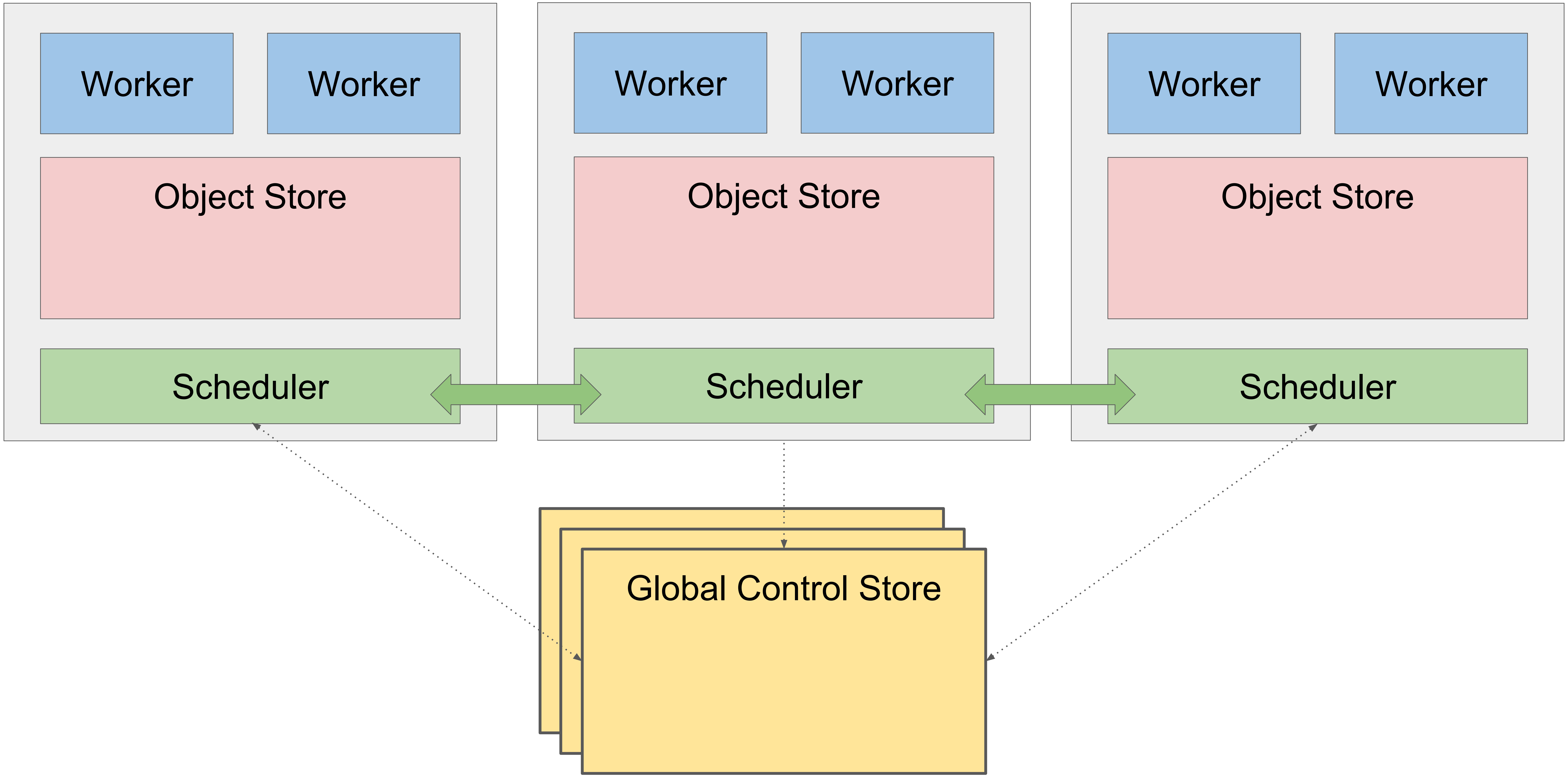
```
@ray.remote
def read_array(file):
    # read array “a” from “file”
    return a

@ray.remote
def add(a, b):
    return np.add(a, b)

id1 = read_array.remote([5, 5])
id2 = read_array.remote([5, 5])
id3 = add.remote(id1, id2)
ray.get(id3)
```



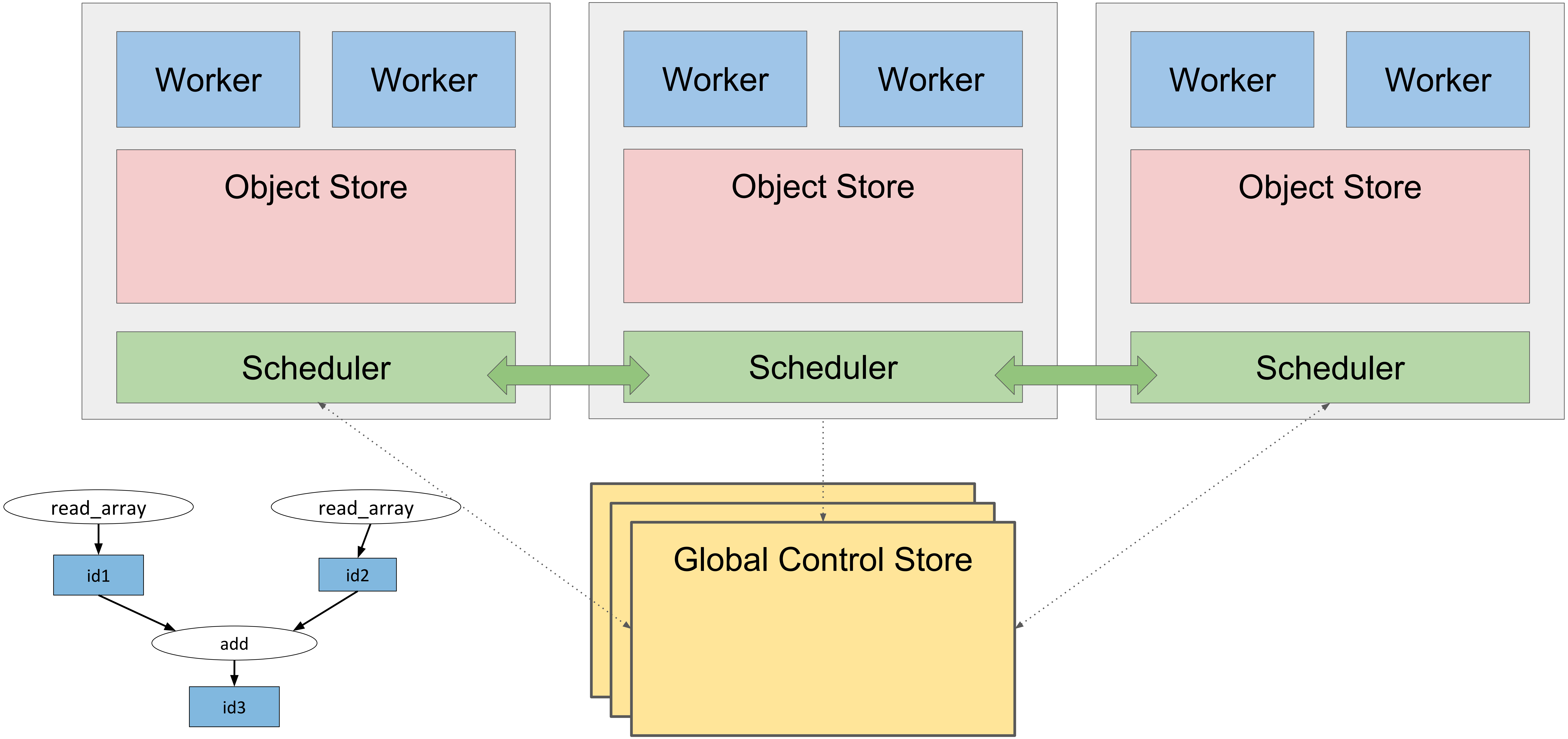
How does this work under the hood?



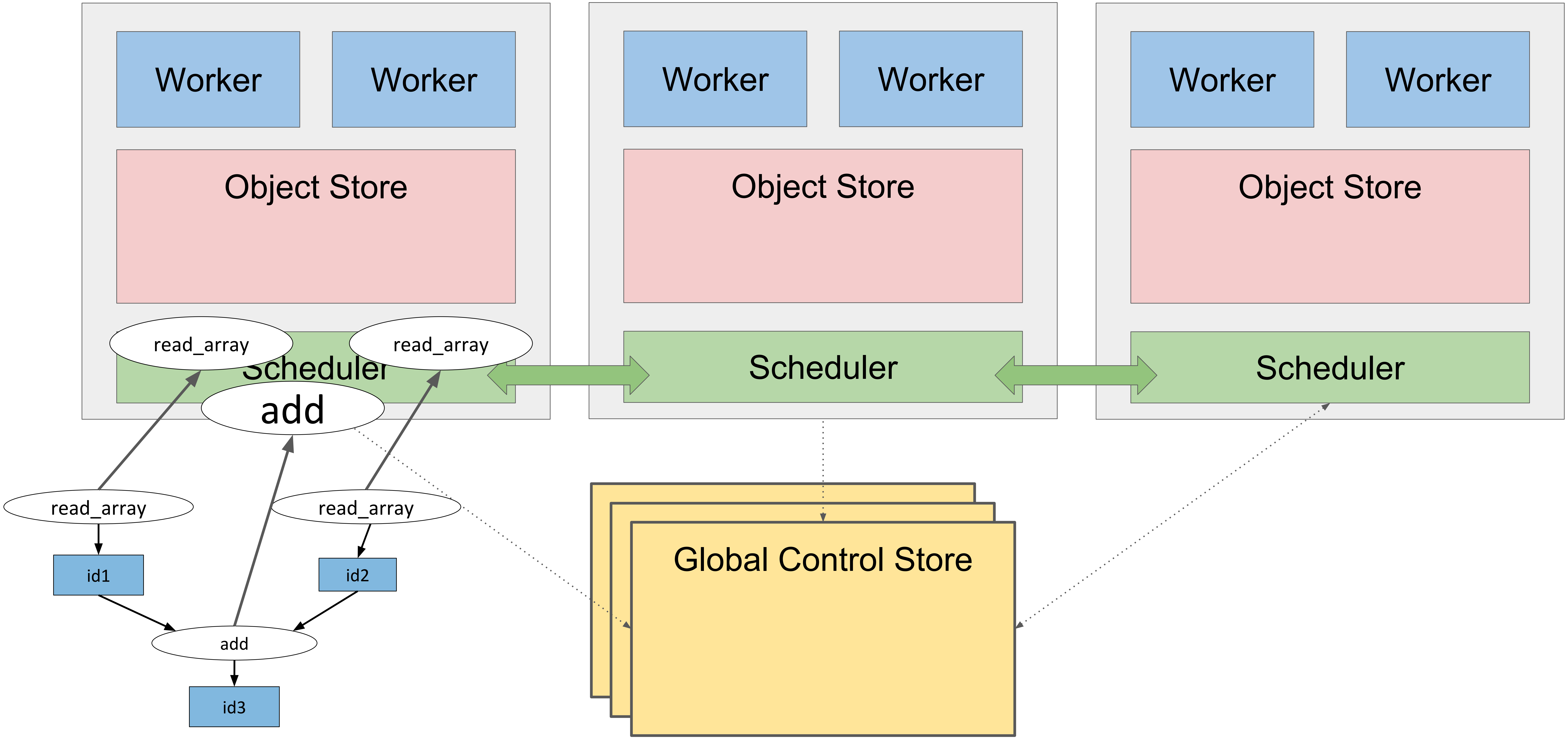
The Ray Architecture



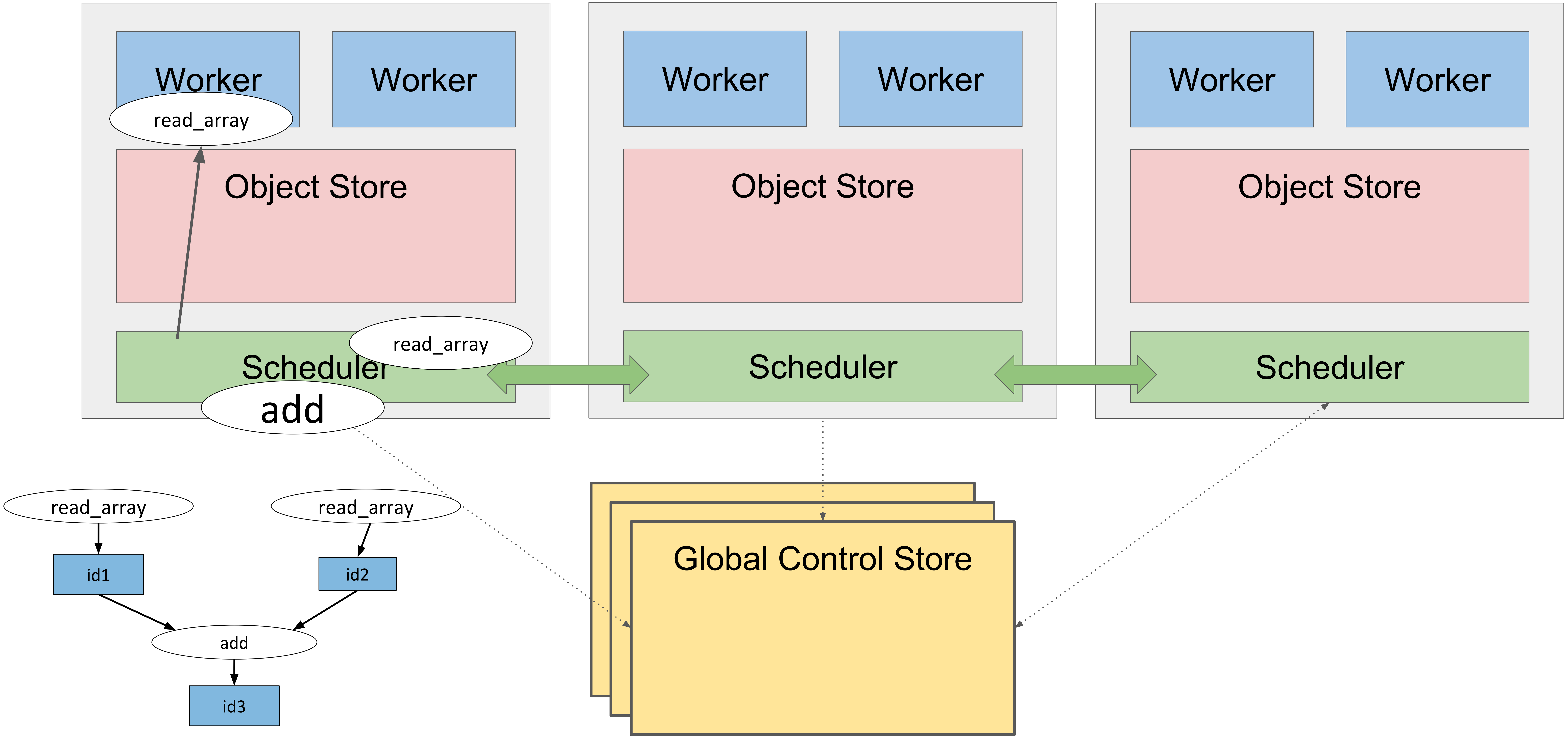
The Ray Architecture



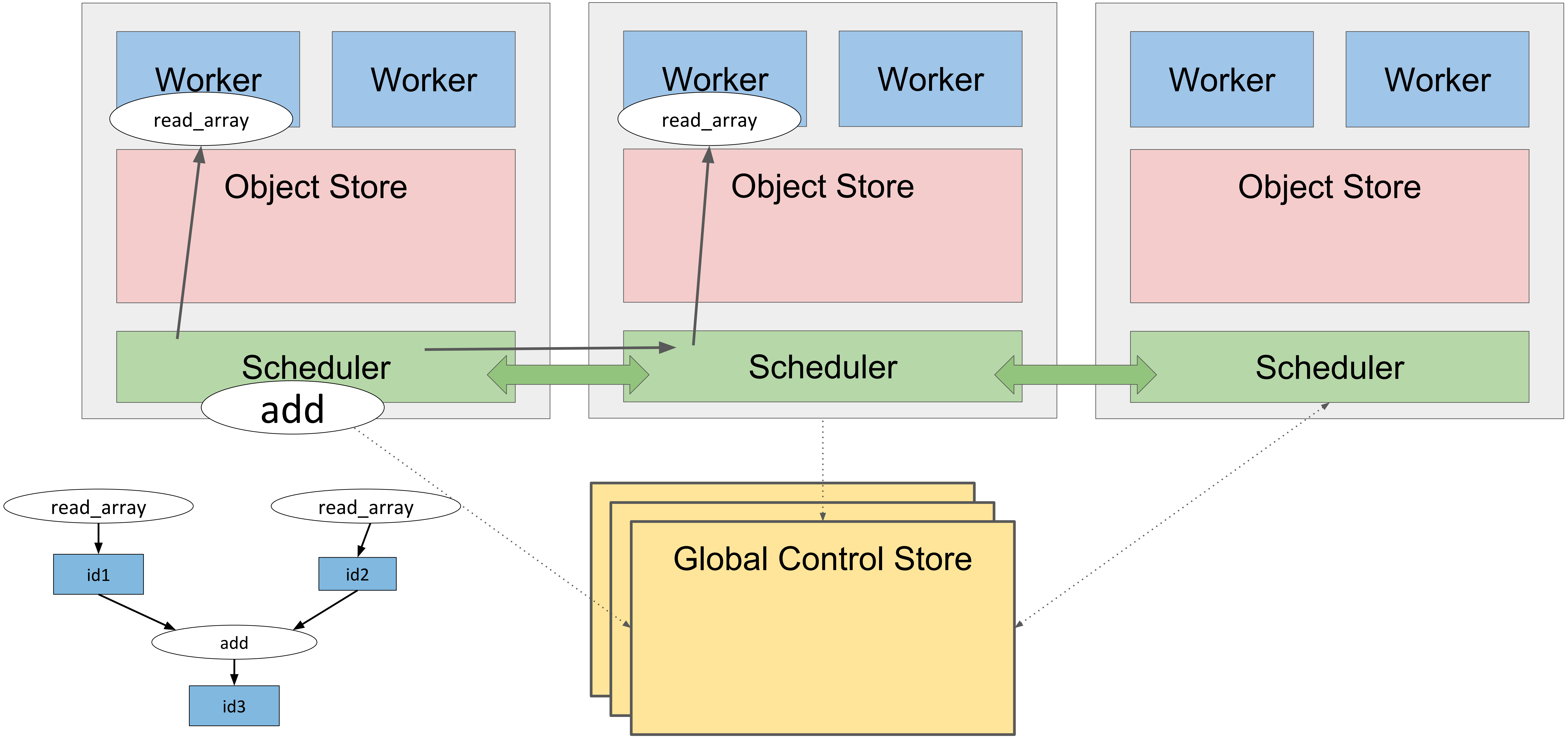
The Ray Architecture



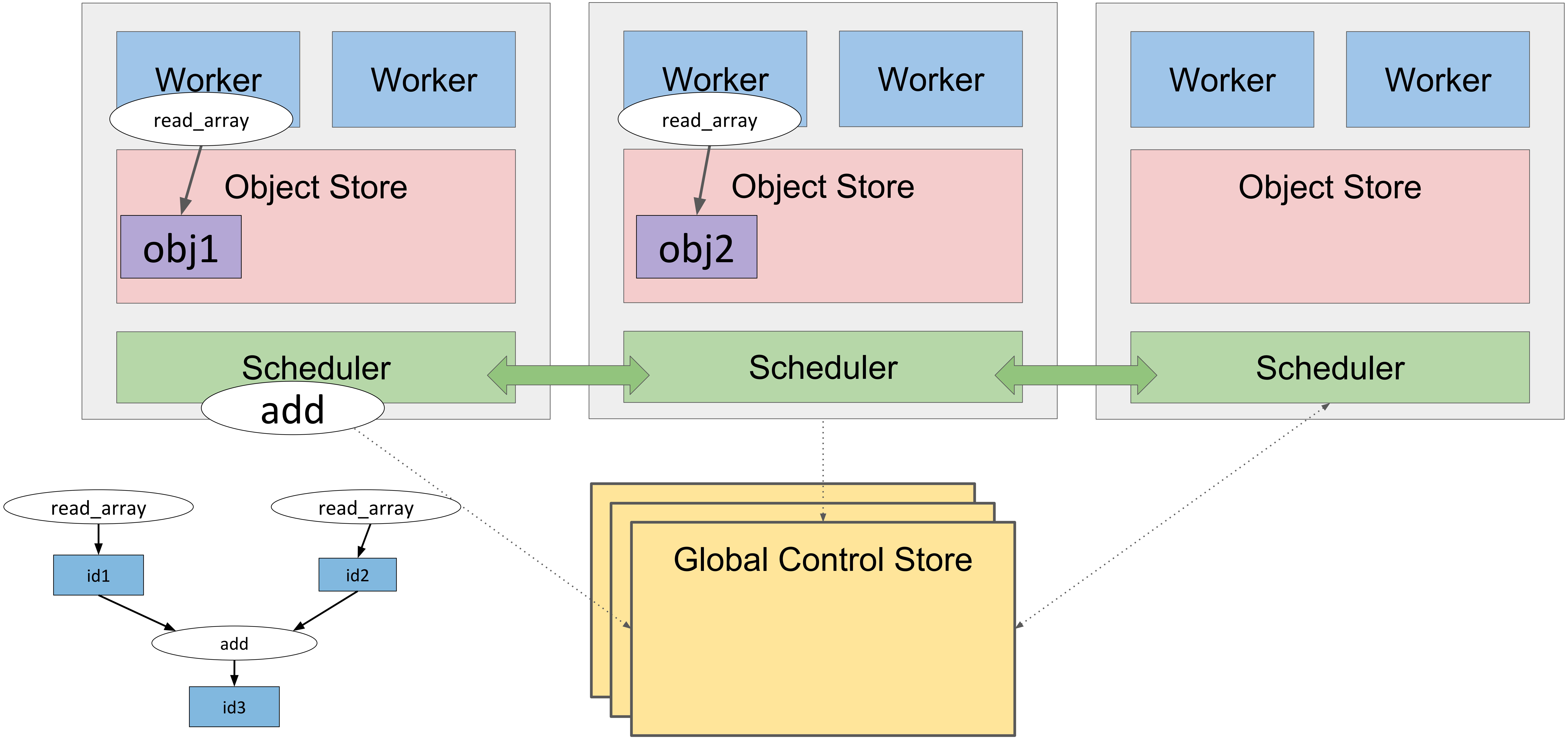
The Ray Architecture



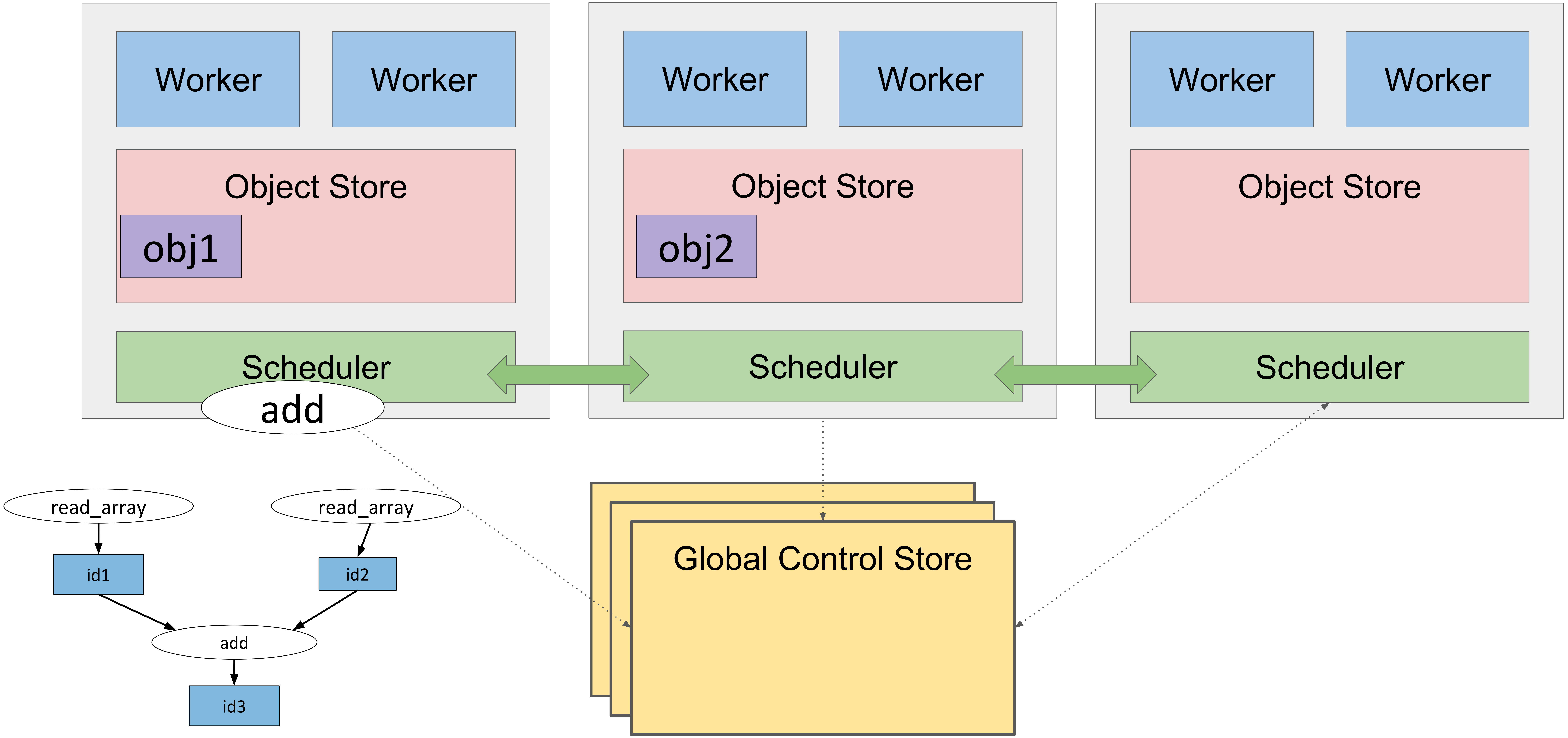
The Ray Architecture



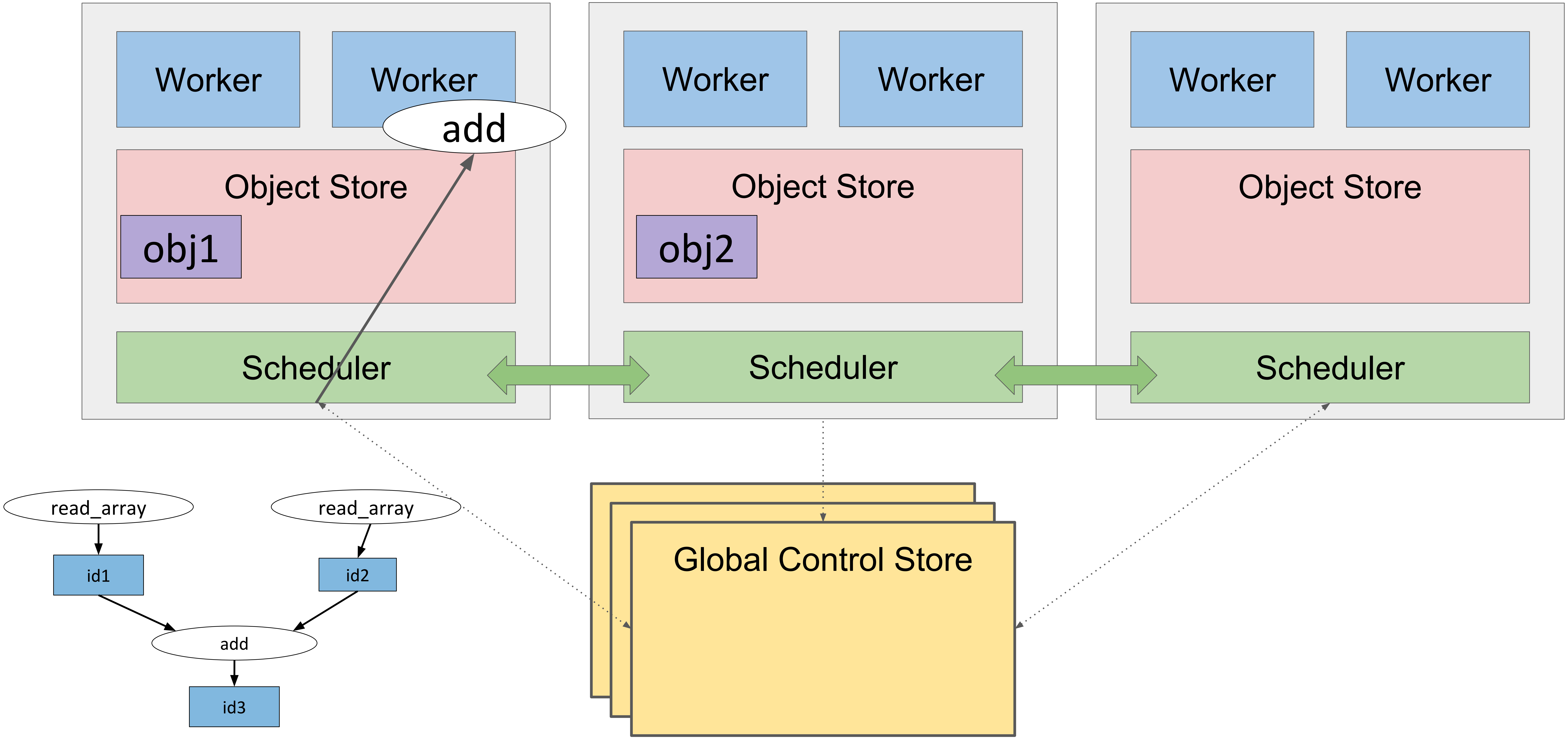
The Ray Architecture



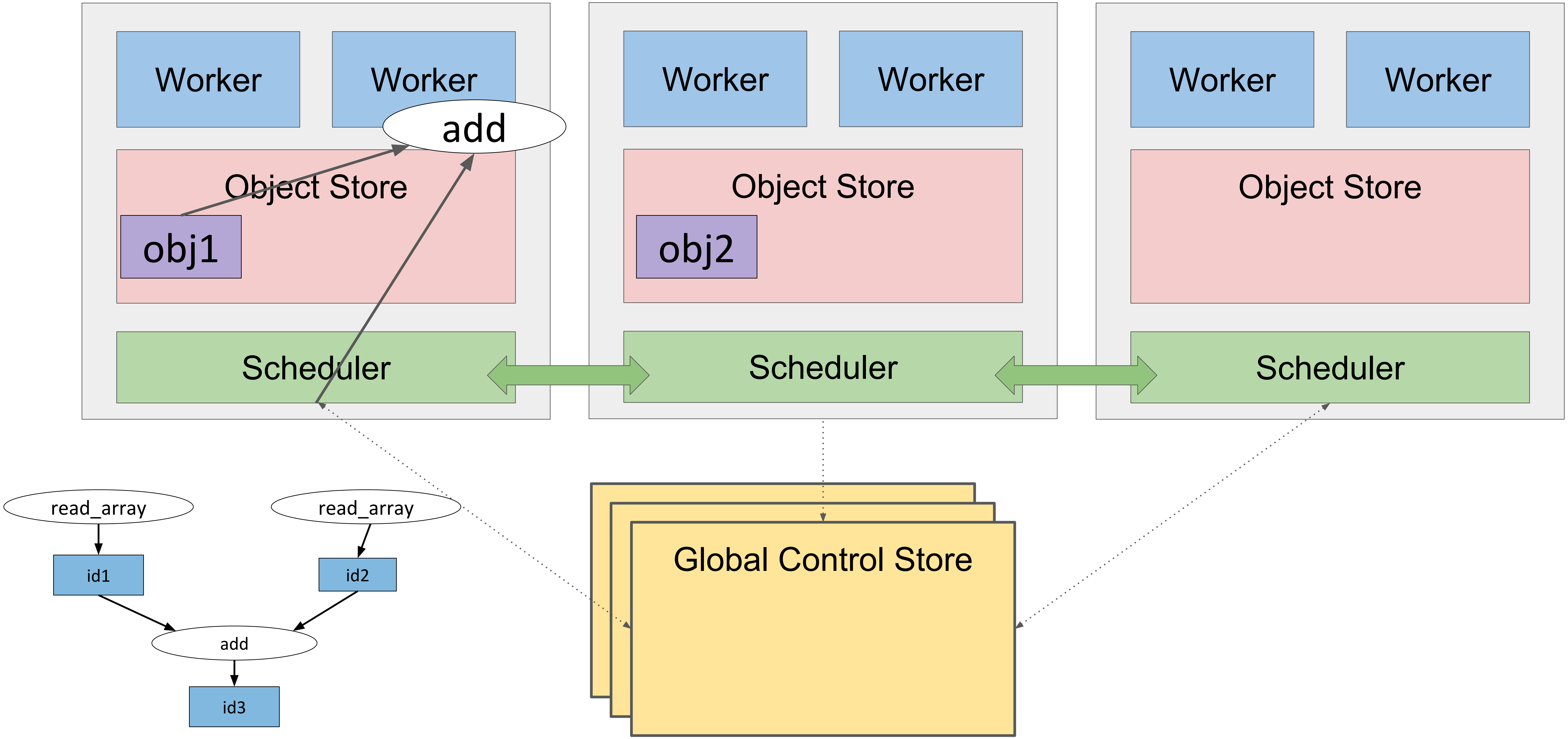
The Ray Architecture



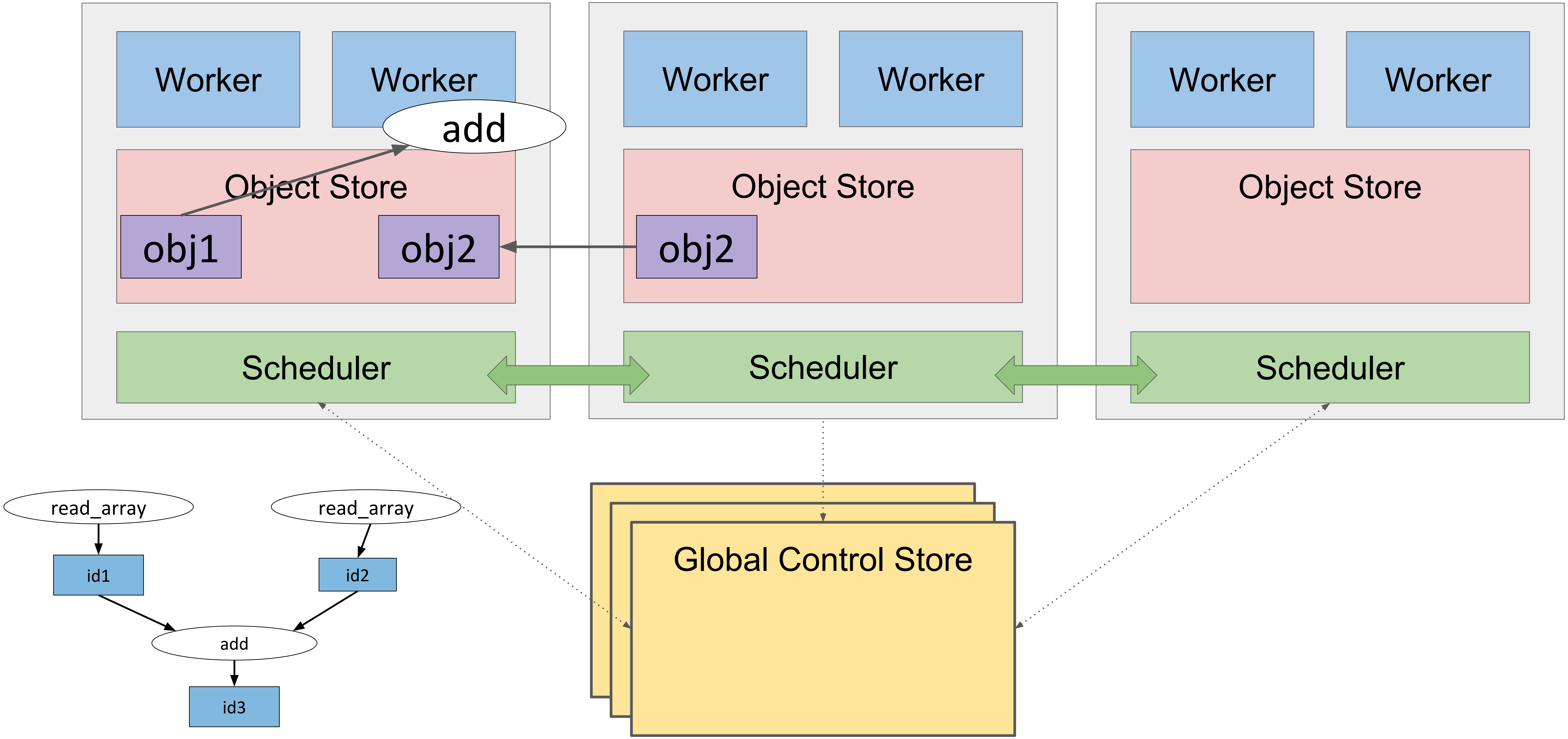
The Ray Architecture



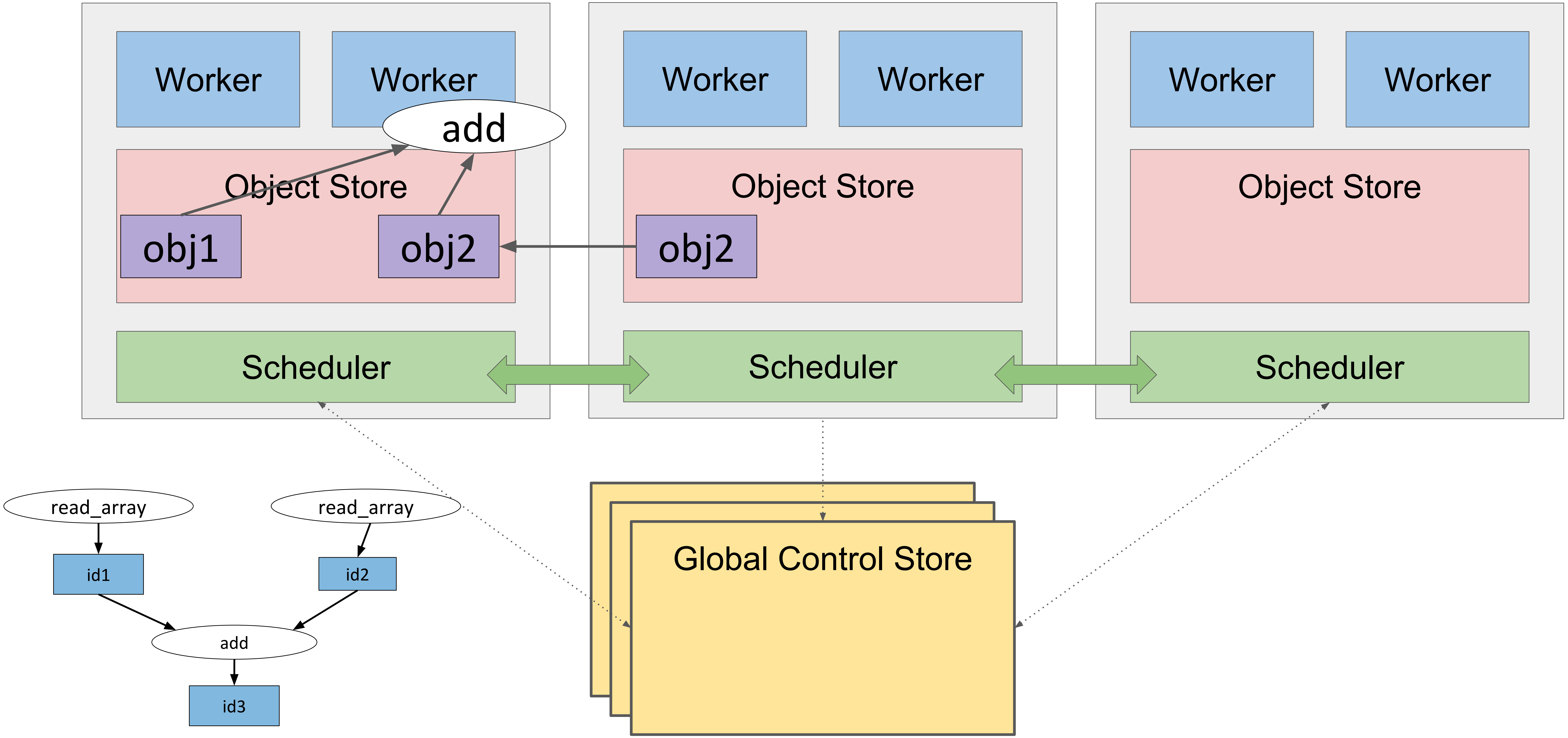
The Ray Architecture



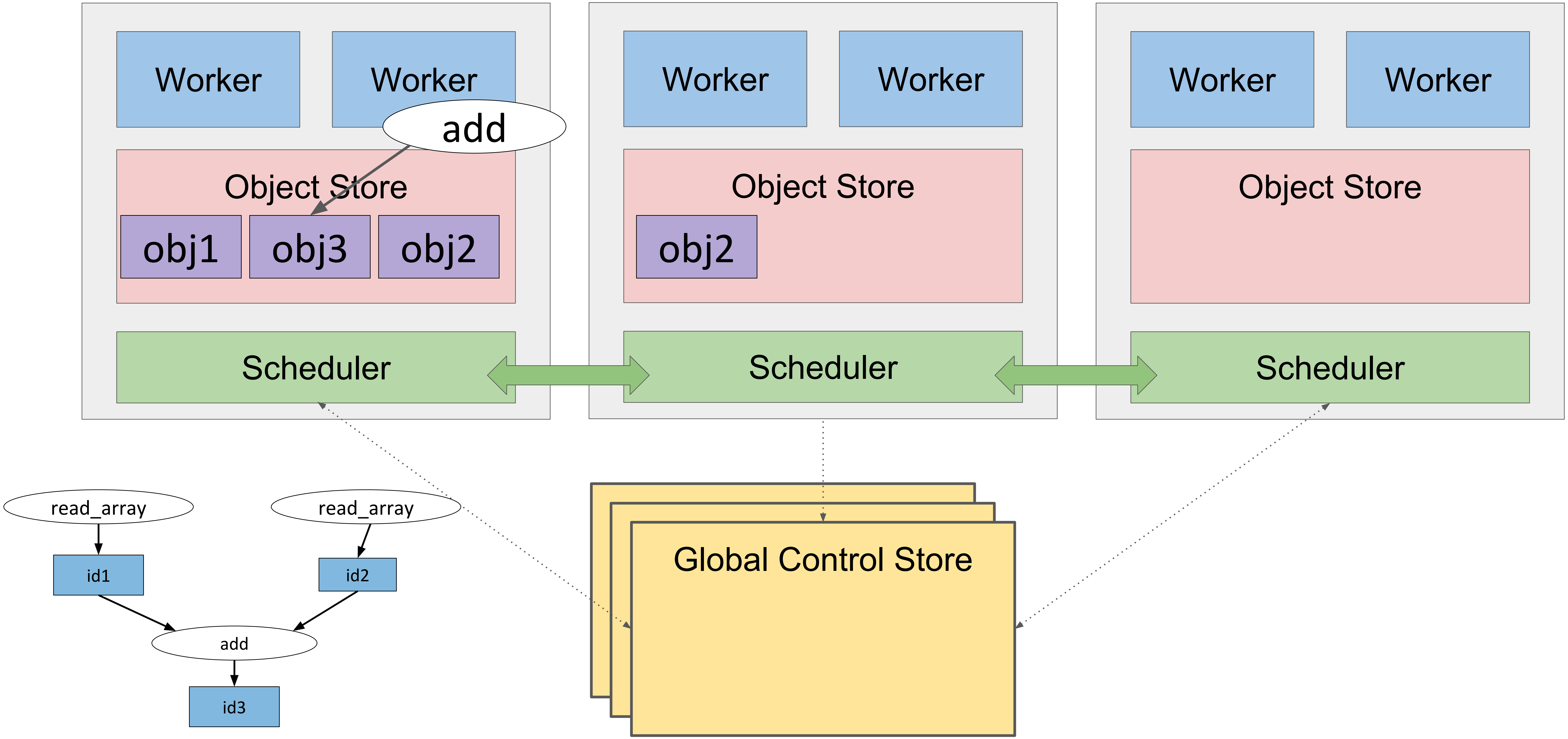
The Ray Architecture



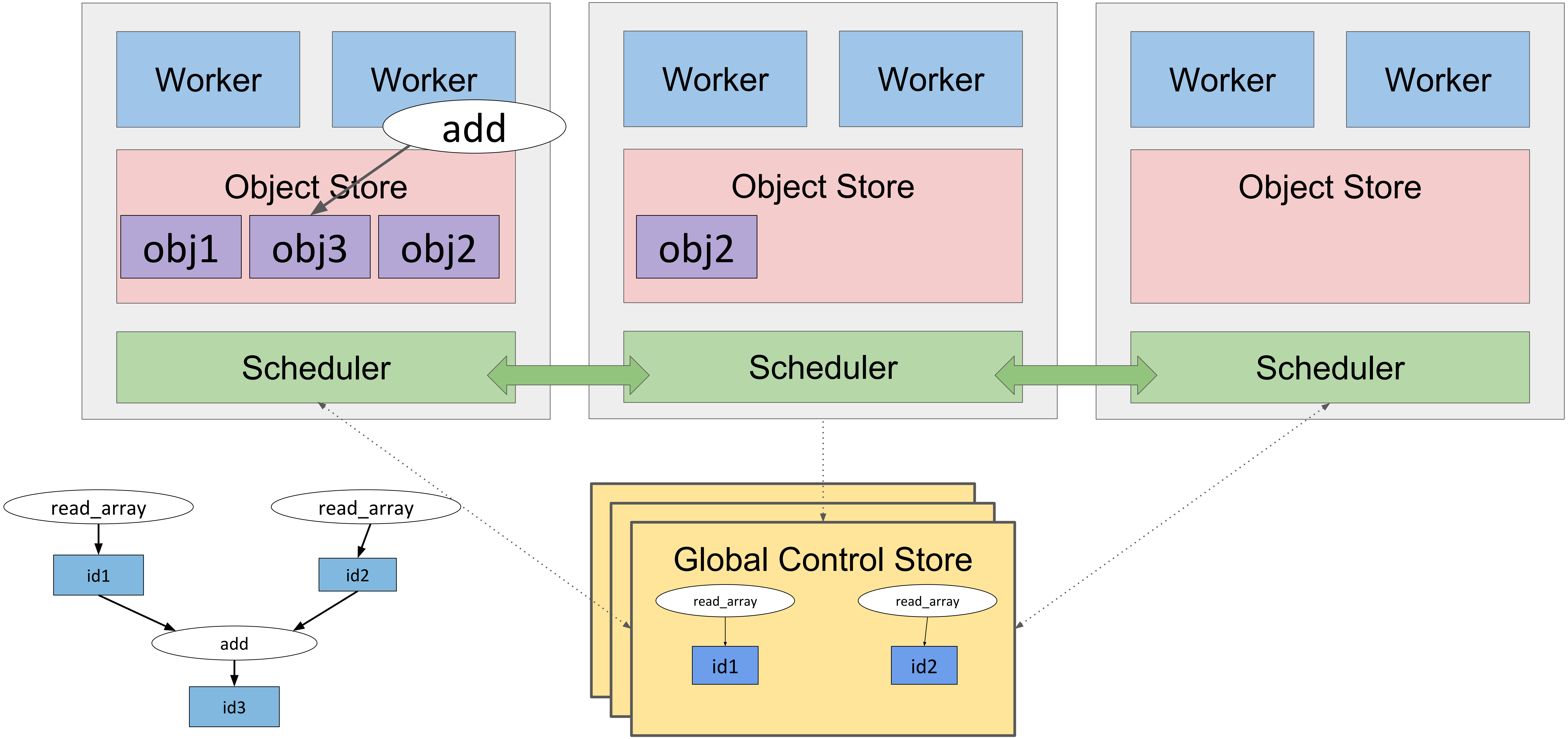
The Ray Architecture



The Ray Architecture



The Ray Architecture



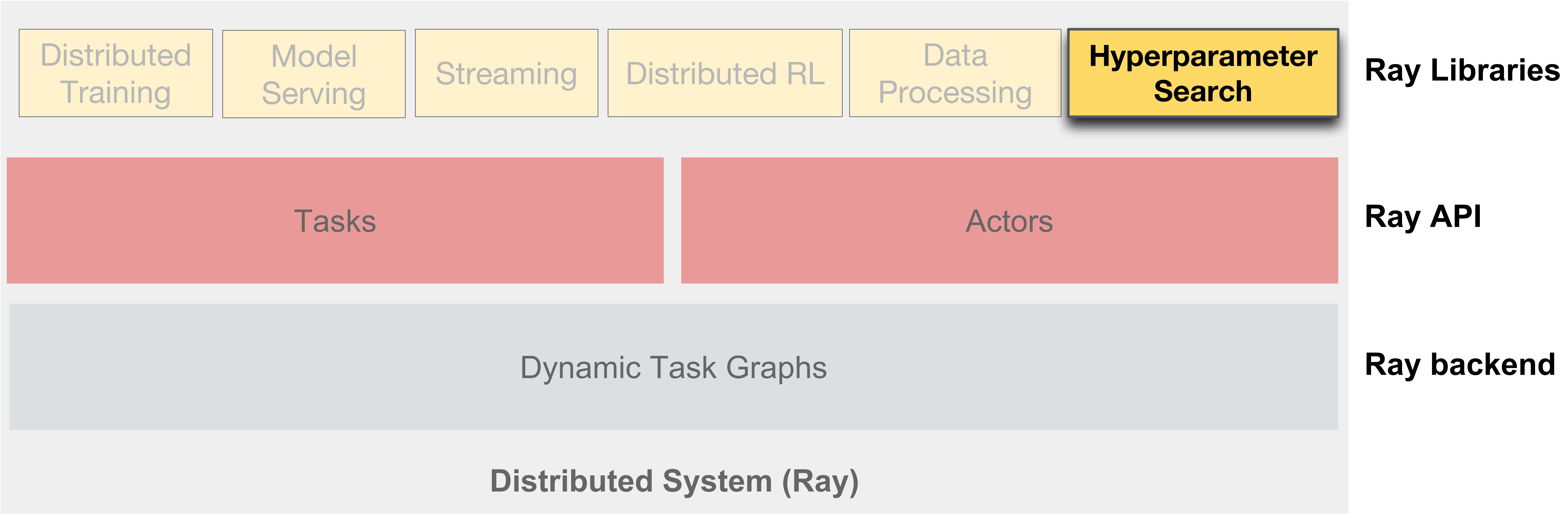
tune

Distributed Hyperparameter Search on Ray

riselab
UC Berkeley

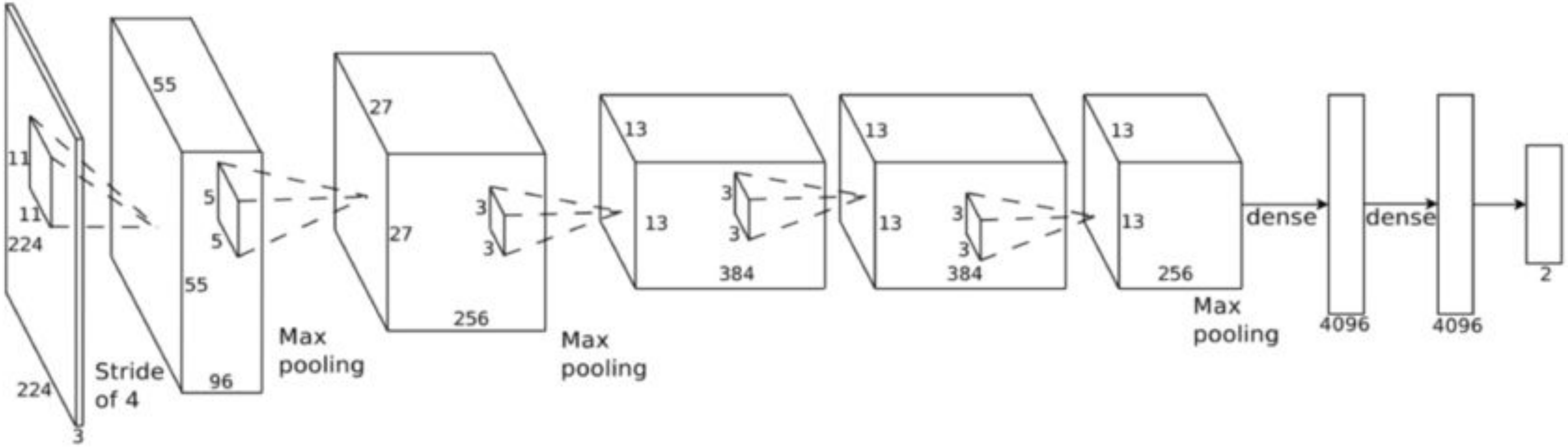
ray.readthedocs.io/en/latest/tune.html

What is Tune?


Distributed Training
Model Serving
Streaming
Distributed RL
Data Processing
Hyperparameter Search
Ray Libraries
Tasks
Actors
Ray API
Dynamic Task Graphs
Ray backend
Distributed System (Ray)

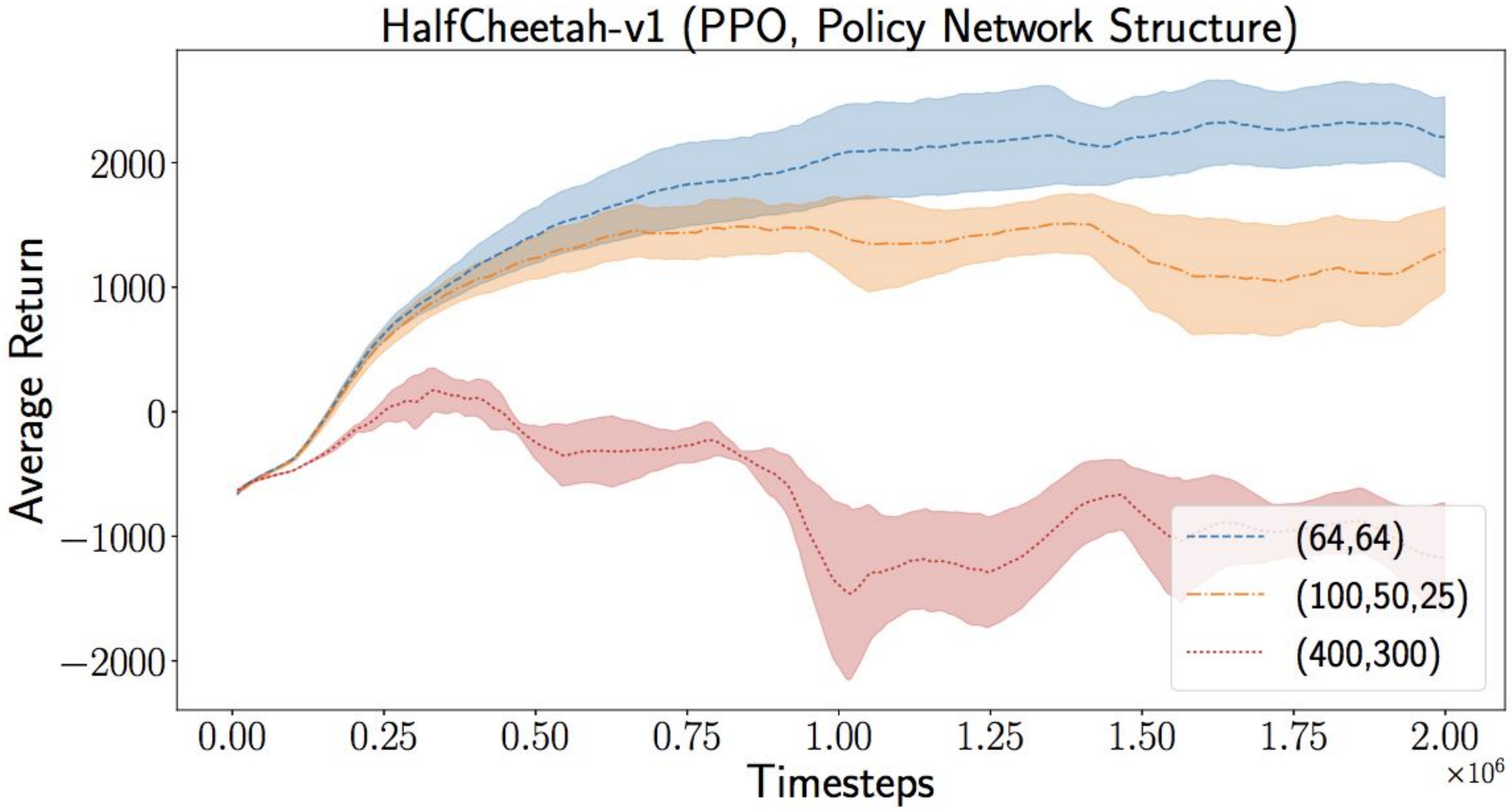
Hyperparameters?

©2017 RISELab



©2017 RISELab

Are hyperparameters actually that important?



Why a framework for tuning hyperparameters?

We want the best model

Resources are expensive

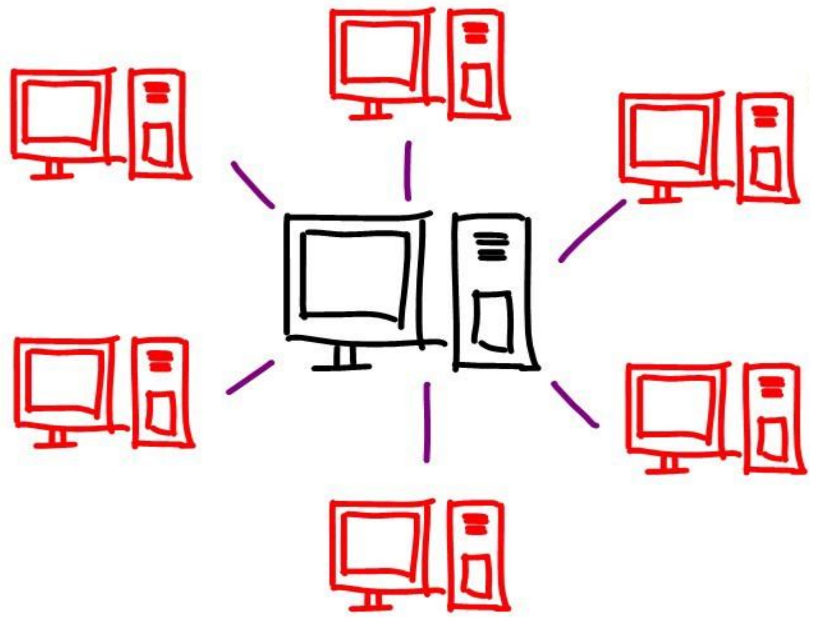
Model training is time-consuming

tune

Tune is built with Deep Learning as a priority.

Resource Aware Scheduling

Seamless Distributed Execution



Simple API for new algorithms

```
class TrialScheduler:
    def on_result(self, trial, result): ...
    def choose_trial_to_run(self): ...
```

Framework Agnostic

ray.readthedocs.io/en/latest/tune.html

Tune is simple to use.

```
# Function-based API
def train():
  for _ in range(N):
    reporter(...)
```

```
# Class-based API
class MyModel(Trainable):
  def _setup(); def _train();
  def _save(); def _restore();
```

Two simple APIs for model training

Tune API

Bayesian Optimization	Grid Search	...	HyperBand	Population Based Training

Pair **search algorithms** and **trial schedulers** to guide your distributed optimization.

ray.readthedocs.io/en/latest/tune.html

Quick Tune API Demo

ray.readthedocs.io/en/latest/tune.html

```python
import ray
import ray.tune as tune

ray.init()
```

```python
def train_func(config):
    model = Model(config)

    ( ... )

    for idx, (data, target) in enumerate(dataset):

        ( ... )

        accuracy = model.train(data, target)
```

ray.readthedocs.io/en/latest/tune.html

```python
import ray
import ray.tune as tune

ray.init()
```

```python
def train_func(config, reporter):  # add a reporter arg
    model = Model(config)

    ( ... )

    for idx, (data, target) in enumerate(dataset):

        ( ... )

        accuracy = model.train(data, target)
        reporter(timesteps_total=idx, mean_accuracy=accuracy) # report metrics
```

ray.readthedocs.io/en/latest/tune.html

```python
def train_func(config, reporter):  # add a reporter arg
    model = Model(config)

    ( ... )

    for idx, (data, target) in enumerate(dataset):

        ( ... )

        accuracy = model.train(data, target)
        reporter(timesteps_total=idx, mean_accuracy=accuracy) # report metrics
```

```python
all_trials = tune.run_experiments({
    "my_experiment": {
        "run": train_func,



    }
})
```

ray.readthedocs.io/en/latest/tune.html

```
run_experiments({
    "my_experiment_name": {
        "run": "my_func",

        "stop": { "mean_accuracy": 100 },
        "config": {
            "alpha": grid_search([0.2, 0.4, 0.6]),
            "beta": grid_search([1, 2]),
        },



    }
}
```

ray.readthedocs.io/en/latest/tune.html

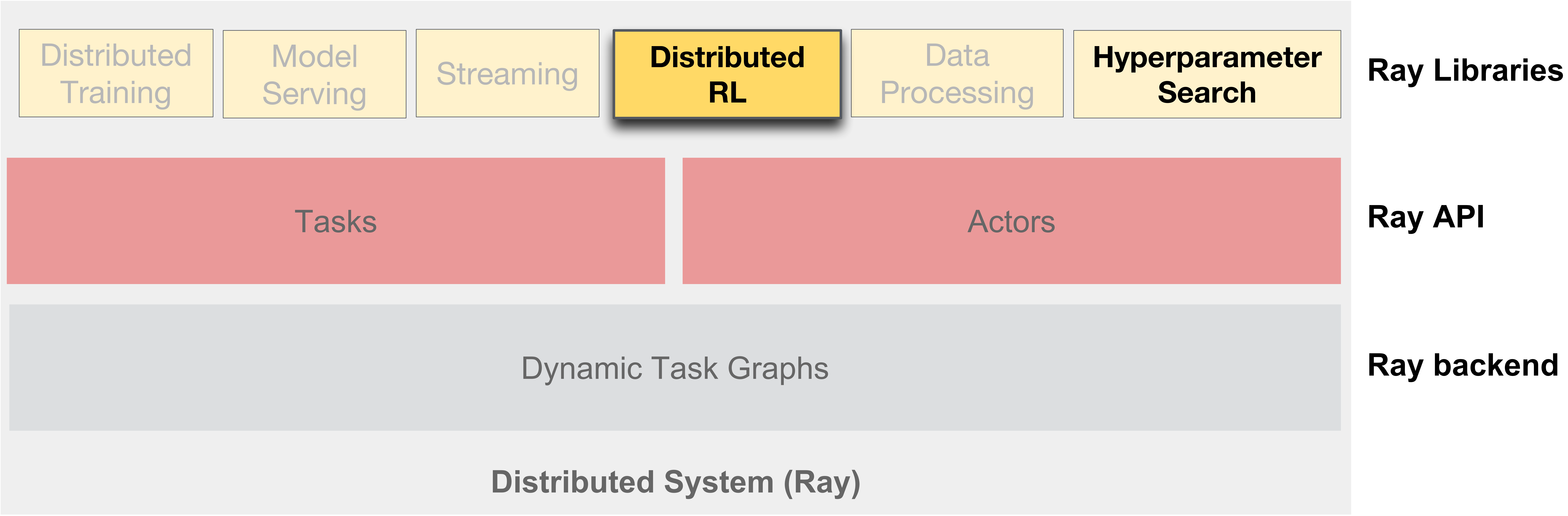
RLlib

A scalable and unified library for reinforcement learning

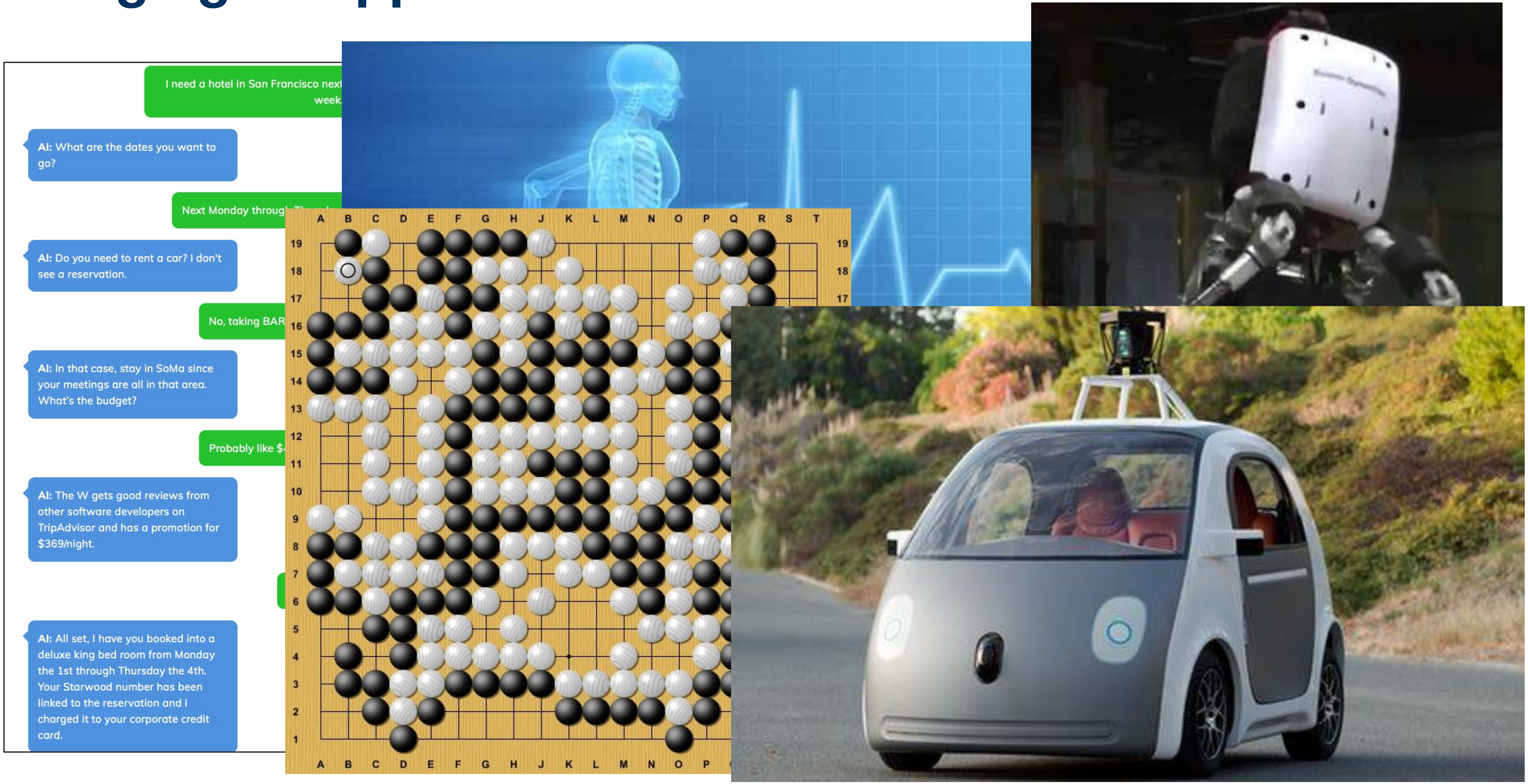
https://rllib.io

risel**ab**
UC Berkeley

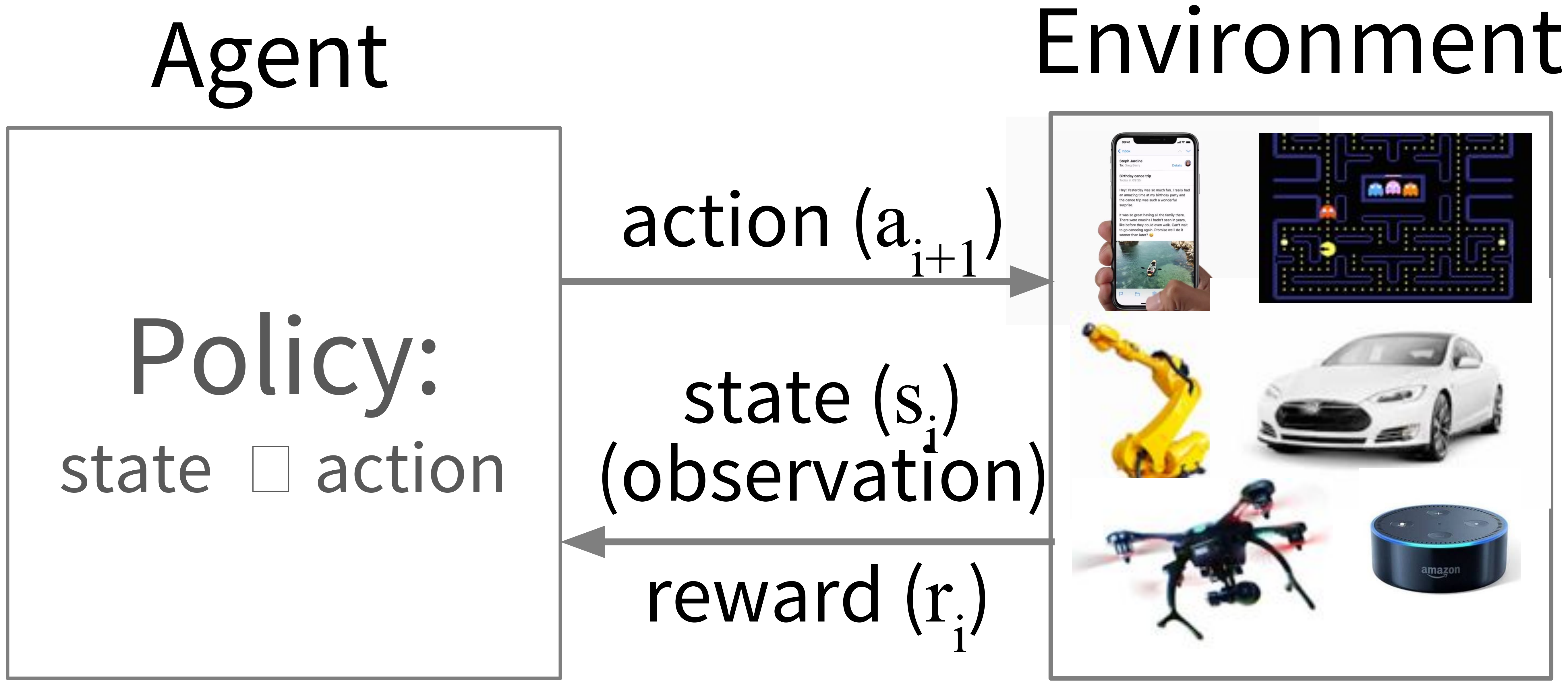
What is RLlib?



Emerging AI Applications



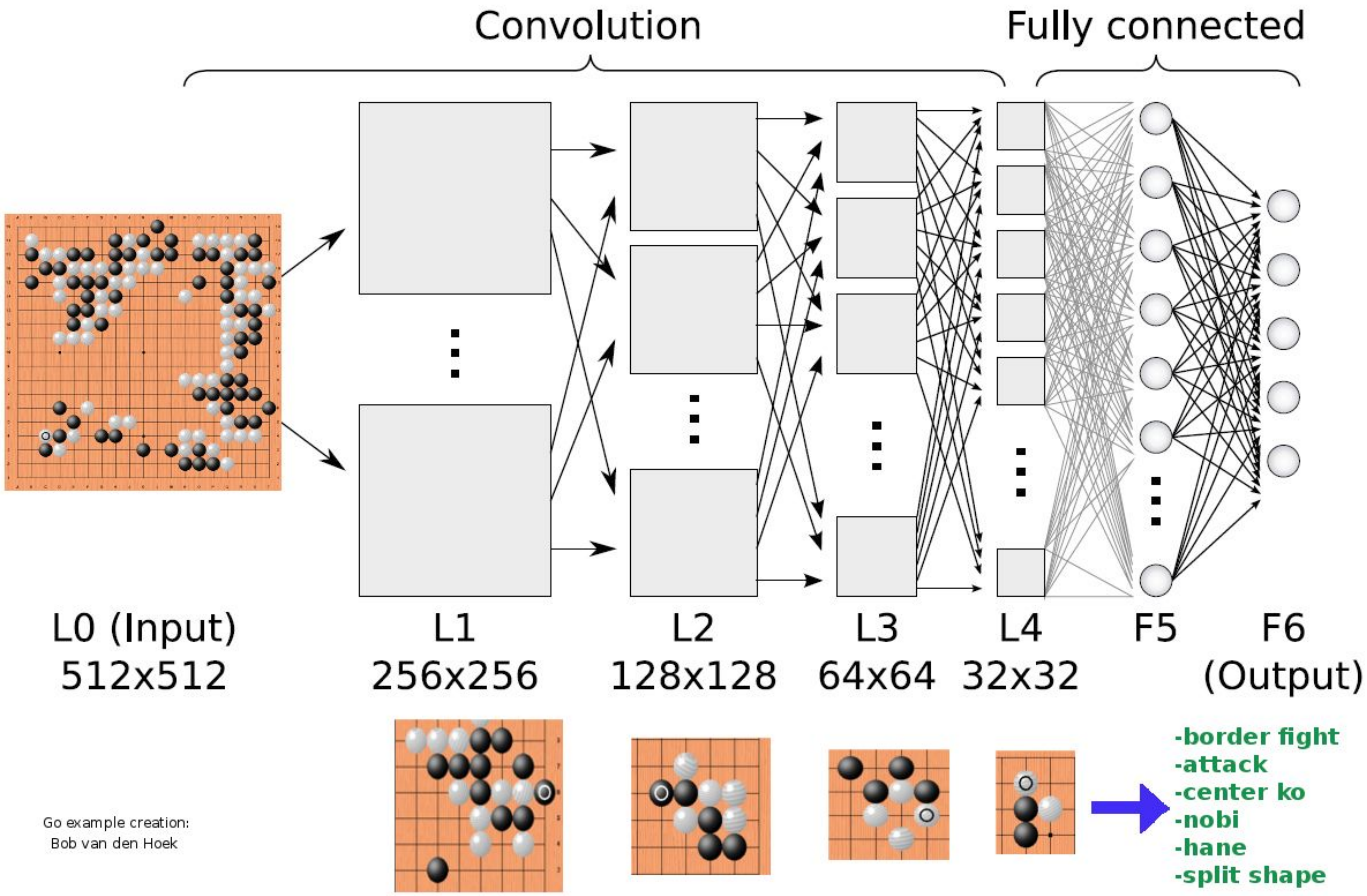
Reinforcement Learning



Applications of Reinforcement Learning

AlphaGo (2016)

- Observations:
  – board state
- Actions:
  – where to place stone
- Rewards:
  – win / lose

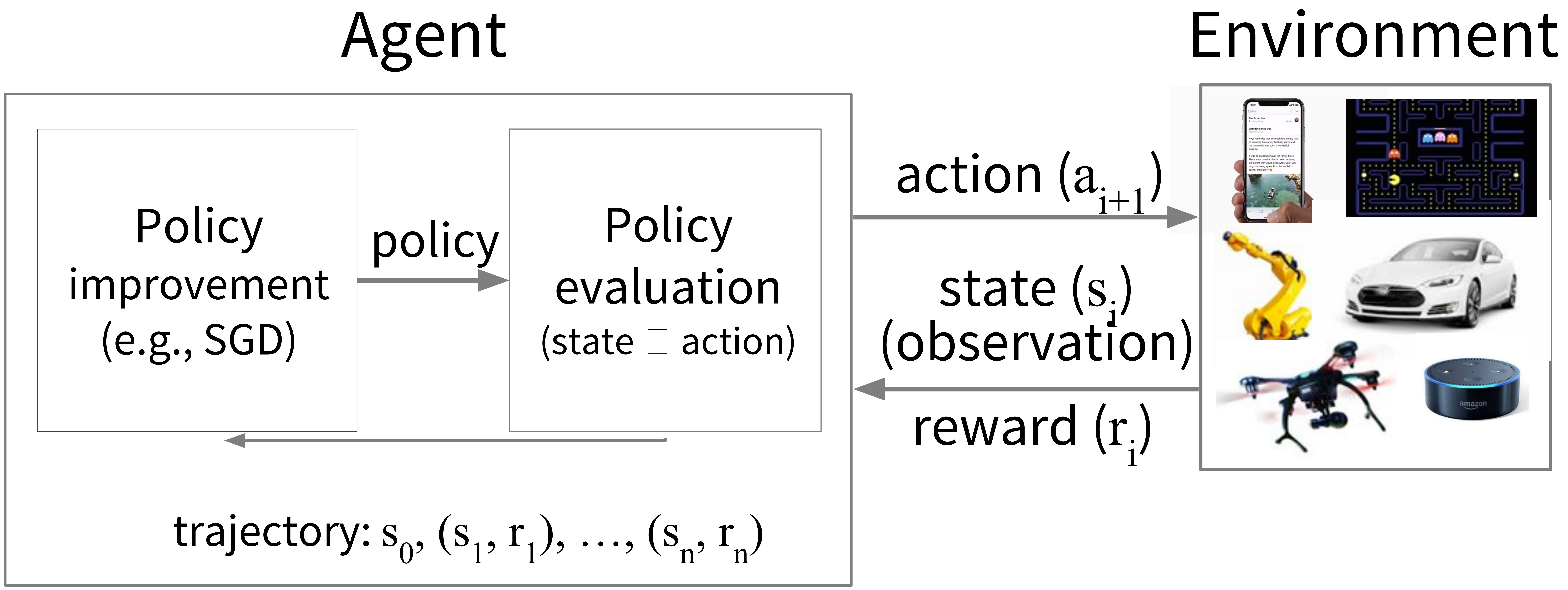


Applications of Reinforcement Learning

Antenna tilt control (research)

- Observations:
  - positions of users
  - user signal strength
- Actions:
  - antenna tilt adjustment
- Rewards:
  - network throughput

Reinforcement Learning



What is Reinforcement Learning?

- Learn which actions are best to take using feedback

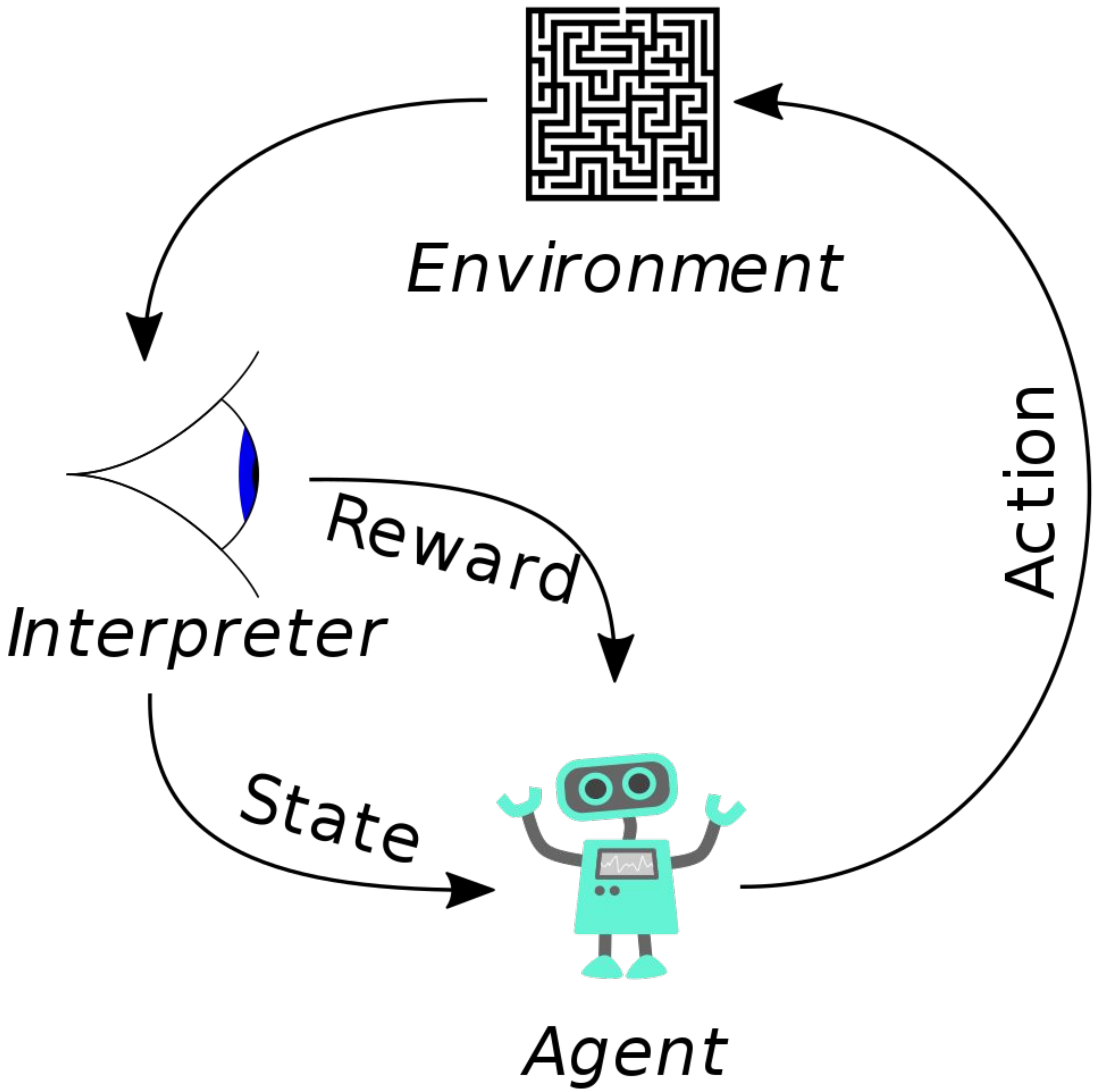


Image from Wikipedia

What is Reinforcement Learning?

- Learn which actions are best to take using feedback
- Agent takes an action based on state
    - Put hand in fire

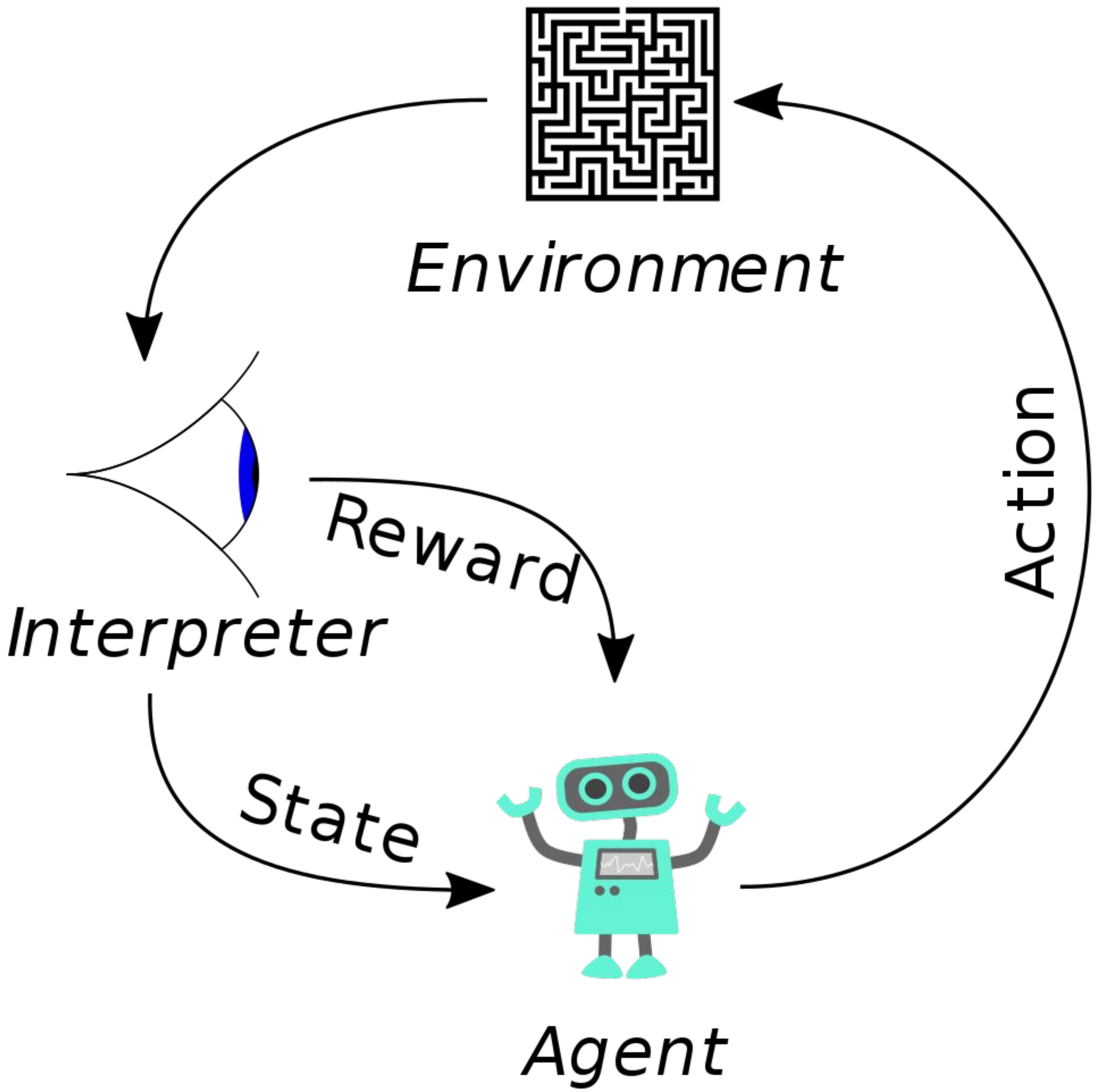


Image from Wikipedia

What is Reinforcement Learning?

- Learn which actions are best to take using feedback
- Agent takes an action based on state
  - Put hand in fire
- Actions change the environment
  - Hand in new location
  - Heat travels to my hand

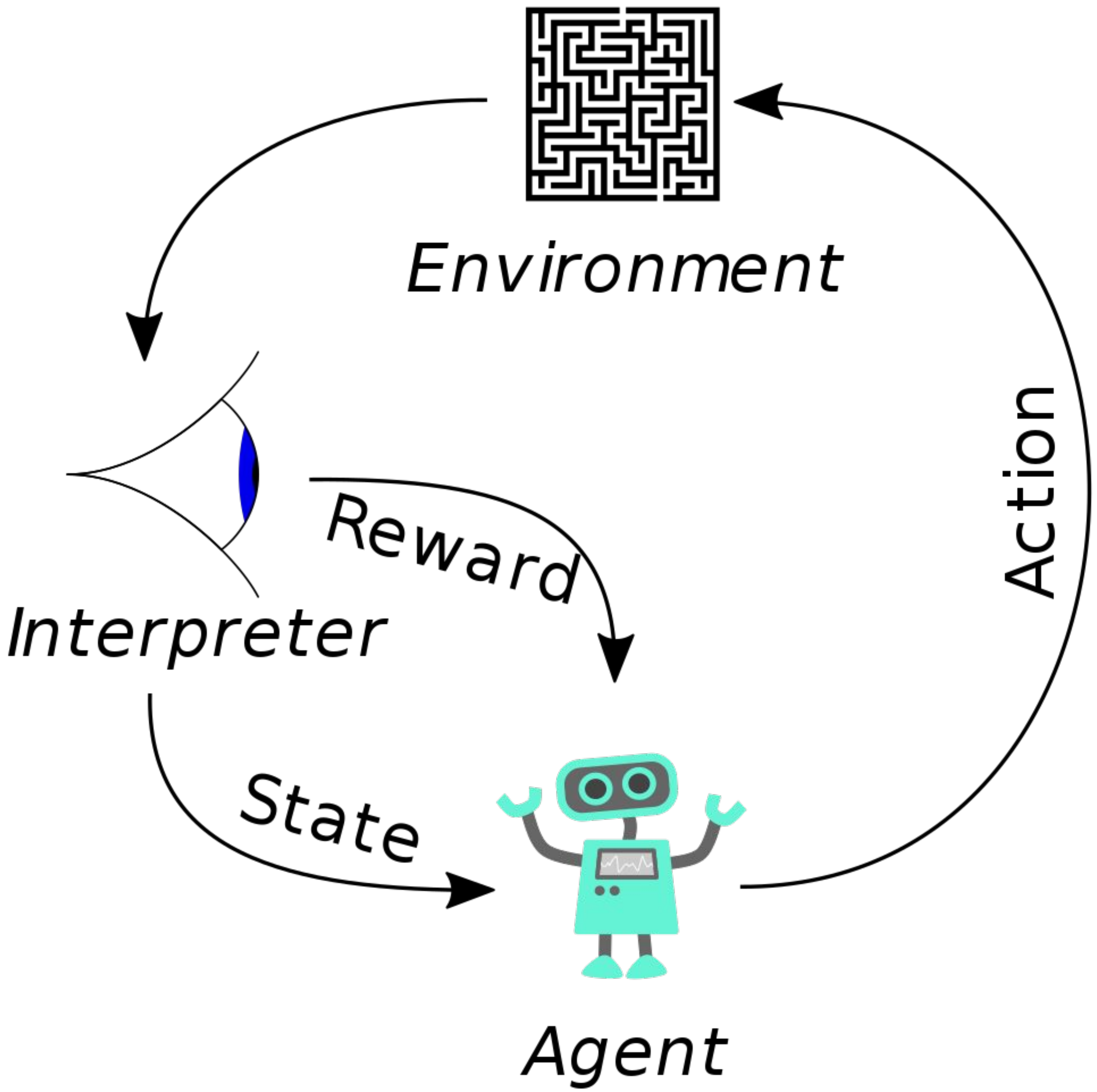


Image from Wikipedia

What is Reinforcement Learning?

- Learn which actions are best to take using feedback
- Agent takes an action based on state
  - Put hand in fire
- Actions change the environment
  - Hand in new location
  - Heat travels to my hand
- Agent observes new state of environment
  - “My hand is hot”
  - Pain -> low reward

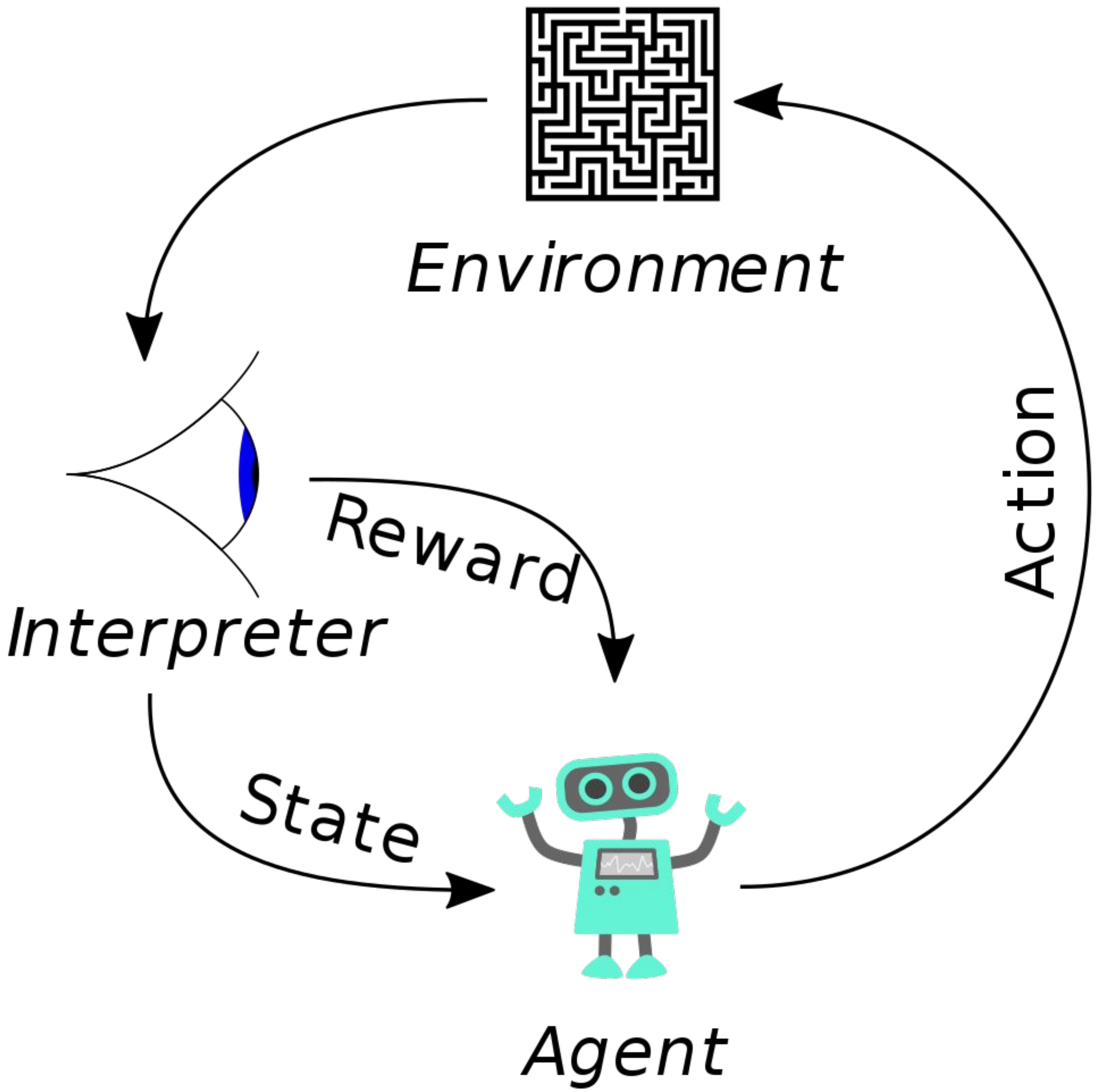


Image from Wikipedia

What is Reinforcement Learning?

- Learn which actions are best to take using feedback
- Agent takes an action based on state
    - Put hand in fire
- Actions change the environment
    - Hand in new location
    - Heat travels to my hand
- Agent observes new state of environment
    - “My hand is hot”
    - Pain -> low reward
- Agent uses reward to update its policy
    - “Don’t put hand in fire”

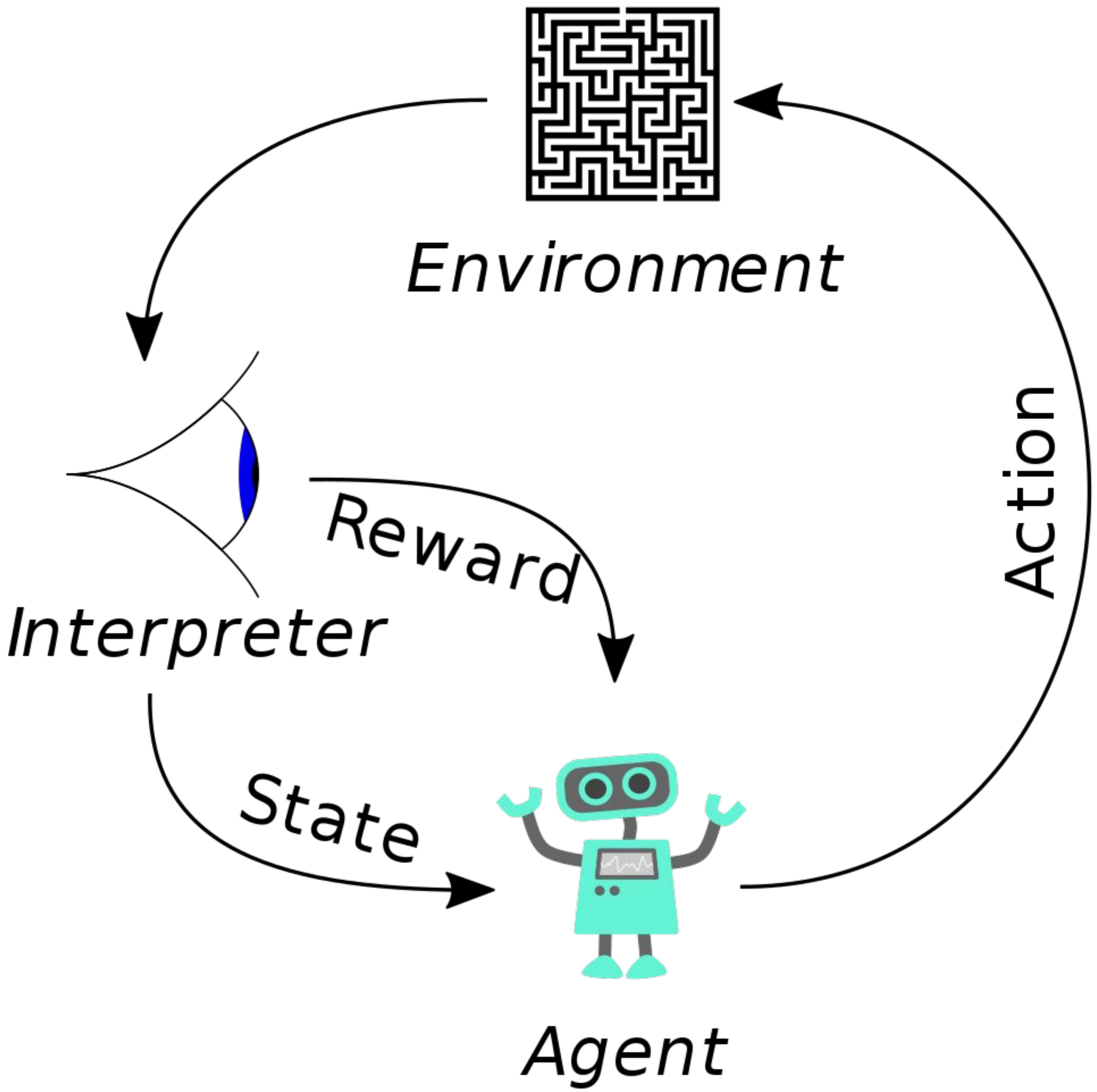
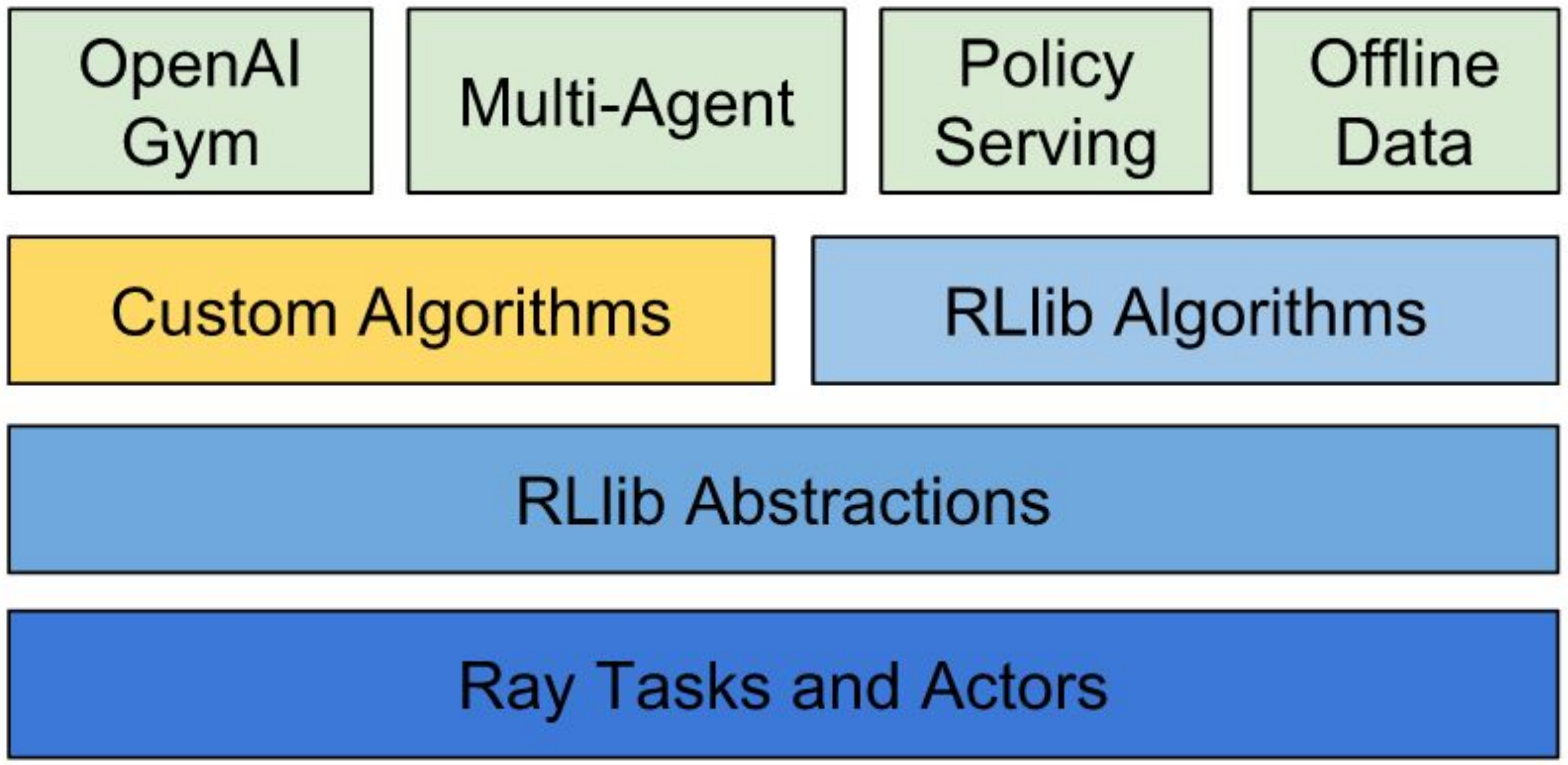


Image from Wikipedia

RLlib: A Unified Library for Reinforcement Learning

Three main value adds:



3. APIs that make algorithms accessible to a variety of applications

2. Collection of scalable reference algorithms

1. Abstractions for implementing distributed RL algorithms (ICML '18)

Reference Algorithms

- **High-throughput architectures**
  - Distributed Prioritized Experience Replay (Ape-X)
  - Importance Weighted Actor-Learner Architecture (IMPALA)
- **Gradient-based**
  - Advantage Actor-Critic (A2C, A3C)
  - Deep Deterministic Policy Gradients (DDPG, TD3)
  - Deep Q Networks (DQN, Rainbow)
  - Policy Gradients
  - Proximal Policy Optimization (PPO)
- **Derivative-free**
  - Augmented Random Search (ARS)
  - Evolution Strategies

Community Contributions

MOBILE SENSING LAB
UNIVERSITY OF CALIFORNIA, BERKELEY

Alibaba Group

APIs

- Stable public APIs (see rllib.io)
- Custom environments
  - OpenAI gym
  - Vectorized
  - Multi-agent
  - External simulators
  - *
- Custom policy network models
  - Recurrent policies
  - Complex observation spaces (dict / tuple spaces)
  - Parametric action spaces (variable-length / infinite space of actions)
- Custom policy losses / algorithms
- Also can "drop down to Ray"

+ Multi-GPU PPO / IMPALA

APIs

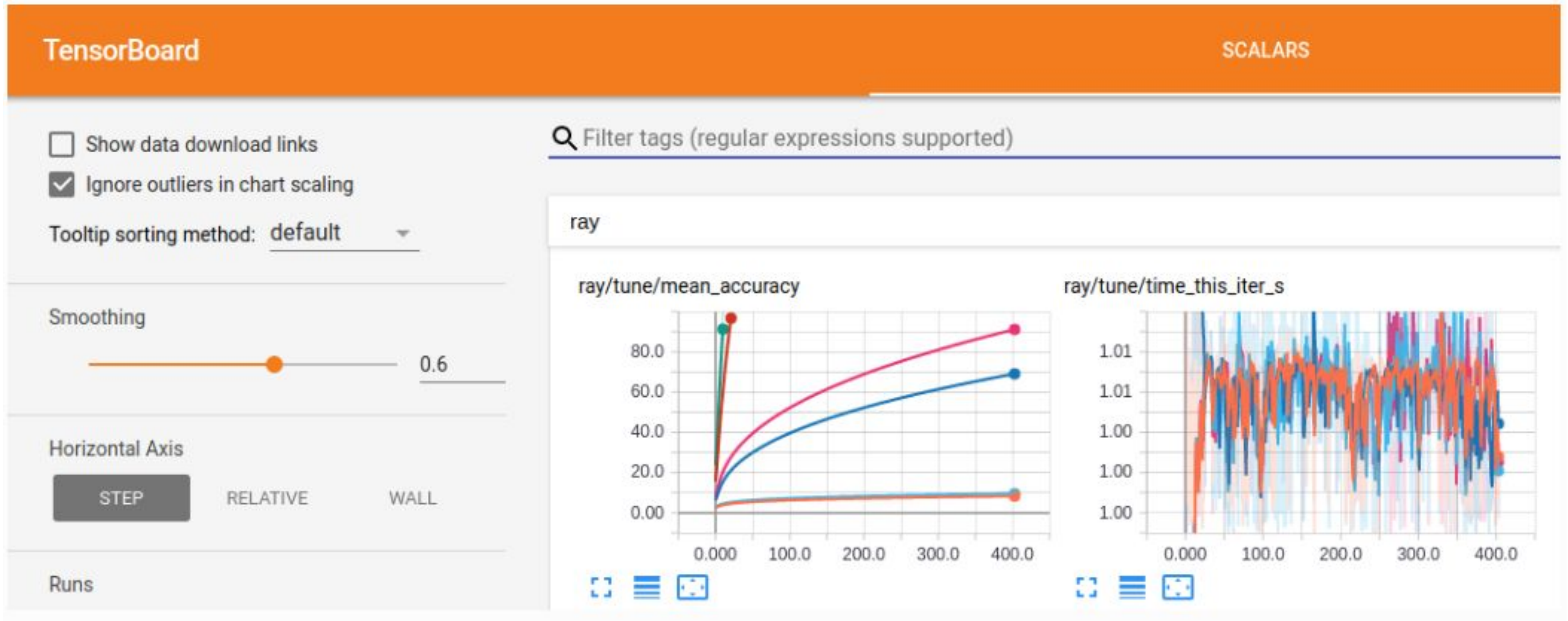
- Integration with Tune

```
import ray
import ray.tune as tune

ray.init()
tune.run_experiments({
    "my_experiment": {
        "run": "PPO",
        "env": "CartPole-v0",
        "stop": {"episode_reward_mean": 200},
        "config": {
            "num_gpus": 0,
            "num_workers": 1,
            "sgd_stepsize": tune.grid_search([0.01, 0.001, 0.0001]),
        },
    },
})
```

APIs

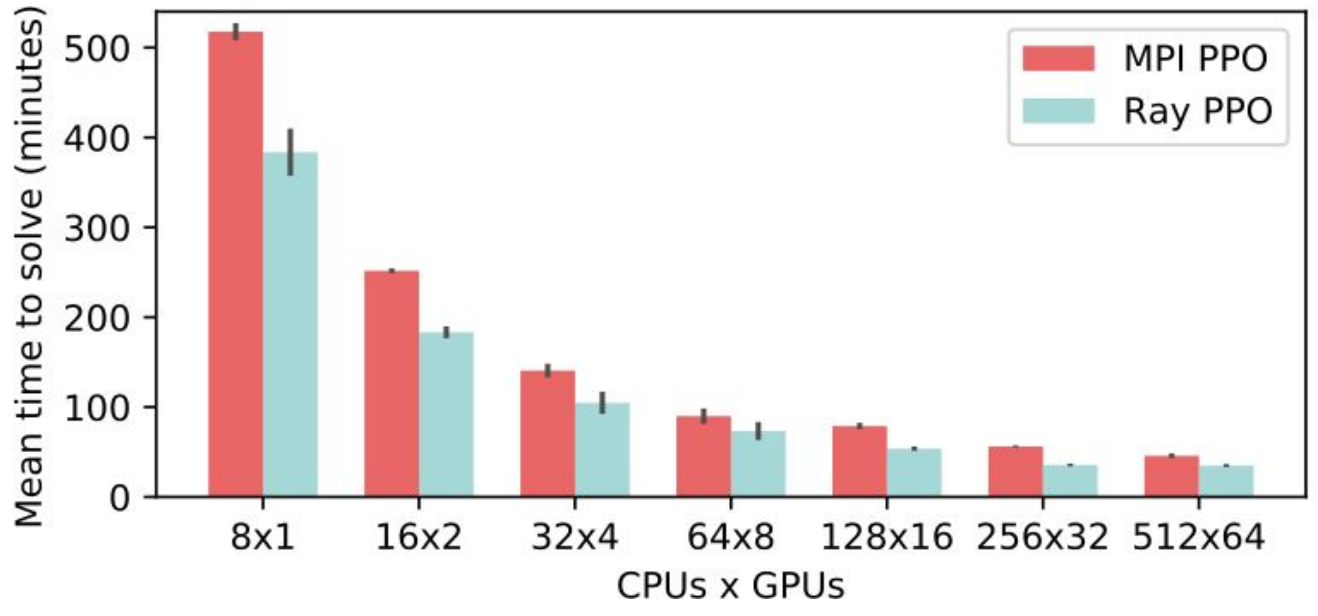
- Integration with Tune

```
== Status ==
Using FIFO scheduling algorithm.
Resources requested: 4/4 CPUs, 0/0 GPUs
Result logdir: ~/ray_results/my_experiment
PENDING trials:
 - PPO_CartPole-v0_2_sgd_stepsize=0.0001:   PENDING
RUNNING trials:
 - PPO_CartPole-v0_0_sgd_stepsize=0.01:     RUNNING [pid=21940], 16 s, 4013 ts, 22 rew
 - PPO_CartPole-v0_1_sgd_stepsize=0.001:    RUNNING [pid=21942], 27 s, 8111 ts, 54.7 rew
```

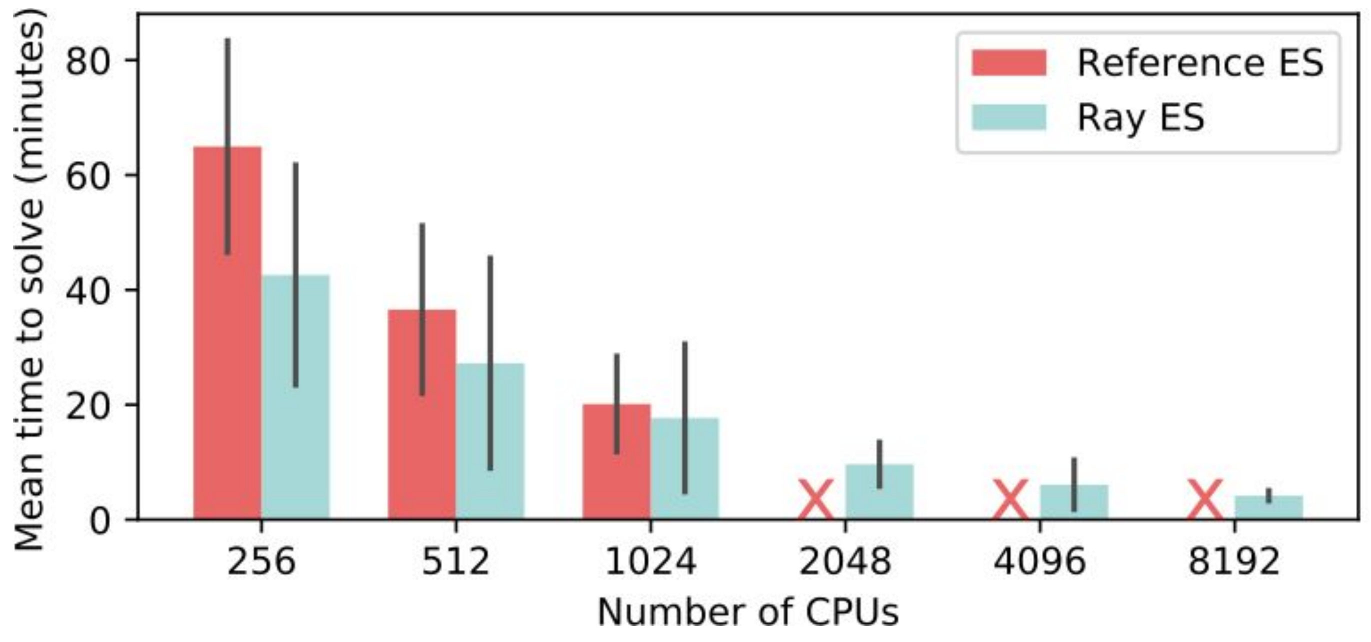


Performance

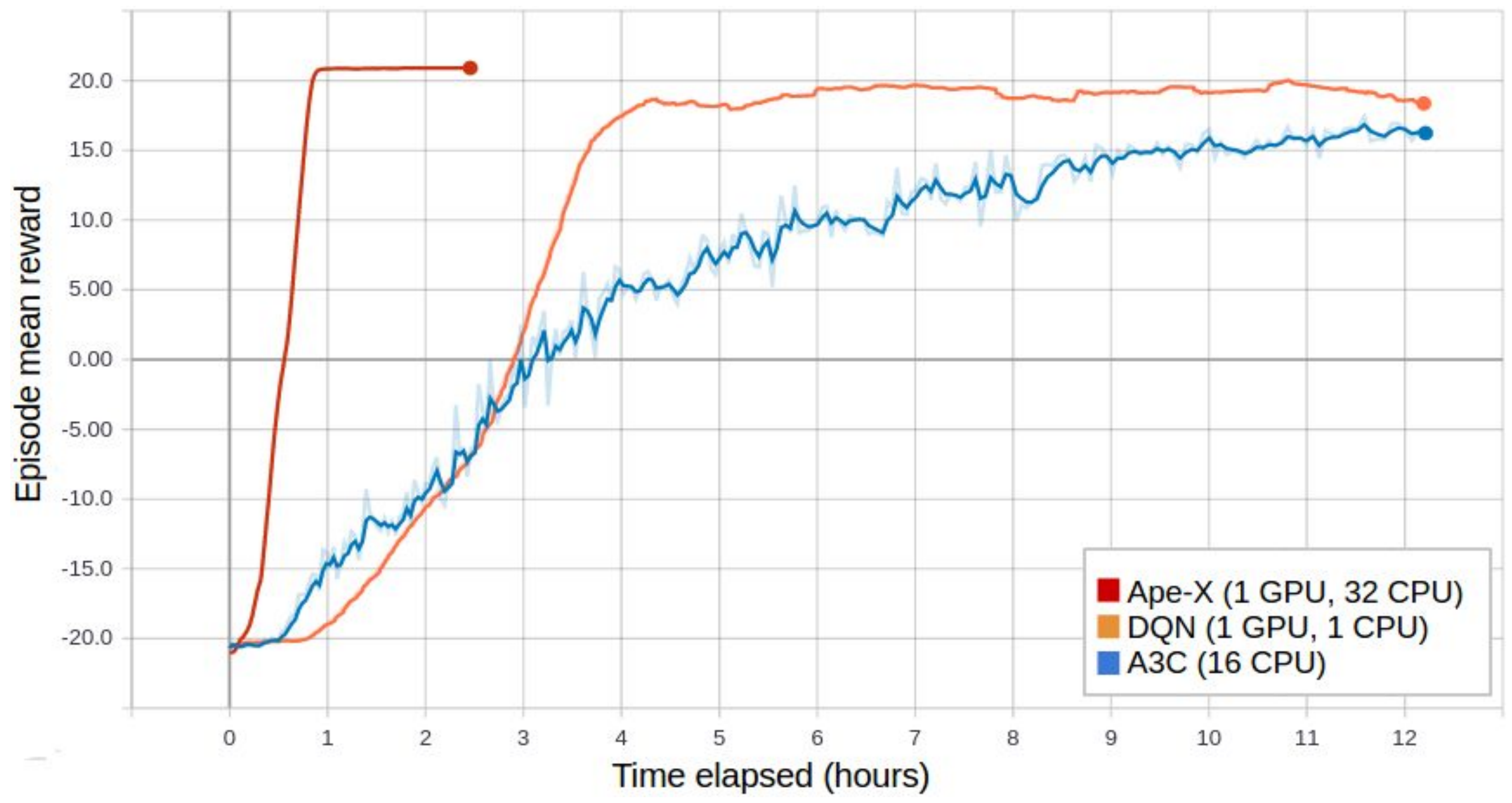
Distributed PPO
(vs OpenMPI)



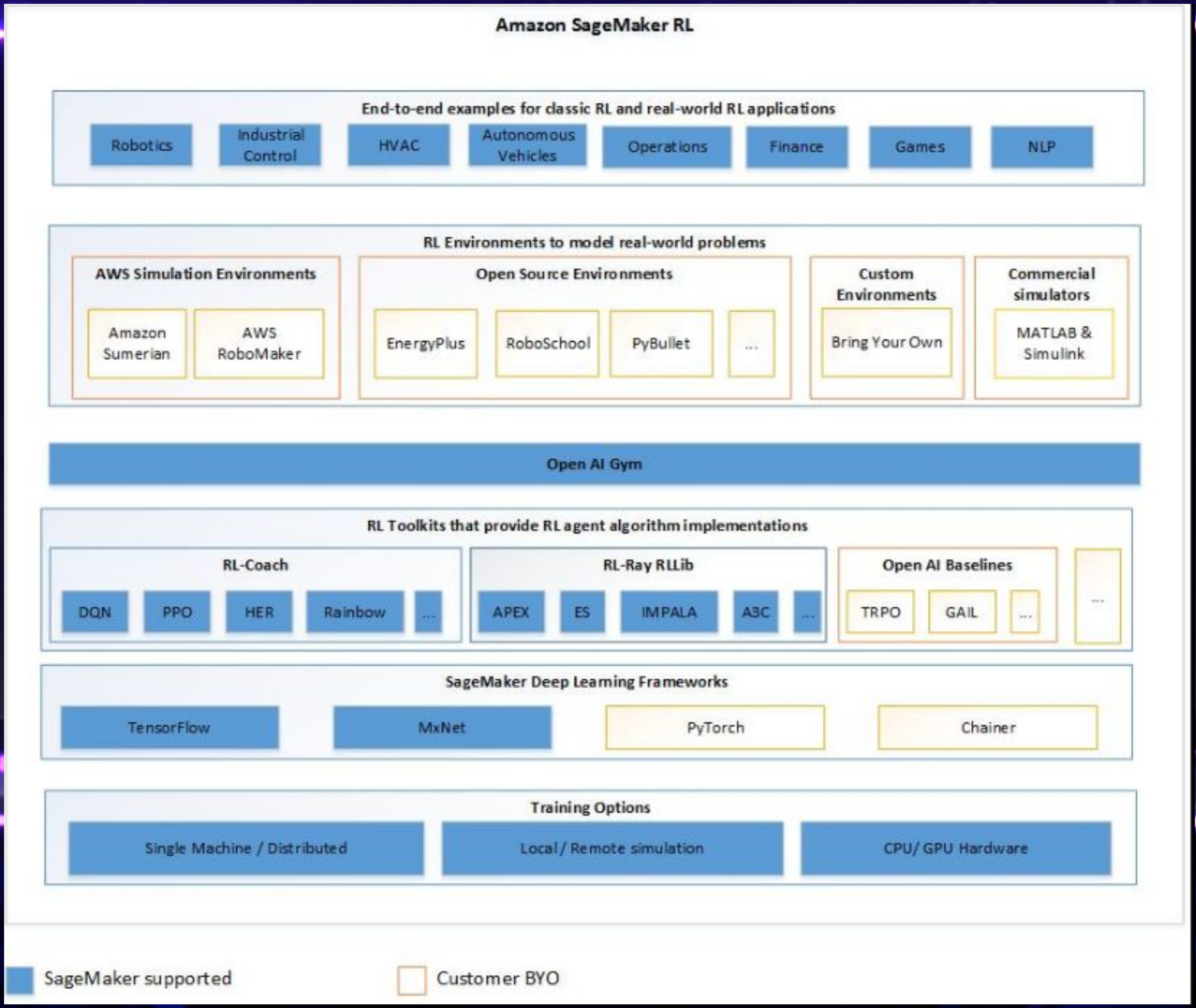
Evolution Strategies
(vs Redis-based)



Ape-X Distributed DQN, DDPG




Amazon
SageMaker RL
Reinforcement learning for every
develope nd data scientist


Amazon SageMaker RL
End-to-end examples for classic RL and real-world RL applications
Robotics
Industrial Control
HVAC
Autonomous Vehicles
Operations
Finance
Games
NLP
RL Environments to model real-world problems
AWS Simulation Environments
Amazon Sumerian
AWS RoboMaker
Open Source Environments
EnergyPlus
RoboSchool
PyBullet
...
Custom Environments
Bring Your Own
Commercial simulators
MATLAB & Simulink
Open AI Gym
RL Toolkits that provide RL agent algorithm implementations
RL-Coach
DQN
PPO
HER
Rainbow
...
RL-Ray RLLib
APEX
ES
IMPALA
A3C
...
Open AI Baselines
TRPO
GAIL
...
...
SageMaker Deep Learning Frameworks
TensorFlow
MxNet
PyTorch
Chainer
Training Options
Single Machine / Distributed
Local / Remote simulation
CPU/ GPU Hardware
SageMaker supported
Customer BYO

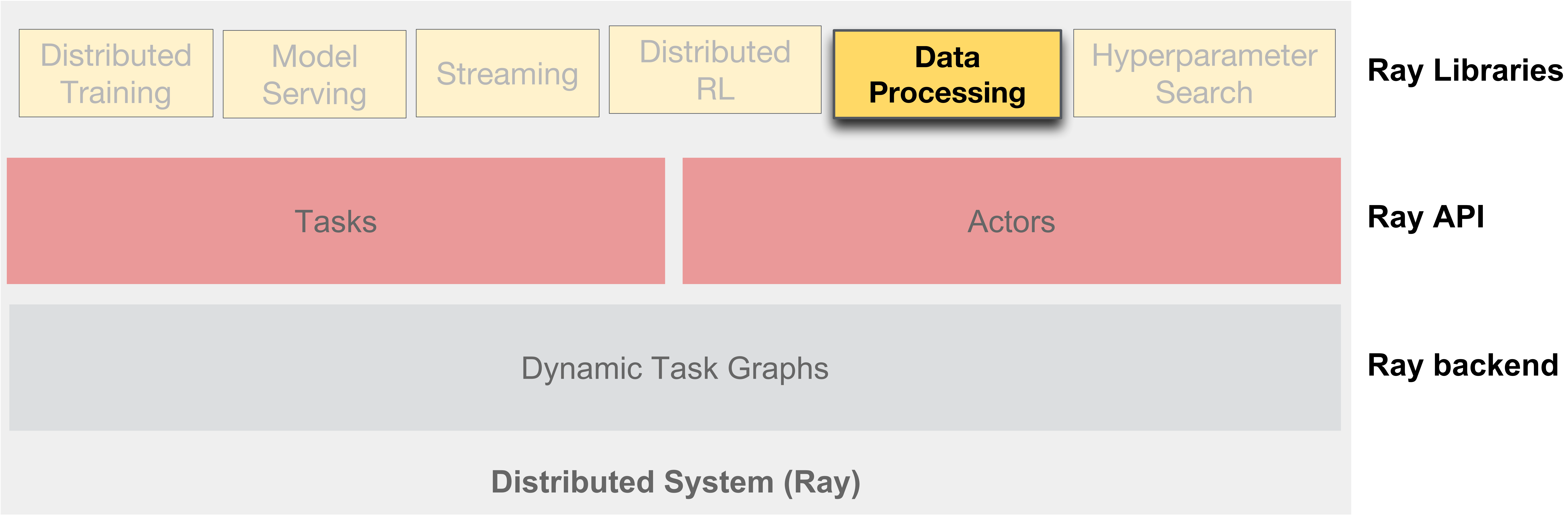
MODIN

Accelerate your Pandas workflows by changing a single line of code

risela b
UC Berkeley

ray.readthedocs.io/en/latest/tune.html

What is Modin?



Modin: Pandas on Ray

Accelerate your pandas workloads by changing one line of code

Tools good at 1MB

Tools good at 1TB+

©2017 RISELab

Modin: Pandas on Ray

Accelerate your pandas workloads by changing one line of code

To use Modin, replace the pandas import:

```
# import pandas as pd
import modin.pandas as pd
```

Installation

Modin can be installed from PyPI:

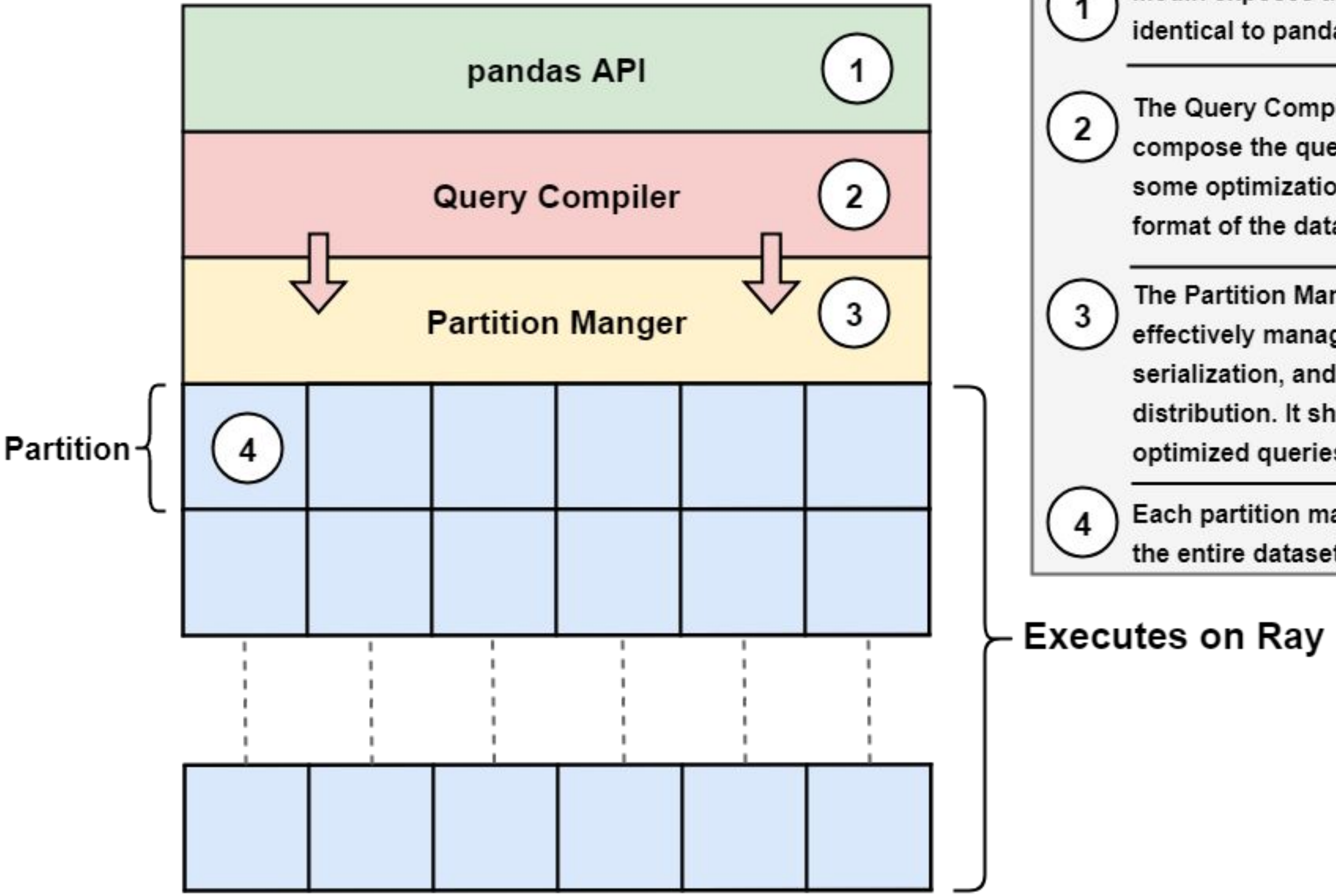
```
pip install modin
```


©2017 RISELab

Why Modin?

- Faster pandas, even on your laptop
  - Up to 4x speed improvement over pandas on 4 physical cores
- Cluster support -- experimental!
- A DataFrame library aimed at bridging the gap between MB-scale and TB-scale data

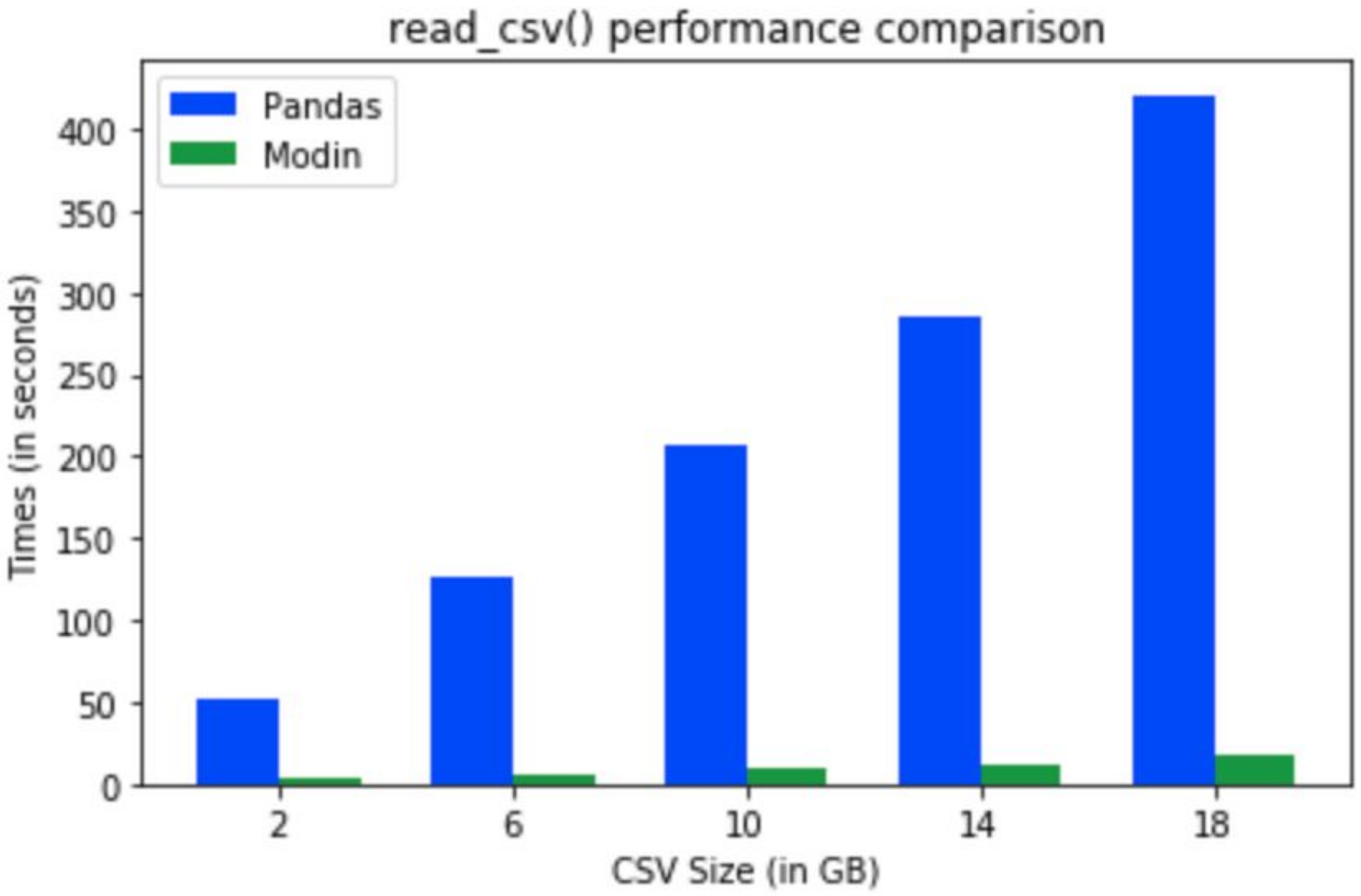
©2017 RISELab



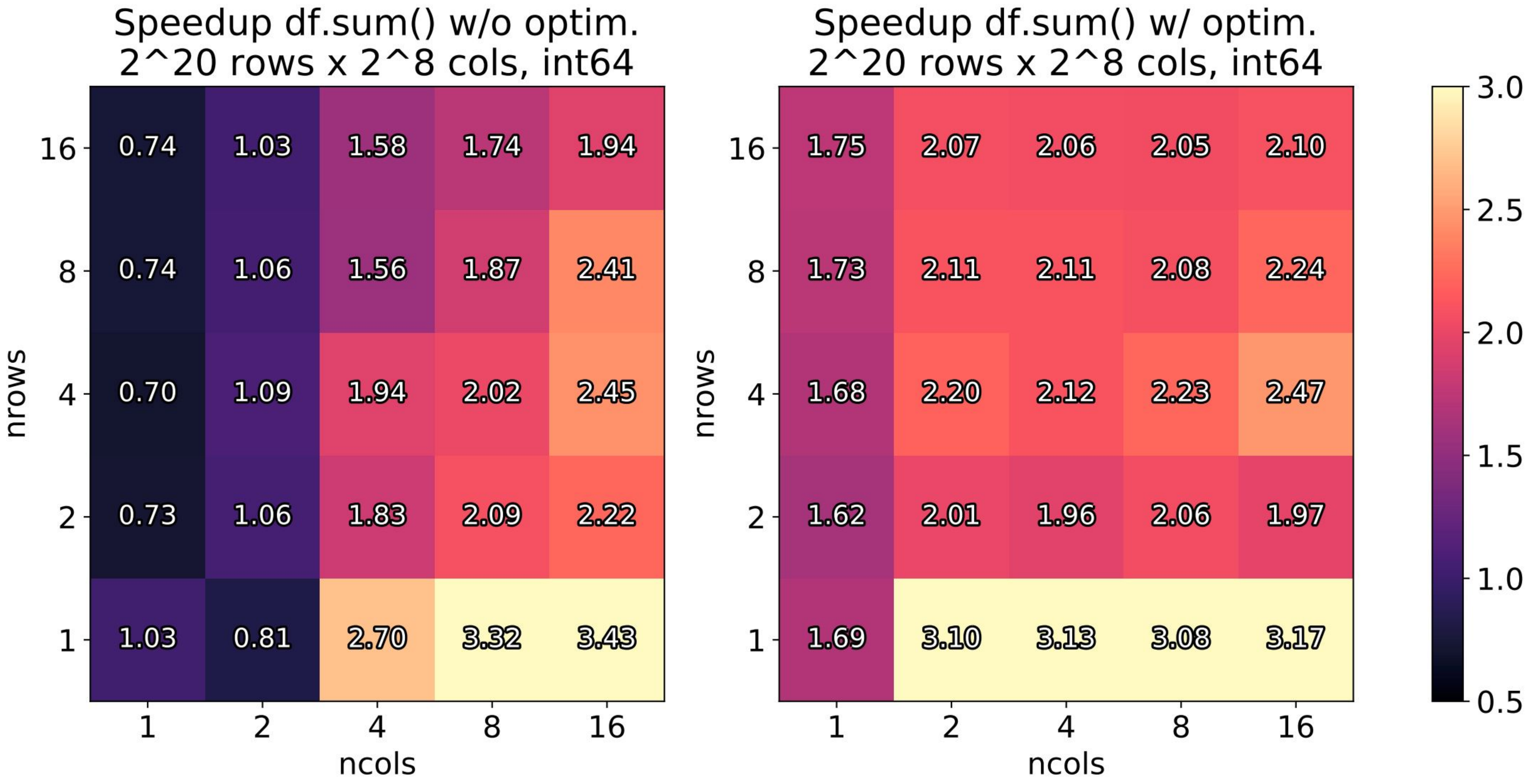
1. Modin exposes an API that is identical to pandas.
2. The Query Compiler layer will compose the query and perform some optimizations based on the format of the data in memory.
3. The Partition Manager layer effectively manages partitioning, serialization, and data distribution. It ships the optimized queries to the data.
4. Each partition maintains a part of the entire dataset.

©2017 RISELab

Performance



Performance



Conclusion

- Ray is an open source project for distributed computing
- **special-purpose** distributed systems -> **general-purpose** distributed system
- Support for the full ML lifecycle (data collection, training, simulation, serving)

RAY

github.com/ray-project/ray

©2017 RISELab