Introduction to Ray & Anyscale

anyscale

Agenda Today

- Introductions
- Overview
    - Intro your instructor
    - Why are we here?
    - What is Ray, what is Anyscale? Why Ray for ML?
- Hands On - How to Ray & Anyscale?

Instructors & TAs

- Instructor: Charles Greer
  - Solutions Architect @ Anyscale
  - Technical Point of Contact for Koch (find me on teams!)
- TAs
  - +1 Solutions Architect: Bill Wang
  - Product Team @ Anyscale (find us on teams!)

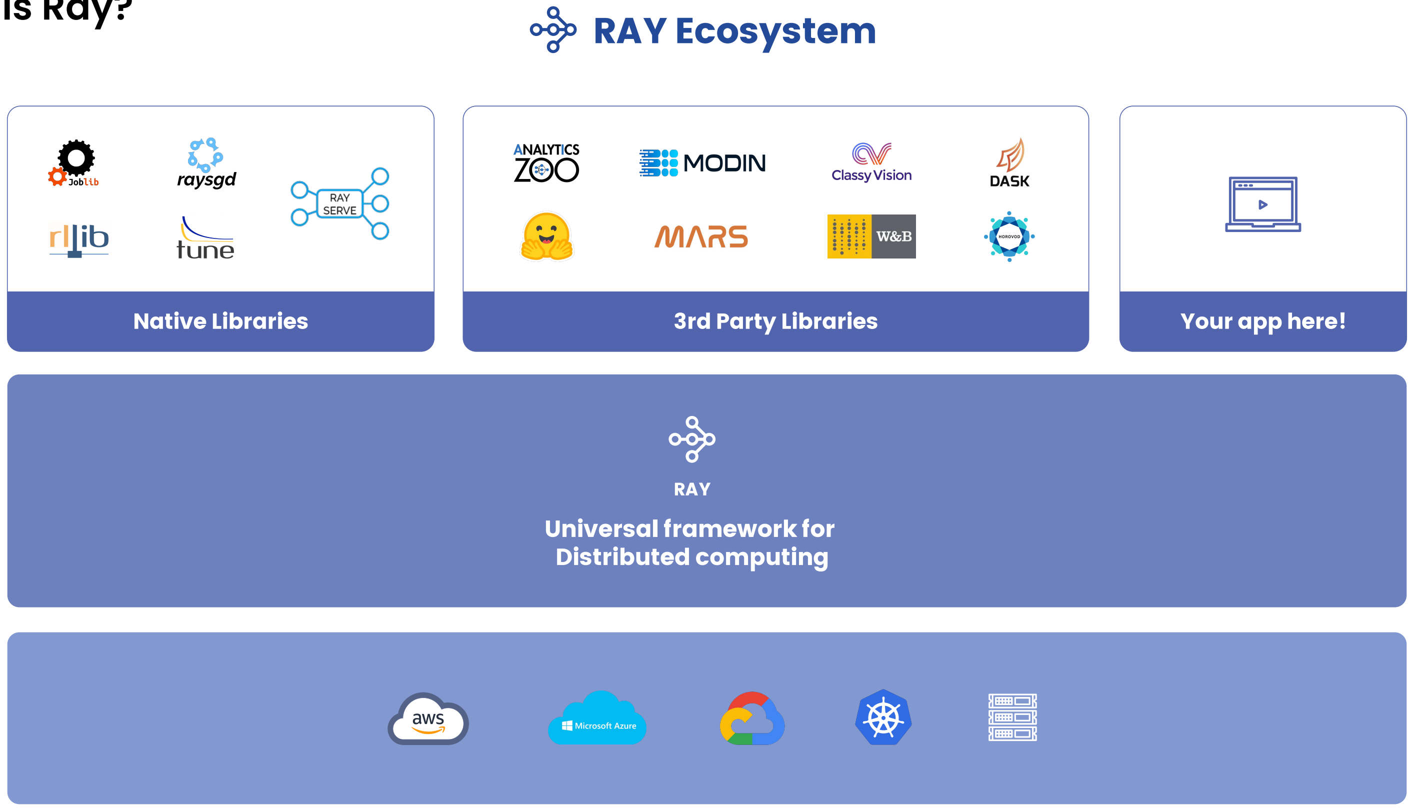
Why are we here today?

Training Schedule

- **Today:** Anyscale + Ray Overview
- **In ~ + 2 Weeks: Deep Dive on Anyscale + Ray for ML dev**
    - Meeting with Maxim + Badrul next week, please send any questions / topics you'd like us to cover!
- **In ~ + 3 Weeks: Deep Dive on Anyscale + Ray for production**
    - Meeting with Dermot + Devin in the 2 weeks, please send any questions / topics you'd like us to cover!
- **Follow ups as needed (e.g., RLlib)**

What is Ray?
What is Anyscale?

What is Ray?


RAY Ecosystem
Joblib
raysgd
RAY SERVE
rllib
tune
Native Libraries
ANALYTICS ZOO
MODIN
Classy Vision
DASK
MARS
W&B
HOROVOD
3rd Party Libraries
Your app here!
RAY
Universal framework for Distributed computing
aws
Microsoft Azure

What is Ray?

- An ecosystem of tools to scale any kind of workload
    - **Ray Core:** A universal framework for executing and managing distributed computation.
    - **Ray Libraries:** A set of high-level ML-oriented libraries that leverage this united computing environment.
- A community of thousands of developers, data scientists, and ML engineers.

Ray Core

- Distribute functions and classes across a cluster of machines without having to think about the infrastructure
- Low level tool to build apps from scratch
- Analogy: The Python Language
  - *But purpose made for scaling applications*

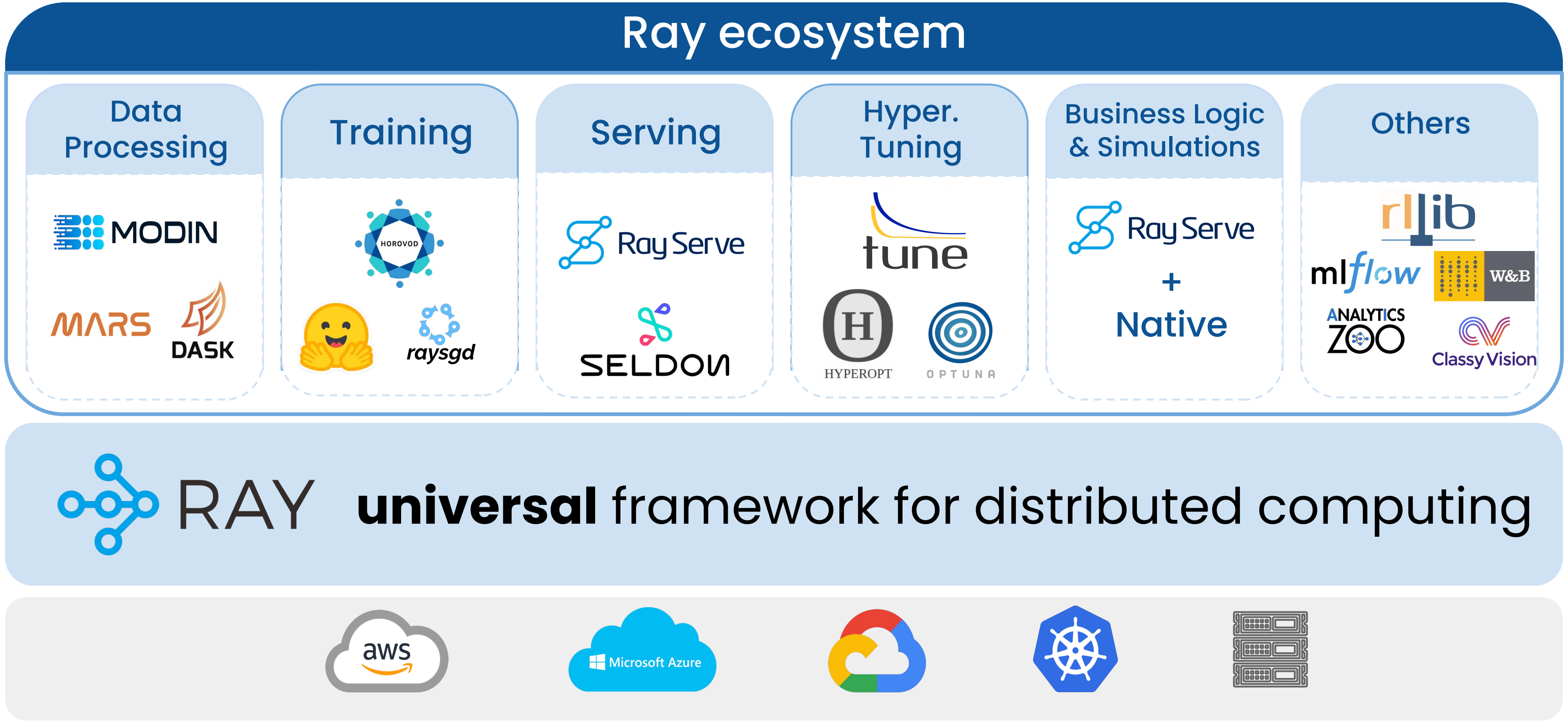
Ray (Native) Libraries

- **Ray Tune** - scaling HPO with cutting edge algos
- **Ray Serve** - scaling model serving
- **Ray SGD** - scaling distributed training (multi-node + multi GPU)
- **Ray RLlib** - scaling reinforcement learning
- **Ray Data** - scaling data processing
- Analogy: The Python Standard Library
  - *But purpose made for scaling and deploying ML applications*

Ray (Integrated) Libraries

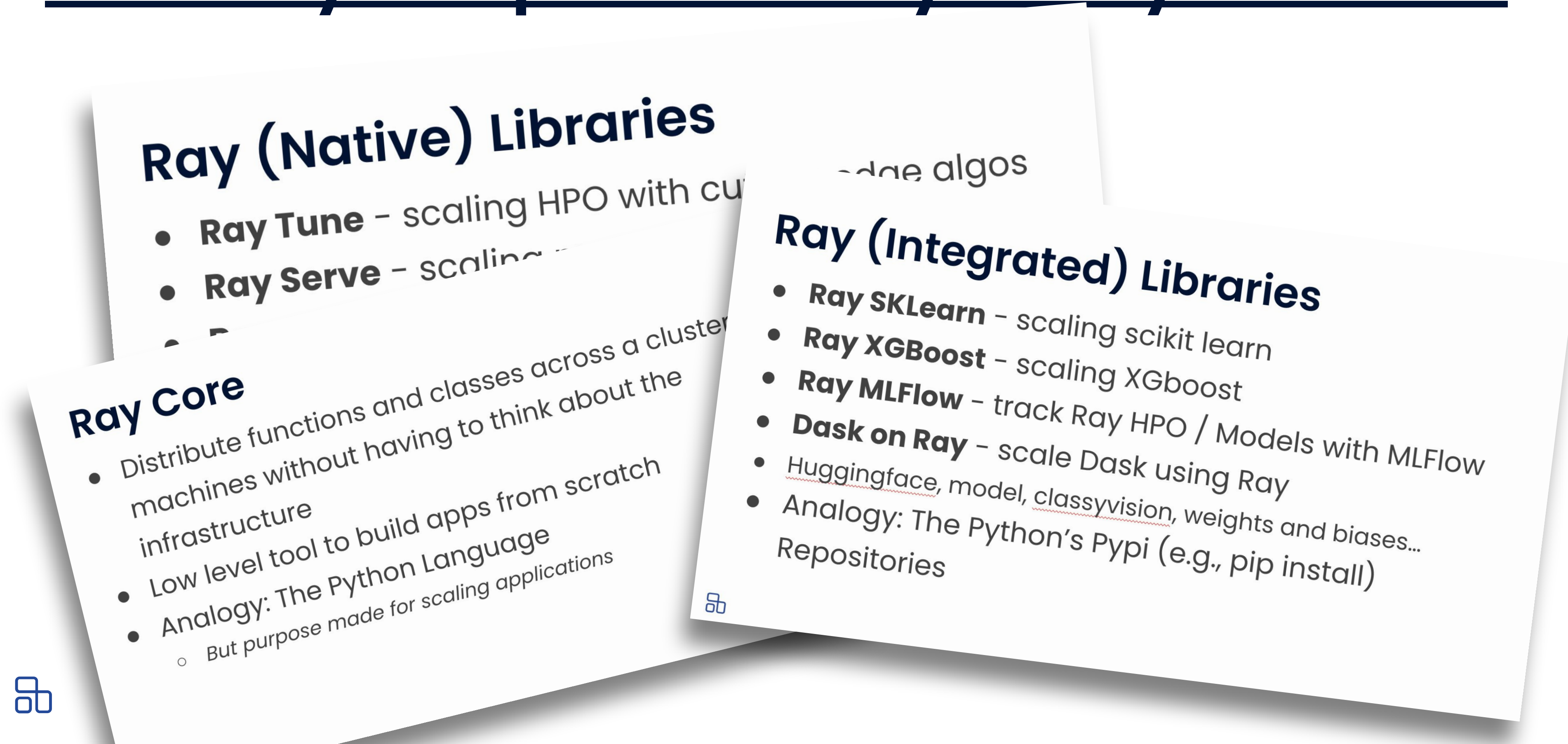
- **Ray SKLearn** - scaling scikit learn
- **Ray XGBoost** - scaling XGboost
- **Ray MLFlow** - track Ray HPO / Models with MLFlow
- **Dask on Ray** - scale Dask using Ray
- Huggingface, model, classyvision, weights and biases...
- Analogy: The Python's Pypi (e.g., pip install) Repositories

Ray's Ecosystem of ML and Data Libraries



Key Point...

Use only the parts of Ray that you need!



Ray (Native) Libraries

- **Ray Tune** - scaling HPO with cu
  dge algos
- **Ray Serve** - scalin

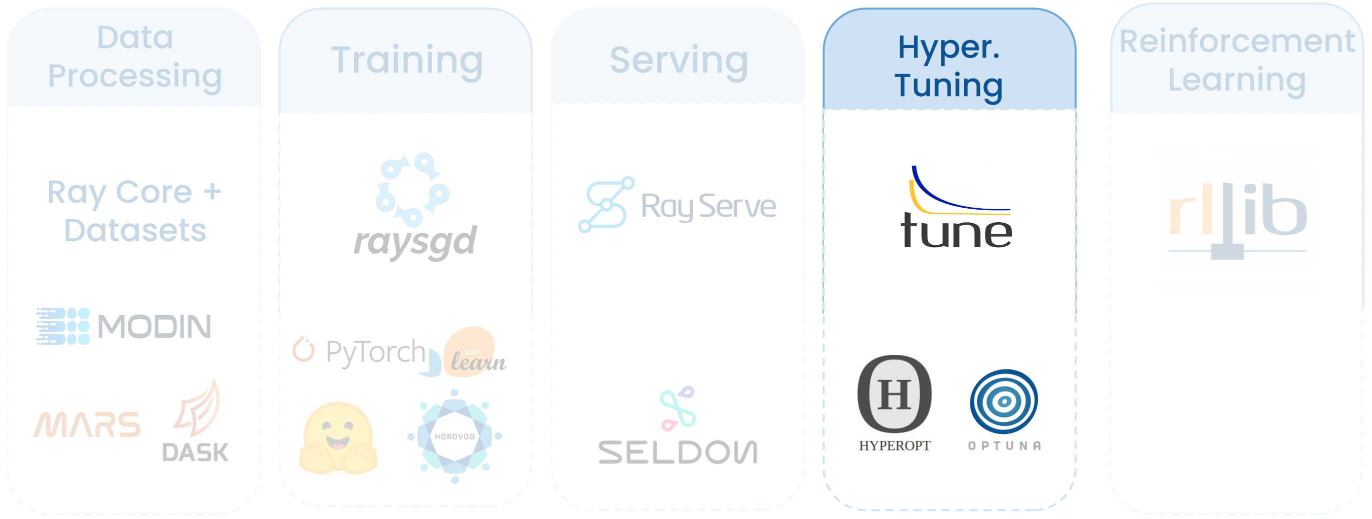
Ray Core

- Distribute functions and classes across a cluster machines without having to think about the infrastructure
- Low level tool to build apps from scratch
- Analogy: The Python Language
  - *But purpose made for scaling applications*

Ray (Integrated) Libraries

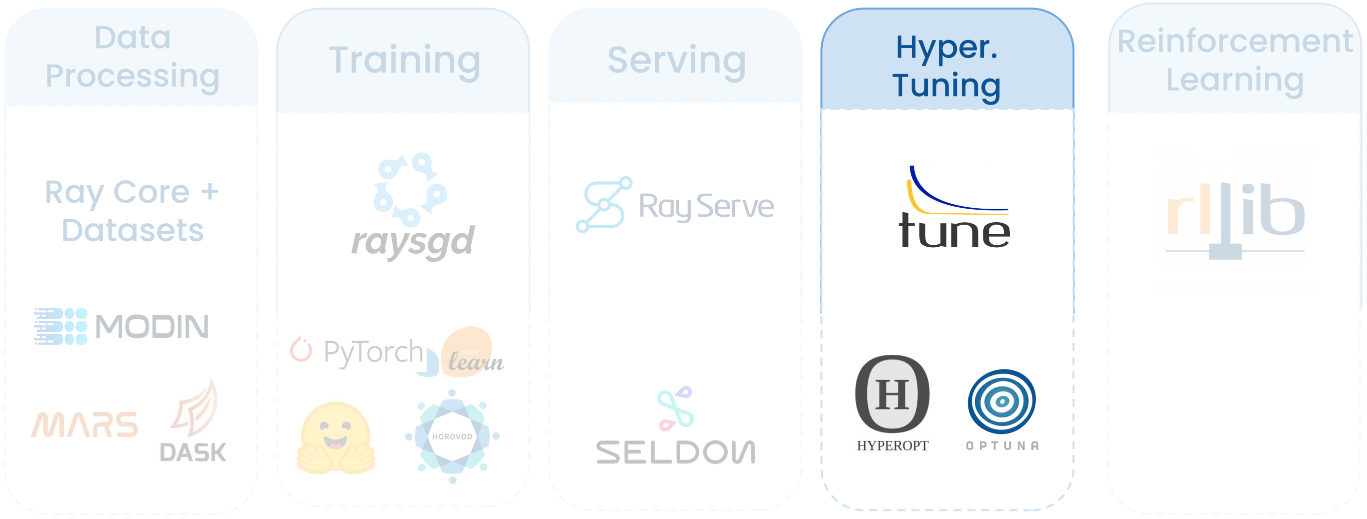
- **Ray SKLearn** - scaling scikit learn
- **Ray XGBoost** - scaling XGboost
- **Ray MLFlow** - track Ray HPO / Models with MLFlow
- **Dask on Ray** - scale Dask using Ray
- Huggingface, model, classyvision, weights and biases...
- Analogy: The Python's Pypi (e.g., pip install) Repositories

Rich ecosystem for scaling ML workloads



Challenges in scaling hyperparameter tuning?

Rich ecosystem for scaling ML workloads



Integrate Ray Tune!
No need to adopt entire Ray framework.

What is Anyscale?

Company founded by the Creators of Ray

Ion Stoica (Leader RISELab, AMPLab, Co-Founder Databricks, Conviva),
Michael Jordan (Leader RISELab, AMPLab),
Robert Nishihara (Ph.D. Berkeley, Ray co-creator),
Philipp Moritz (Ph.D. Berkeley, Ray co-creator)

Experts in Distributed Computing + ML

ex-Uber, Stripe, Databricks, Google Brain

Backed by Top Tier VCs:

A16z, NEA, Intel Capital

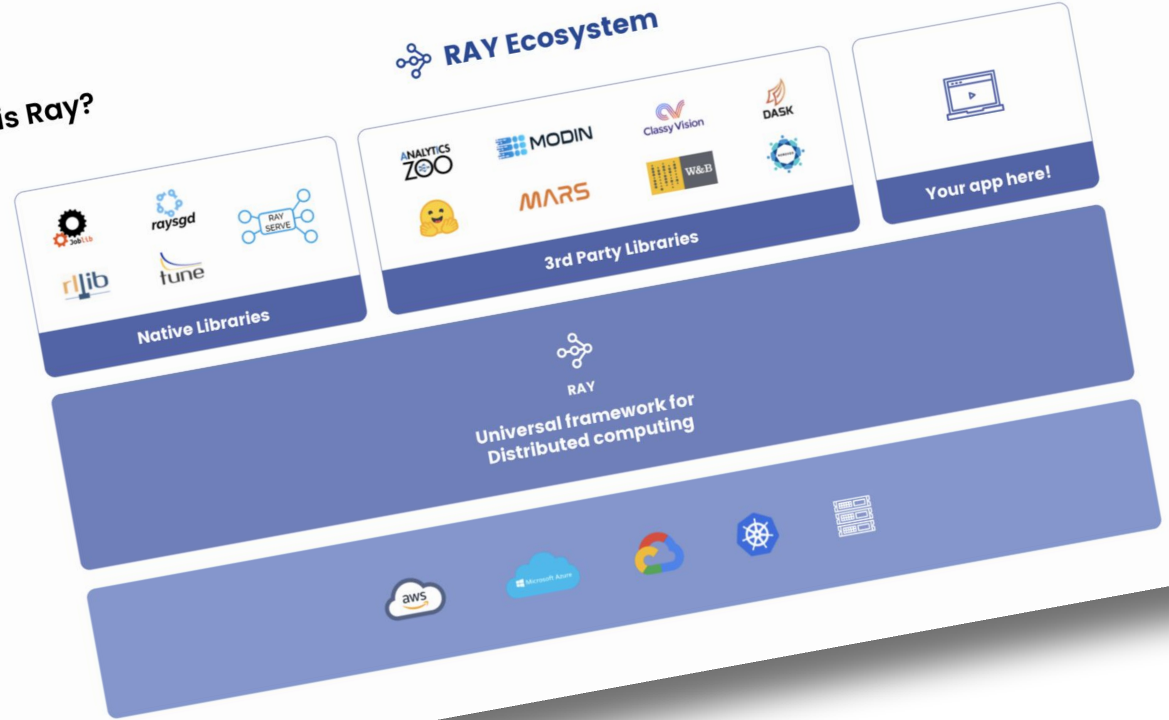
Anyscale's Mission

- Enable data teams to build ML applications faster and deploy applications more reliably by providing a managed Ray service.

How do we do this?

  - Enable Koch to focus on *ML applications* that make a difference for the business, not *infrastructure to get a service out the door.*
  - Provide simple ways of tapping into scaling (e.g., remote connect to a Ray cluster)
  - Provide robust APIs for CI/CD + deployment

What is Ray?


RAY Ecosystem
Joblib
raysgd
RAY SERVE
rllib
tune
Native Libraries
ANALYTICS ZOO
MODIN
Classy Vision
DASK
MARS
W&B
3rd Party Libraries
Your app here!
RAY
Universal framework for
Distributed computing
aws
Microsoft Azure

RAY Ecosystem

Joblib
raysgd
RAY SERVE
rllib
tune

Native Libraries

ANALYTICS ZOO
MODIN
Classy Vision
DASK
MARS
W&B
HOROVOD

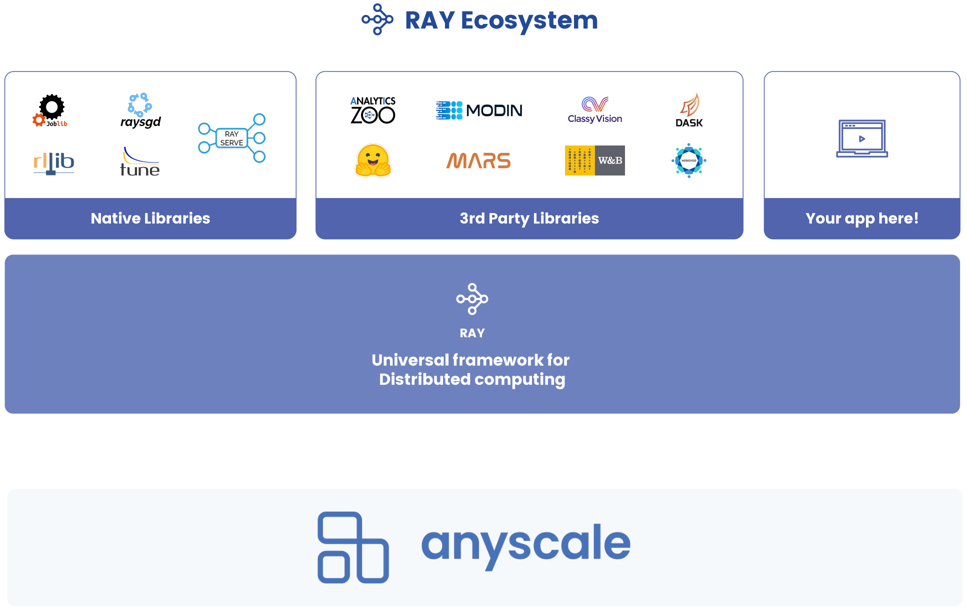
3rd Party Libraries

Your app here!

RAY

Universal framework for Distributed computing

anyscale



Anyscale leverages the same Ray Open Source APIs

Workflow Advantages:

- Simple access to scale in dev
- Streamlined APIs to move to production

System Advantages:

- Robust APIs for managing 10s - 100s of clusters
- Proprietary Cluster Manager
- Optimized Ray Runtime
- Integrated metrics & monitoring

Hands-on Demo & Lab Using Ray on Anyscale

What is Ray - 3 Key Ideas

Execute functions remotely as tasks, and instantiate classes remotely as actors

- **Support both stateful and stateless computations**

Asynchronous execution using futures

- **Enable parallelism**

Distributed (immutable) object store

- **Efficient communication** (send arguments by reference)

What is Ray? API looks like:

Function	→	Task
Class	→	Actor
Object	→	(Distributed) Object

What is Ray - API

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


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

a = read_array(file1)
b = read_array(file2)
sum = add(a, b)
```

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

c = Counter()
c.inc()
c.inc()
```

What is Ray – API

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

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

a = read_array(file1)
b = read_array(file2)
sum = add(a, b)
```

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

c = Counter()
c.inc()
c.inc()
```

What is Ray – API

```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)

ref1 = read_array.remote(file1)
ref2 = read_array.remote(file2)
ref = add.remote(ref1, ref2)
sum = ray.get(ref)
```

```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()
ref4 = c.inc.remote()
ref5 = c.inc.remote()
```

What is Ray - API (Actor Handles)

Invoke actor methods from other tasks/actors/applications.

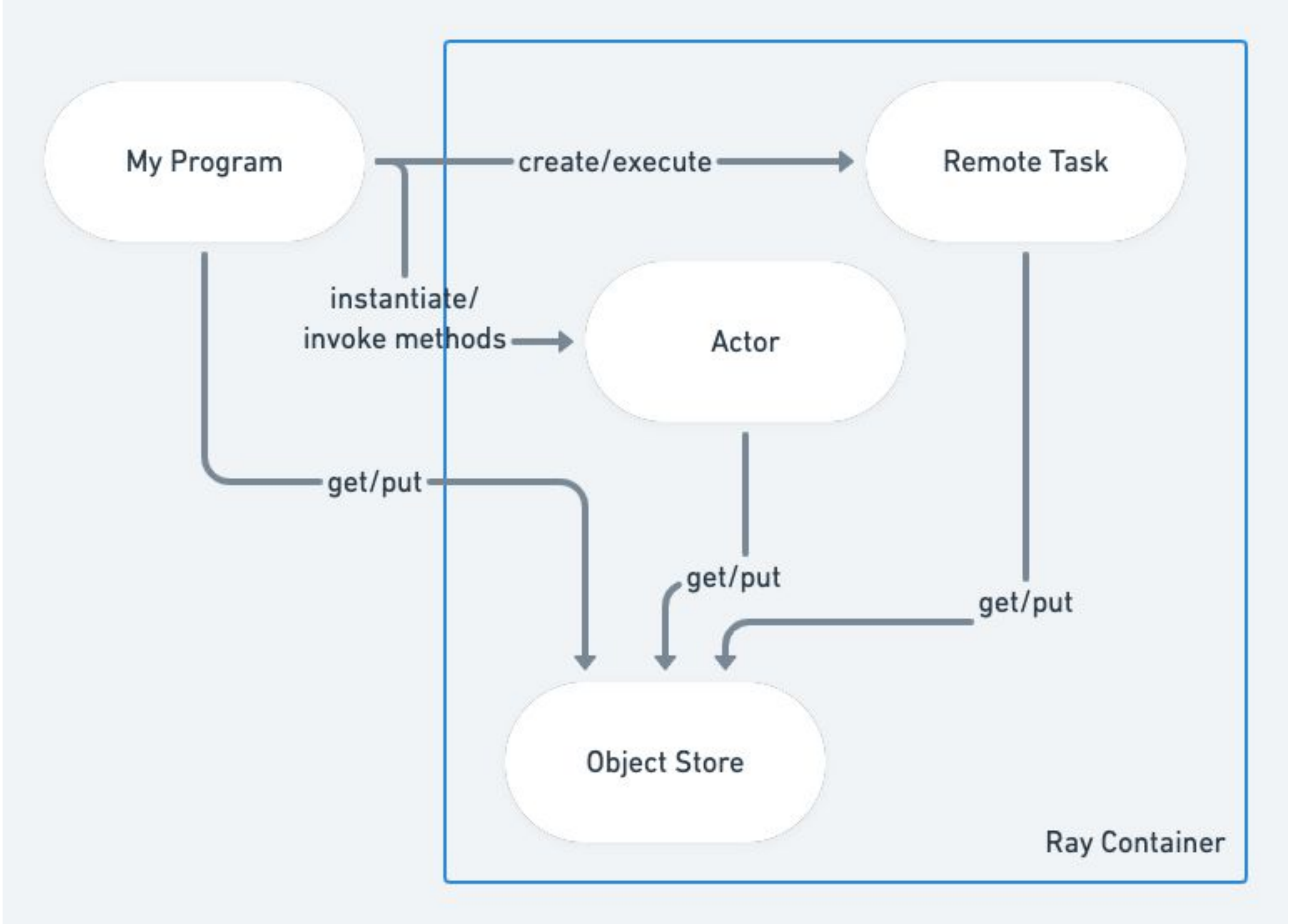
```python
@ray.remote(num_gpus=1)
class Counter(object):
    def __init__(self):
        self.value = 0
    def inc(self):
        self.value += 1
        return self.value

c = Counter.remote()
id = c.inc.remote()
```

```python
# Use the actor from a
# different task

@ray.remote
def use_actor(c):
    id = c.inc.remote()
    ray.get(id)

use_actor.remote(c)
```

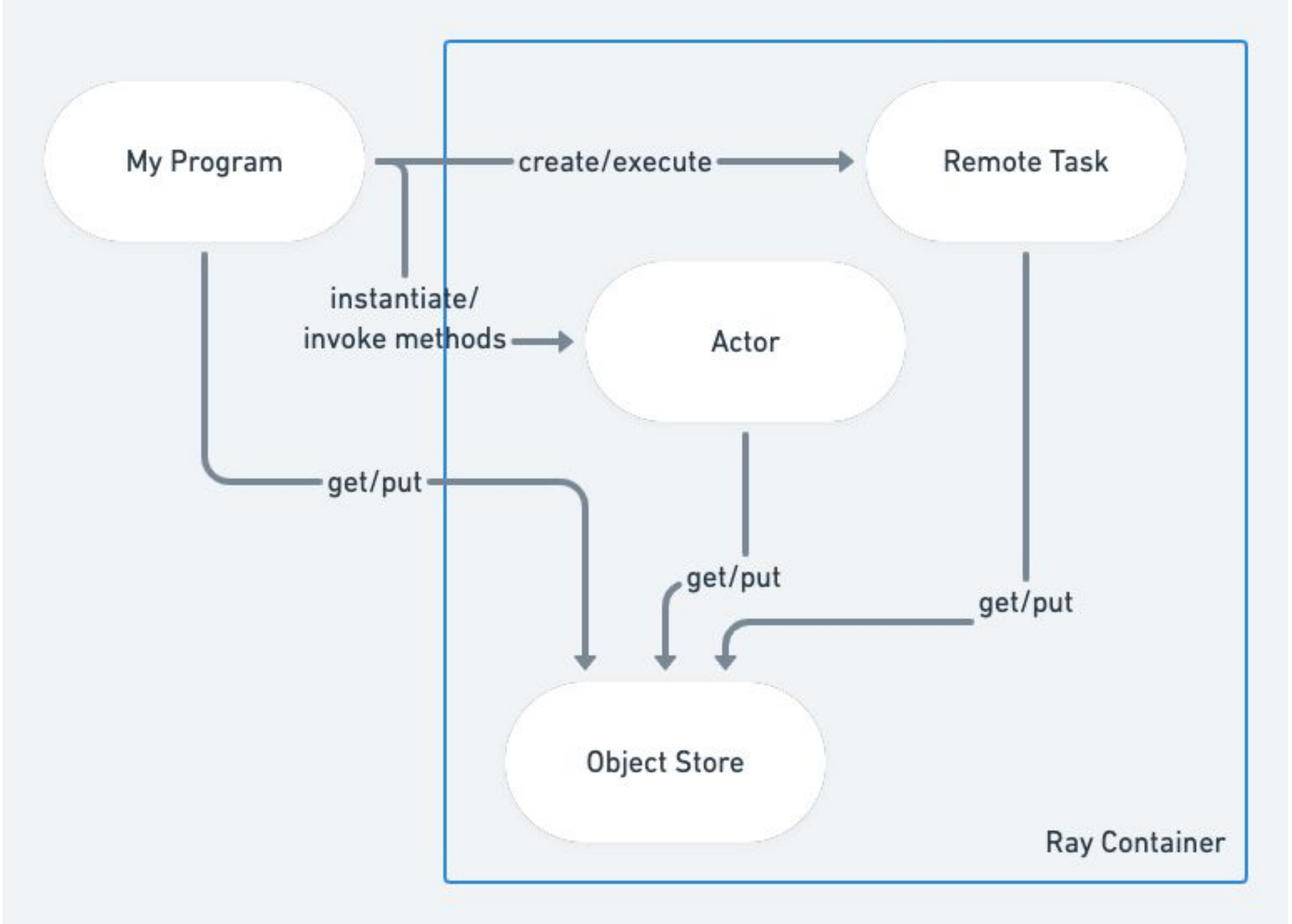

My Program
create/execute
Remote Task
instantiate/
invoke methods
Actor
get/put
get/put
get/put
Object Store
Ray Container

Ray Application Lifecycle

- Experiment/Design on a Single Node
  - Jupyter Notebook on Anyscale
  - Laptop and local ray
- Optimize and Scale
- Automate and Deploy

Let us Code

- Tasks
- Actors
- Remote calls
- Put and get
- Best Practices
- Scaling

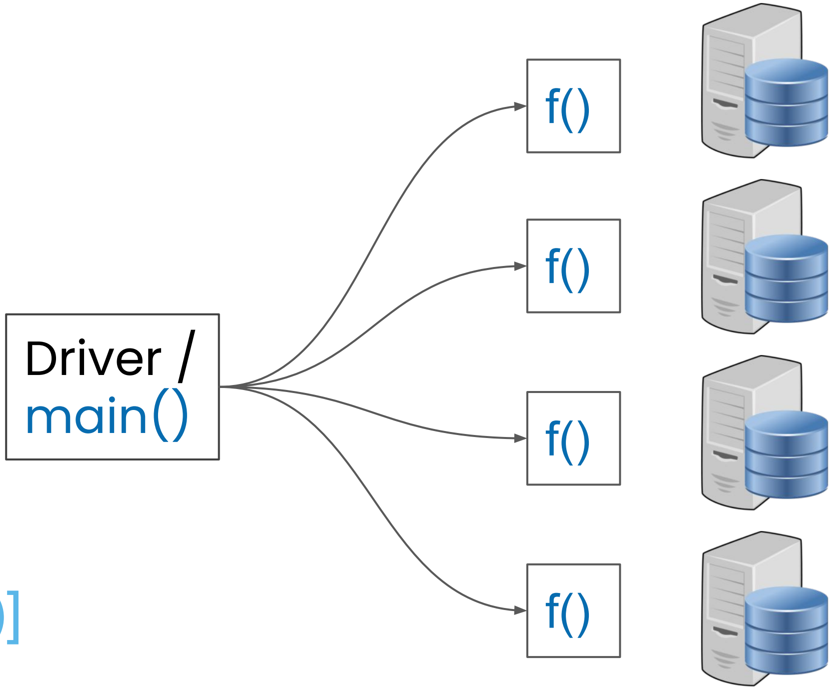

My Program
create/execute
Remote Task
instantiate/
invoke methods
Actor
get/put
get/put
get/put
Object Store
Ray Container

What is Ray? Hello World...

```
import ray
ray.init()

@ray.remote
def f(x):
    return x * x

futures = [f.remote(i) for i in range(4)]
print(ray.get(futures)) # [0, 1, 4, 9]
```

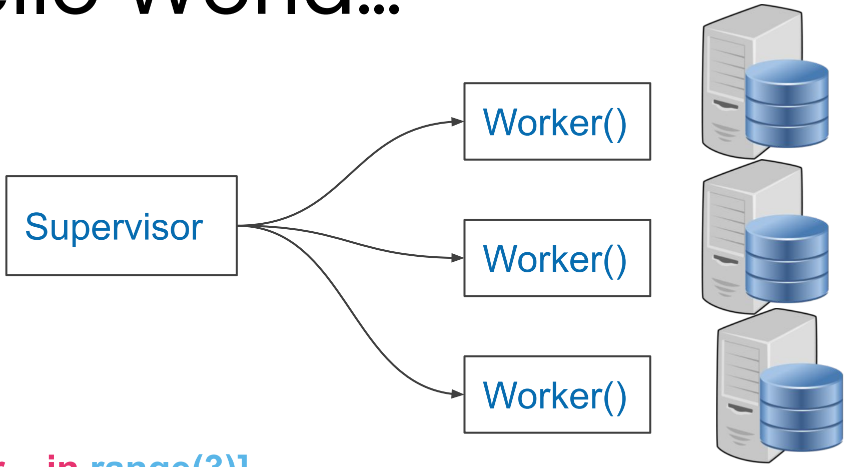


What is Ray? Hello World...

```
@ray.remote(num_cpus=1)
class Worker:
    def work(self):
        return "done"

@ray.remote(num_cpus=1)
class Supervisor:
    def __init__(self):
        self.workers = [Worker.remote() for _ in range(3)]
    def work(self):
        return ray.get([w.work.remote() for w in self.workers])

ray.init()
sup = Supervisor.remote()
print(ray.get(sup.work.remote()))  # outputs ['done', 'done', 'done']
```



Additional Resources Ray

Documentation (docs.ray.io)

Quick start example, reference guides, etc

Forums (discuss.ray.io)

Learn / share with broader Ray community, including core team

Ray Slack

Connect with the Ray team and community

Anyscale Support

Find us on Teams, on the Open Source Ray Slack, or over email.

Charles Greer
Bill Wang
Bill Chambers
Javier Redondo
Tricia Fu
Will Drevo

anyscale