Enable RISC-V Accelerated Ray for AI Workloads

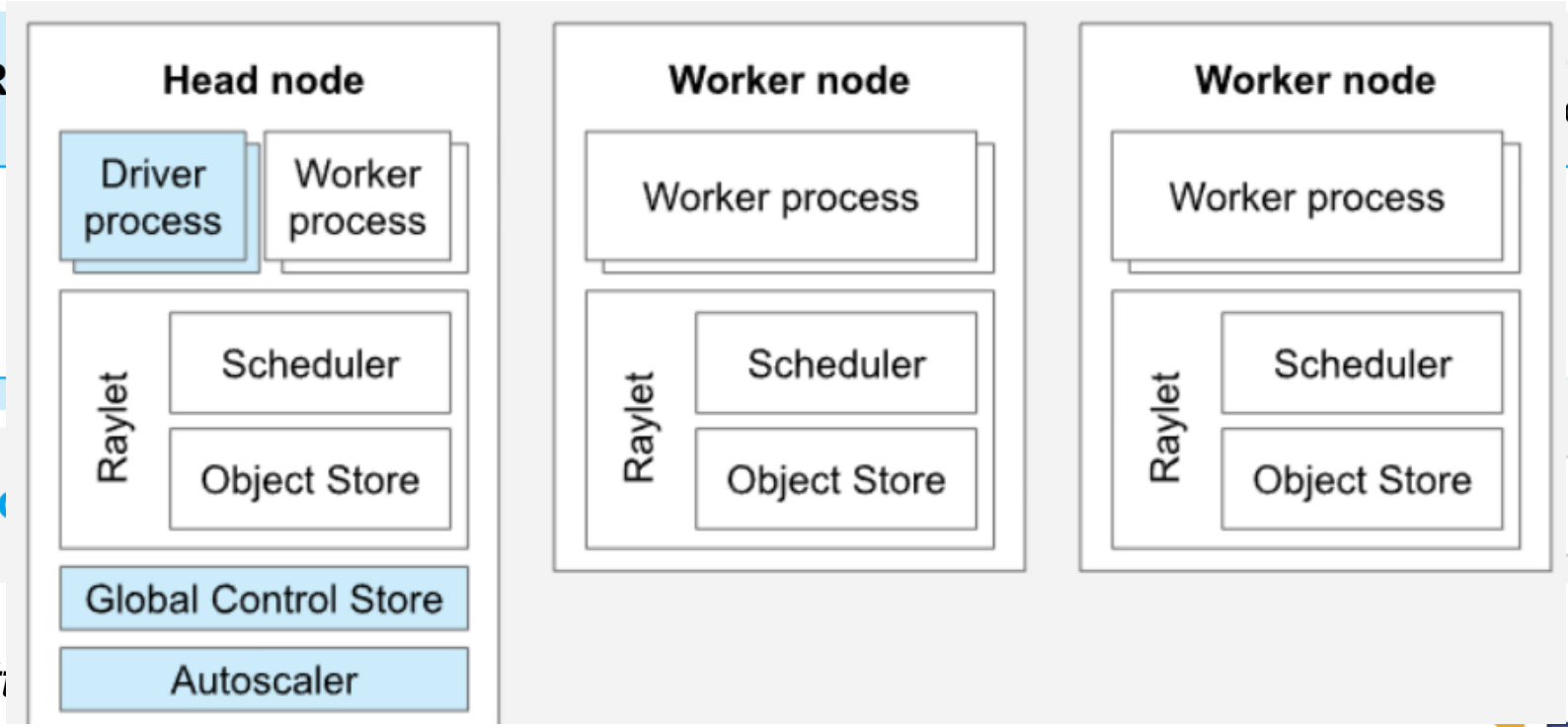
Tiejun Chen
RISC-V International Ambassador

Agenda

- Ray Overview
- Enable Ray on RISC-V
  - demos
- Ray extensions
  - Accelerate Ray
  - Remote ML/AI
- What's likely next

Ray Overview I

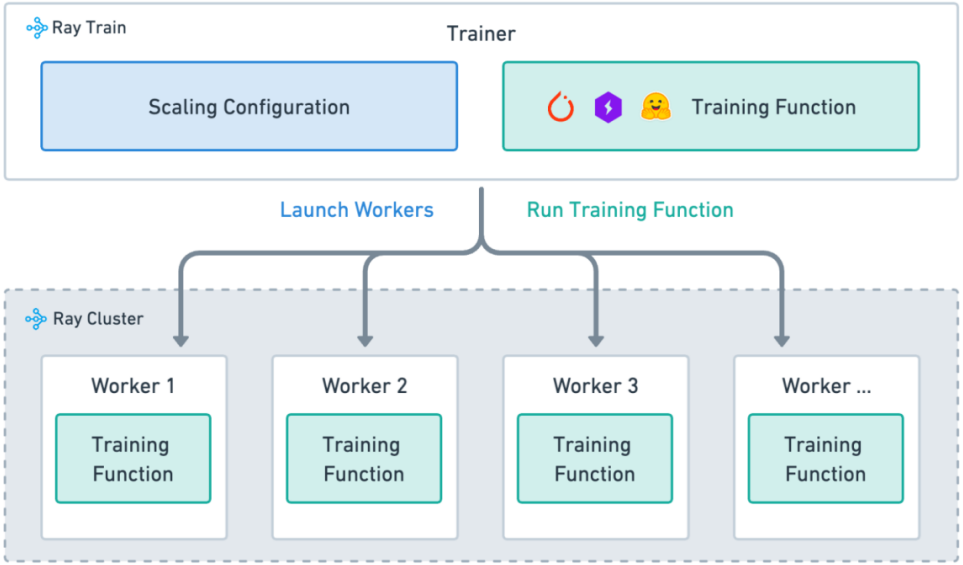
Brief



Ray Overview II

Ray Train

- A scalable machine learning library for distributed training and fine-tuning
- Main components
  - Training function
  - Worker
  - Scaling configuration
  - Trainer



❖ *https://docs.ray.io/en/master/train/overview.html*

Ray Overview III

Rat Serve

- A scalable model serving library for building online inference APIs
  - Be framework-agnostic
    - ❑ Tensorflow, Pytorch ,etc

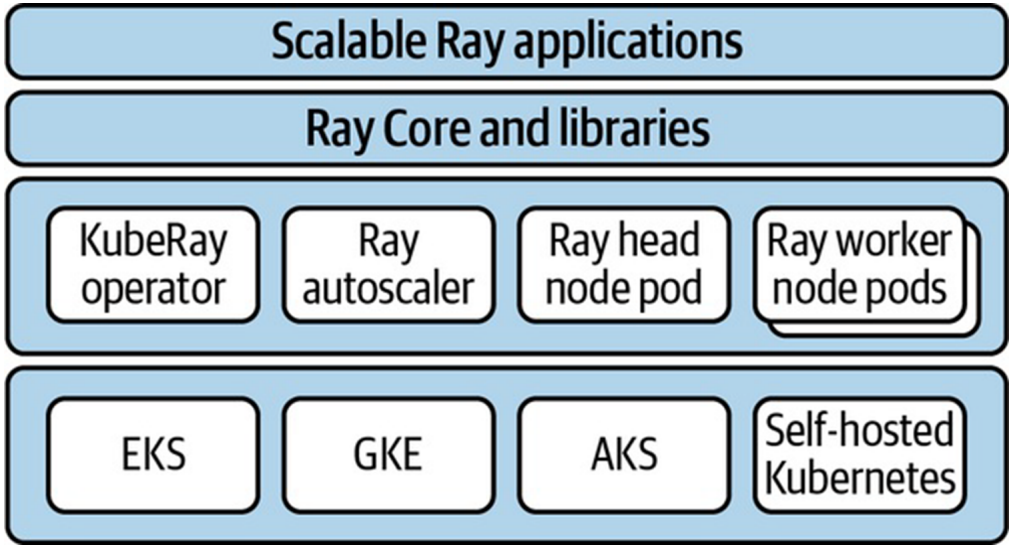
```python
@serve.deployment
class TFMnistModel:
    def __init__(self, model_path: str):
        import tensorflow as tf

        self.model_path = model_path
        self.model = tf.keras.models.load_model(model_path)

    async def __call__(self, starlette_request: Request) -> Dict:
        # Step 1: transform HTTP request -> tensorflow input
        # Here we define the request schema to be a json array.
        input_array = np.array((await starlette_request.json())["array"])
        reshaped_array = input_array.reshape((1, 28, 28))

        # Step 2: tensorflow input -> tensorflow output
        prediction = self.model(reshaped_array)

        # Step 3: tensorflow output -> web output
        return {"prediction": prediction.numpy().tolist(), "file": self.model_path}
```

Doesn't perform any model-specific optimizations

Ray Overview

KubeRay – Managing Ray clusters on Kubernetes

- RayService
- RayJob
- RayCluster



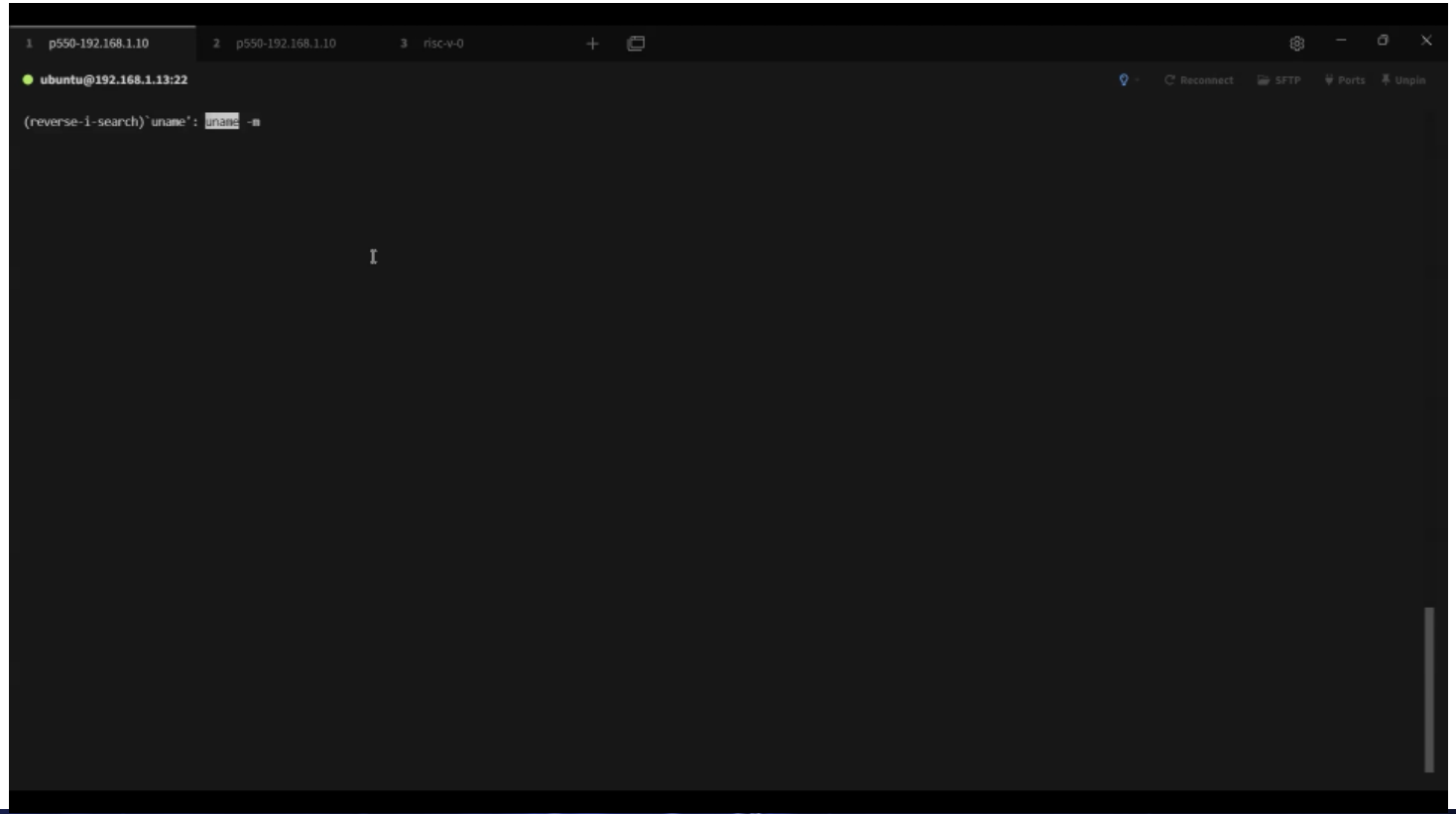
Enable Ray to RISC-V

Basic requirements | Challenges

- *ML frameworks – Pytorch, Tensorflow, etc*
- *Bazel == 6.5.0*
  - abseil-cpp (RDCYCLE)
- *Golang*
- *Docker*
- *50+packages*
  - pyarrow
  - relocate-perl

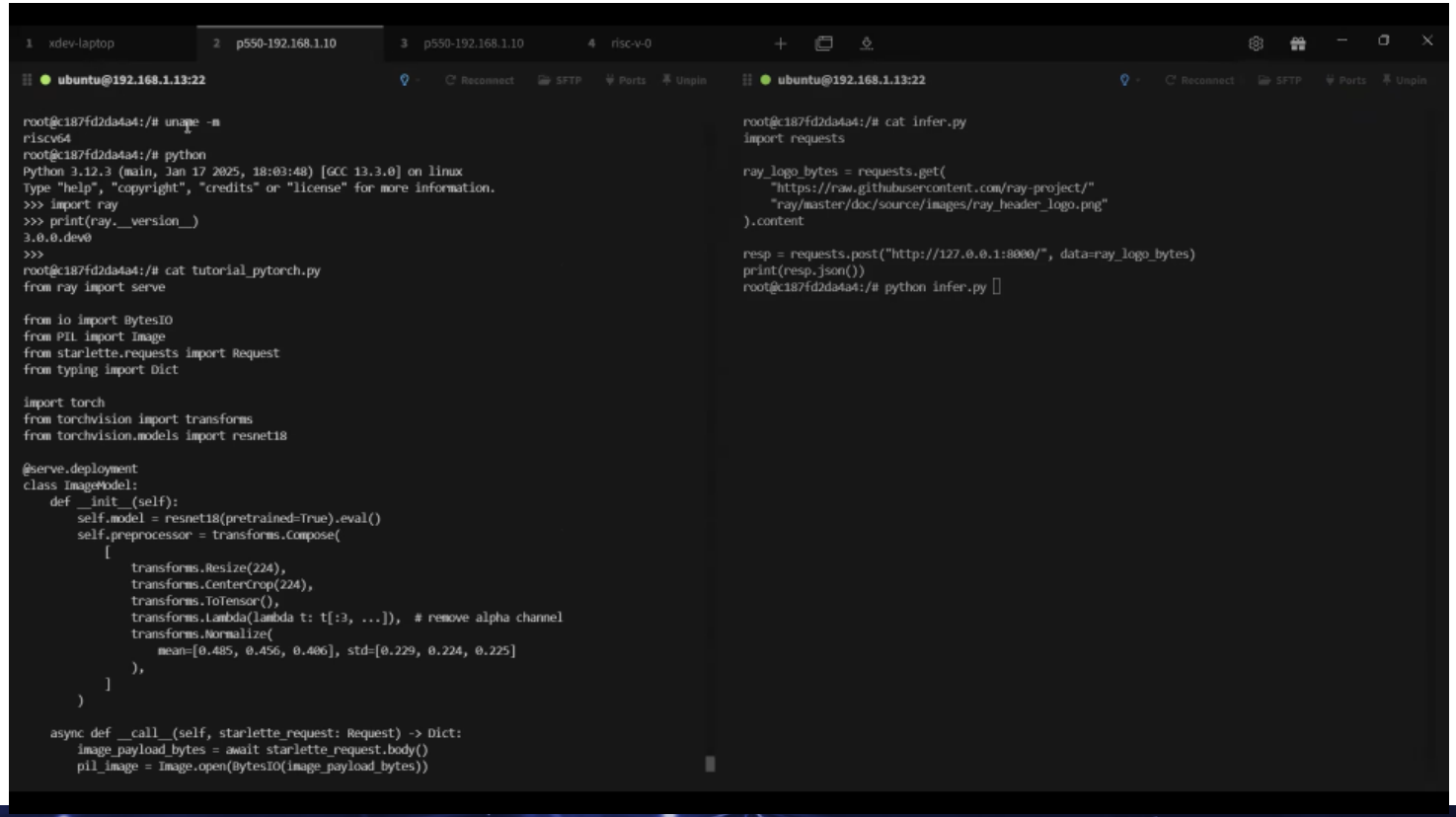
Enable Ray to RISC-V – Demo I

Ray Train on RISC-V



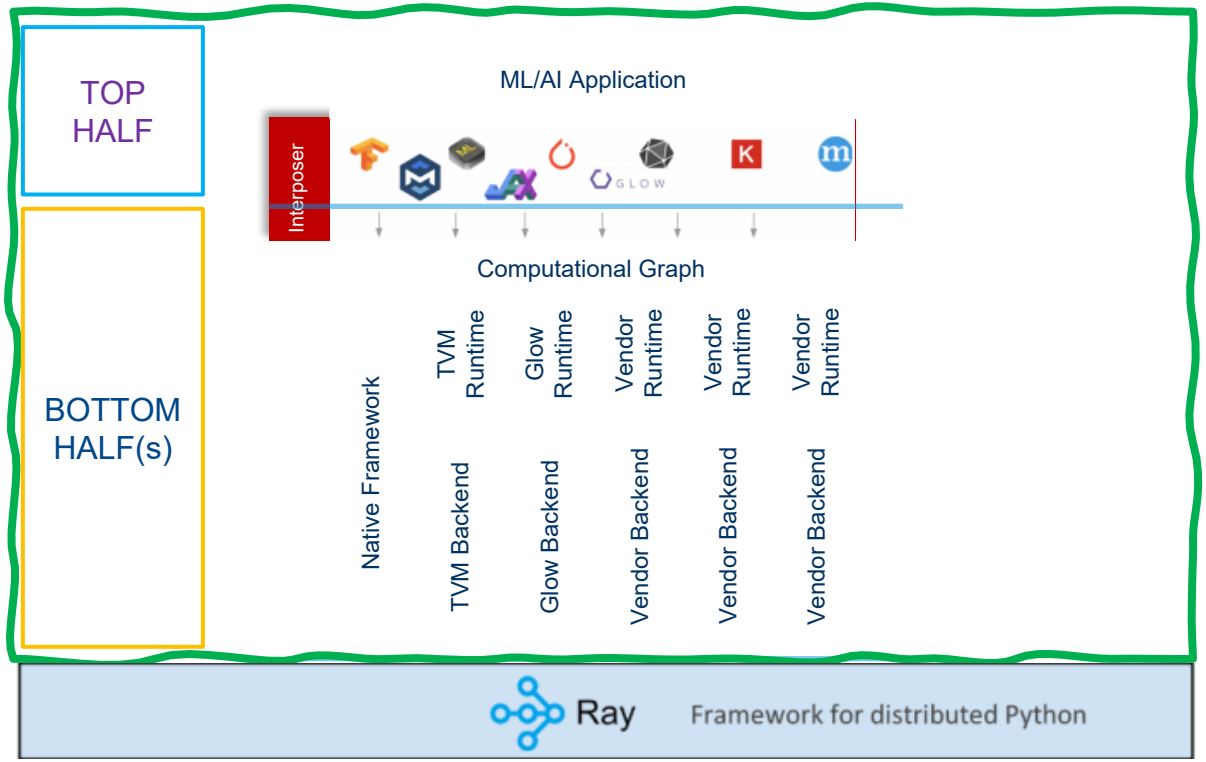
Enable Ray to RISC-V – Demo II

Ray Serve on RISC-V



Ray extensions – Accelerate Ray

Leverage graph compliers



ML Frameworks

↕

Computational Graph

↕

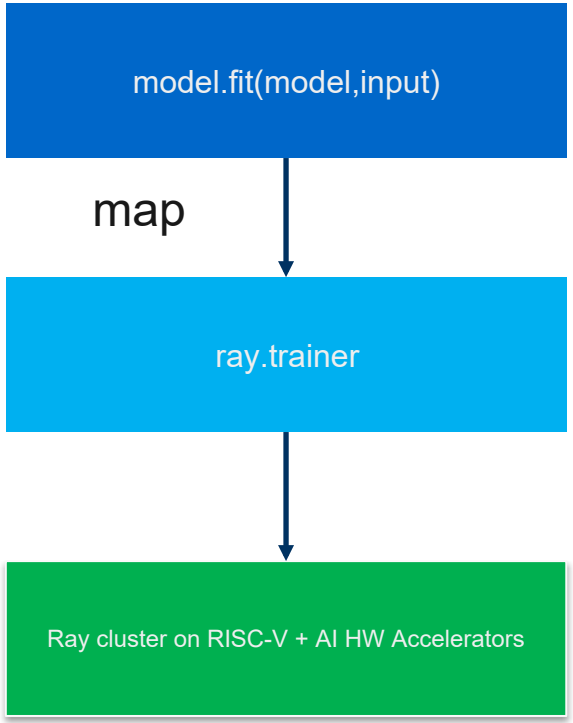
Low-level Libs/Device Drivers

Ray extensions – Remote ML/AI I

Empower native ML frameworks

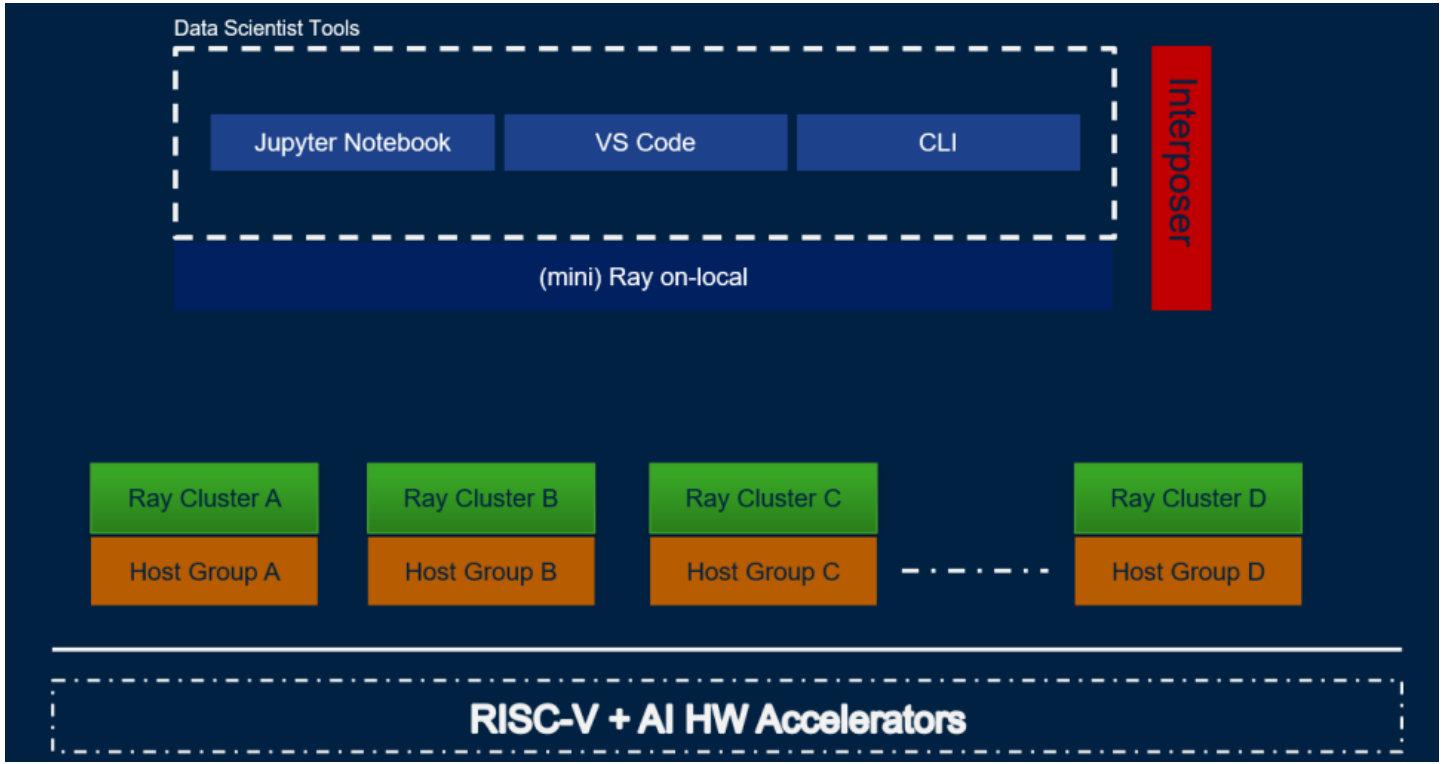
- How - Intercept key APIs at runtime
- e.g – Tensorflow Training

1. loading data
2. define model
3. model.compile()
4. model.fit(model, input, epoch)



Ray extensions – Remote ML/AI II

Empower data science tools



What's likely next ?

Empower AI everywhere with RISC-V

- Get into Ray community
- Get on Ray cluster (KubeRay)
- Get more cases
  - RISC-V host with AI hardware accelerators

Q & A

Thank you!

@Tiejun_Chen

tiejun.china@gmail.com

https://www.linkedin.com/in/tiejun-chen-41a6654b/