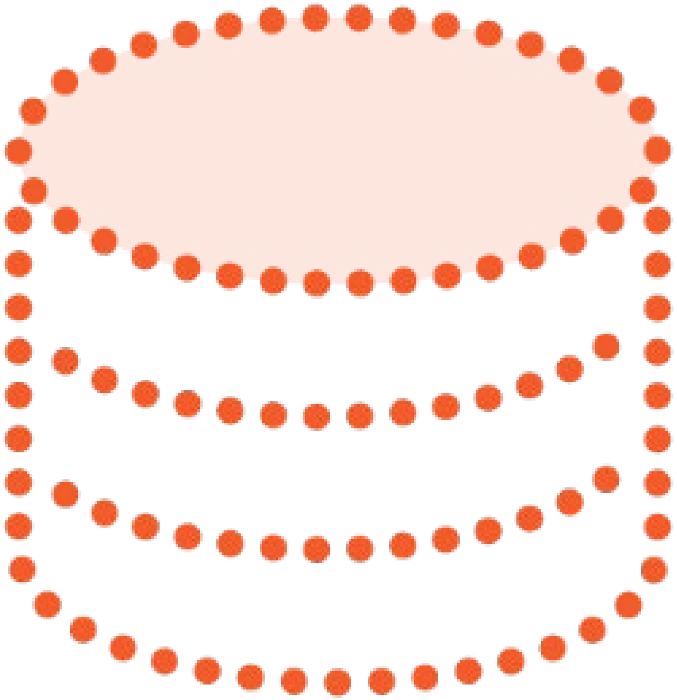
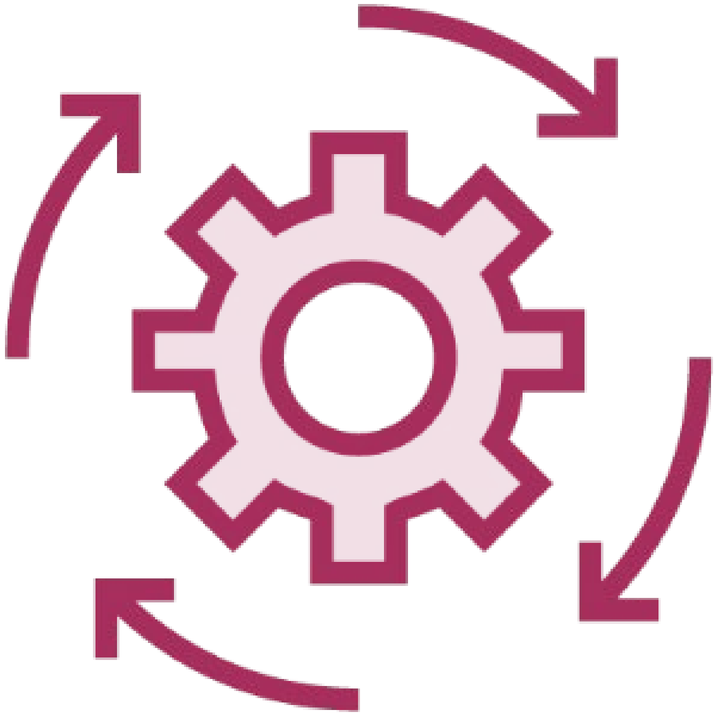
Exploring Distributed Computing with Ignite

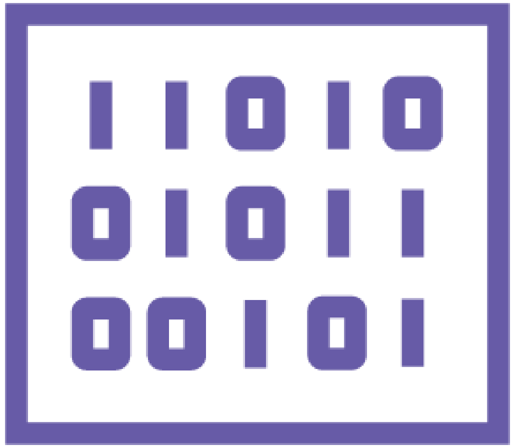
Edward Curren

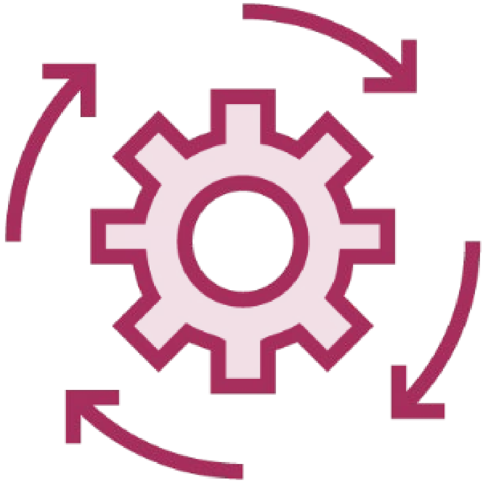
ENTERPRISE ARCHITECT

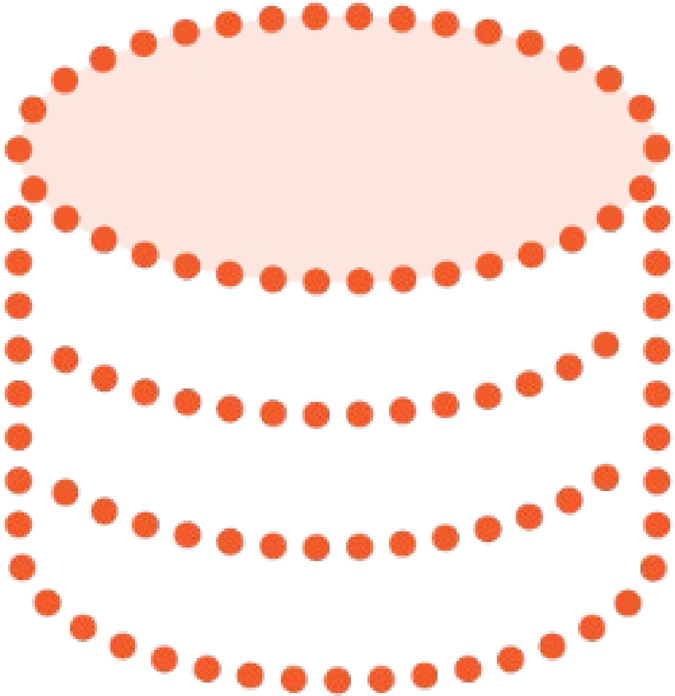
@EdwardCurren http://www.edwardcurren.com

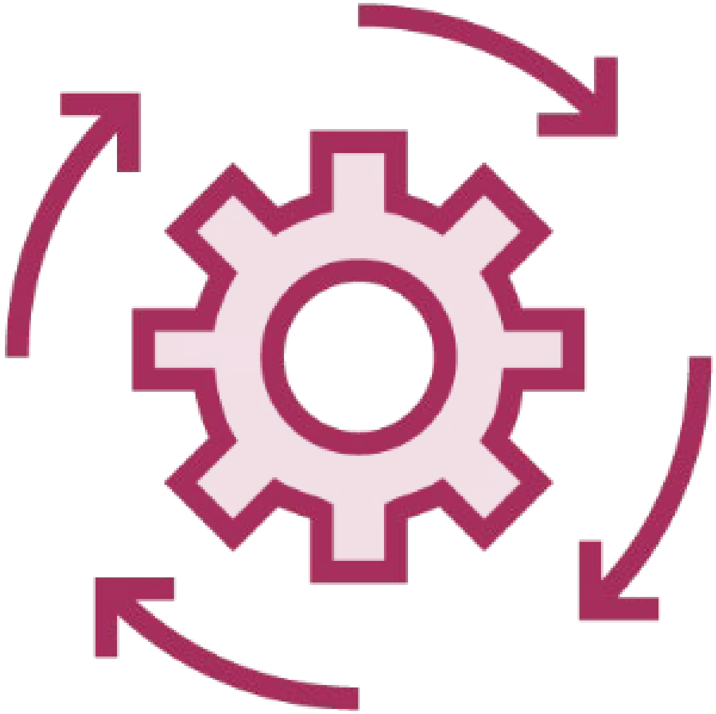





11010
01011
00101







Scalable

fail-over

Cluster

parallel execution of tasks

distribution mechanics

Overview

Distributed closures

ComputeTask, ComputeJob and ComputeTaskSession

Job scheduling

Checkpointing

Collocation with data

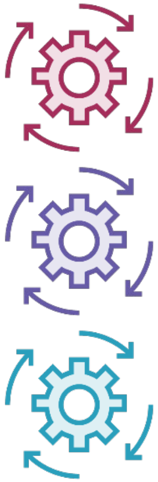
Distributing a Task’s Workload

Distributing a Task’s Workload

Task Parallelism

Call & Run

Distributing a Task’s Workload



Task Parallelism

Call & Run

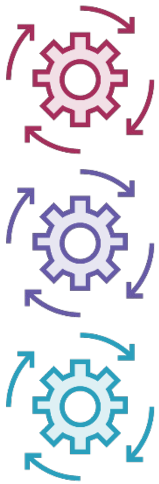
Data Parallelism

Apply

DA781	Enroute
DA294	Ready
DA319	Enroute
DA6804	Enroute
DA984	Arrived
DA681	Ready
DA965	Enroute
DA201	Arrived
DA881	Enroute
DA4901	Arrived
DA198	Ready
DA681	Enroute
DA615	Arrived
DA485	Enroute
DA354	Enroute

DA781	Enroute
DA294	Ready
DA319	Enroute
DA6804	Enroute
DA681	Ready
DA965	Enroute
DA881	Enroute
DA198	Ready
DA681	Enroute
DA485	Enroute
DA354	Enroute

Distributing a Task’s Workload

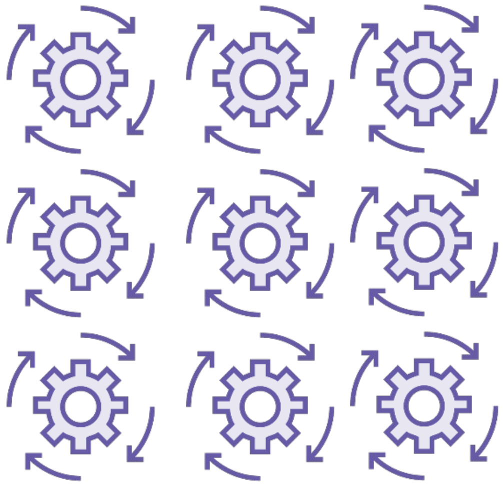


Task Parallelism

Call & Run

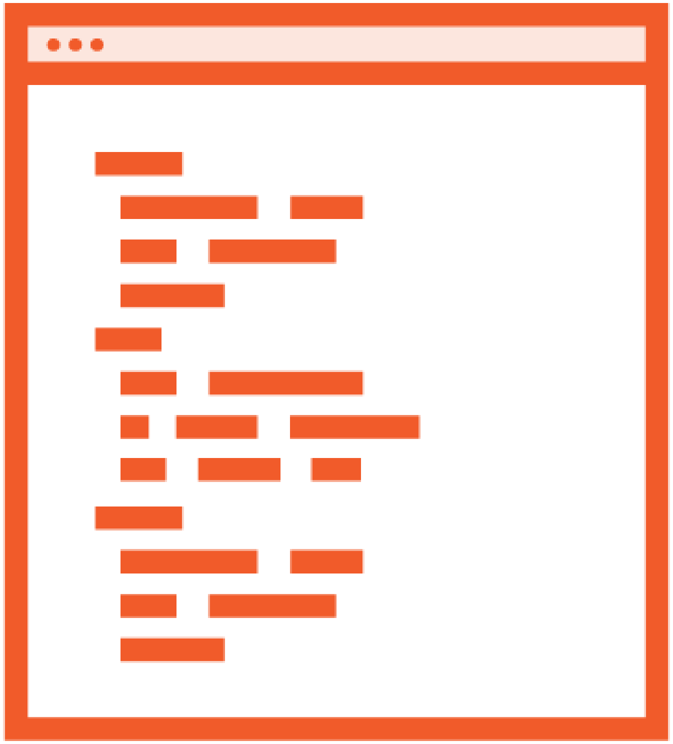
Data Parallelism

Apply



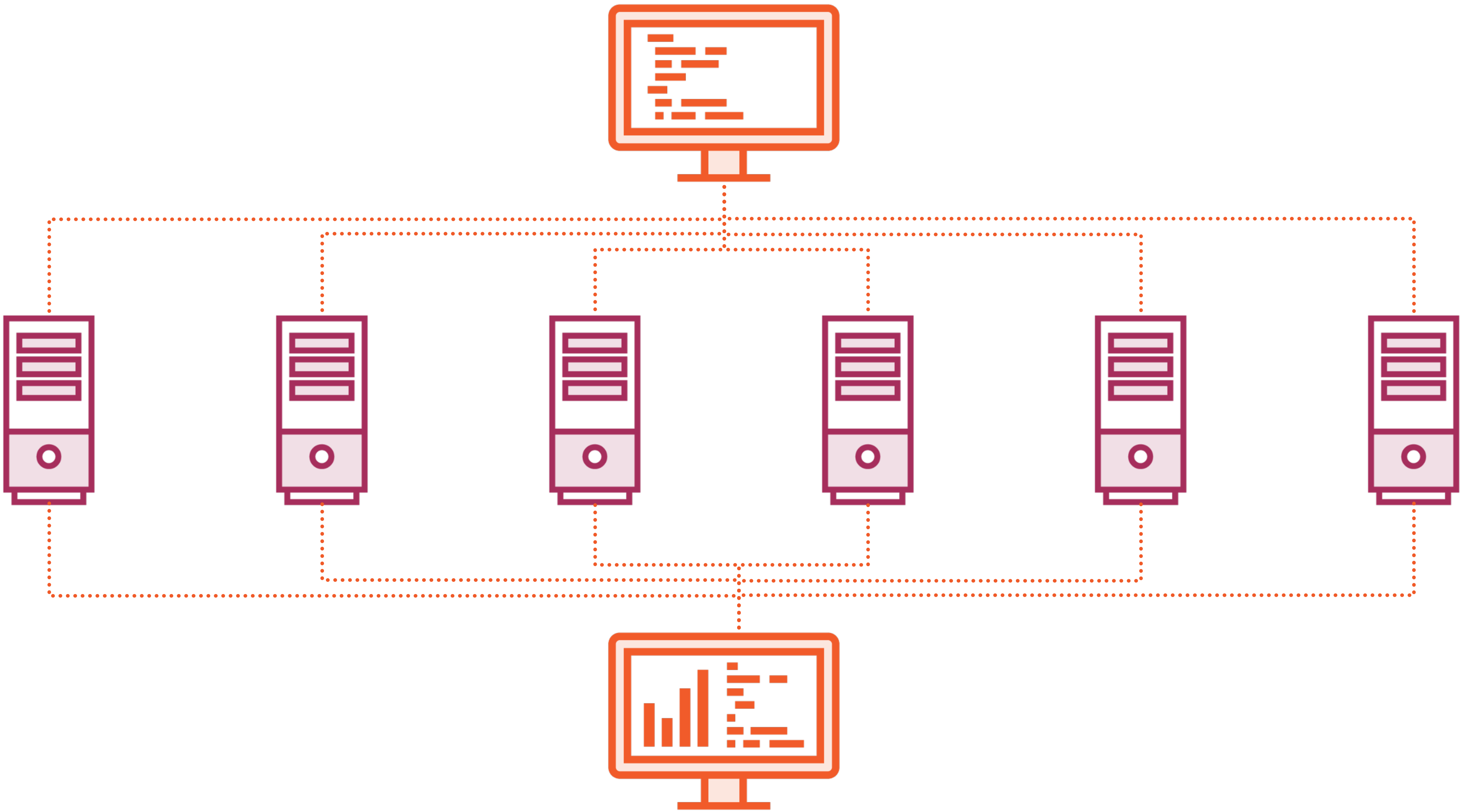
Broadcast

Use ComputeTask and ComputeJob When

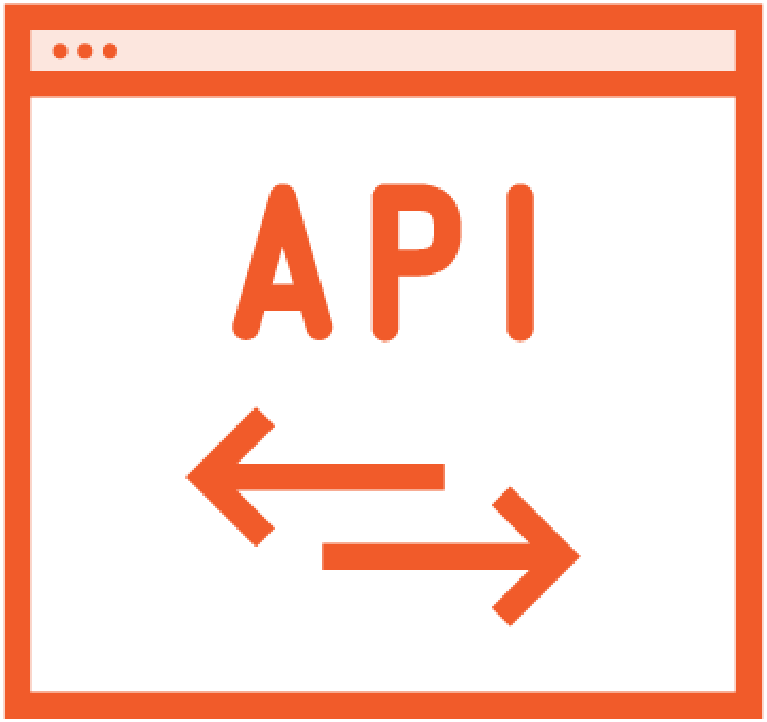


You need to control how jobs are mapped to nodes or define more complex reduce logic

You need to implement custom fail-over logic



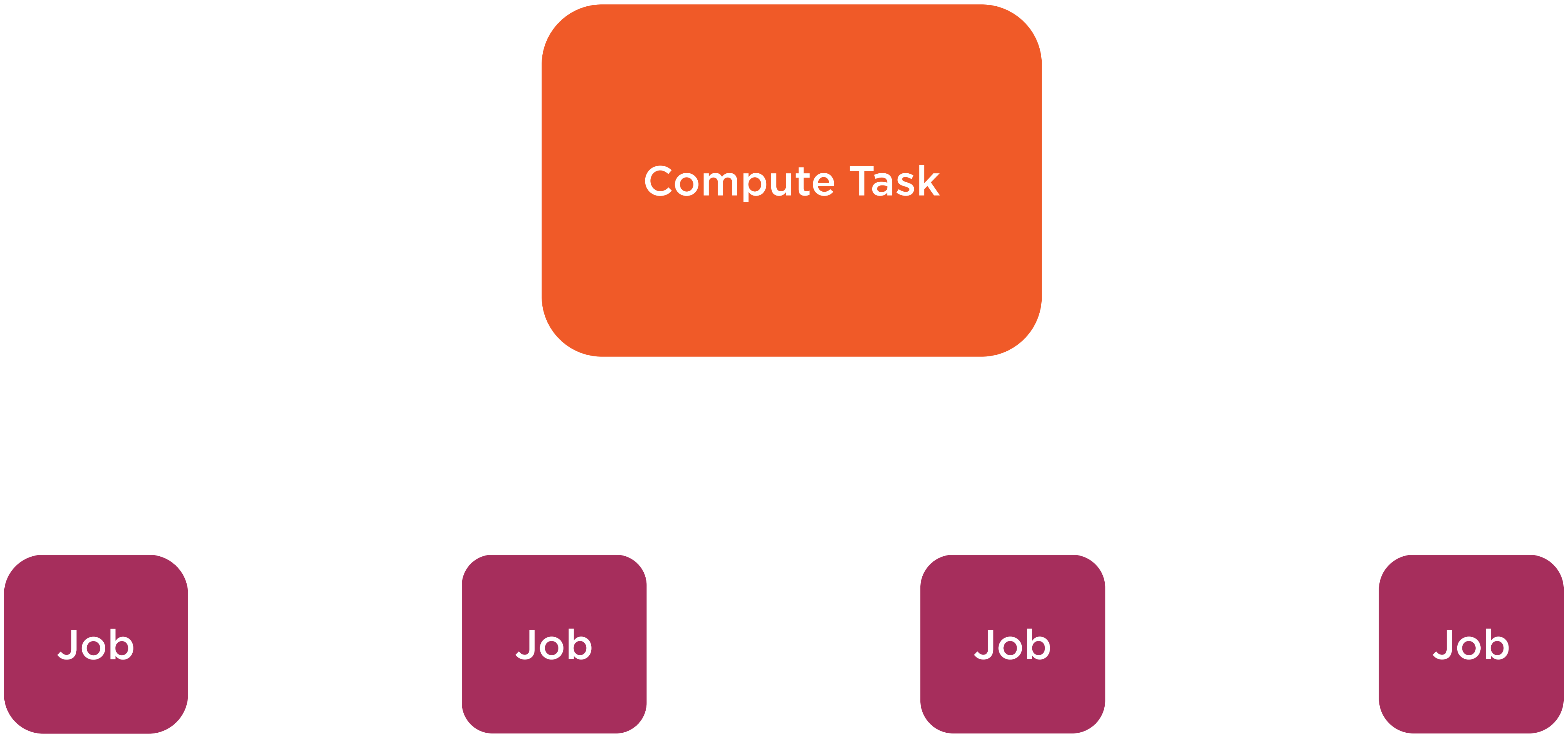
Apache Ignite's ComputeTask API



map()

reduce()

Compute Task



Compute Task

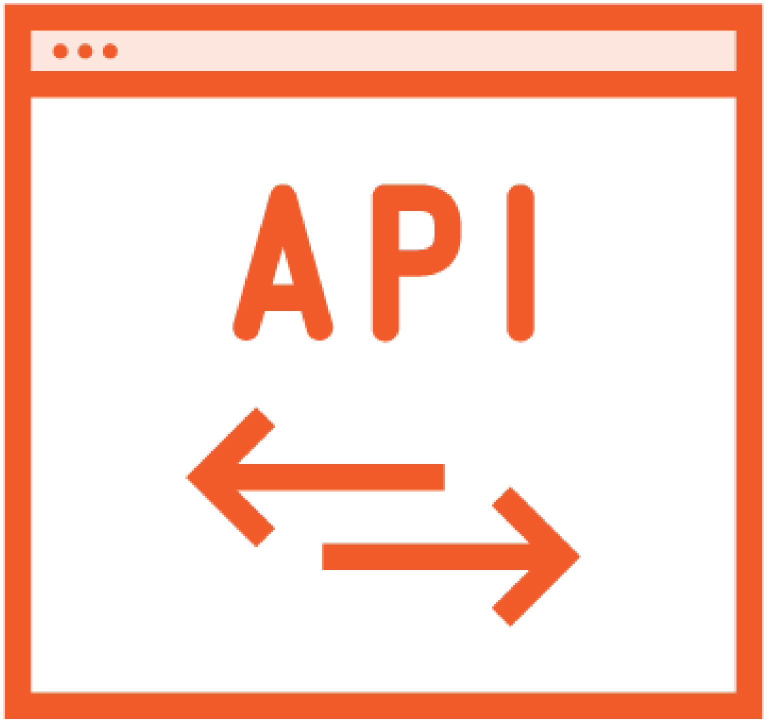
Job

Job

Job

Job

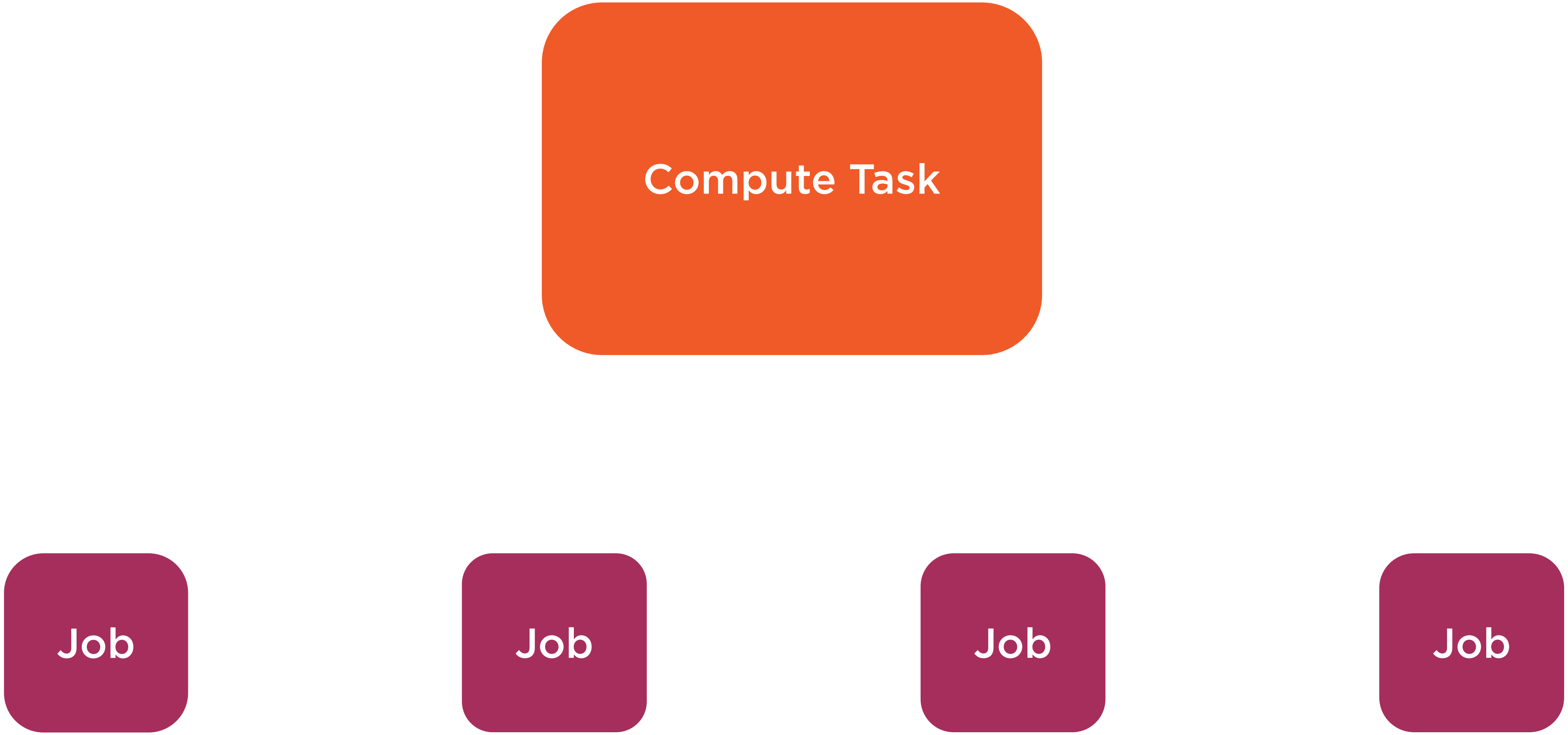
Apache Ignite's ComputeTask API



map()

reduce()

```
        <node,data>
        <node,data>
map(    <node,data>    )
        <node,data>
```



Compute Task

Job

Job

Job

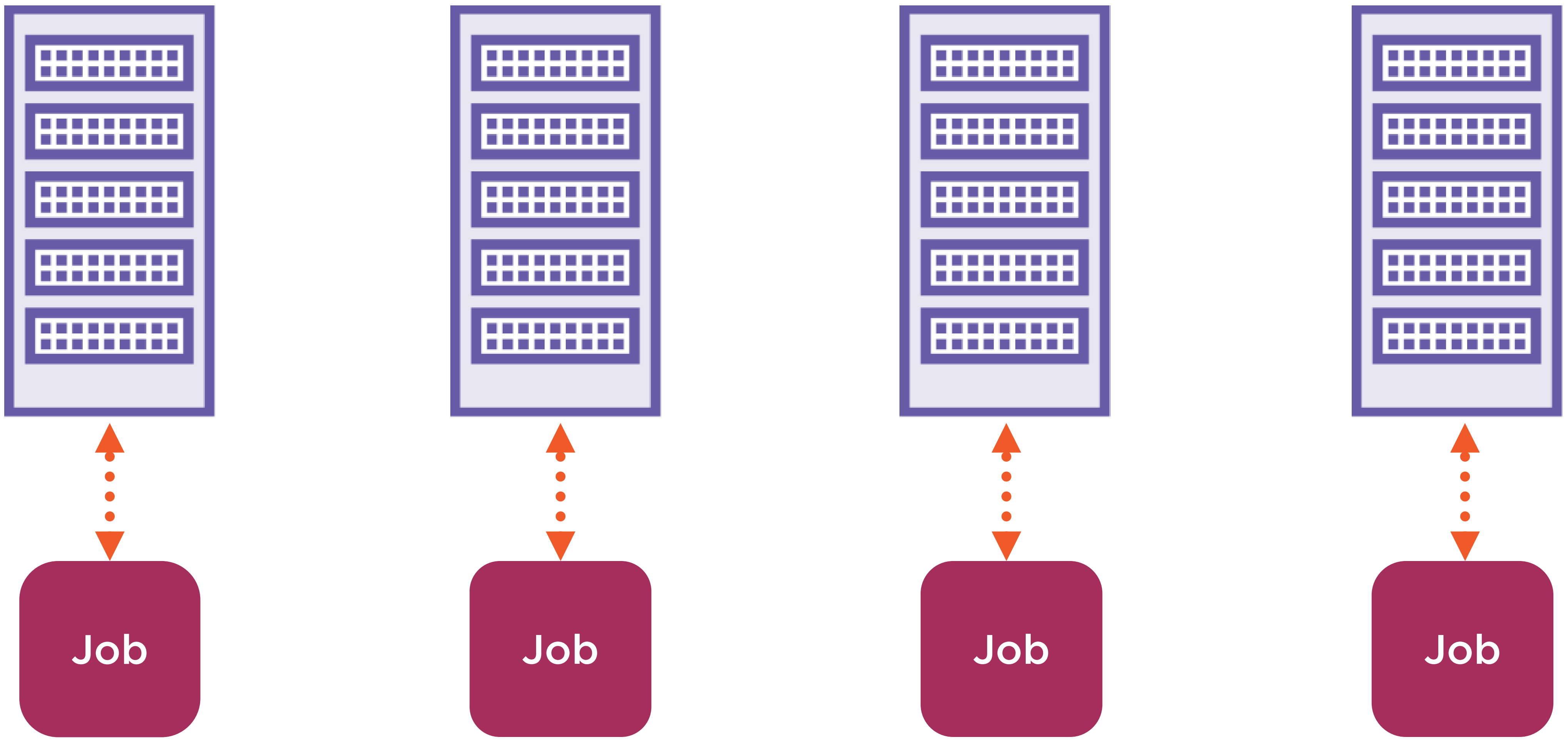
Job

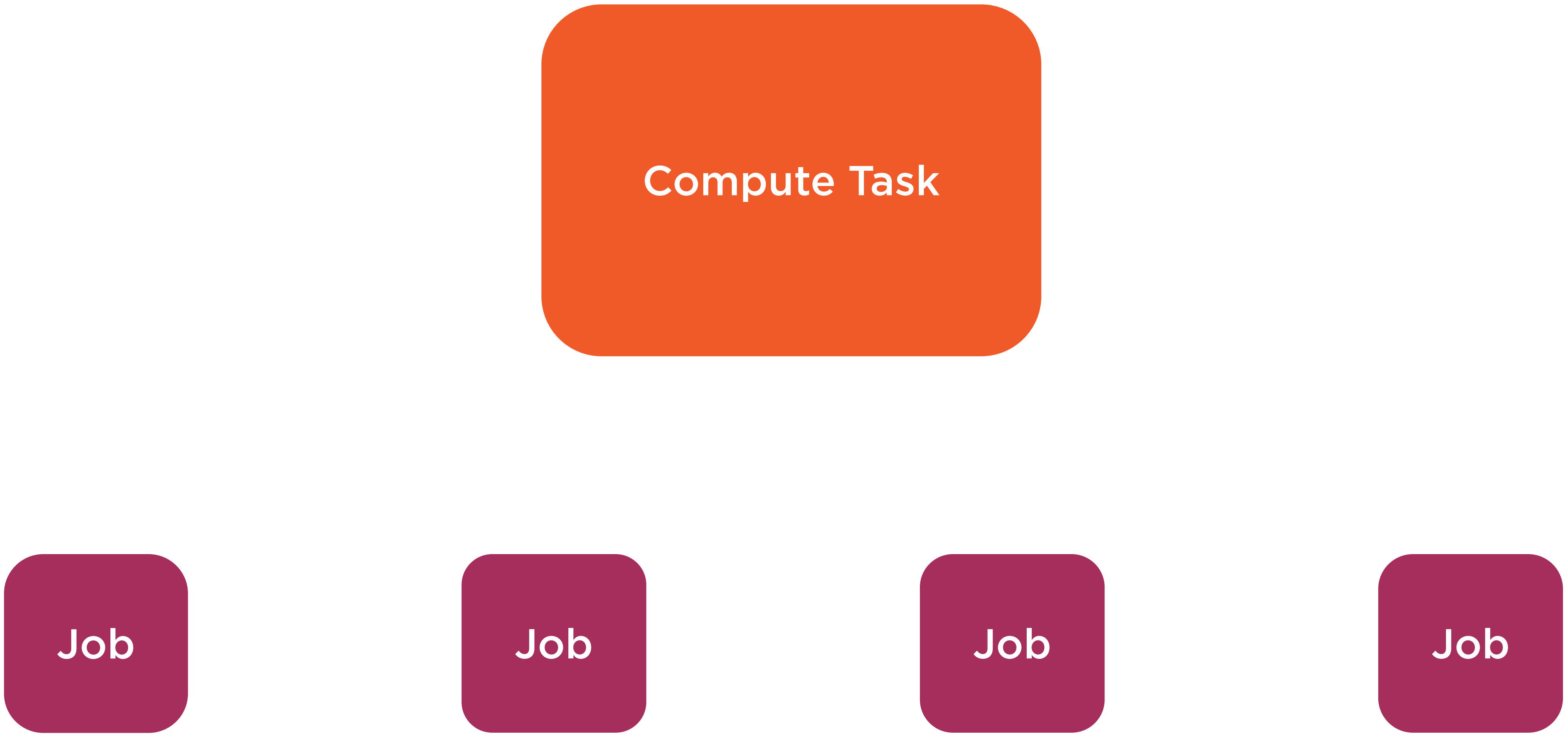
Job

Job

Job

Job


Job
Job
Job
Job



Compute Task

Job

Job

Job

Job

Compute Task

Implementations of ComputeTask

ComputeTaskSplitAdapter() {

- **Split()**
- **Reduce()**

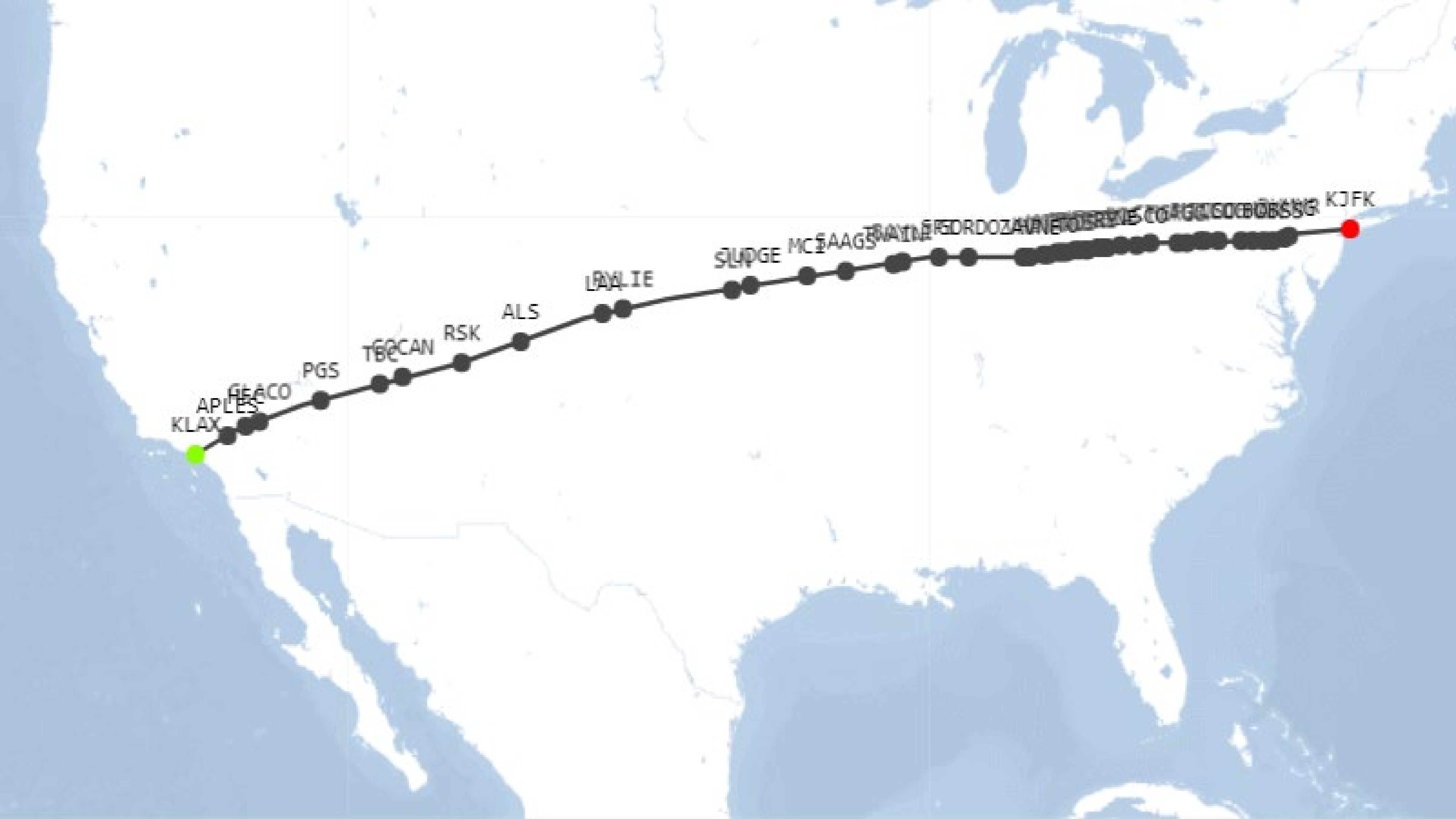
Implementations of ComputeTask

ComputeTaskSplitAdapter()

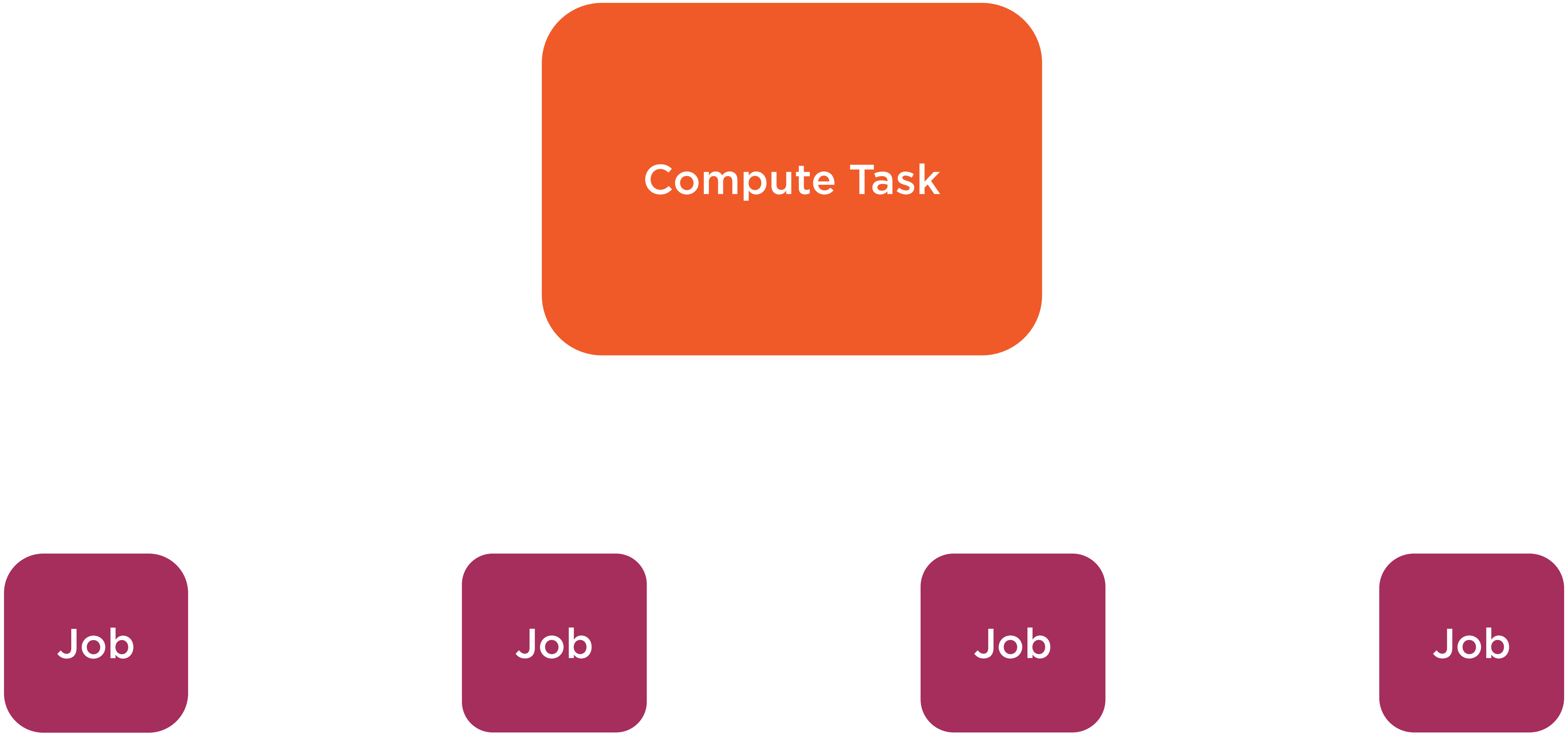
- Split()
- Reduce()

ComputeTaskAdapter()

- Map()
- Reduce()


KLAX
APLES
CLACO
PGS
TBC
GOCAN
RSK
ALS
LAA
RYLIE
SLN
JUDGE
MCI
SAAGS
KJFK

Compute Task



Compute Task

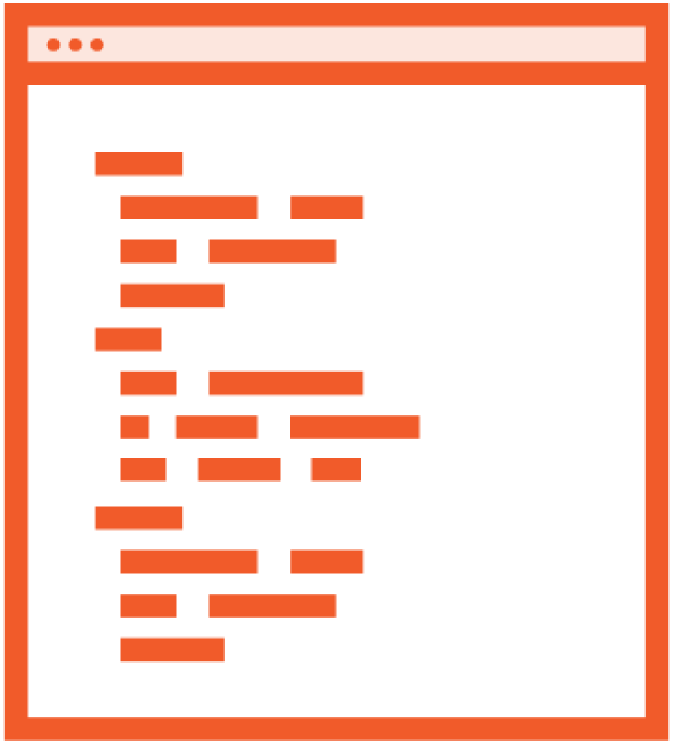
Job

Job

Job

Job

ComputeJobAdapter

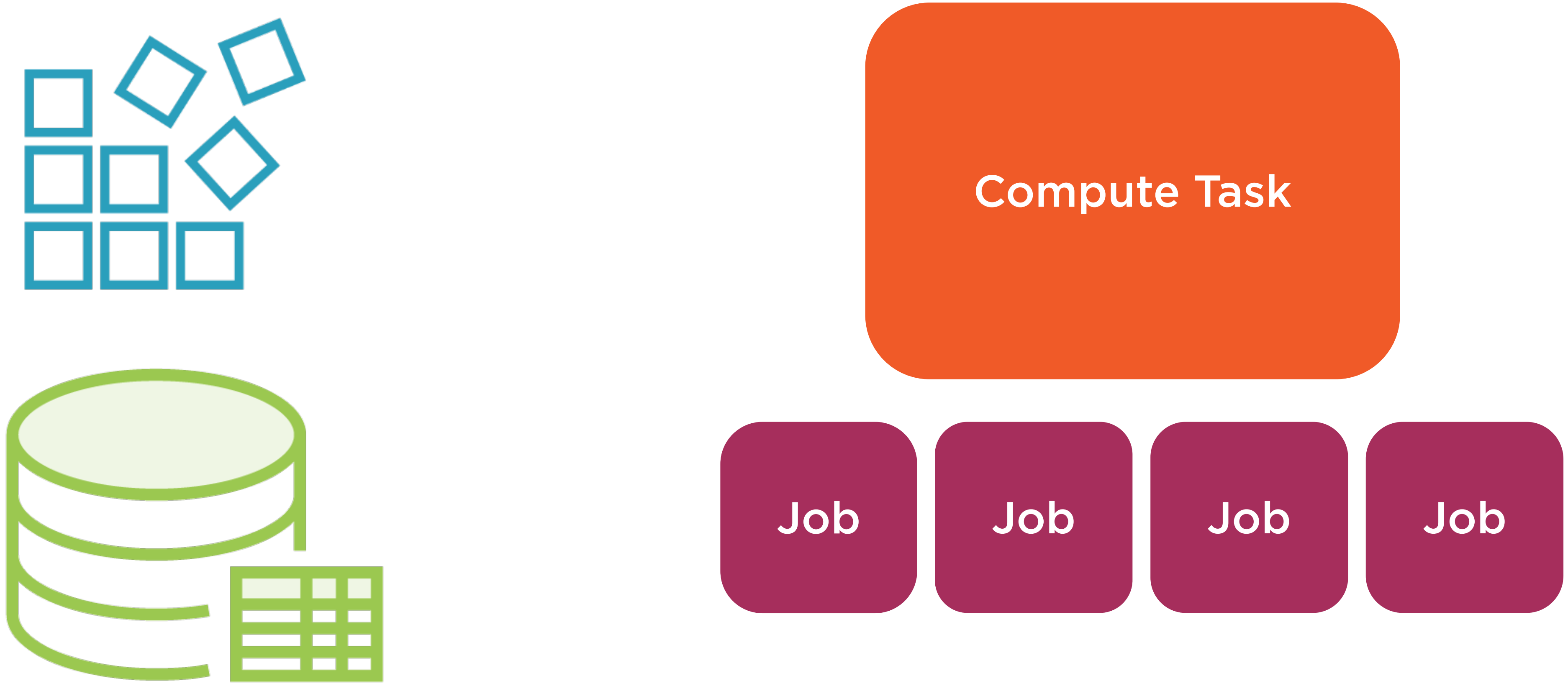


Is an abstract class

Exposes execute() and cancel() methods

Default ‘no-op’ cancel() method provided

Distributed Task Session



Compute Task

Job

Job

Job

Job

Flight

FlightId

Reservation

ReservationId
FlightId
PassengerId

Passenger

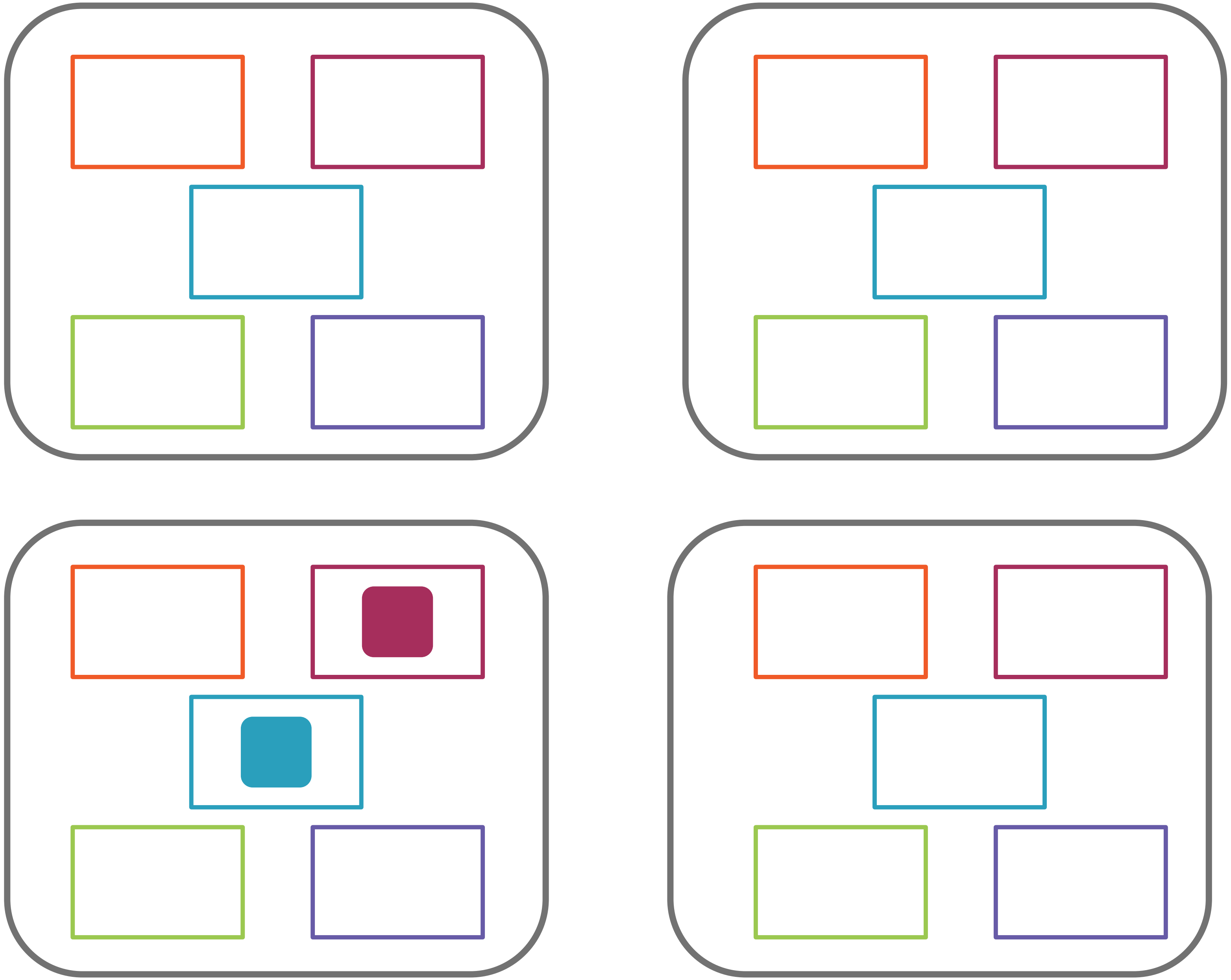
PassengerId
FreqentFlyterId

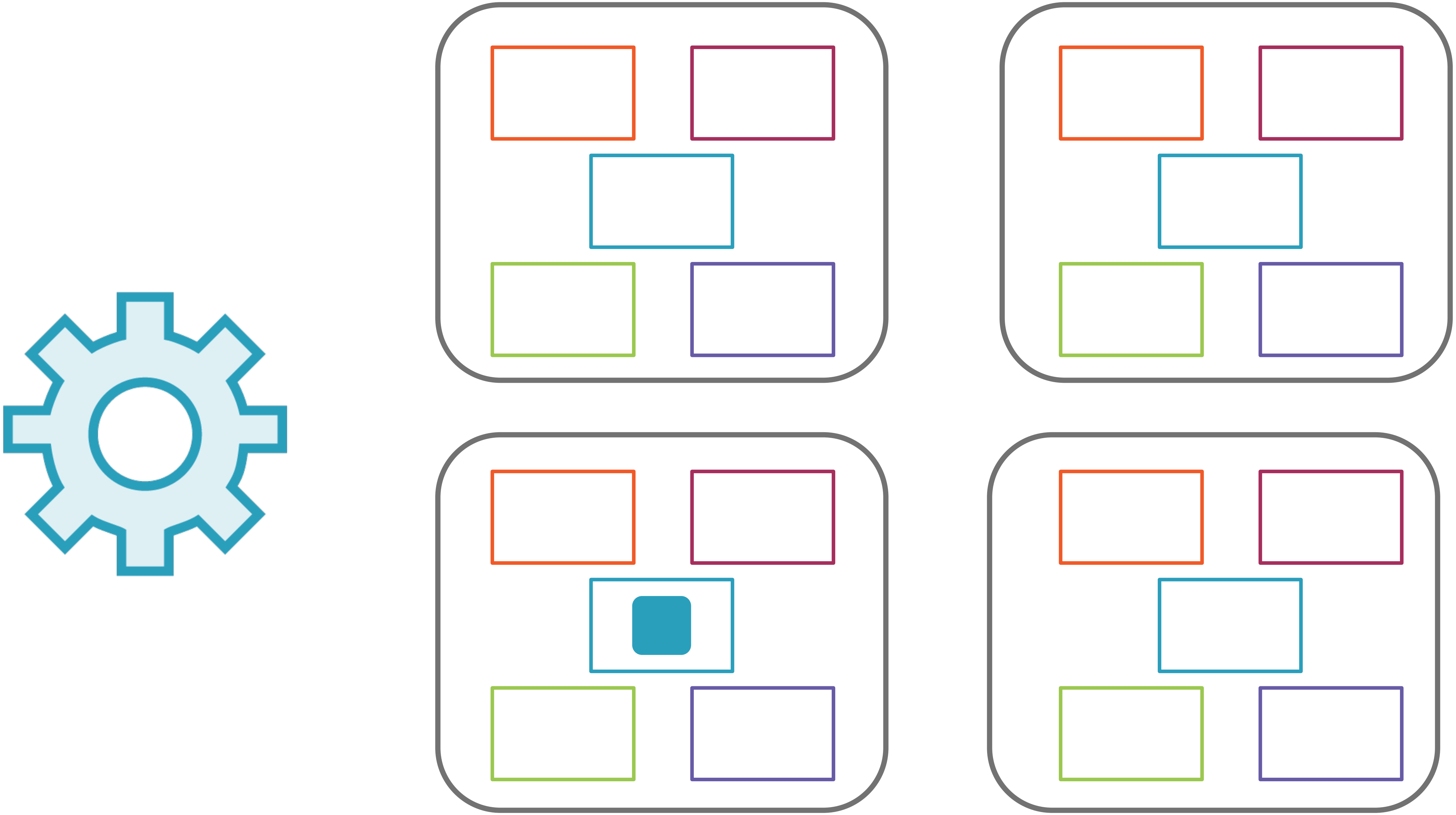
FQFL

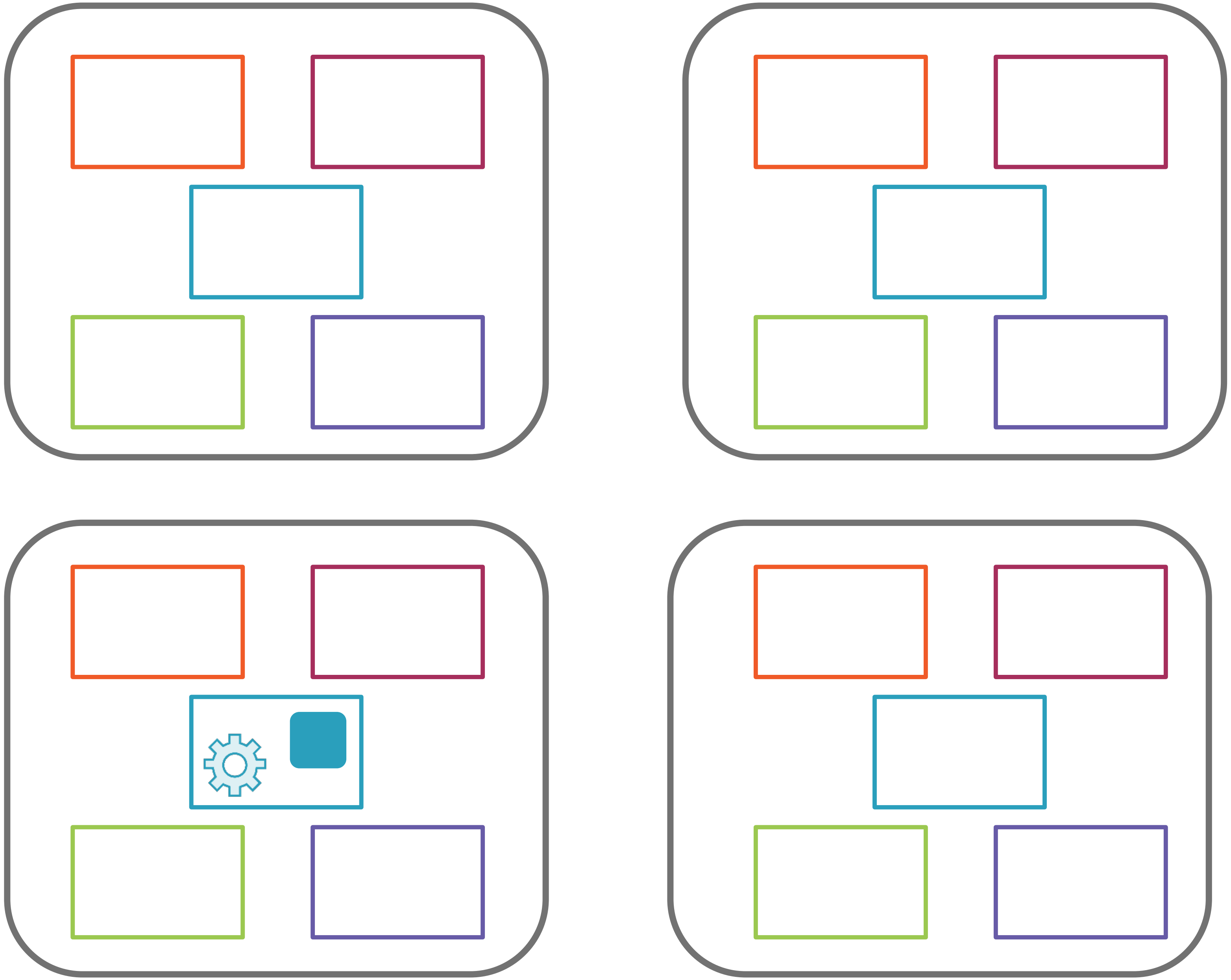
FrequentFlyerId

FFFH

FrequentFlyerId
FlightId







Task-Parallelism
(“Run” and “Call”)

Data-Parallelism
(“Apply”)

Broadcast

Affinity Run & Affinity Call

Cache Name 1.. ∞

IgniteRunnable<>()

Affinity Key

Partition Number

~~SQL Query~~

~~SQL Fields Query~~

Scan Query

~~Continuous Query~~

~~Text Query~~

Summary

Distributed execution of tasks

The “Compute API”

Task parallelism and Data parallelism

ComputeTask and ComputeJob

The “Distributed Task Session”

Compute collocation