

Pinball: Workflow System of Pinterest

Shuo Xiang / Jack Ye

Pinterest: The World's Catalog of Ideas

Data and Workflow in Pinterest

Scale@Pinterest

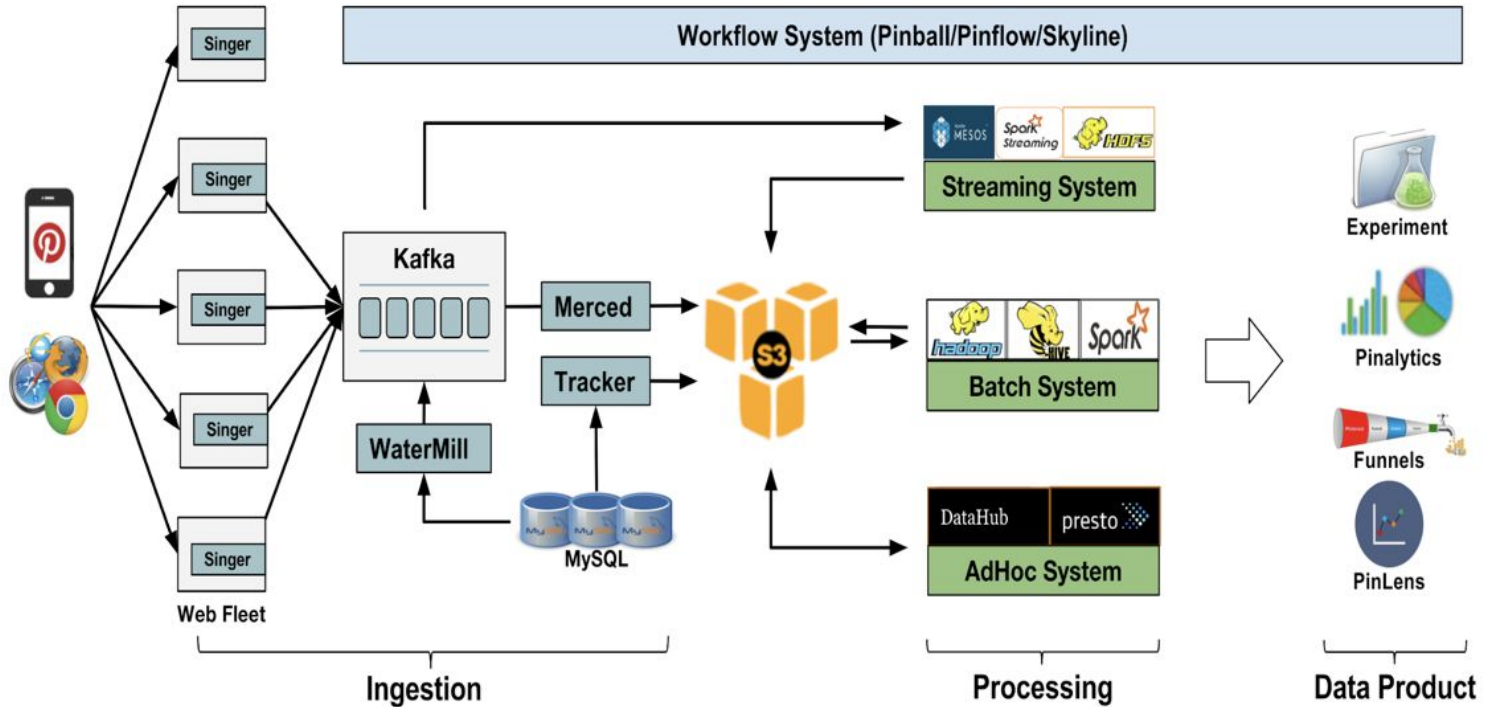
Run on the Cloud(AWS) Business Scale.

- 200M+ MAUs
- Hundred billions of Pins
- Billions of Boards

Data Scale

- 200+ PB @ S3
- 6000+ Hive/Hadoop nodes
- 400+ Presto nodes

Data@Pinterest



Workflow System

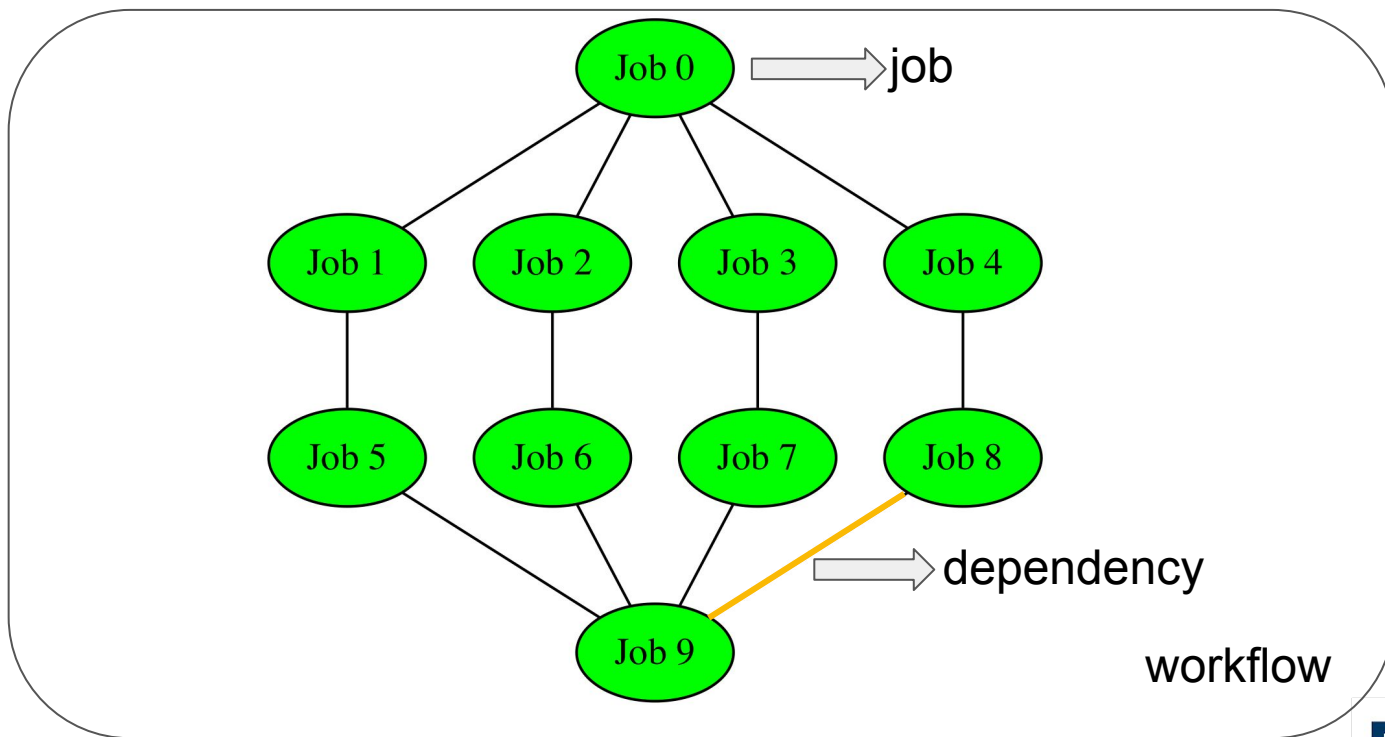
- Workload
 - 5000+ workflows
 - 30,000+ jobs
- Computation types
 - Hadoop, Hive, Spark ...
 - Cascading, Scalding ...
 - General script

Workflow use

- External/internal metrics reporting
- ETL for data warehouse
- Search, recommendation
- ML pipelines

Pinball: Pinterest's Workflow System

Notations



Framework

- Composer to define workflows
- Scalable scheduler
- Multi-platform support
 - Local,
 - Hadoop/Yarn,
 - Presto
 - Mesos, K8s etc.
- Operate, report, monitor and alert

User Features

- Rich job templates
 - Hadoop/Hive/Spark/Local job template
 - Condition and waiter job
- Workflow overrun policy
- Job execution record
 - Automatic retry & backfill
 - Manual mark/erase
 - x-workflow and x-cluster dependency

Workflow Overrun

Decisions for multiple workflow instances

- skip
- delay
- abort
- start_new

1535599570053	RUNNING	20 hrs 1 min 24 secs
1535513177018	SUCCESS	20 hrs 13 mins 1 sec
1535426758978	SUCCESS	18 hrs 18 mins 9 secs
1535340351179	SUCCESS	19 hrs 2 mins 32 secs
1535253984604	ABORTED	1 day 8 secs
1535167542572	SUCCESS	19 hrs 37 mins 41 secs

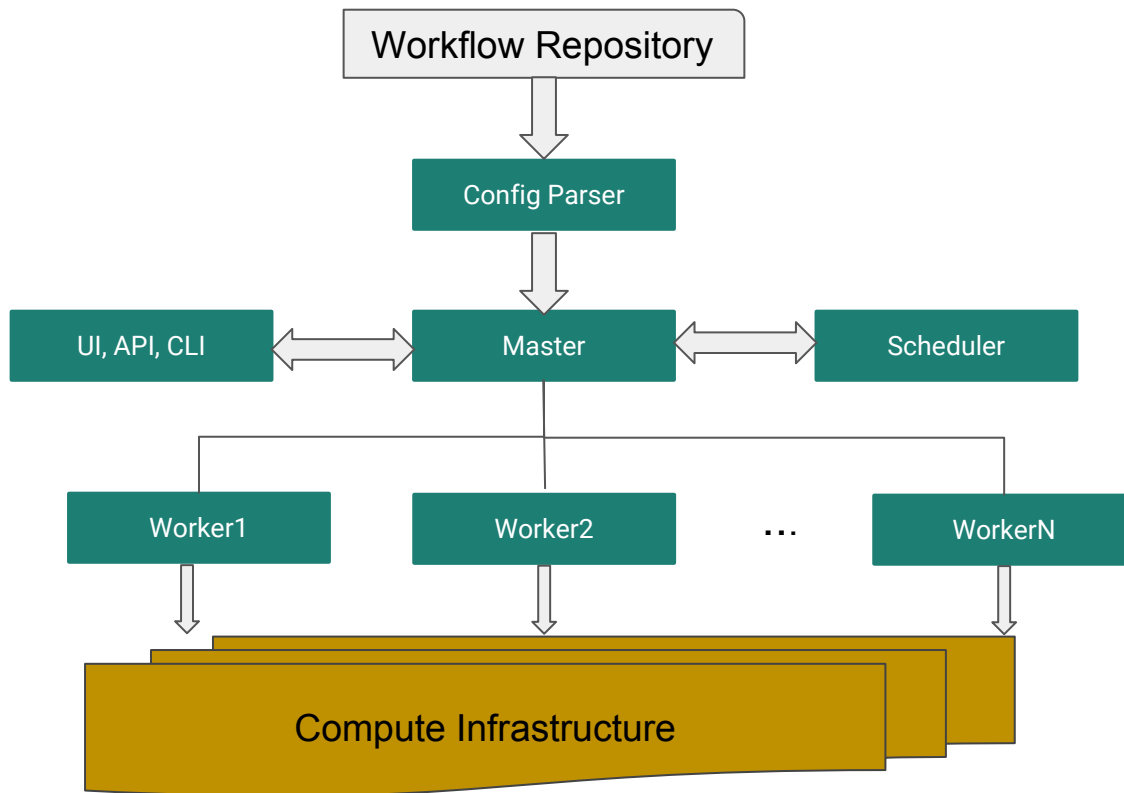
Job Record

- External storage component
- Automatic retry & backfill
- Manual mark/erase
- X-workflow and X-cluster dependency

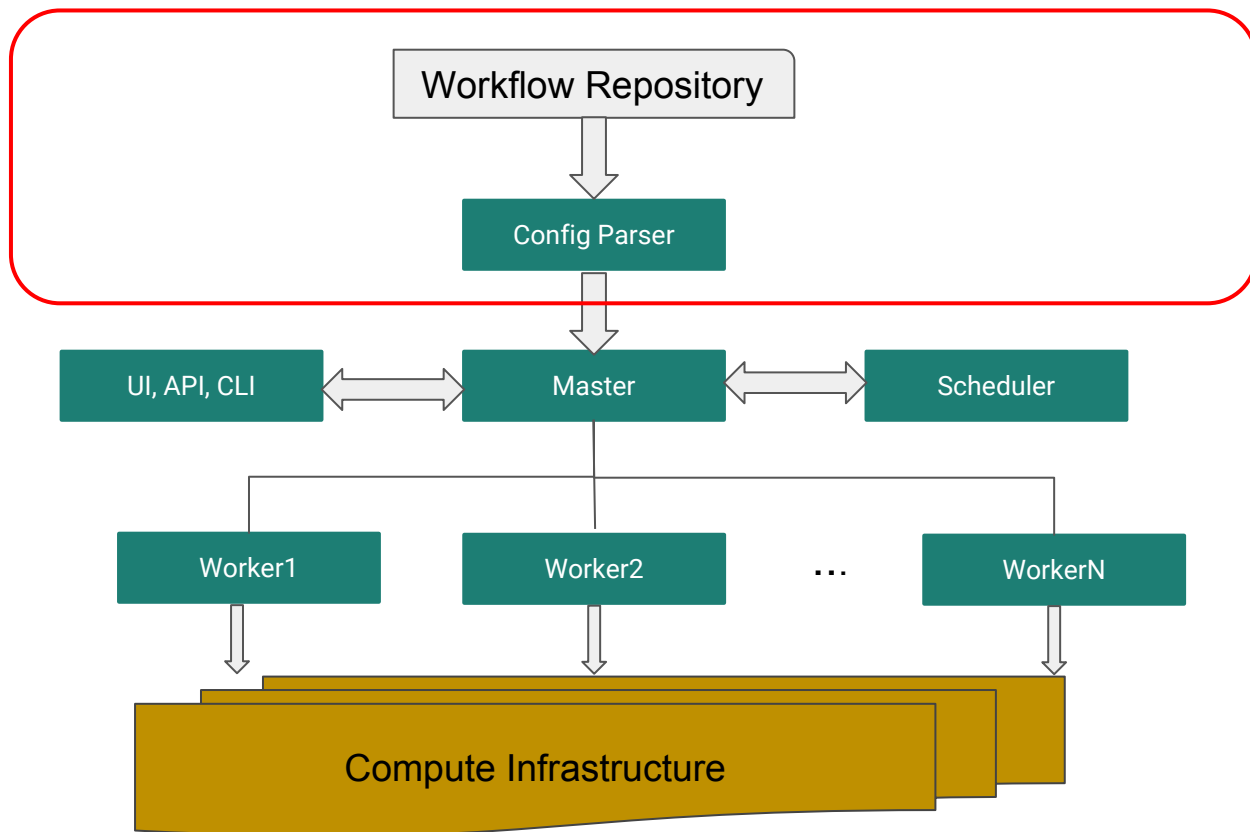
Design Choices

- Composer scheduler isolation
- Pluggable job/task
- Extensible

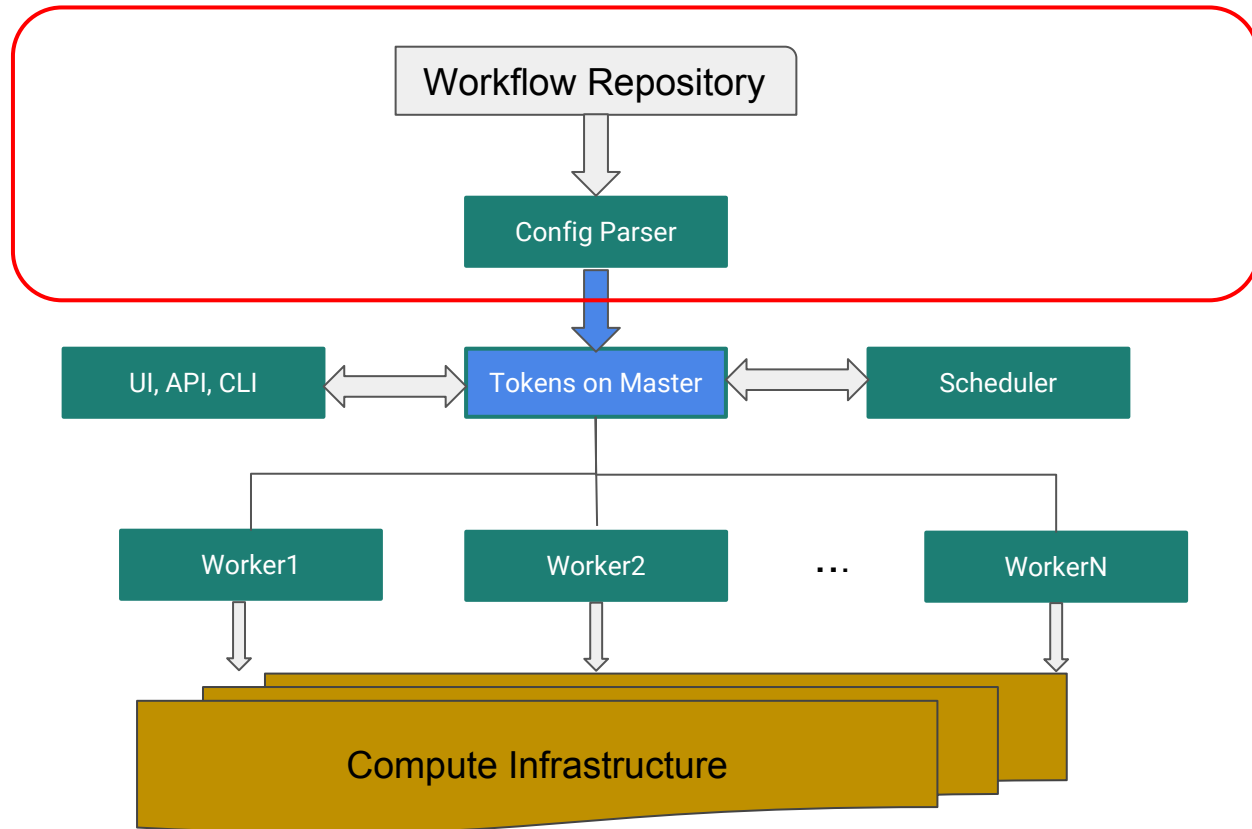
Architecture



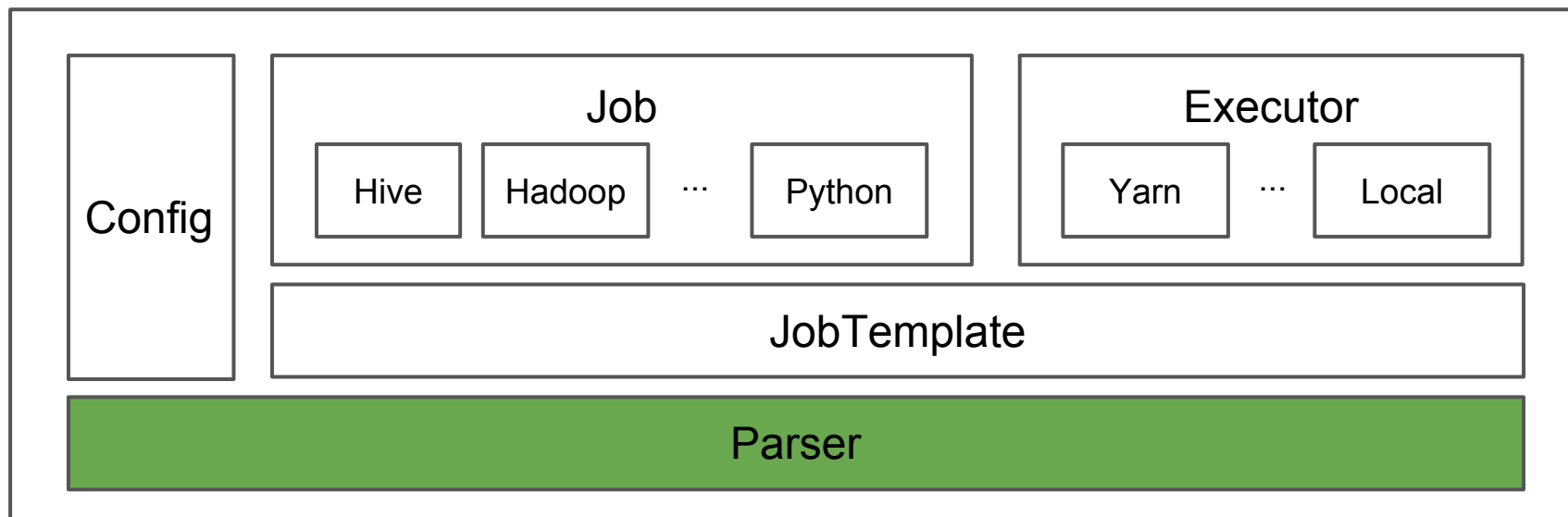
Multi-composer



Multi-composer



Multi-composer



Pinball Scheduler

config.yaml

```
parser:  
parser params:
```

Example composer #1

```
class ExamplePythonJob(data_job.PythonJob):  
    _OWNERS = 'ldap_user_a'  
  
    def _execute(self):  
        print('this is a PythonJob')
```

Example composer #1

```
class ExampleHiveJob(HiveJob):
```

```
    # True for date range, False for individual dates
```

```
    _MULTIDATE_EXECUTE = False
```

```
def _setup(self):
```

```
    super(data_job.HiveJob, self)._setup()
```

```
    self.params['k1'] = 'v1'
```

```
    self.params['k2'] = 'v2'
```

```
_QUERY_TEMPLATE = """SELECT ... %(k1)s %(k2)s"""
```

Example composer #1

```
class ExampleHadoopJob(data_job.HadoopJob):
    def _setup(self):
        self._output_path = 's3://path/to/output/data'
        self.jobconf_args['mapred.job.name'] = str(self._get_class_name())
        self.jobconf_args['mapred.reduce.tasks'] = 200
        self.jobconf_args['cascading.combine_input_max_size'] = 3 * 1024 ** 3
        self.jobconf_args["mapred.output.compress"] = 'true'
        self.jobconf_args["mapred.output.compression.codec"] = 'org.apache.hadoop.io.compress.SnappyCodec'
        super(ExampleHadoopJob, self)._setup()

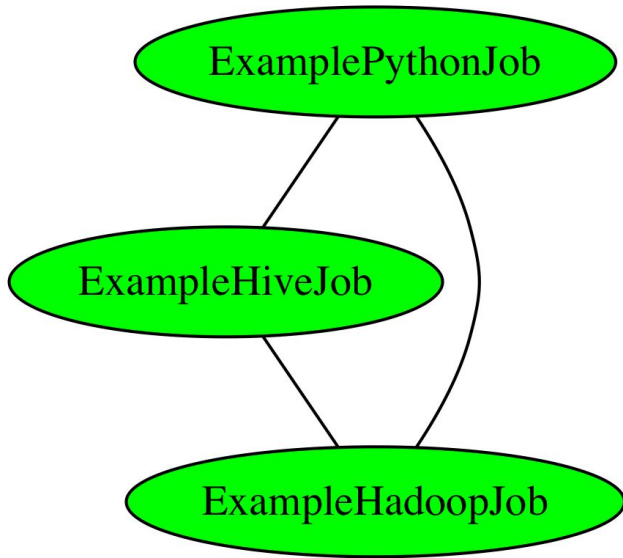
    def _get_class_name(self):
        return "com.pinterest.hadoop.businessA.JobDefA"

    def _execute(self):
        s3_utils.rm(self._output_path)
        super(ExampleHadoopJob, self)._execute()

    def _complete(self):
        # some post execution logic
```

Example Composer #1

```
EXAMPLE_WORKFLOW = {  
    LocalConditionTemplate('ExamplePythonJob'): [],  
  
    HadoopJobTemplate(  
        name='ExampleHiveJob',  
        job_full_class='pinterest.example.ExampleHiveJob'  
    ): ['ExamplePythonJob'],  
  
    HadoopJobTemplate(  
        name='ExampleHadoopJob',  
        job_full_class='pinterest.example.ExampleHadoopJob'  
    ): ['ExamplePythonJob', 'ExampleHiveJob'],  
}
```



Example Composer #2

Skyline

Workflow Dashboard Workflow Composer **Job Composer** Backfill Jobs

Workflow Name *

AdEngMetricsWorkflow

Job/Condition Name *

WaitForAdEngMetricsTable

Job/Condition Type *

HIVE_QUERY_CONDITION

Parent Job/Condition(s)

Owners *

edmarc@pinterest.com

Advanced Options

Query *

```
1 set mapred.min.split.size.per.node=1024000000000;
2 set mapred.min.split.size.per.rack=1024000000000;
3 set mapred.max.split.size=1024000000000;
4
5 SELECT true FROM
6   ad.ad_eng_metrics
7 WHERE
8   the_dt='%(end_date)s' LIMIT 1
```

Example Composer #2

Skyline

[ETL Manager](#) [Data Warehouse](#)

[Workflow Dashboard](#) [Workflow Composer](#) [Job Composer](#) [Backfill Jobs](#)

Workflow Name *

AdEngMetricsWorkflow

Job/Condition Name *

WaitForAdEngMetricsTable

Job/Condition Type *

HIVE_QUERY_JOB

Parent Job/Condition(s)

WaitForAdEngMetricsTable

AdsEngOKRPanalyticsSegmentJob

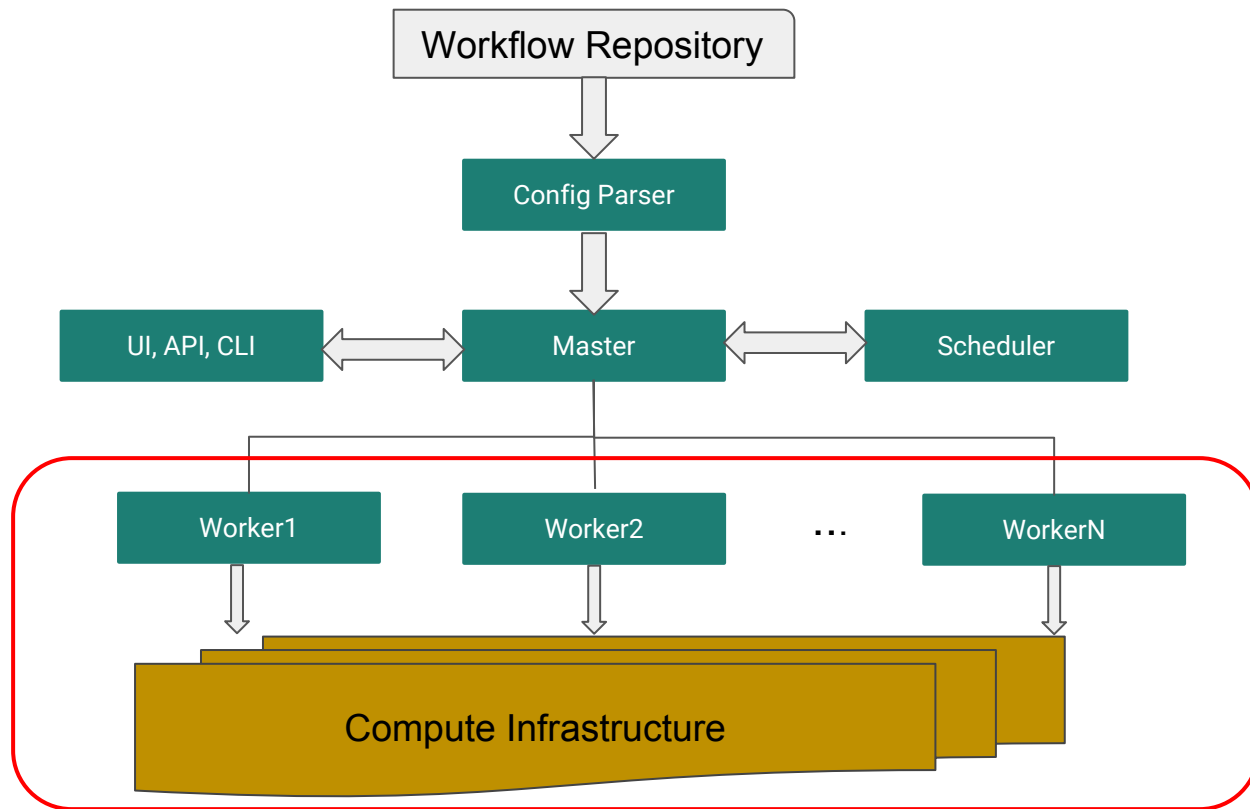
Owners *

yourteam@pinterest.com

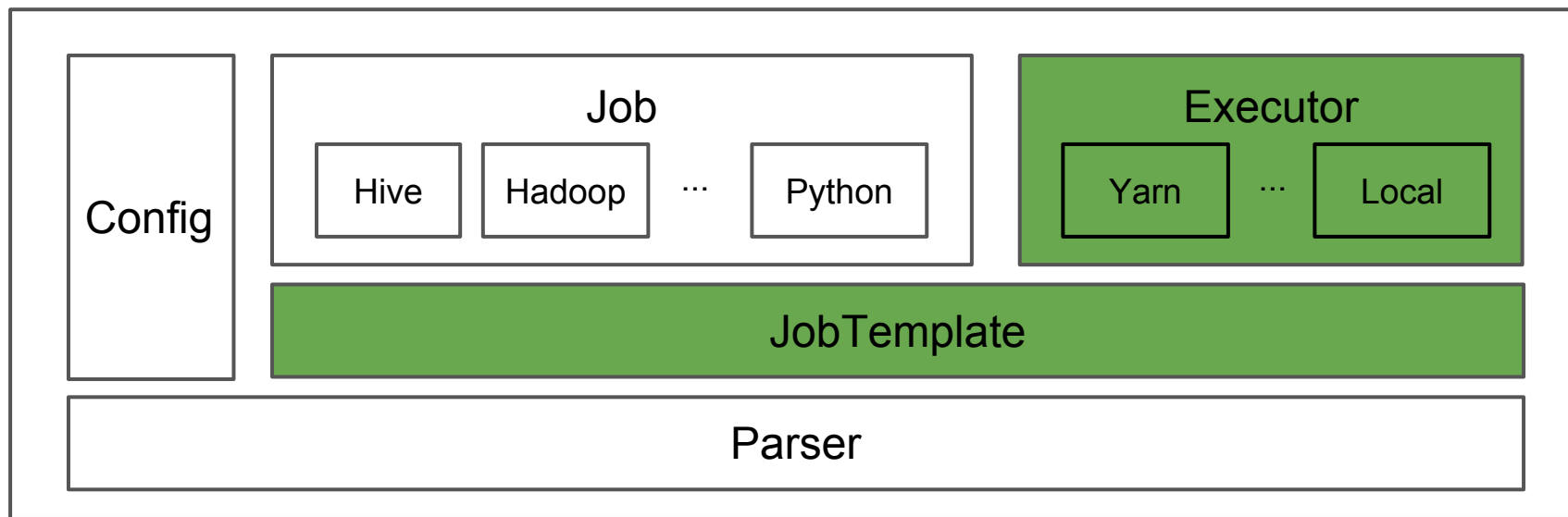
Create a Job/Condition

Check SVG Graph

Multi-platform



Multi-platform



Pinball Scheduler

config.yaml

```
parser:  
parser params:
```

Pinball Master

Service for

- composer-independent tokens
- workflow and job status
- scheduling status
- user interactions

Pinball Worker

- Claims runnable jobs from master
- Applies lease on job for fault-tolerance
- Runs general job submission command
- Scales up by increasing parallelism

Summary

- Flexible workflow system supports multiple composers and platforms
- Distributed master-worker design
- Rich user features and extensible

Upcoming features

- Enhanced composer
 - Release process (test/canary/prod)
 - Resource abstraction
 - Dynamic versionized workflow
- Improved execution isolation

We are hiring !



Jack Ye wye@pinterest.com

Shuo Xiang sxiang@pinterest.com



THE LINUX FOUNDATION **OPEN SOURCE SUMMIT**