

Automating Netflix ML Pipelines With Meson

| QCon SF 2017 | Eugen Cepoi, Davis Shepherd

NETFLIX



NETFLIX

Browse ▾

Kids

DVD

Search



OneRingTo... ▾

NETFLIX ORIGINAL
STRANGER THINGS

98% Match 2016 TV-14 1 Season

Season 2 Coming October 27

When a young boy vanishes, a small town uncovers a mystery involving secret experiments, terrifying supernatural forces and one strange little girl.

Starring: Winona Ryder, David Harbour, Finn Wolfhard,

Millie Bobby Brown, Gaten Matarazzo

Creator: The Duffer Brothers

Genres: Halloween Favorites, Scary Halloween Favorites, TV Shows



MY LIST



OVERVIEW

EPISODES

TRAILERS & MORE

MORE LIKE THIS

DETAILS

Popular on Netflix



Goal

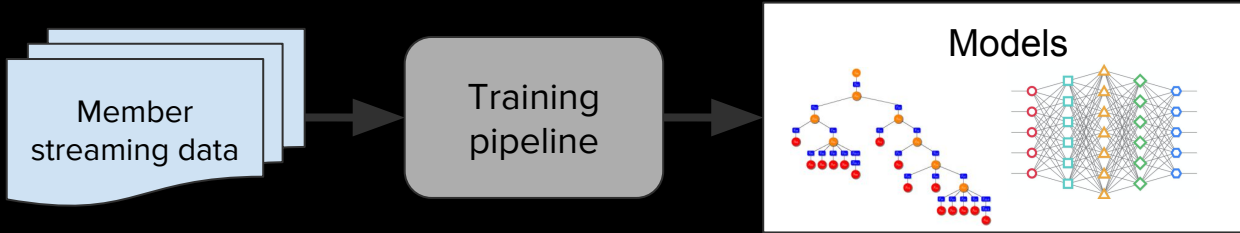
Create a **personalized experience** to help members find content to **watch** and **enjoy**

Recommendation Context

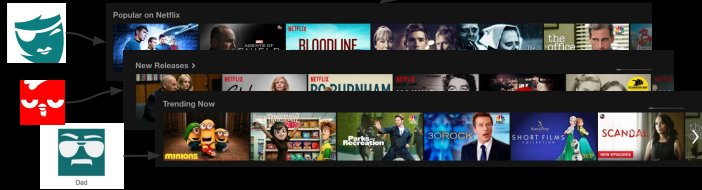
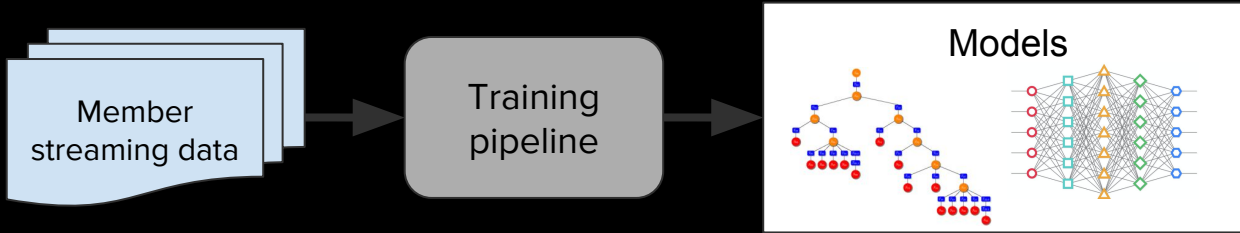
Member
streaming data



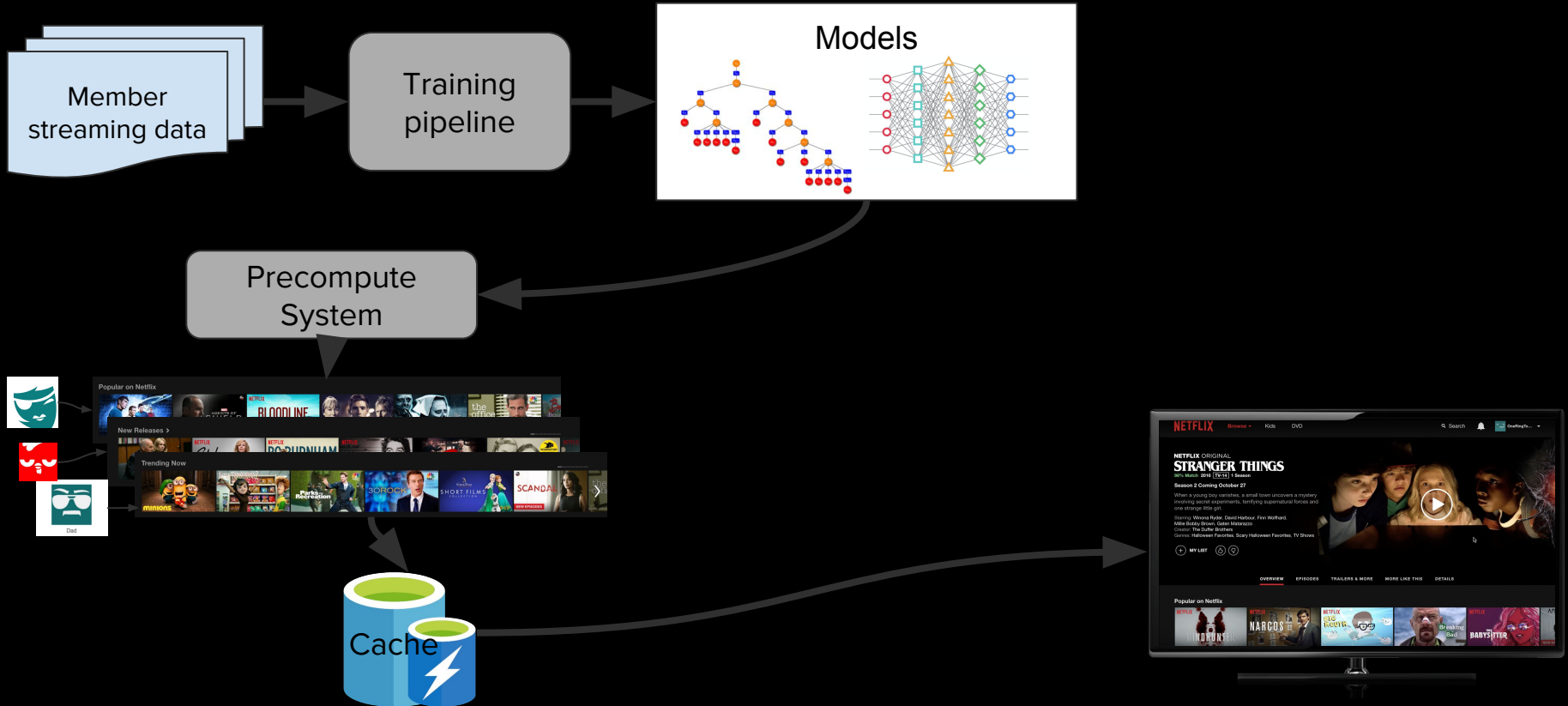
Recommendation Context



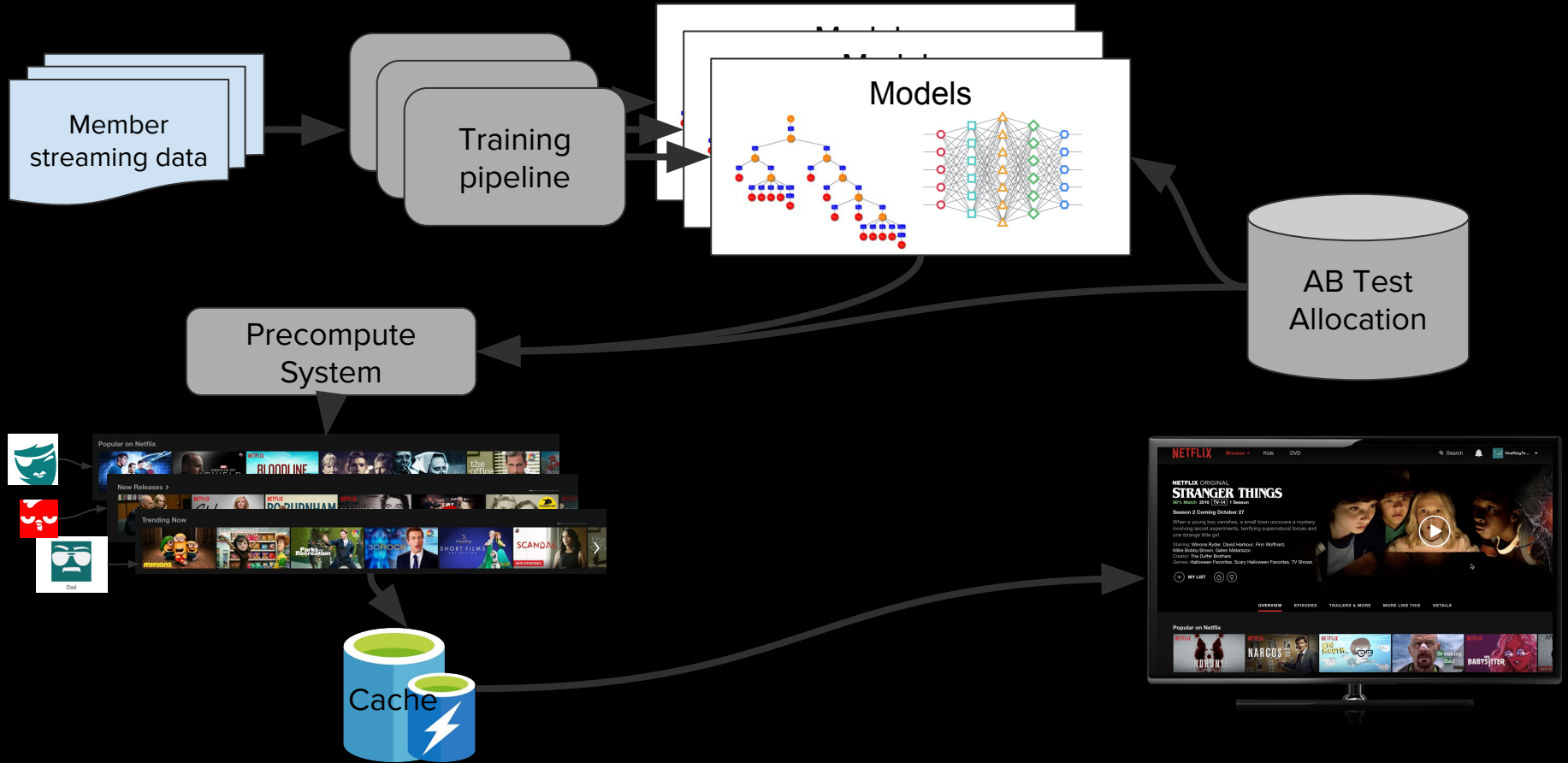
Recommendation Context



Recommendation Context

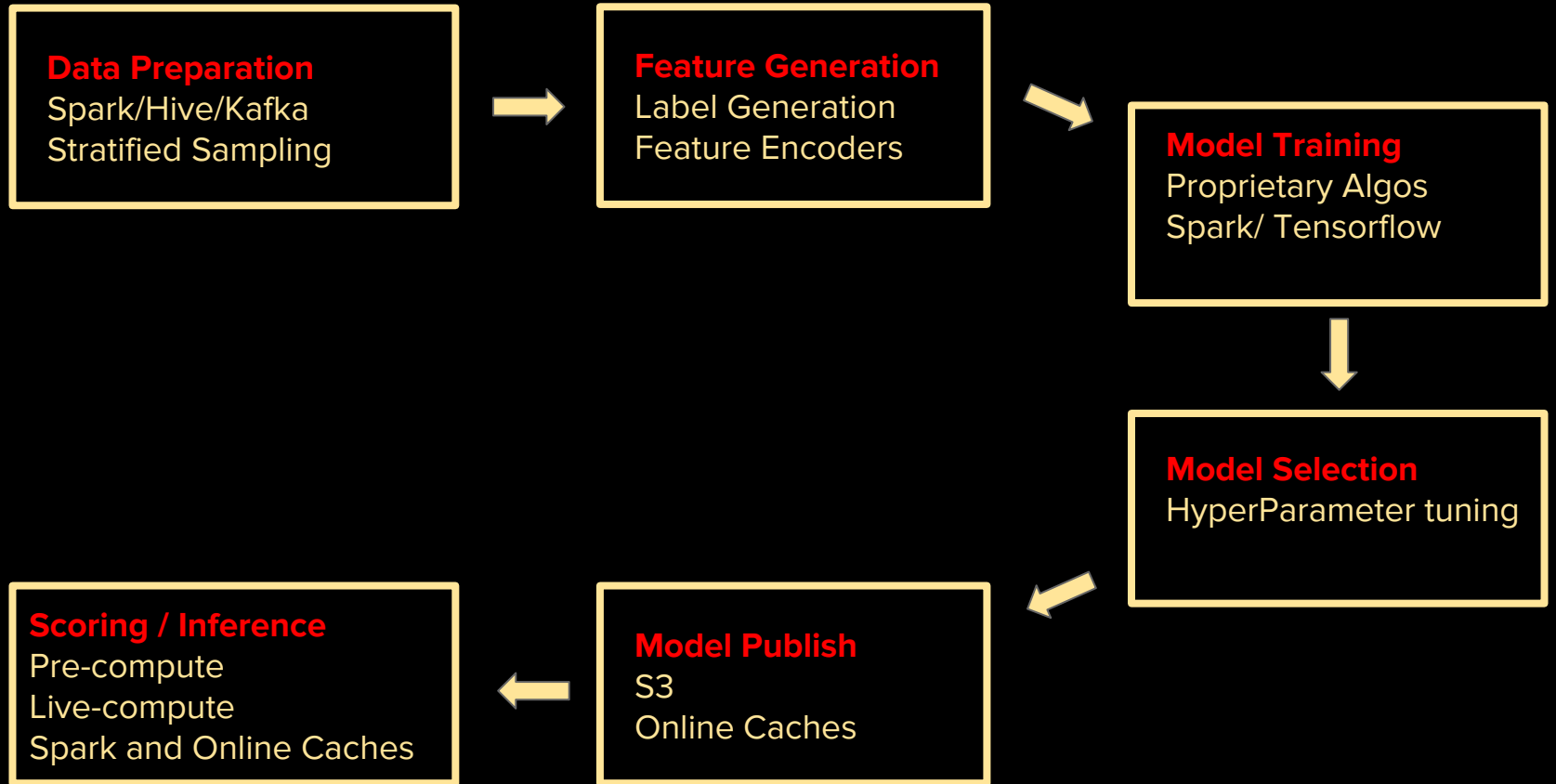


Recommendation Context



Innovation is driven by experimentation

Training Pipelines



Before Meson

- A collection of operators
- Little to no orchestration
- Often limited to single machine

Desired Properties

- Support Heterogeneous systems
- Highly flexible generic orchestration
- Handle failures
- Provide Reproducibility
- Support Multi-tenancy
- Support External Triggers

Why didn't an out of the box solution work?

- Spark and Scala support was paramount
- Options available didn't have the flexibility and scalability that we needed

Meson overview

Meson Overview

General purpose workflow orchestration engine

Delegates execution to Mesos

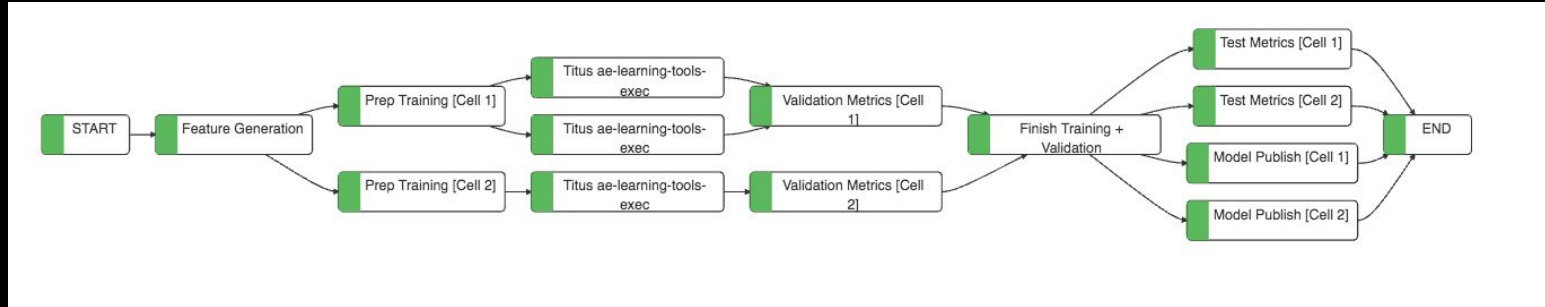
Initially built for Machine Learning pipelines for personalization

Supports complex workflow patterns (branching, loops, foreach)

Concepts

Workflow Directed Graph of steps, global parameters, triggers...

Step Describes a job and its configuration



Defining workflows

Scala DSL

Python DSL

UI

REST API

```
val sparkStep = Spark(  
  jobClass = "netflix.MySparkJob",  
  sparkArgs = Seq("--driver-memory" -> "8g")  
)  
  
val cpStep = DistCp(from = "...", to = "...")  
  
Workflow(  
  id = "my-cool-workflow",  
  triggers = Seq(Trigger.CRON("0 0 0 1/1 * ? *")),  
  notification = Notification(  
    whenStart = true, whenFinish = true, whoCustom = Option("me@gmail.com")  
  ),  
  parameters = Seq(Parameter.STRING(name = "country", value = "US")),  
  definition = sparkStep.sequence(cpStep).end()  
)
```

Parameters

Used to configure steps, job arguments, and step transitions

MVEL expression to derive parameter values at runtime

Predefined macros

```
Workflow(  
  id = "my-cool-workflow",  
  parameters = Seq(Parameter.LONG(name = "dateint", expression = Option("tsToDateInt(RUN_TS)")),  
  definition = Step(  
    name = "jobX",  
    parameters = Seq(  
      Parameter.LONG(name = "day", expression = Option("dayFromDateInt(dateint)"))  
    )).end()  
  )  
)
```

Workflow patterns

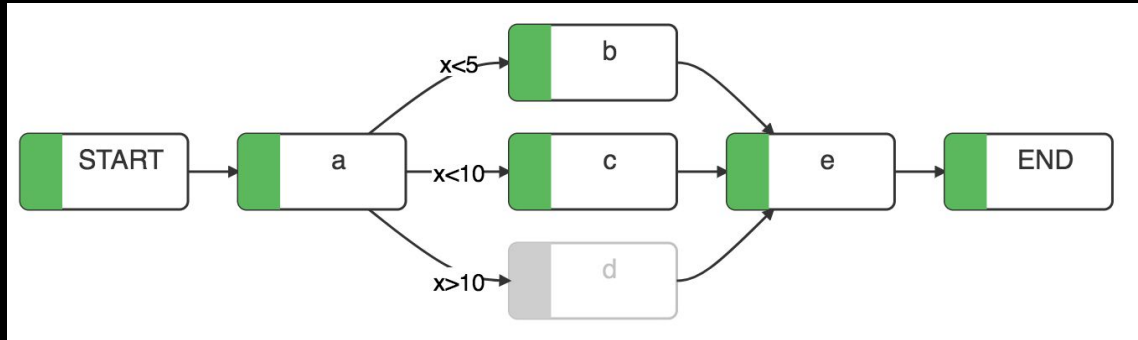
Branches (OR, XOR, AND)

Loops with XOR

Foreach

```
Step("a").or(  
  Condition("x<5").branch(Step("b")),  
  Condition("x<10").branch(Step("c")),  
  Condition("x>10").branch(Step("d"))  
).sequence(Step("e"))
```

Using parameters and
MVEL



Data artifacts

Data artifact defined by a name and a set of partitions (parameters)

Cross workflow
dependencies

External triggers

```
Step(  
  name = "someJob",  
  inputs = Seq(  
    DataArtifact(  
      name = "views",  
      params = Seq(Parameter.STRING(name = "country", value = "US"))  
    )  
  ),  
  outputs = Seq(  
    DataArtifact(  
      name = "aggViews",  
      params = Seq(  
        Parameter.STRING(name = "ts", value = "$RUN_TS"),  
        Parameter.STRING(name = "country", value = "US")  
      )  
    )  
  )  
)  
)
```

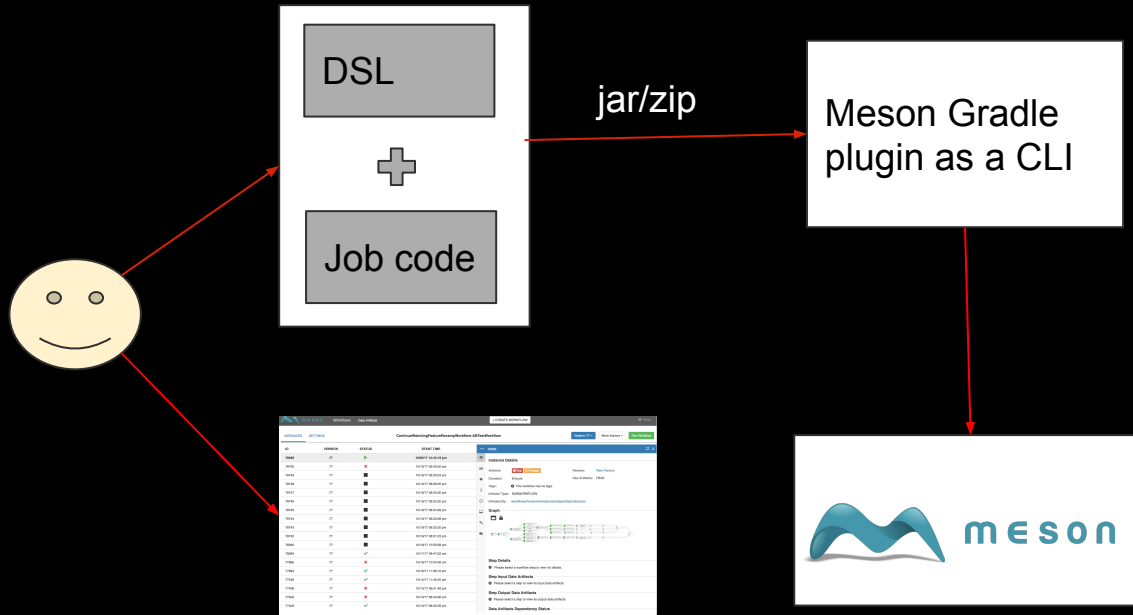
Workflow versions

Workflows have immutable versions

Enables:

- Better collaboration
- Rollbacks
- Reproducibility

Deploying workflows with the Gradle plugin



Link and version
workflow with job
code

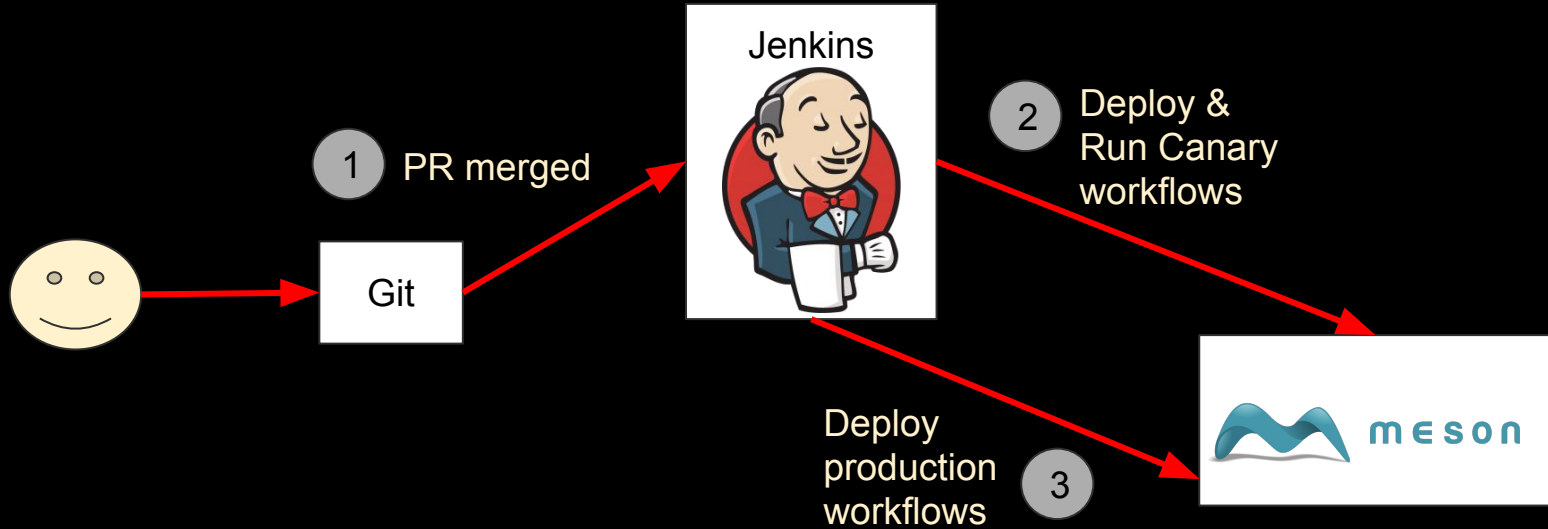
Add metadata (SCM)

Validate

Deploy

Run

Automated releases with canary workflows



Monitor & Debug

The screenshot shows the Meson Workflows interface. At the top, there's a navigation bar with 'meson', 'Workflows', and 'Data Artifacts'. A '+ CREATE WORKFLOW' button is on the right. Below the navigation, the current workflow is 'ContinueWatchingFeatureRevampWorkflow-ABTestWorkflow' with 'Version: 77'. A 'Run Workflow' button is visible.

ID	VERSION	STATUS	START TIME
78929	77	▶	10/20/17 12:43:19 pm
78750	77	✖	10/19/17 06:32:55 pm
78749	77	■	10/19/17 06:30:04 pm
78748	77	■	10/19/17 06:26:25 pm
78747	77	■	10/19/17 06:25:30 pm
78746	77	■	10/19/17 06:25:20 pm
78745	77	■	10/19/17 06:24:59 pm
78744	77	■	10/19/17 06:23:06 pm
78743	77	■	10/19/17 06:22:55 pm
78742	77	■	10/19/17 06:21:23 pm
78340	77	■	10/18/17 10:55:58 am
78094	77	✔	10/17/17 09:47:02 am
77865	77	✖	10/16/17 10:24:39 am
77694	77	✔	10/15/17 11:36:12 am
77548	77	✔	10/14/17 11:45:45 am
77406	77	✖	10/13/17 06:51:46 pm
77404	77	✖	10/13/17 06:44:06 pm
77402	77	✔	10/13/17 06:35:39 pm

The right-hand pane shows 'Instance Details' for instance 78929. It includes 'Actions' (Stop, Retrigger), 'Duration: 6 hours', 'Tags: This workflow has no tags.', 'Initiator Type: SUBWORKFLOW', and 'Initiated By: /workflows/versions/instances/steps/stepinstances/'. Below this is a 'Graph' showing a workflow flow. The 'Step Details' section has a message: 'Please select a workflow step to view its details.' The 'Step Input Data Artifacts' section has a message: 'Please select a step to view its input data artifacts.' The 'Step Output Data Artifacts' section has a message: 'Please select a step to view its output data artifacts.' The 'Data Artifacts Dependency Status' section is currently empty.

This screenshot shows the 'Step Input Data Artifacts' section. It lists two artifacts:

- dataArtifacts/prodhive_dsa_user_interaction_day_agg2b77f86b506c31ea12f234ed6b0409ce33bd546da0135b576b...**
Announced Time: 11/08/17 11:45 pm
Params: partition: 20171107
- dataArtifacts/prodhive_dsa_login_member_allocation_df8dcbca3247b9826d44bdf10bb2c5c60844a243325f0dcbf1...**
Announced Time: 11/08/17 04:23 am
Params: partition: 20171107

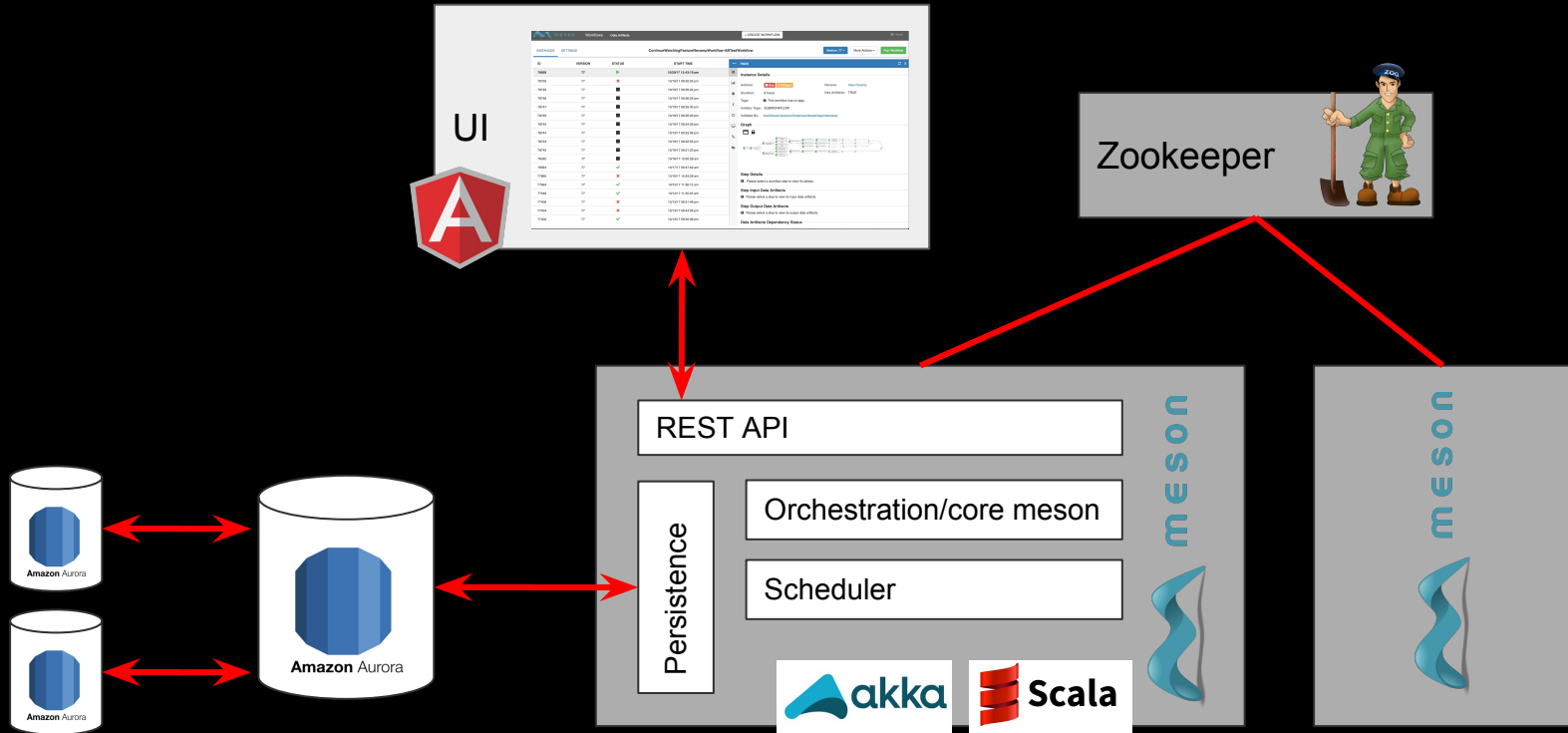
Below this is the 'Step Output Data Artifacts' section, which states: 'No output data artifacts for this step.' The 'Data Artifacts Dependency Status' section states: 'Dependency status is populated only for the steps in WAITING_PRE_CONDITION'. The 'Step Links' section shows a 'MILESTONE: 54043:2 (JSON)' with a visual timeline showing three states: 'Titus-4107039 Queued' (11-09-17, 17:11:24), 'Titus-4107039 Running' (11-09-17, 17:11:54), and 'Titus-4107039 Finished' (11-09-17, 17:31:25). There are also links for 'execution report: 54043:2:0 (HTML)' and 'execution script: 54043:2:0 (HTML)'.

This screenshot shows the 'Step Links' section. It lists several links:

- full_command: 83226:2:0 (HTML)
- spark-submit --ver 2.0 --conf spark.cores.max=64 --conf spark.executor.cores=2 --driver-s...
- mesos_sandbox_link: 83226:2:0 (URL)
- spark_history_server_url: 83226:2:0 (URL)
- spark_mesos_framework: 83226:2:0 (URL)
- spark_ui_url: 83226:2:0 (URL)

The 'Step Messages' section at the bottom states: 'No messages for this step.'

Architecture

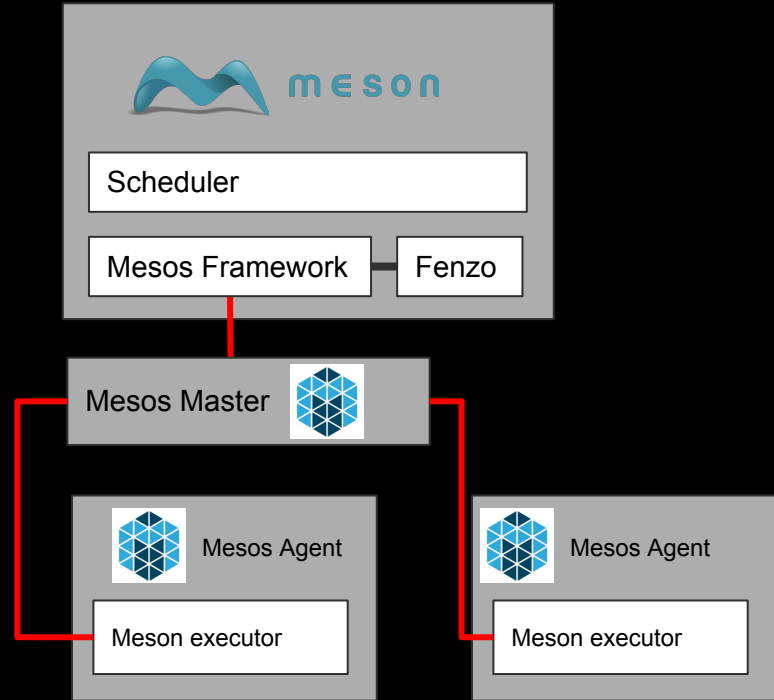


Scheduling

Meson as a Mesos Framework

Mesos offers resources and runs the steps

[Fenzo](#) (Netflix OSS) makes scheduling decisions

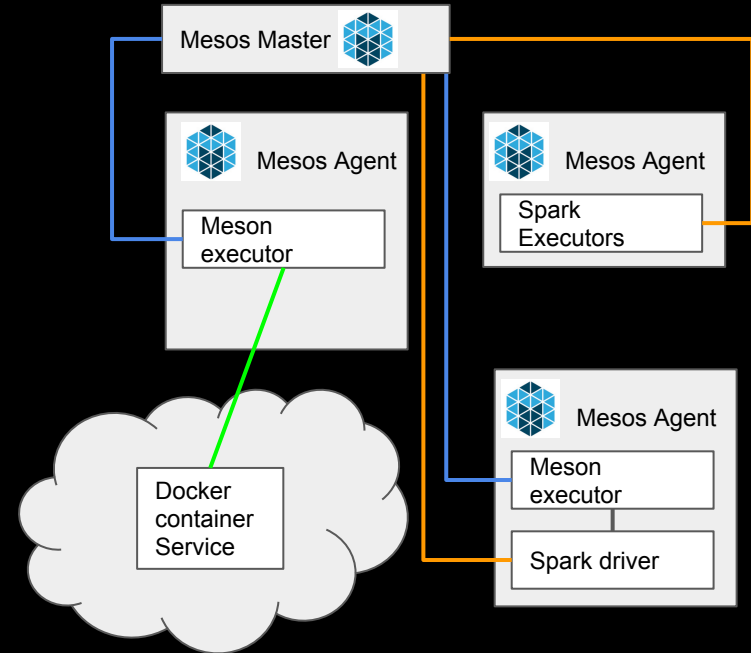


Execution

Custom executor code for different runtime systems (spark, bash...)

Publish runtime debug infos (logs, url to monitoring UIs...)

Meson executors survive Meson scheduler failure



Looking Forward

Closing the loop

Interact with Meson from the running job

Output parameters to leverage loops and foreach

Expose debugging information through Artifacts

Progress Milestones, Links, Counters, Images, etc.

A day in the life of a workflow...

Backfills, work prioritization and parallelism

Avoid re-doing work after fixing a bug and re-deploying a workflow

Explicit (data) lineage

Looking back

Adoption

2+ years in production

10+ managed and self-service deployed clusters

1000+ daily Production and A/B Test ML pipelines

2000+ EC2 instances in Spark/Mesos compute pool

20000+ of steps run per day

One Abstraction doesn't fit all

Evidenced by the many names:

- Workflow
- ProcessFlow
- Pipeline
- DAG
- DataFlow

Over specialization will inevitably weaken other use cases.

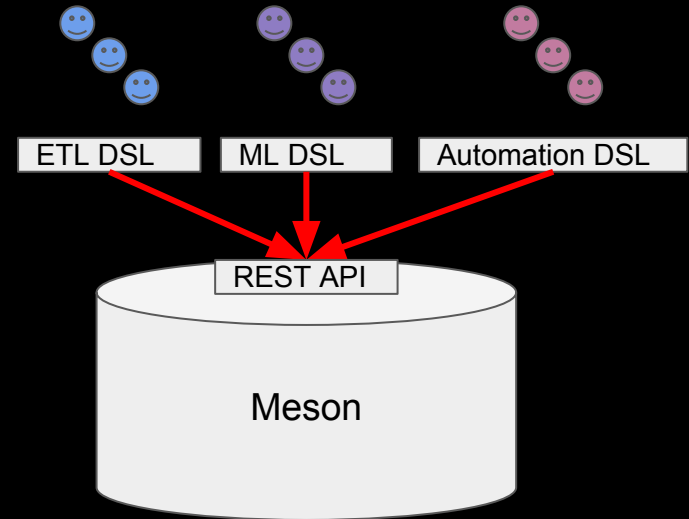


One Abstraction doesn't fit all

With the REST API Meson provides “workflows as a service”.

Enables many domain specialized abstractions:

- A/B test orchestration
- ML orchestration
- ETL pipelines
- Notebook Automation
- And more..



Questions?