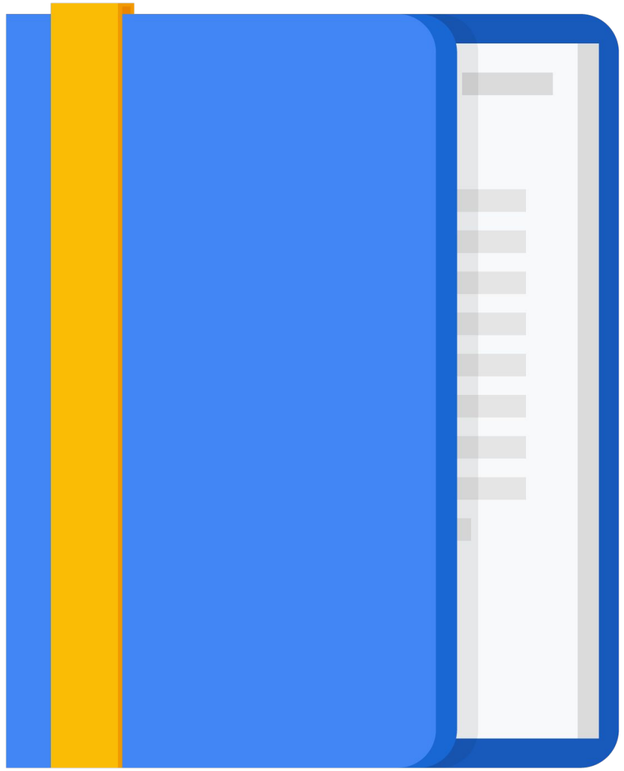
ML Metadata with TFX

Doug Kelly
ML Solutions Engineer, Google Cloud

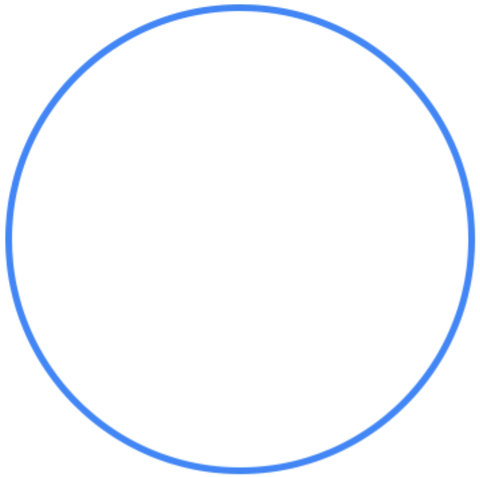
Agenda

TFX pipeline metadata



Review: What is machine learning metadata?

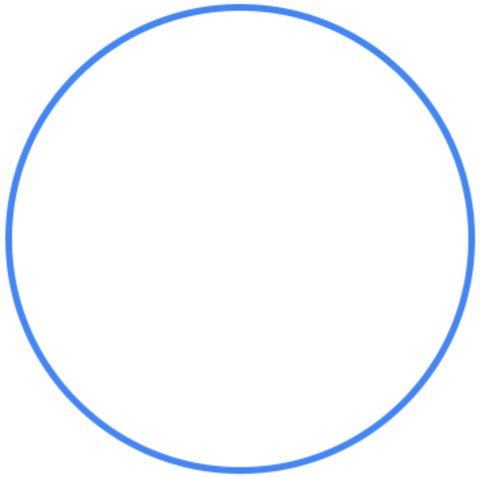
Metadata is “**data about data, but not the data itself.**”



Review: What is machine learning metadata?

Metadata is “**data about data, but not the data itself.**”

Who triggered this pipeline run?

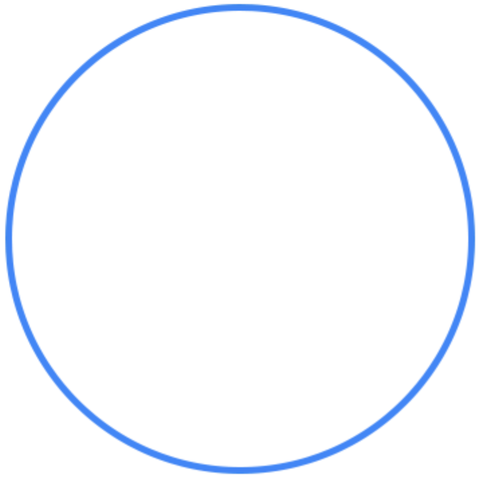


Review: What is machine learning metadata?

Metadata is “**data about data, but not the data itself.**”

Who triggered this pipeline run?

What hyperparameters were used to the train the model?



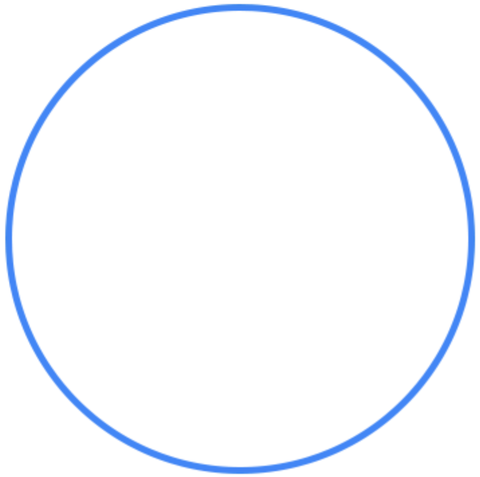
Review: What is machine learning metadata?

Metadata is “**data about data, but not the data itself.**”

Who triggered this pipeline run?

What hyperparameters were used to the train the model?

Where is the model file stored?



Review: What is machine learning metadata?

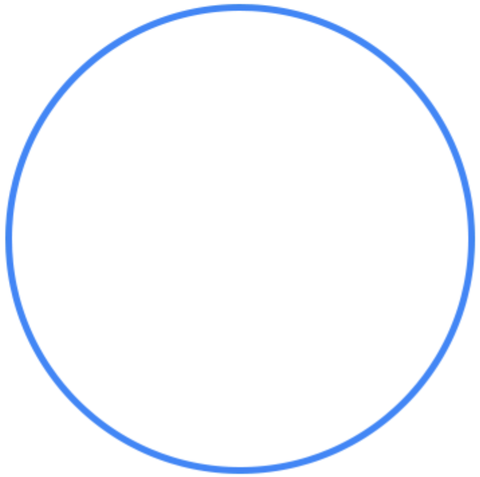
Metadata is “**data about data, but not the data itself.**”

Who triggered this pipeline run?

What hyperparameters were used to the train the model?

Where is the model file stored?

When was the model pushed to production?



Review: What is machine learning metadata?

Metadata is “**data about data, but not the data itself.**”

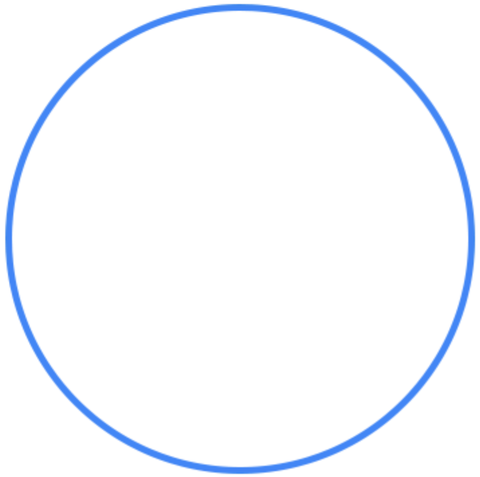
Who triggered this pipeline run?

What hyperparameters were used to the train the model?

Where is the model file stored?

When was the model pushed to production?

Why was model A preferred over model B?



Review: What is machine learning metadata?

Metadata is “**data about data, but not the data itself.**”

Who triggered this pipeline run?

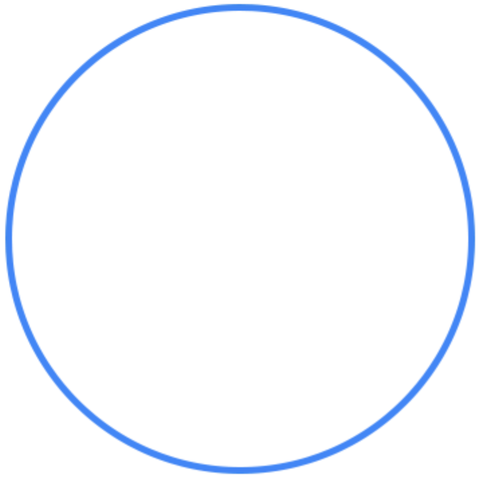
What hyperparameters were used to the train the model?

Where is the model file stored?

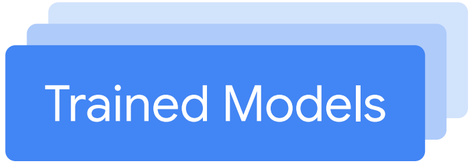
When was the model pushed to production?

Why was model A preferred over model B?

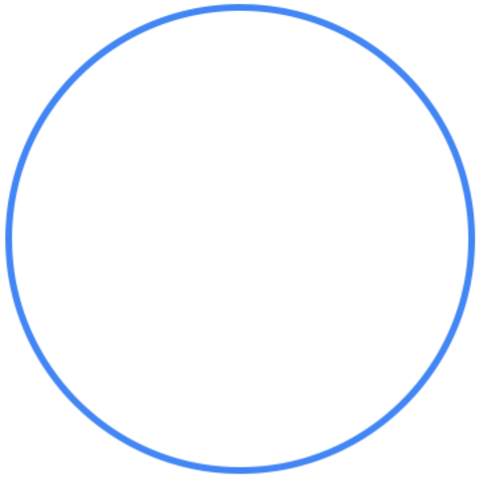
How was the training environment configured?



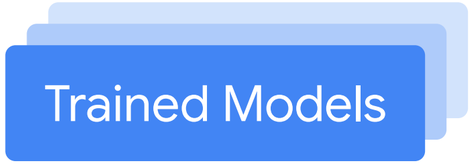
What is in TFX's Metadata Store?



Type definitions of artifacts and their properties



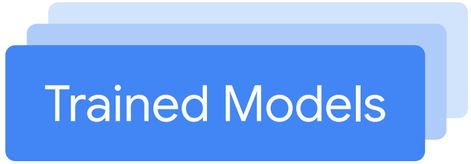
What is in TFX's Metadata Store?



Type definitions of artifacts and their properties

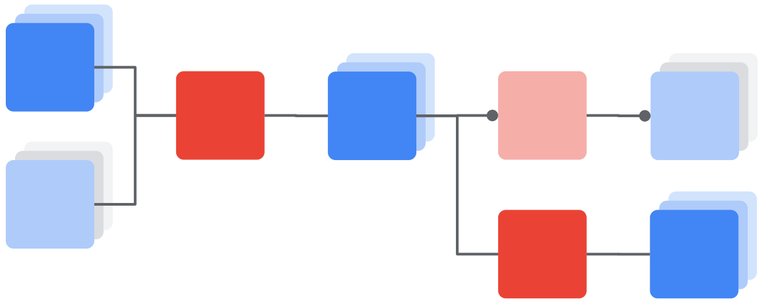
Execution records (runs) of components

What is in TFX's Metadata Store?



Type definitions of artifacts and their properties

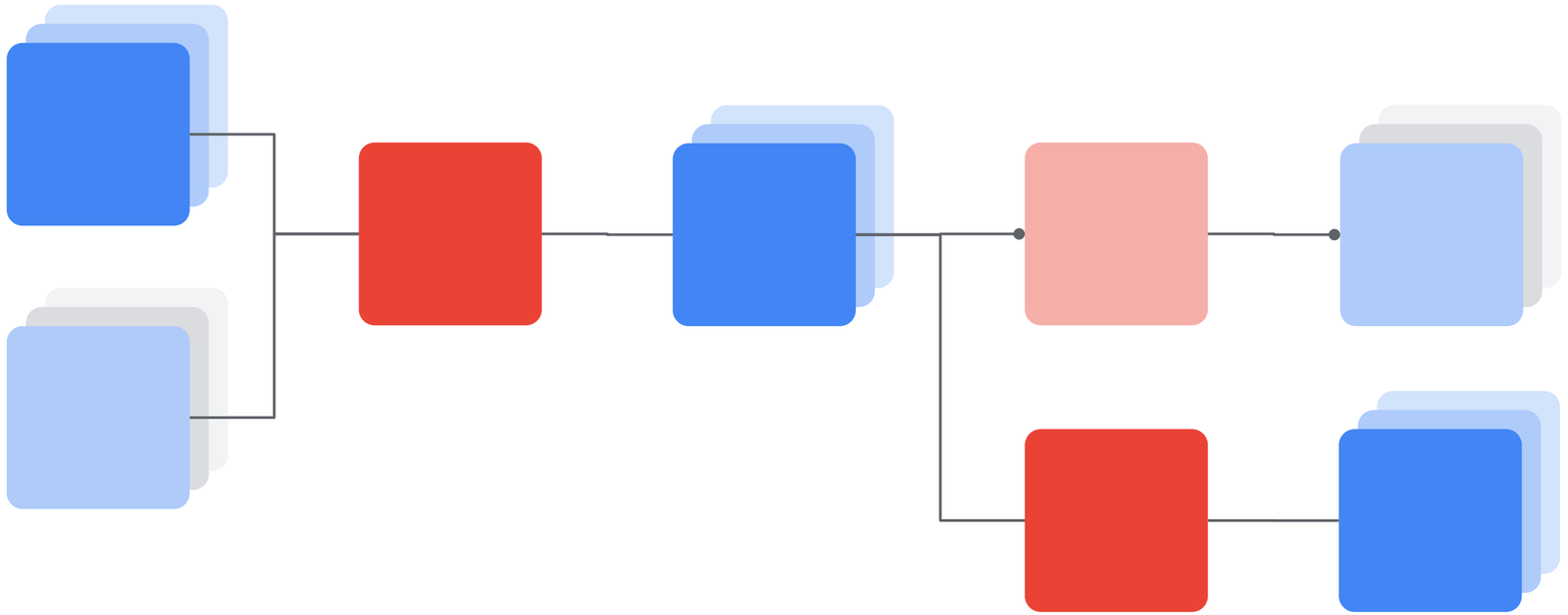
Execution records (runs) of components



Data provenance across all executions

TFX Metadata-powered functionality

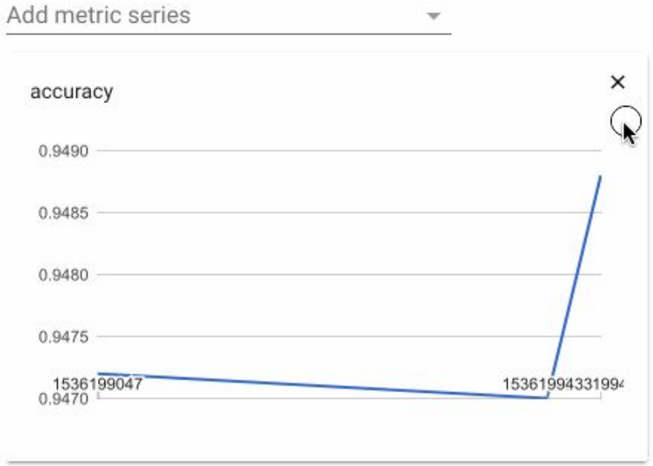
Trace model runs back to the data it was trained on



TFX Metadata-powered functionality

Compare previous model runs

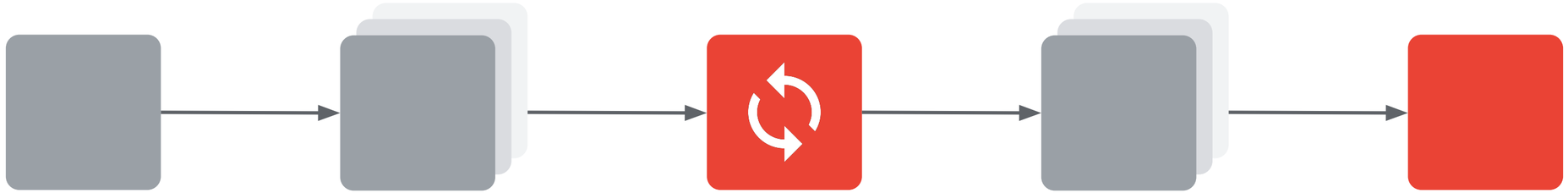
```
eval_results = tfma.make_eval_results([tfma_result_1, tfma_result_2, tfma_result_3],
                                      tfma.constants.MODEL_CENTRIC_MODE)
tfma.view.render_time_series(eval_results, OVERALL_SLICE_SPEC)
```

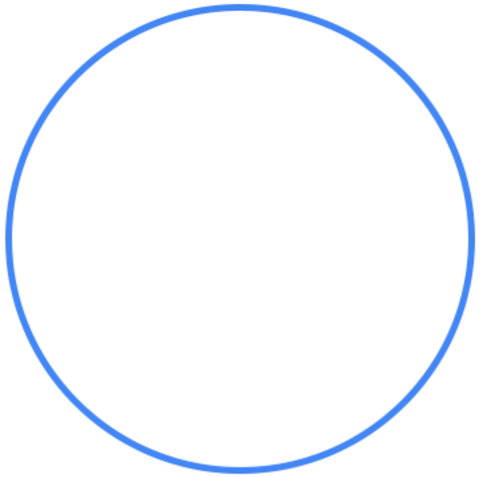


Model	Data	accuracy	accuracy_baseline	auc	auc_precision_recall	average_loss	label/mean
1536199479	data.csv	0.94880	0.94220	0.93168	0.98516	0.13980	0.94220
1536199433	data.csv	0.94700	0.94220	0.93165	0.98170	0.13979	0.94220
1536199047	data.csv	0.94720	0.94220	0.92914	0.99480	0.14103	0.94220

TFX Metadata-powered functionality

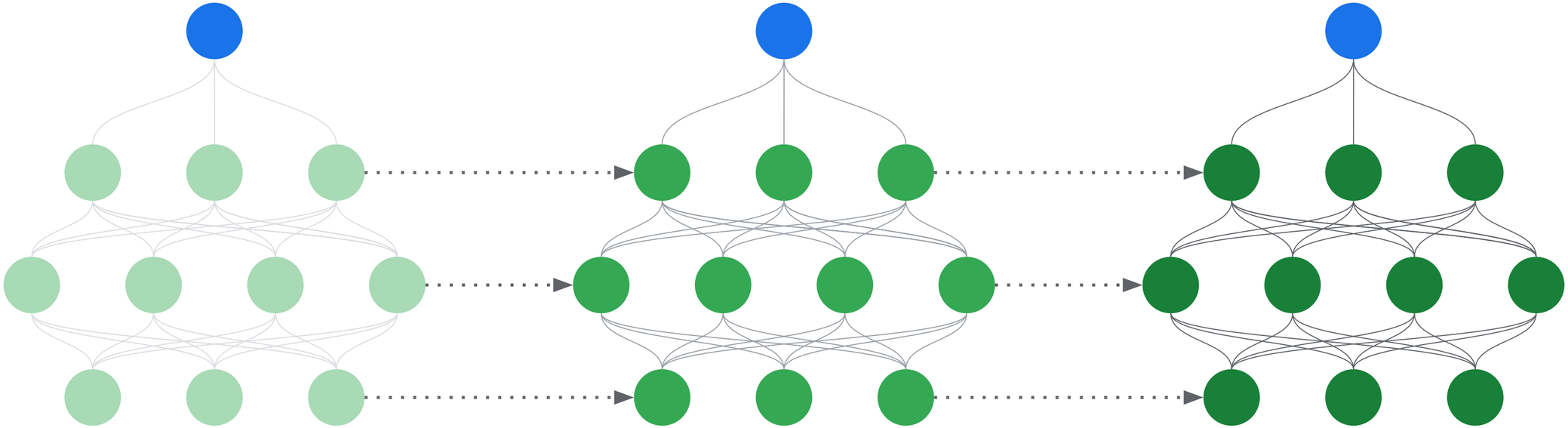
Re-use previously computed outputs





TFX Metadata-powered functionality

Re-use previously computed outputs



Advanced TFX use case: pipeline warm starting

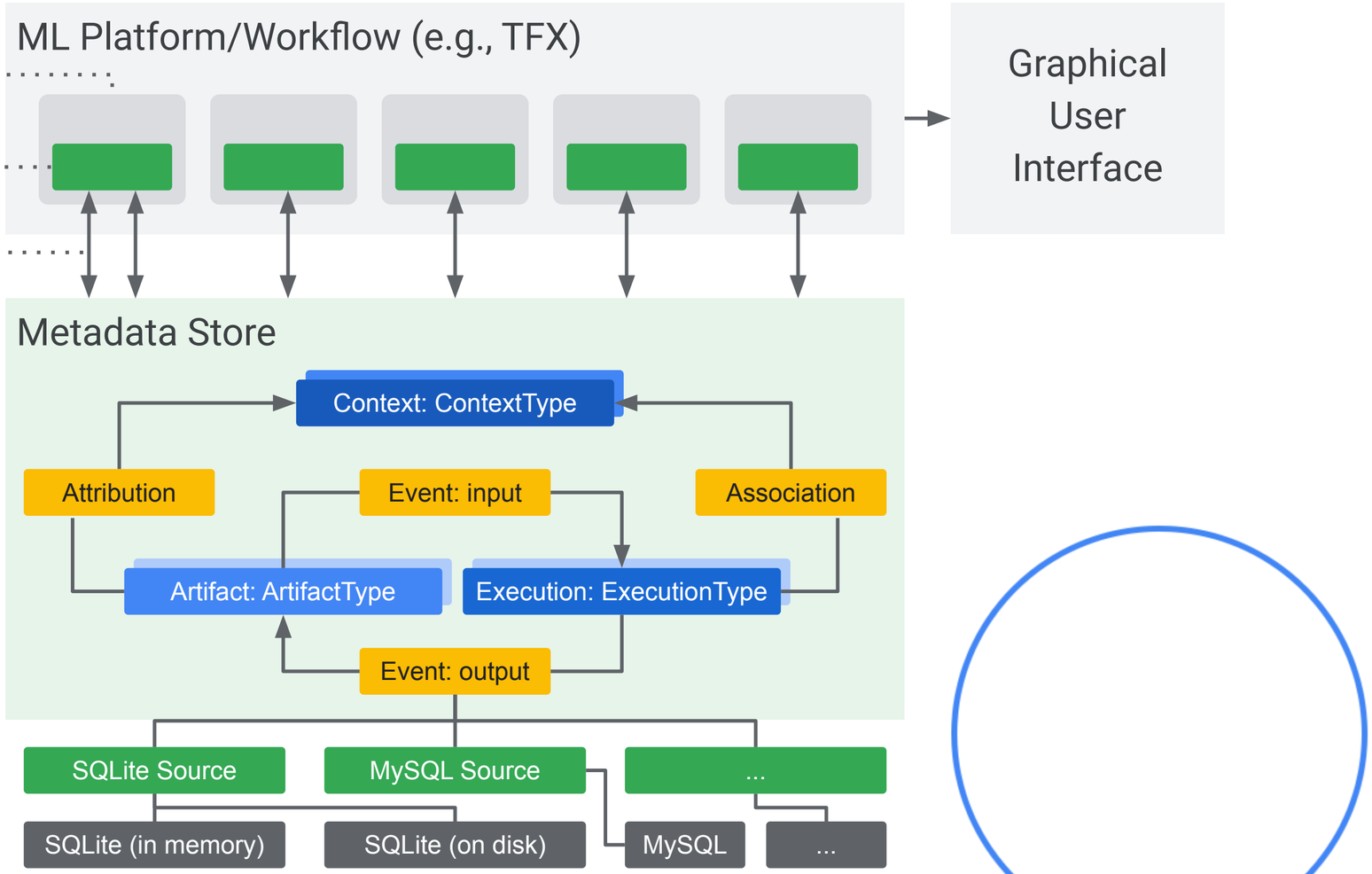
```
# Get the latest model so that we can warm start from the model.
 latest_model_resolver = ResolverNode(
     instance_name='latest_model_resolver',
     resolver_class=latest_artifacts_resolver.LatestArtifactsResolver,
     latest_model=Channel(type=Model))

 # Uses user-provided Python function that implements a model using TF-Learn.
 trainer = Trainer(
     module_file=module_file,
     transformed_examples=transform.outputs['transformed_examples'],
     schema=schema_gen.outputs['schema'],
     base_model=latest_model_resolver.outputs['latest_model'],
     transform_graph=transform.outputs['transform_graph'],
     train_args=trainer_pb2.TrainArgs(num_steps=10000),
     eval_args=trainer_pb2.EvalArgs(num_steps=5000))
```

ML Metadata (MLMD) library data model

Pipeline components/steps

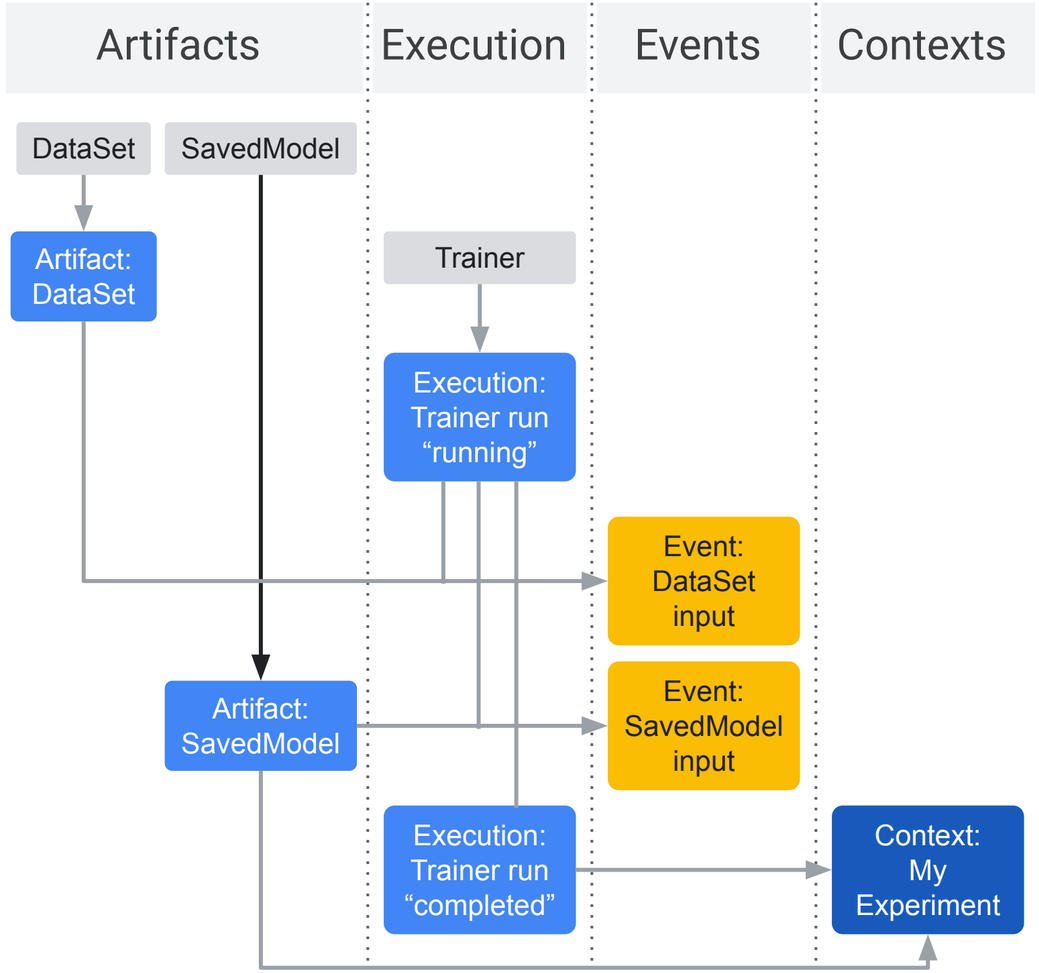
MLMD client libraries

Metadata and lineage about input/output artifacts



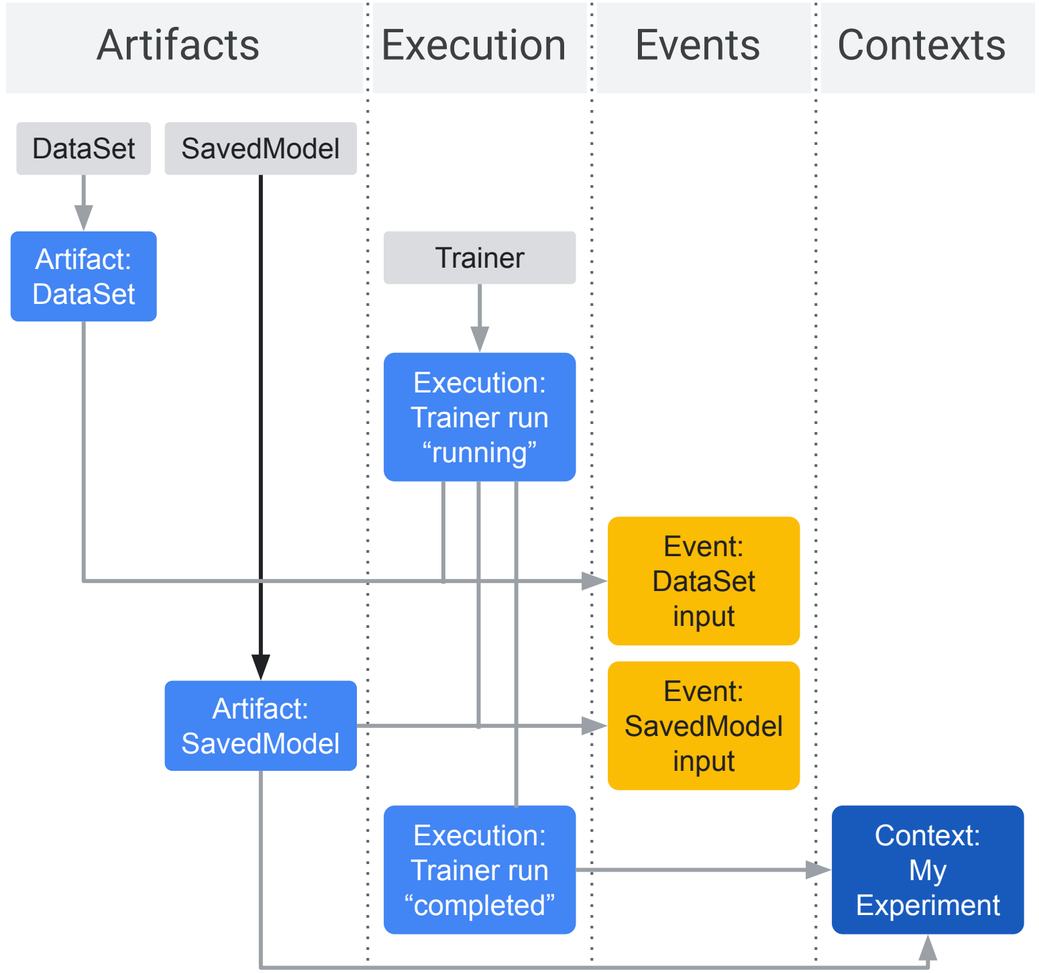
ML Metadata in action

1. Register ArtifactTypes.



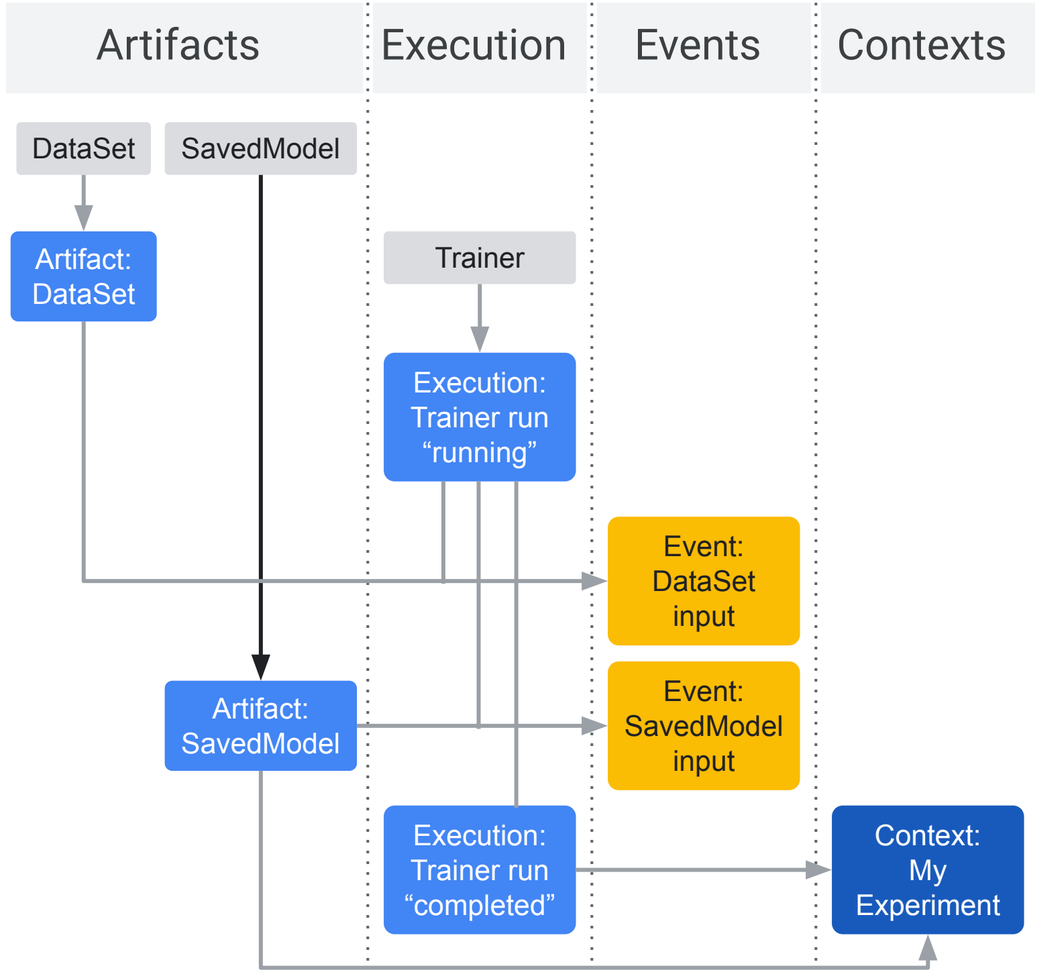
ML Metadata in action

1. Register ArtifactTypes.
2. Register ExecutionTypes.



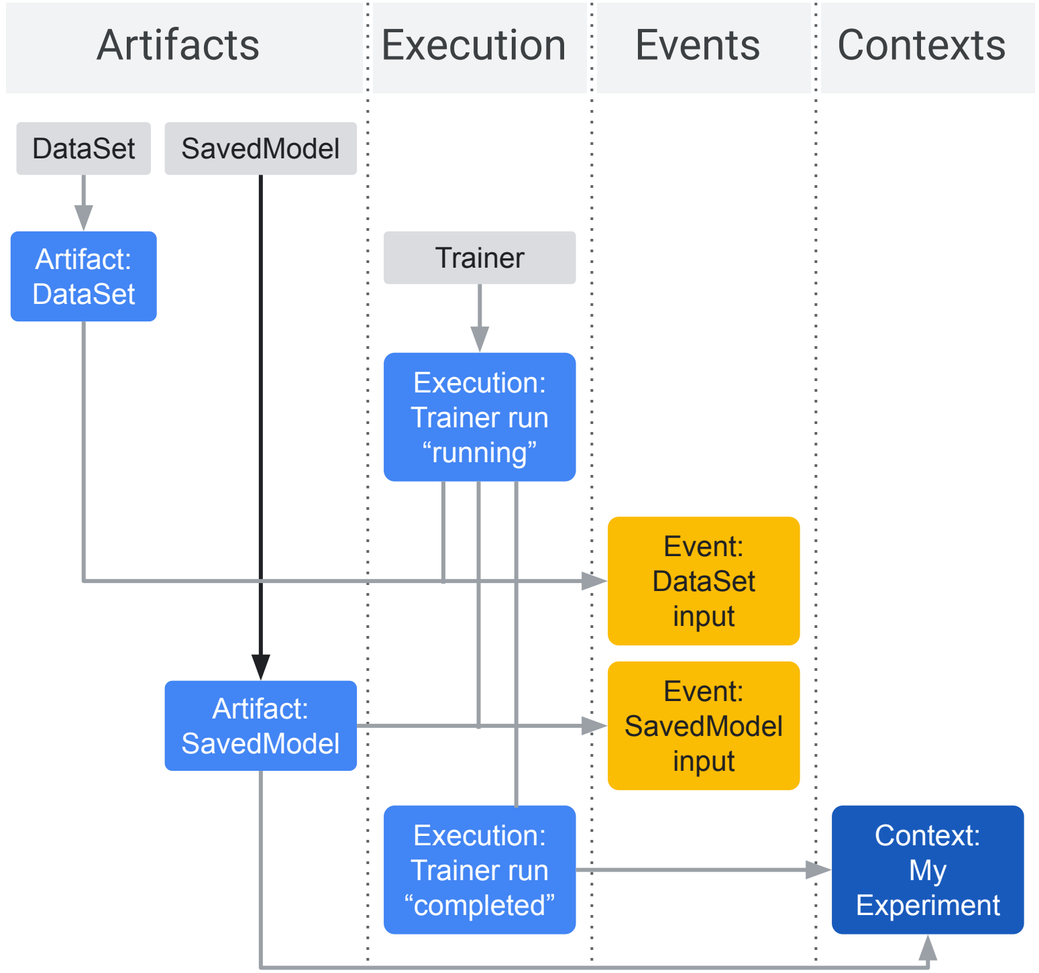
ML Metadata in action

1. Register ArtifactTypes.
2. Register ExecutionTypes.
3. Create DataSet Artifact.



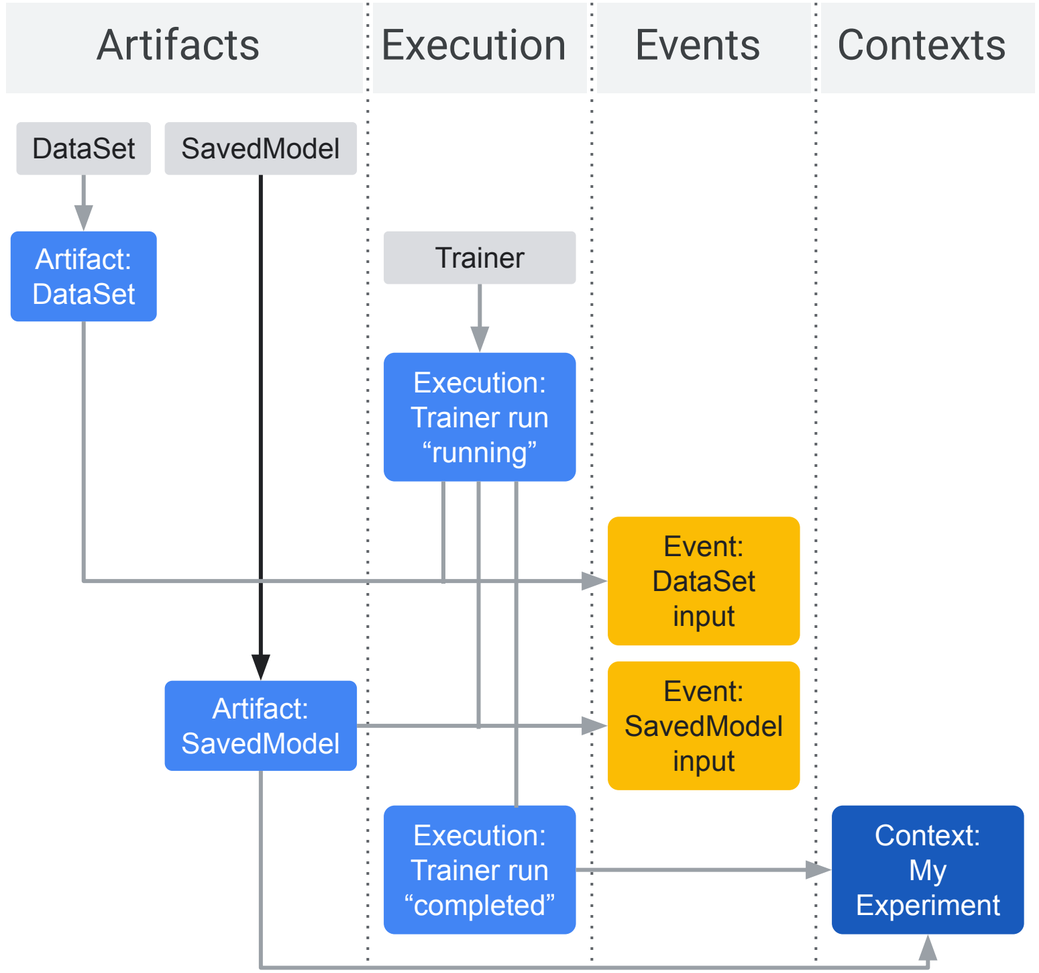
ML Metadata in action

1. Register ArtifactTypes.
2. Register ExecutionTypes.
3. Create DataSet Artifact.
4. Create Execution for Trainer.



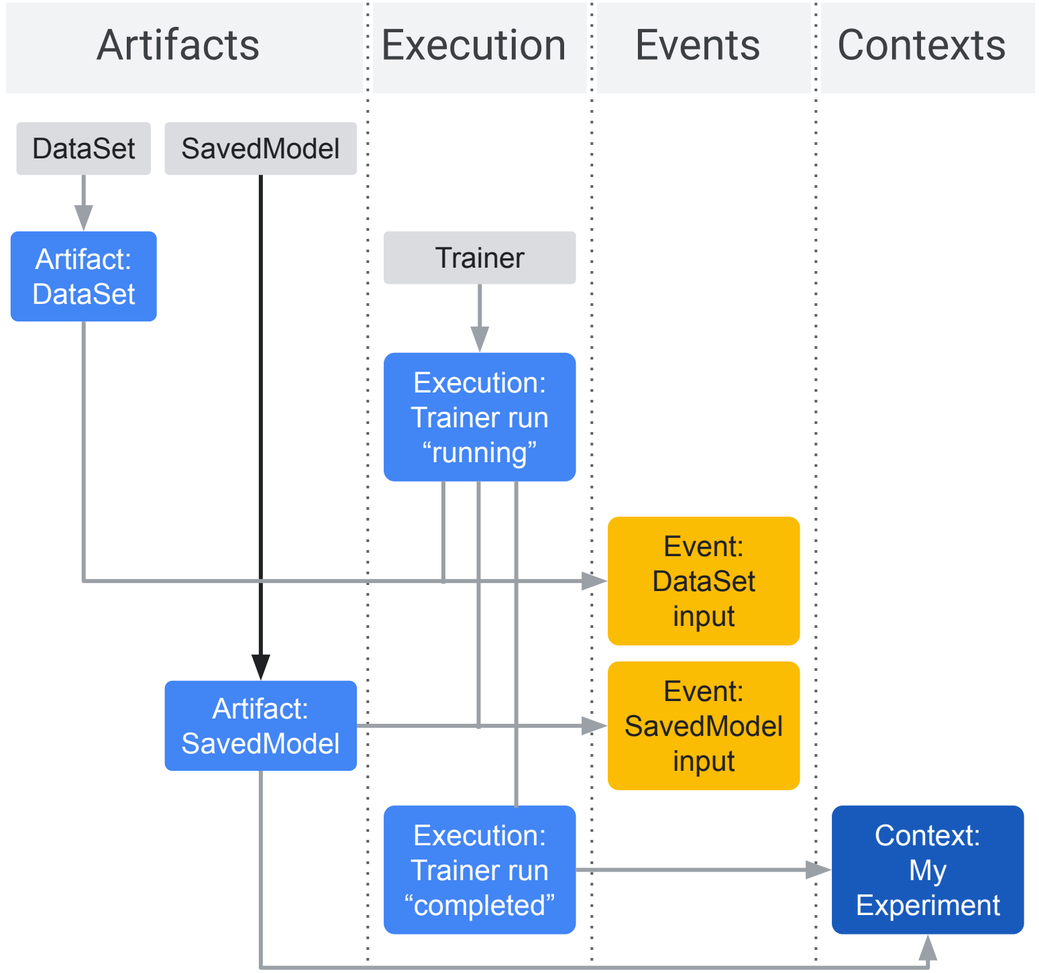
ML Metadata in action

1. Register ArtifactTypes.
2. Register ExecutionTypes.
3. Create DataSet Artifact.
4. Create Execution for Trainer.
5. Read DataSet and record input event.



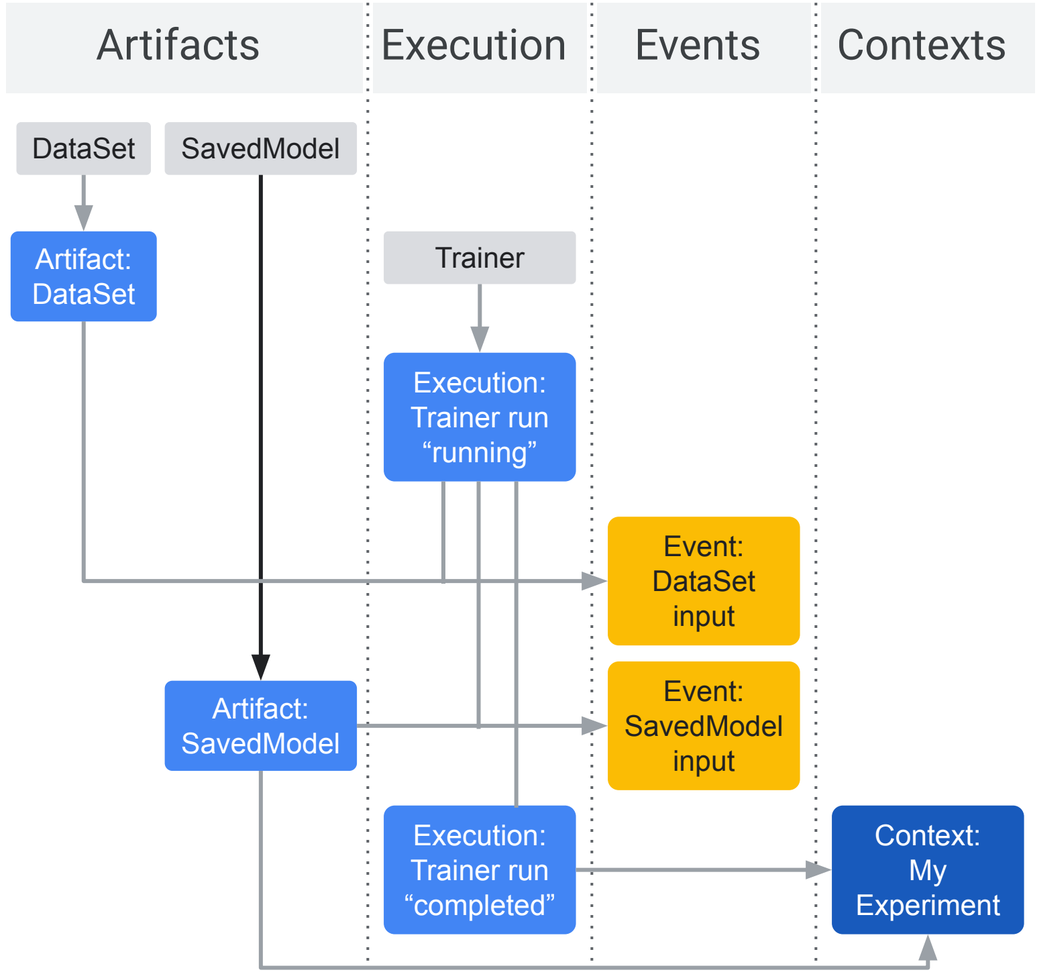
ML Metadata in action

1. Register ArtifactTypes.
2. Register ExecutionTypes.
3. Create DataSet Artifact.
4. Create Execution for Trainer.
5. Read DataSet and record input event.
6. Train Model and Create SavedModel Artifact.



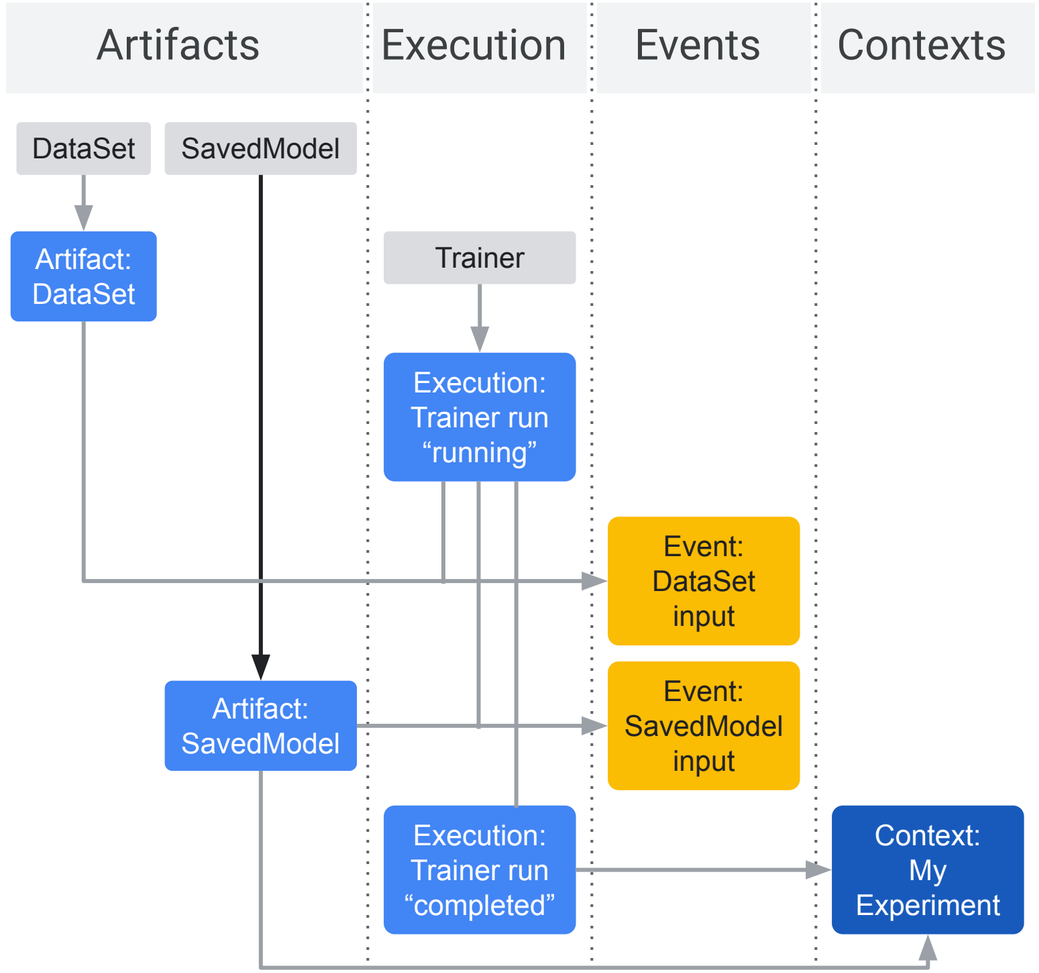
ML Metadata in action

1. Register ArtifactTypes.
2. Register ExecutionTypes.
3. Create DataSet Artifact.
4. Create Execution for Trainer.
5. Read DataSet and record input event.
6. Train Model and Create SavedModel Artifact.
7. Write SavedModel and record output event.



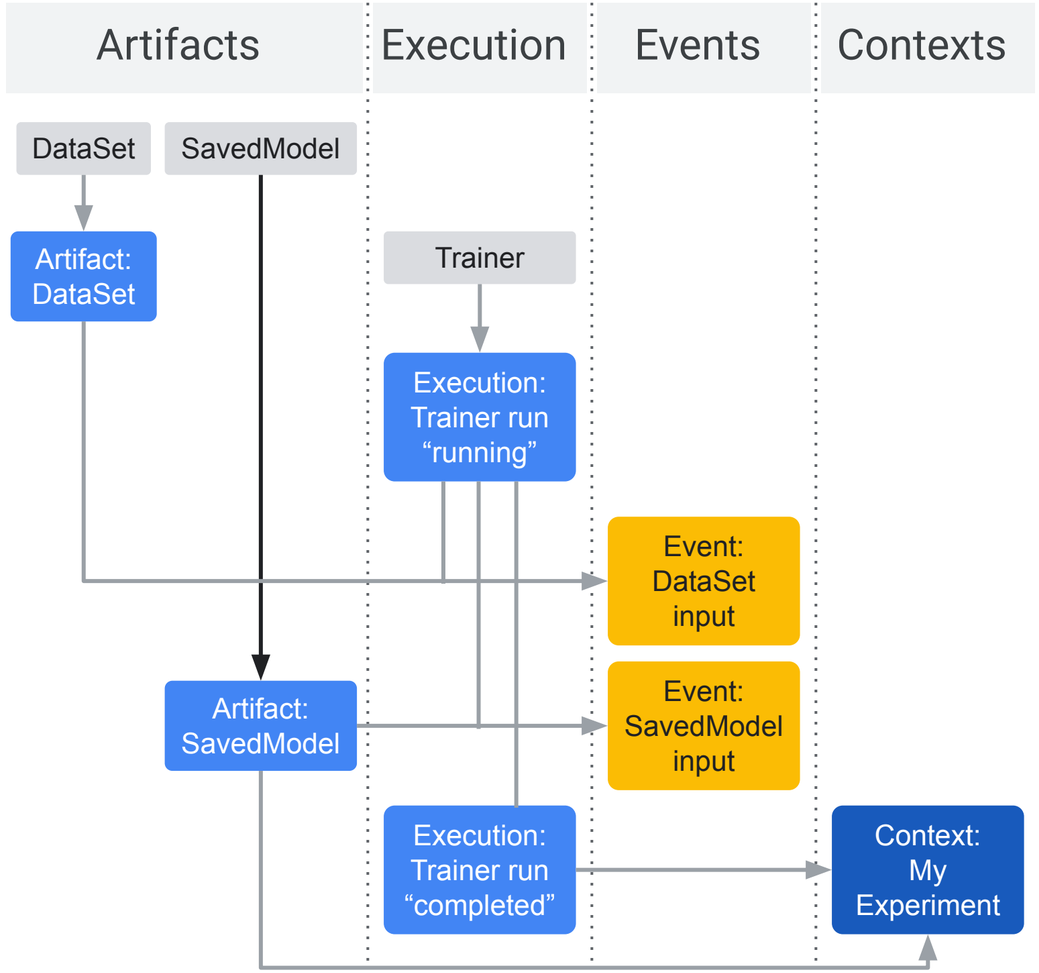
ML Metadata in action

1. Register ArtifactTypes.
2. Register ExecutionTypes.
3. Create DataSet Artifact.
4. Create Execution for Trainer.
5. Read DataSet and record input event.
6. Train Model and Create SavedModel Artifact.
7. Write SavedModel and record output event.
8. Mark Execution completed.

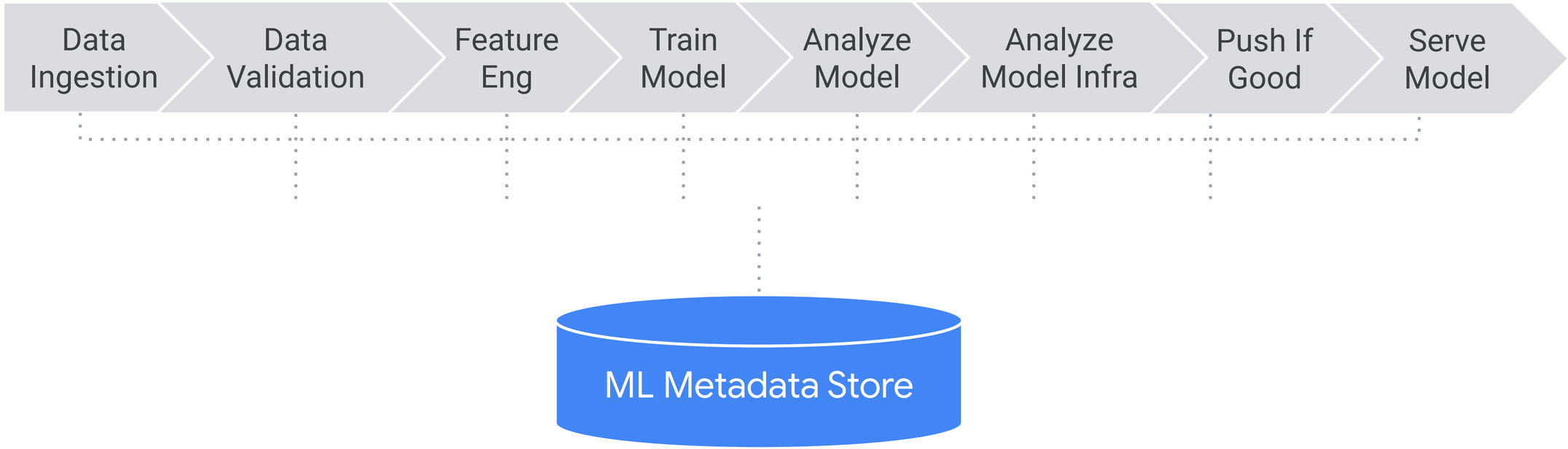


ML Metadata in action

1. Register ArtifactTypes.
2. Register ExecutionTypes.
3. Create DataSet Artifact.
4. Create Execution for Trainer.
5. Read DataSet and record input event.
6. Train Model and Create SavedModel Artifact.
7. Write SavedModel and record output event.
8. Mark Execution completed.
9. Annotate the experiment with a Context.



Metadata enables TFX task and data-aware pipelines



Lab

TFX Pipeline Metadata

Google Cloud