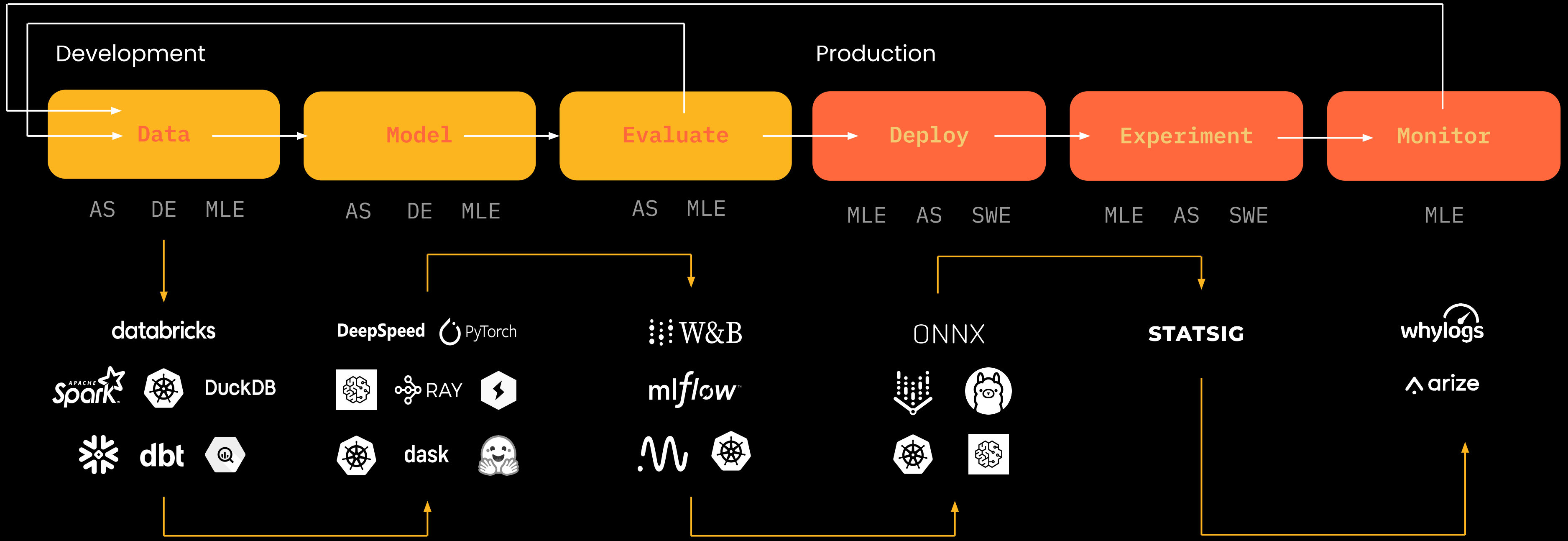
union

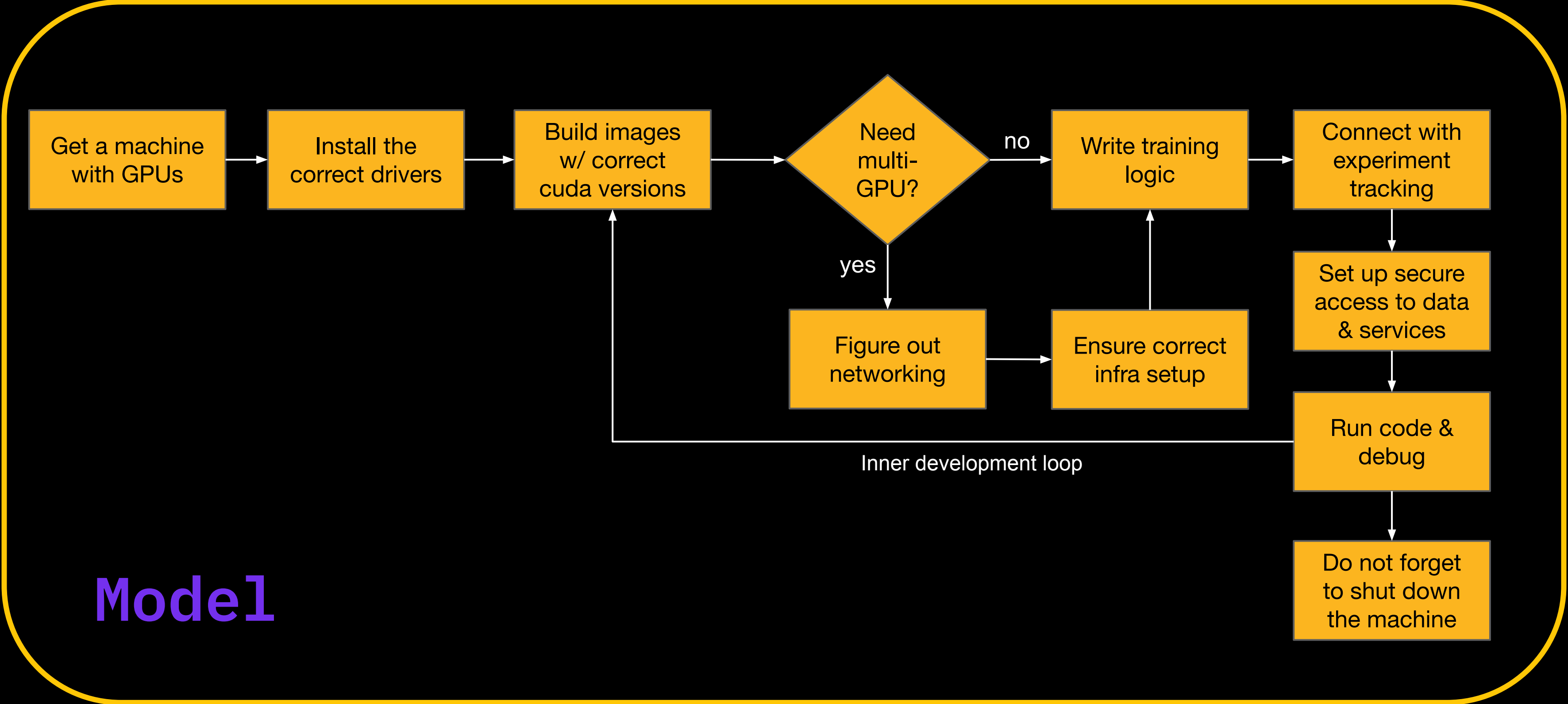
Orchestrate Your AI

In the Era of AI every company must become an AI company

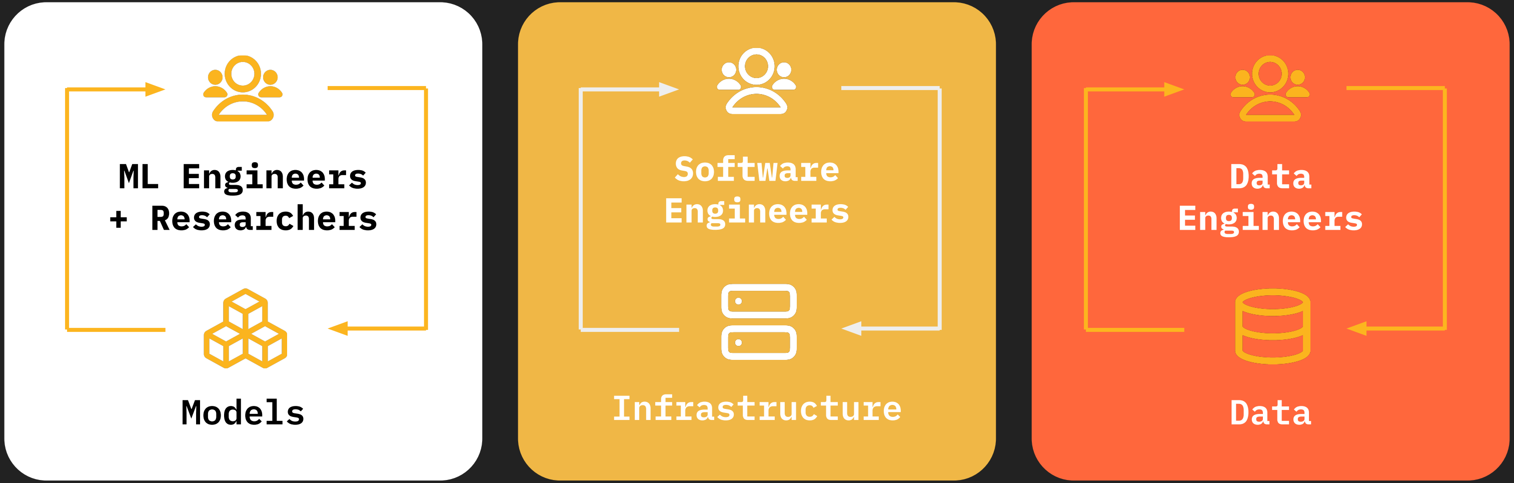
But, AI product development is **chaotic**



And, solving this problem is hard


Get a machine with GPUs
Install the correct drivers
Build images w/ correct cuda versions
Need multi-GPU?
no
Write training logic
Connect with experiment tracking
yes
Figure out networking
Ensure correct infra setup
Set up secure access to data & services
Run code & debug
Inner development loop
Do not forget to shut down the machine
Model

Because teams are often siloed.



ML Engineers + Researchers

Models

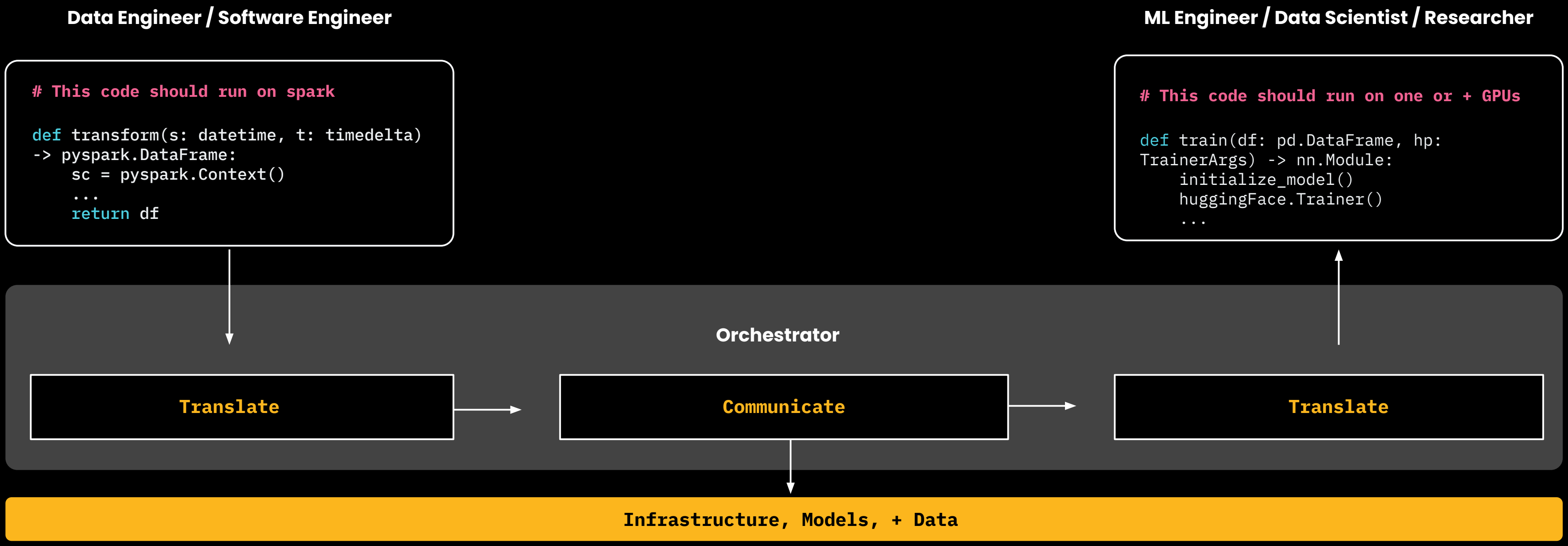
Software Engineers

Infrastructure

Data Engineers

Data

Companies need a unified platform to create AI products



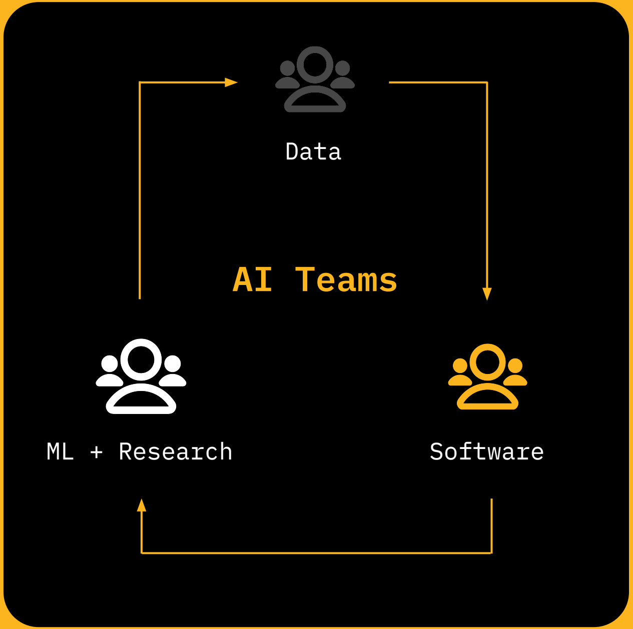
Data Engineer / Software Engineer

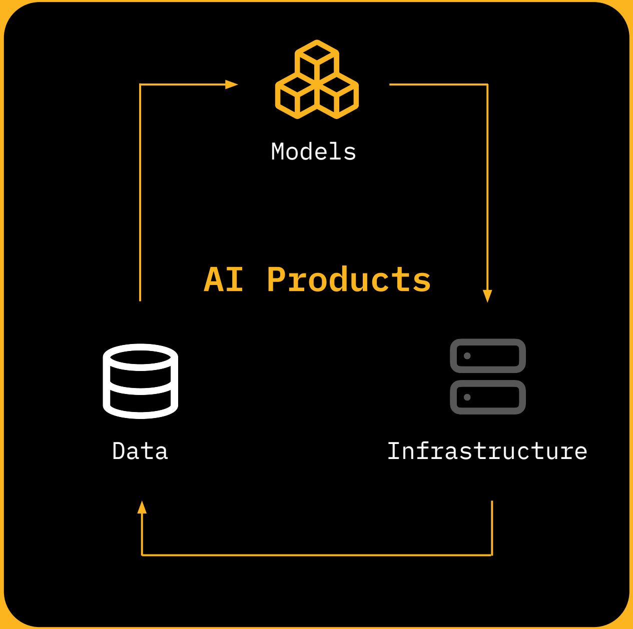
```
# This code should run on spark

def transform(s: datetime, t: timedelta)
-> pyspark.DataFrame:
    sc = pyspark.Context()
    ...
    return df
```

ML Engineer / Data Scientist / Researcher

```
# This code should run on one or + GPUs

def train(df: pd.DataFrame, hp:
TrainerArgs) -> nn.Module:
    initialize_model()
    huggingFace.Trainer()
    ...
```

Union brings engineering, ops, + data science together… into **one efficient AI team.**

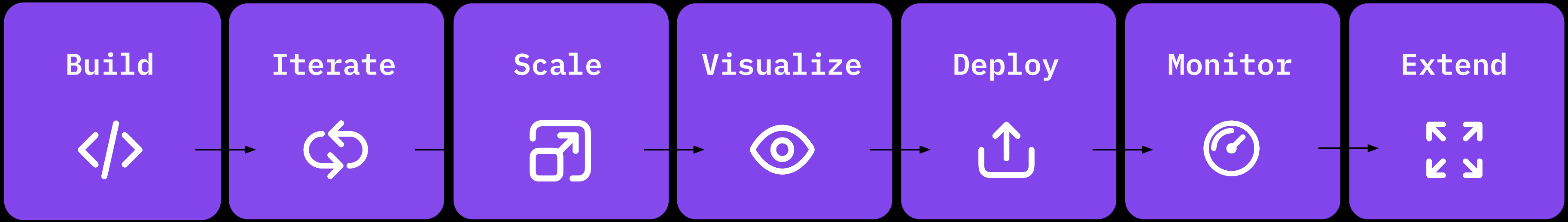




Open source + closed source, ensures flexibility without lock-in

Flyte open source to unlock AI

One platform for your ML pipelines



Build

Write code in Python and test locally

Iterate

Rapidly ship to remote with a tight inner loop

Scale

Scale out and leverage multiple frameworks

Visualize

Track experiments, view plots, and use results

Deploy

CRON jobs and API-based executions

Monitor

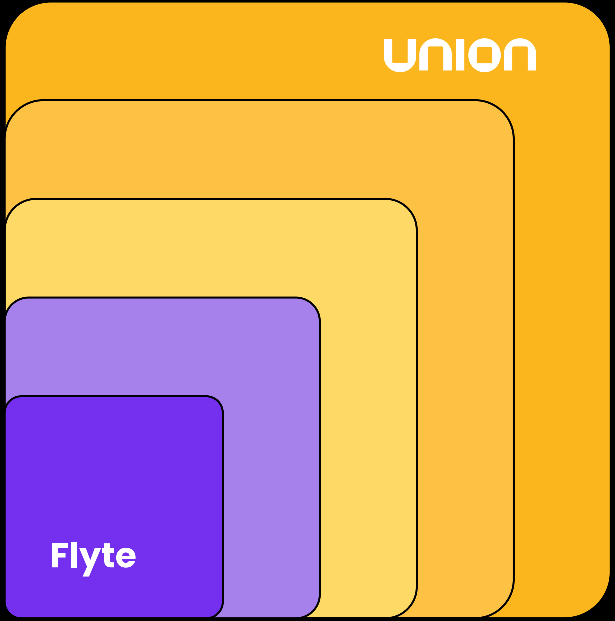
Notifications via slack, email + pagerduty

Extend

Integrate with 2p and 3p external services

Flyte

Union supercharges & extends Flyte



Fully managed & secure
Leverage a robust platform that meets rigorous standards for security, compliance, and operational reliability - in your cloud.

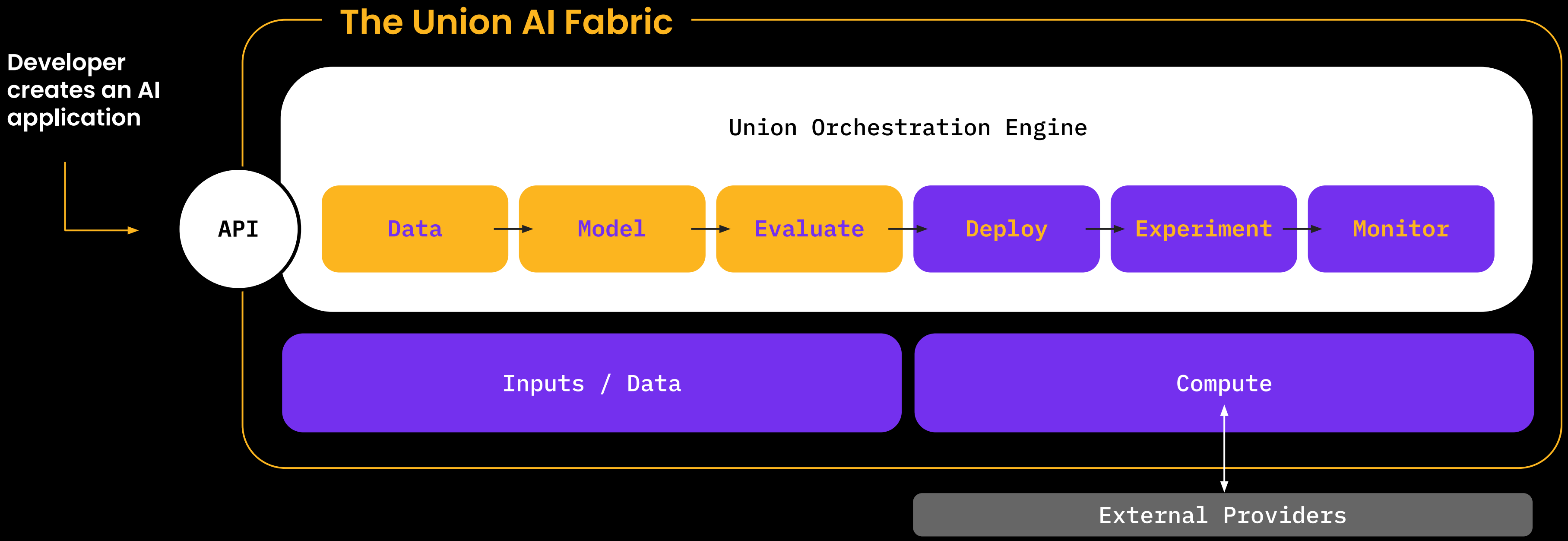
Supercharged performance
Run complex AI workloads with unparalleled performance, scale, and efficiency. (25x faster)

Enhanced developer experience
Shorten the development loop from hours to seconds while writing production-ready code.

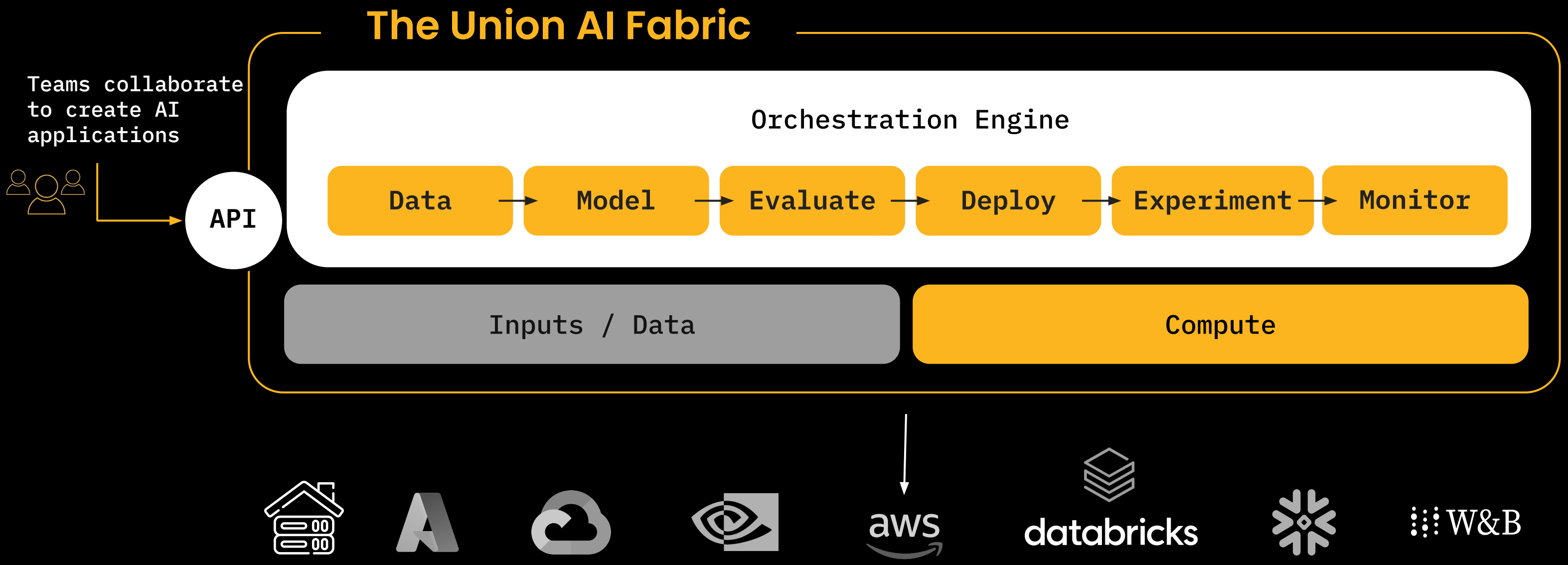
More efficient
Boost ROI by enabling teams to access the resources they need while sharing underlying infrastructure.

SLAs + customer success from the team that built Flyte

Our mission is to make AI products **reliable, secure, + easy**



Our mission is to make AI products **reliable, secure, + easy**



Workflows on Union are...

- Reproducible
- Programmable
- Composable
- Interoperable
- Scalable
- Production Ready
- Reliable
- Efficient

Workflows on Union are...

Reproducible

Automatic Versioning
Containerization
Data Immutability
Durable State & Results

Programmable

Declarative Infra
Declarative data flow
Type-Aware
Pythonic

Composable

Reusable components
Reusable data
Heterogeneous workloads
Accurate caching

Interoperable

Integrations
API-driven development
Framework agnostic
Multi-language

Scalable

Local-Remote parity
Multi-tenant
Integrated compute
Multi-cluster

Prod-ready

Scheduling
Notifications
Observability
Dev/prod isolation

Reliable

Retries
Checkpointing
Failure Recovery
Multi-AZ

Efficient

Ephemeral Compute
Spot Instances
Checkpointing
Fractional GPUs

Demo

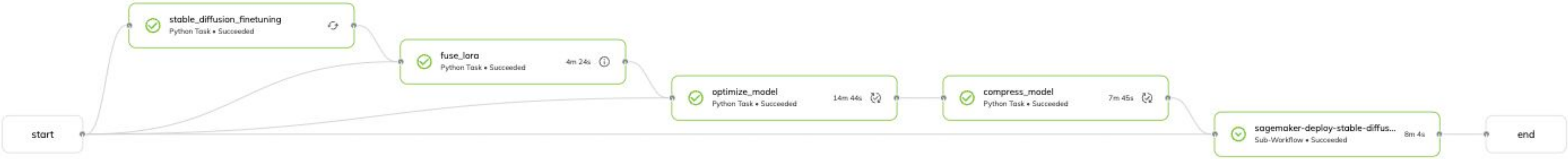
Demo: Stable Diffusion Fine Tuning

Training
Train LORA adapters on Multiple GPUs

Optimize the model for deployment
Use ONNX + Tensorrt to optimize the model for Nvidia hardware

Use built-in Sagemaker deployment integration
Automatically deploy the model

- □ Simple development and hardware targeting
- □ Caching (save money and time)
- □ Automated container builds
- □ UI based triggering
- □ Model Registry and Model cards
- □ Extensible integration system - Sagemaker
- □ Sharable building blocks - reuse optimization code!



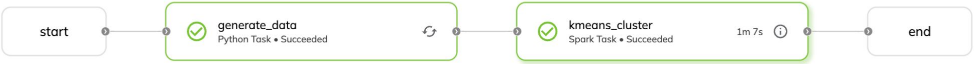
Demo: Spark three ways (local, open-source on k8s, and Databricks)

Use Spark
Use spark local, or on K8s or on Databricks (and others)

Simplified dependency management
Automatically build containers and manage clusters

Visualize
Visualize data using FlyteDecks and debug Spark using Spark History Server

- □ Test spark code locally
- □ Declarative infrastructure
- □ Automated container builds
- □ Extensible - Databricks vs others
- □ Visualize data

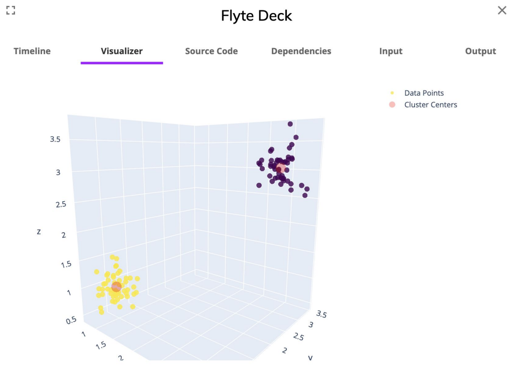


start

generate_data
Python Task • Succeeded

kmeans_cluster
Spark Task • Succeeded
1m 7s

end



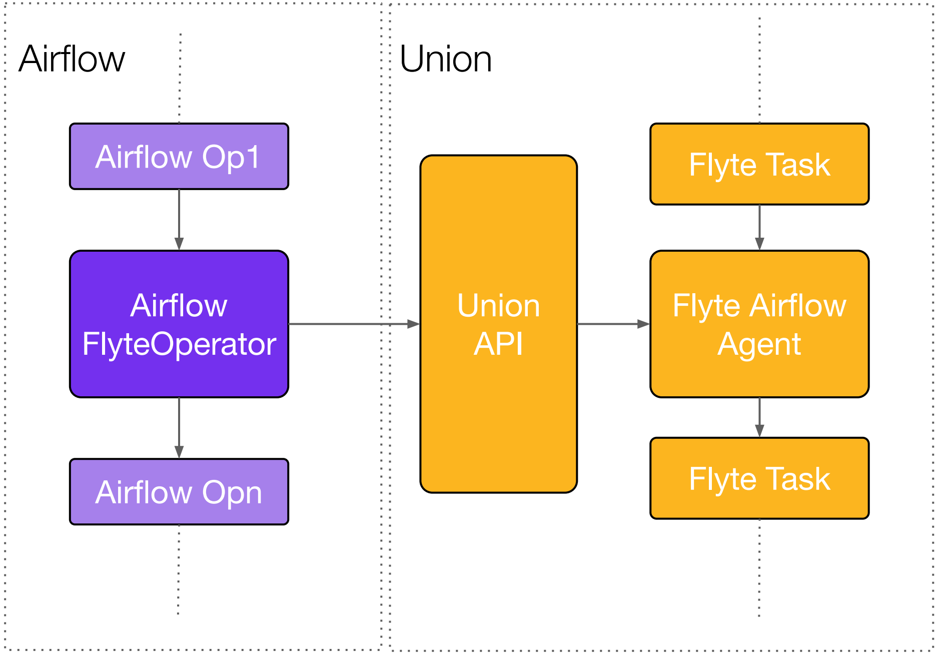
Flyte Deck

Demo: Integrate with Airflow

Trigger from Airflow
Seamless integration of Union Workflows / Tasks using FlyteAirflowProvider

Migrate Airflow pipelines to Union
Use most existing Airflow Operators and let Union handle the scaling

Interoperate
Use Flyte features like caching, data movement, type-safety and interoperate with Airflow operators



"We were able to ***save nine months of engineering time*** *by avoiding any code changes, and simply* ***lifting and shifting*** *our Airflow code and running it with Union."* — Shih-Gian Lee, Senior Machine Learning Engineer, Porch.

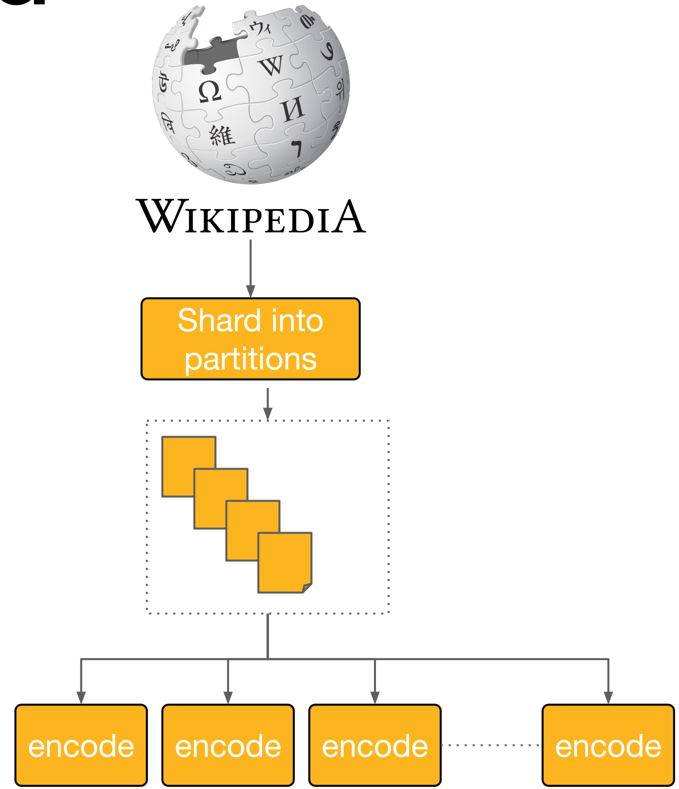
Demo: Embed Wikipedia

Scale
Scale to multiple GPU's and cpu's efficiently

Simple
Code is simple, no need to learn complicated frameworks

Efficient
Native caching and high performance

Model and Reuse
Model the workflow as small tasks and reuse them



Demo: Embed PDFs fast

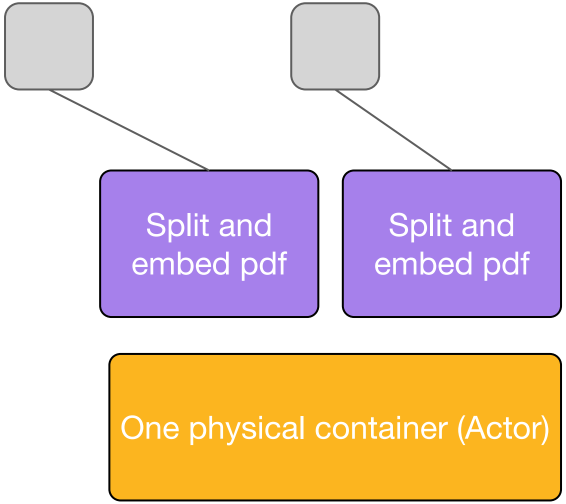
Scale
Scale to multiple GPU’s and cpu’s efficiently

Simple
Code is simple, no need to learn complicated frameworks

Efficient
Native caching and high performance

Model and Reuse
Model the workflow as small tasks and reuse them

Capture artifacts
Capture the lineage and artifacts



Customer Innovation

Training & Fine Tuning

Confidently run large-scale training or fine-tuning on GPU clusters across clouds and on-premise

Data Processing

Seamlessly connect to your data stack. Focus on data, not infrastructure.

Near-Line Inference

Deliver high-throughput, reliable, and fault-tolerant inference for production AI apps

Generative AI & LLMs

Take generative AI applications to production faster

Bioinformatics & Pharma

Effortlessly carry out scientific computing workflows with must-have features out of the box

LinkedIn Spotify TOYOTA freenome WAYVE NVIDIA

stripe ANDURIL gojek LOCKHEED MARTIN Expedia TESLA amazon hopper

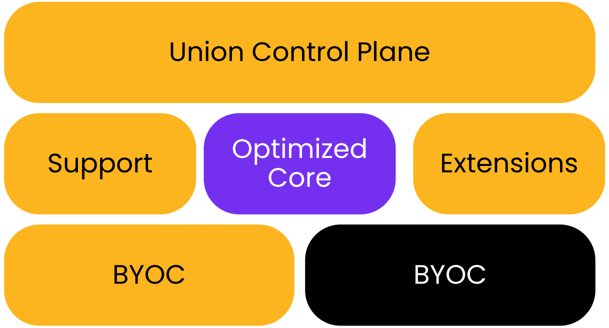
The standard AI orchestration platform

Spotify: Luigi (2013) -> **Migrated to Flyte → Union**
LinkedIn: Pro-ML (2018) -> **Migrated to Flyte**
Flipkart: Hunch (2018) -> **Working with them**
Stripe: Railyard (2019) -> **Migrated to Flyte**
Gojek: Machine Learning Platform (2019) -> **Migrated to Flyte**
Lyft: Flyte (2020) -> **:)**
DoorDash: ML platform (2020) -> **Doordash EU migrated to Flyte**

Deployment options

- Managed by Union
- Managed by Flyte OSS Core + Union features
- Managed by Customer

Union BYOC
(Bring your own cloud)



Cloud native
enterprises

Thank you / Union.ai

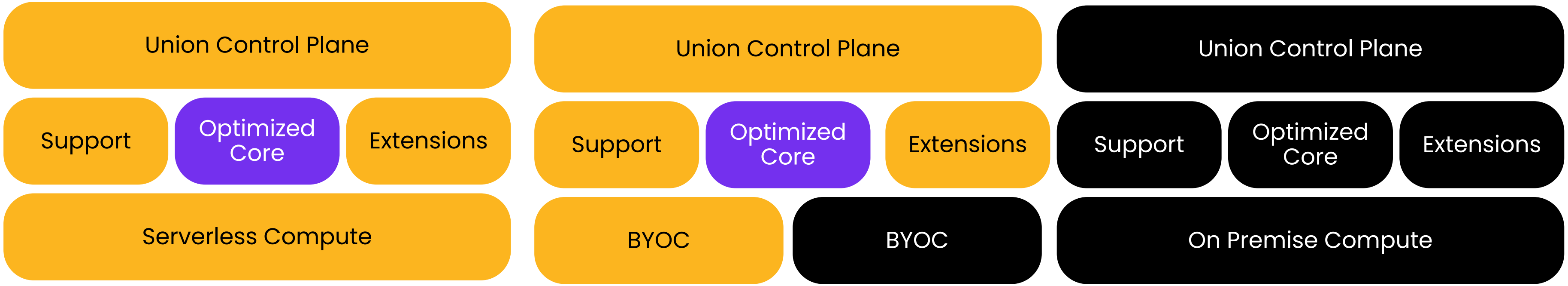
Deployment options

- Managed by Union
- Managed by Flyte OSS Core + Union features
- Managed by Customer (Planned)

Union Serverless

Union BYOC (Bring your own cloud)

Union On Premise



Small teams
+ individuals

Cloud native
enterprises

Enterprises with
on-prem requirements

Pricing

Utilize Union and only pay for the resources managed by the platform. Benefit from spot instances with recovery, enhanced caching, and the ability to scale to zero.

Flyte

Free
(support options available)

DIY ML orchestration for teams with on-premise or bare metal deployments.

- Open source (Apache 2.0)
- Battle-tested at scale
- Deployable on-premise
- Vibrant community
- Union support plans available

UNION
Serverless

Pay-as-you-go*
$30 in free credit

Ship your first production model in seconds without worrying about infrastructure. Ideal for individuals.

- Optimized and expanded Flyte
- Limited to 1 seat
- Scalable serverless environment
- Pay only for consumed resources
- Sign up instantly with GitHub

UNION
BYOC - Startup

$500/mo + % of compute
(on-demand retail pricing)

A secure and scalable platform ideal for small teams and early stage companies.

- Optimized and expanded Flyte
- Up to 5 seats
- Single cluster/cloud
- Purchase on AWS or GCP marketplace
- Standard bring-your-own-cloud (BYOC) deployment

UNION
BYOC - Enterprise

Custom
(w/ committed use discounts)

Built for enterprises that require large-scale, highly available, and customizable deployments.

- Optimized and expanded Flyte
- Unlimited seats
- Multi-cluster/cloud
- Purchase on AWS or GCP marketplace
- Customizable bring-your-own-cloud (BYOC) deployment

ARCHIVE

Use Cases

Generative AI
- Fine tuning
- RAG data ingestion
- Embedding
- Multimodal training & inference

Logos: Flawless, LinkedIn

Finance/FinTech
- AML (JPM)
- Fraud detection (Stripe)
- Time-series forecasting
- FP&A (Spotify)

Geospatial
- Satellite imagery
- Mapping
- Data Extraction

Logos: MethaneSAT, Muon Space

Bioinformatics/Pharma
- Protein Engineering
- Therapeutics
- Drug discovery
- Antibodies
- Compound discovery (Zymergen)

Consumer
- Recommendation Systems (HBO)
- Personalization (LinkedIn)

Logistics
- ETA
- Operational Research

Logos:
- Gojek

Autonomy & Robotics
- Computer Vision
- Perception
- SLAM

Logos:
- Tesla, Physical Intelligence, Toyota, Mercedes, Wayve, StackAV

Retail
- Churn prediction
- Pricing

Wallapop

Use Cases

Generative AI

- Fine tuning
- RAG data ingestion
- Embedding
- Multimodal training

Finance / FinTech

- Fine tuning
- RAG data ingestion
- Embedding
- Multimodal training

Geospatial

- Fine tuning
- RAG data ingestion
- Embedding
- Multimodal training

Bioinformatics + Pharma

- Fine tuning
- RAG data ingestion
- Embedding
- Multimodal training

Copy needs to be updated

Consumer

- Fine tuning
- RAG data ingestion
- Embedding
- Multimodal training

Logistics

- Fine tuning
- RAG data ingestion
- Embedding
- Multimodal training

Autonomy + Robotics

- Fine tuning
- RAG data ingestion
- Embedding
- Multimodal training

Retail

- Fine tuning
- RAG data ingestion
- Embedding
- Multimodal training

Why Flyte + Union

Spotify: Luigi (2013) -> **Migrated to Flyte**
Facebook: FBLearner Flow (2016) -> **We have been talking with them**
Uber: Michaelangelo (2017)
Google: TFX (2017) **-> I have a talk at Google today**
LinkedIn: Pro-ML (2018) -> **Migrated to Flyte**
Flipkart: Hunch (2018) -> **Working with them**
Netflix: Metaflow (2018)
Stripe: Railyard (2019) -> **Migrated to Flyte**
Pinterest: Galaxy (2019)
Gojek: Machine Learning Platform (2019) -> **Migrated to Flyte**
Lyft: Flyte (2020) -> **:)**
DoorDash: ML platform (2020) -> **Doordash EU migrated to Flyte**

Rate limit velocity is impacted by org silos

ML Engineers
+ Researchers

Models

Software
Engineers

Infrastructure

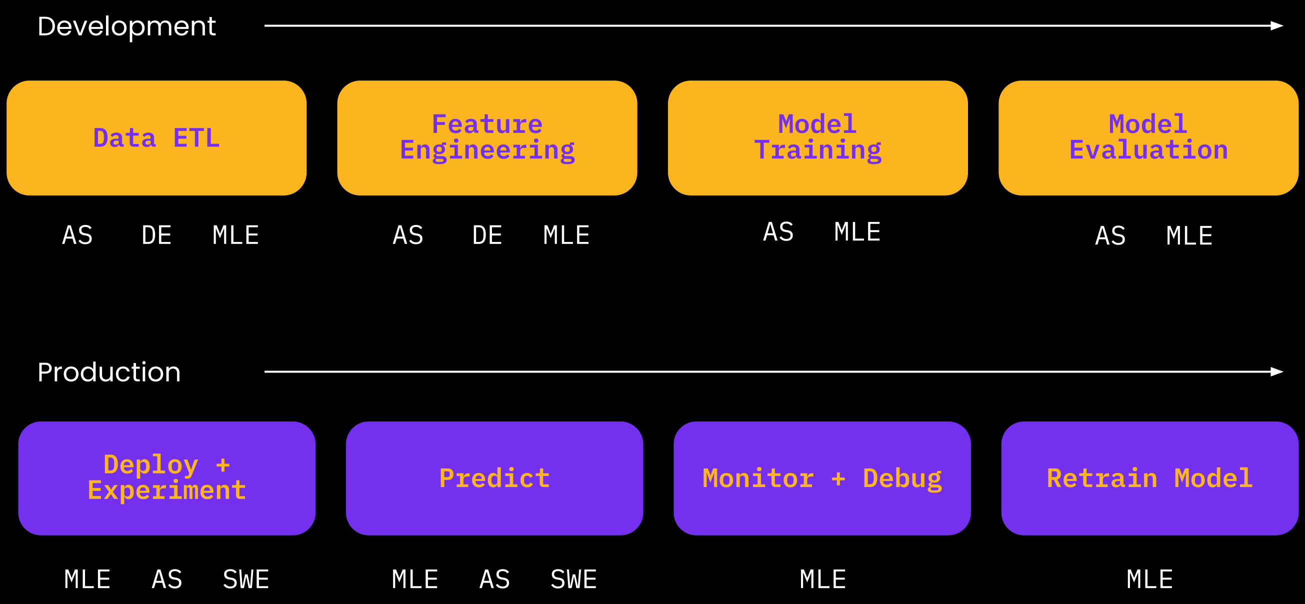
Data
Engineers

Data

But, AI product development is HARD

The ML lifecycle is iterative + collaborative with many different roles

- Applied Scientist (AS)
- Data Engineer (DE)
- ML Engineer (MLE)
- Software Engineer (SWE)



Development

Data ETL — AS DE MLE

Feature Engineering — AS DE MLE

Model Training — AS MLE

Model Evaluation — AS MLE

Production

Deploy + Experiment — MLE AS SWE

Predict — MLE AS SWE

Monitor + Debug — MLE

Retrain Model — MLE

But, AI Product development is HARD

AI Researchers
Data Science
Applied Research

Models

ML Eng/
Software
Engineers

Code < / >

Platform
Engineers

Infrastructure

Data
Engineers

Data

But, AI product development is HARD

AI Researchers & Data Scientists

Models

ML & Software Engineers

Code

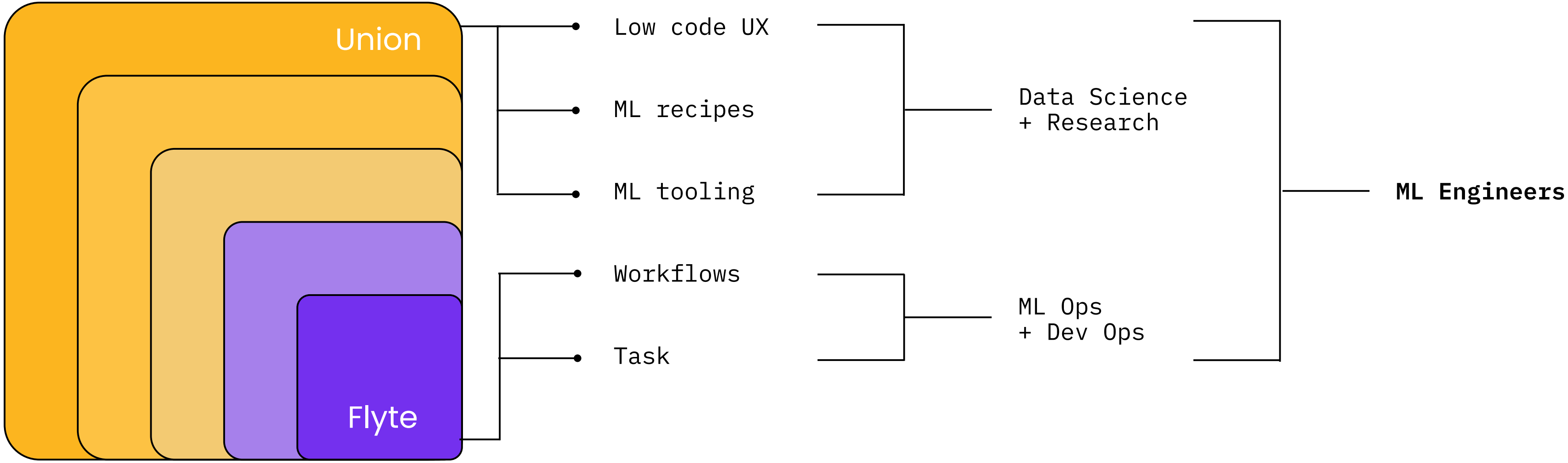
Platform Engineers

Infrastructure

Data Engineers

Data

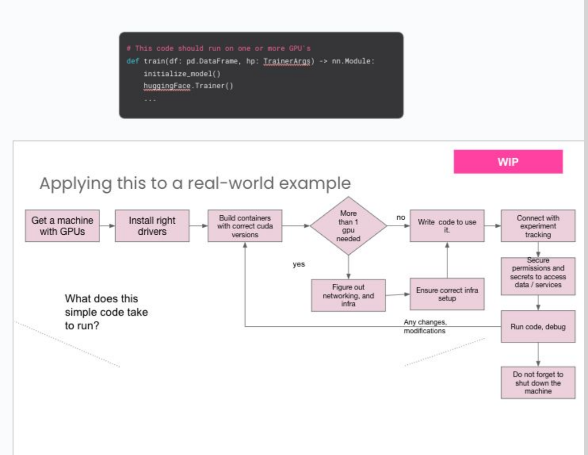
Union products expand the Flyte system



Companies switch to Flyte & Union

From this to this

Today



```
# This code should run on one or more GPU's
def train(df: pd.DataFrame, hp: TrainerArgs) -> nn.Module:
    initialize_model()
    huggingFace.Trainer()
    ...
```

WIP

Applying this to a real-world example

Get a machine with GPUs

Install right drivers

Build containers with correct cuda versions

More than 1 gpu needed

no

yes

Write code to use it.

Figure out networking, and infra

Ensure correct infra setup

Connect with experiment tracking

Secure permissions and secrets to access data / services

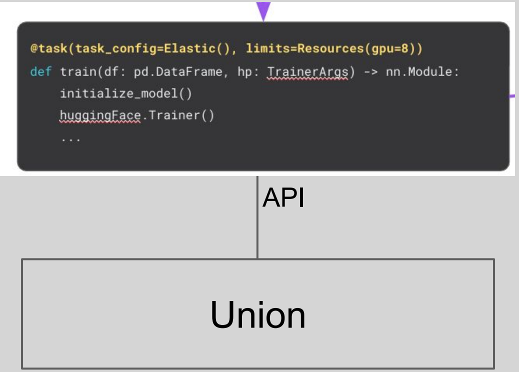
Run code, debug

Any changes, modifications

Do not forget to shut down the machine

What does this simple code take to run?

With Union

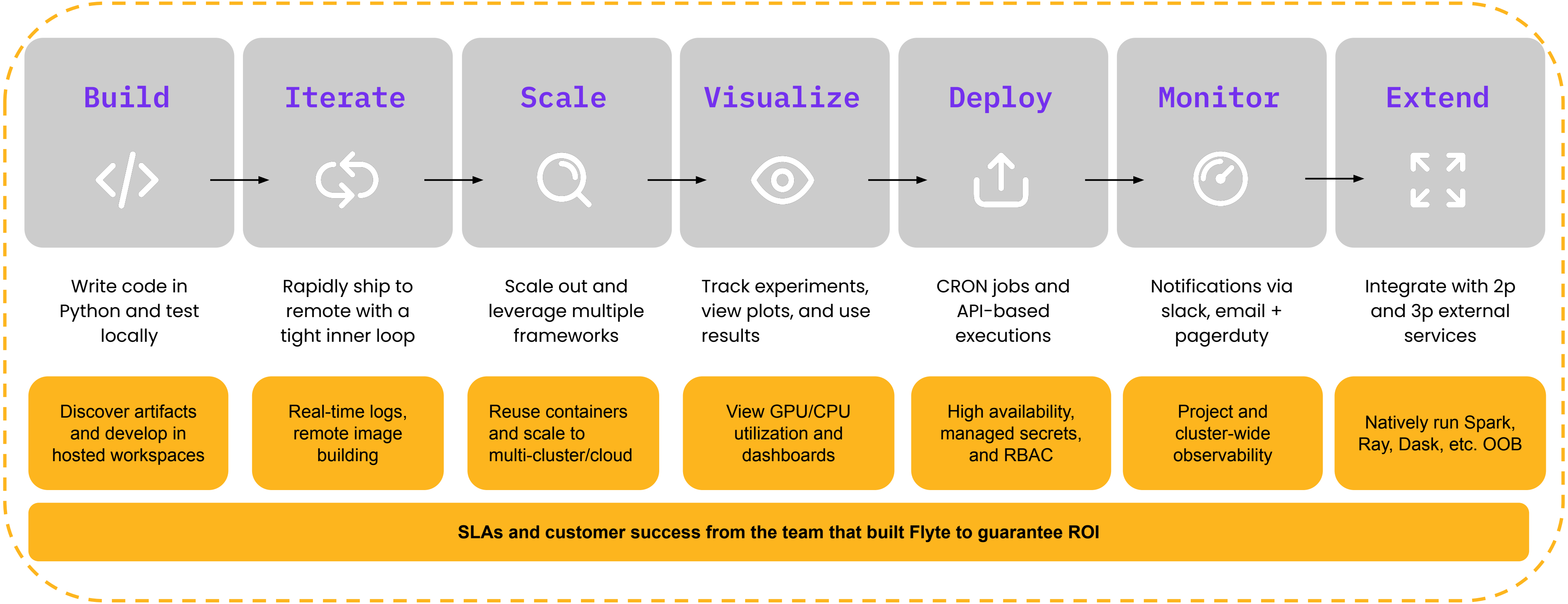


```
@task(task_config=Elastic(), limits=Resources(gpu=8))
def train(df: pd.DataFrame, hp: TrainerArgs) -> nn.Module:
    initialize_model()
    huggingFace.Trainer()
    ...
```

Union meets the requirements for AI dev

- **Reproducible:** Automatic Versioning, Containerization, Data Immutability, Durable State & Results
- **Programmable**: Declarative Infrastructure, Declarative Data Flow, Type-Safety & Type-Awareness, Pythonic (easy to adopt)
- **Composable**: Reusable Tasks & Workflows, Reusable Data (Artifacts), Heterogeneous Workflows
- **Interoperable**: Agents (Connect to 2p and 3p s), API-driven development, Framework-Agnostic (PyTorch, Tensorflow, etc)
- **Scalable**: Local-Remote Parity, Multi-tenant, Scalable Compute Fabric
- **Production-ready**: Scheduling, Notifications, Observability, Isolated Dev/Prod Environments
- **Reliable**: Retries, Checkpointing, Failure Recovery, No SPOF, Multi-AZ, Multi-Cluster
- **Efficiency**: Caching, Ephemeral Compute, Spot Instances, Fractional GPU

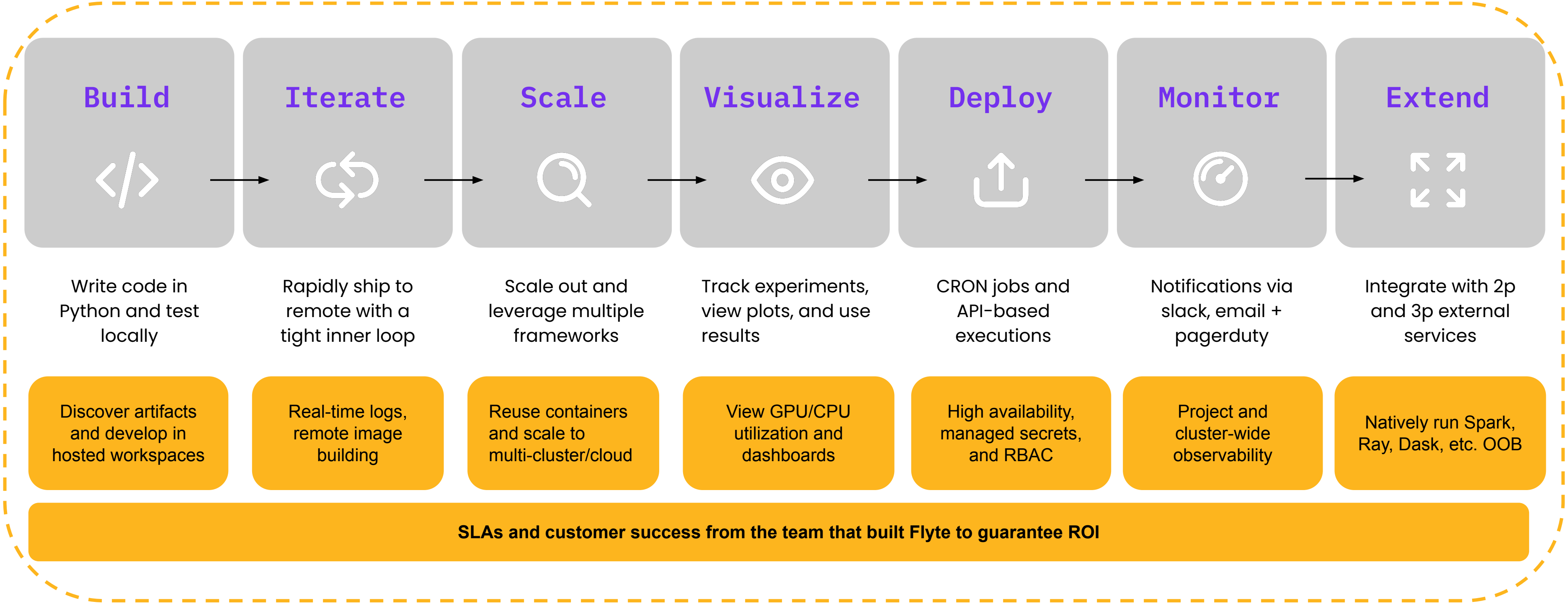
Union is Flyte, supercharged



Build	Iterate	Scale	Visualize	Deploy	Monitor	Extend
Write code in Python and test locally	Rapidly ship to remote with a tight inner loop	Scale out and leverage multiple frameworks	Track experiments, view plots, and use results	CRON jobs and API-based executions	Notifications via slack, email + pagerduty	Integrate with 2p and 3p external services
Discover artifacts and develop in hosted workspaces	Real-time logs, remote image building	Reuse containers and scale to multi-cluster/cloud	View GPU/CPU utilization and dashboards	High availability, managed secrets, and RBAC	Project and cluster-wide observability	Natively run Spark, Ray, Dask, etc. OOB

SLAs and customer success from the team that built Flyte to guarantee ROI

Union is Flyte, supercharged



Build

Write code in Python and test locally

Discover artifacts and develop in hosted workspaces

Iterate

Rapidly ship to remote with a tight inner loop

Real-time logs, remote image building

Scale

Scale out and leverage multiple frameworks

Reuse containers and scale to multi-cluster/cloud

Visualize

Track experiments, view plots, and use results

View GPU/CPU utilization and dashboards

Deploy

CRON jobs and API-based executions

High availability, managed secrets, and RBAC

Monitor

Notifications via slack, email + pagerduty

Project and cluster-wide observability

Extend

Integrate with 2p and 3p external services

Natively run Spark, Ray, Dask, etc. OOB

SLAs and customer success from the team that built Flyte to guarantee ROI

Customer innovation

Spotify TOYOTA LOCKHEED MARTIN LinkedIn stripe freenome

TESLA amazon Microsoft NVIDIA Cradle

Training & Fine Tuning

Confidently run large-scale training or fine-tuning on GPU clusters across clouds and on-premise

Data Processing

Seamlessly connect to your data stack. Focus on data, not infrastructure.

Near-Line Inference

Deliver high-throughput, reliable, and fault-tolerant inference for production AI apps

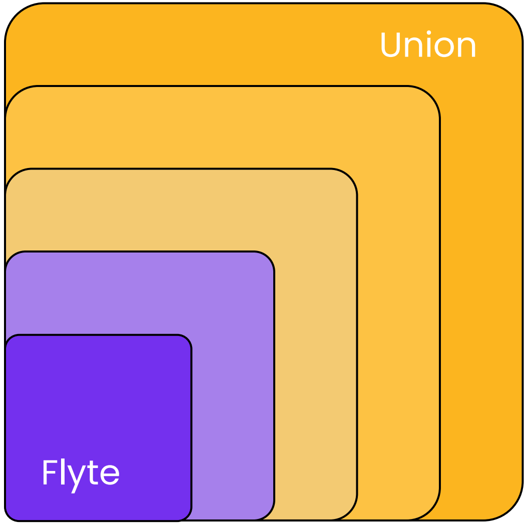
Generative AI & LLMs

Take generative AI applications to production faster

Bioinformatics & Pharma

Effortlessly carry out scientific computing workflows with must-have features out of the box

Union supercharges and extends Flyte



More efficient
Boost ROI by enabling teams to access the resources they need while sharing underlying infrastructure.

Better developer experience
Shorten the development loop from hours to seconds while writing production-ready code.

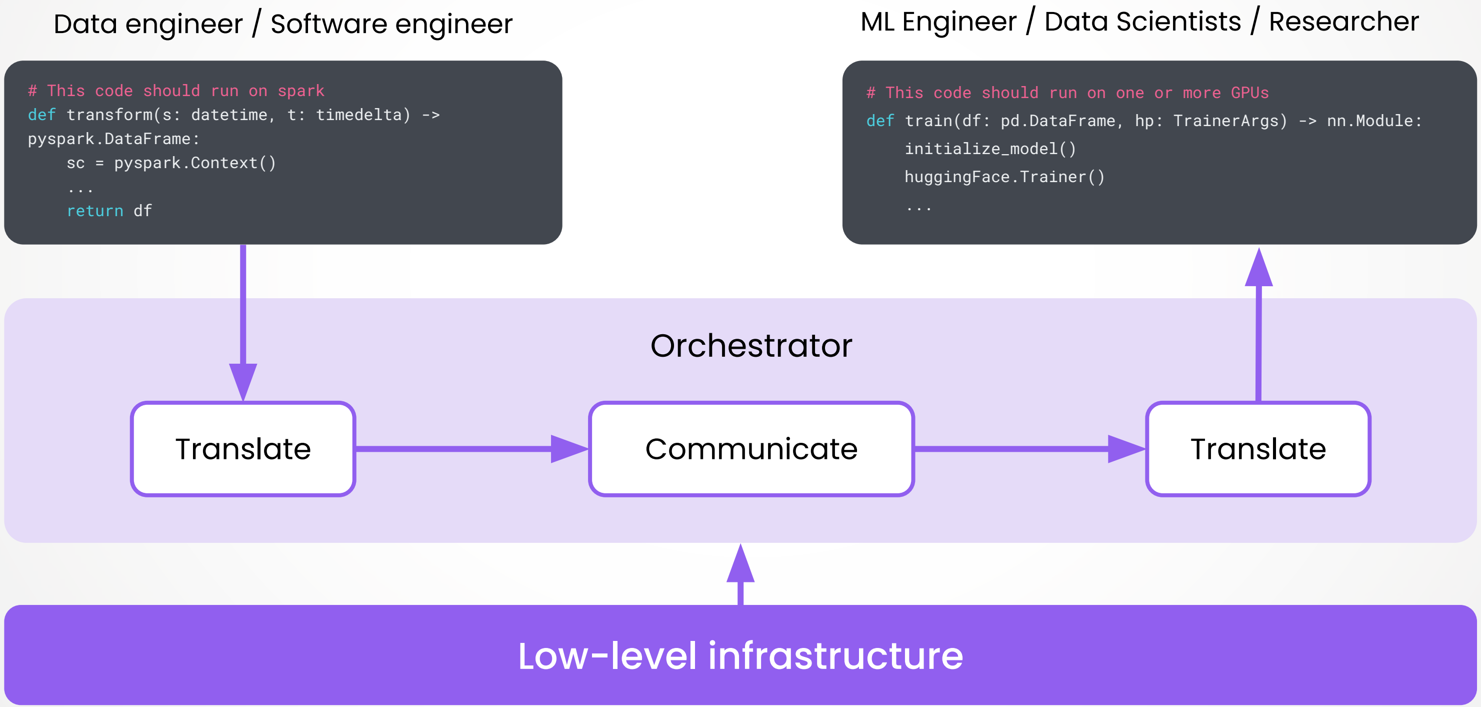
Supercharged performance
Run complex AI workloads with unparalleled performance, scale, and efficiency.

Fully managed & secure
Leverage a robust platform that meets rigorous standards for security, compliance, and operational reliability - in your cloud.

SLAs and customer success from the team that built Flyte

WIP

You need a system that abstracts sharing & Infrastructure



Data engineer / Software engineer

```
# This code should run on spark
def transform(s: datetime, t: timedelta) ->
pyspark.DataFrame:
    sc = pyspark.Context()
    ...
    return df
```

ML Engineer / Data Scientists / Researcher

```
# This code should run on one or more GPUs
def train(df: pd.DataFrame, hp: TrainerArgs) -> nn.Module:
    initialize_model()
    huggingFace.Trainer()
    ...
```

Orchestrate Your AI

Bring together ML, Platform, Data and Ops teams to create AI products efficiently

Flyte, supercharged

All of the features in flyte, optimized for speed and enhanced for dynamic execution and managed K8s

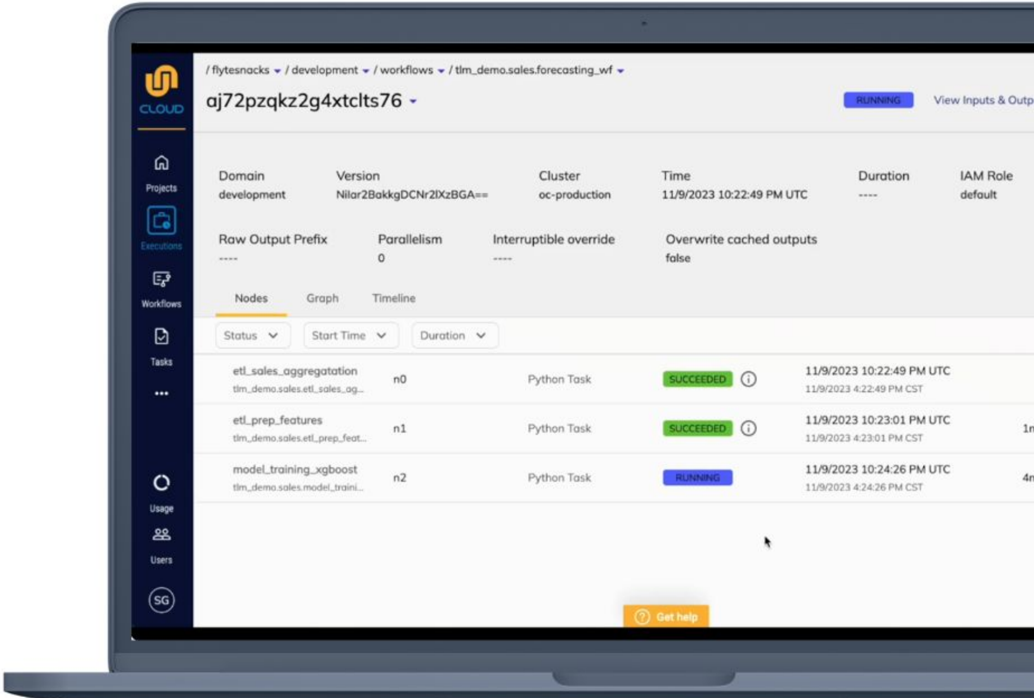
Unified workstreams

Modern AI orchestration that joins teams to productionize AI apps, process and workflows

Maximized AI ROI, derisked

Reduce operating costs with efficient resource management, while increasing velocity

All built on a foundation of trust



/ flytesnacks / development / workflows / tlm_demo.sales.forecasting_wf

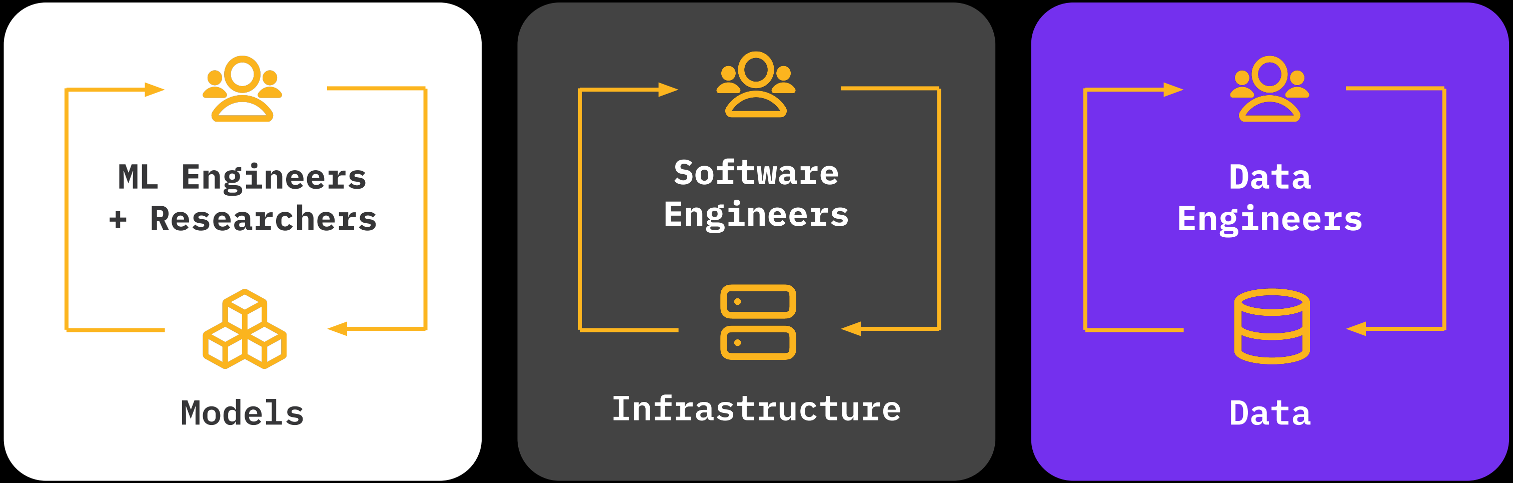
aj72pzqkz2g4xtclts76

Domain	Version	Cluster	Time	Duration	IAM Role
development	Nilar2BakkgDCNr2lXzBGA==	oc-production	11/9/2023 10:22:49 PM UTC	----	default

Raw Output Prefix	Parallelism	Interruptible override	Overwrite cached outputs
----	0	----	false

etl_sales_aggregatation tlm_demo.sales.etl_sales_ag...	n0	Python Task	SUCCEEDED	11/9/2023 10:22:49 PM UTC 11/9/2023 4:22:49 PM CST
etl_prep_features tlm_demo.sales.etl_prep_feat...	n1	Python Task	SUCCEEDED	11/9/2023 10:23:01 PM UTC 11/9/2023 4:23:01 PM CST
model_training_xgboost tlm_demo.sales.model_traini...	n2	Python Task	RUNNING	11/9/2023 10:24:26 PM UTC 11/9/2023 4:24:26 PM CST

Rate limit velocity is impacted by org silos



ML Engineers + Researchers

Models

Software Engineers

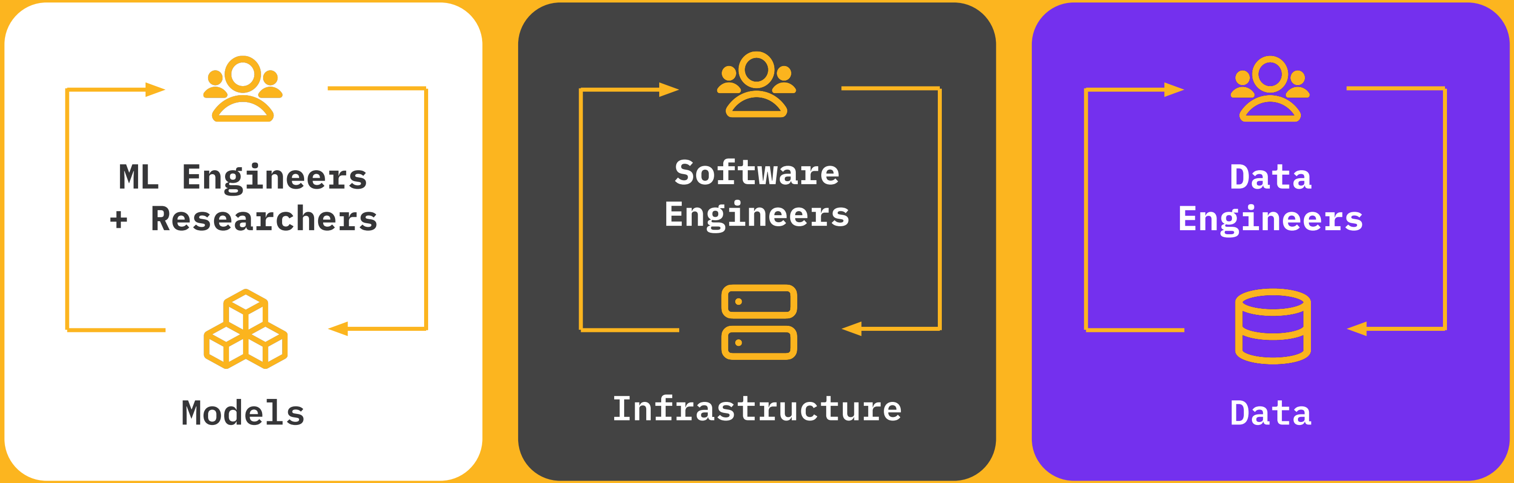
Infrastructure

Data Engineers

Data

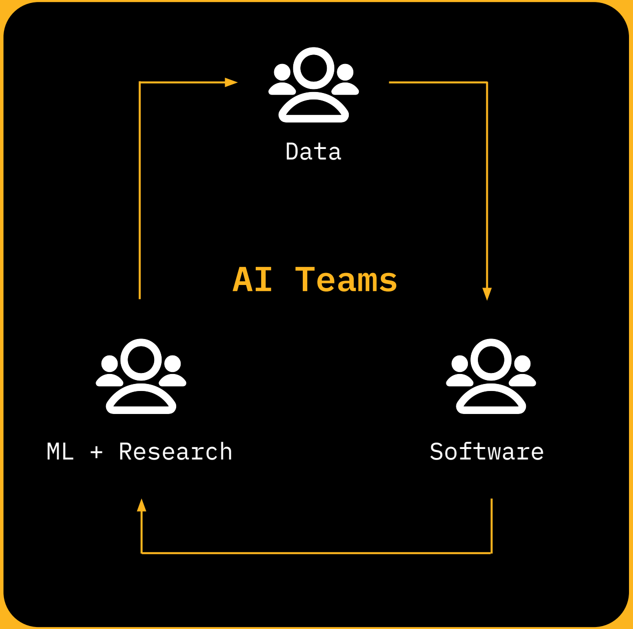
UNION

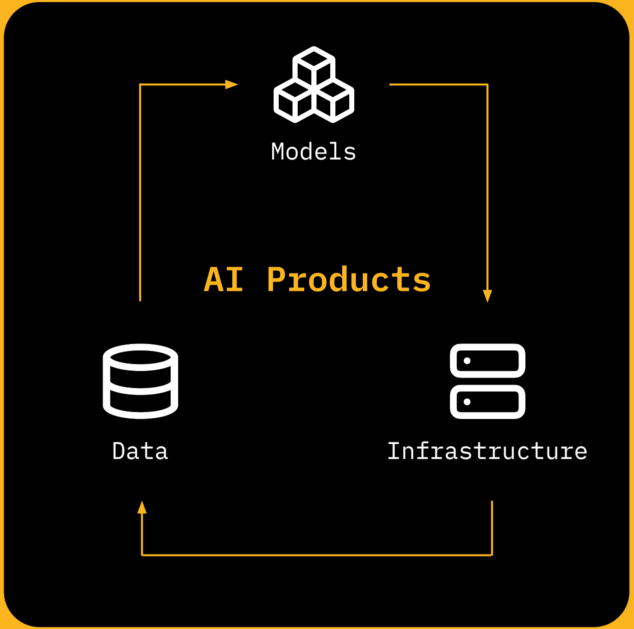
Bring engineering, ops, and data science together to create AI products efficiently



Bring engineering, ops, and data science together to create AI products efficiently

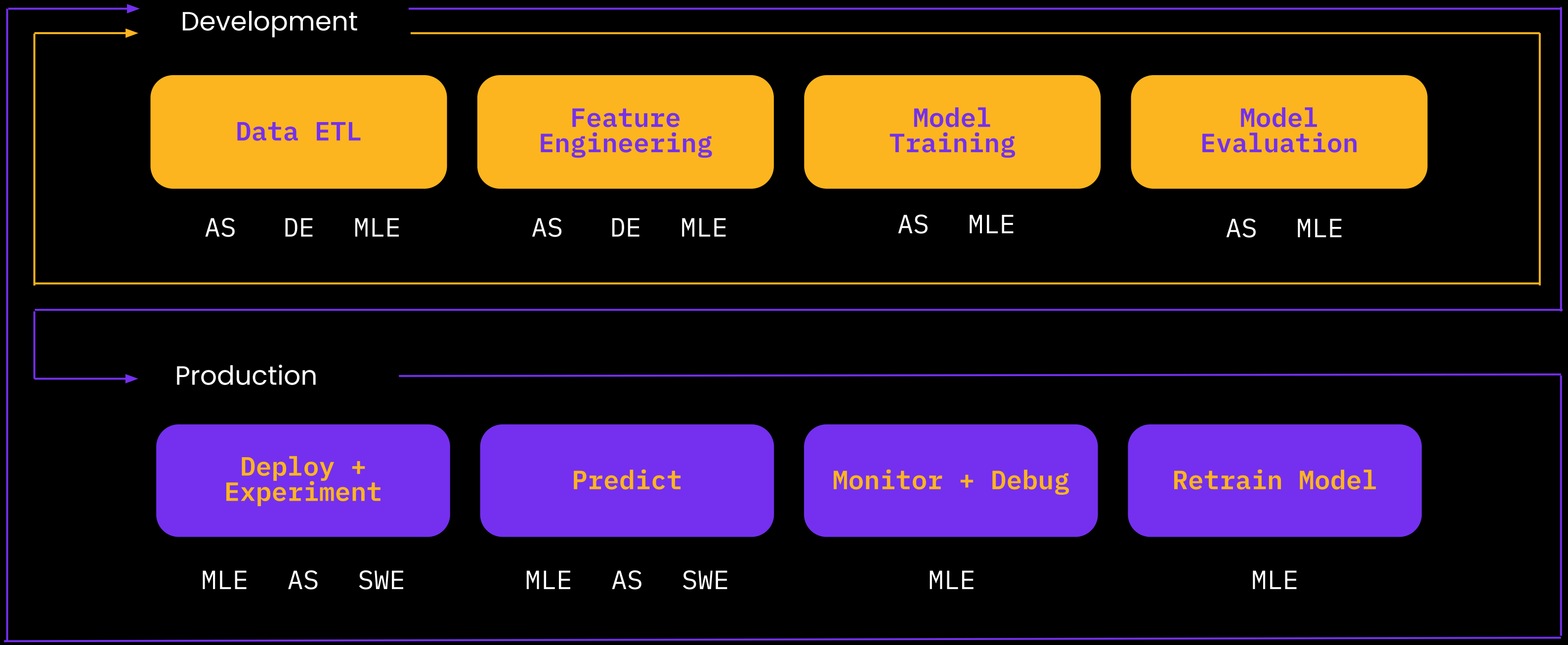
WIP





AI product development is **hard + chaotic**

WIP



The standard AI orchestratoin platform

WIP

Spotify: Luigi (2013) -> **Migrated to Flyte → Union**
Facebook: FBLearner Flow (2016) -> **We have been talking with them**
Uber: Michaelangelo (2017)
Google: TFX (2017) **-> I have a talk at Google today**
LinkedIn: Pro-ML (2018) -> **Migrated to Flyte**
Flipkart: Hunch (2018) -> **Working with them**
Netflix: Metaflow (2018)
Stripe: Railyard (2019) -> **Migrated to Flyte**
Pinterest: Galaxy (2019)
Gojek: Machine Learning Platform (2019) -> **Migrated to Flyte**
Lyft: Flyte (2020) -> **:)**
DoorDash: ML platform (2020) -> **Doordash EU migrated to Flyte**

Storyline

AI product development is CHAOTIC
→ need to anchor on a customer story/challenge - all the different tools and systems, the constant loop
→ the fact that the models are iterative - this is new type of development

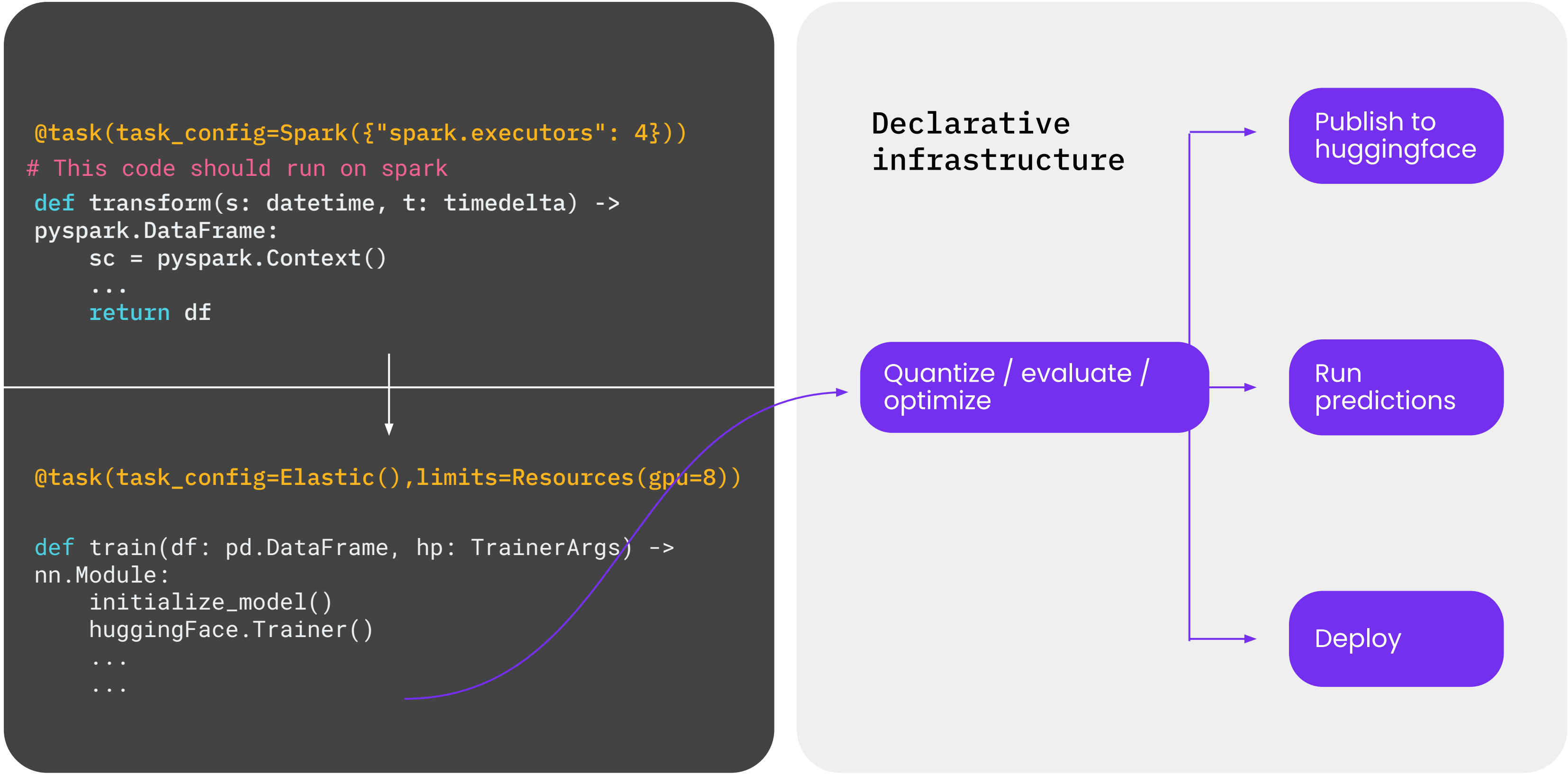
Solving this problem is Hard
→ because it is deep tech
→ and we have the issue of silo’s and people/process

We are solving this problem today (SHOW THE HOW)

- Solve the silo problem
- Solve the deep tech problem

→ the result is a simple platform where you build your AI products (Union is the fabric)

Applying this to a real-world example

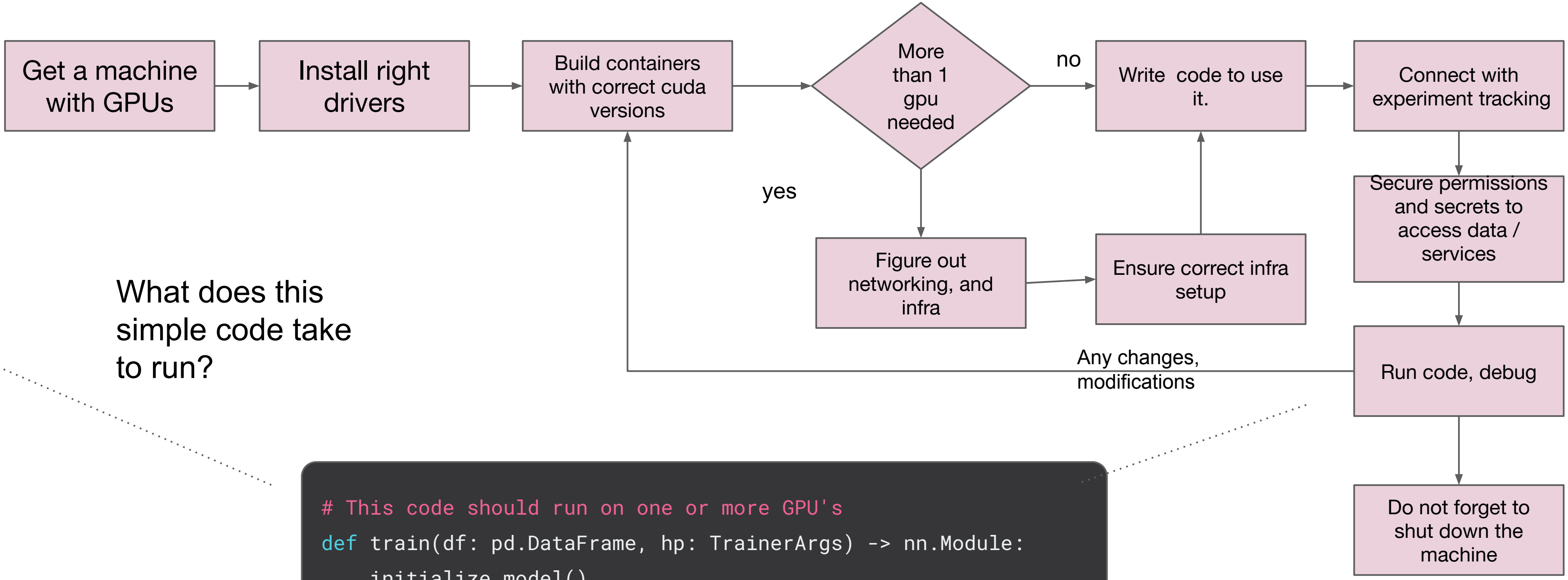


```
@task(task_config=Spark({"spark.executors": 4}))
# This code should run on spark
def transform(s: datetime, t: timedelta) ->
pyspark.DataFrame:
    sc = pyspark.Context()
    ...
    return df
```

```
@task(task_config=Elastic(),limits=Resources(gpu=8))

def train(df: pd.DataFrame, hp: TrainerArgs) ->
nn.Module:
    initialize_model()
    huggingFace.Trainer()
    ...
    ...
```

WIP

Applying this to a real-world example

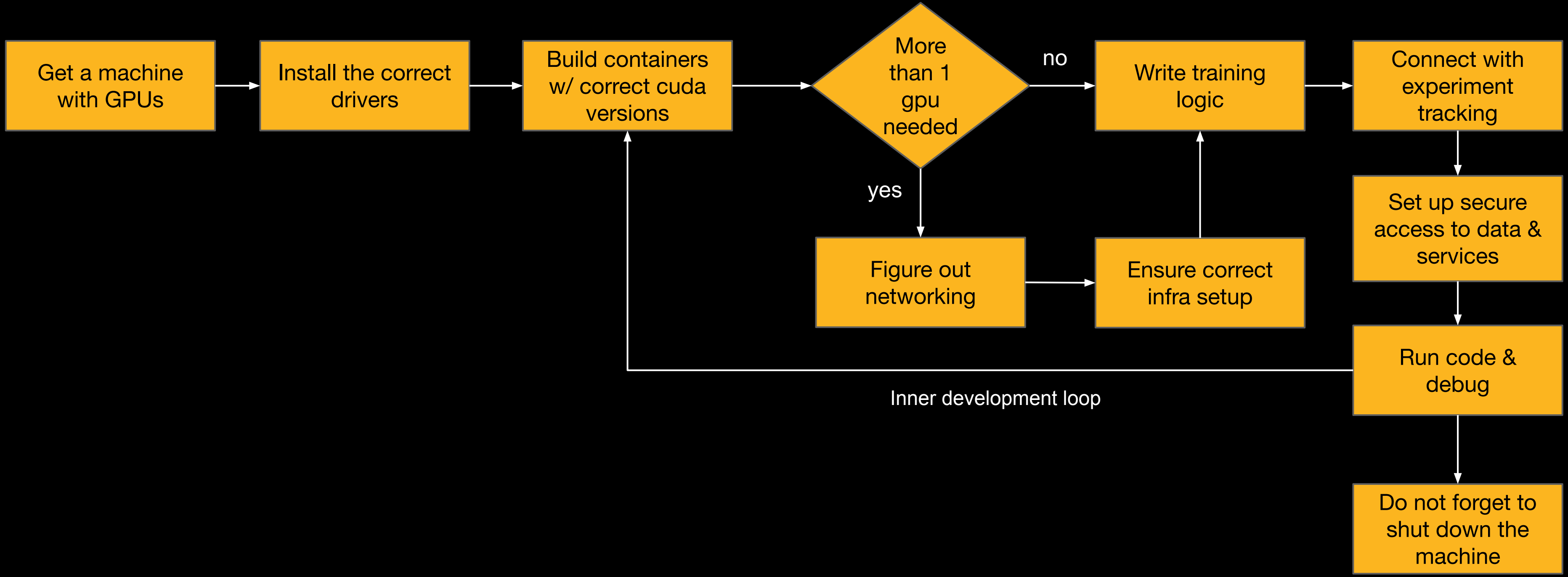

Get a machine with GPUs
Install right drivers
Build containers with correct cuda versions
More than 1 gpu needed
no
yes
Write code to use it.
Connect with experiment tracking
Secure permissions and secrets to access data / services
Figure out networking, and infra
Ensure correct infra setup
Any changes, modifications
Run code, debug
Do not forget to shut down the machine

What does this simple code take to run?

```
# This code should run on one or more GPU's
def train(df: pd.DataFrame, hp: TrainerArgs) -> nn.Module:
    initialize_model()
    huggingFace.Trainer()
    ...
```

Imagine every mle / data scientists has to do this. The cost, access management and scale?

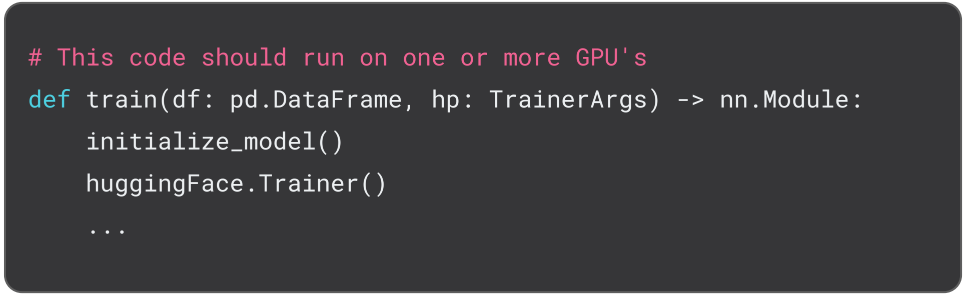
Real-world example



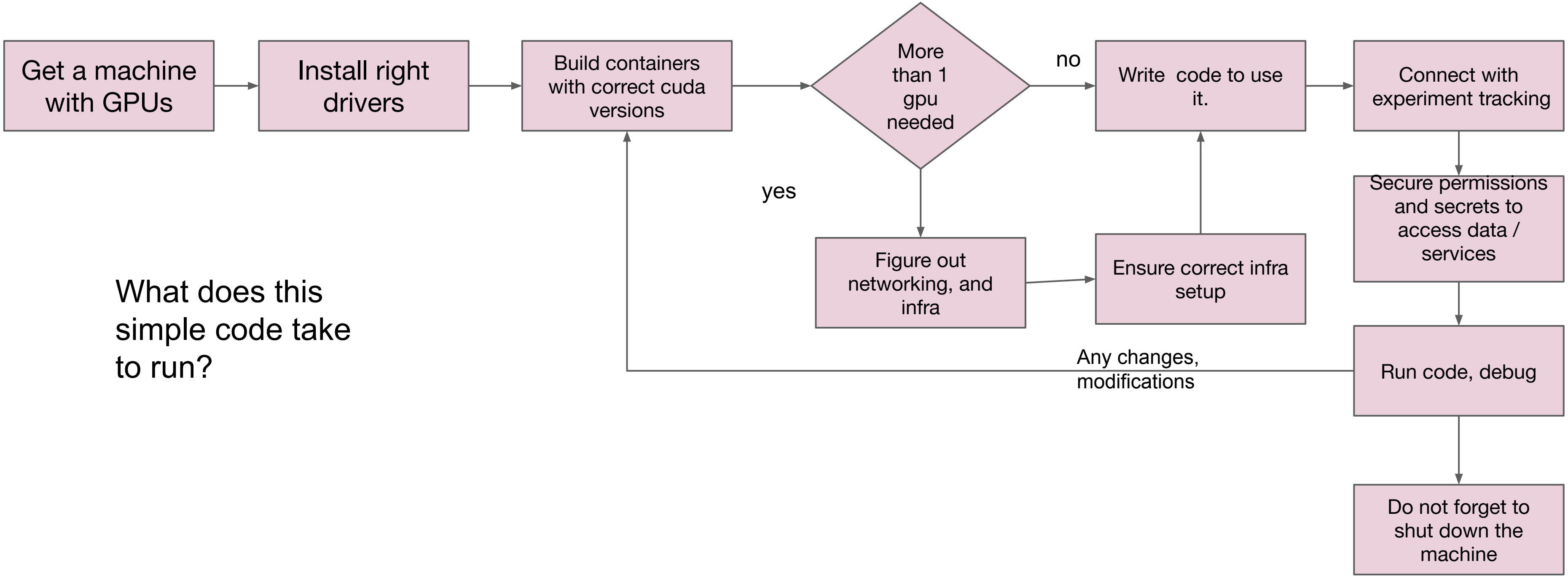
Get a machine with GPUs
Install the correct drivers
Build containers w/ correct cuda versions
More than 1 gpu needed
no
yes
Figure out networking
Ensure correct infra setup
Write training logic
Connect with experiment tracking
Set up secure access to data & services
Run code & debug
Do not forget to shut down the machine
Inner development loop

Applying this to a real-world ex

WIP



```
# This code should run on one or more GPU's
def train(df: pd.DataFrame, hp: TrainerArgs) -> nn.Module:
    initialize_model()
    huggingFace.Trainer()
    ...
```



Get a machine with GPUs

Install right drivers

Build containers with correct cuda versions

More than 1 gpu needed

no

Write code to use it.

Connect with experiment tracking

yes

Figure out networking, and infra

Ensure correct infra setup

Secure permissions and secrets to access data / services

Run code, debug

Any changes, modifications

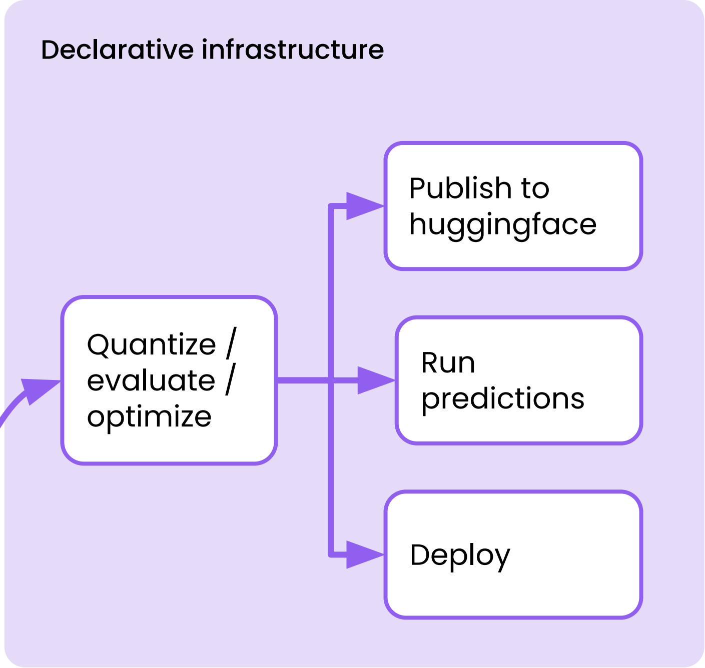
Do not forget to shut down the machine

What does this simple code take to run?

Applying this to a real-world example

```python
@task(task_config=Spark({"spark.executors": 4}))
def transform(s: datetime, t: timedelta) -> pyspark.DataFrame:
    sc = pyspark.Context()
    ...
    return df
```

```python
@task(task_config=Elastic(), limits=Resources(gpu=8))
def train(df: pd.DataFrame, hp: TrainerArgs) -> nn.Module:
    initialize_model()
    huggingFace.Trainer()
    ...
```



Declarative infrastructure

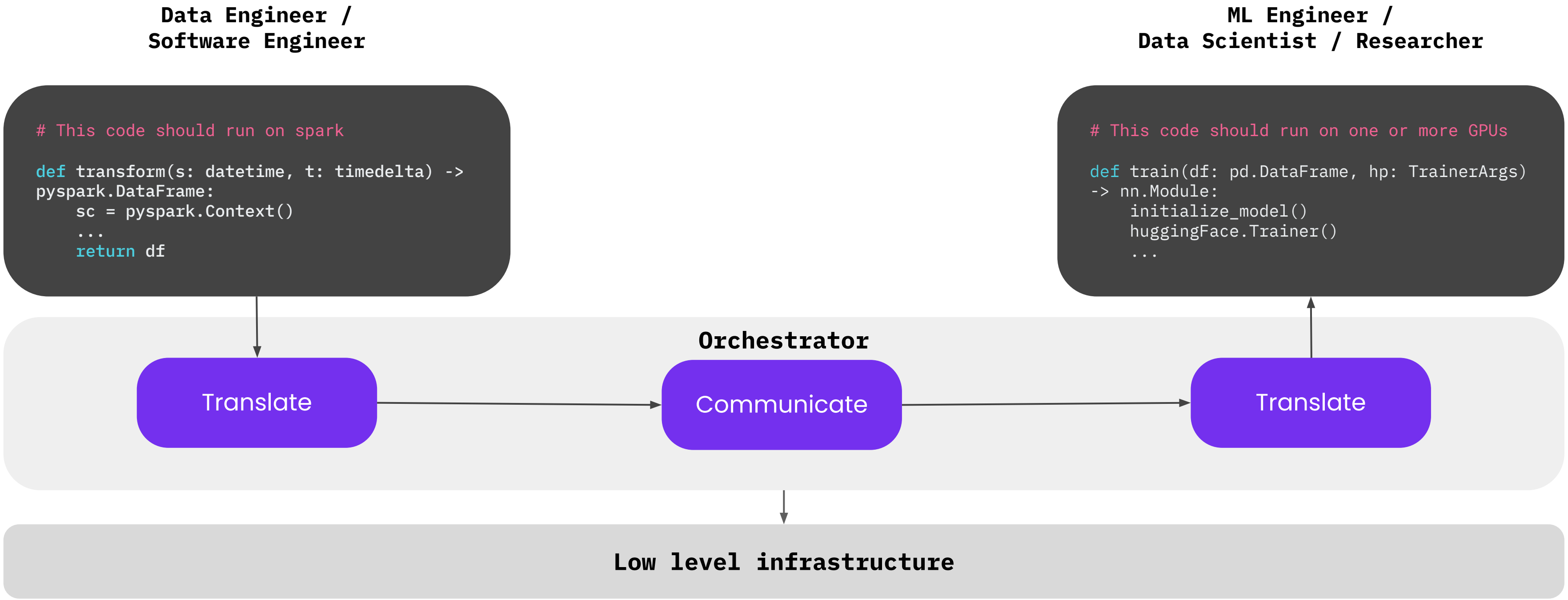
Quantize / evaluate / optimize

Publish to huggingface

Run predictions

Deploy

You need a system that abstracts sharing + infrastructure (which is HARD)



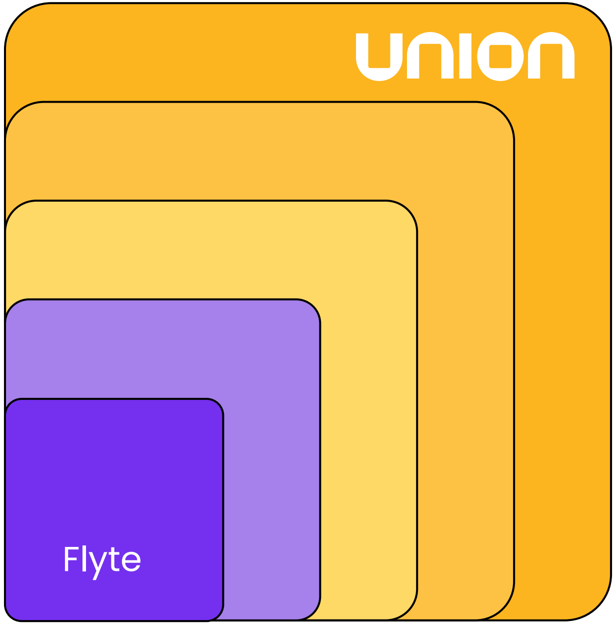
Data Engineer /
Software Engineer

```
# This code should run on spark

def transform(s: datetime, t: timedelta) ->
pyspark.DataFrame:
    sc = pyspark.Context()
    ...
    return df
```

ML Engineer /
Data Scientist / Researcher

```
# This code should run on one or more GPUs

def train(df: pd.DataFrame, hp: TrainerArgs)
-> nn.Module:
    initialize_model()
    huggingFace.Trainer()
    ...
```

Union supercharges + extends Flyte



More efficient
Boost ROI by enabling teams to access the resources they need while sharing underlying infrastructure.

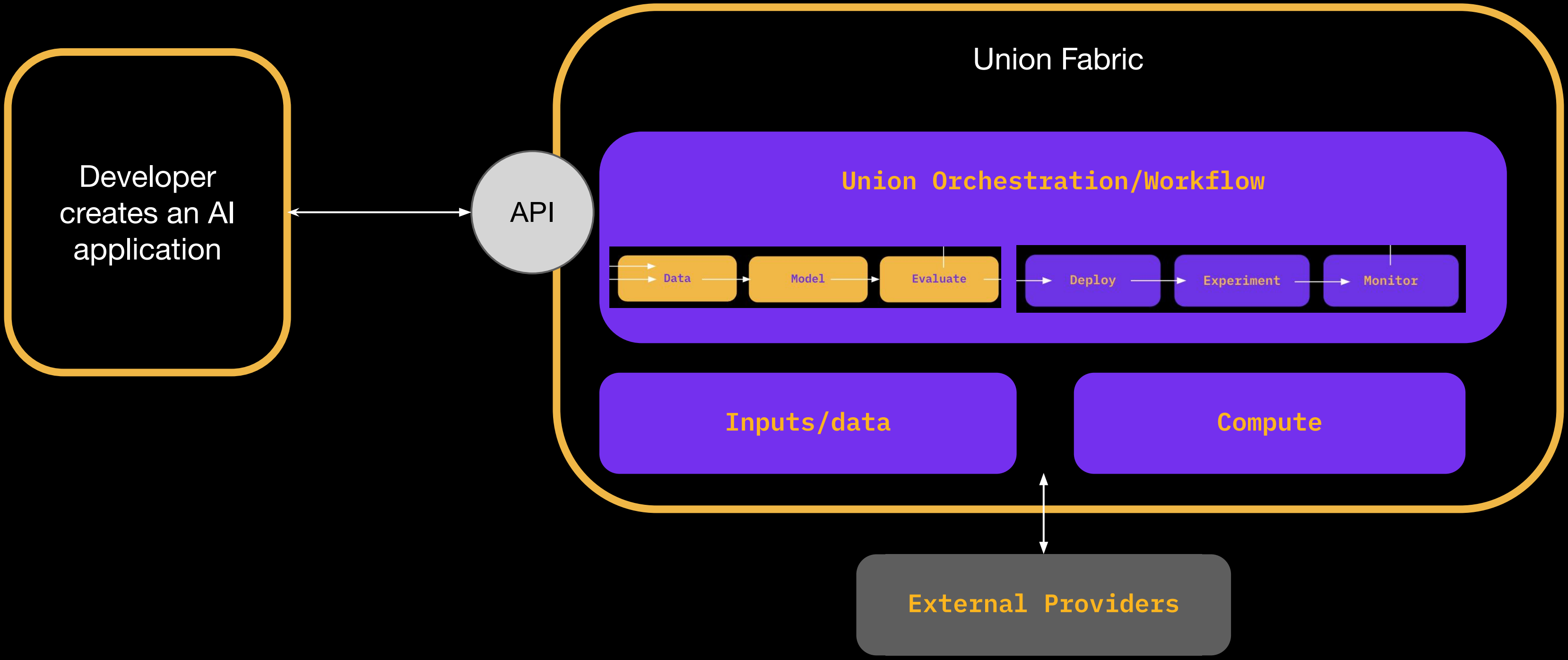
Enterprise developer experience
Shorten the development loop from hours to seconds while writing production-ready code.

Supercharged performance
Run complex AI workloads with unparalleled performance, scale, and efficiency.

Fully managed & secure
Leverage a robust platform that meets rigorous standards for security, compliance, and operational reliability - in your cloud.

SLAs + customer success from the team that built Flyte

Our mission at Union: Make creating AI products **reliable**, **secure** and **easy**


Developer creates an AI application
API
Union Fabric
Union Orchestration/Workflow
Data
Model
Evaluate
Deploy
Experiment
Monitor
Inputs/data
Compute
External Providers