

DO NOT DISTRIBUTE

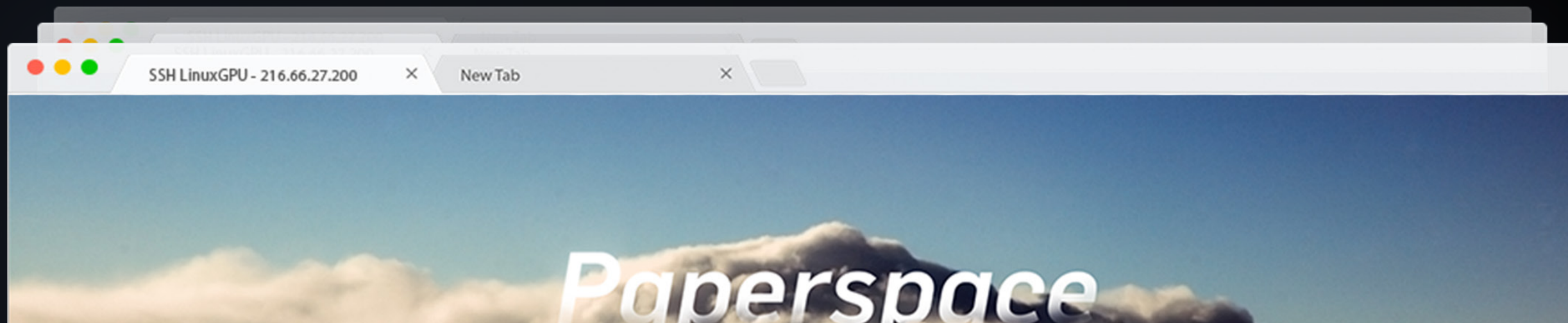
Paperspace

www.paperspace.com



Paperspace

Serverless AI for the future of intelligence.



Introduction

Deep Learning platform built for developers.

Infrastructure automation and software layer to build intelligent applications.



A new generation of AI
developers require a rethinking
of tooling and workflows.

Why this matters

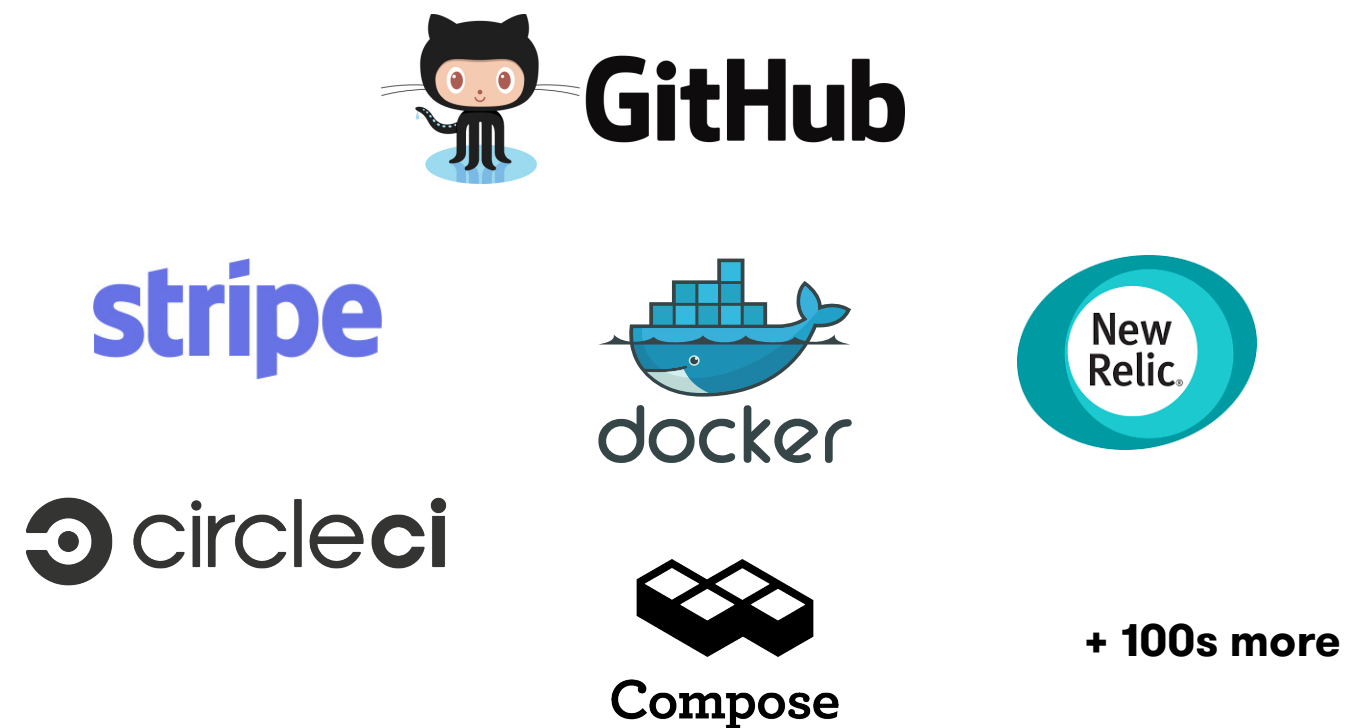
**Developers spend 75% of their time
managing infrastructure.**

So what's the underlying problem?

The cloud was built for a different use-case (**web servers**) and a different audience (**DevOps**).

The DevOps ecosystem is rich

Traditional Web Services



storage, CDN, deploy, monitor, VPC,
load balance, IPsec, CI/CD, DNS ...

Deep Learning

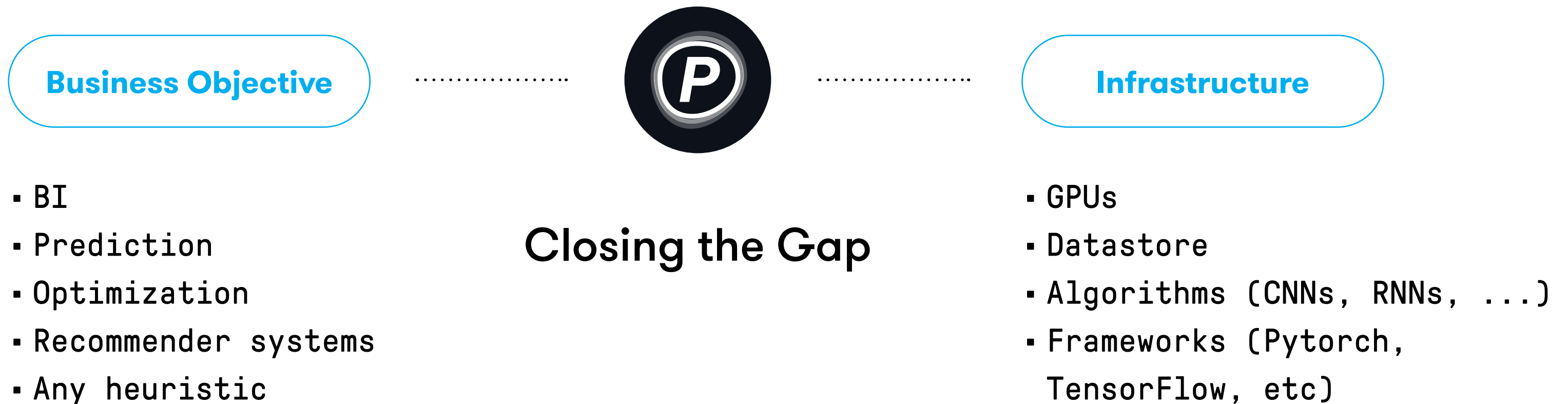


data, notebooks, train, visualize,
collaborate, version, hyperparameters ...

The key to solving this problem
is finding the right layer of
abstraction.

Put Uber/Facebook-grade AI platform in the hands of every developer

There is a huge disconnect between modern business objectives and the DL tools that can fulfill them.



Paperspace abstracts powerful infrastructure behind a simple software layer making cloud ML as easy as modern web services.

A complete platform for modern deep learning

Ingest → Train →
Analyze → Deploy

+

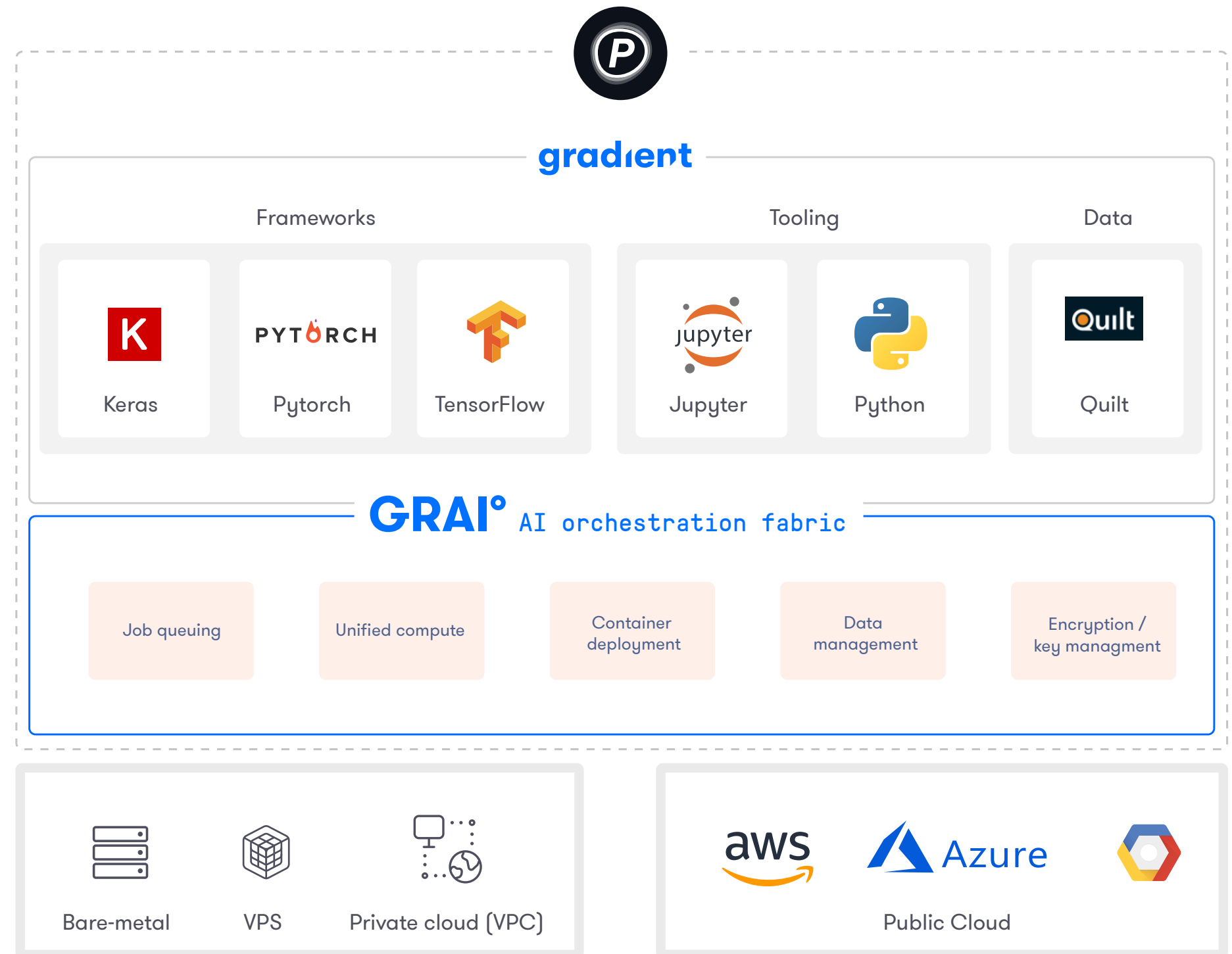
Manage, collaborate, share

- Fully-managed GPU infrastructure
- Unified dev experience
- 1 click Jupyter Notebooks/Lab
- API & language integrations
- ACL/team controls



GRAI° Model building AI orchestration fabric

- Job queuing / management
- Cloud agnostic
- Accelerator architecture native (GPU, FPGA, ASIC, TPU, etc)
- Unified compute
- Extensible
- Built on best practices (containers, kubernetes, and data policies)

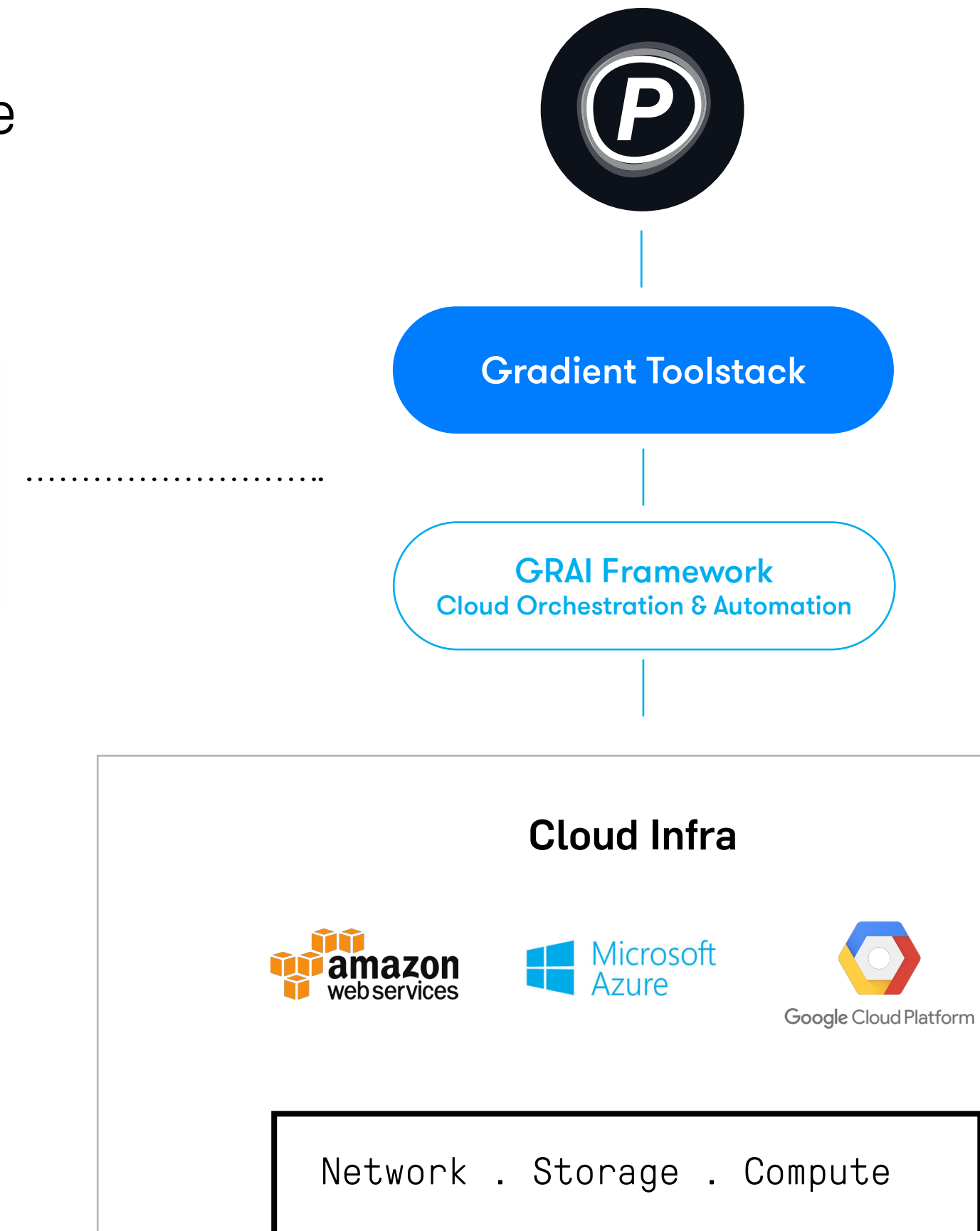


Cloudscale with a single line of code

```
> import paperspace as ps  
  
# Run job on GPU cluster  
  
> ps({'Type': 'TPU', 'container': 'TensorFlow' ... })
```

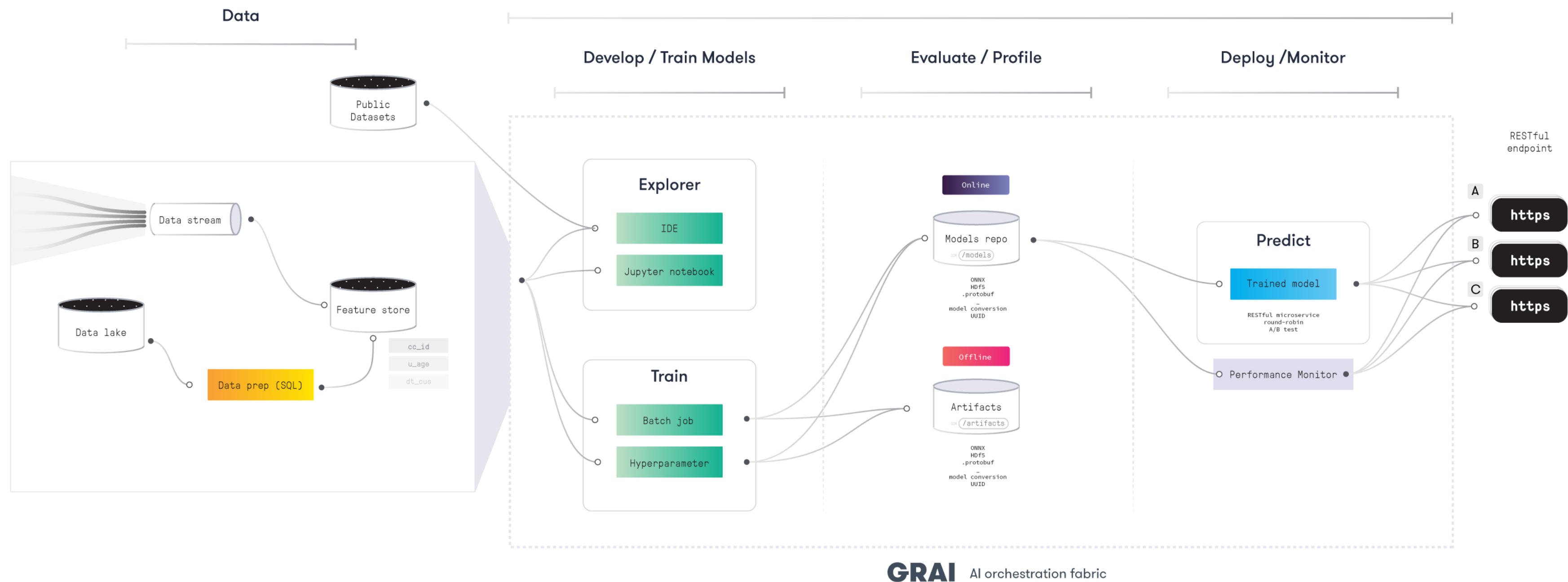
Connecting modern ML and the cloud
by converting infrastructure into
code.

Raw compute is not sufficient.



The Pipeline

gradient°



Remarks from the trenches

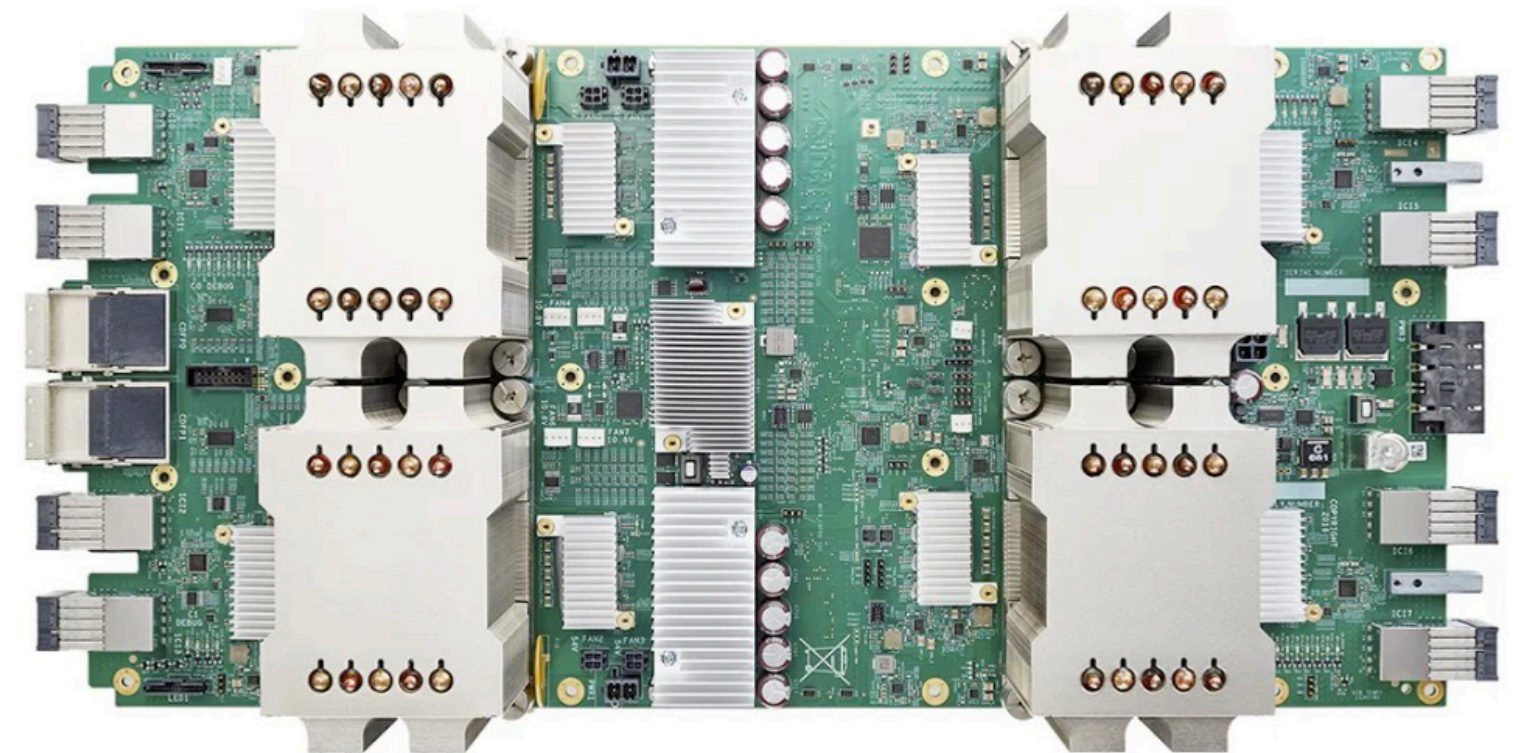
Trends:

1. Chip renaissance
2. Evolution of ML/AI in practice
3. Consolidation around best practices

Chip renaissance

- Graphcore
- Cerebras
- Nervana
- Wave
- Google TPU
- . . .

The big question today is whether accelerator architectures will follow commodity CPU x86 or lead to a golden era for high-end, use-specific hardware.



The evolution of ML/AI in practice

2016

2018

Consumable API → Refit the Model → Model as core IP

- Clarifai
- AWS Rekognition
- Google Cloud Vision
- MS cognitive services

- Paperspace
- Algorithmia
- FloydHub
- ClusterOne

Consolidation around best practices

- Containerization
- Jupyter
- Job runner architecture
- Pipeline
- etc.

Jupyter Notebook Example

Run ▶

Run some Python code!

To run the code below:

1. Click on the cell to select it.
2. Press **SHIFT+ENTER** on your keyboard or press the play button (▶) in the toolbar above.

A full tutorial for using the notebook interface is available [here](#).

In []:

```
%matplotlib notebook

import pandas as pd
import numpy as np
import matplotlib

from matplotlib import pyplot as plt
import seaborn as sns

ts = pd.Series(np.random.randn(1000), index=pd.date_range('1/1/2000', periods=1000))
ts = ts.cumsum()

df = pd.DataFrame(np.random.randn(1000, 4), index=ts.index,
                  columns=['A', 'B', 'C', 'D'])
df = df.cumsum()
df.plot(); plt.legend(loc='best')
```

Feel free to open new cells using the plus button (+), or hitting shift-enter while this cell is selected.

Behind the scenes, the software that powers this is [tmppnb](#), a Tornado application that spawns [pre-built Docker containers](#) and then uses the [jupyter/configurable-http-proxy](#) to put your notebook server on a unique path.

```
from matplotlib import pyplot as plt
import seaborn as sns

ts = pd.Series(np.random.randn(1000), index=pd.date_range('1/1/2000', periods=1000))
ts = ts.cumsum()

df = pd.DataFrame(np.random.randn(1000, 4), index=ts.index,
                  columns=['A', 'B', 'C', 'D'])
df = df.cumsum()
df.plot(); plt.legend(loc='best')
```

Feel free to open new cells using the plus button (+), or hitting shift-enter while this cell is selected.

Behind the scenes, the software that powers this is [tmppnb](#), a Tornado application that spawns [pre-built Docker containers](#) and then uses the [jupyter/configurable-http-proxy](#) to put your notebook server on a unique path.

Jupyter Notebook Example

Run ▶

Paperspace

20 Jay St. Suite 312

Brooklyn, NY 11201

hello@paperspace.com

www.paperspace.com

(718) 619 4325

Thank you.

