Managing AI workloads on cloud

Challenges & Takeaways

Developer Relations Engineer @ Mia-Platform
Former Technical Product Manager, Software Engineer

A bit of myself

Developer Relations Engineer @ Mia-Platform
Former Technical Product Manager, Software Engineer

Graziano Casto ✌🏻

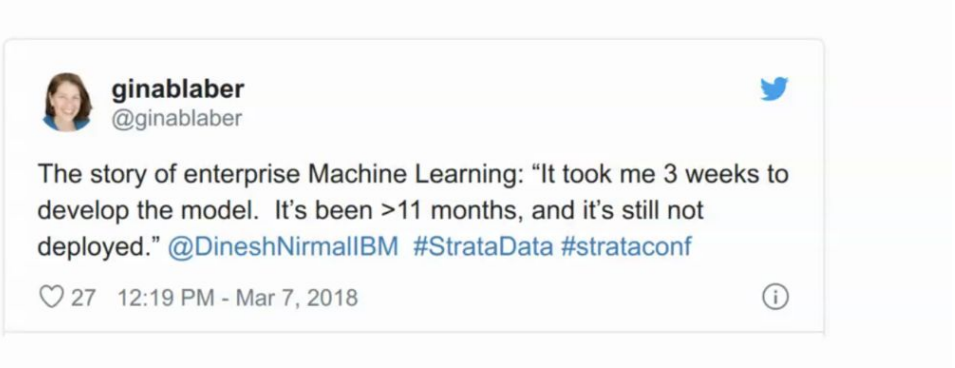
castograziano.com | blog.mia-platform.eu

@graz-dev

This is me while he explains the **misfortunes** of his latest project.

Leonardo is a **Data Scientist** and a friend of mine

Every time I think things are under control, **something breaks**—network issues, cloud costs, or unstable pipelines. **Infrastructure ends up taking more time than model improvements**.



Data Preparation

Model Training and Tuning

Deployment and Monitoring

Iterative Updates

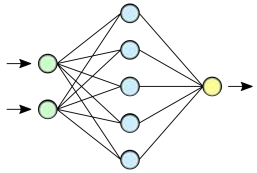
Data management is a mess. Preparing datasets means losing track of who did what and how data changed over time. I'm never sure if it's up to date, risking inaccurate models. Tracing lineage or accountability takes hours to piece together manually.

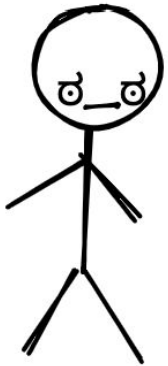
Locally everything runs smoothly, but then when we move to production, it's endless problems: configuration, scalability, integration... it's like we're on a **different planet**!

Model monitoring is chaos! Once in production, we lose control—chasing drift, performance drops, and hidden biases, often noticing issues only when they’re critical.

How can I combine **my experience** with these aspects to help my friend and his team **overcome these challenges**?

I feel like every day I have to **reinvent the wheel** just to get the models to work in the cloud. There's no break at all.

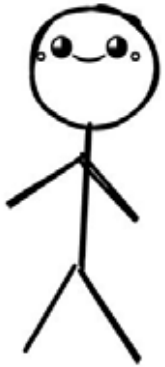




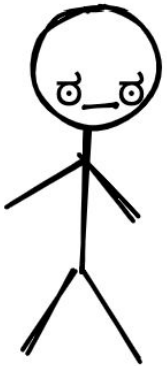
DevOps Guy

Data Science Guy

While thinking at **ML Models**



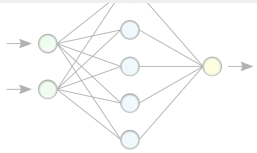
DevOps Guy

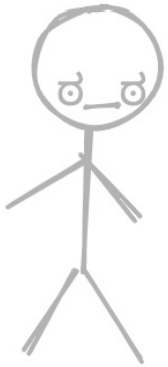


Data Science Guy

While thinking at **Production Deployments**

Doesn't it remind you of anything?

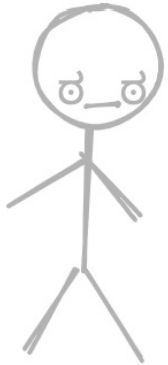




DevOps Guy

Data Science Guy

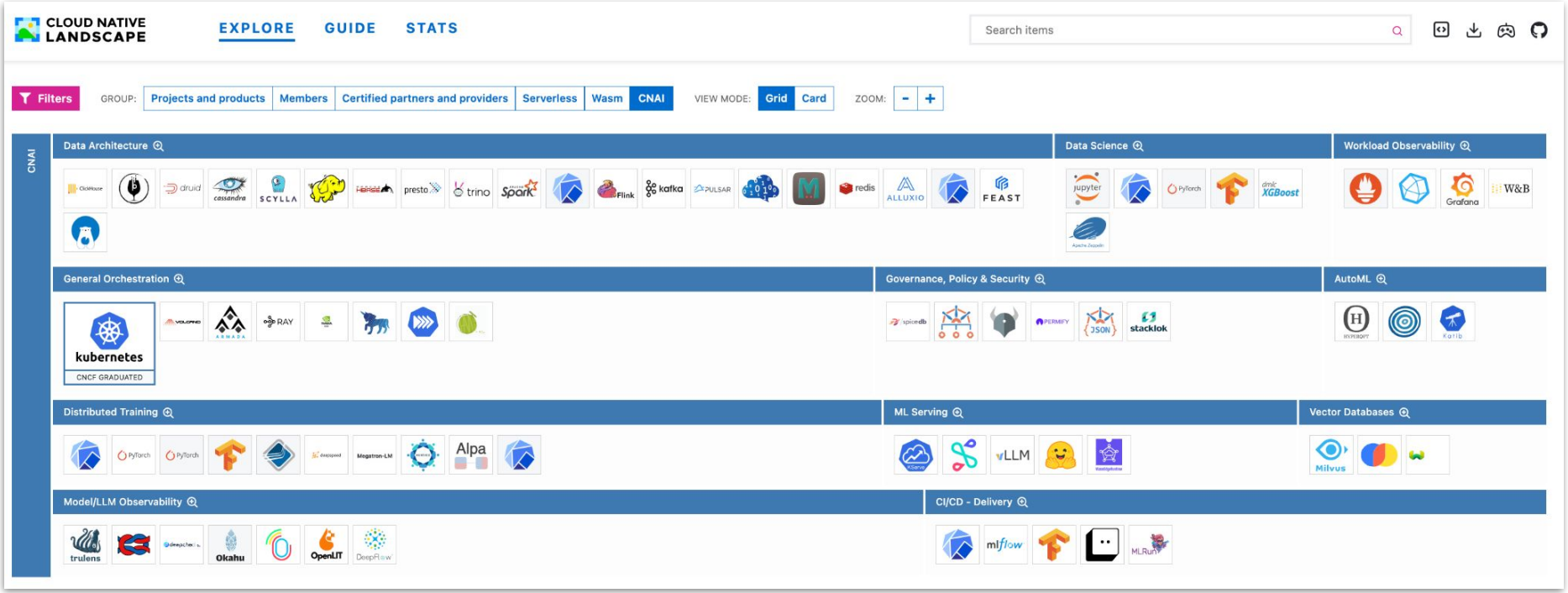
DevOps Guy



Data Science Guy

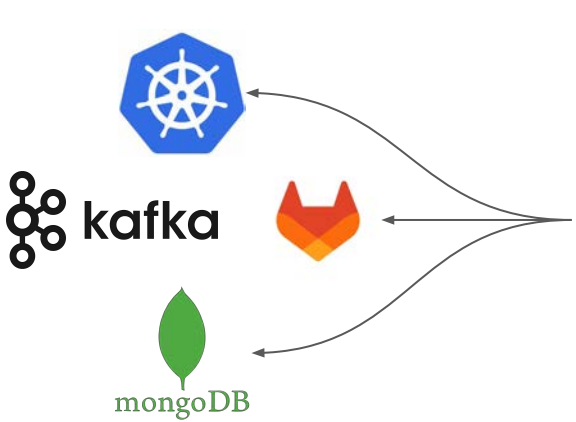
They are the same troubles that existed between Ops and Devs 😱

Leonardo, imagine a world where all of this gets simpler. What if cloud configs, deployments, and scalability weren't daily worries? **If tools and frameworks could automate these headaches**, you'd finally be free to focus on what you love most: the model itself.



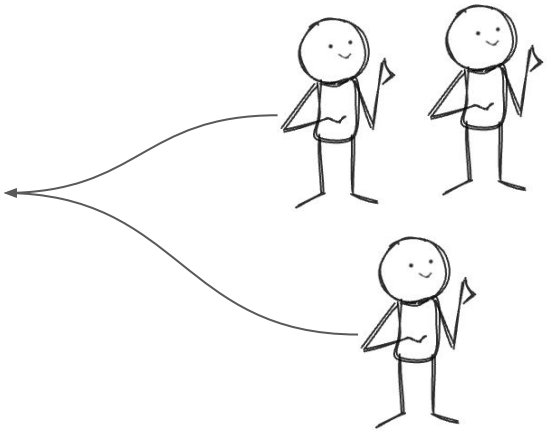
The **CNAI WG** mission is to streamline the integration of artificial intelligence within cloud-native ecosystems, equipping the community with **robust frameworks**, **tools**, and **best practices** that not only ensure scalable, secure, and efficient AI deployments but also simplify bringing **product-ready AI workloads to market**.

Platform Engineering was created precisely to solve these problems. The idea is to build a **self-service platform** that allows ML engineers like Leonardo to focus on models and data, while infrastructure and operations are handled in an **automated and optimized way**.



Dependencies

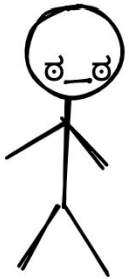
Your IDP



Engineers

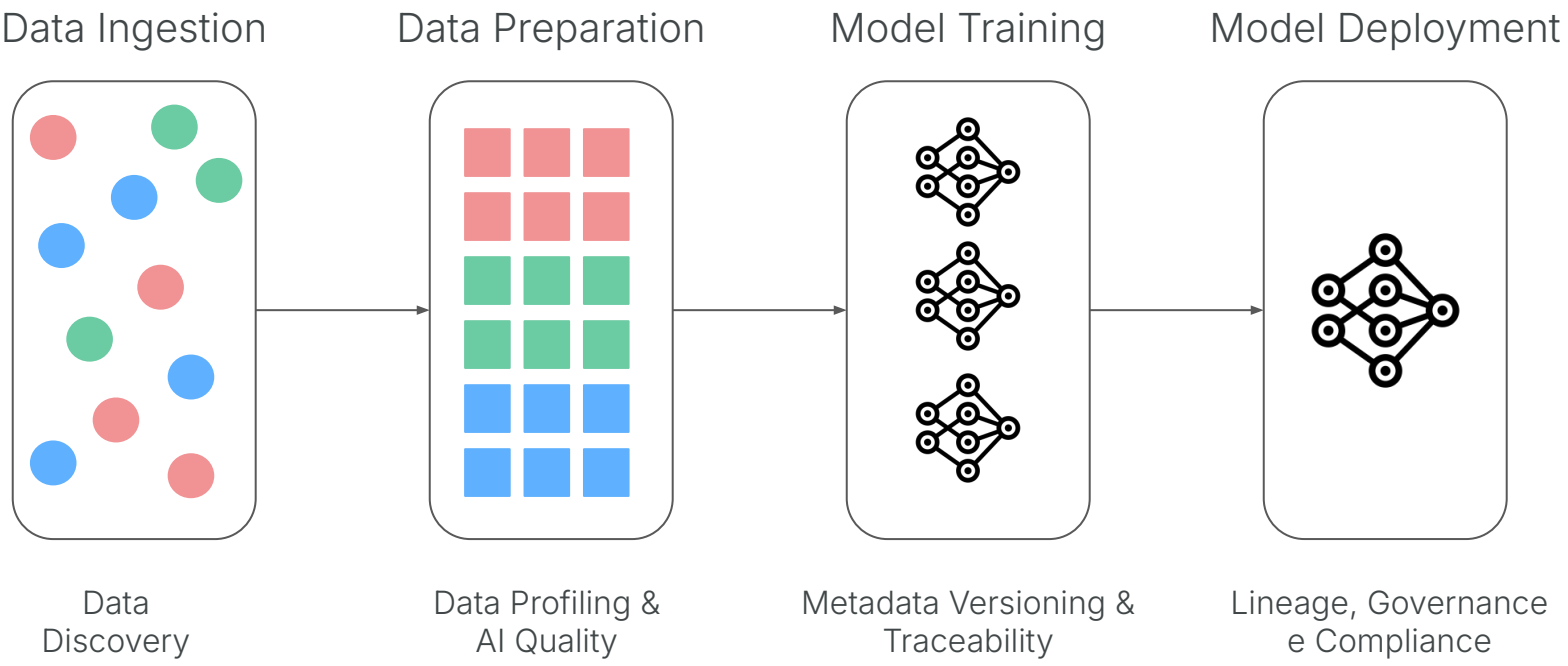
Hey, Leonardo, have you ever thought about introducing a **data catalog** into your workflow? Imagine being able to **browse through datasets like it's Netflix**, without getting lost in all the versions and transformations!

Data Scientist thinking about data preparation without a clear governance

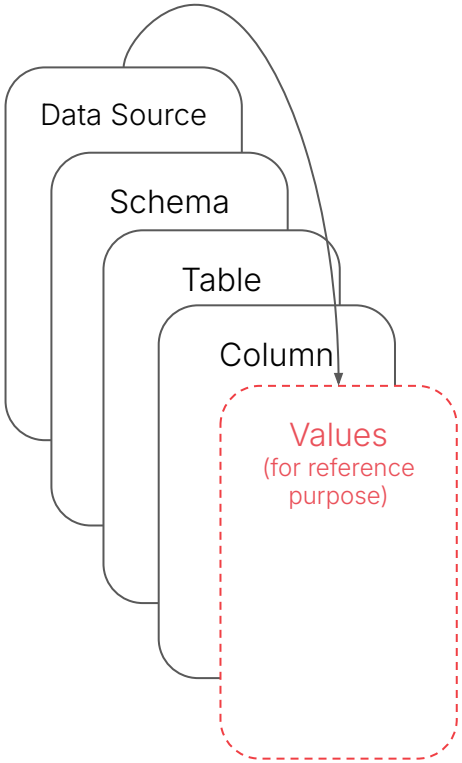


Data Scientist thinking about data preparation using a Data Catalog

The path to AI-readiness



Data Catalogs



Technical Metadata

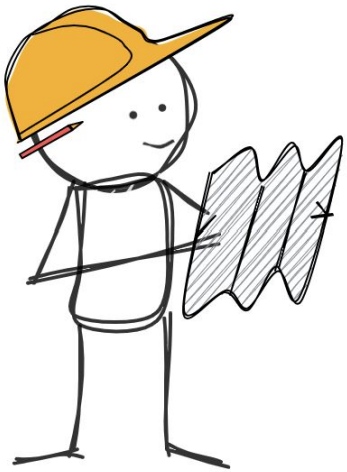
- Name
- Location
- Lineage
- Type

Business Metadata

- Description
- Steward
- Classification
- Custom Fields

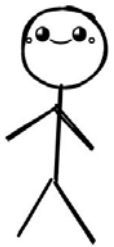
Use Cases

- Search
- Curate
- Audit
- Govern
- Collaborate



Leonardo, why not team up with the platform teams and bring in tools like **Kubeflow** to **bridge local development and deployment?** You could establish shared golden paths across the company to streamline these tools and **make everyone's work easier**.

Kubeflow

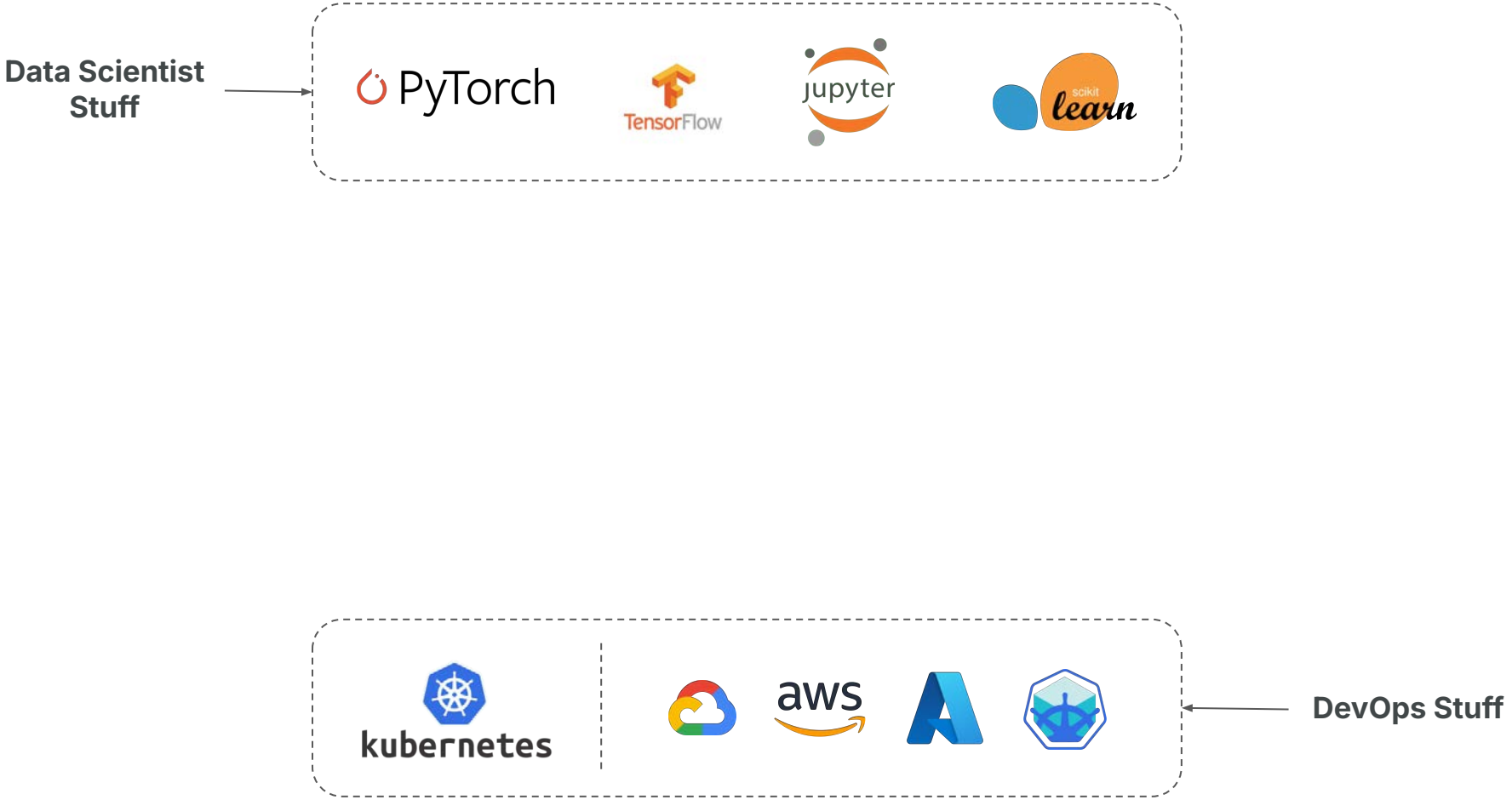


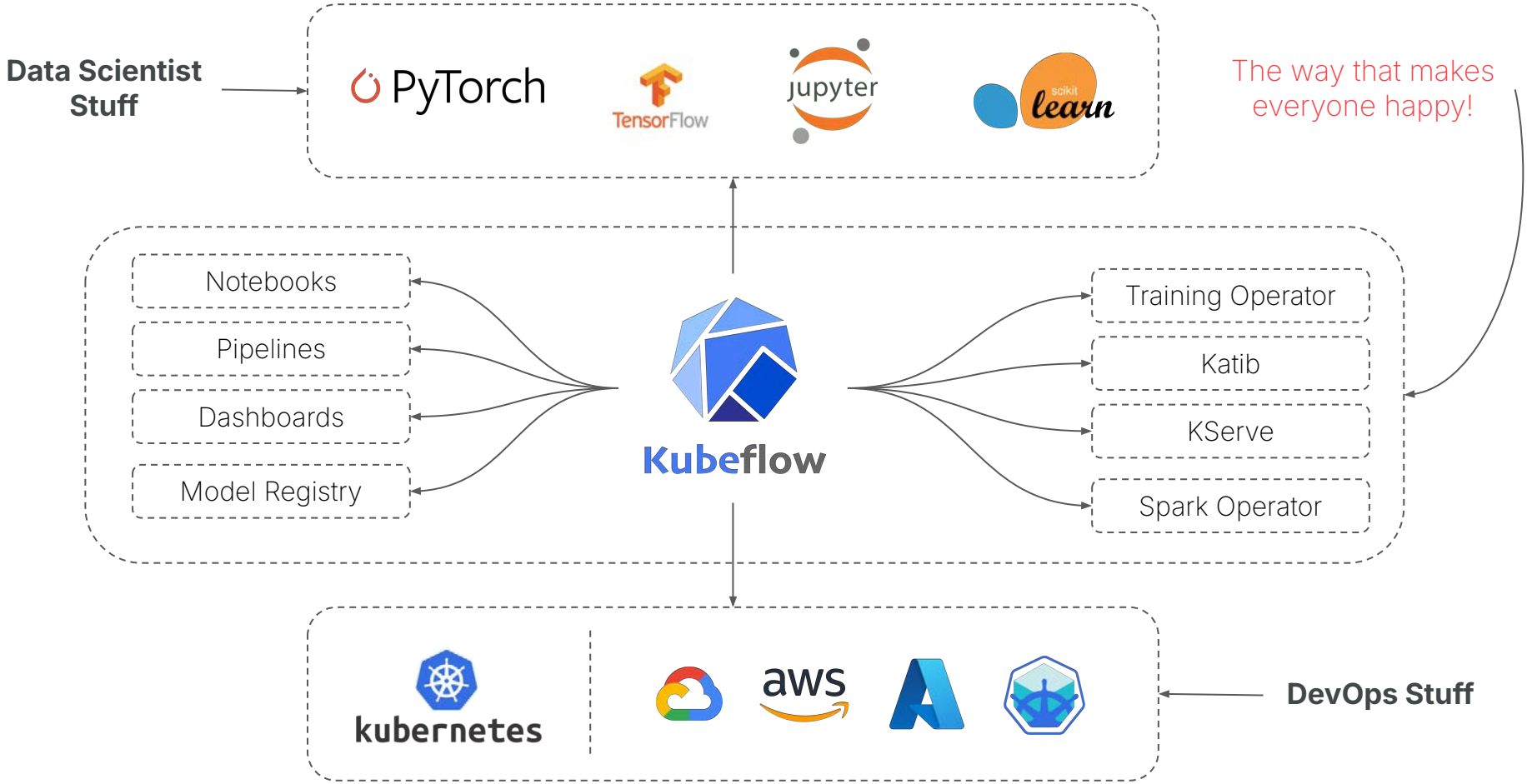
DevOps Guy

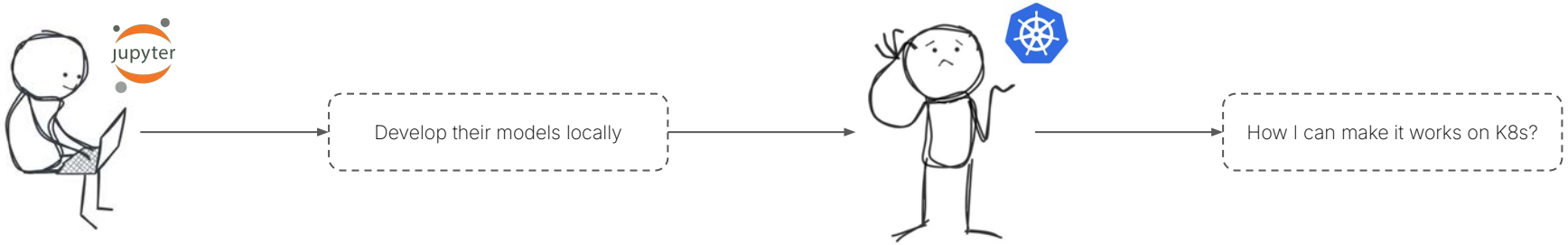
Data Science Guy

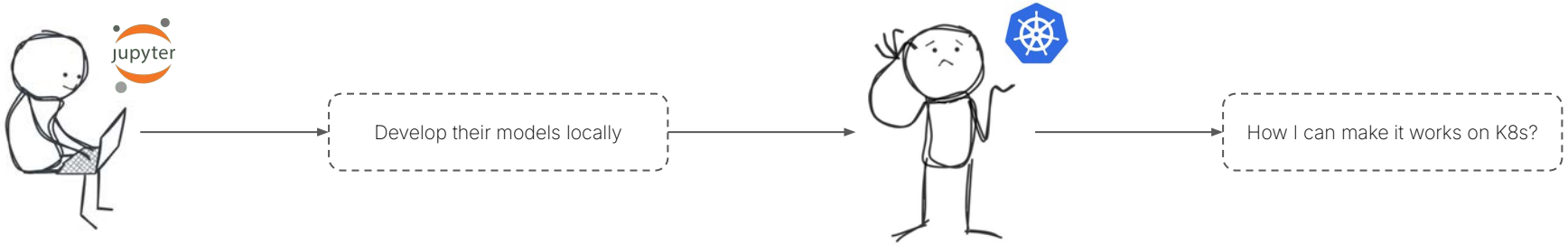
While thinking at their E2E ML Pipeline

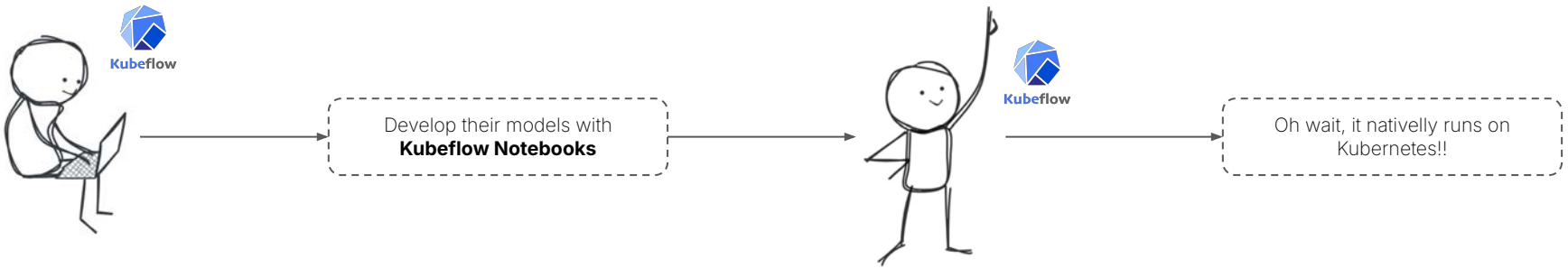

Data Scientist
Stuff
PyTorch
TensorFlow
jupyter
scikit
learn


Data Scientist Stuff
PyTorch
TensorFlow
jupyter
scikit learn
kubernetes
aws
DevOps Stuff

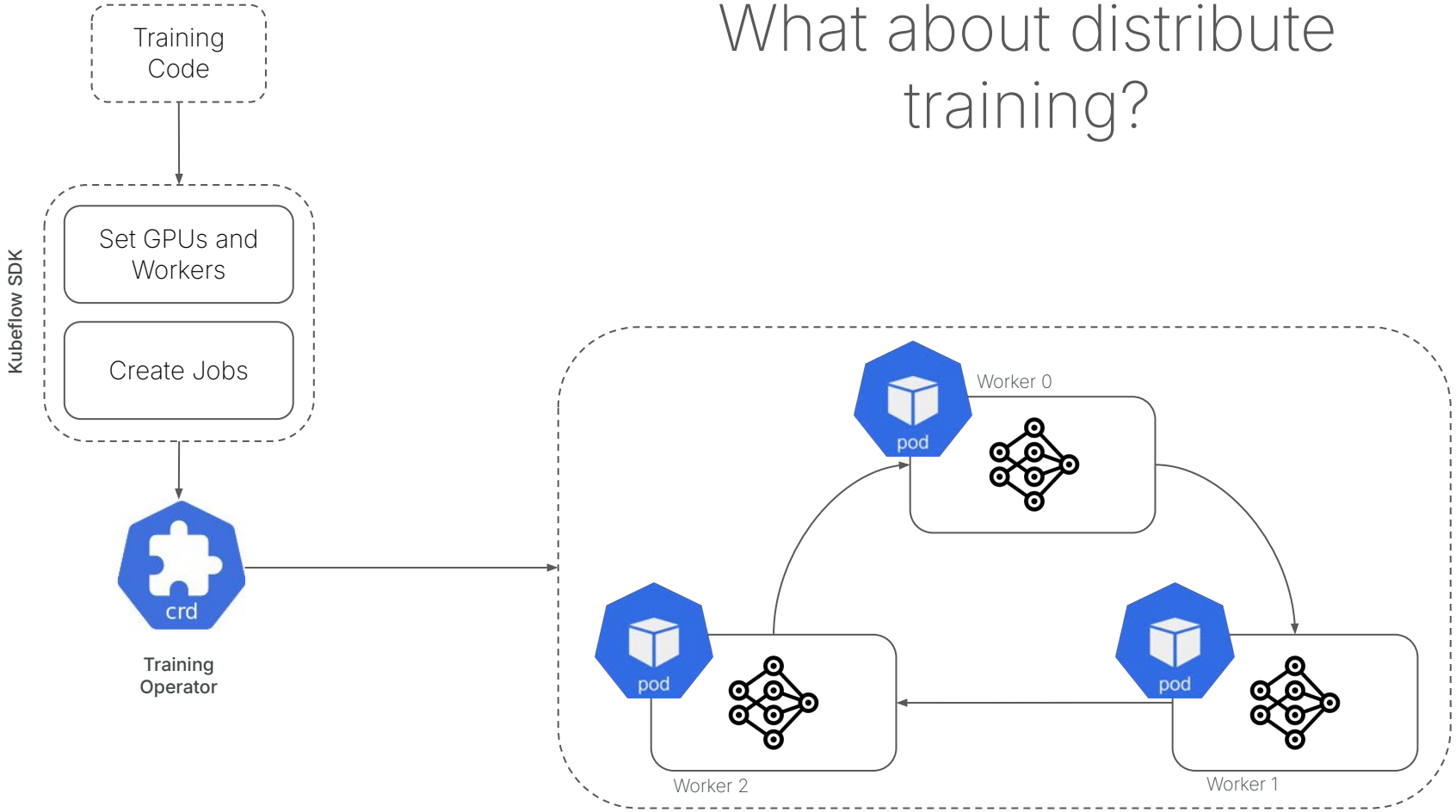

Data Scientist Stuff
PyTorch
TensorFlow
jupyter
scikit learn
The way that makes everyone happy!
Notebooks
Pipelines
Dashboards
Model Registry
Kubeflow
Training Operator
Katib
KServe
Spark Operator
kubernetes
aws
DevOps Stuff


jupyter
Develop their models locally
How I can make it works on K8s?

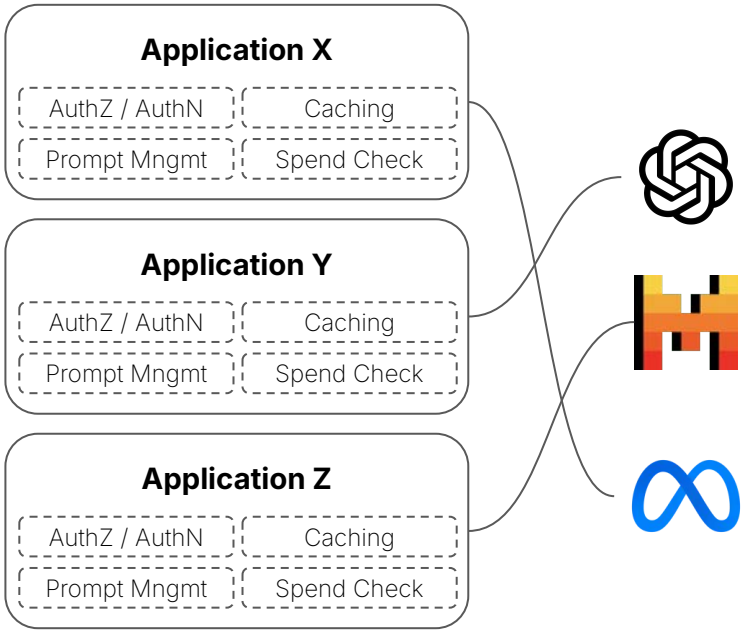

jupyter
Develop their models locally
How I can make it works on K8s?


Kubeflow
Develop their models with
Kubeflow Notebooks
Kubeflow
Oh wait, it natively runs on
Kubernetes!!

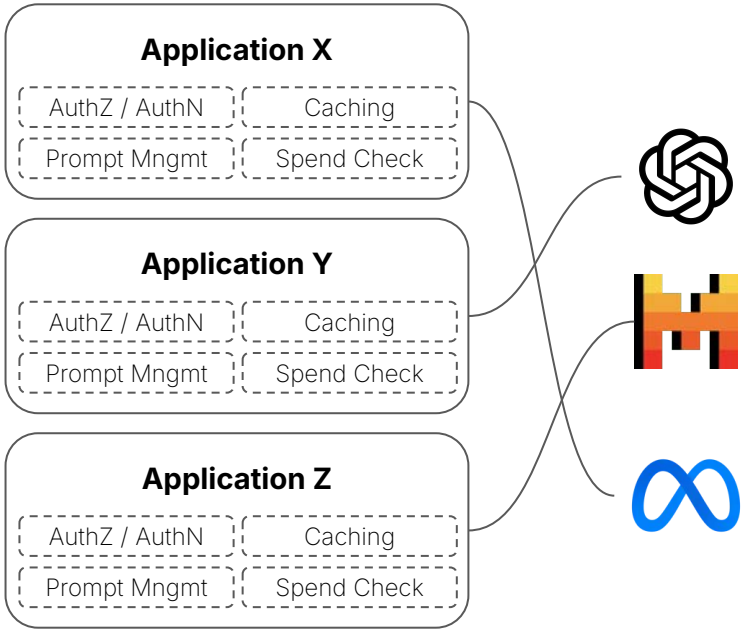
What about distribute training?

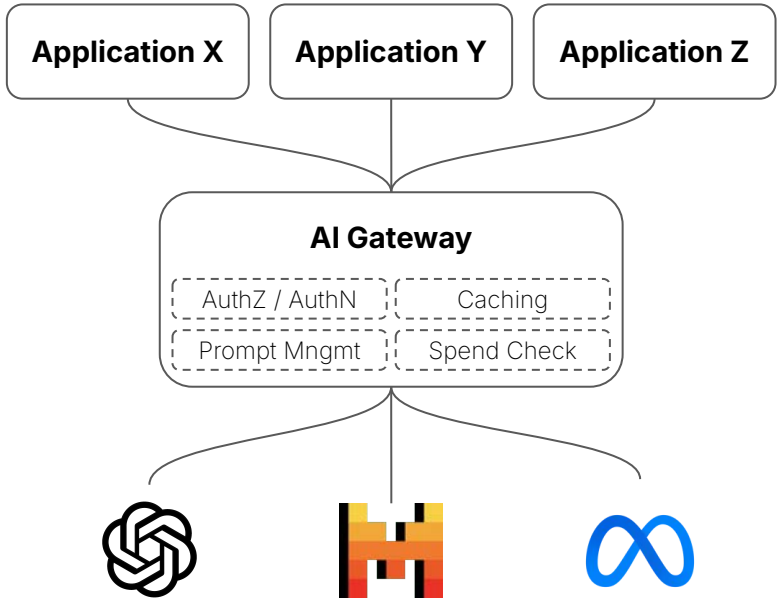


How to integrate third-party models?



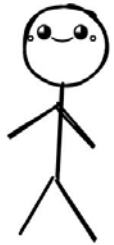
How to integrate third-party models?





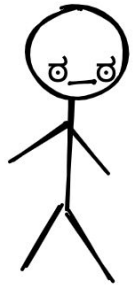
For **observability**, why not integrate the best practices of monitoring and observability from the DevOps world? You can incorporate tools like **Prometheus** and **Grafana**.

We just need system and resource metrics right?



DevOps Guy

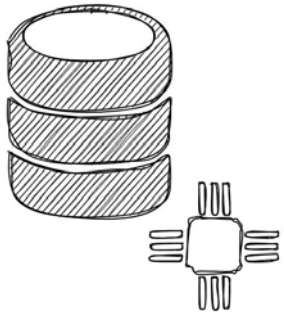
What about model/prediction drifts, classification and regression metrics?



Data Science Guy

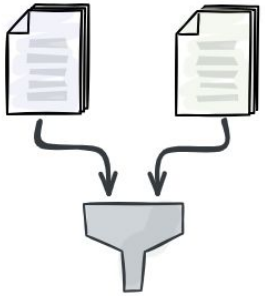
In the realm of MLOps the importance of monitoring starts from **early detection of issues.** This is to ensure the model can explain predictions.

Understanding what to monitor is key.



Operational Metrics
(Is it working?)

Model Outputs
(Are the predictions accurate?)

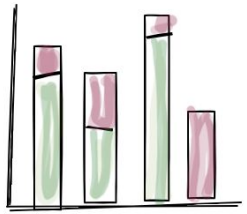


Model Inputs
(Is the data what is expected?)

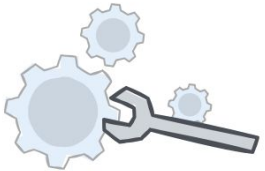
Drifts in ML refer to changes that occur over time.

Understanding and managing different types of drifts is pivotal.

Data Drifts

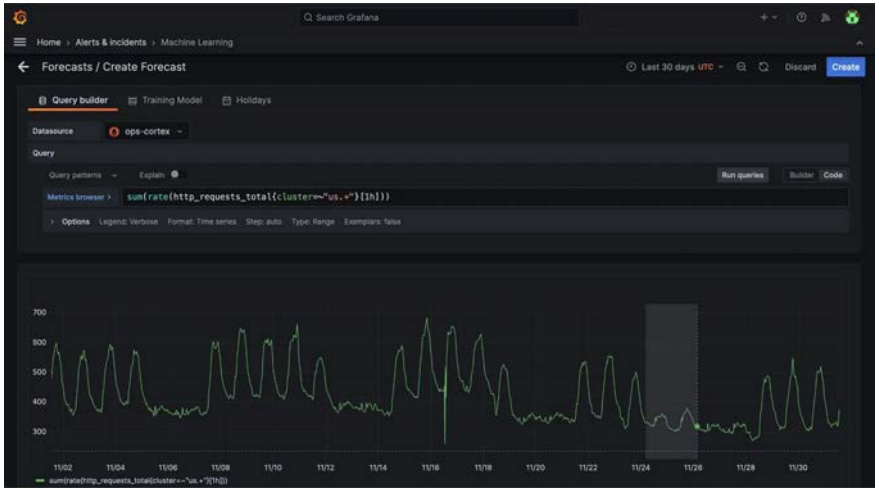


Prediction Drifts



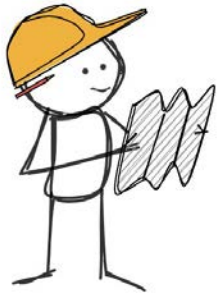
Concept Drifts

Continuous monitoring is the key!



Prometheus + Grafana = ❤
(Also in the realm of MLOPS)

Real-time Monitoring &
Alerting



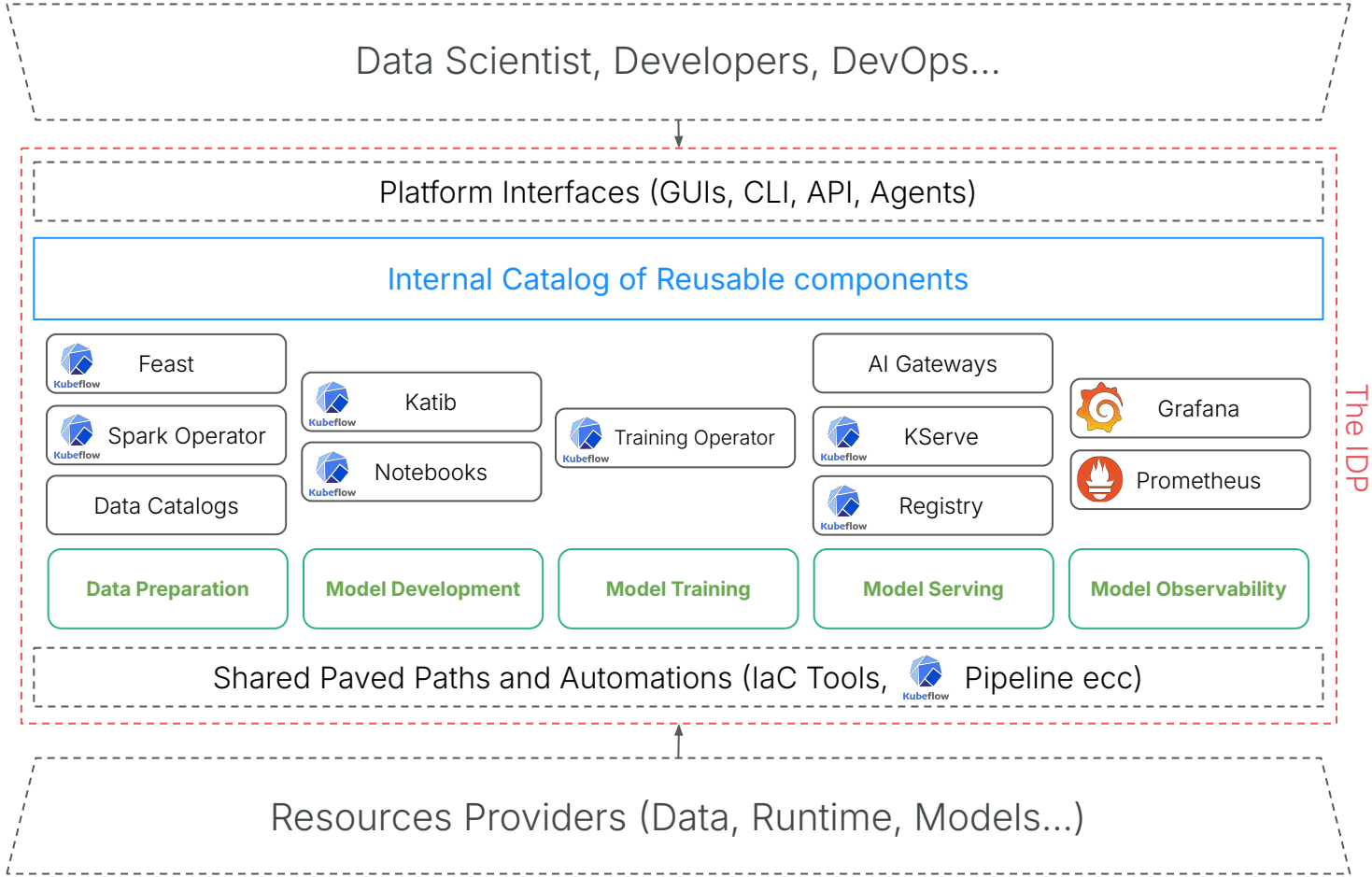
Customizable and
In-Depth Insights

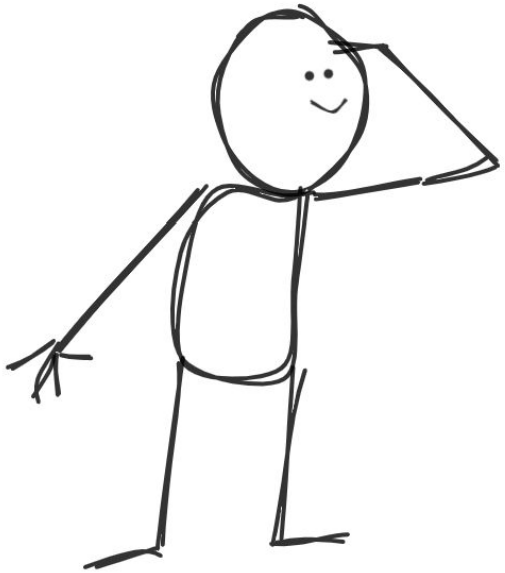
Scalability and
Flexibility

Enhanced Troubleshooting and
Debugging

Wrap Up

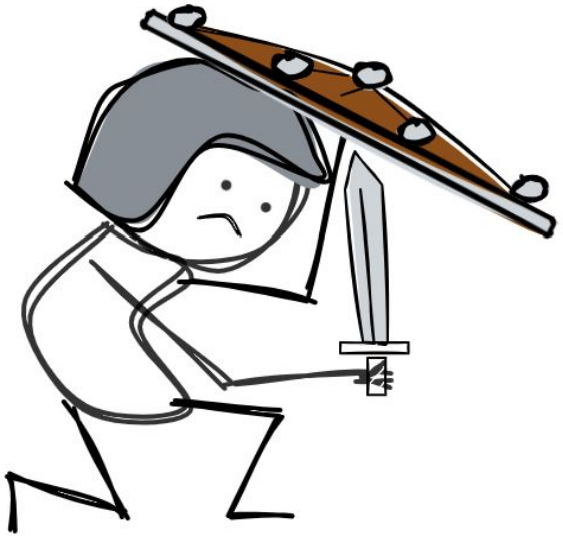
Creates **paved paths** to simplify the adoption of the tools engineers need through the entire lifecycle to make their life easier accessing a **catalog of reusable components**.





The cloud-native world can offer a lot to the AI domain, but what can we take from the AI world to **improve the experience of developing applications in the cloud?**

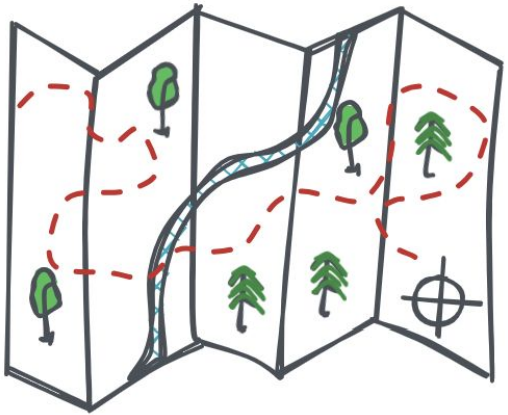
WHAT IF...



We deal with **Platforms** and Platforms are **made of**:

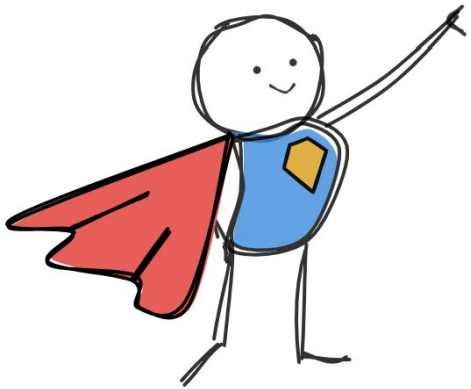
- Metadata
- Schemas
- Pipelines
- Code Repos
- Reusable Components
- Standards
- Policies
- Documentation

And many others...



What if, instead of **fighting against** all these assets...

...we made them **work in synergy for us**?



RAG + Platforms

=

Conversational DevEX

❤

SPOILER: want to build a RAG system over your own documentation in few seconds? Scan the QR Code in the last slide to get the open-source way to **bring your DevEX to the next level!** 🤫

Your Platform

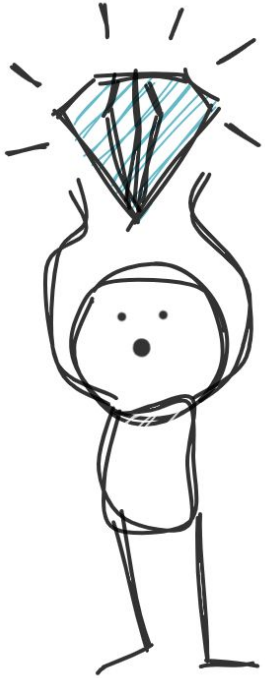
- Metadata
- Schemas
- Pipelines
- Standards
- Reusable Components
- Code Repository
- Policies
- Documentation

Conversational DevEx

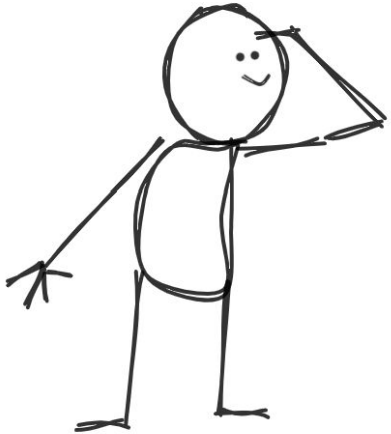
Configurations are provided to AI as comprehensive context of the entire platform.

- Why does my application keep giving me an error?
- Which policies are we applying to this set of data?
- How can I configure a specific service?

Shift old solution to new challenges is often a good choice!



Platforms are a powerful tool when tailored on your needs.



AI on platforms is a game changer that won't steal your job.

https://shorturl.at/mxkNI

Leave a feedback and access
in-depth resources! ☝

Thanks

→ LET’S KEEP IN TOUCH