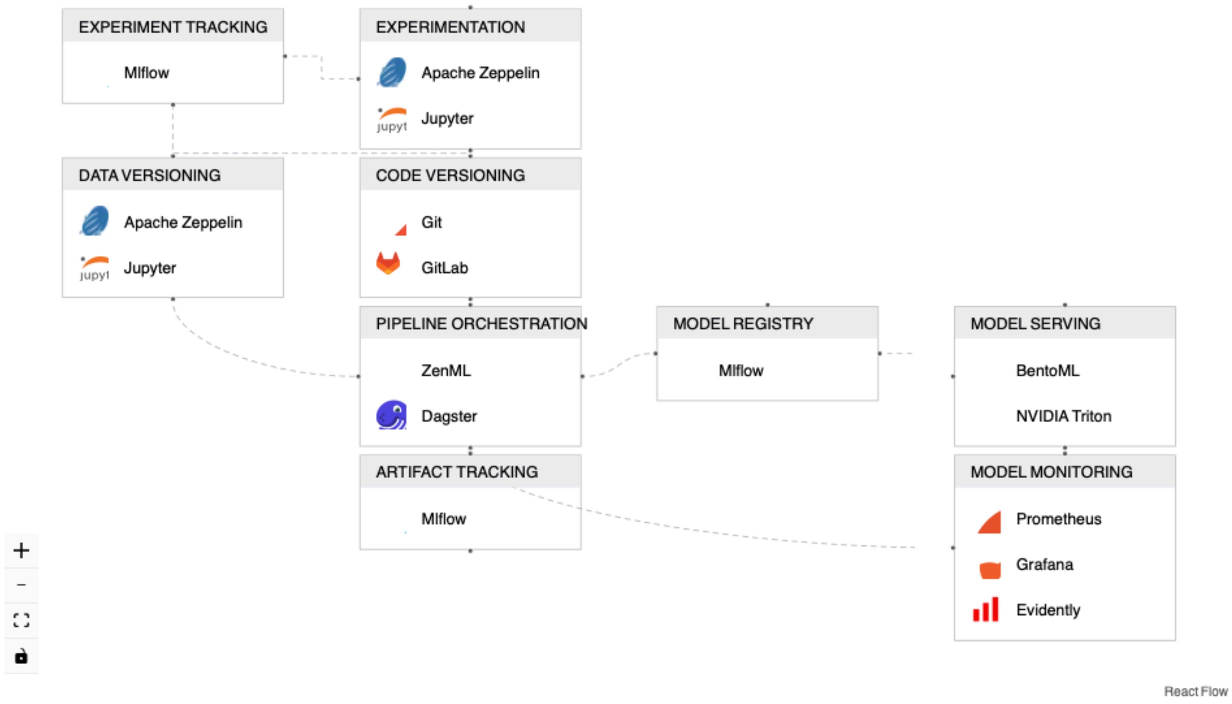
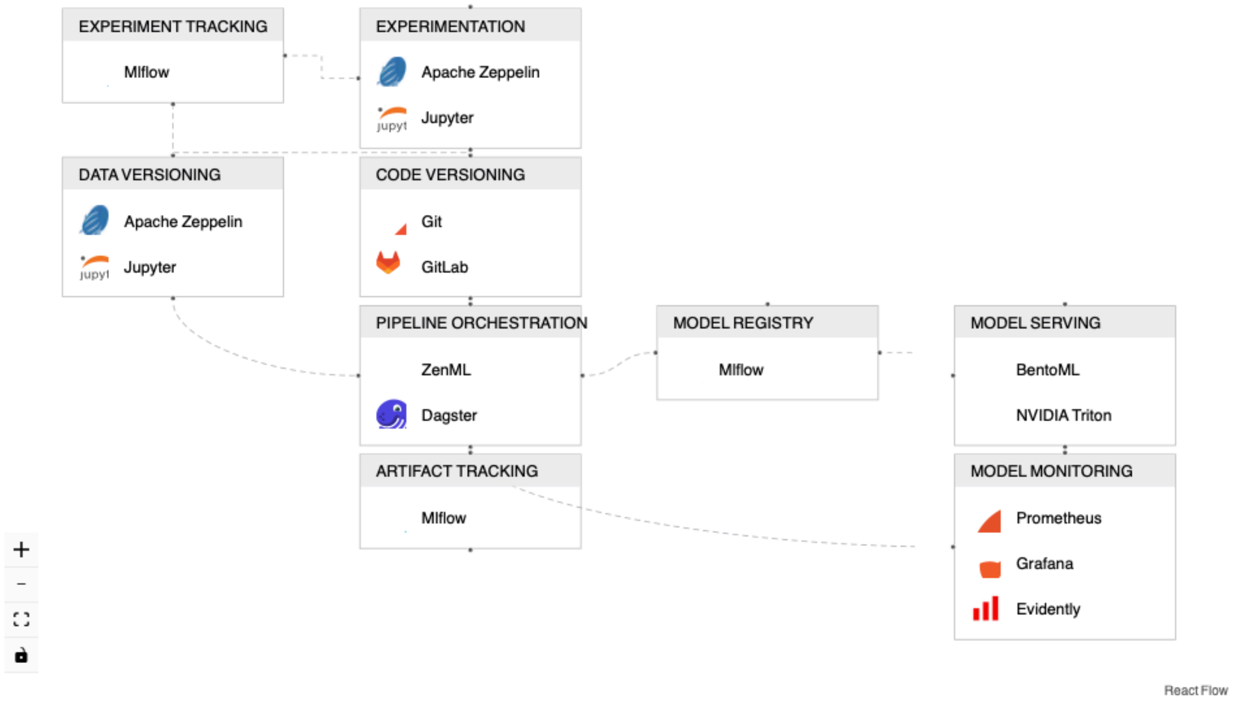
Projet 3a CS - Présentation ML Ops

1. Schéma global d’une stack de ML Ops

https://mllops.vercel.app/



2. Présentation succincte de chaque technologie



DS		(Notebook) / App python : - Data preprocessing - Model definition - Model training - Model testing / evaluation
DE	ML Eng / ML Ops	Model packaging (Bento ML, Kserve, triton) Model registry / versioning (mlflow, bentoml x yatai) Model deployment (Bento ML, triton) Model Serving (Bento ML, triton) Data versioning (DVC) Experiment / Artifact tracking de prod / live (mlflow, W&B, Neptune) -> équipe DS sur le sujet Pipeline orchestration (ré-entrainement, validation, de déploiement) (ZenML x airflow, kubeflow) Model monitoring (prometheus, grafana, loki)
	Runtime engine / pipeline	Kafka, Spark, Flink, Temporal

2 grandes familles de pipeline :

- Training / fine-tuning (exemple : fineweb)
- Inference (exemple : fineweb)

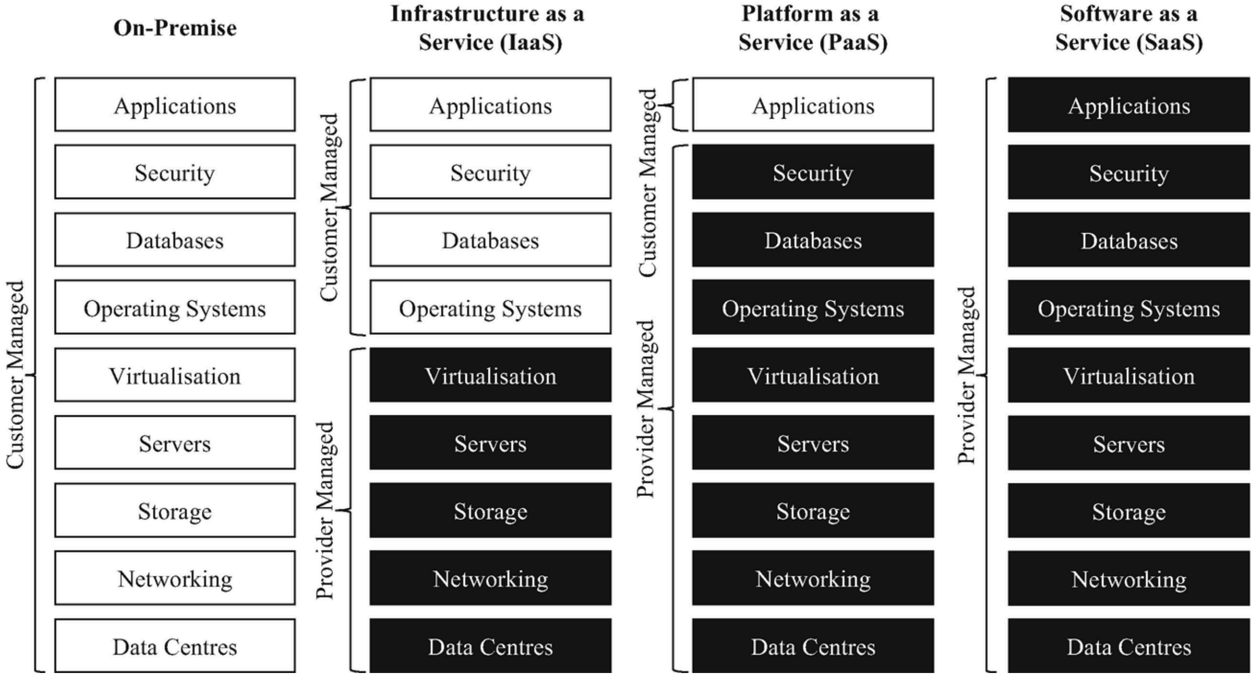
3. Présentation de frameworks de ML Ops (ZenML, Flyte, Ray, ...)

Zen ML
https://www.zenml.io/
https://docs.zenml.io/
https://docs.zenml.io/user-guide/starter-guide
https://docs.zenml.io/how-to/build-pipelines
https://docs.zenml.io/stack-components/component-guide

Flyte
https://flyte.org/

Ray
https://ray.io

4. Présentation des solutions Cloud (GCP Vertex AI, AWS Sagemaker, Azure ML)



Source wikipedia

- Ready to use task-specific solutions (almost SaaS)
  - Vision
  - LLM
  - Document parsing
  - Etc…
- Ready to deploy models
  - Azure model catalog
  - GCP model garden
  - AWS Sagemaker Marketplace
- MLOps bricks
  - Data catalog
  - Pre-built environment registries (VM, training, inference)
  - Training
  - Inference RT / batch
  - Monitoring
  - Model registry