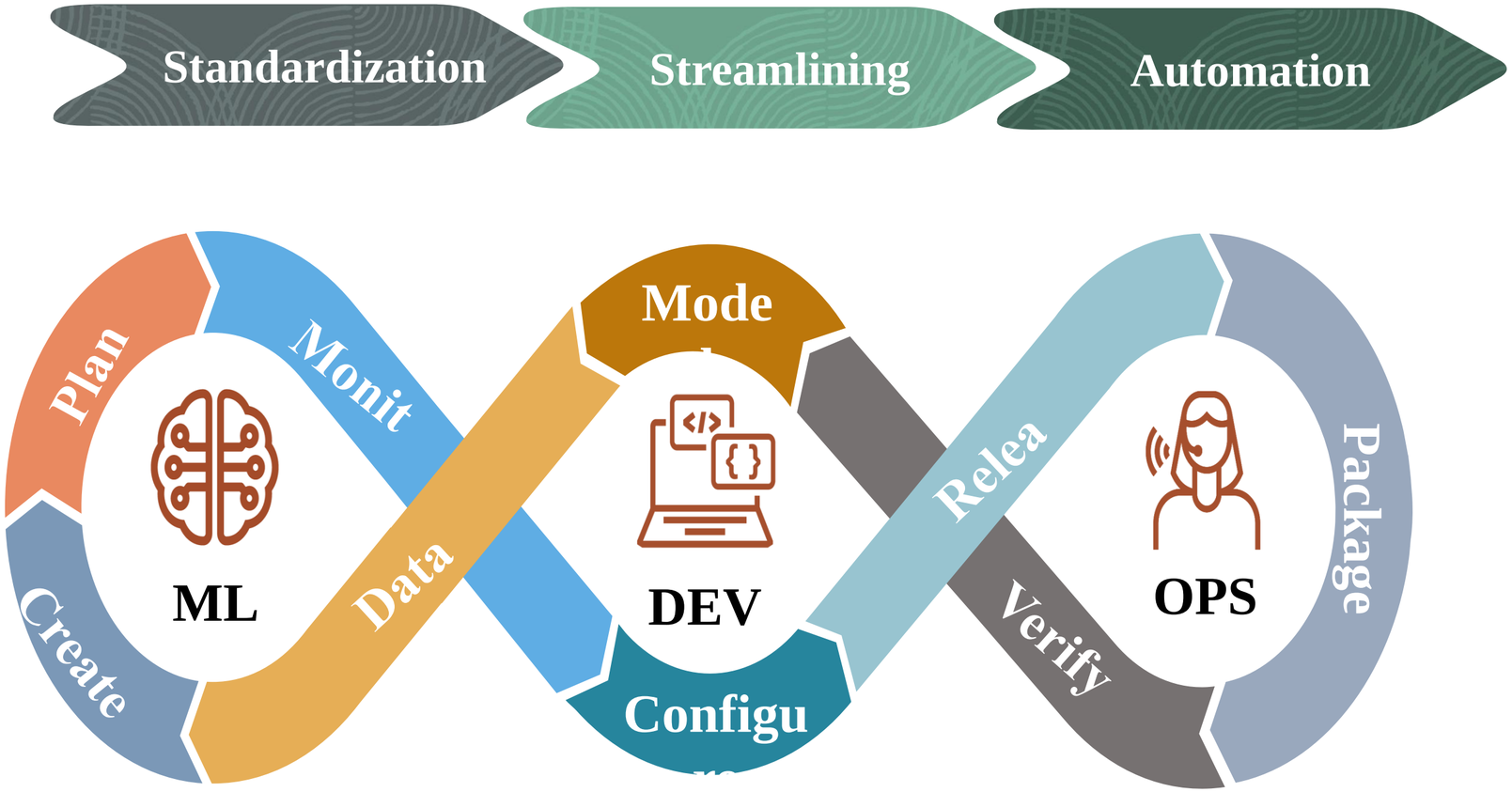
MLOps Practices

MLOps Architecture

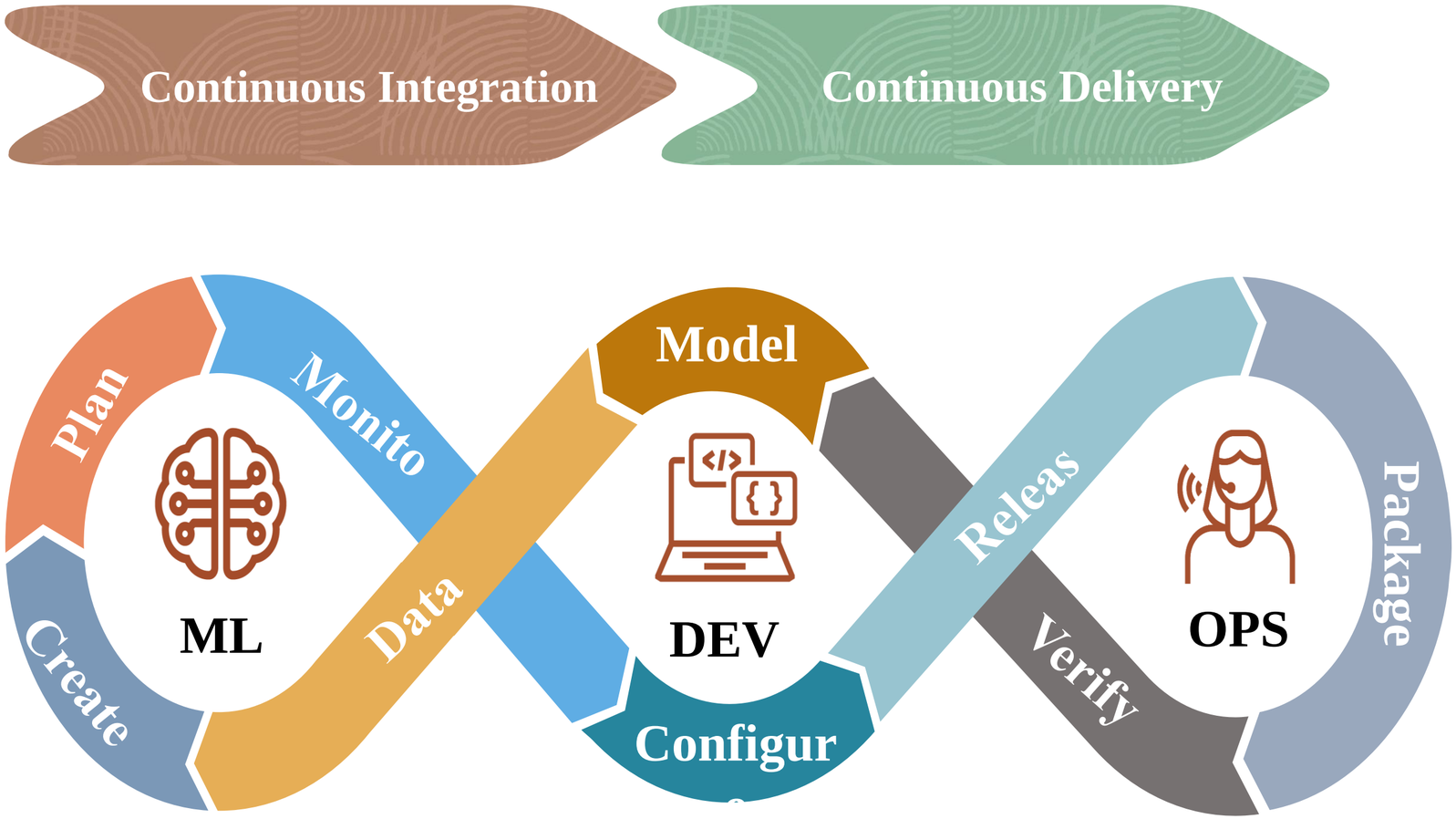
MLOps is the standardization, streamlining, and automation of machine learning lifecycle management. ML assets are treated like other software assets within an iterative, continuous-integration, continuous-delivery environment. ML models are deployed alongside the services that wrap them and the services that consume them as part of a unified release process. Ultimately, MLOps is a set of technologies and practices to rapidly deploy and manage scalable and governed ML applications in production environments.

What is MLOps?



MLOps is the standardization, streamlining, and automation of machine learning lifecycle management. ML assets are treated like other software assets within an iterative, continuous-integration, continuous-delivery environment. ML models are deployed alongside the services that wrap them and the services that consume them as part of a unified release process. Ultimately, MLOps is a set of technologies and practices to rapidly deploy and manage scalable and governed ML applications in production environments.

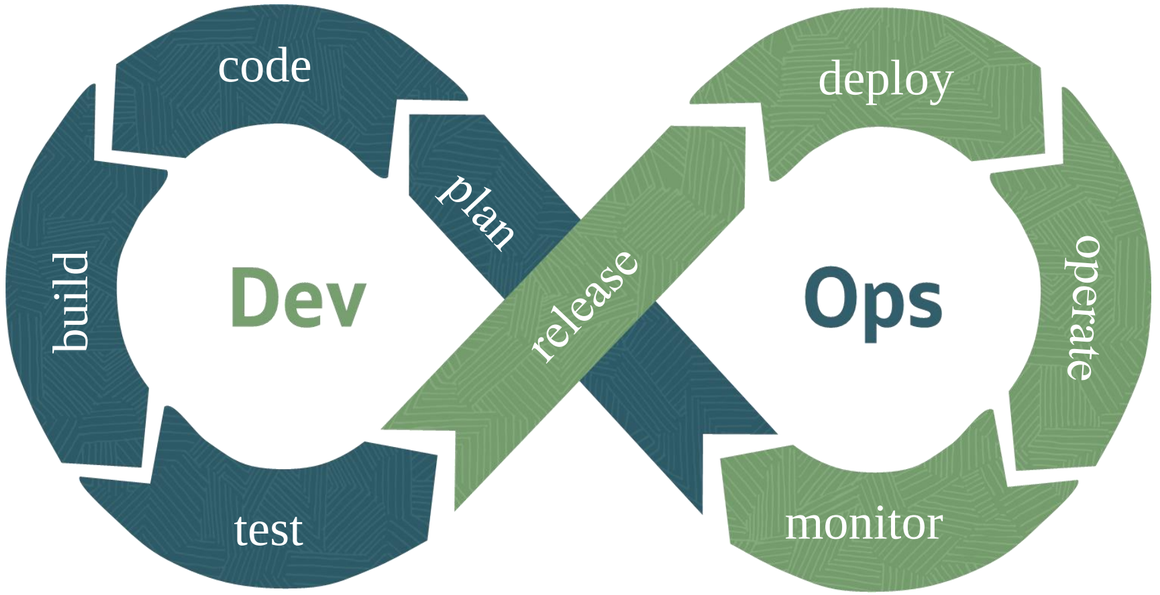
What is MLOps?



What is MLOps?

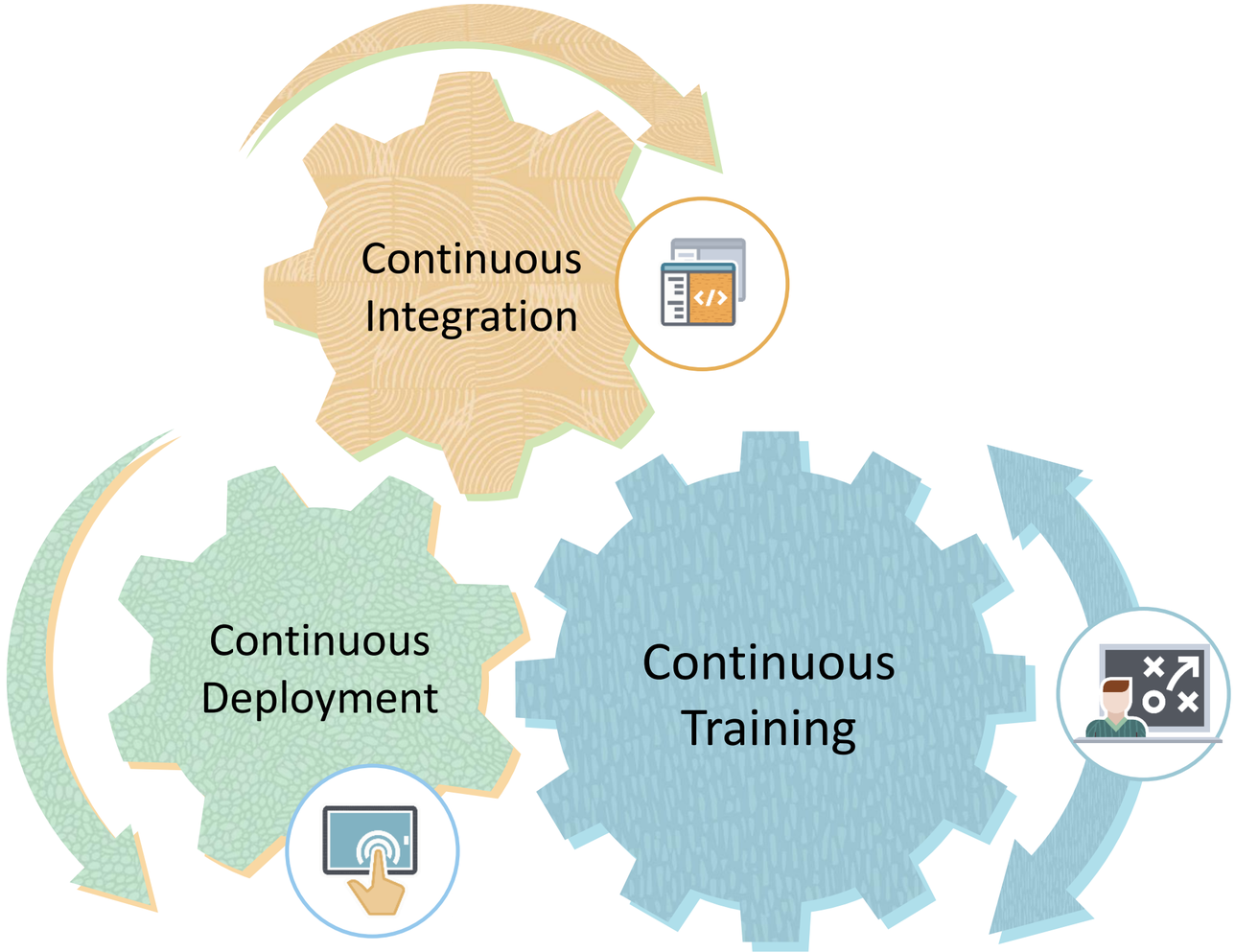
MLOps pulls heavily from the concepts of DevOps, which streamlines the practice of software updates. You might be familiar with the DevOps workflow diagram, which is similar to the MLOps diagram. The DevOps life cycle includes practices like building, testing, continuous deployment, and monitoring.

The MLOps life cycle adds the ML loop to the DevOps lifecycle below

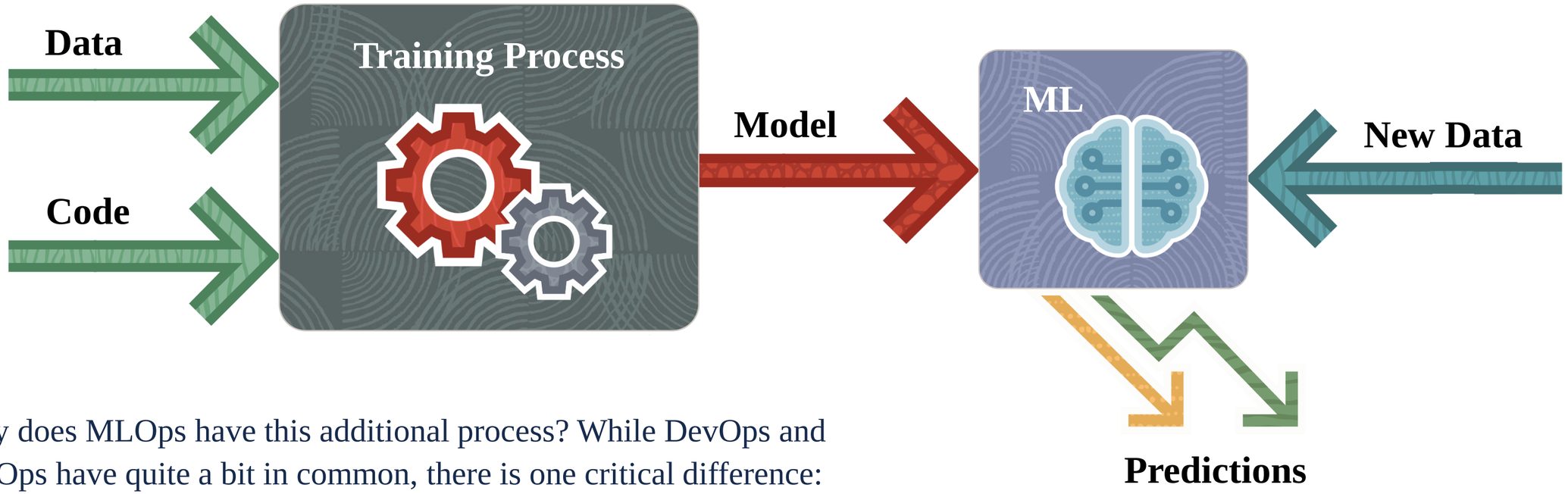


Continuous Practices in MLOps

In DevOps, continuous integration refers to the validation and integration of updated code into the central repository, and continuous deployment refers to the redeployment of those changes into production.
In MLOps, continuous integration refers to the validation and integration of new data and ML models, and continuous deployment refers to releasing that model into production.
Continuous training is unique to MLOps and refers to the automatic retraining of ML models for redeployment.

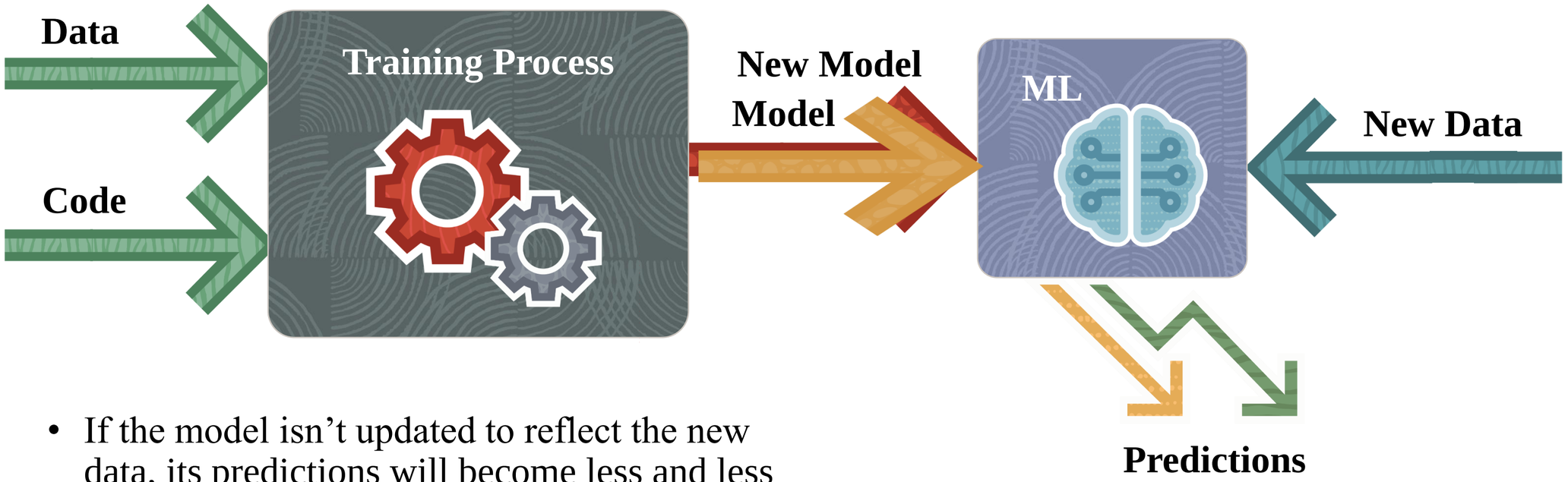


Why is MLOps important?



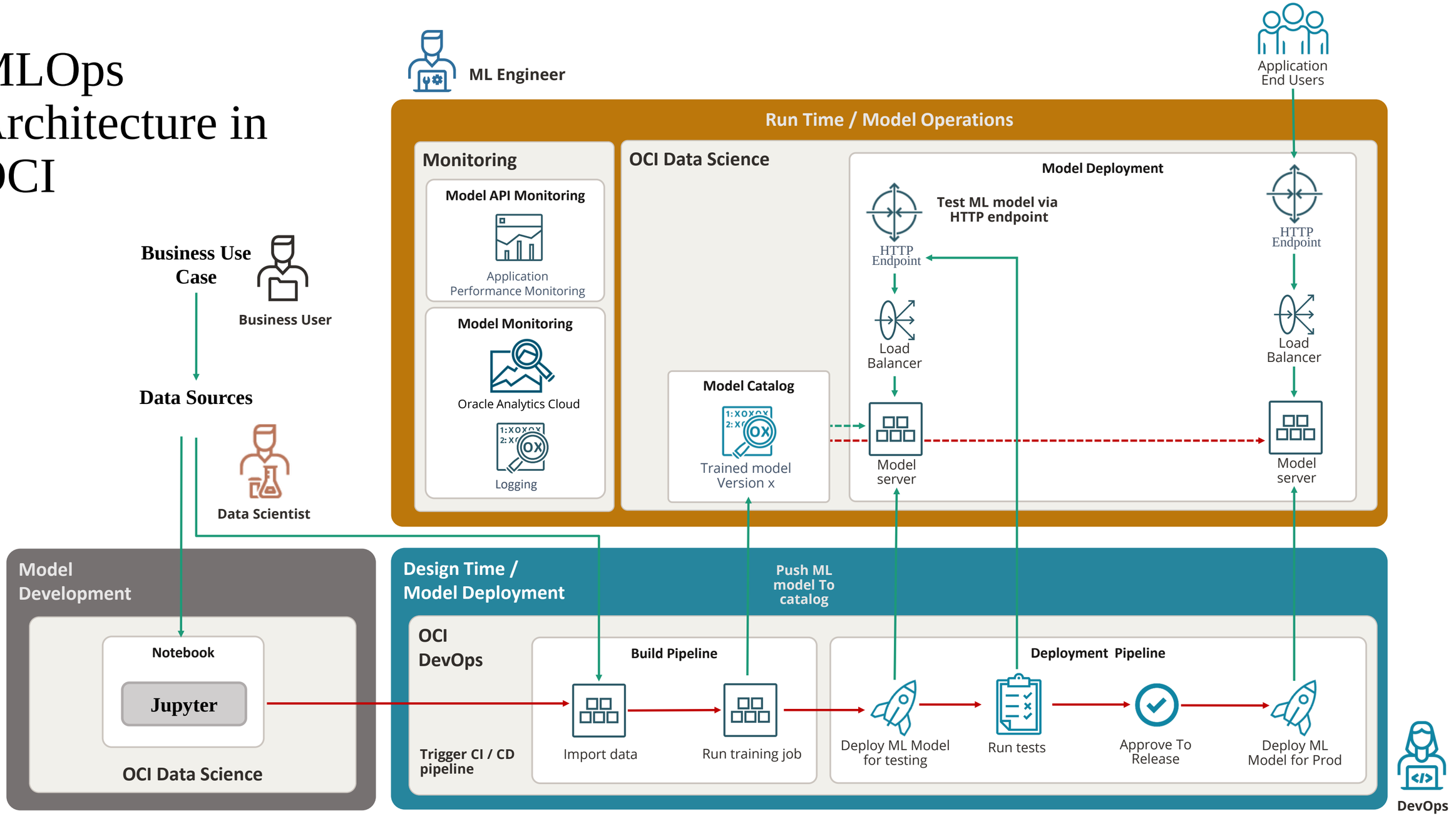
Why does MLOps have this additional process? While DevOps and MLOps have quite a bit in common, there is one critical difference: whereas software code is relatively static, data is always changing. That means machine learning models are constantly learning and adapting to new inputs. This drift in data is why the continuous retraining of ML models is so critical.

Why is MLOps important?



- If the model isn’t updated to reflect the new data, its predictions will become less and less accurate. Due to this degradation in the model’s performance over time, it is important to retrain the model on new data as quickly as possible. This is where the automation of continuous training in MLOps provides great benefits.

MLOps Architecture in OCI



This is the MLOps architecture, which shows the life cycle from data ingestion to deploying the model in an application to end users.

1. Business information from business and data scientists is sent into the Jupyter notebooks for model development.
2. The raw data and notebook sessions are imported into OCI DevOps, which triggers the CI/CD pipeline.
3. The DevOps Build Pipeline runs the training job and sends the output to the model catalog.
4. The completion of the Build Pipeline triggers the Deployment Pipeline.
5. The ML model is deployed for testing.
6. The model is tested at an internal endpoint.
7. If the ML model is approved, it is deployed for production.
8. Finally, the model is sent to applications for end users.

- **SOURCES**
- https://k21academy.com/microsoft-azure/dp-100/machine-learning-operations-mlops-on-azure/
- https://www.youtube.com/watch?v=zSLQaPIkVu8&ab_channel=ParshwaShah
- https://neptune.ai/blog/mlops
- https://databricks.com/glossary/mlops
- https://ml-ops.org/
- https://neptune.ai/blog/mlops