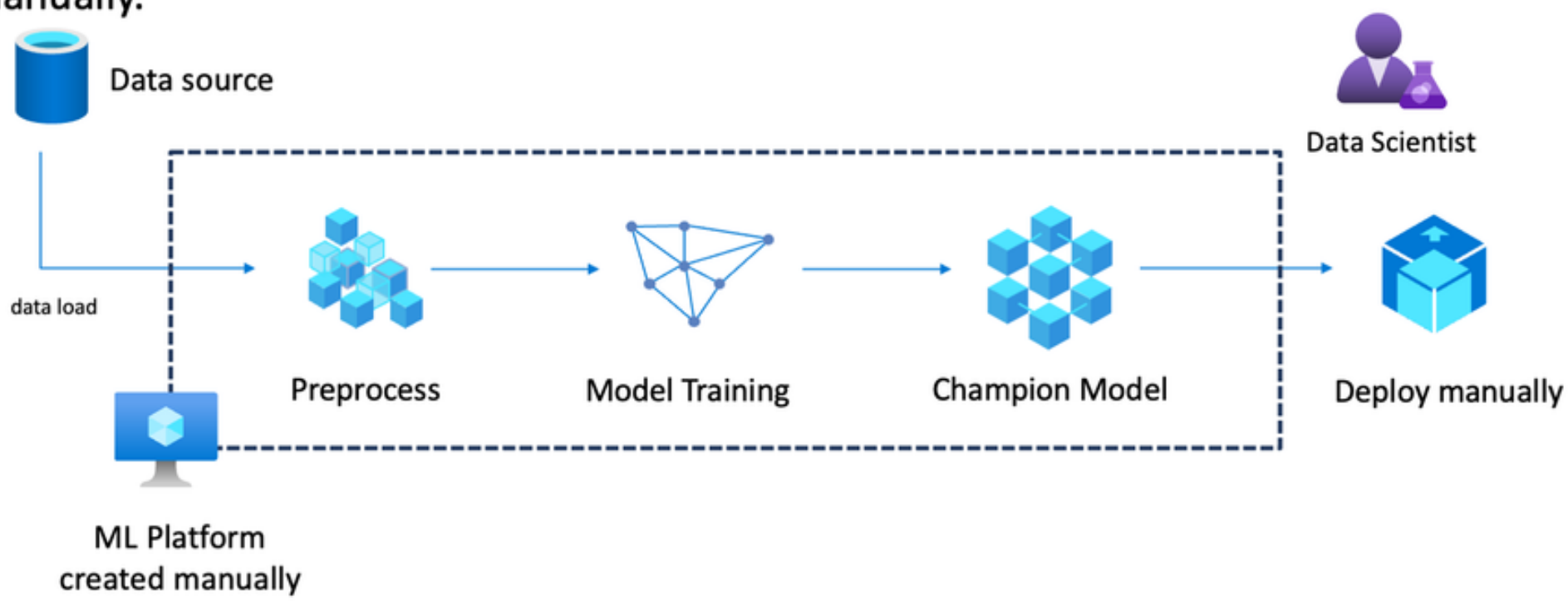
Microsoft

MACHINE LEARNING OPERATIONS MATURITY MODEL

5 Stages Architecture View

Level 0 – No MLOps

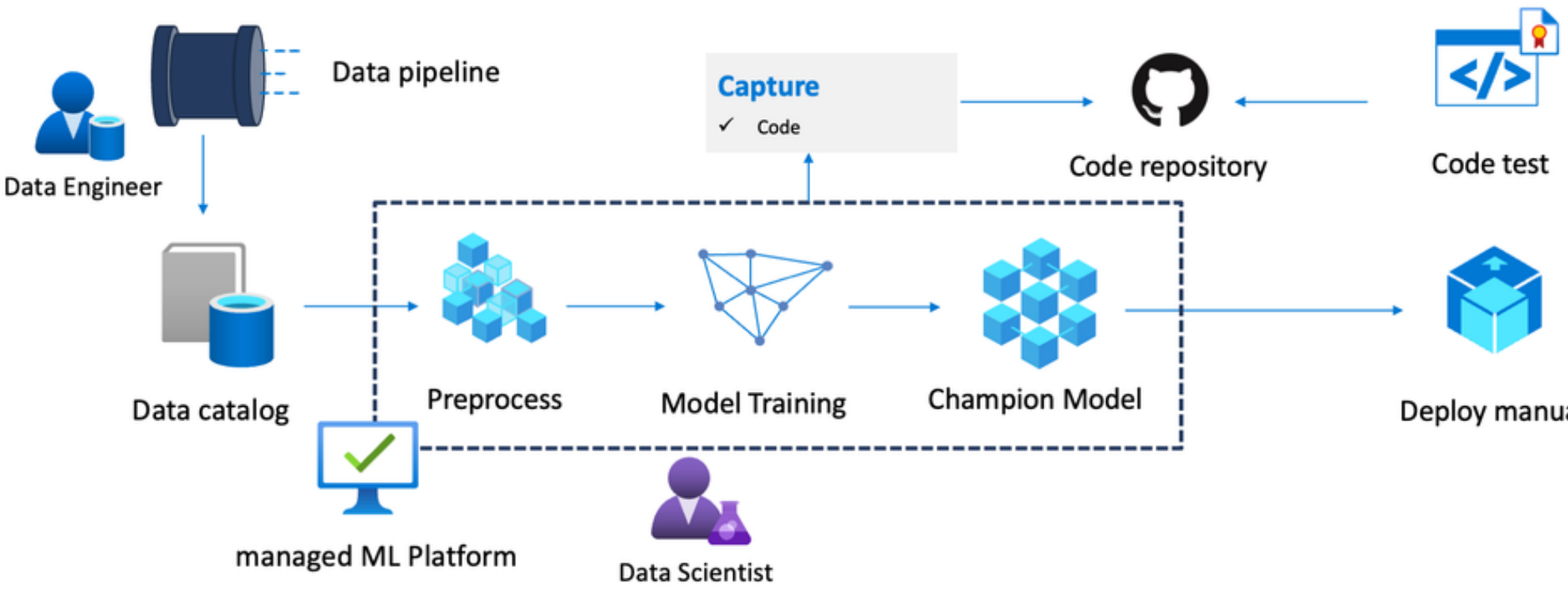
- Find best model interactively and exploratory.
- Create environment, gather and preprocess data, model training, deploy and test manually.



People	Model Creation
• Data scientists: siloed, not in regular communications with the larger team • Data engineers (*if exists*:(siloed, not in regular communications with the larger team • Software engineers: siloed, receive model remotely from the other team members	• Data gathered manually • Compute is likely not managed • Experiments aren't predictably tracked • End result may be a single model file manually handed off with inputs/outputs
Model Release	**Application Integration**
• Manual process • Scoring script may be manually created well after experiments, not version controlled • Release handled by data scientist or data engineer alone	• Heavily reliant on data scientist expertise to implement • Manual releases each time

Level 1 – DevOps no MLOps

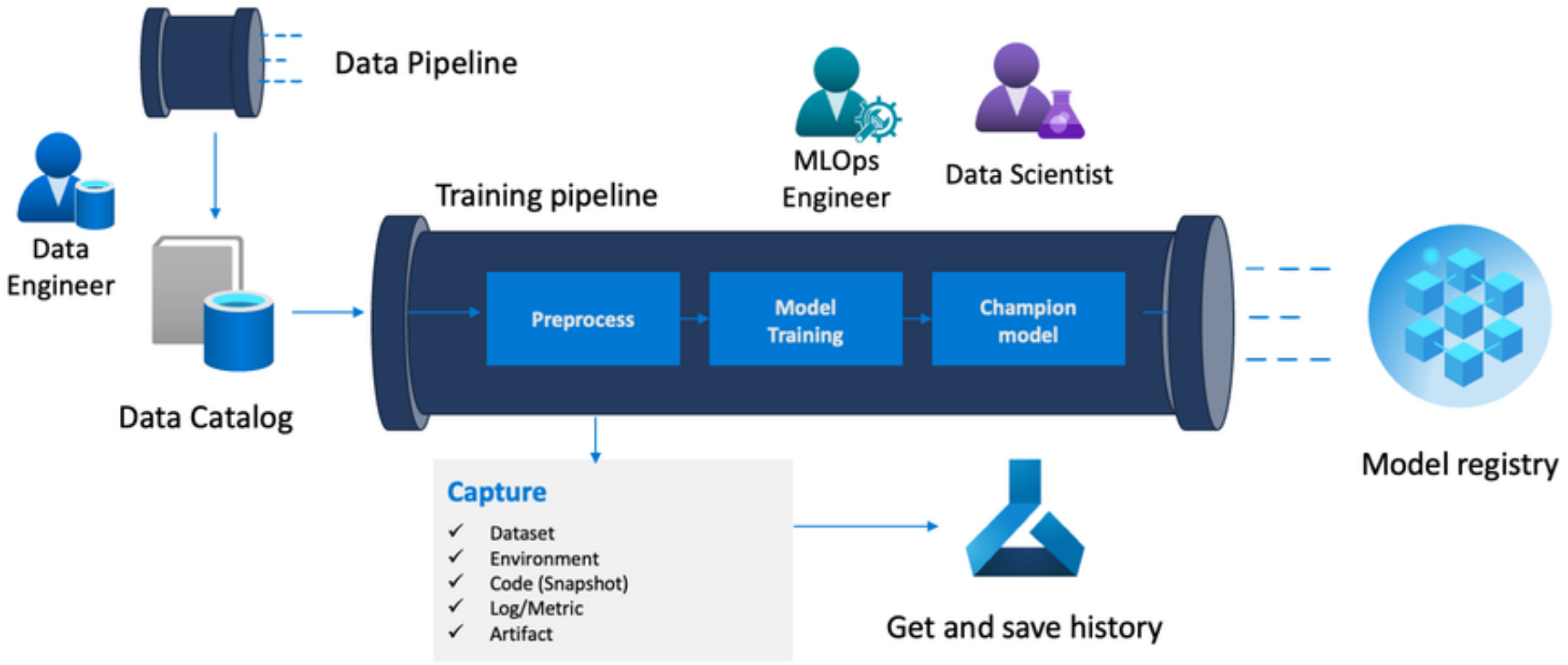
- Create managed ML platform.
- Maintain code test against application and training/inference scripts.



People	Model Creation
• Data scientists: siloed, not in regular communications with the larger team • Data engineers (if exists): siloed, not in regular communication with the larger team • Software engineers: siloed, receive model remotely from the other team members	• Data pipeline gathers data automatically • Compute is or isn't managed • Experiments aren't predictably tracked • End result may be a single model file manually handed off with inputs/outputs
Model Release	**Application Integration**
• Manual process • Scoring script may be manually created well after experiments, likely version controlled • Is handed off to software engineers	• Basic integration tests exist for the model • Heavily reliant on data scientist expertise to implement model • Releases automated • Application code has unit tests

Level 2 – Automated Training

- Make Code, Data, Model tracked, saved and version controlled.
- Automate training process using pipeline.



People

- Data scientists: Working directly with data engineers to convert experimentation code into repeatable scripts/jobs
- Data engineers: Working with data scientists
- Software engineers: siloed, receive model remotely from the other team members

Model Creation

- Data pipeline gathers data automatically
- Compute managed
- Experiment results tracked
- Both training code and resulting models are version controlled

Model Release

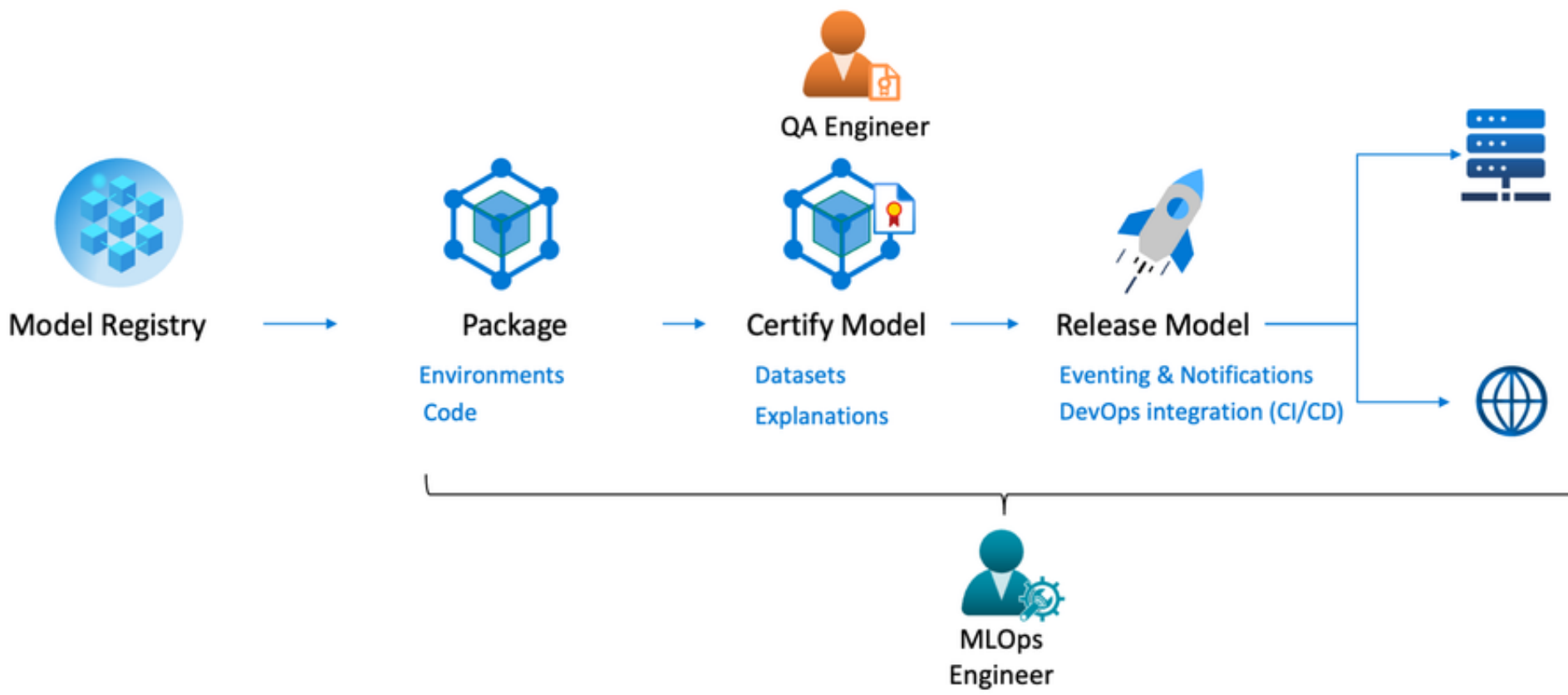
- Manual release
- Scoring script is version controlled with tests
- Release managed by Software engineering team

Application Integration

- Basic integration tests exist for the model
- Heavily reliant on data scientist expertise to implement model
- Application code has unit tests

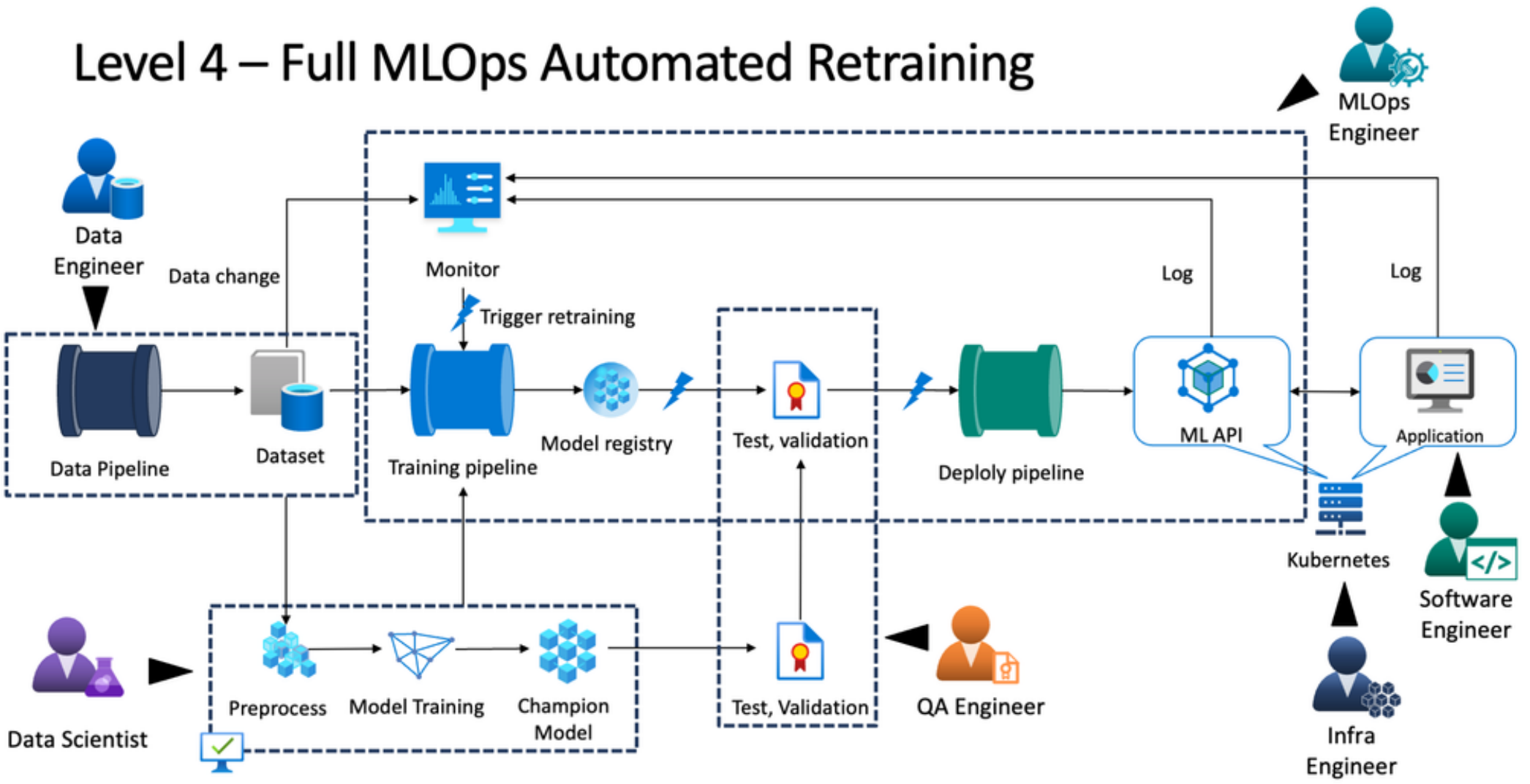
Level 3 – Automated Model Deployment

- Package model, Certify Model and release model are semi-automated.



People	Model Creation
• Data scientists: Working directly with data engineers to convert experimentation code into repeatable scripts/jobs • Data engineers: Working with data scientists and software engineers to manage inputs/outputs • Software engineers: Working with data engineers to automate model integration into application code	• Data pipeline gathers data automatically • Compute managed • Experiment results tracked • Both training code and resulting models are version controlled

Model Release	Application Integration
• Automatic release • Scoring script is version controlled with tests • Release managed by continuous delivery (CI/CD) pipeline	• Unit and integration tests for each model release • Less reliant on data scientist expertise to implement model • Application code has unit/integration tests



People	Model Creation
• Data scientists: Working directly with data engineers to convert experimentation code into repeatable scripts/jobs. Working with software engineers to identify markers for data engineers • Data engineers: Working with data scientists and software engineers to manage inputs/outputs • Software engineers: Working with data engineers to automate model integration into application code. Implementing post-deployment metrics gathering	• Data pipeline gathers data autom atically • Retraining triggered automatically based on production metrics • Compute managed • Experiment results tracked • Both training code and resulting models are version controlled

Model Release	Application Integration
• Automatic Release • Scoring Script is version controlled with tests • Release managed by continuous integration and CI/CD pipeline	• Unit and Integration tests for each model release • Less reliant on data scientist expertise to implement model • Application code has unit/integration tests

FOLLOW ME

For more similar post

ashishpatel2604