Challenges and Experiences with MLOps for Performance Diagnostics in Hybrid-Cloud Enterprise Software Deployments

Amitabha Banerjee

Chien-Chia Chen

Chien-Chun Hung

Xiaobo Huang, Yifan Wang, Razvan Chevesaran

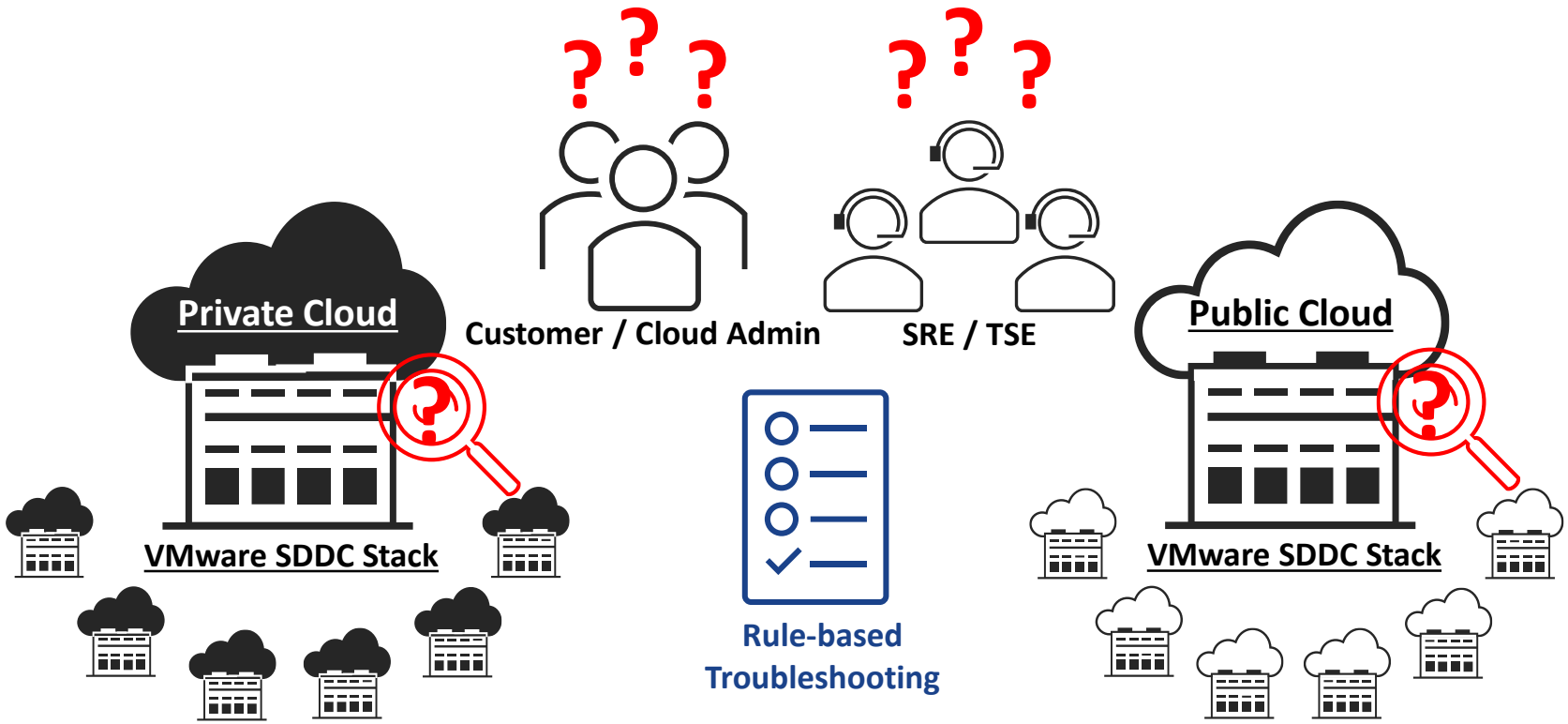
VMware, Inc.

@2020 USENIX Conference on Operational Machine Learning

July 27 – August 3, 2020

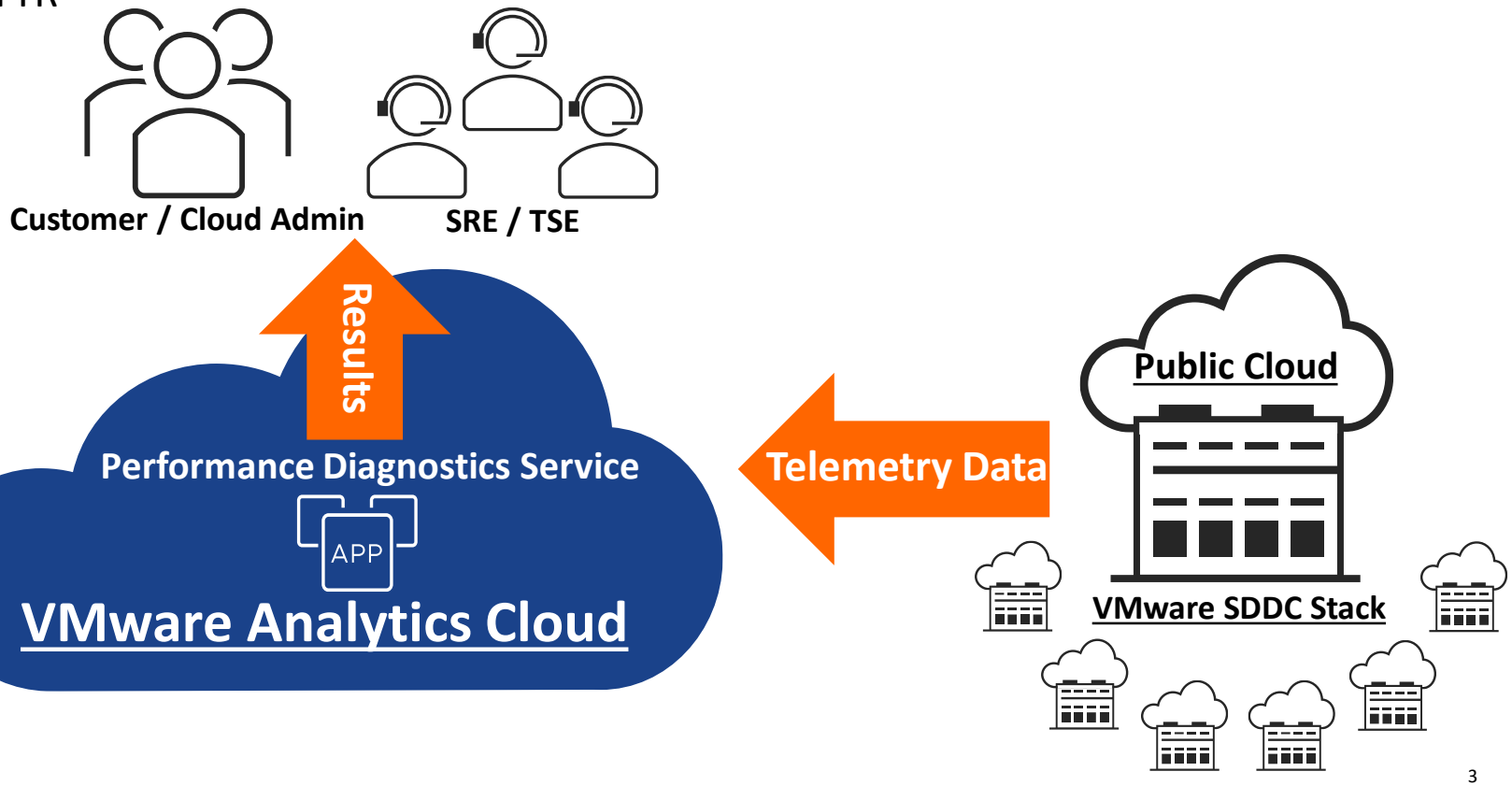
Performance Troubleshooting for Hybrid-Cloud Deployments

- VMware has the most large-enterprise customers who deploy our Software-Defined Datacenter (SDDC) stack in both their on-premises datacenters (private cloud)
  as well as VMware managed public clouds (VMC).
- **Detect** and **root-cause** performance issues at **scale** is extremely challenging.
- Traditional rule-based approach has limitations.

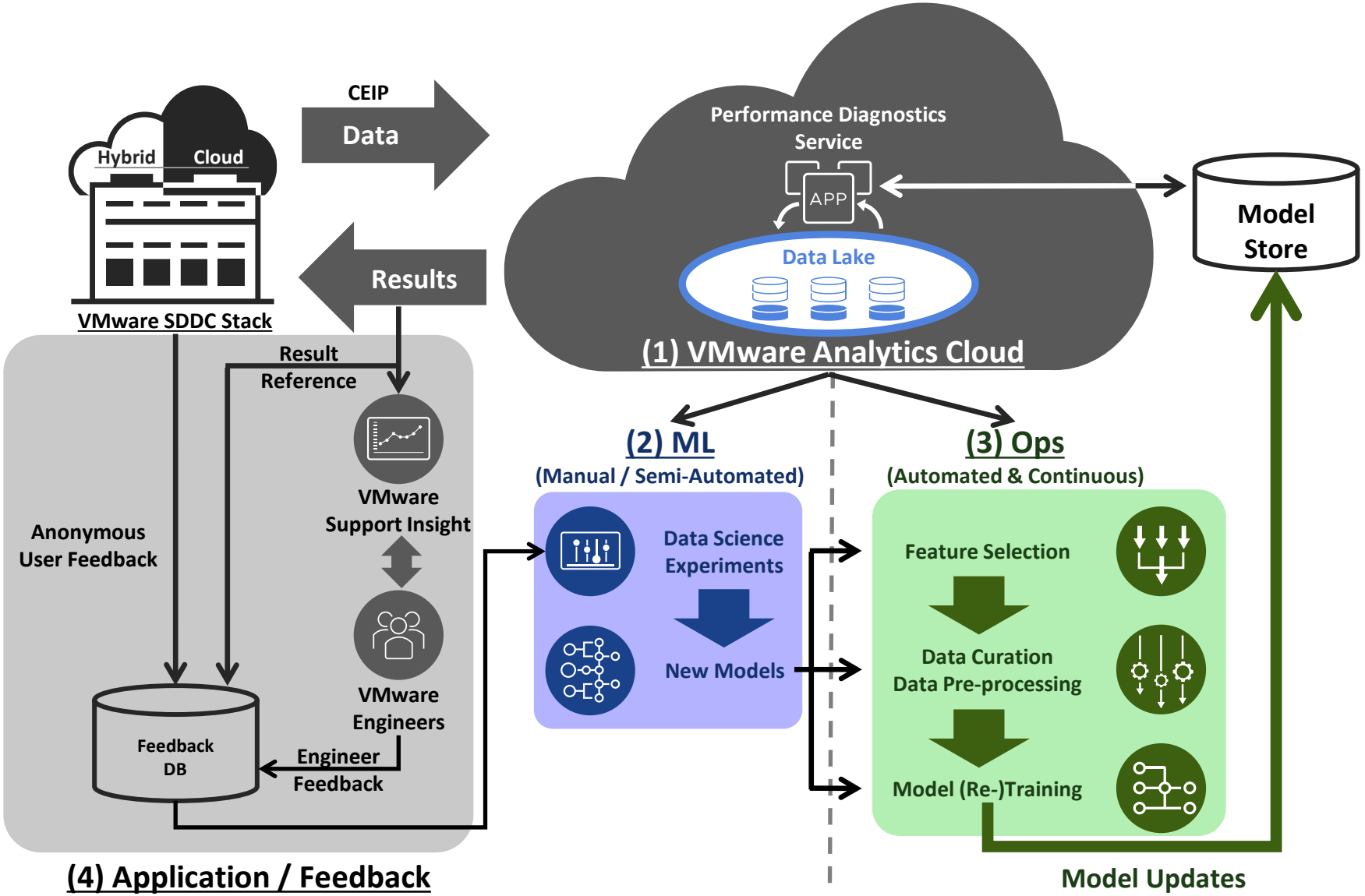


ML-Based Performance Diagnostics Service

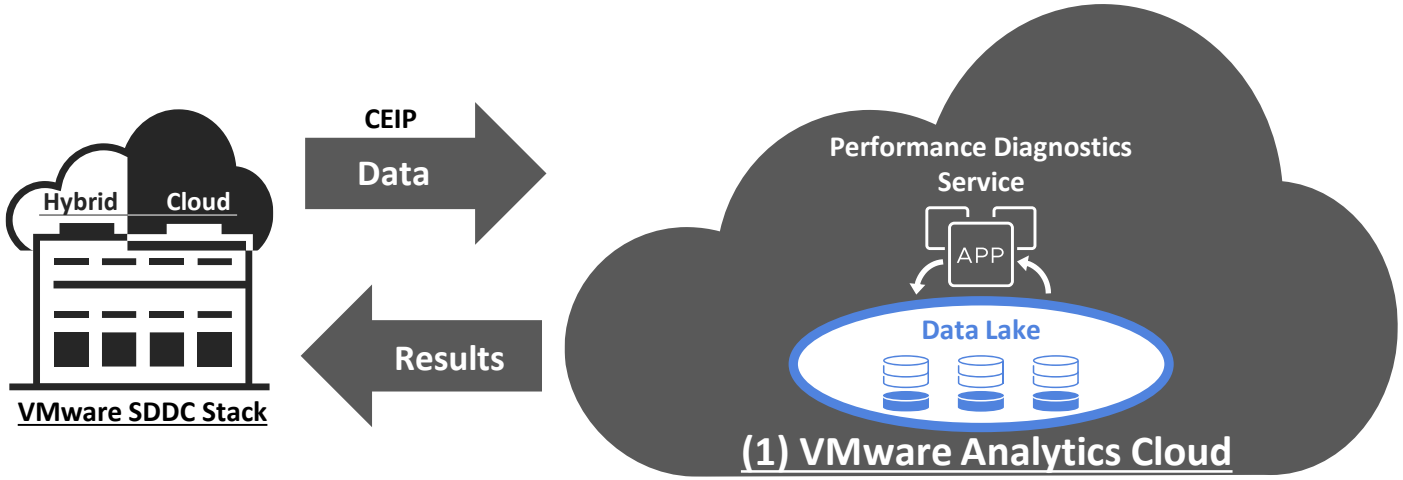
- Performance Diagnostics Service
  - Data-driven, ML-based
  - Detection and RCA
  - Unified UX for both on-prem and VMC
  - Decouple intelligence from product releases
  - Proactive: Reduce MTTD and MTTR



Building Performance Diagnostics Service

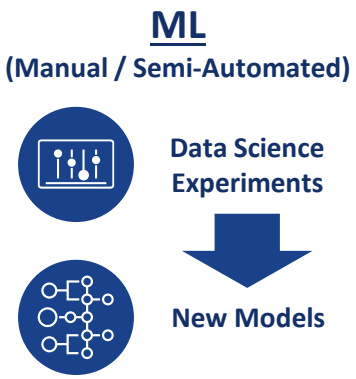


(1) VMware Analytics Cloud



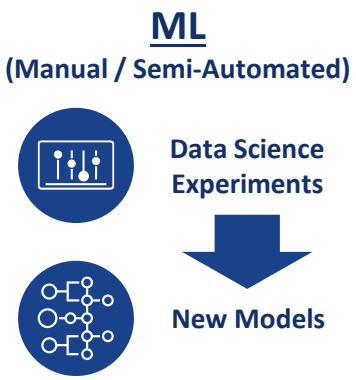
- Governed by VMware **CEIP**
  - Privacy: Customers agree to send anonymized data
    - Telemetry data streamed from **all** product deployments (under CEIP)
    - Usage, hardware/software configurations, performance counter readings
    - NO contents and NO logs
  - ACL: Data only available for VMware internal purposes
  - Other data compliance (GDPR/CCPA)
- Consumed by a cloud service designed using Apache Spark
- Performance Diagnostics Service runs as a job in VAC

(2) ML—Performance Issue Detection and RCA



- Input: performance counter readings
  - IOPS, I/O throughput, CPU utilization, queue utilization, etc
  - Thousands of counters across SDDC stack
- Problem (1): Detect performance **anomalies**
  - Does the SDDC perform normally?
    - E.g., disk I/O latency is “normal”, memory usage is “normal”
  - Data scientists experiment and develop ML models
- Problem (2): **Root Cause Analysis (RCA)**
  - RCA:
    - What is the cause of the anomaly?
    - How to remediate?
    - Explain the anomaly and provide recommendation
  - Statistical learning / unsupervised learning
    - Decision trees, RCA rules, clustering, correlations
    - E.g., if packet drops are abnormally high and I/O latency is also abnormally high,
      Root cause: packet drops -> latency anomaly
      Remediation: Examine physical links/switches and network utilization

(2) ML—Validation

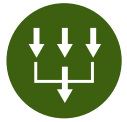


- Labels
  - Anomaly labels
  - RCA labels
- **Manual** label
  - Highly depend on product experts
- **Synthesized** label
  - Inject controlled performance perturbation
    - Artificial packet drops, artificial I/O latency, etc
  - Run various synthetic workload on internal testbeds

(3) Ops—Feature Selection

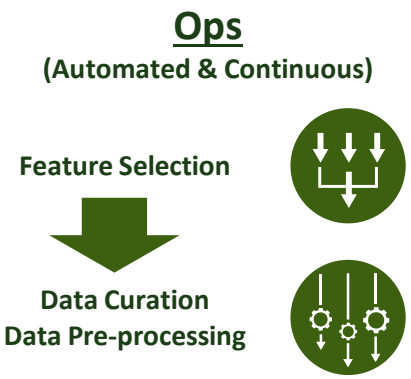
Ops
(Automated & Continuous)

Feature Selection



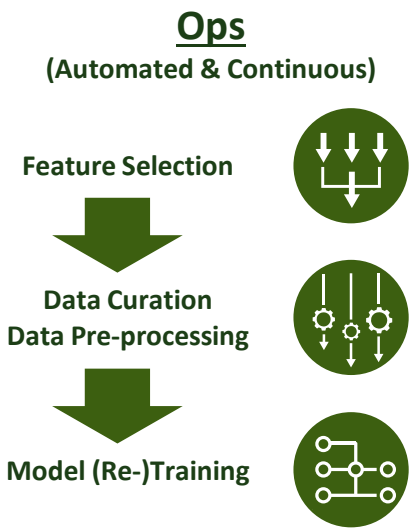
- Fully automated and continuous Ops pipeline
- **Feature selection**
  - Extremely high dimension
  - Product experts provide candidates sets of features
  - Feature selection pipeline to train models with every feature set
  - Automatic retrain once feature sets change
- *Why not all feature combinations?*
  - Too many of them → too much resource consumption
- *Why not dimension reduction?*
  - Doesn't work; statistical relations ≠ consequential relations
- *Why not dimension reduction + some feature engineering?*
  - Performance gain doesn't justify engineering cost

(3) Ops—Data Curation and Pre-processing



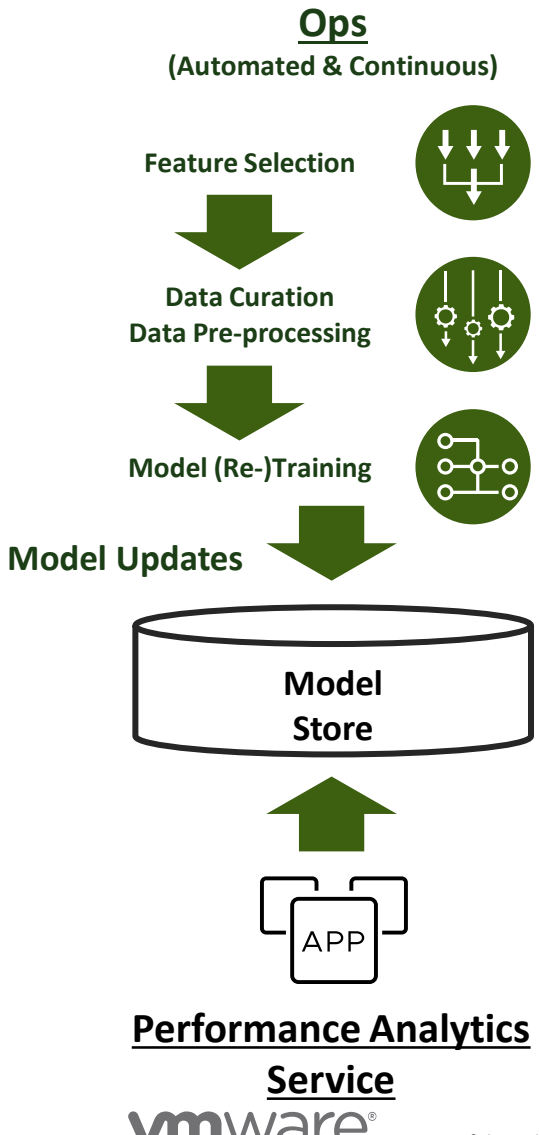
- Some models are sensitive to data distribution
- **Data Pre-processing** methods
  - Normalization (standardization, Box-Cox transform, …etc)
  - Band pass filter → remove outlier
  - Other curation → avoid dividing by zero
- **Pre-processing chain**
  - Series of data transformation
  - Multiple chains according to permutation
- Automatic selection of the best pre-processing chain

(3) Ops—Model (Re-)Training



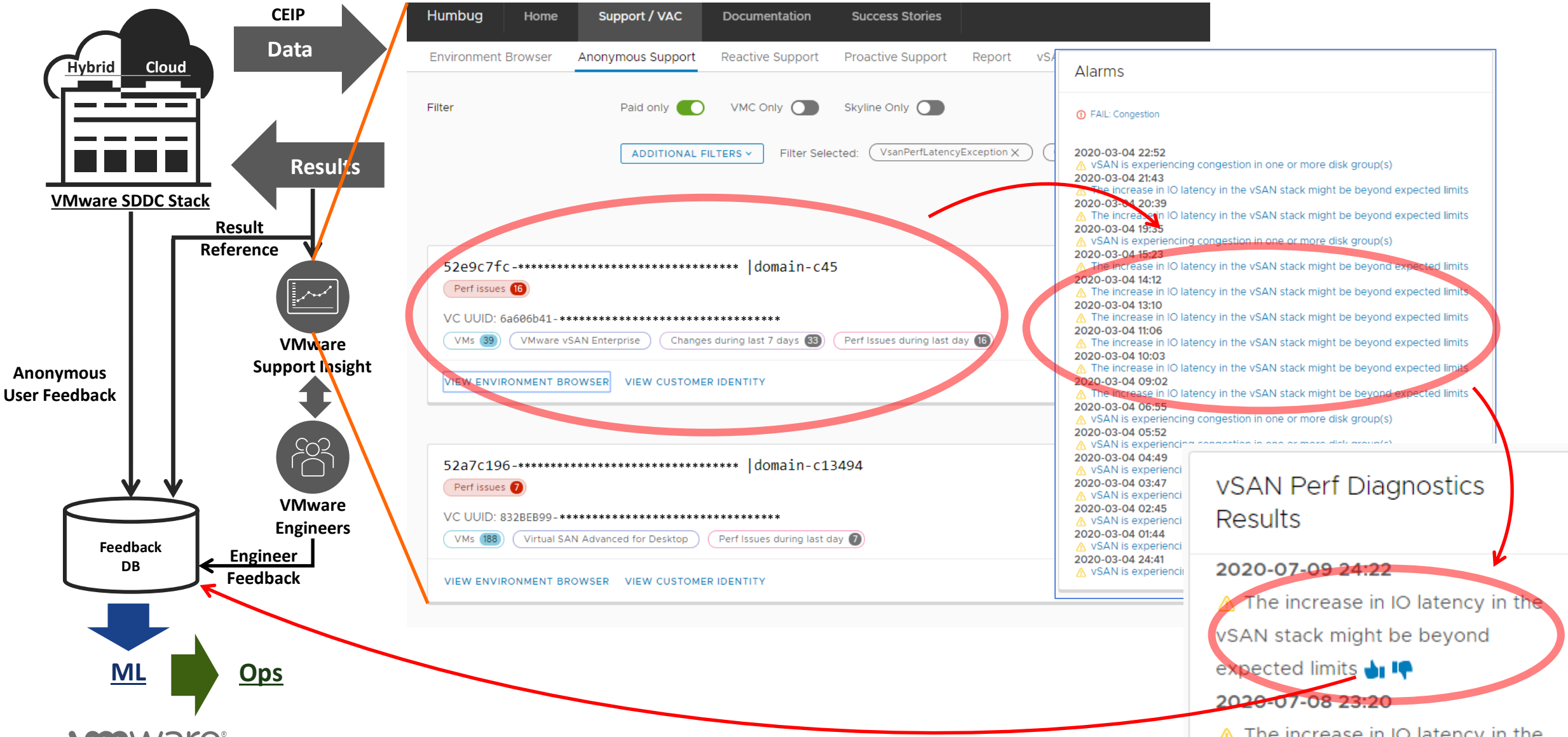
- Model offering
  - Regression, autoencoder, isolation forest, principal component analysis
- **Ensemble**
  - Hyper-parameters: ***models***, ***feature sets***, ***pre-processing chains***
  - Determine hyper-parameters with labeled dataset
  - Aggregation of model predictions: boosting, majority vote
- Custom accuracy metric: **percentage of correct predictions**
  - Balance false positives and false negatives
  - F1-score does not work well in anomaly detection scenario (i.e., *skewed distribution*)

(3) Ops—Model Serving



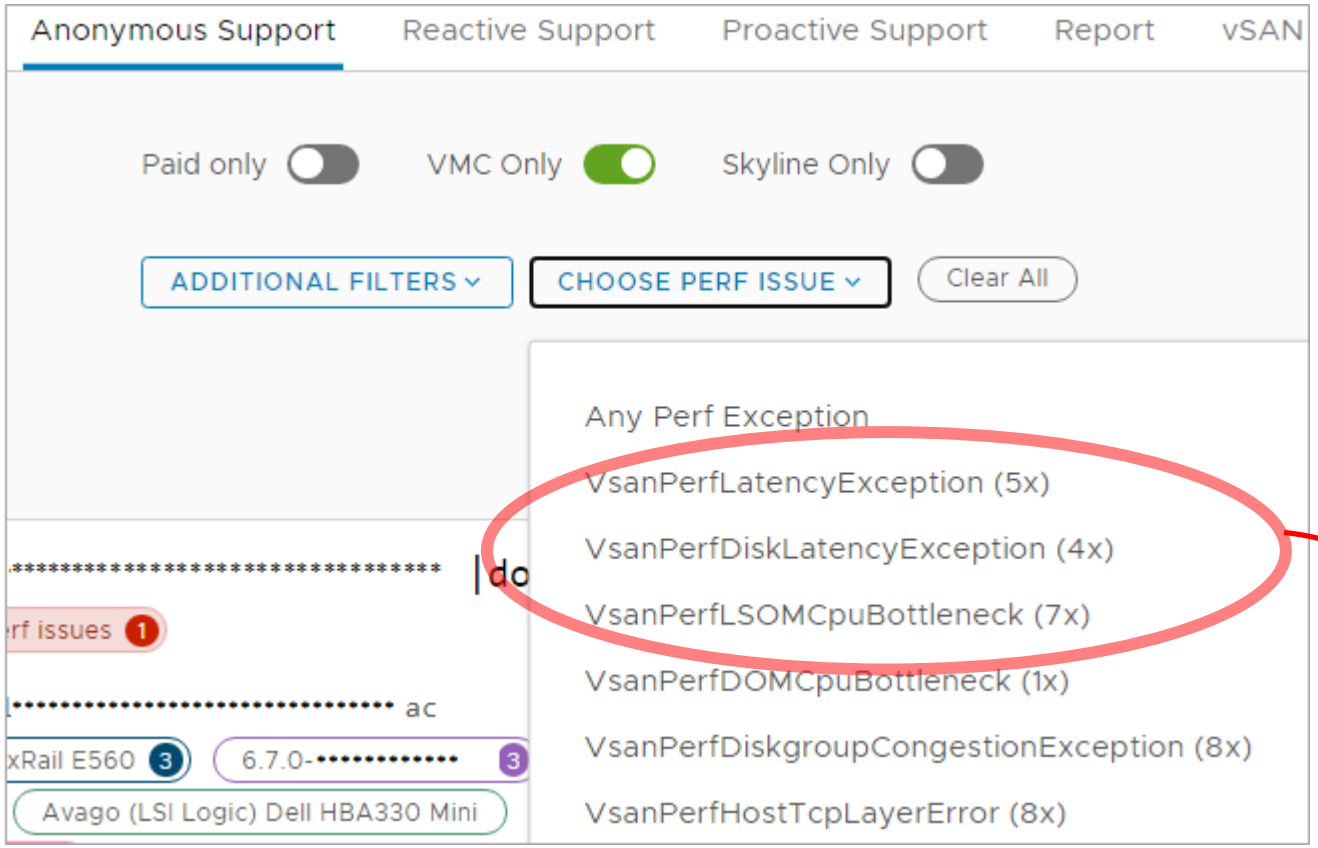
- Model store for trained models
  - Model instance
  - Active
  - Version
  - Timestamp
  - Feature set
  - Pre-processing chain
  - Measured accuracy
  - Training dataset
- **Performance Diagnostics Service** chooses a model for inference
  - Most recent
  - Most accurate
  - Specific version (*e.g., rollback*)

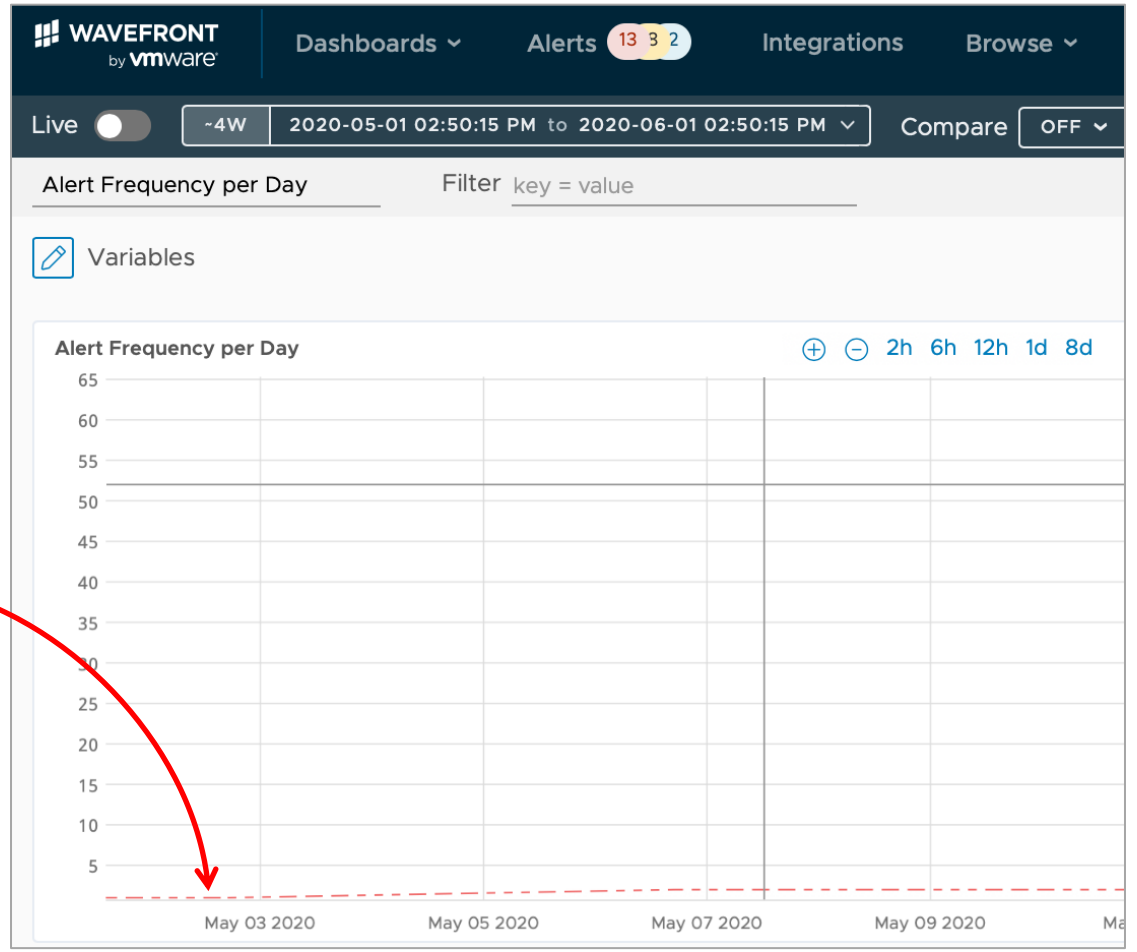
(4) Application / Feedback



Performance Drift Monitoring

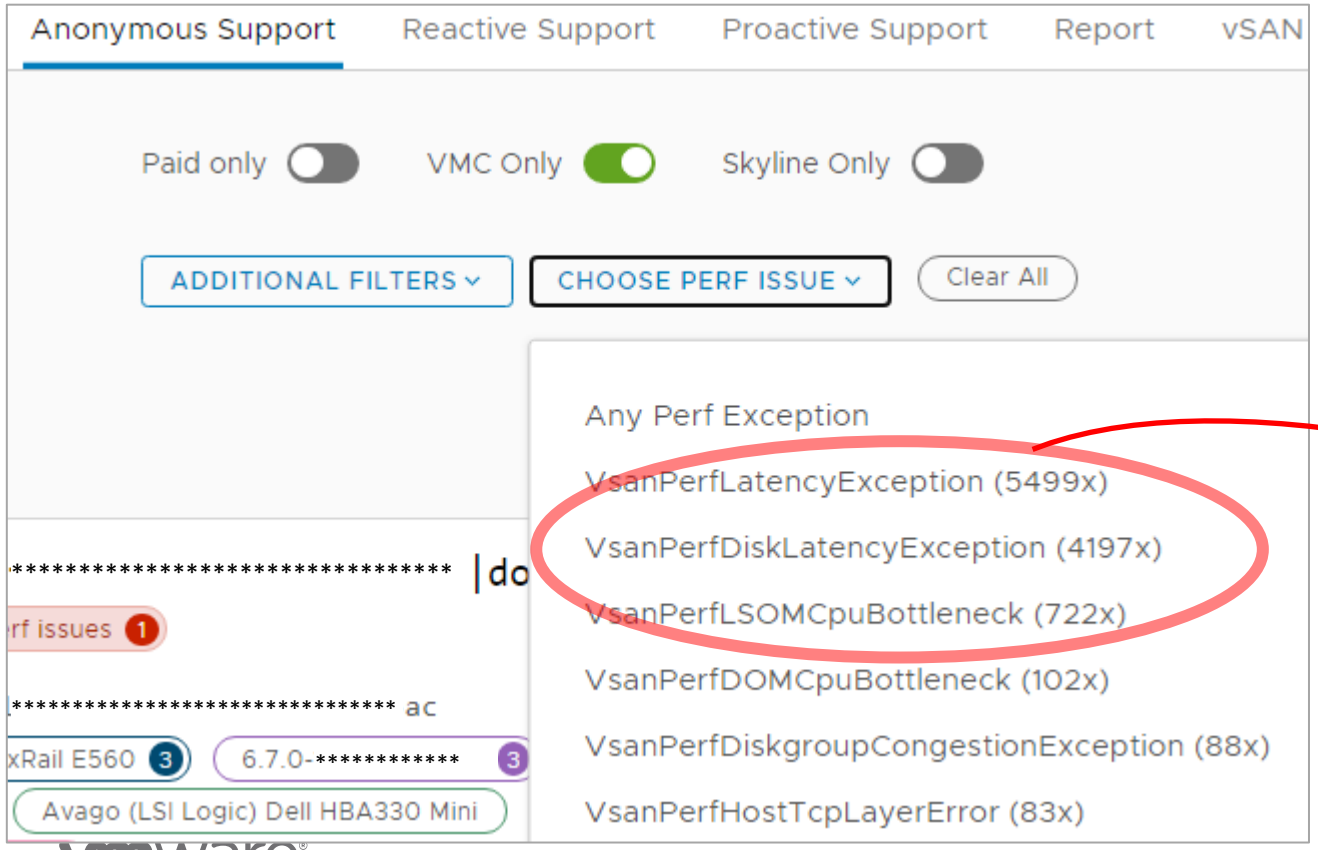
- Monitor changes in accuracy and anomaly detection rate
- Spikes or dips imply performance changes
- Good drifts: thumb-up label to reinforce model training

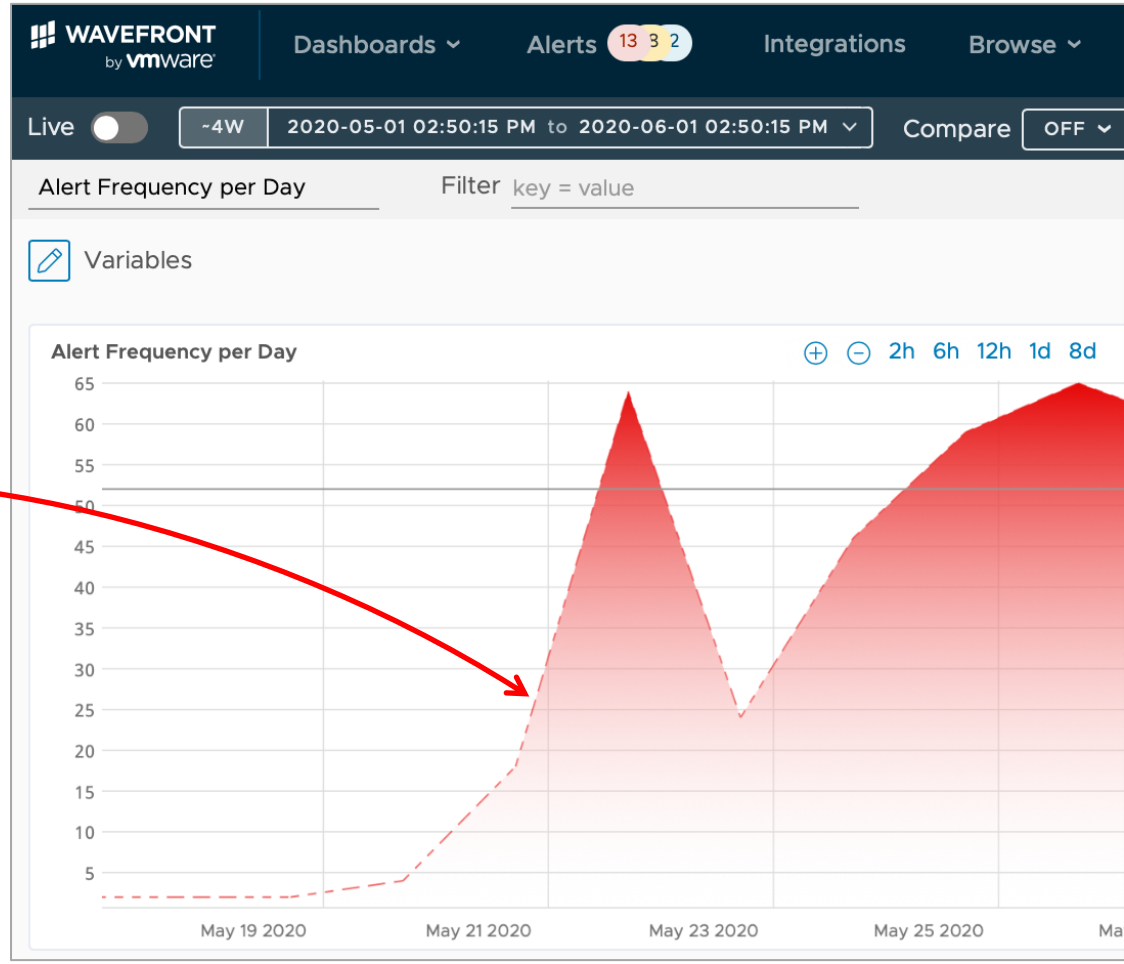




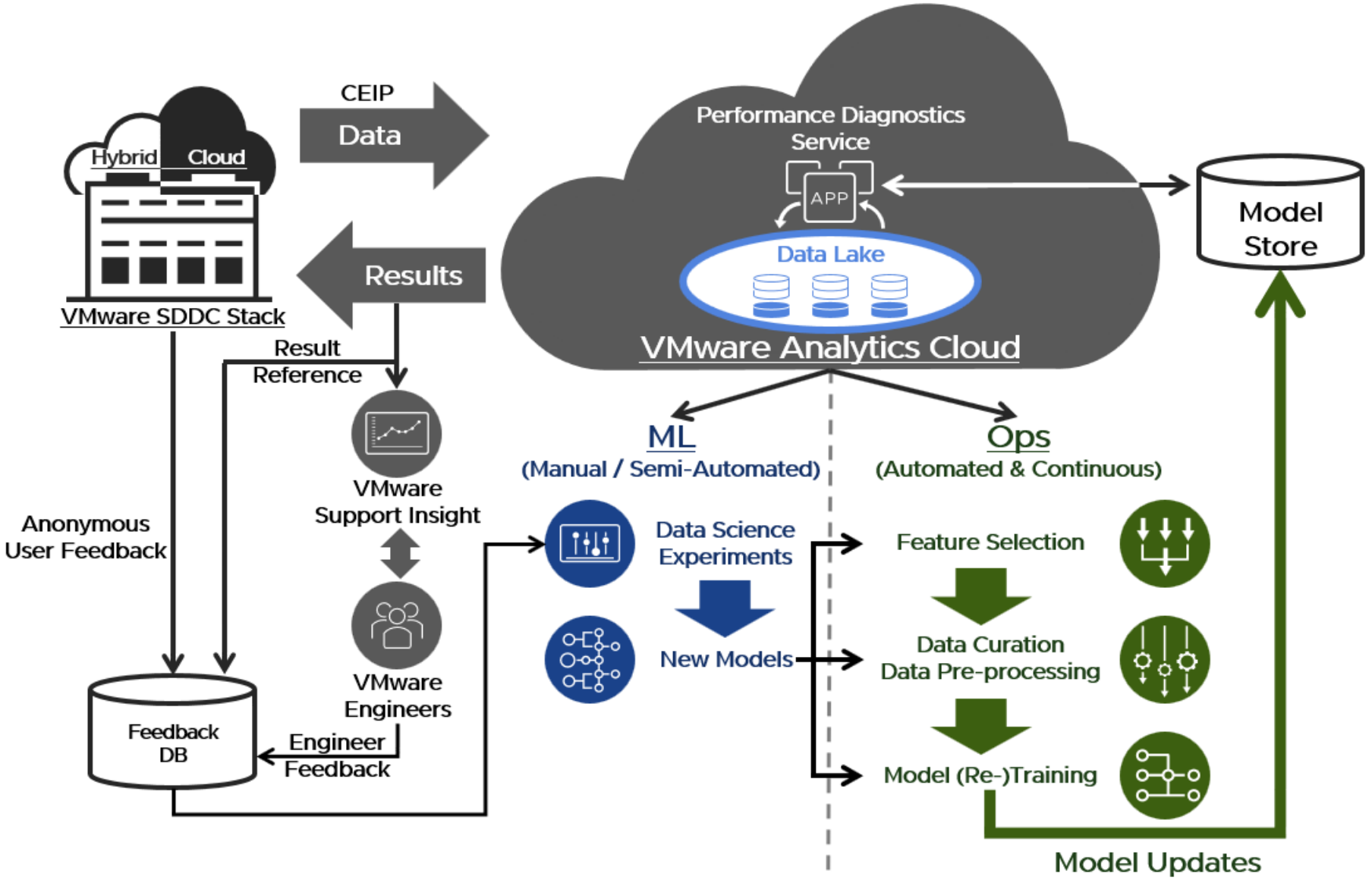
Handling New Performance Issues

- Bad drifts: might indicate new, unseen performance issues
- Require manual RCA to determine actions
  - Either fix product or change model



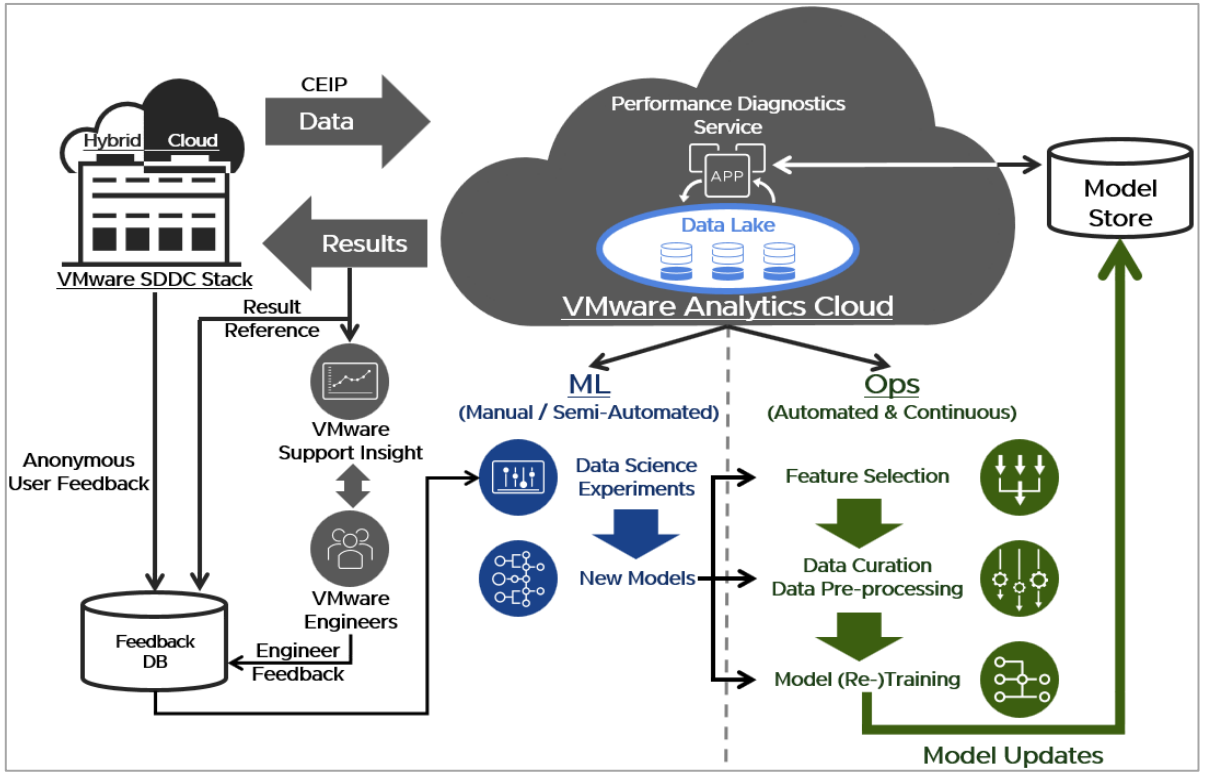


Put Everything in Production!



Takeaways

- **Continuous and automated training and serving**
  - Automatic feedback consumption
  - Keep the production models up-to-date without human intervention
- **Monitoring dashboard in production**
  - Visualize the deployment performance for easier tracking and alerting
- **Orchestrated experiment environment**
  - Validate model behavior with synthesized setup and data



Thank You



{banerjeea,chien-chiachen,hchienchun}@vmware.com

ACKNOWLEDGEMENT

VMware Analytics Cloud Team
VMware Support Insight Team
VMware Research Group
Parikshit Gopalan
Udi Wieder
VMware Performance Group
Rajesh Somasundaran
Bruce Herndon
Chuck Lintell