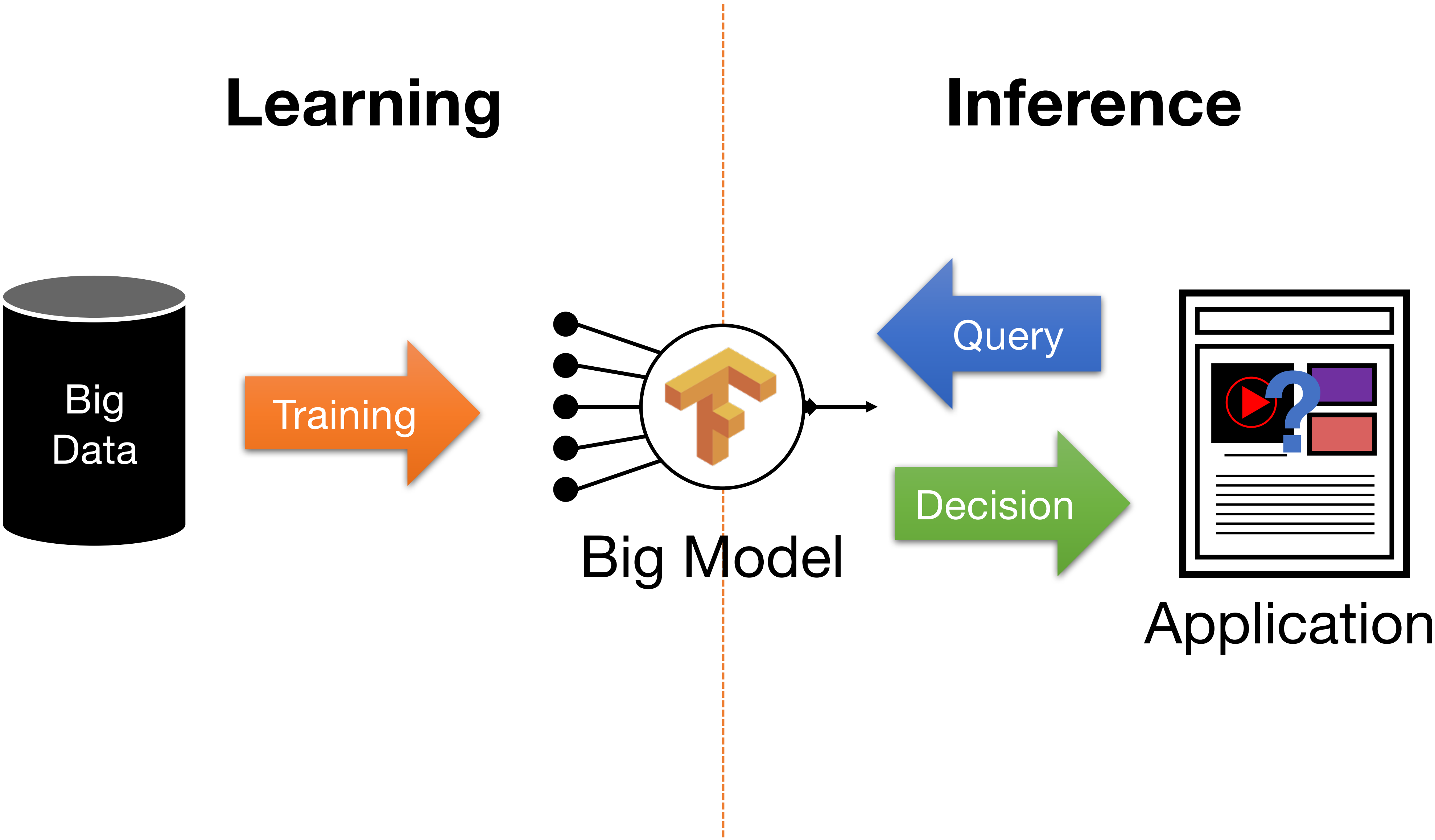
RISE to the challenges of Intelligent systems

A brief overview of machine learning research topics in 294

Machine Learning

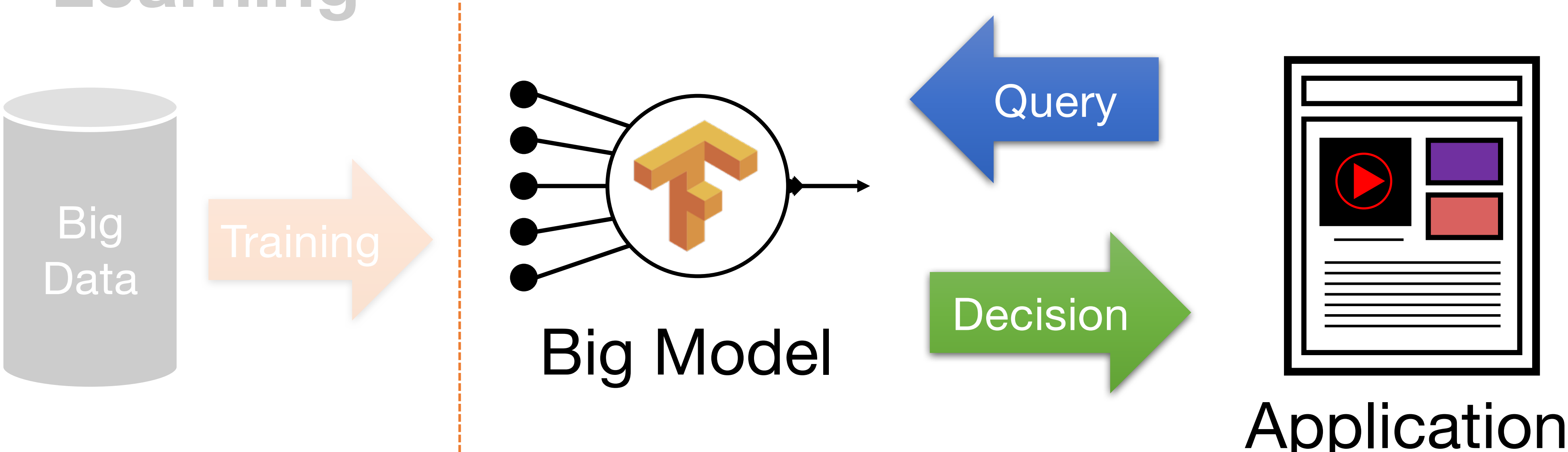
Timescale: minutes to days

*Heavily studied ... primary focus of the **ML research***


Learning
Inference
Big
Data
Training
Query
Decision
Big Model
Application

Learning

Inference



Often **overlooked**

Timescale: ~10 milliseconds

A new focus in the RISELab

Improving the Perf. of **Inference**

Reducing Latency

- **Approximate caching** to address high-dim continuous features
- **Anytime predictions** to tradeoff accuracy and latency during inference
- **Model compression** to reduce inference costs (memory and CPU)

Improving Throughput

- **Batching & scheduling technique** to leverage parallel hardware
- **Model cascades** to separate simple and complex queries

System Failures

- **Graceful degradation** as models and resources fail
- **Abstractions** to communicate loss of performance to end-user app.

High Perf. Prediction Serving

- Caching techniques fail on continuous features
  - Study locality sensitive hash functions for **approximate caching**
- Less accurate pred. often better than no pred.
  - Derive **anytime inference algs.** with error bounds
- Recent models (DNNs) are often large → costly
  - **Compress models** using context (e.g., condition on class bias)
  - **Cascade models** to separate easy from challenging queries
- Where are predictions rendered? (mobile, GPU, cloud, …)
  - **Split computation** across mobile and cloud
  - **Schedule models** across accelerators to maximize performance

Robust **Inference**

How do we

- identify inputs that are **outside the domain** of the model
  - nighttime images for a daytime model
- recognize **poorly performing models** without feedback
  - e.g., feature and label distribution deviates from training data
- **Calibrate** and **communicate uncertainty** in predictions
  - e.g., ensembles & CIs → increased **overhead** …

at scale with rapidly changing models and data?

Ensuring Robust Predictions

- Data is constantly changing (e.g., new features, signal…)
  - detect **stale models**
  - correct for **real-time covariate shift**
- Often predicting “unsure” is better than bad predictions:
  - use **bootstrap** in **real-time** to quantify uncertainty
  - discriminatively train models that **decline to predict**
- How do we know when models are performing poorly?
  - identify **failing models** without explicit feedback
  - help **diagnose models** that have failed?

Inference is moving beyond the cloud

Opportunities

- Reduce latency and improve privacy
- Address network partitions

Research Challenges

- Minimize **power consumption**
- **Limited hardware** & long life-cycles
- **Protect models** from attack
- Develop new **hybrid models** to leverage cloud and devices

Security: Protecting Models

Data is a core **asset** & models capture the **value** in data

- **Expensive**: many engineering & compute hours to develop
- Models can **reveal private information** about the data

How do we **protect models** from being stolen?

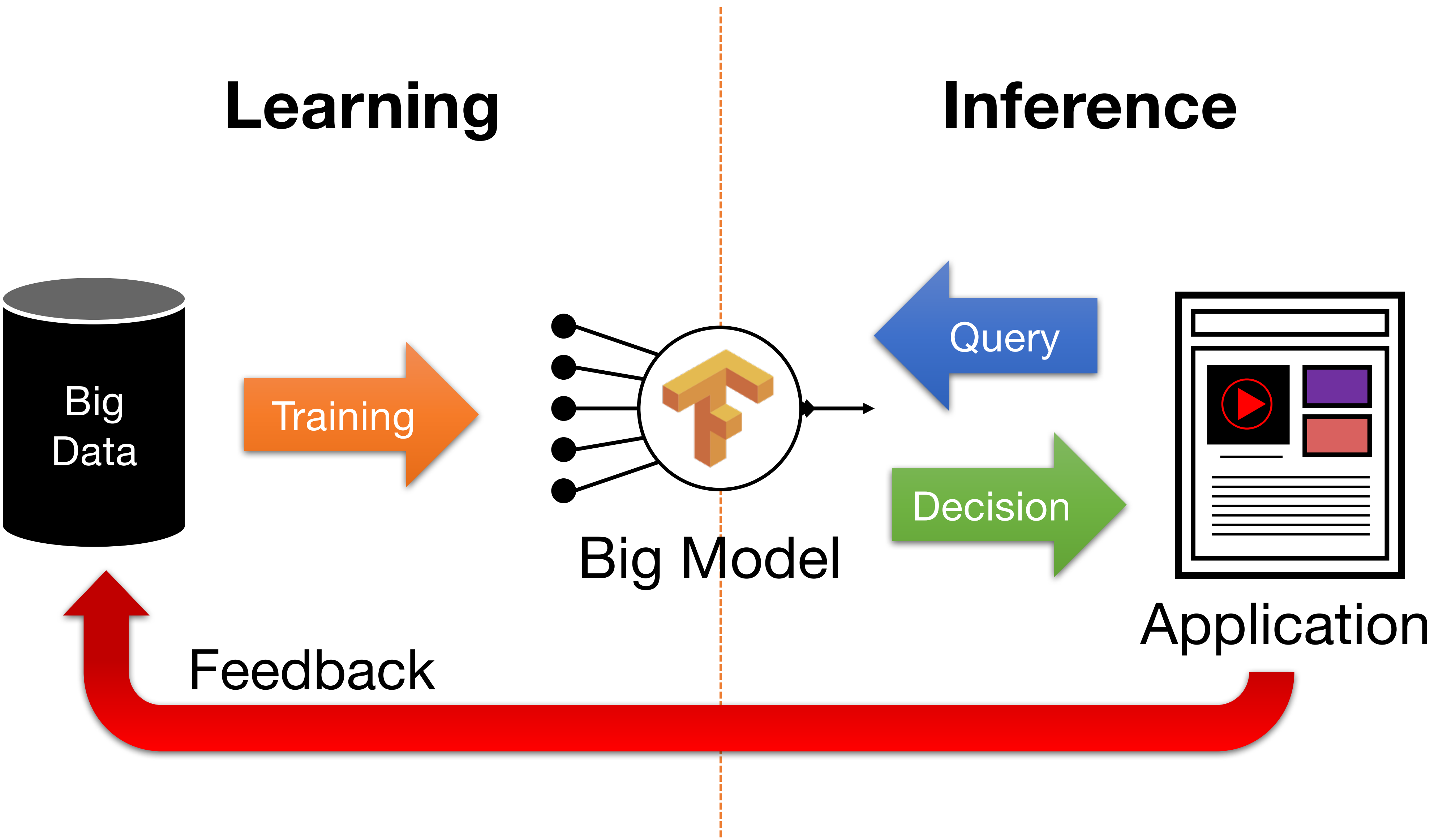
- Prevent them from being copied from devices (**DRM**? **SGX**?)
- Defend against **active learning attacks** on decision boundaries

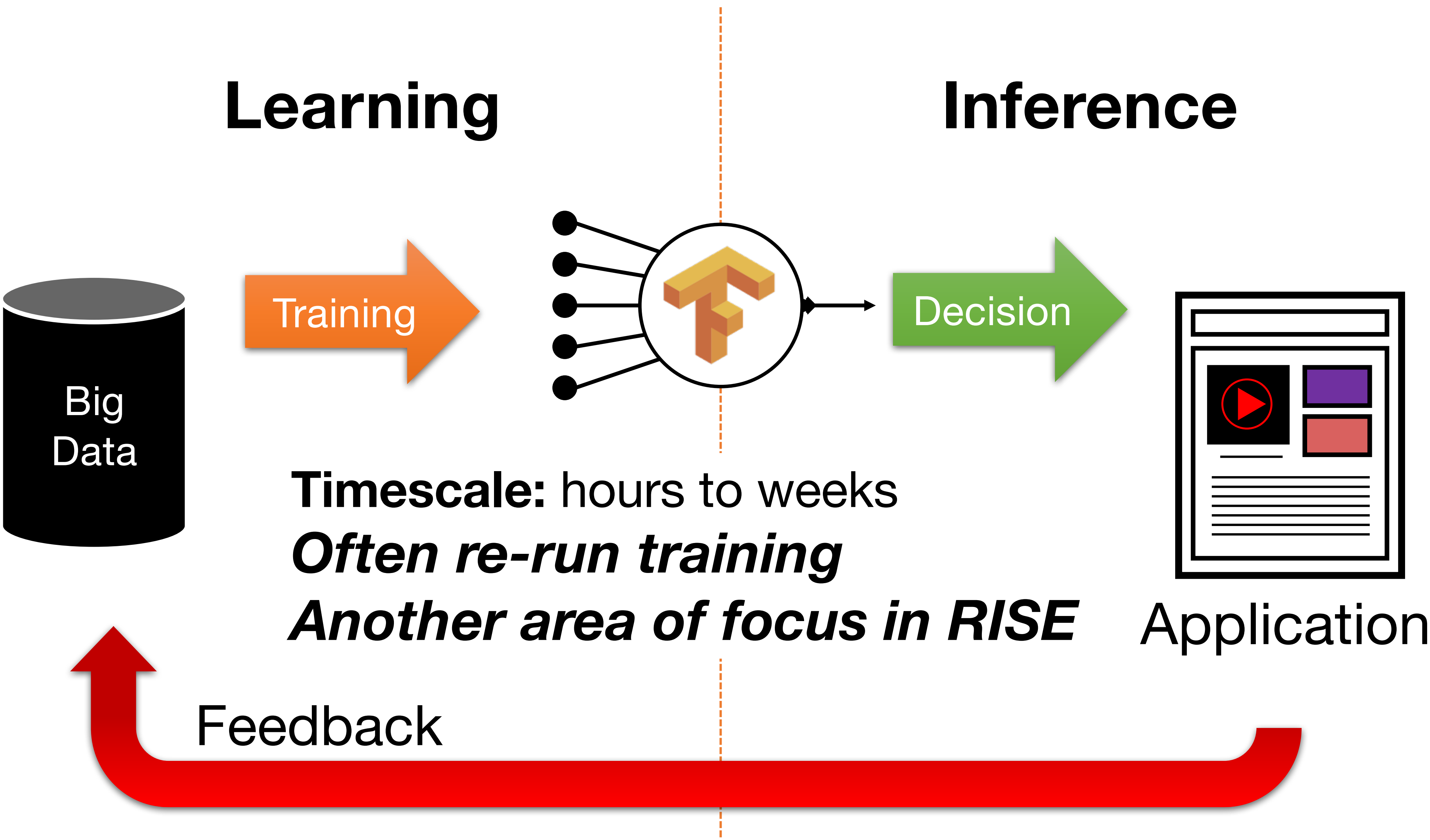
How do we identify when models have been stolen?

- **Watermarks** in decision boundaries?

Machine Learning and Security

- Private Predictions
  - partially **homomorphic encryption** on advanced models
  - combined **secure** hardware with **accelerators** (SGX + GPU)
- Active Learning Attacks
  - efficiently **learn through queries** to prediction services
  - can we **identify active learning attacks**
- Securing Models
  - protect models **deployed on mobile** devices
  - **watermark** a model and its predictions


Learning
Inference
Big
Data
Training
Query
Decision
Big Model
Application
Feedback

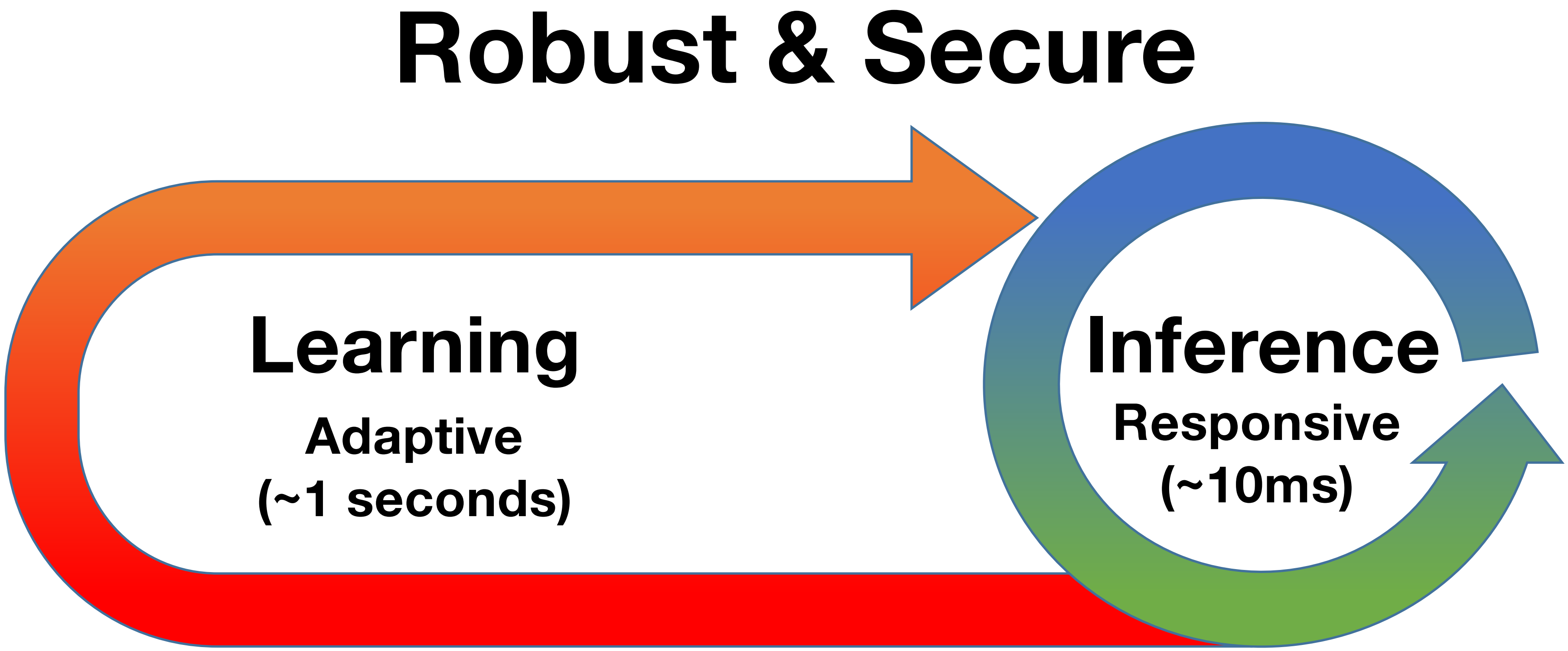

Learning
Inference
Big Data
Training
Decision
Application
Timescale: hours to weeks
Often re-run training
Another area of focus in RISE
Feedback

Why is **Closing the Loop** challenging?

- Combines multiple **systems** with different design goals
  - **Latency** vs **Throughput**
- Exposes system to **feedback loops**
  - *If we only play the top songs how will we discover new hits?*
- Must address **concept drift** and **temporal variation**
  - How do we **forget the past** and **model time directly**
  - **Model complexity** should evolve with data
- Personalization and delayed reward → emphasis on **MTL** and **RL**
- Learning with complex **model dependencies**
- **Robust learning** against **adverserial data**

Dealing with Feedback

- Need to respond to feedback in real-time
  - avoid costly **retraining** process
  - how (at what rate) do we **forget** old data
- Feedback may not be explicit or timely
  - leverage **implicit feedback**? (select between models …)
  - deal with **delayed feedback**? (stream processing, RL, …)
- Feedback is biased by our actions
  - leverage advances in **bandits** within serving systems
  - actions offten affect future decisions → **RL in serving** systems



The focus of **Learning Systems** in **RISE**

The focus of **Learning** **Systems** in **RISE**

Prediction Serving

Deep Learning

Reinforcement Learning