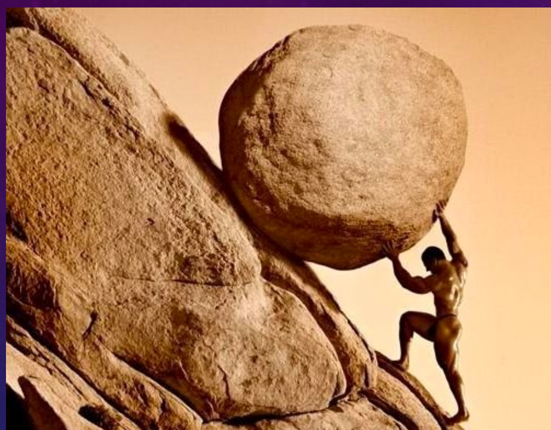


Architecting ML-Enabled Systems

Rick Kazman

University of Hawaii

1

Architecture for ML-intensive Systems

- Premise: Including an ML component in a software system has architectural implications
- Do we all agree that is true?
- When does it matter?
- What do we do about it?

2

My Position

- Architecture matters for all systems, not just ML-intensive ones.
- But the challenges of ML-intensive systems are considerable.
- My research goal is to explore these challenges.
- In particular I am interested in how an architect responds to the various system drivers that are commonly found in ML-intensive systems, e.g.
  - key quality attribute requirements (such as performance, scalability, integrability, testability, etc.)
  - constraints (legal, regulatory, cost, etc.)
  - deployment options

3

The Problem

Gartner reports that 85% of AI projects fail and 53% never leave the prototype stage.

- Challenges to fielding these projects can be introduced in the DS process, the design and deployment, and during system evolution
- How can architecture help in the second phase?
- Currently this process is poorly structured.

4

The Desired End State

- My desired outcome from this research is to identify a set of architectural mechanisms—patterns and tactics—that are commonly used in architecting ML-intensive systems.
- So we are interviewing architects to discover:
  - the architectural mechanisms that they have employed,
  - the quality attributes they believe these mechanisms address,
  - the costs, risks, and tradeoffs that they have experienced.

5

ML-Enabled Systems Are Not Monoliths

Some premises

- ML-enabled systems are highly variable, and there is no single reference architecture that can apply to all these systems.
- The extent to which integrating an ML component introduces architecture concerns varies based on several factors about the ML component.
- ML-enabled systems are still software systems and share many common concerns with conventional software systems.

6

ML-Enabled Systems Are Not Monoliths

Some questions

- How stable are the problem and data over time?
- How tightly coupled is the ML component with the rest of the system?
- How much monitoring is required to assure model performance? How much is possible?
- How complex is the ML component?

7

How Does ML Influence System Architecture?

System-level concerns

8

How Does ML Influence System Architecture?

System-level concerns

Component-level concerns

9

How Does ML Influence System Architecture?

System-level concerns

Component-level concerns

System environment concerns

10

How Does ML Influence System Architecture?

- System-level concerns
- Component-level concerns
- System environment concerns
- ML-specific concerns

11

The Questionnaire

- Let us now consider each of these categories of concerns in more detail.
- For each category we provide questions to the architect, along with examples of possible answers to those questions, as a means of stimulating discussion.

12

How Does ML Influence System Architecture?

System-level concerns	Component-level concerns
System environment concerns	ML-specific concerns

13

System-level concerns

System-level Concerns

- These are concerns that influence the system as a whole and the ML component does not function differently than any other software component.
- This includes, for example, concerns related to security, performance, modularity, maintainability, availability, and testability.

14

System-level concerns

Time sensitivity of prediction

- How important is it that predictions are produced at a specified interval or by a specified deadline? How does prediction time influence system performance as a whole?
- Examples:
  - BI Dashboard
  - Image-based navigation

15

System-level concerns

Data quality

- Can the system detect and adapt to different levels of data quality? If so, can this be done dynamically?
- Examples:
  - Fraud detection system
  - IoT monitoring system

16

System-level concerns

Degree of model modularity

- How tightly coupled is the model to the rest of the system?
- How stable are the APIs, e.g. If the model algorithm changes do the interfaces to it change?
- Is there a choice of algorithms that can solve the problem with comparable performance and accuracy?
- Is the selected or preferred algorithm supported within more than one ML framework? Are the data interfaces for producing the prediction compatible with the APIs for an algorithm in a chosen framework?
- Does the ML function depend on a sequence of models to produce intermediate results that serve the prediction?
- Are the discrete tasks well-understood, cohesive, and well supported, e.g. computer vision?

17

System-level concerns

Degree of model modularity

- Example:
  - text classification

18

System-level concerns

Portability

- Is it expected that the model will need to be served on multiple hardware platforms? Or is it anticipated that it will need to be ported to other platforms in the future?
- Examples:
  - single cloud platform
  - vendor-agnostic solution

19

System-level concerns

Testability / Model verifiability

- Can the output of the ML component be independently verified?
- Can model performance be judged based on knowledge of false positives and false negatives?
- Does the system allow for arbitrary capture of system state or intermediate results for more fine-grained testing?
- Examples:
  - basic model testing
  - conditional estimates or business logic

20

System-level concerns

Monitorability

- Can the state of the system and of the ML components within the system be monitored at runtime?
- Can these monitoring needs be achieved by a single system component?
- Does the ML component monitoring require the deployment of an additional model or models?
- Examples:
    - system resources
    - ML-specific measures

21

How Does ML Influence System Architecture?

System-level concerns	**Component-level concerns**
System environment concerns	ML-specific concerns

22

Component-level concerns

Component-level Concerns

- These are concerns that involve the ML component, or components, which have significant non-local effects on the system.

23

Component-level concerns

Retraining cadence and Deployment effort

- How frequently is it anticipated that a model will need to be retrained and redeployed?
- How much are deployments automated (versus requiring human-intensive steps)?
- Examples:
  - medical devices
  - recommender system

24

Component-level concerns

Drift Rate

- Is it expected that the data might drift considerably over time?
- Are the conditions under which drift occurs predictable? Will these conditions need to be monitored?
- What kind of drift is anticipated – distributional drift, concept drift, or both?
- Examples:
  - blood glucose readings
  - consumer preferences

25

How Does ML Influence System Architecture?

System-level concerns	Component-level concerns
System environment concerns	ML-specific concerns

26

System environment concerns

System Environment Concerns

- These concerns focus on how the system environment needs to be appropriately provisioned for, in particular, the ML function.

27

System environment concerns

Distribution

- Is it anticipated that the ML functionality within the system will be:
  - fully centralized
  - federated (with a centralized model interacting with edge devices, each of which holds its own domain-specific model)
  - fully decentralized (where each node of the system is a peer)?
- Examples:
  - blood glucose monitor
  - image detection on a phone
  - highly-personalized federated learning

28

System environment concerns

Computational intensity (training)

- If training is within the scope of the system, what is the computational power required by the ML algorithm, in terms of both space and time, for this training?
- This may be a function of both the inherent complexity of the approach chosen and the size of the dataset.
- Examples:
  - Ordinary least squares is O(nm) [n = observations, m = features] in time and space,
  - Random forest is O(t m n log n) in time and O(md) in space [t = number of trees, m = features, n = observations, d = number of nodes].

29

System environment concerns

Computational intensity (prediction)

- What is the computational power required by the ML algorithm, in terms of both space and time, for prediction?
- This again may be a function of both the inherent complexity of the approach chosen and the size of the dataset.
- Examples:
  - Ordinary least squares is O(nm) in time and space [n = observations, m = features]
  - Random forest is O(t log n) in time and O(td) in space [t = number of trees, d = number of nodes].

30

System environment concerns

Computational power

- What is the computational power provided by the platform on which the ML algorithm is being run?
- Examples:
  - Single IoT device
  - GPU or NPU cluster hosted in a cloud infrastructure

31

System environment concerns

Data Fusion

- Does information from multiple sources or sensors need to be composed into a single data representation before it is presented to the model?
- Is it anticipated that these sources or sensors will change in the future?
- Examples:
  - fraud detection
  - cookies for user behavior

32

System environment concerns

Analytic redundancy

- Is there a requirement for a redundant component (or set of components) to provide additional sources of information or truth, perhaps with lower accuracy and less demanding performance characteristics?
- Examples:
  - expert system vs. ML model
  - ad placement

33

System environment concerns

Data volume

- What amount of data must the system ingest and process to produce predictions?
- Examples:
  - hi-res satellite images
  - hand-created .xlsx file

34

How Does ML Influence System Architecture?

System-level concerns	Component-level concerns
System environment concerns	**ML-specific concerns**

35

ML-specific concerns

ML-Specific Concerns

- These are concerns that are important to overall system success, but where the solution to these concerns lives primarily outside the decisions made for the software architecture.
- Such solutions are typically centered around the details of the functions of the ML components, and how they are envisioned, designed, and created.

36

ML-specific concerns

Non-architectural concerns

- Does your system have requirements to meet some standards for fairness, privacy, or ethics?
- Are there regulatory considerations (e.g. GDPR) that constrain what kind of data can be collected or stored, or how it may be used?
- Are these concerns complex enough that they require their own components (or sub-systems) within the system?
- Are they handled within the model and data interfaces for the model?

37

ML-specific concerns

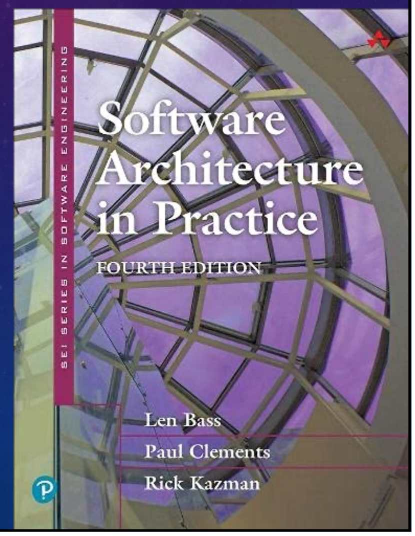
Non-architectural concerns

- Examples:
  - license plate information
  - facial images

38

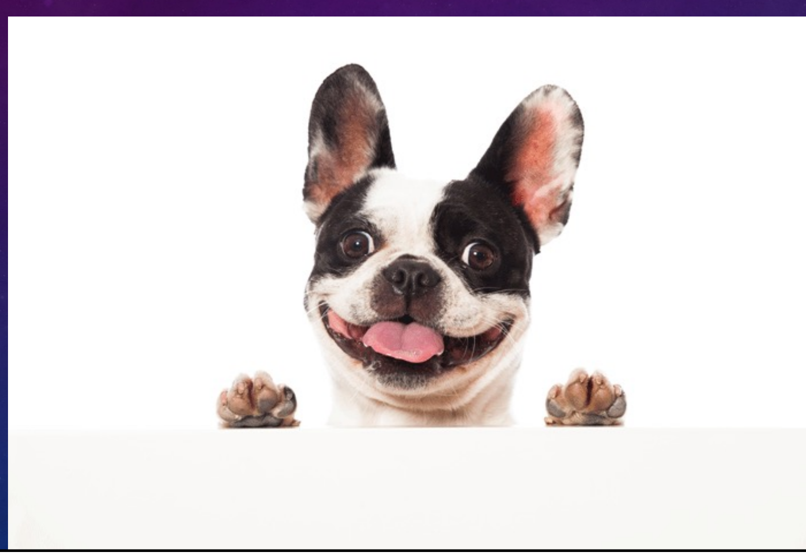
Final Thoughts

- Most of the concerns of ML-enabled systems are not unique!
- All of our existing patterns and tactics apply.
- This is good news!
- The biggest challenge is model modularity.
- We have early empirical validation.



39

Thank You!



40