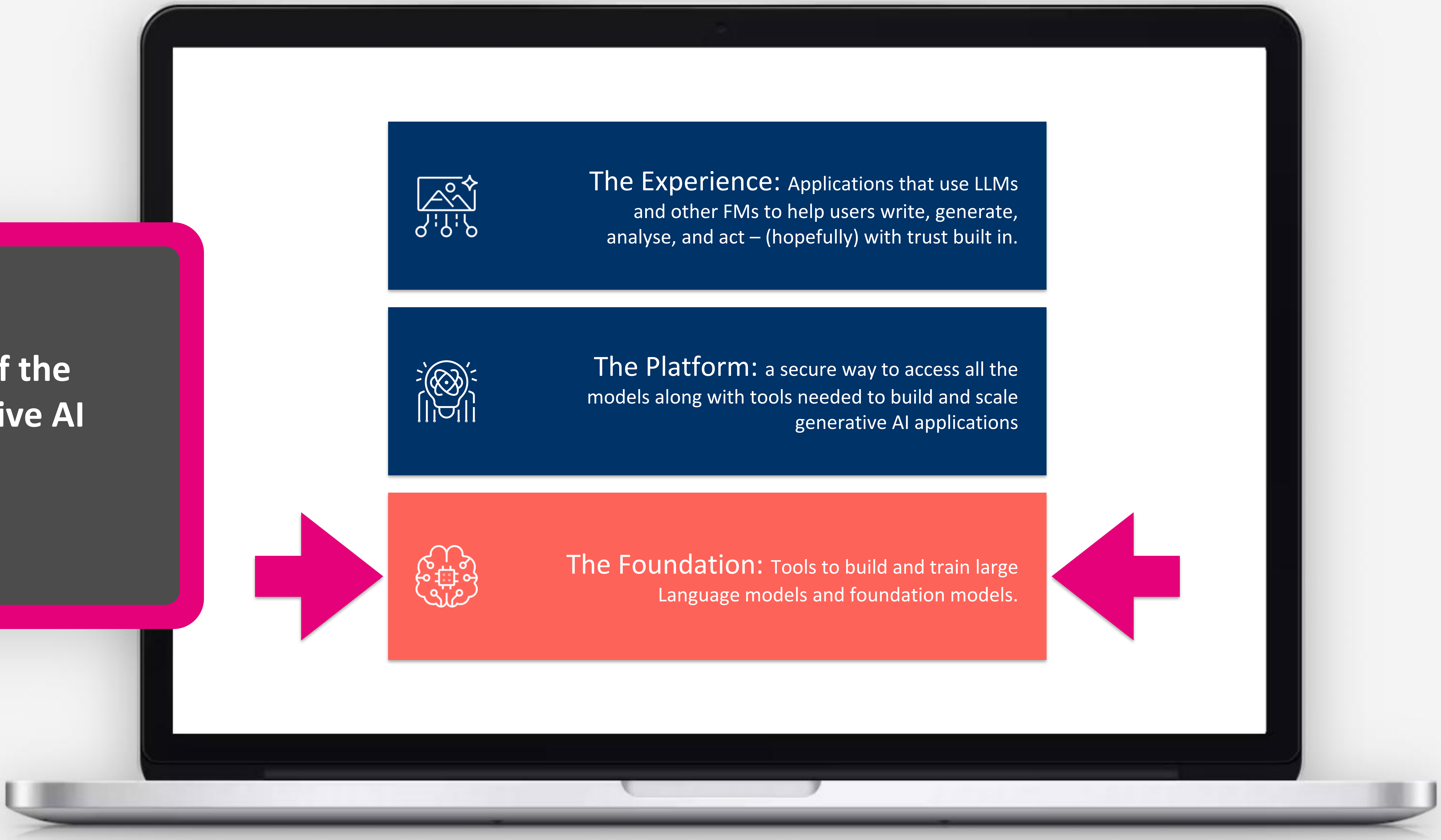
Vriti Magee

→

Embedding security into the AI/ML lifecycle — without slowing down innovation.

Demystifying MLSecOps

Layers of the Generative AI Stack

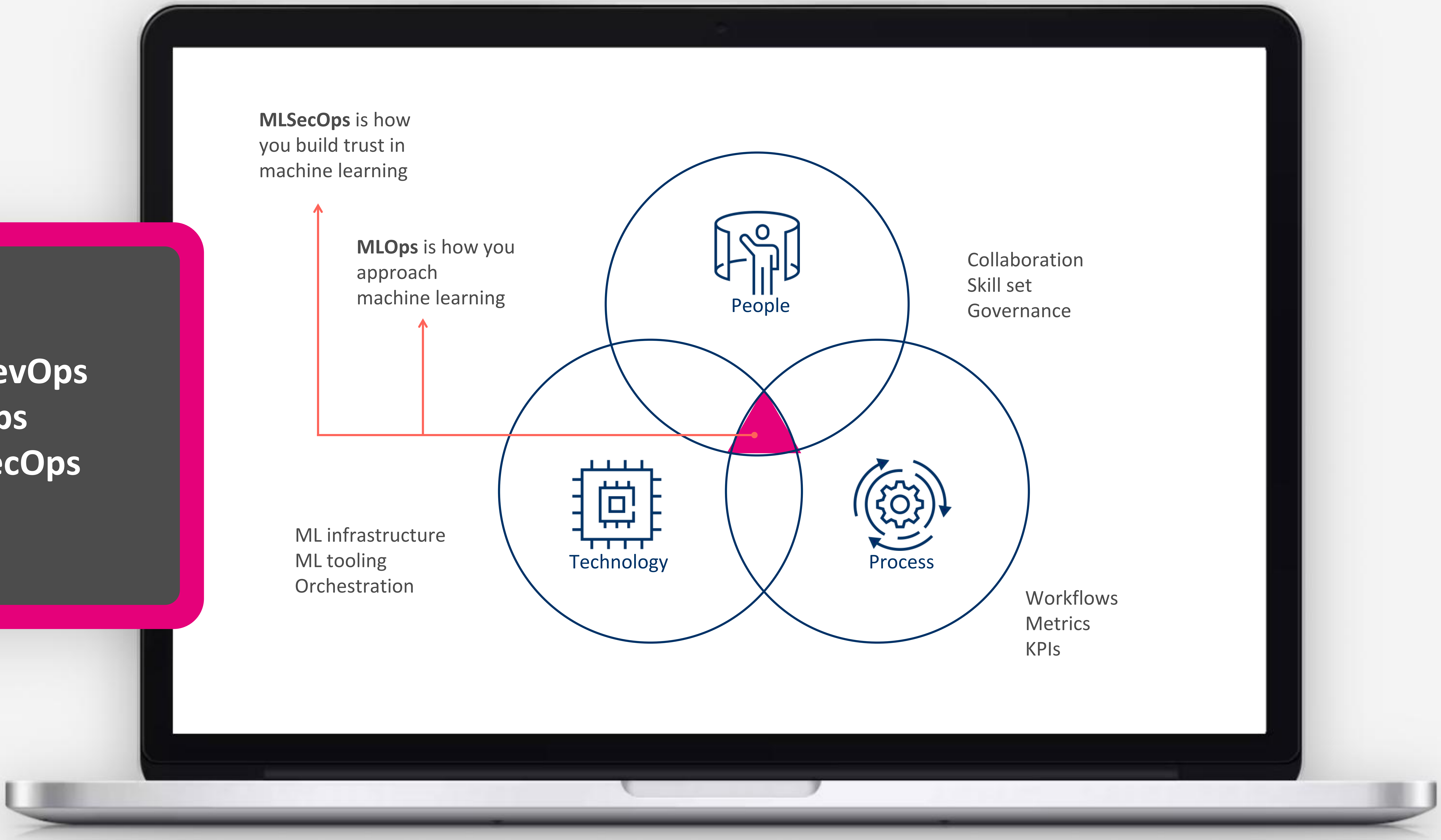


The Experience: Applications that use LLMs and other FMs to help users write, generate, analyse, and act – (hopefully) with trust built in.

The Platform: a secure way to access all the models along with tools needed to build and scale generative AI applications

The Foundation: Tools to build and train large Language models and foundation models.

From **DevOps**
to **MLOps**
to **MLSecOps**



MLSecOps is how you build trust in machine learning

MLOps is how you approach machine learning

People

Collaboration
Skill set
Governance

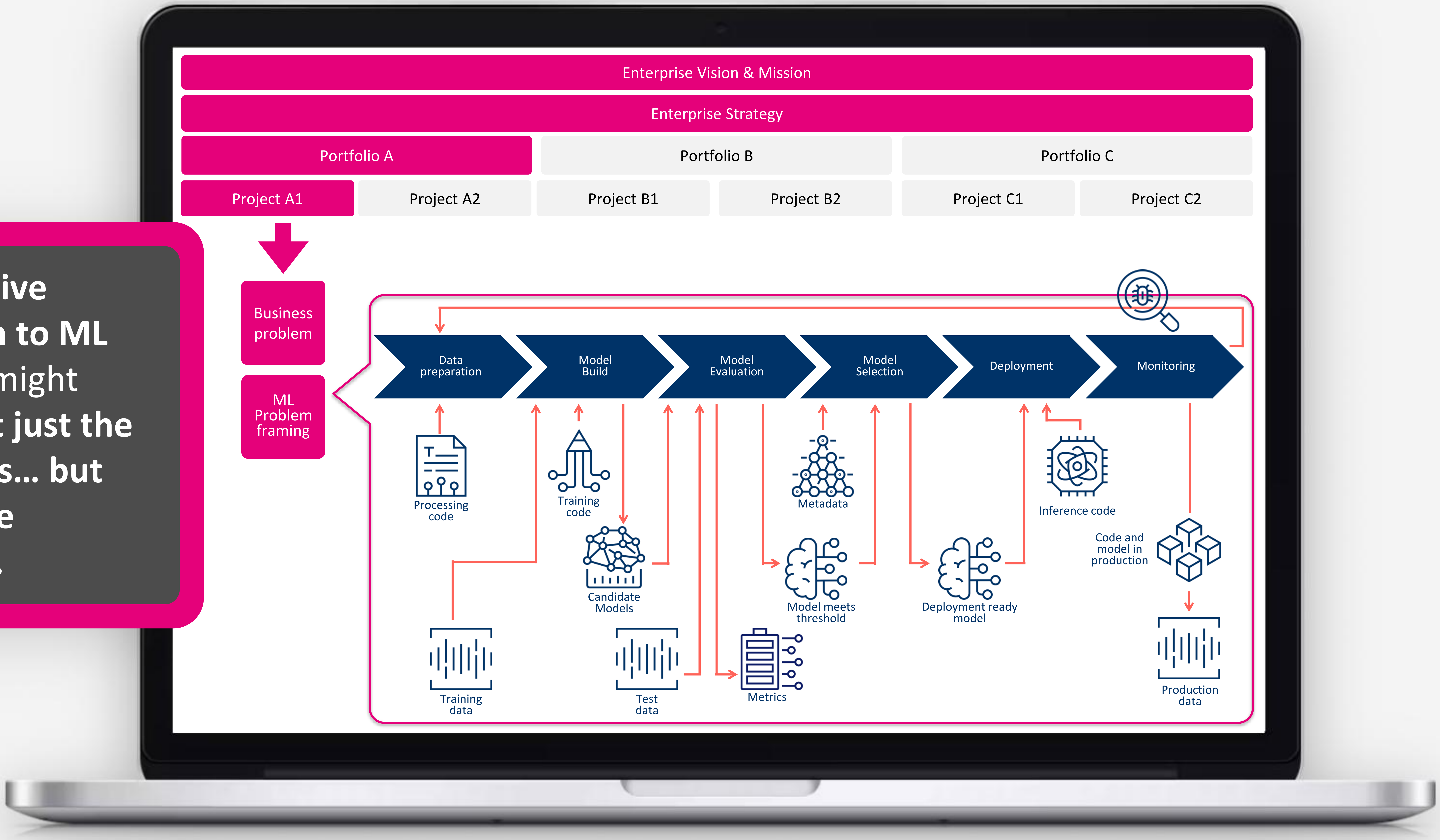
Technology

ML infrastructure
ML tooling
Orchestration

Process

Workflows
Metrics
KPIs

An **adaptive approach to ML** security might serve **not just the engineers… but the entire business.**



Enterprise Vision & Mission

Enterprise Strategy

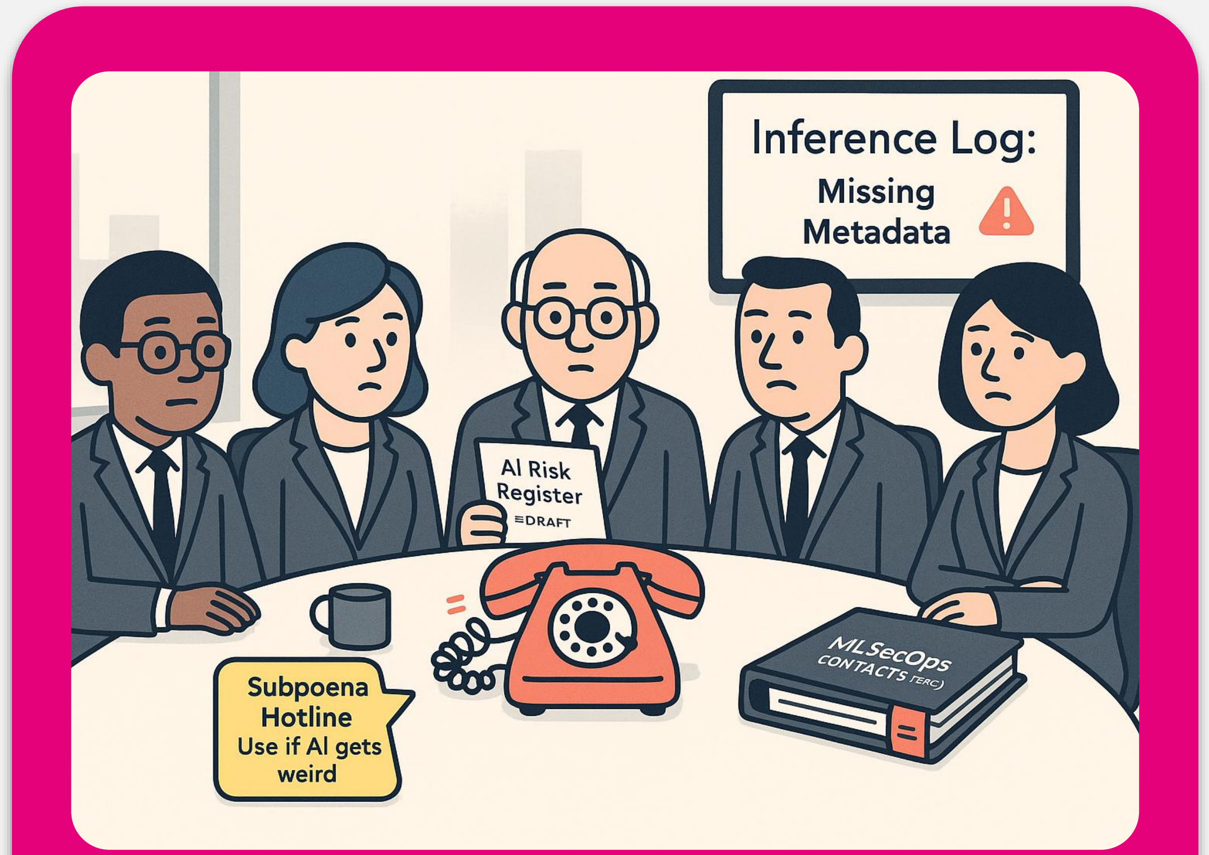
Portfolio A		Portfolio B		Portfolio C	
Project A1	Project A2	Project B1	Project B2	Project C1	Project C2

Business problem

ML Problem framing

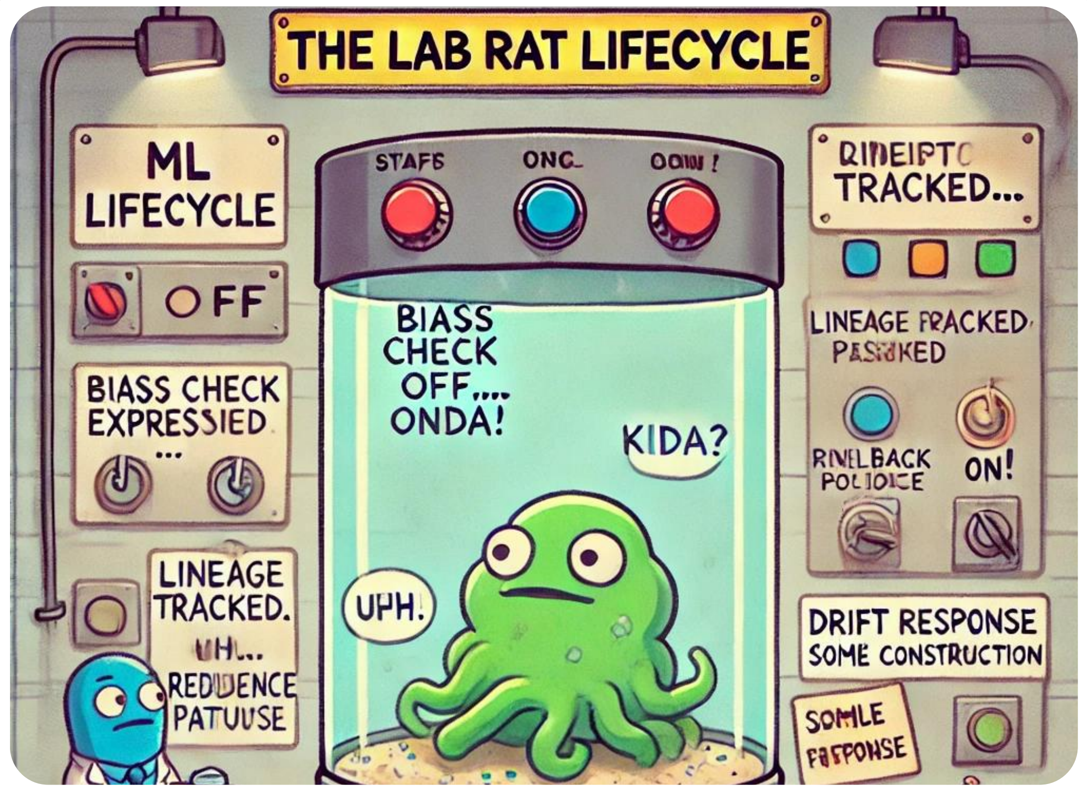
Data preparation → Model Build → Model Evaluation → Model Selection → Deployment → Monitoring

What Matters, Where



Why Boards Care (Even If They Don’t Say ‘MLSecOps’)

The ML Lifecycle, Revisited

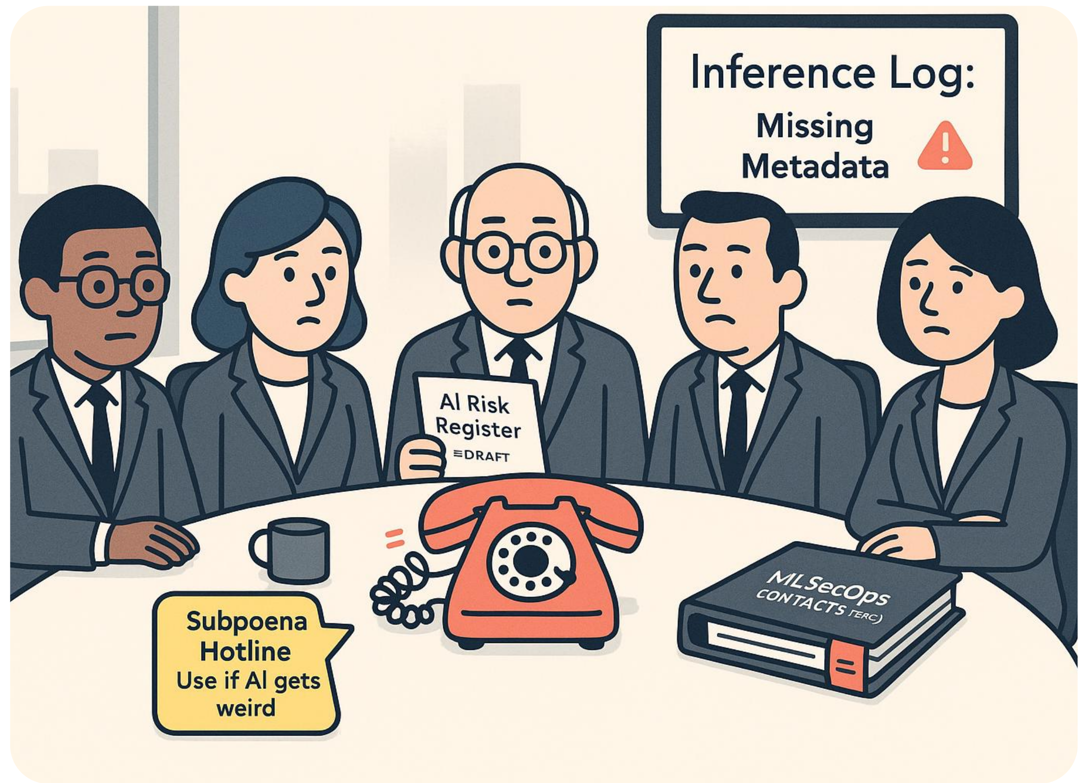


Designing Resilience In (Not Bolting It On)

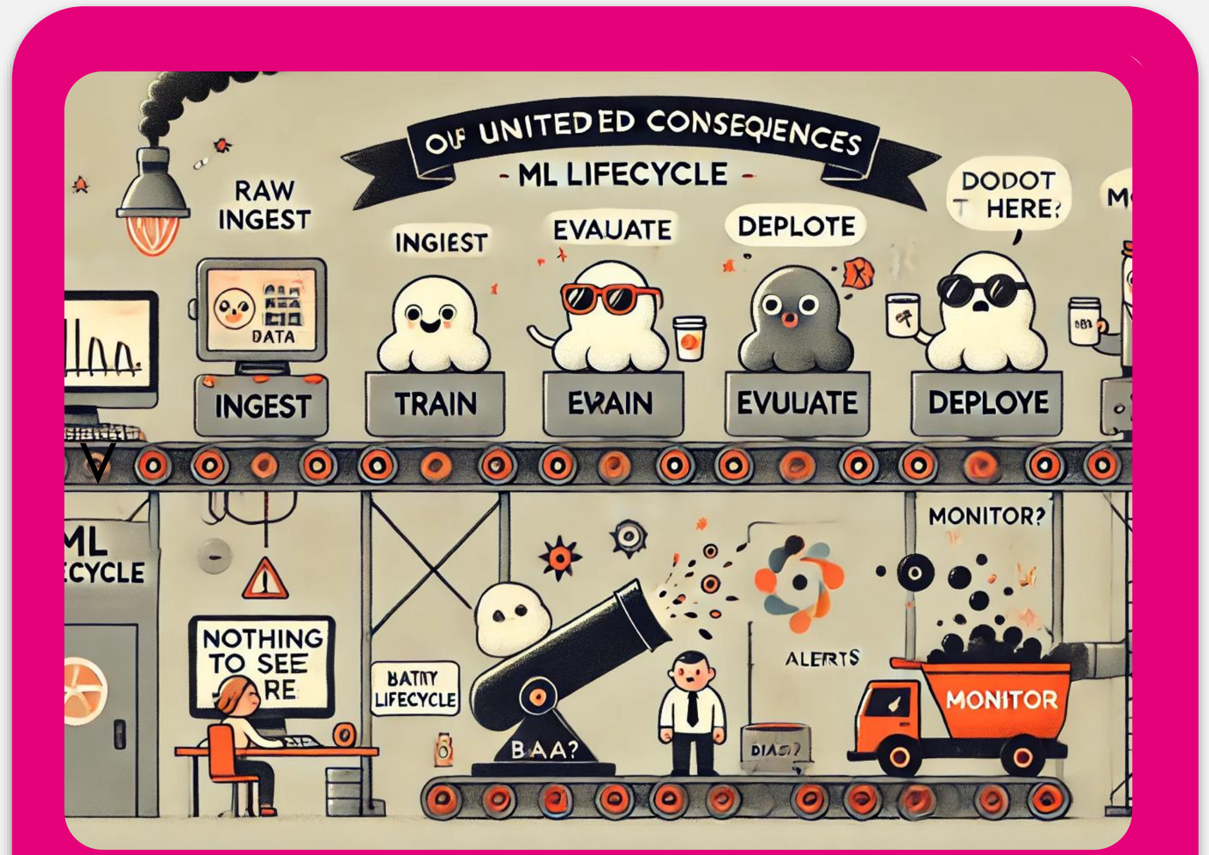
Why Boards Care (Even If They Don’t Say ‘MLSecOps’)

AI strategy is no longer just about models. It’s about governance, readiness, and trust.

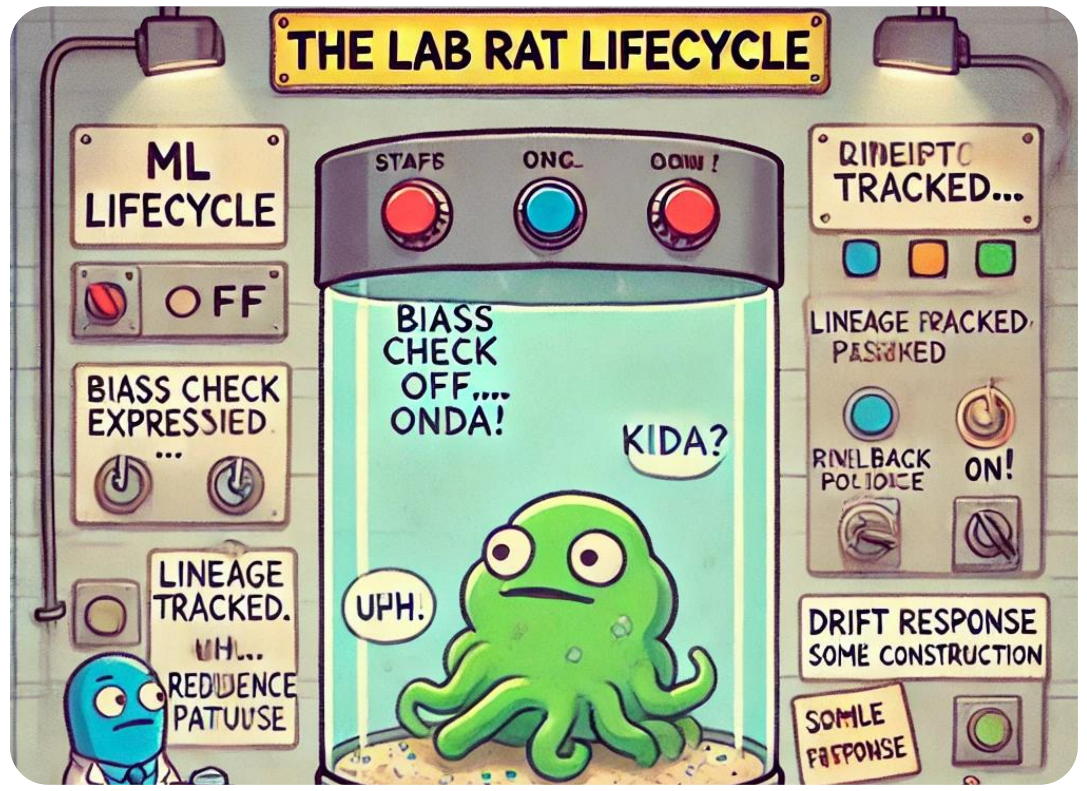
What Matters, Where



Why Boards Care (Even If They Don’t Say ‘MLSecOps’)



The ML Lifecycle, Revisited



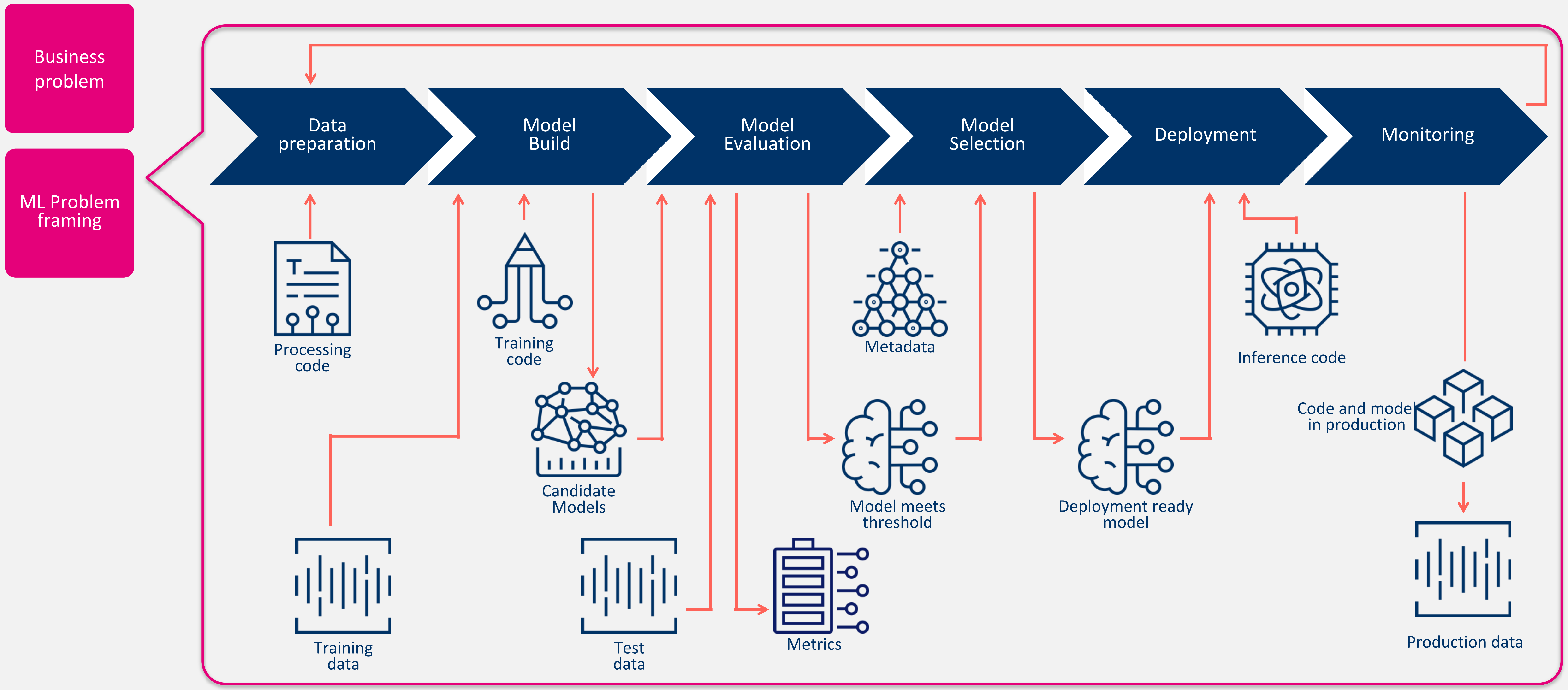
Designing Resilience In (Not Bolting It On)

The ML Lifecycle

Every model follows a lifecycle. But not every lifecycle is visible.

The ML Lifecycle

Every model follows a lifecycle. But not every lifecycle is visible.



Business problem

ML Problem framing

Data preparation

Model Build

Model Evaluation

Model Selection

Deployment

Monitoring

Processing code

Training code

Candidate Models

Training data

Test data

Metrics

Metadata

Model meets threshold

Deployment ready model

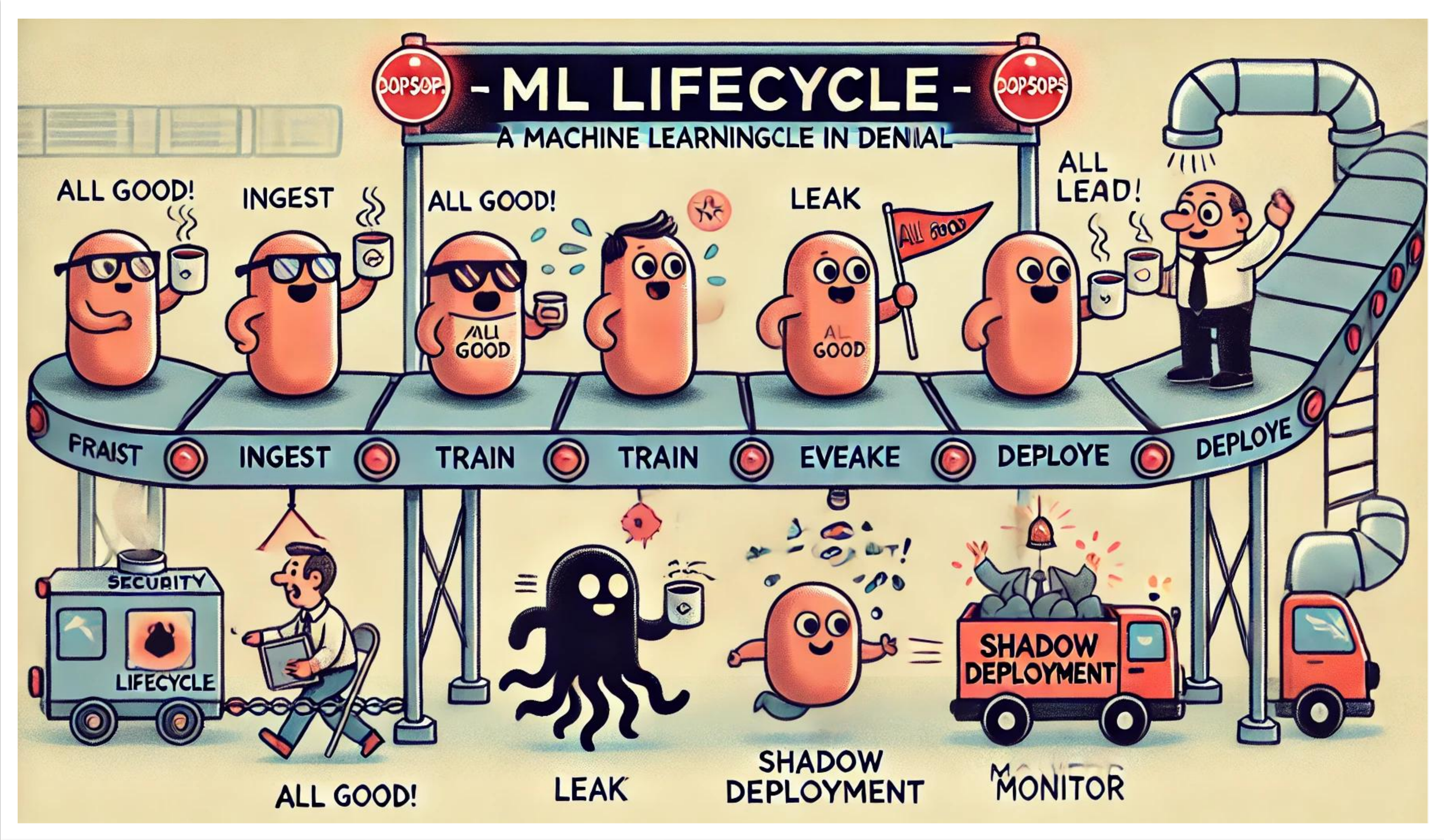
Inference code

Code and model in production

Production data

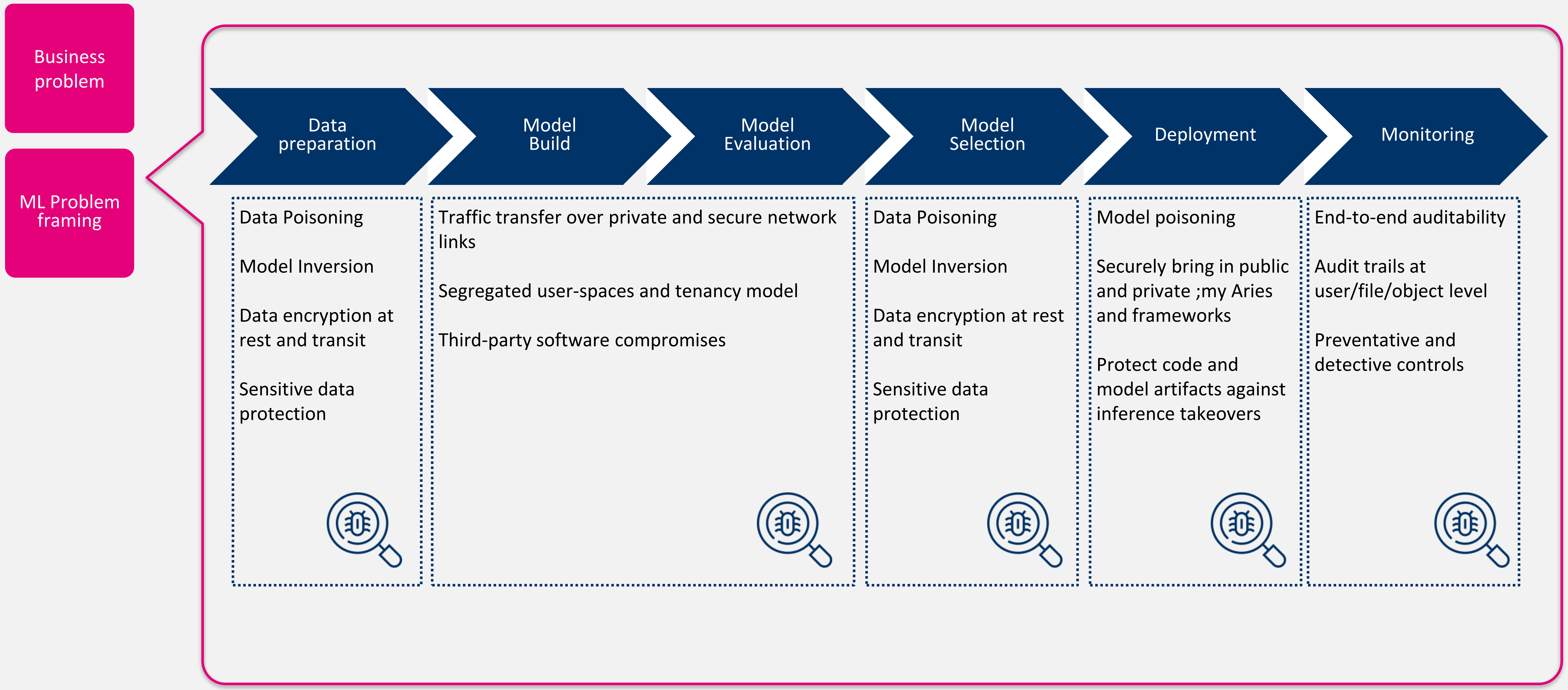
The ML Lifecycle, Where the Gaps Usually Start

Security often enters late. Risk doesn’t wait.



The ML Lifecycle, Where the Gaps Usually Start

Security often enters late. Risk doesn’t wait.



Business problem

ML Problem framing

Data preparation

Model Build

Model Evaluation

Model Selection

Deployment

Monitoring

Data Poisoning

Model Inversion

Data encryption at rest and transit

Sensitive data protection

Traffic transfer over private and secure network links

Segregated user-spaces and tenancy model

Third-party software compromises

Data Poisoning

Model Inversion

Data encryption at rest and transit

Sensitive data protection

Model poisoning

Securely bring in public and private ;my Aries and frameworks

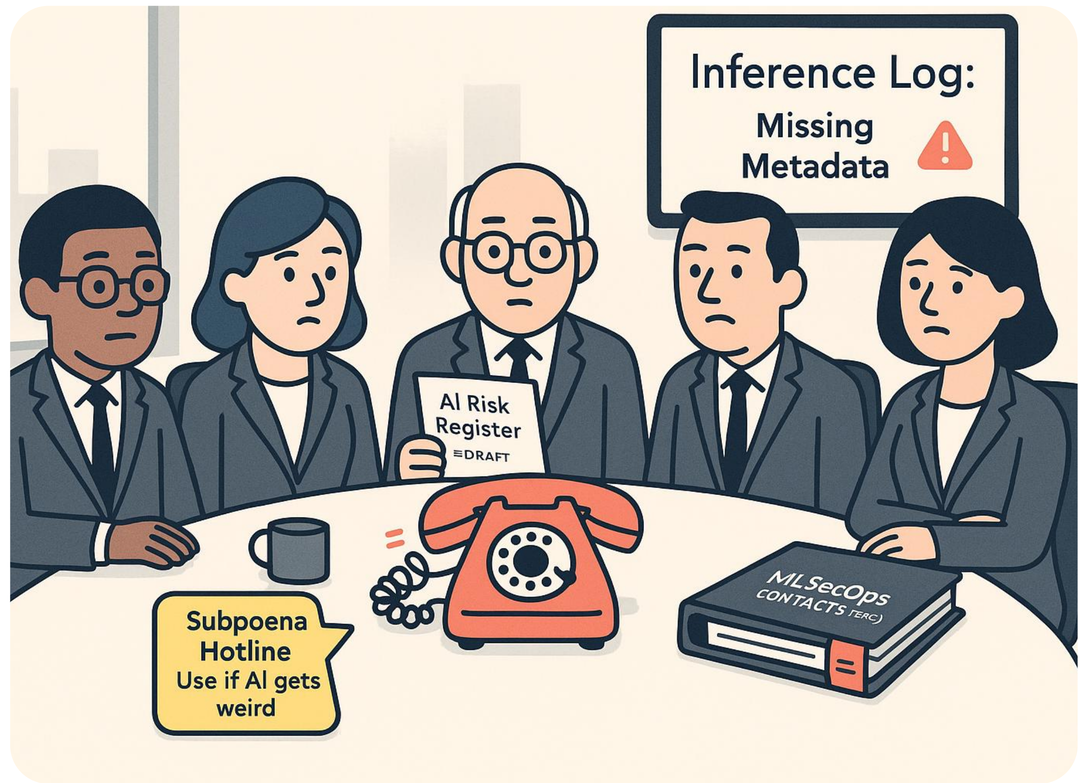
Protect code and model artifacts against inference takeovers

End-to-end auditability

Audit trails at user/file/object level

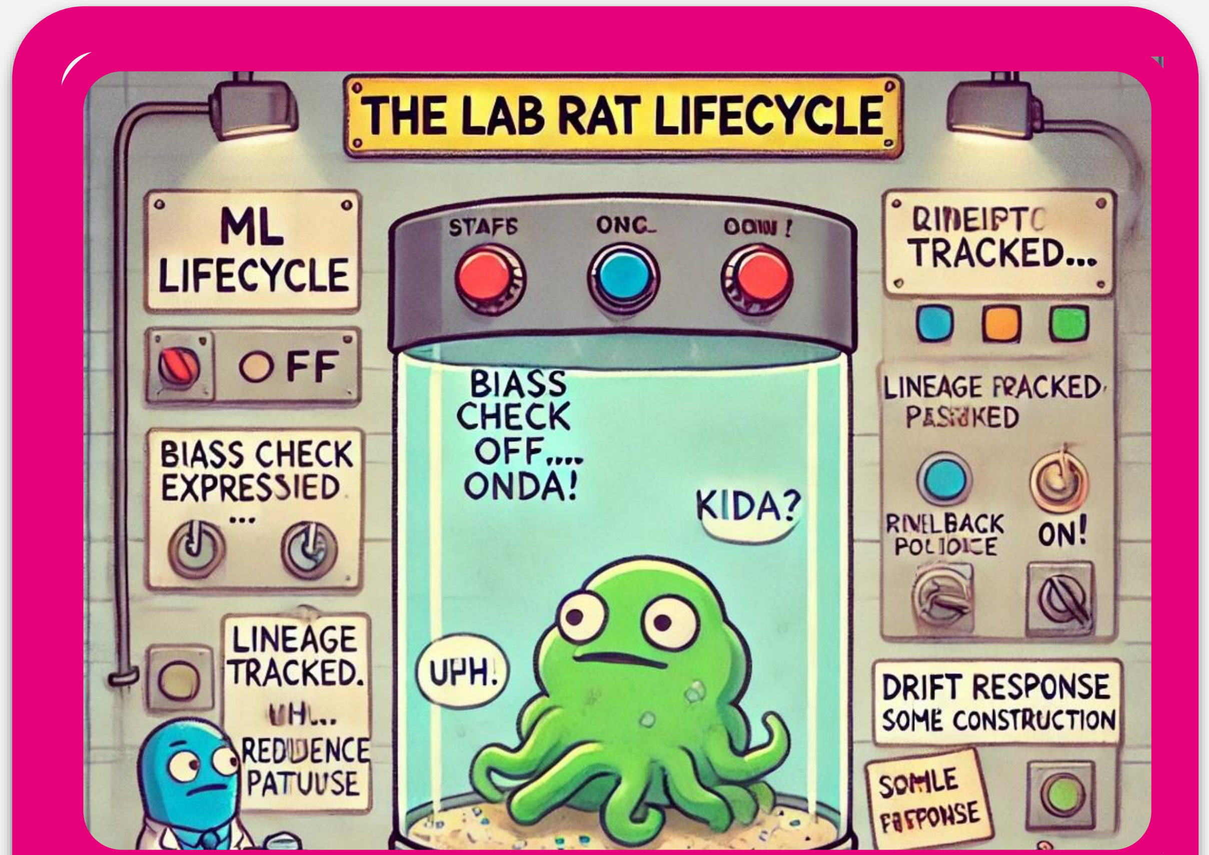
Preventative and detective controls

What Matters, Where



Why Boards Care (Even If They Don’t Say ‘MLSecOps’)

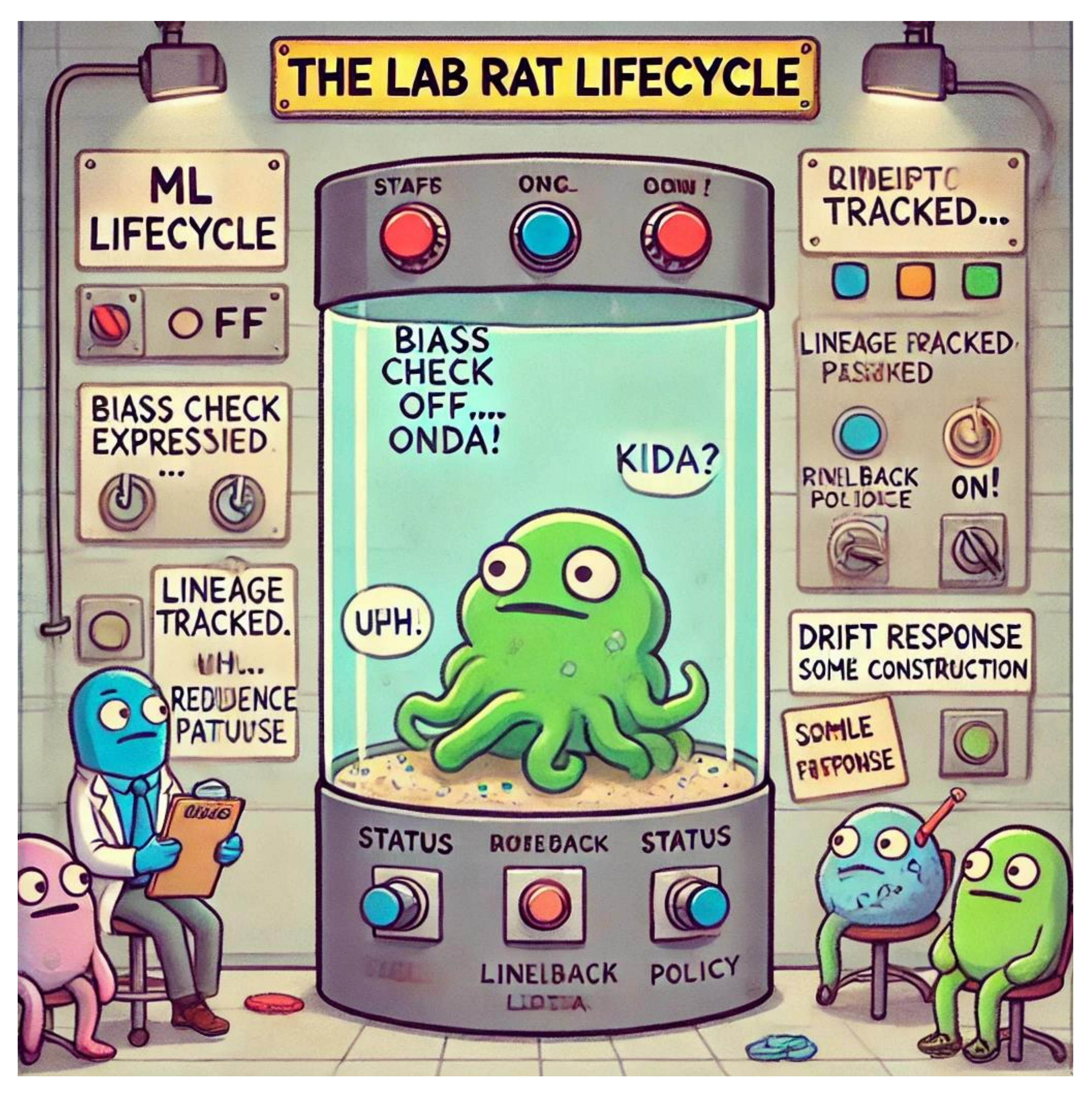
The ML Lifecycle, Revisited



Designing Resilience In (Not Bolting It On)

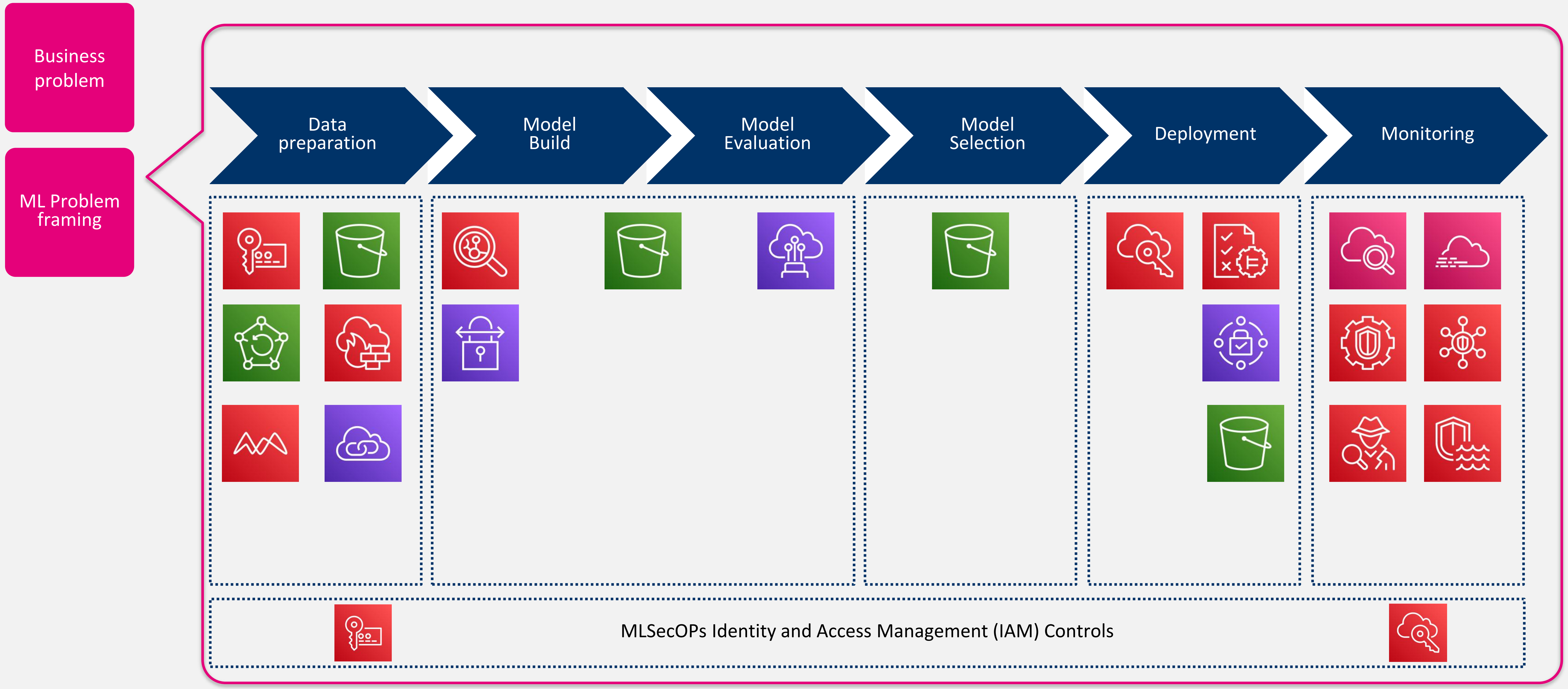
Designing Resilience In, Not Bolting It On

MLSecOps is how we express trust across the lifecycle — not once, but continuously.



Designing Resilience In, Not Bolting It On

MLSecOps is how we express trust across the lifecycle — not once, but continuously.



Business problem

ML Problem framing

Data preparation

Model Build

Model Evaluation

Model Selection

Deployment

Monitoring

MLSecOPs Identity and Access Management (IAM) Controls

Designing Resilience In, Where the Friction Really Lives

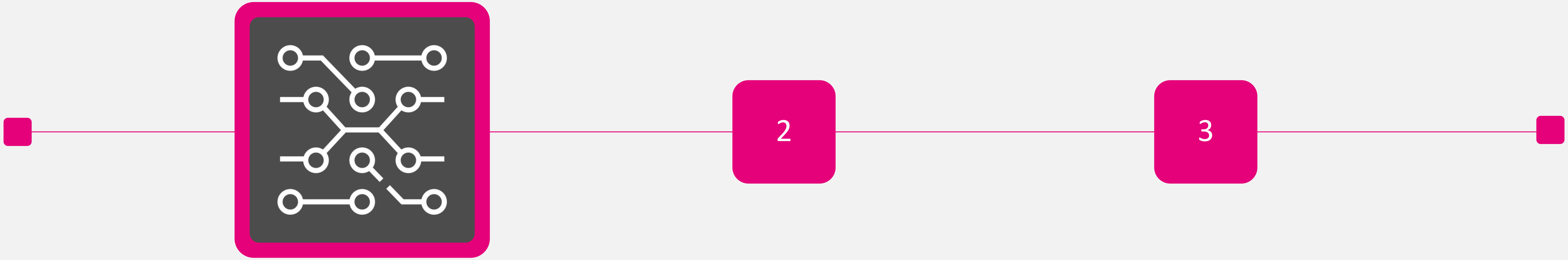
It’s not a tooling problem. It’s a coordination problem.

MODEL OPS PLAN
~~D~~RIFT?
NOT MY API
IT WORKED IN MY NOTEBOOK
WHOSE MODEL IS THIS?
ROLLBACK PLAN?

Ideas To Get You Started

1. Map your real lifecycle
2. Assign shared ownership
3. Make model behaviour explainable

Ideas To Get You Started



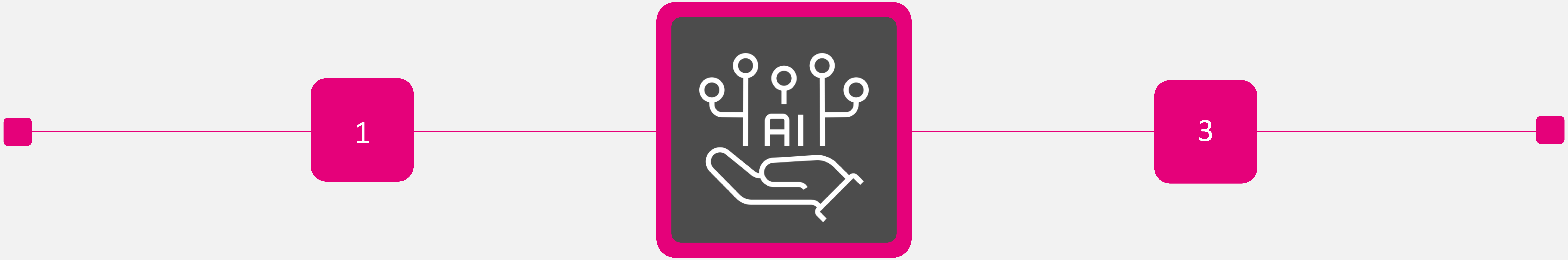
2

3

Map your real lifecycle

Not what’s in the docs — what’s actually happening.

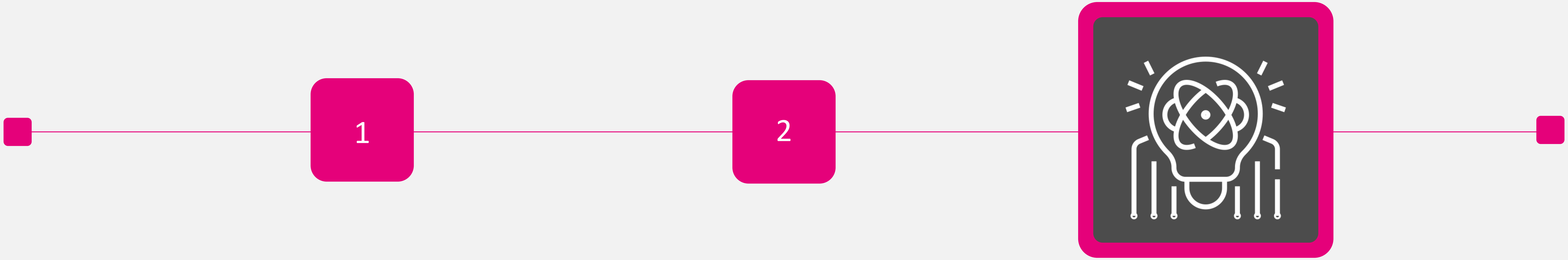
Ideas To Get You Started



Assign shared ownership

Clearly define who owns what

Ideas To Get You Started



1

2

Make model behaviour explainable

You need just enough
traceability to answer:
“Why did this model do that?”

Ideas To Get You Started

1

Map your real lifecycle

2

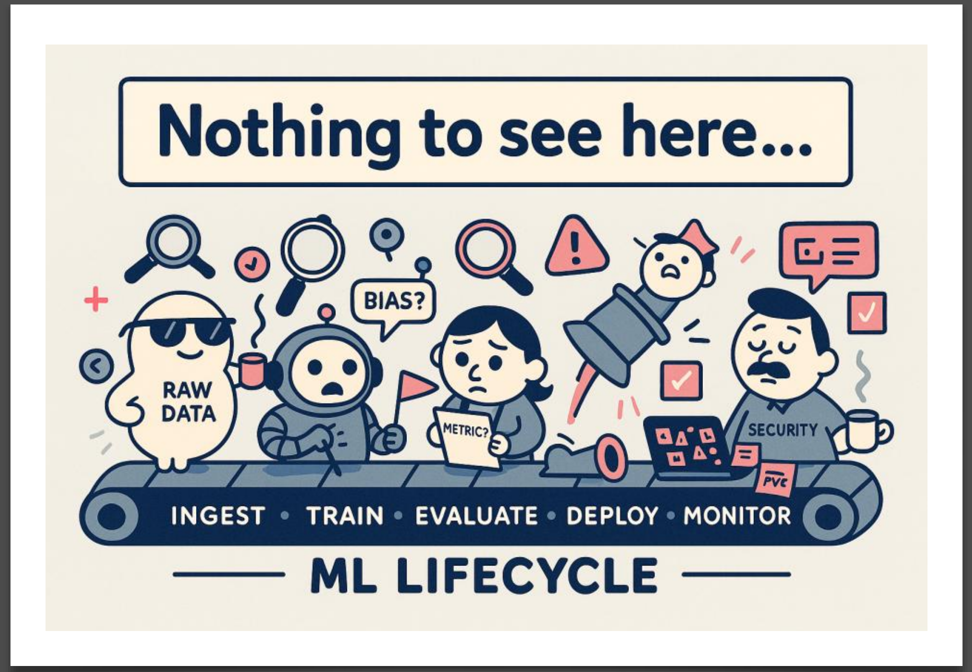
Assign shared ownership

3

Make model behaviour explainable

What Storage Knows
(That You Might Not)

Vriti Magee
www.linkedin.com/in/vriti



Thank you