

Let's talk about MLOps

Christian Barra // PyCon SK 2019

@christianbarra

A couple of questions.....

@christianbarra

A couple of questions.....

@christianbarra

"You are not paid to program, you're not even paid to maintain someone else's program, you're paid to deliver solutions to the business."

David Cheney

@christianbarra



@christianbarra

1. Let's talk about DevOps

@christianbarra

It's a culture

@christianbarra

and a set of practices

@christianbarra

and capabilities.

@christianbarra

"... to shorten the systems development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives."

Wikipedia

@christianbarra

- **CI/CD**
- **Version control**
- **Monitoring**
- **Learning organization**
- **Logging**
- **ADR (Architecture Design Review)**
- **Runbooks**
- **Lean development**
- **Agile stuff**
- **Documentation**
- **Testing**
- **Alerting**
- **Metrics**
- **SLA**
- **Deploy strategy**
- **....**

It's not a set of tools

- **It's not just CI/CD**
- **It's not Jenkins**
- **It's not Airflow**
- **It's not K8S**
- **It's not the morning standup**
- **...**

@christianbarra

It's not a ROLE.

IMHO.

@christianbarra

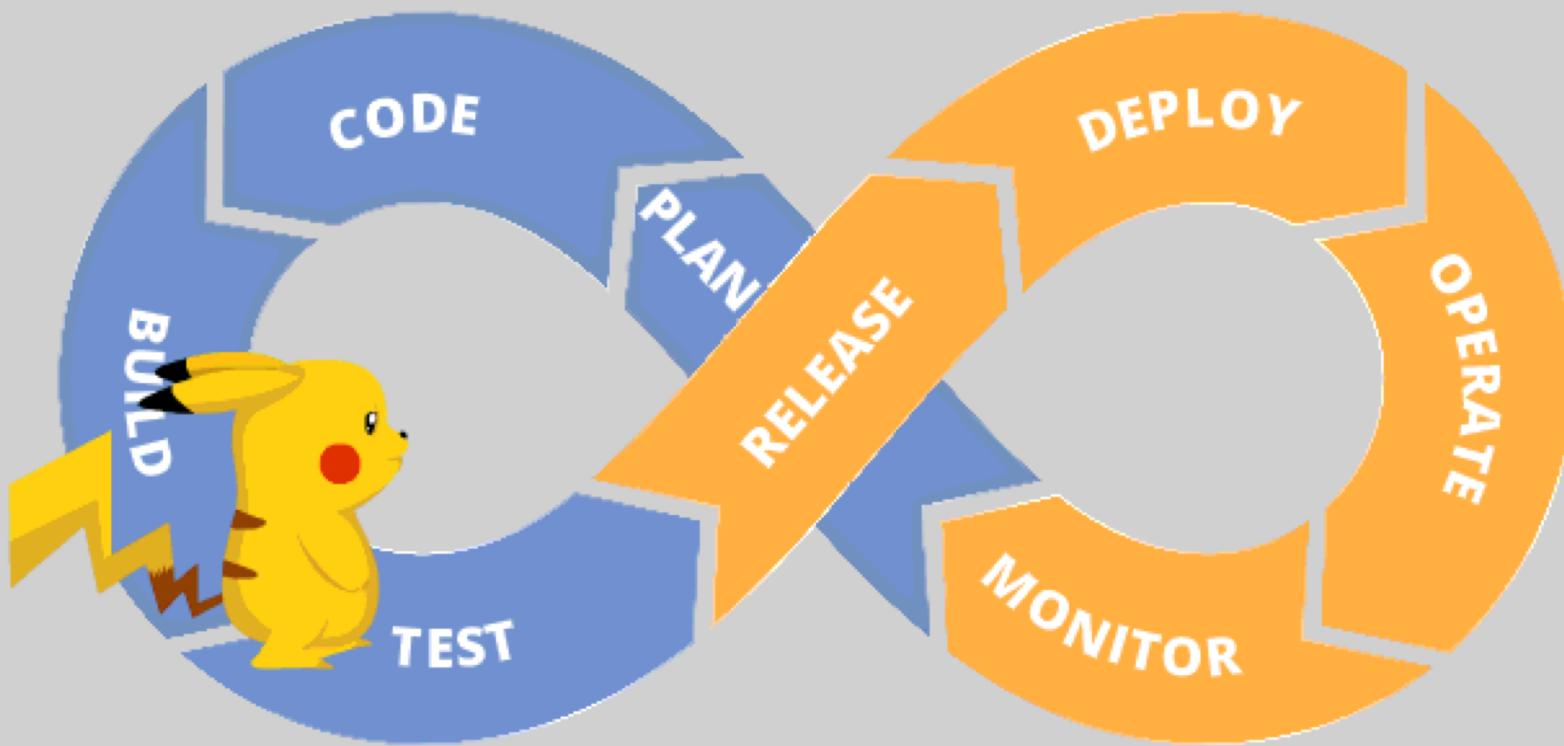
**The true essence of
DevOps is empathy.**

Jeff Sussna

@christianbarra

Let's talk about *Pokemon.*

@christianbarra



Pokemon Go - Riding on DevOps principles

**It's a never ending process of
continuous improvement....**

@christianbarra

**using the feedback
that you get...**

@christianbarra

**when you have
end to end ownership....**

@christianbarra

of your systems.

@christianbarra

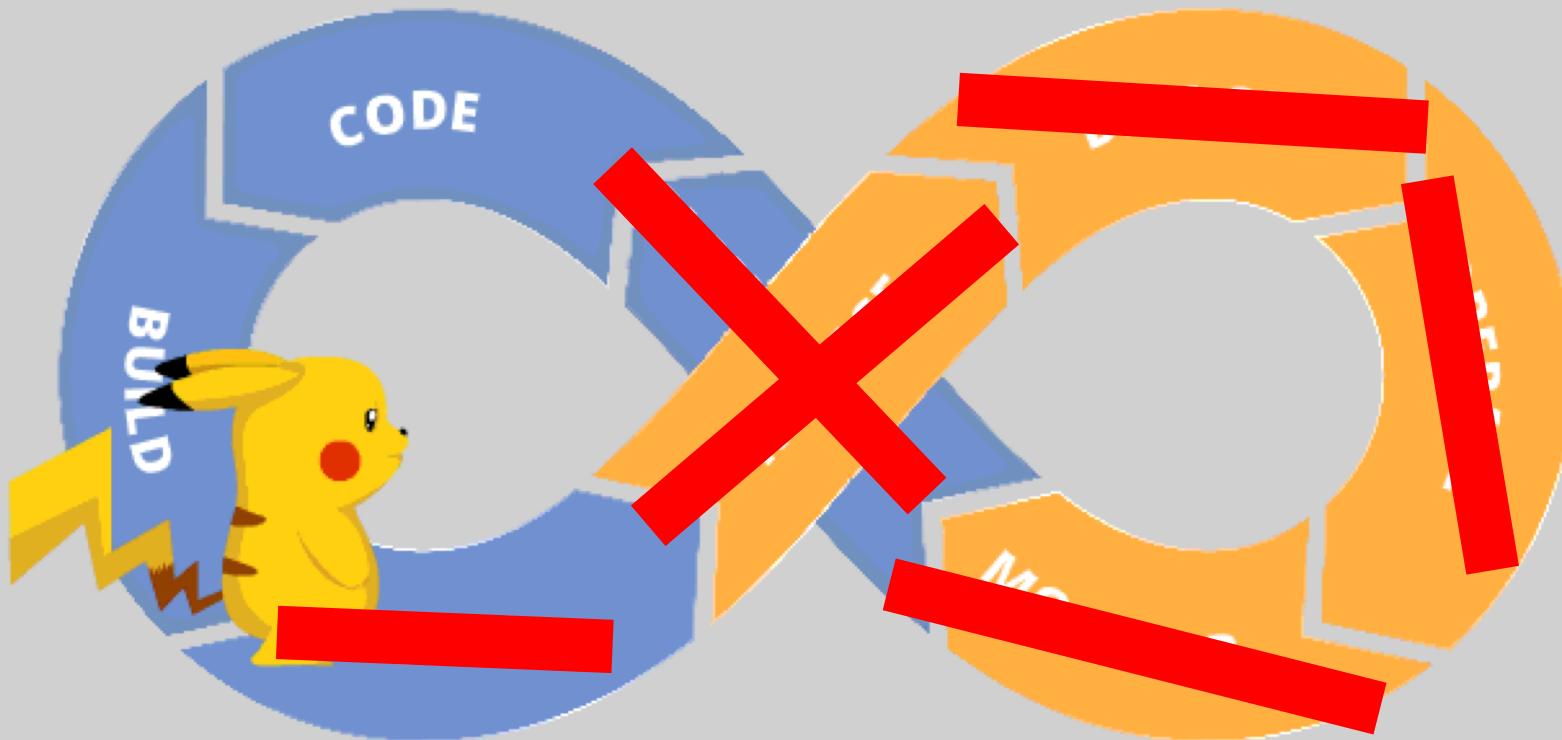
You build it, you ship it.

And keep it running.

@christianbarra

2. What's the problem with DS/ML?

@christianbarra



Pokemon Go - Riding on DevOps principles

How did it happen?

@christianbarra

No idea (🤔).

But let's focus on the solutions.

@christianbarra

3. MLOps

@christianbarra

Machine Learning Operations

@christianbarra

“[...] applies to the entire lifecycle – from integrating with *model generation* (software development lifecycle, CI/CD), orchestration, and deployment, to health, diagnostics, governance, and business metrics.”

Wikipedia

@christianbarra

WAT???

@christianbarra

Isn't DevOps 🤔 ?

@christianbarra

Or maybe not 🤔 ?

@christianbarra

You code.

@christianbarra

You train the model.

@christianbarra

You *ship* the model.

@christianbarra

You *operate* the model.

@christianbarra

They don't look so different 🤔

@christianbarra

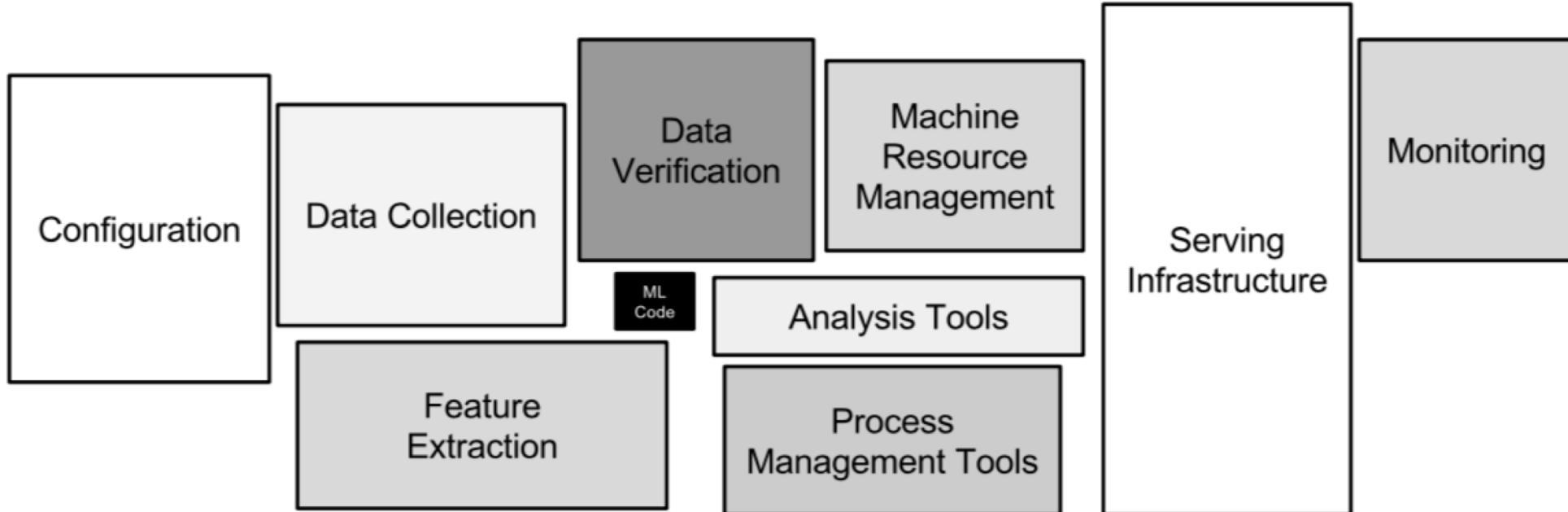


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

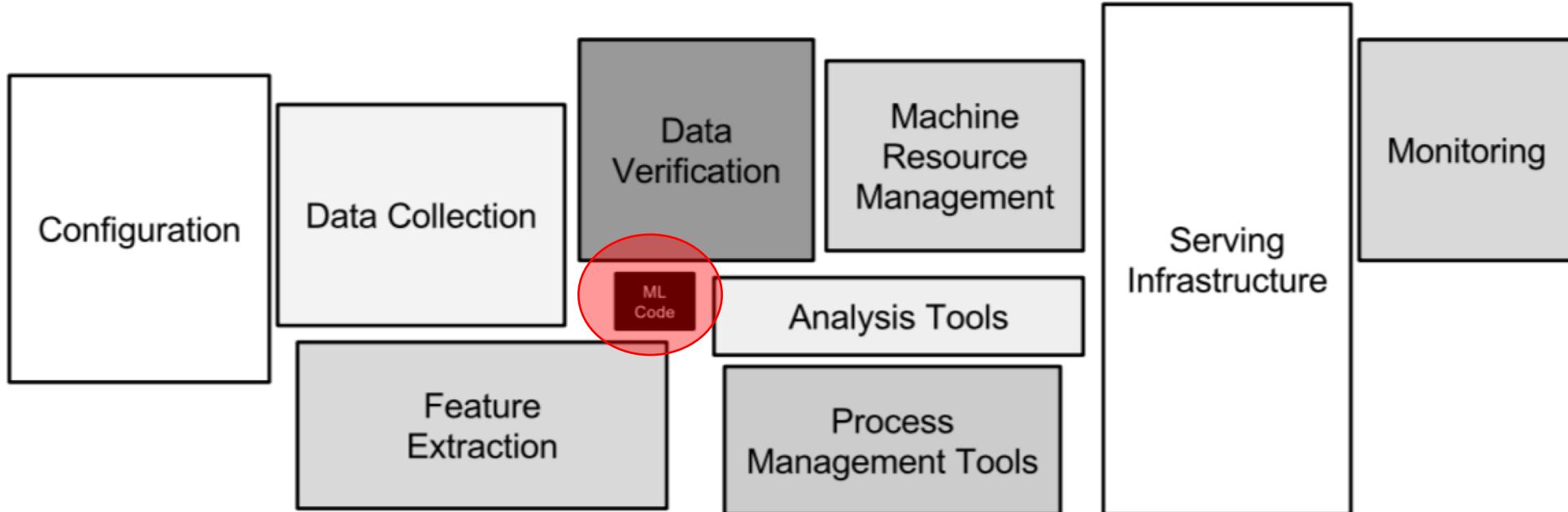


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

**Don't try reinvent every
software best practice from
scratch.**

Again, IMHO.

@christianbarra

4. Production Readiness

@christianbarra

**A measure of your capability of
shipping business features in a reliable
way, constantly and without
compromising quality.**

@christianbarra

A measure of your capability of shipping business features in a *reliable* way, *constantly* and *without compromising quality.*

@christianbarra

Production readiness is a spectrum.

@christianbarra

Things that contribute to production readiness

- **Version control**
- **Business requirements are satisfied**
- **CI/CD**
- **Observability (monitoring, logging, tracing, ...)**
- **Runbooks**
- **Orchestration system**
- **1 click deployment/rollback**
- **Canary deployments**
- **Testing spectrum**
- **Documentation**
- **Health checks**
- **Many more stuff...**

5. [WIP] Production readiness checklist for ML

@christianbarra

What's the goal of having a checklist?

@christianbarra

Increase quality by setting clear requirements/processes.

@christianbarra

**You do not release
until everything is**

Ideally.....

@christianbarra

1. You use version control.

@christianbarra

**2. (Almost) everything is
automated.**

@christianbarra

3. You can introspect your system

@christianbarra

4. You have metrics about your system

@christianbarra

5. You release often and don't have a release day.

Every day (apart from Friday) is a good day!

@christianbarra

6. Your models are services.

@christianbarra

7. You can run different versions of your models at the same time.

Which means you can compare them with real data and rollback easily.

@christianbarra

7. You defined what reproducibility is

@christianbarra

**These are all things that you can
learn.**

**It's much more easier through osmosis and
cross pollination.**

@christianbarra

**Which means putting DS and
SW closer.**

Remember? empathy!

@christianbarra

**And PLS stop putting random
words together....**

@christianbarra



DevOps Production Readiness Engineer

Zurich, Switzerland

To sum it up...

The market leader in developing distributed ledger technology for settlements in the financial markets.

Role overview

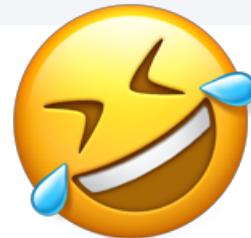
Distributed Ledger Technology is emerging within the financial markets, offering the chance for outdated, archaic systems to be modernized for the transaction driven global electronic markets





DevOps Production Readiness Engineer

Zurich, Switzerland



To sum it up...

The market leader in developing distributed ledger technology for settlements in the financial markets.

Role overview

Distributed Ledger Technology is emerging within the financial markets, offering the chance for outdated, archaic systems to be modernized for the transaction driven global electronic markets



Thanks

@christianbarra