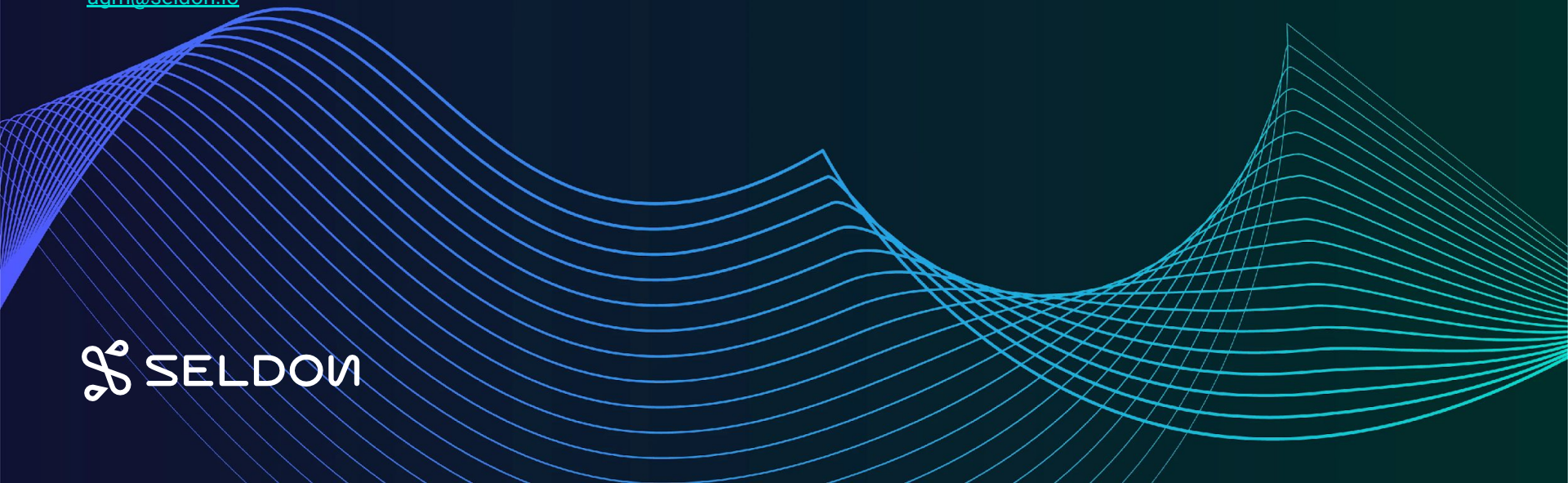
# Poisoned Pickles Make You ill

Adrian Gonzalez-Martin

Head of ML Serving

agm@seldon.io



SELDON

# About me


Adrian Gonzalez-Martin

**Head of ML Serving**, Seldon

**Fellow**, The Institute for Ethical AI & ML

@adriangonz@mastodon.social

github.com/adriangonz

# Security Challenges in Machine Learning

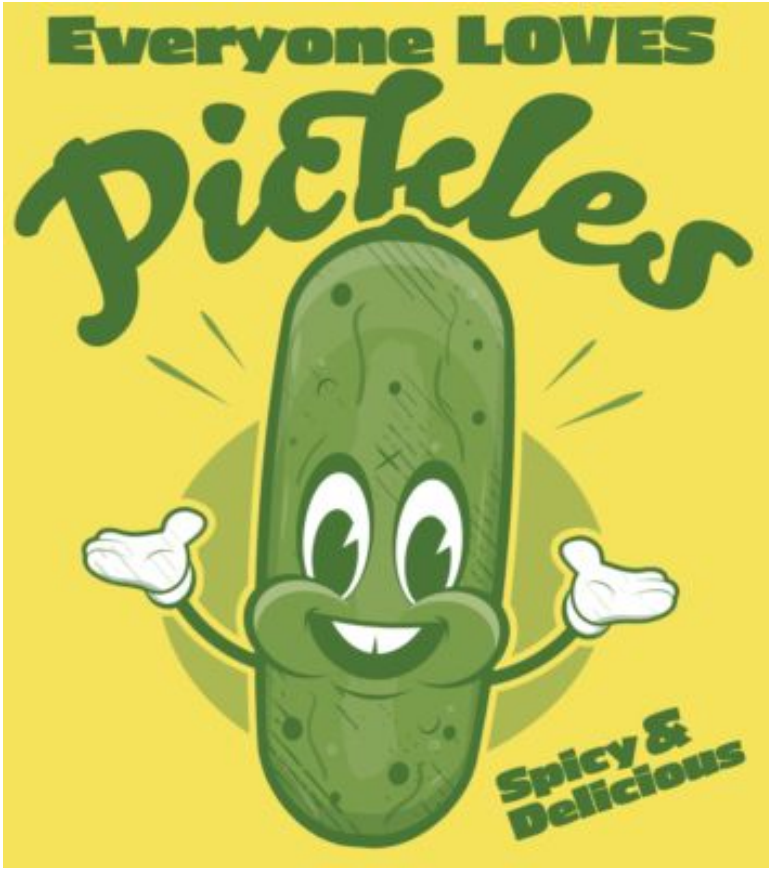
- ➔ Security challenges in devops and software space more known
- ➔ Key challenges in machine learning still being defined as best practice
- ➔ Impossible to make systems **unhackable**, but possible to **mitigate undesired outcomes**

**The solutions will be technical in nature but will ultimately still rely on humans and process.**

# Everyone loves Pickles!

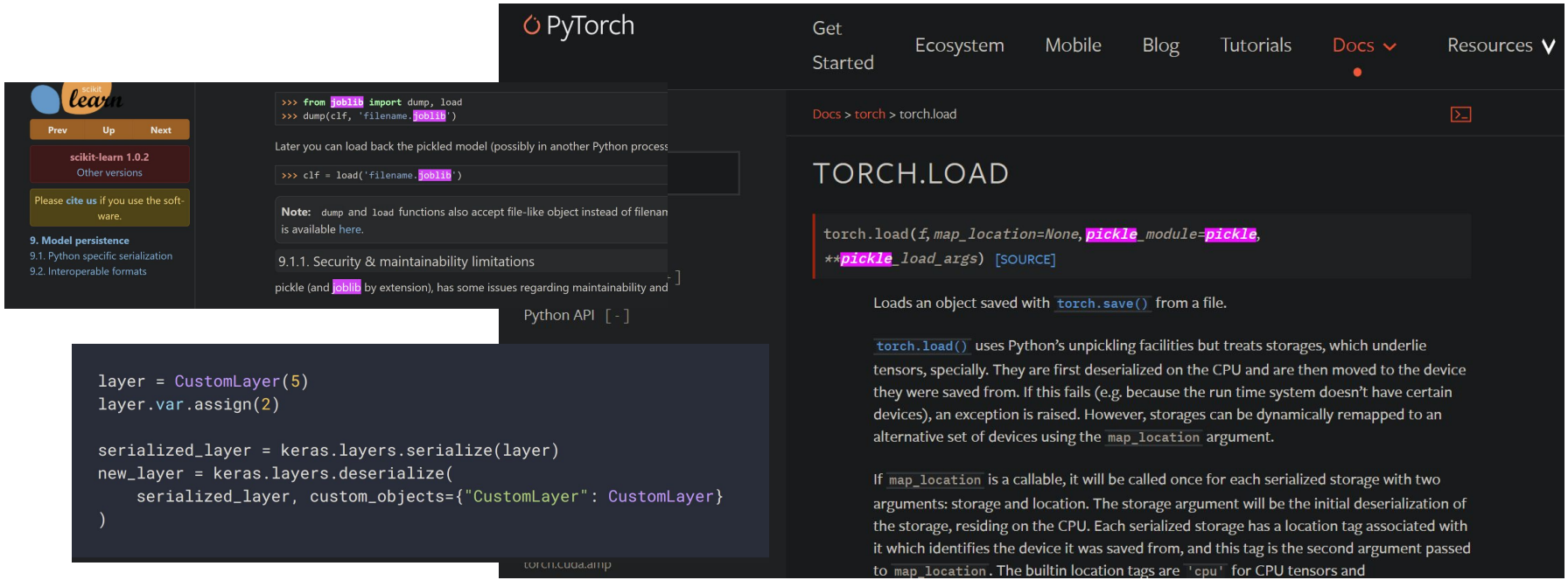
# Everyone loves pickles!

- ➔ Pickles are Python's native **serialisation format**
- ➔ **Any Python object** can be serialised into a Pickle
  - ◆ Including functions and **arbitrary Python code**
- ➔ Hugely popular in the ML space



# Pickles are everywhere

➔ Major ML libraries rely on Pickles (i.e. they are exposed to Pickle's issues)



# Security Risks with Pickles

By design, Pickles can serialise **any Python object**

Dump model binary with pickle

```
[7]: !mkdir -p fml-artifacts/safe/

[8]: import joblib

     joblib.dump(model, "fml-artifacts/safe/model.joblib")

[8]: ['fml-artifacts/safe/model.joblib']

[9]: with open("fml-artifacts/safe/model.joblib", "rb") as f: print(f.readlines())

     [b'\x80\x04\x95\xe5\x01\x00\x00\x00\x00\x00\x00\x8c\x1esklearn.linear_model._logistic\x94\x8c\x12LogisticRegression\x94\x93\x94)\x81\x94}\x94
     (\x8c\x07penalty\x94\x8c\x02l2\x94\x8c\x04dual\x94\x89\x8c\x03tol\x94G?\x1a6\xe2\xeb\x1cC-\x8c\x01C\x94G?\xf0\x00\x00\x00\x00\x00\x00\x8c\rfit
     _intercept\x94\x88\x8c\x11intercept_scaling\x94K\x01\x8c\x0cclass_weight\x94N\x8c\x0crandom_state\x94N\x8c\x06solver\x94\x8c\tliblinear\x94\x8
     c\x08max_iter\x94Kd\x8c\x0bmulti_class\x94\x8c\x03ovr\x94\x8c\x07verbose\x94K\x00\x8c\n', b'warm_start\x94\x89\x8c\x06n_jobs\x94N\x8c\x08l1_ra
     tio\x94N\x8c\x0en_features_in_\x94K\x04\x8c\x08classes_\x94\x8c\x13joblib.numpy_pickle\x94\x8c\x11NumpyArrayWrapper\x94\x93\x94)\x81\x94}\x94
     (\x8c\x08subclass\x94\x8c\x05numpy\x94\x8c\x07ndarray\x94\x93\x94\x8c\x05shape\x94K\x03\x85\x94\x8c\x05order\x94h\t\x8c\x05dtype\x94h\x1fh%\x93
     \x94\x8c\x02i8\x94\x89\x88\x87\x94R\x94(K\x03\x8c\x01<\x94NNNJ\xff\xff\xff\xffJ\xff\xff\xff\xffK\x00t\x94b\x8c\n', b'allow_mmap\x94\x88ub\x0
     0\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x02\x00\x00\x00\x00\x00\x00\x00\x95M\x00\x00\x00\x00\x00\x00\x00\x8c\x05coef_\x9
     4h\x1b)\x81\x94}\x94(h\x1eh!h"K\x03K\x04\x86\x94h$\x8c\x01F\x94h%h&\x8c\x02f8\x94\x89\x88\x87\x94R\x94(K\x03h*NNNJ\xff\xff\xff\xffJ\xff\xff\xf
     f\xffK\x00t\x94bh,\x88ub, ?T\xff@\xda?\xf6_5nM\\\xdb?.z\xa2\x86\xfbQ\xfb\xbf\x0bh|N5m\xf7?w\xfa$3:\xcb\xf9\xbf\xbc\x99m\xbff\x8c\xf8\xbf{\xc8\
     x01\x01\x8c\x14\x02\xc0l\xcb\xc4e\x18m\xe2?\xb3s\x82\xa2\x8a\xc4\x03@`\xf08\xe4(V\xf0\xbf"\\}\x85\xaf\x7f\xf6\xbf\x03M#\n', b'fq\x04@\x95,\x0
     0\x00\x00\x00\x00\x00\x00\x00\x8c\n', b'intercept_\x94h\x1b)\x81\x94}\x94(h\x1eh!h"K\x03\x85\x94h$h\th%h4h,\x88ub\xb5~?\xd6\xf4\xe8\xd0?\x8d\xd5\x
     fc\'\xb7\x80\xf1??\xc3\xdc\xe0ro\xf3\xbf\x95K\x00\x00\x00\x00\x00\x00\x8c\x07n_iter_\x94h\x1b)\x81\x94}\x94(h\x1eh!h"K\x01\x85\x94h$h\th%h
     &\x8c\x02i4\x94\x89\x88\x87\x94R\x94(K\x03h*NNNJ\xff\xff\xff\xffJ\xff\xff\xff\xffK\x00t\x94bh,\x88ub\x07\x00\x00\x00\x95\x1f\x00\x00\x00\x00\x
     00\x00\x00\x8c\x10_sklearn_version\x94\x8c\x060.24.2\x94ub.']
```

# Security Risks with Pickles II

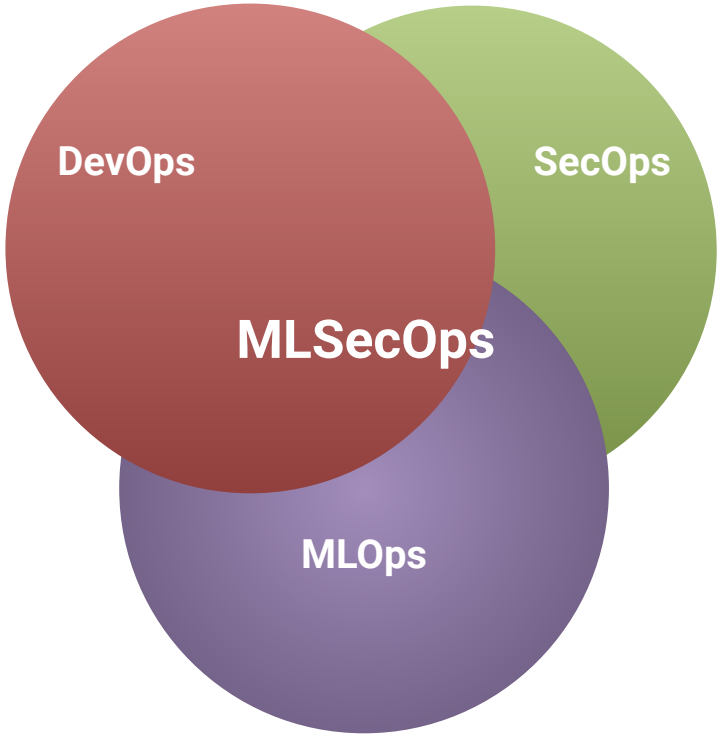
Including **arbitrary code executions**!

```
[50]: import joblib

      model_safe = joblib.load("fml-artifacts/safe/model.joblib")

[51]: model_safe.predict(X[:1])

[51]: array([0])

 [1]: import types, os, base64

      def __reduce__(self):
          # This is basically base64 for cmd = "env > pwnd.txt"
          cmd = base64.b64decode("ZW52ID4gcHduZC50eHQ=").decode()
          return os.system, (cmd,)

 [2]: model_safe.__class__.__reduce__ = types.MethodType(__reduce__, model_safe.__class__)

      •••

 [ ]: !mkdir -p fml-artifacts/unsafe/

 [ ]: joblib.dump(model_safe, "fml-artifacts/unsafe/model.joblib")

 [3]: with open("fml-artifacts/unsafe/model.joblib", "rb") as f: print(f.readlines())

      [b'\x80\x04\x95)\x00\x00\x00\x00\x00\x00\x8c\x05posix\x94\x8c\x06system\x94\x93\x94\x8c\x0eenv > pwnd.txt\x94\x85\x94R\x94.']
```

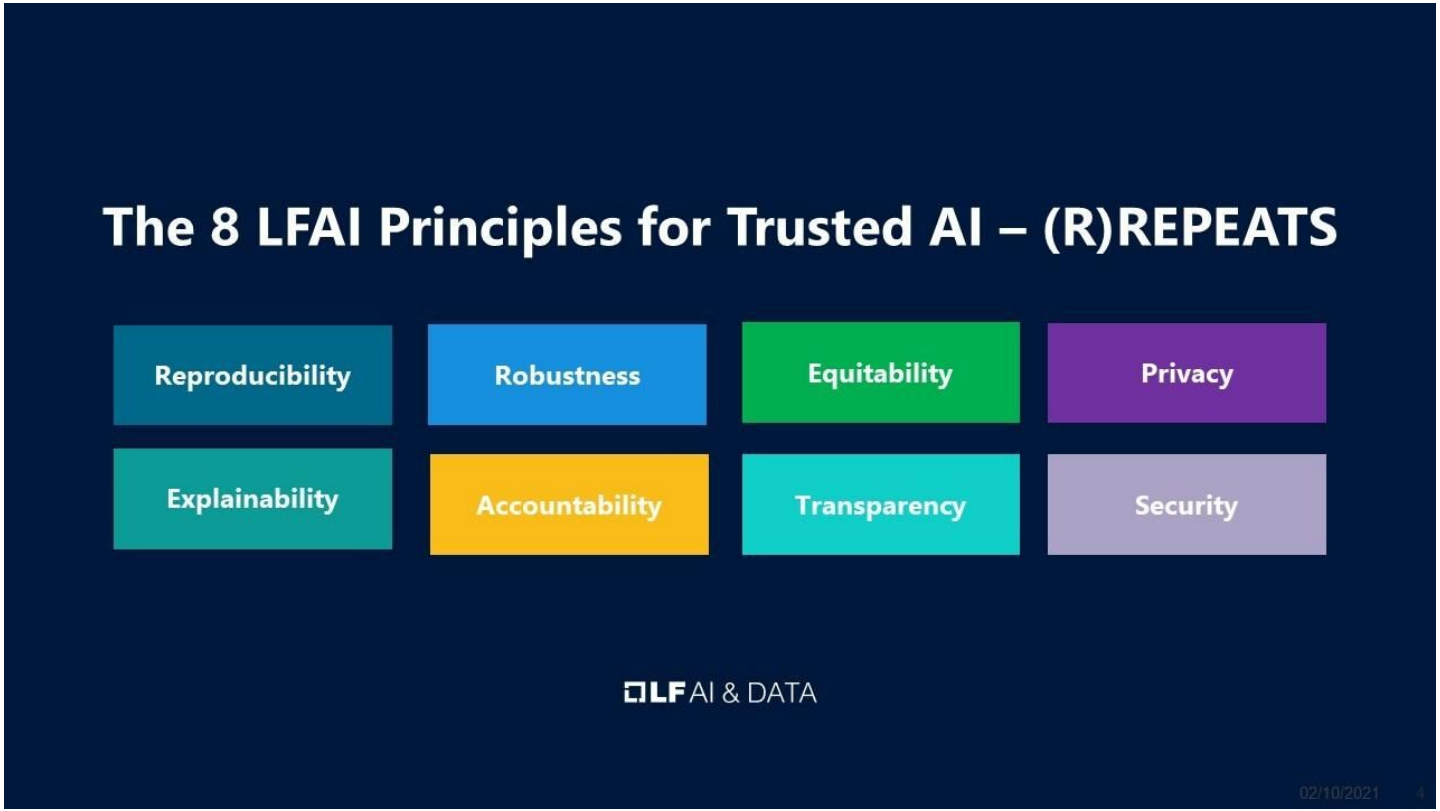
# What is MLSecOps?

➔ The Extension of DevOps and Security with machine learning infrastructure as first class citizen

➔ The Intersection of ML infrastructure, developer operations / automation, and security policies



# Trusted AI Principles



# LFAI MLSecOps Top 10 for Machine Learning

[Flawed ML Security talk from KubeCon 2022]

LF AI & DATA

| # | MLSecOps Top 10 |
|---|---|
| 1 | Unrestricted Model Endpoints |
| 2 | Access to Model Artifacts |
| 3 | Artifact Exploit Injection |
| 4 | Insecure ML Systems/Pipeline Design |
| 5 | Data & Infra Misconfigurations |
| 6 | Supply Chain Vulnerabilities in ML Code |
| 7 | IAM & RBAC Failures for ML Services |
| 8 | ML Infra / ETL / CI / CD Integrity Failures |
| 9 | Observability, Reproducibility & Lineage |
| 10 | ML-Server Side Request Forgery |

# Other issues with Pickles

Besides security issues, **Pickles also suffer from other issues**:

- ➔ **Python version incompatibility** issues
  - ◆ Pickles generated in one version may not work with other versions
- ➔ **Framework version incompatibility** issues
  - ◆ No way to version framework in Pickles
- ➔ Hard to **troubleshoot** the above

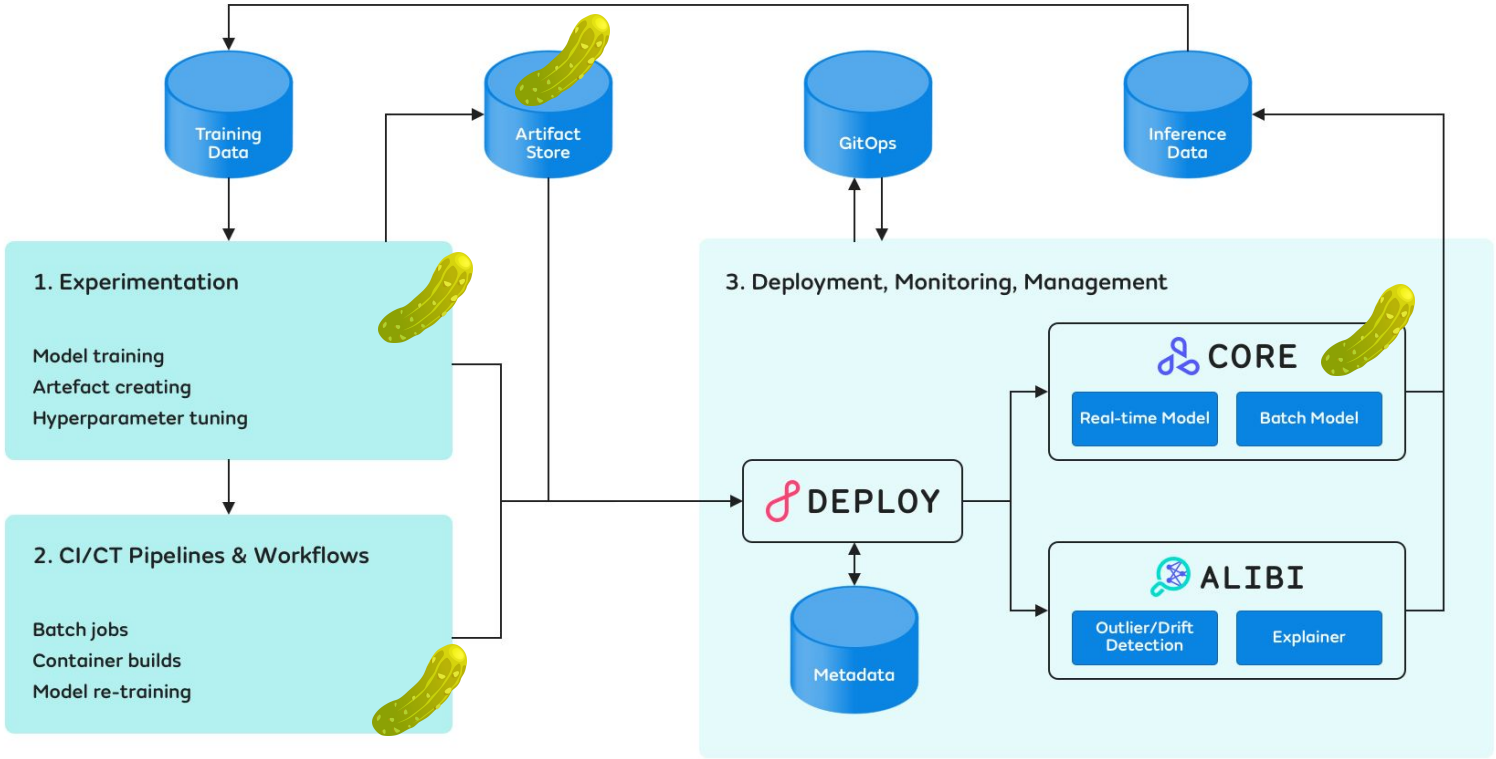
e.g. error from using the wrong sklearn version


```
  File "/opt/conda/lib/python3.8/site-packages/joblib/numpy_pickle.py", line 577, in _unpickle
    obj = unpickler.load()
  File "/opt/conda/lib/python3.8/pickle.py", line 1212, in load
    dispatch[key[0]](self)
  File "/opt/conda/lib/python3.8/site-packages/joblib/numpy_pickle.py", line 402, in load_build
    Unpickler.load_build(self)
  File "/opt/conda/lib/python3.8/pickle.py", line 1705, in load_build
    setstate(state)
  File "sklearn/tree/_tree.pyx", line 714, in sklearn.tree._tree.Tree.__setstate__
  File "sklearn/tree/_tree.pyx", line 1418, in sklearn.tree._tree._check_node_ndarray
ValueError: node array from the pickle has an incompatible dtype:
- expected: {'names': ['left_child', 'right_child', 'feature', 'threshold', 'impurity', 'n_node_samples', 'weighted_n_node_samples', 'missi
_go_to_left'], 'formats': ['<i8', '<i8', '<i8', '<f8', '<f8', '<i8', '<f8', 'u1'], 'offsets': [0, 8, 16, 24, 32, 40, 48, 56], 'itemsize': 6
- got      : [('left_child', '<i8'), ('right_child', '<i8'), ('feature', '<i8'), ('threshold', '<f8'), ('impurity', '<f8'), ('n_node_samples
'<i8'), ('weighted_n_node_samples', '<f8')]
```

# Poisoned Pickles make you ill

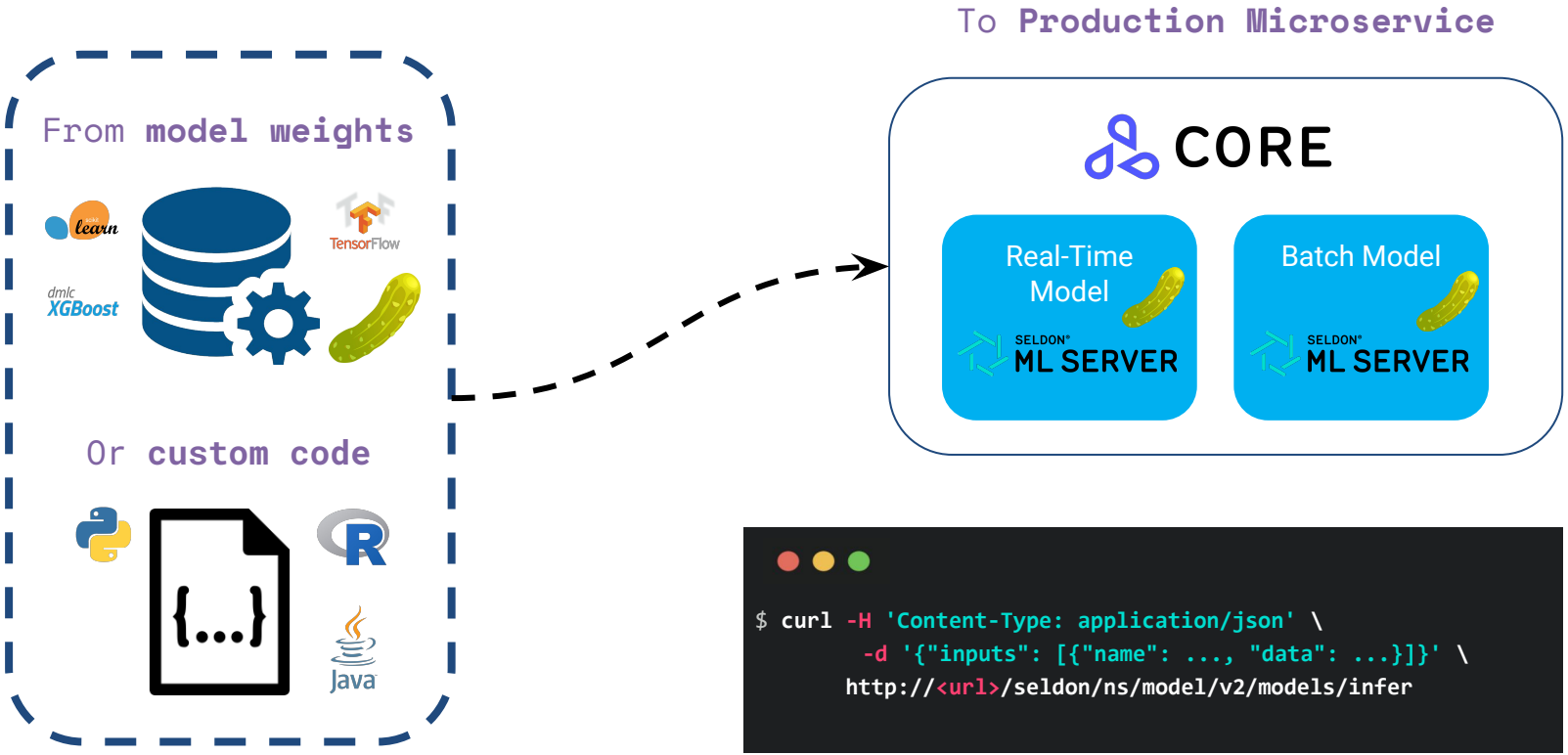
# Some background knowledge...

# MLOps Serving Reference Architecture



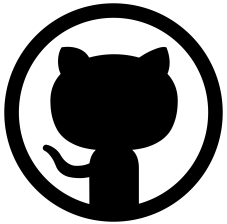

To learn more about production MLOps, catch up [Alejandro's talk on the State of Production ML]

# TLDR; Machine Learning Deployed ✅




```
$ curl -H 'Content-Type: application/json' \
       -d '{"inputs": [{"name": ..., "data": ...}]}' \
      http://<url>/seldon/ns/model/v2/models/infer
```

# Demo



https://github.com/adriangonz/sigstore-talk

# Poisoned Pickles Make You ill

- ➔ It’s incredibly **easy to poison** a Pickle
- ➔ It’s incredibly **hard to detect** if a Pickle has been poisoned

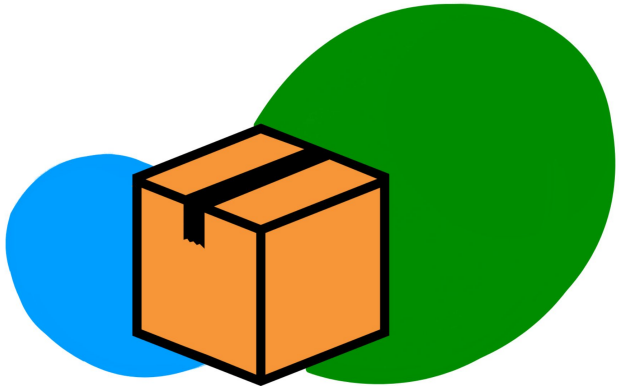

https://huggingface.co/docs/hub/security-pickle

# Ok… Pickles may not be that great

What can we do? Can we have *“higher-quality Pickles”*?

➔ **Don’t use Pickle** (e.g. use ONNX instead)

➔ Use tools like **SKOps** to mitigate Pickle risk [see talk by Merve Noyan on PyData 2022]

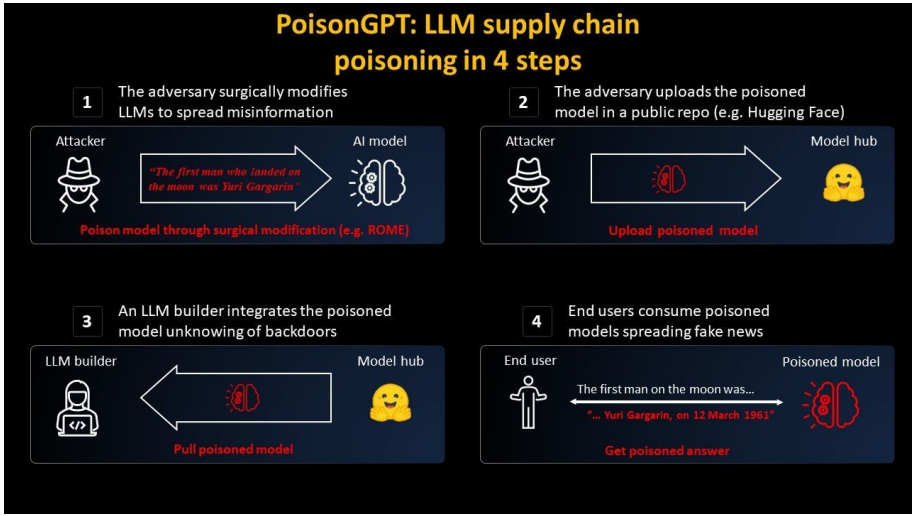


https://skops.readthedocs.io/en/stable/

# Ok… Pickles may not be that great

Even with *“higher-quality Pickles”*, there are risks….

**PoisonGPT:**
https://blog.mithrilsecurity.io/poisongpt-how-we-hid-a-lobotomized-llm-on-hugging-face-to-spread-fake-news/

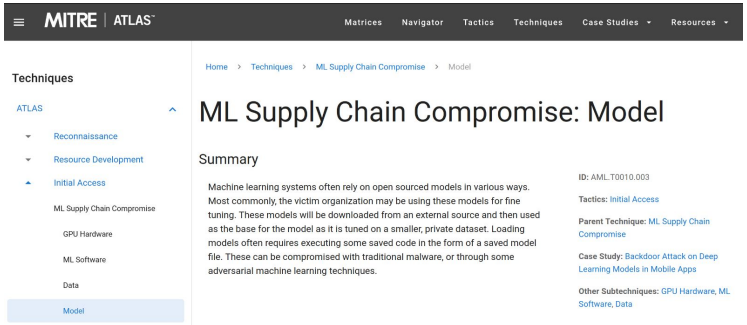


# Tamper-proof Pickle jars

# Tamper-proof Pickle jars

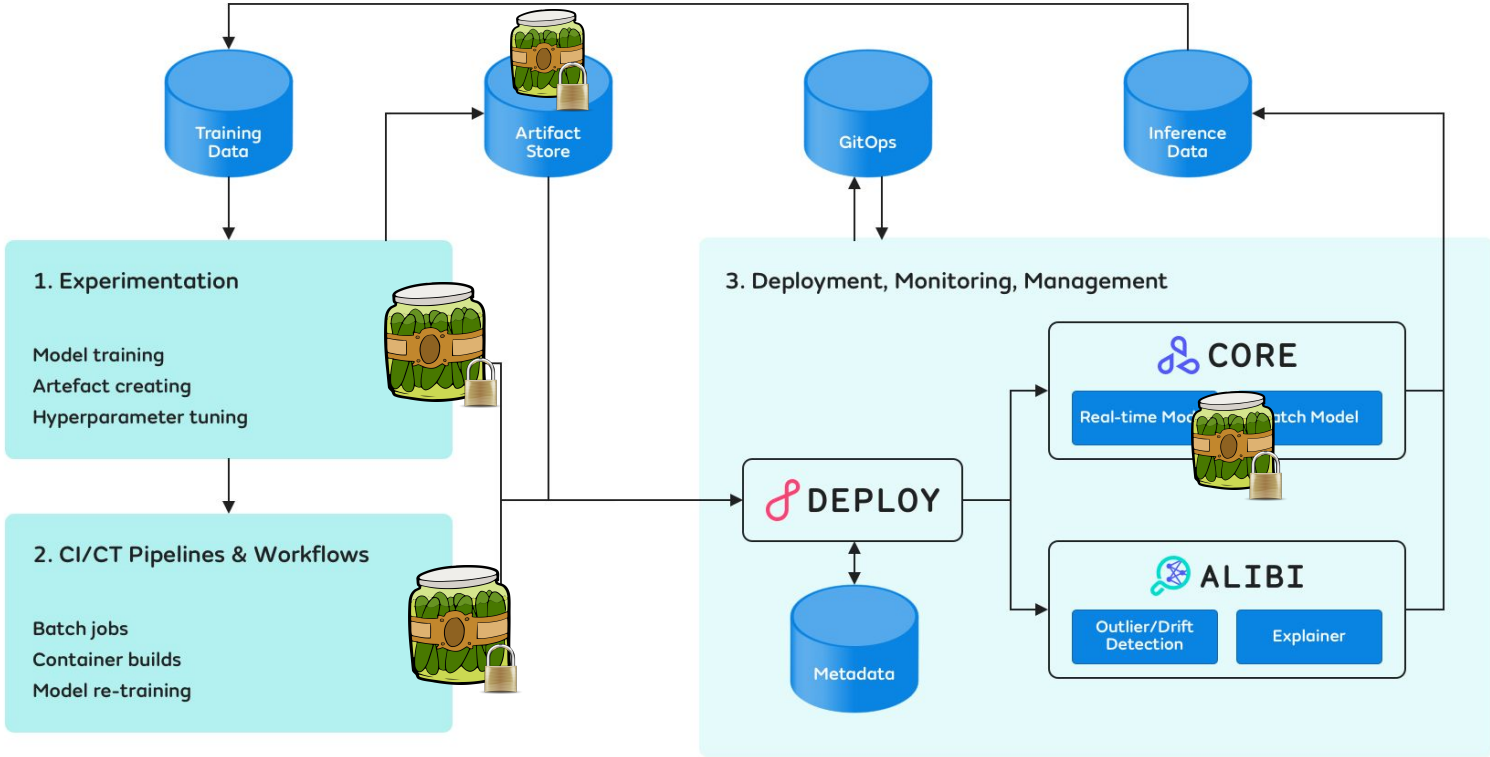
There's always a risk of an attacker tampering with the artefact:

- ➔ Security requirements are analogous to application development, which **cannot expect to be solved with just magical scanning**
- ➔ We need "**trust or discard**" mechanisms that need to be in play through the automation touchpoints of CI / CD / ETL




Catalogue of **attack vectors into ML systems**: https://atlas.mitre.org/

# Tamper-proof Pickle jars



# Tamper-proof Pickle jars

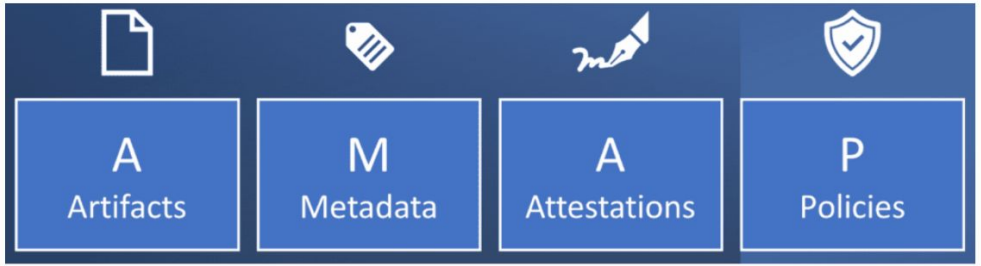
Can't we just *sign* the artefact?

- ➔ Who guarantees that the key used for the signature is valid?
- ➔ Who guarantees that the guarantee we have for the key is valid?
- ➔ And so on… Turtles all the way down!

# Supply Chain Security

Luckily, we can borrow ideas from ***"classic"*** **DevSecOps** to ensure the integrity of our model artefacts!

- ➔ Large landscape of projects!
- ➔ Sign **Artefacts**, **Metadata** and **Attestations**
- ➔ Apply **policies** to verify these


A MAP for software supply chain security

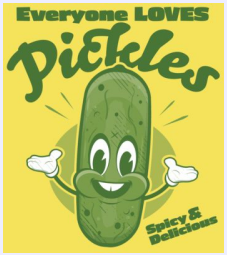
https://www.cncf.io/blog/2022/04/12/a-map-for-kubernetes-supply-chain-security/

# Supply Chain Security

## Artefacts

Our **ML "binary" artefacts**

- Pickles, ONNX, TF's SavedModel, etc.




## Metadata

- **Provenance Data**: Who trained this model? When did they train? What training pipeline did they use?
- **Software Bill of Materials (SBOMs)**: What dataset was used? What package dependencies does the model have? We can use CycloneDX ML-BOM or (soon) SPDX AI.
- **Vulnerability Scan Reports**: Known vulnerabilities within the model or its dependencies. We can use VEX format.

## Attestations

**Assurance** that the artefacts and metadata come from the right persons and **haven't been tampered with**.

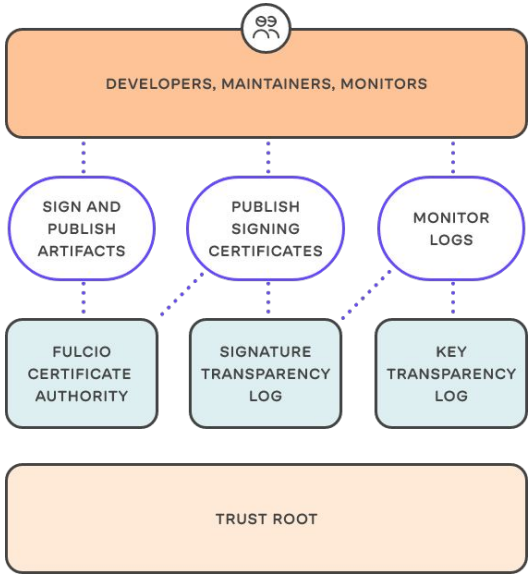
- Signature of Docker image
- **Signature** of our ML artefacts
- etc.

sigstore

# Supply Chain Security

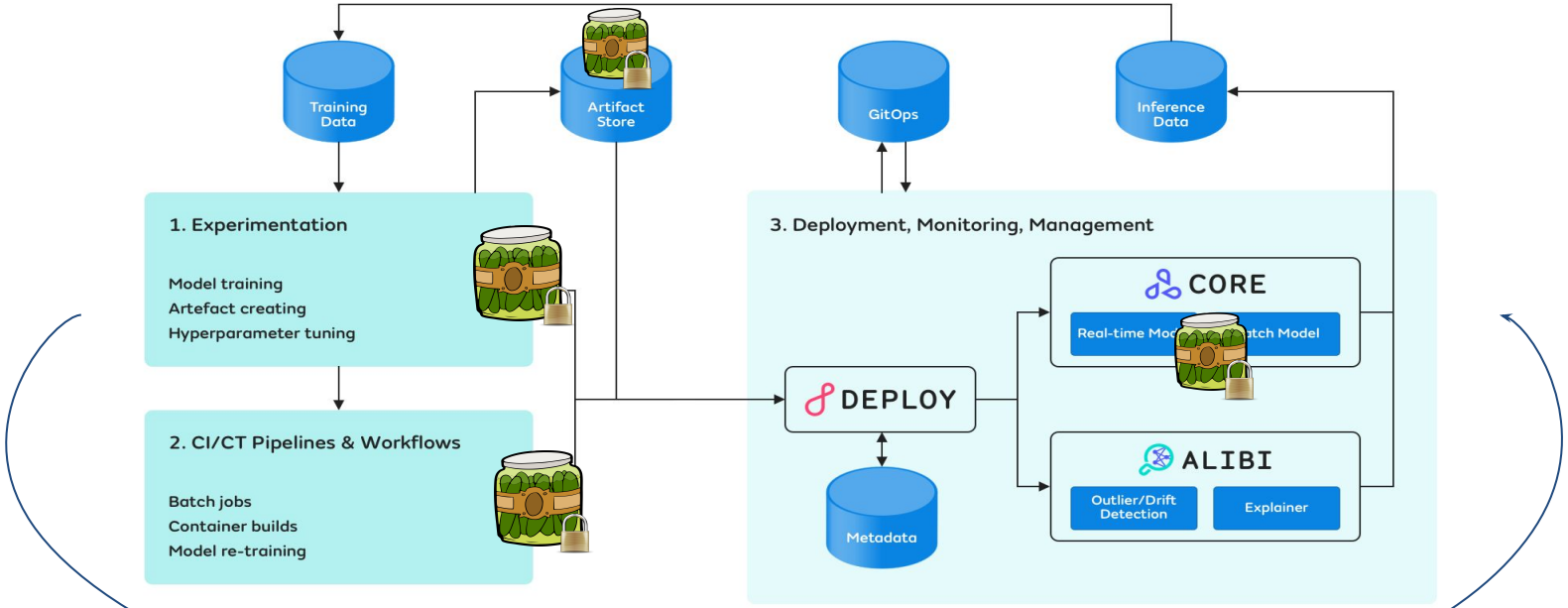
Sigstore orchestrates the whole signing and verification process across a number of components:

- **Fulcio**: A free certificate authority.
- **Rekor**: Ledger with signed metadata to validate details.
- **OIDC Gateway**: Sigstore offers a simple OIDC Gateway, which can federate against Google, GitHub and Microsoft.
- The sigstore project runs hosted versions of the above, however they can also be run on-prem.




sigstore

# Tamper-proof Pickle jars

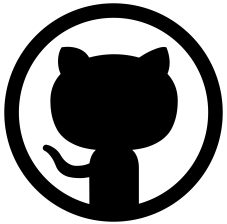



sigstore

**1**. After training, we sign the artefact with Sigstore

**2**. On deployment, we check Sigstore to validate the artifact's signature

# Demo



https://github.com/adriangonz/sigstore-talk

# Next Steps

There don't seem to be many *"best practices"* around supply chain security for MLSecOps.

- ➔ No vendors provide signing - or the ability to apply policies (**yet**).
- ➔ Can we leverage other **OSS projects to apply policies** automatically?
- ➔ There's still work to do!

# Monthly MLSecOps Wg Meetings

## Join Monthly meetings:

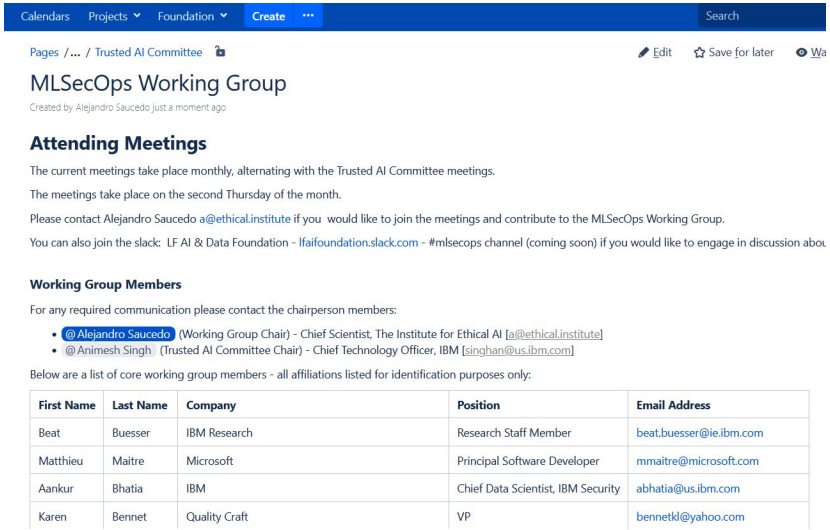
forms.gle/N96w6h4wLDkfgsEs8

## Key milestones

➔Publish MLSecOps Top 10

➔ML Security Standards

➔Best Practices and Examples of Secure ML



Calendars Projects Foundation Create ··· Search

Pages /... / Trusted AI Committee Edit Save for later Wa

MLSecOps Working Group

Created by Alejandro Saucedo just a moment ago

**Attending Meetings**

The current meetings take place monthly, alternating with the Trusted AI Committee meetings.

The meetings take place on the second Thursday of the month.

Please contact Alejandro Saucedo a@ethical.institute if you would like to join the meetings and contribute to the MLSecOps Working Group.

You can also join the slack: LF AI & Data Foundation - lfaifoundation.slack.com - #mlsecops channel (coming soon) if you would like to engage in discussion abou

**Working Group Members**

For any required communication please contact the chairperson members:

- @Alejandro Saucedo (Working Group Chair) - Chief Scientist, The Institute for Ethical AI [a@ethical.institute]
- @Animesh Singh (Trusted AI Committee Chair) - Chief Technology Officer, IBM [singhan@us.ibm.com]

Below are a list of core working group members - all affiliations listed for identification purposes only:

| First Name | Last Name | Company | Position | Email Address |
|---|---|---|---|---|
| Beat | Buesser | IBM Research | Research Staff Member | beat.buesser@ie.ibm.com |
| Matthieu | Maitre | Microsoft | Principal Software Developer | mmaitre@microsoft.com |
| Aankur | Bhatia | IBM | Chief Data Scientist, IBM Security | abhatia@us.ibm.com |
| Karen | Bennet | Quality Craft | VP | bennetkl@yahoo.com |

# Thank You

Adrian Gonzalez-Martin

Head of ML Serving

agm@seldon.io