

Privacy-Preserving and Distributed Machine Learning

Cláudia Brito

February 2023

claudia.v.brito@inesctec.pt

Data is Everywhere
Everything is Data



Do you accept the terms and conditions?

When you accept all the Terms and conditions:

Me who didn't read it:



The company who doesn't
want you to read it:



But what about **my privacy?**

What is private data?

Private Data:

- Name
- Address
- Financial Status
- Health Records

Big Collectors:

- Hospitals
- Banks
- Finances
- Biocenters

Other Collectors:

- Browsers
- Social Networks
- Games

Do you allow us to collect your data?



So, how about we do all of this in a private way?

The real problem?

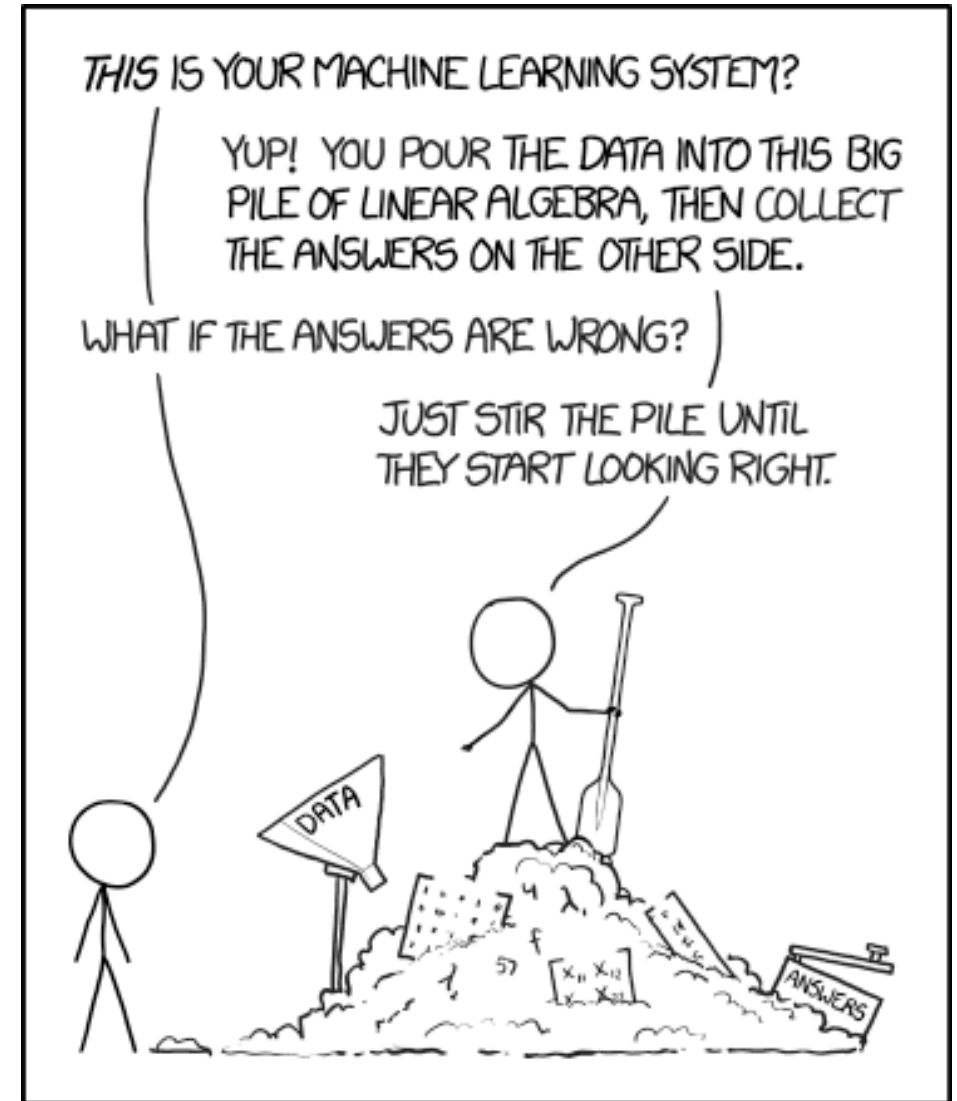
Large amounts of sensitive data generated.

Use of AI techniques to extract valuable insights.

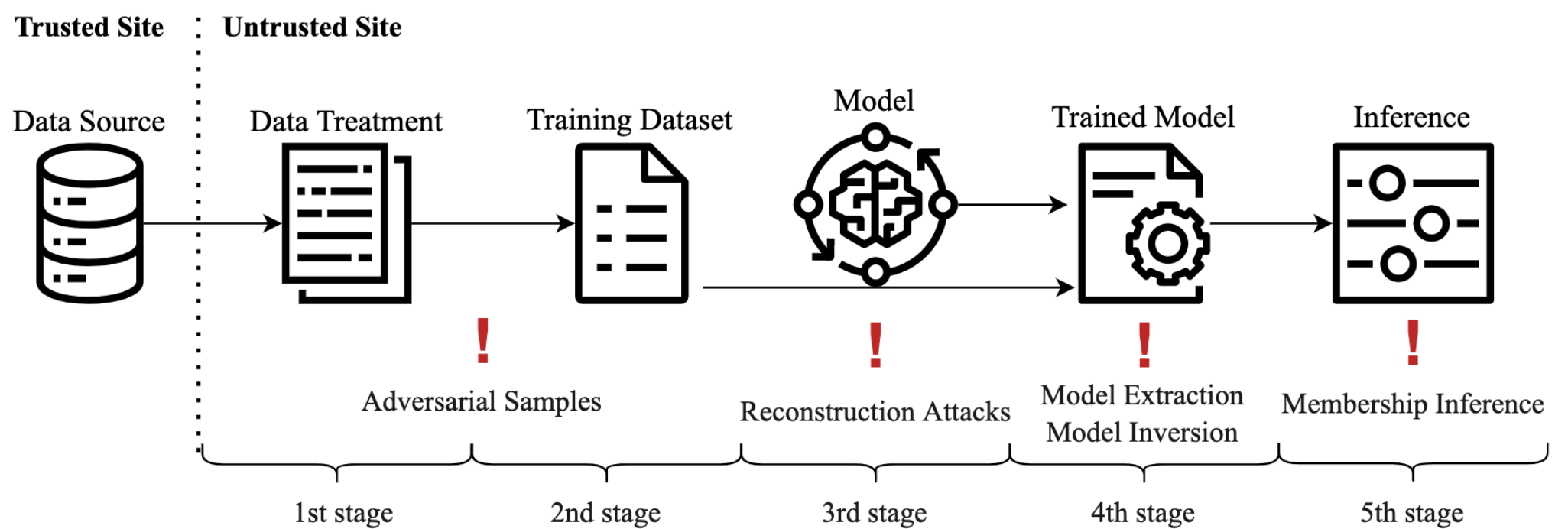
Regulations to avoid the misuse of sensitive data.

Why does ML comes in the picture?

*“A computer program is said to learn from **experience E** with respect to some **task T** and some performance **measure P** , if its performance on T , as measured by P , improves with experience E .”*



The ML Workflow



New Limitations and Challenges

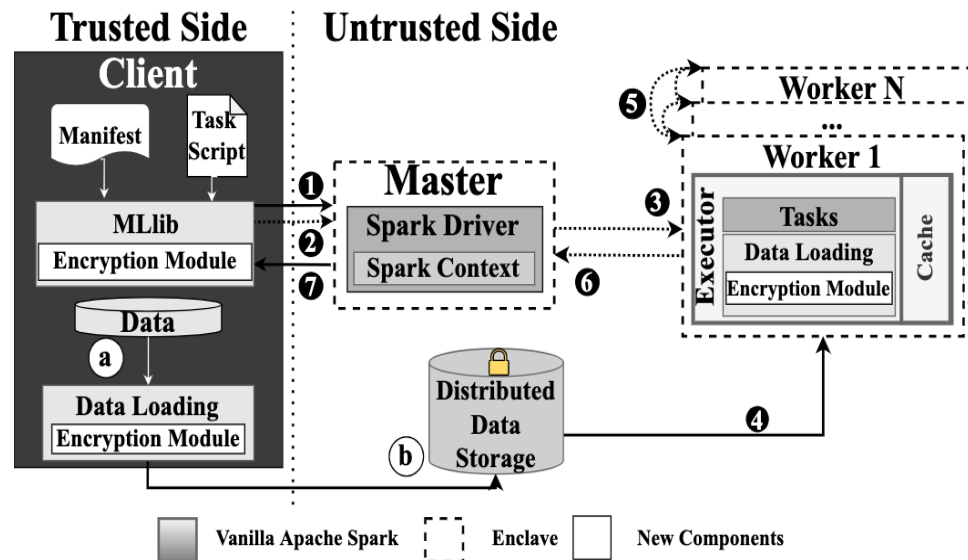
- ML datasets and models are stored and processed in plaintext.
- Large amounts of data to be processed and heavy computation.
- Common cryptographic schemes impose impractical overheads.
- Lack of local infrastructures.

Solutions

1. Outsource computation to third-party infrastructures.
2. Provide secure environments.
3. Allow the private collaboration between all the entities.

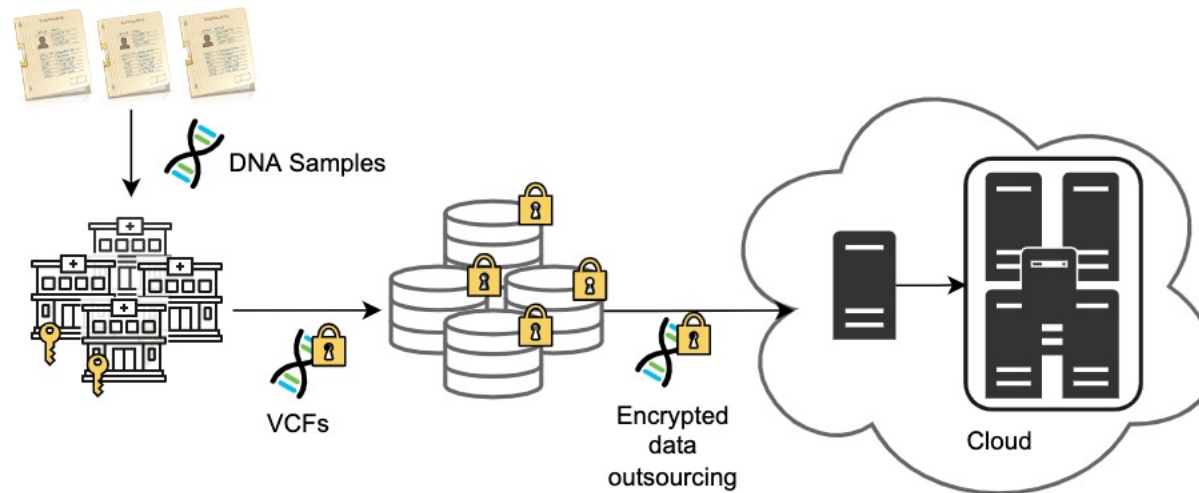
Soteria: Preserving Privacy in Distributed Machine Learning (SAC'23)

- Allows the outsourcing of computation.
- Provides secure environments for the outsourced computation.
- Redefines which computation needs to be kept private.



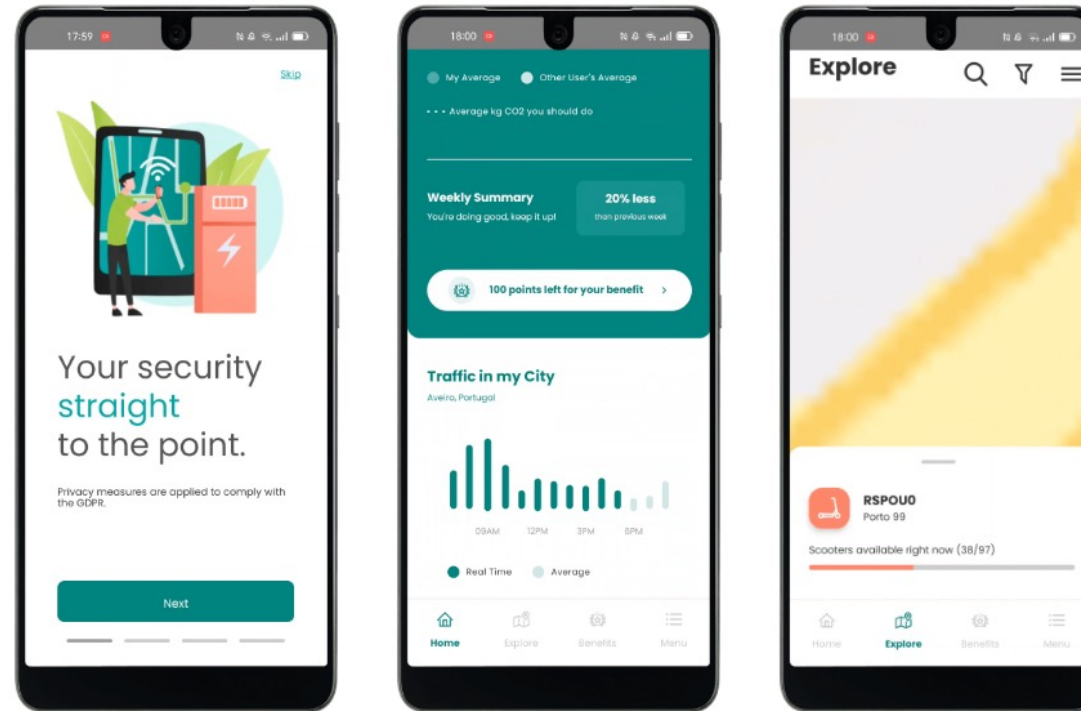
Work in Progress: Privacy-Preserving Framework for Genomic-Wide Association Studies

- Allows the outsourcing of genomic computation.
- Provides secure distributed environments for the computation of the genome.
- Promotes the collaboration between entities.



Promoting sustainable and personalised travel behaviours while preserving data privacy (TRA2022)

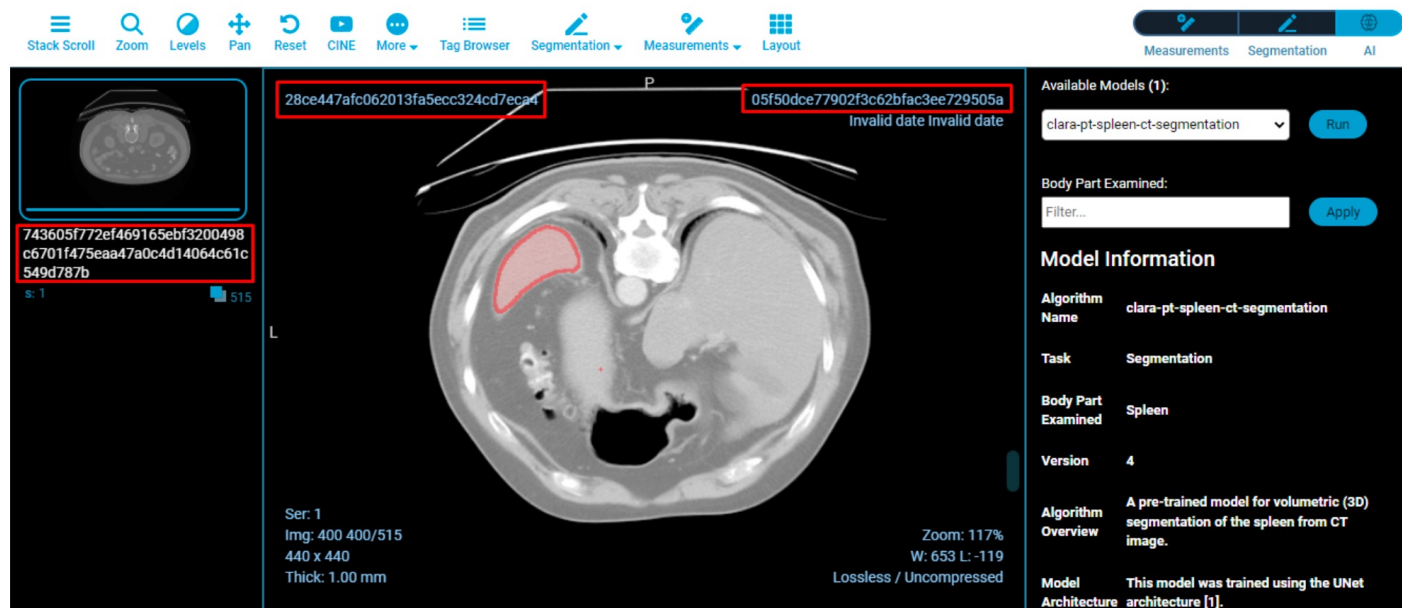
- ✓ **Assessing GHG emissions** from transport modes
- ✓ **Improving data experience** with real-time access
- ✓ **Privacy-preserving data** by not uploading it to centralised servers (leveraged by **AI approach**)
- ✓ **Local incentives** to reduce GHGs



AI4CITIES franchet.ai

Other Research Works

MCC - MedCloudCare



Medical platform for visualizing and analyzing DICOM images by resorting to machine learning algorithms.

requirements.txt
template
models

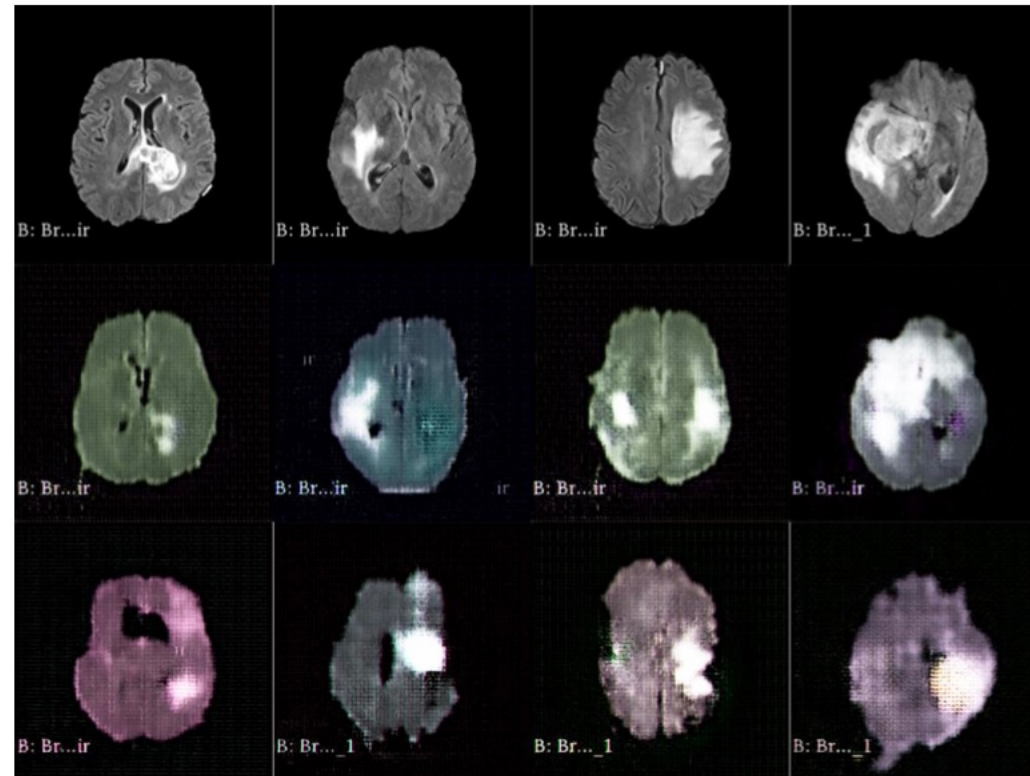
```
4 from monai.networks.nets import UNet
5 from monai.inferers import SlidingWindowInferer
6 from monai.transforms import (
7     Activationsd,
8     AddChanneld,
9     AsDiscreted,
10     LoadImaged,
11     ScaleIntensityRanged,
12     Spacingd,
13     ToNumpyd,
14     ToTensord,
15 )
16 from monai.label.transform.post import BoundingBoxd, Restored
17 from monai.label.interfaces.utils.transform import run_transforms
18 import torch
19
20
21 class SegmentationSpleen(InferTask):
22     """
23     This provides Inference for pre-trained spleen segmentation (UNet) model over MSD Data
24     """
25
26     def __init__(
27         self,
28         root_dir,
29         type=InferType.SEGMENTATION,
30         labels=["spleen"],
31         description="A pre-trained model for volumetric (3D) segmentation of the spleen from CT image."
32     ):
33         super().__init__(
34             type=type,
35             labels=labels,
36             description=description,
37         )
```

requirements.txt
template
models

+ Upload pre-trained model files main.py

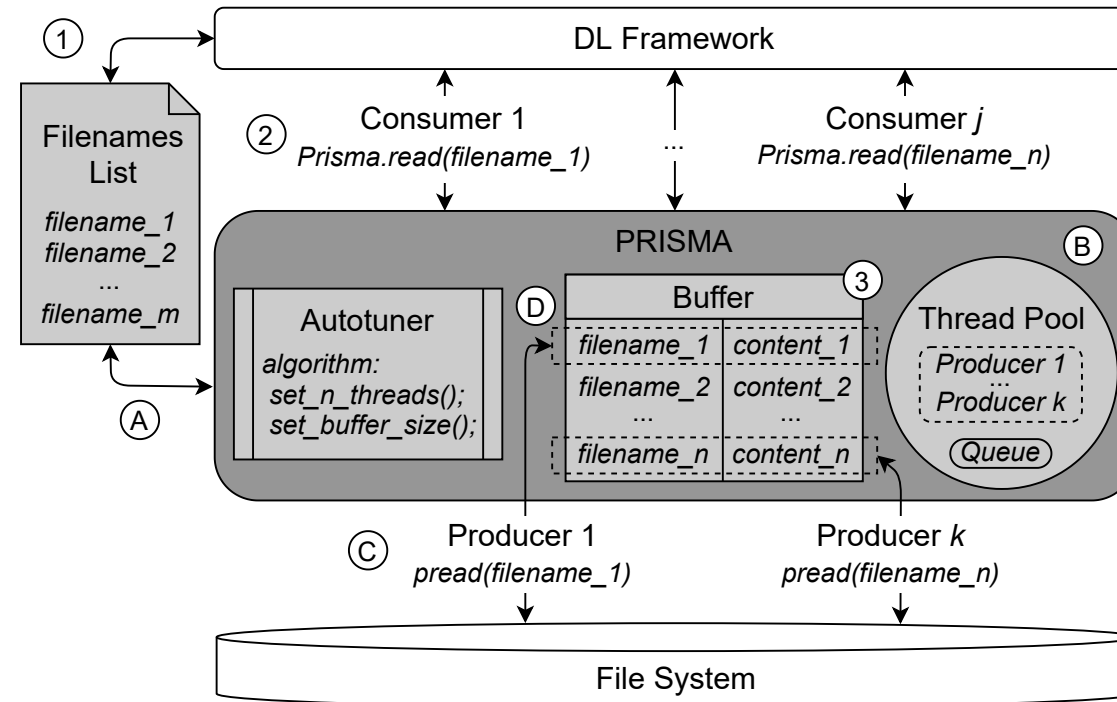
```
91 return network
92
93
94 def run(self, input_data):
95     data = self.pre_processing(input_data)
96
97     inferer = SlidingWindowInferer(roi_size=[160, 160, 160])
98
99     if not torch.cuda.is_available():
100         device = "cpu"
101     else:
102         device = "cuda"
103
104     network = self.get_model_network(device)
105     if network:
106         inputs = data["image"]
107         inputs = inputs if torch.is_tensor(inputs) else torch.from_numpy(inputs)
108         inputs = inputs[None]
109         if device == "cuda":
110             inputs = inputs.cuda()
111         with torch.no_grad():
112             outputs = inferer(inputs, network)
113         if device == "cuda":
114             torch.cuda.empty_cache()
115         outputs = outputs[0]
116         data["pred"] = outputs
117     else:
118         data = run_transforms(data, inferer, log_prefix="INF", log_name="Inferer")
119
120     output_data = self.post_processing(data)
121
122     return output_data
123
124
125 def post_processing(self, data):
126     transforms = [
127         Activationsd(keys="pred", softmax=True),
```

Generation of Medical Images with Deep Learning



Generation of medical images with high resolution to increase the medical data used for developing new models of diagnosis.

PRISMA



*Storage optimizations
for Deep Learning
Workloads.*

Questions?

If you have any questions send an e-mail to:
claudia.v.brito@inesctec.pt