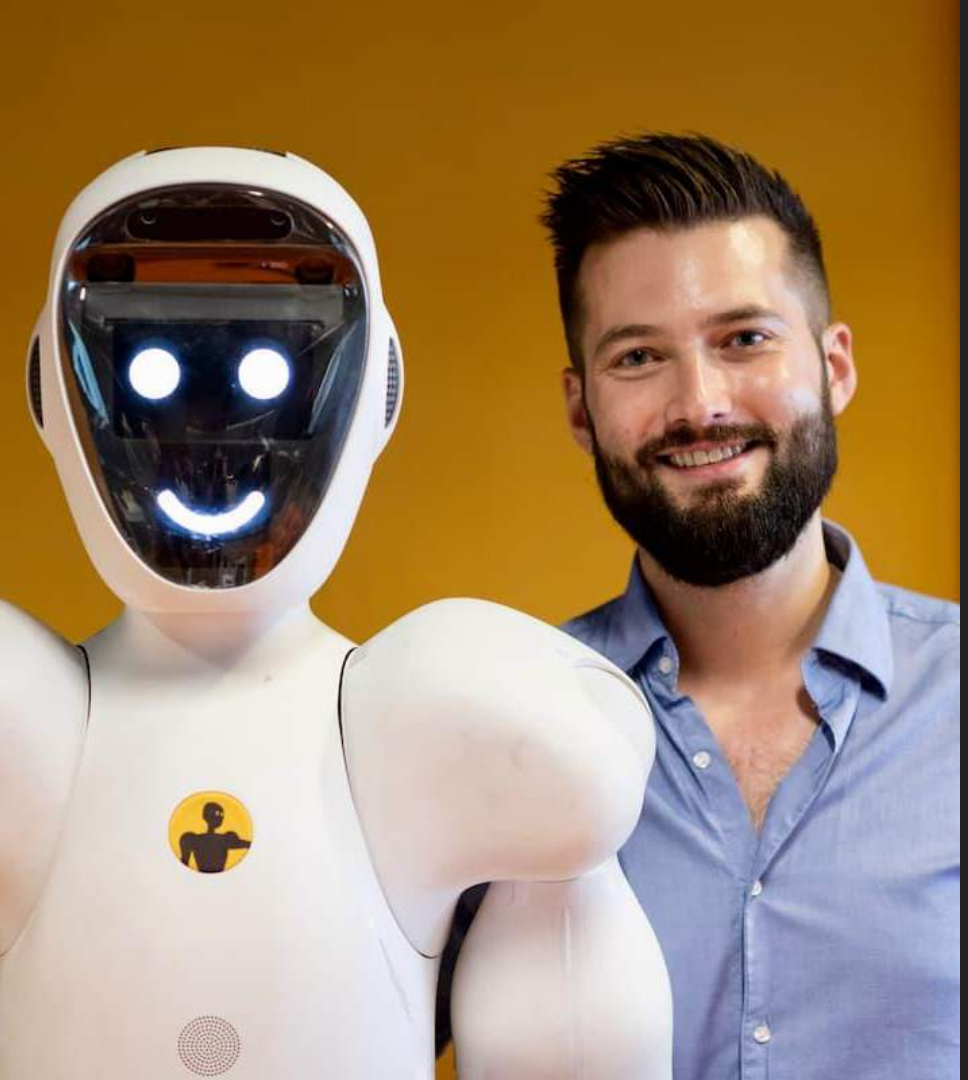
Harnessing Python for Hard Tech Applications

Nicholas Nadeau, Ph.D., P.Eng.



Nicholas Nadeau

nadeau innovations

HALODI robotics

AON3D

ÉTS
Le génie pour l'industrie

McGill

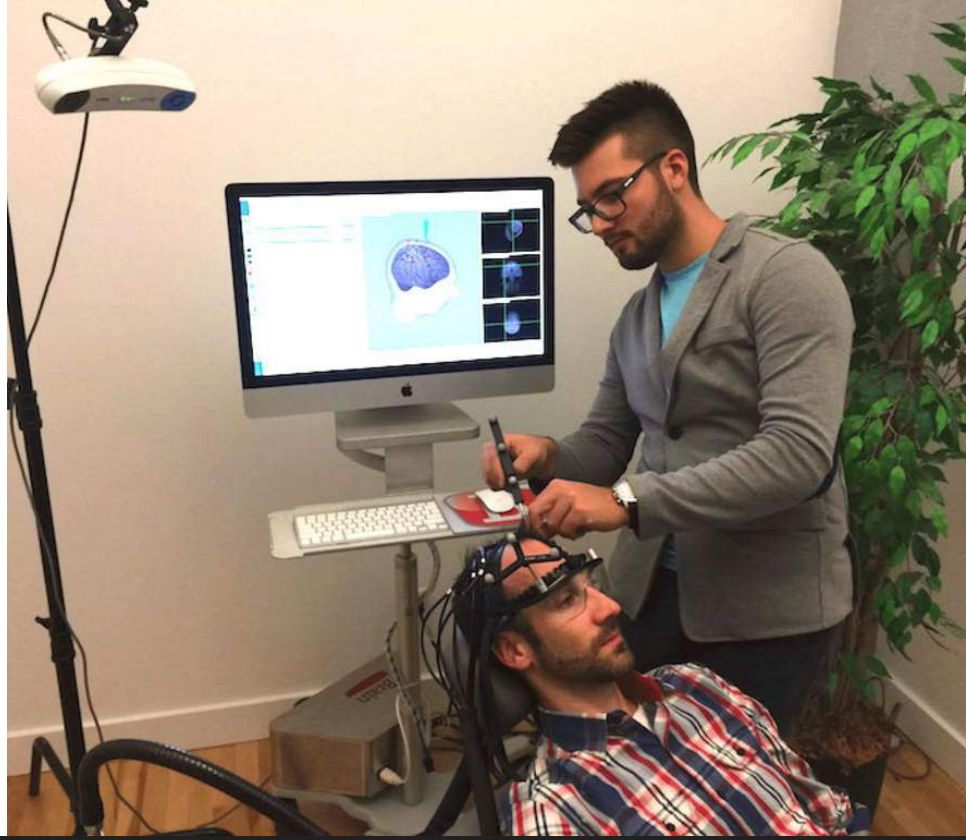
techstars_

NEXTAI

CREATIVE DESTRUCTION LAB

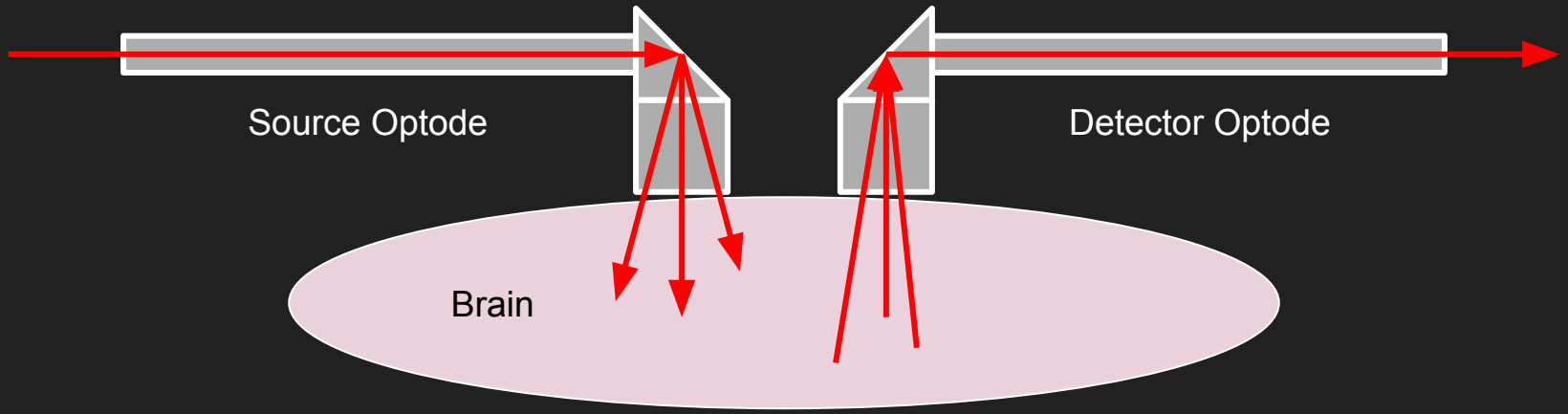
FounderFuel


Rogue Research Inc.
Rogue Research
Brainsight®
NIRS
Brainsight®

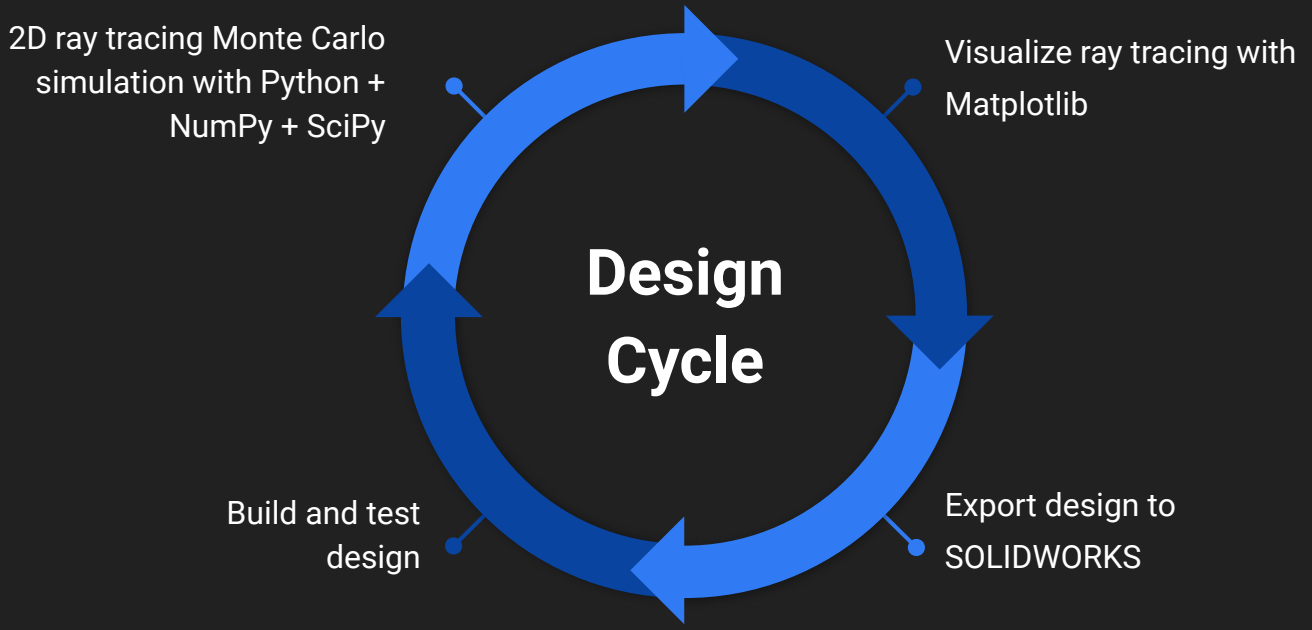


Near-infrared Spectroscopy (NIRS)

Optode Design



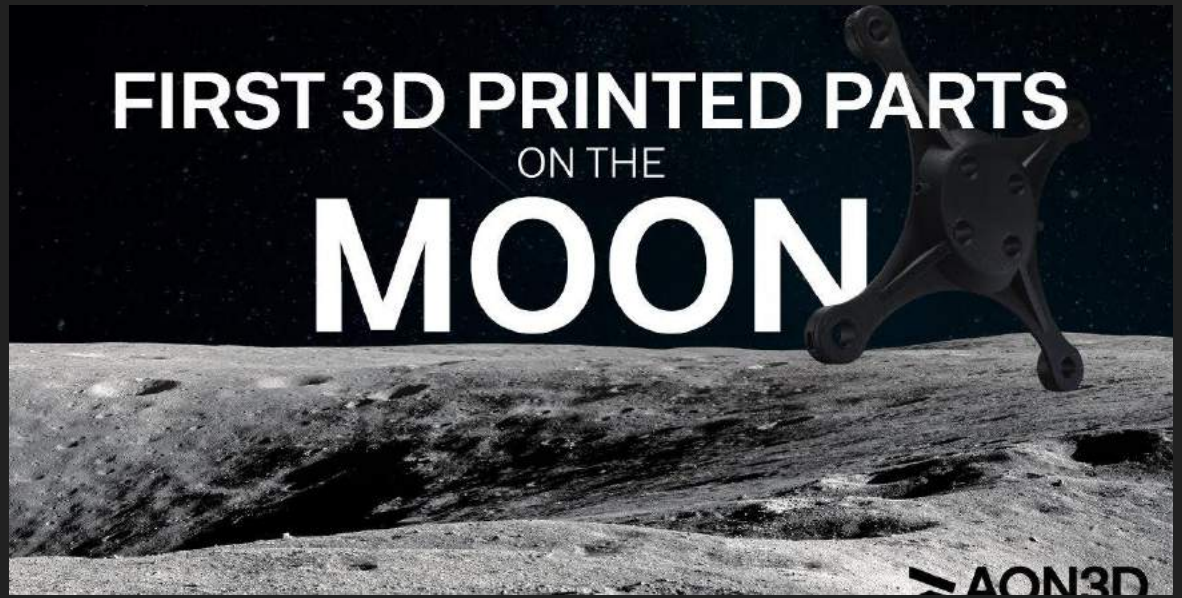
nadeau innovations



nadeau innovations


AON3D

High-performance Parts



Human-Machine Interface (HMI)

Touchscreen HMI

ARM-based SBC (Single Board Computer)

- React UI/UX Container
- Python Flask Backend Container
- Balena Device Management

ATMega-based Microcontroller

- C/C++ Firmware

Hardware System

- Motion Control
- Thermal Control

nadeau innovations

```python
from datetime import datetime

import qrcode

data = {
    "machine": "machine_1",
    "print_file": "test_print.gcode",
    "temp_bed": 60,
    "temp_chamber": 20,
    "temp_nozzle": 200,
    "time": datetime.now().astimezone().isoformat(),
}

img = qrcode.make(data)
img.save("qr_log.png")
```

Making Logs Fun

Scraping Data MVP

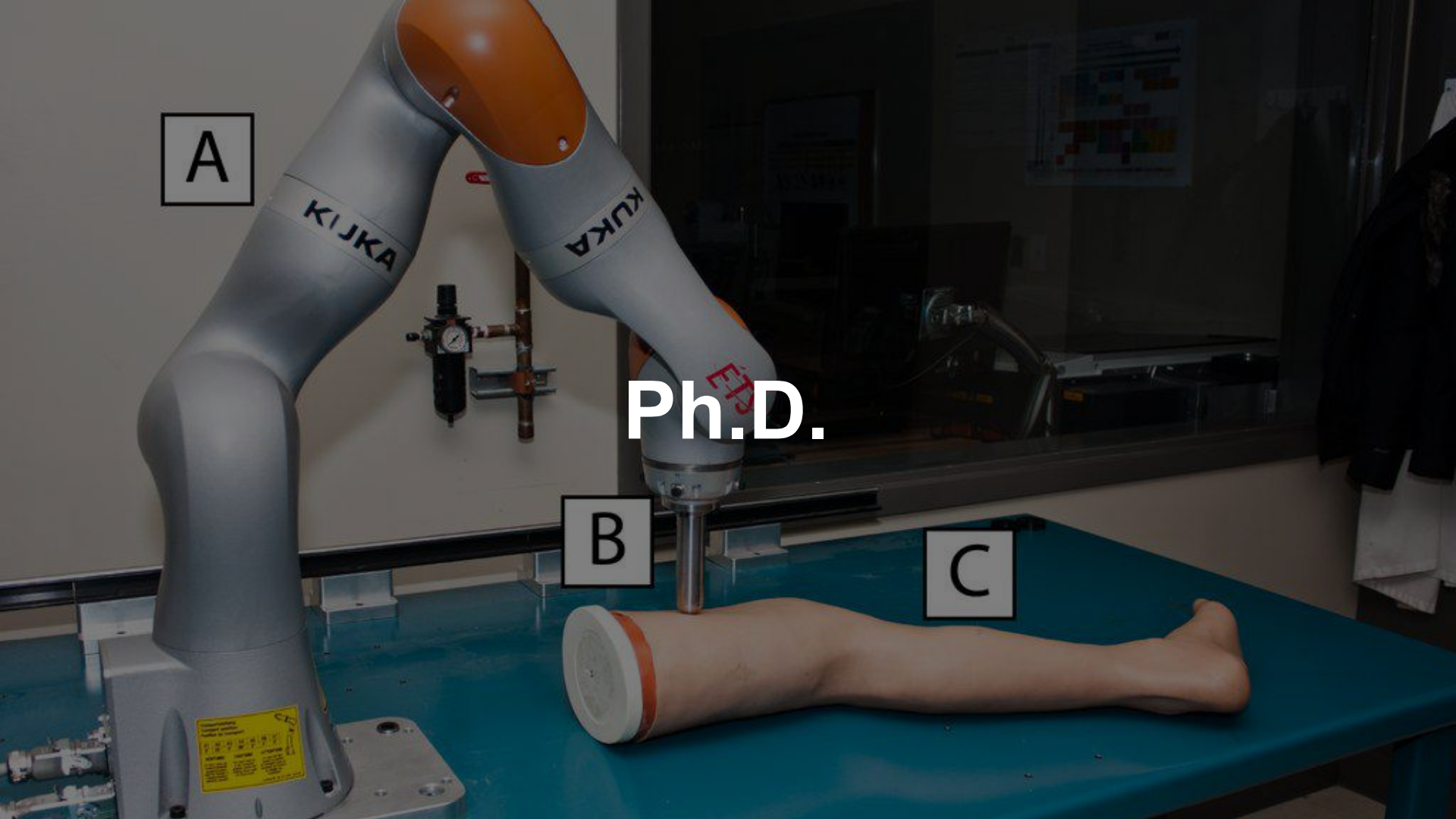
Grafana	**Metabase**
PostgreSQL	**Excel + CSV**

Python Cron Job + ETL

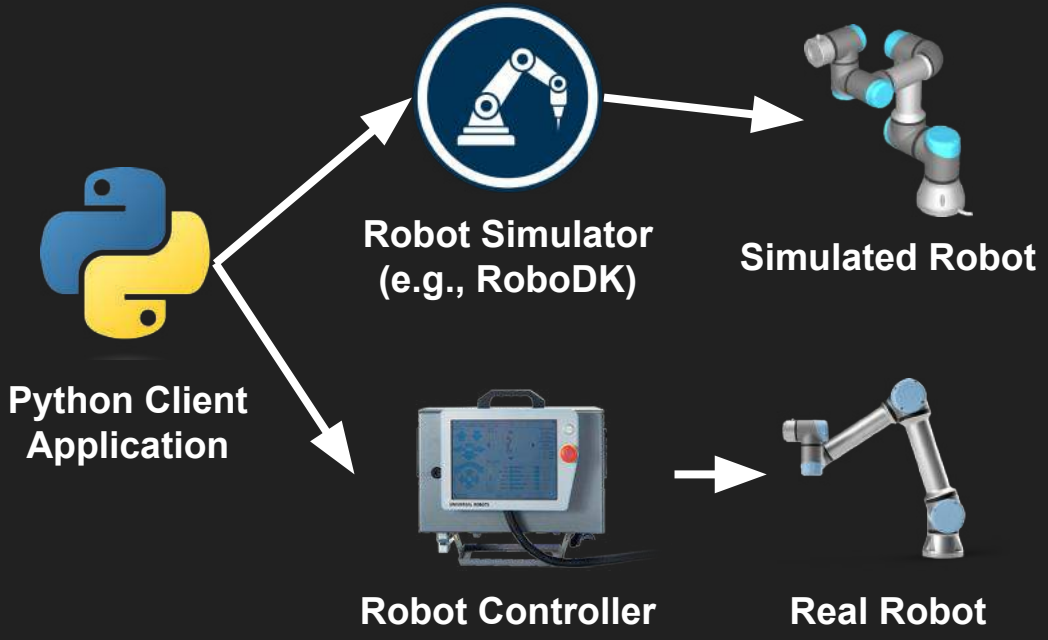
- Requests with REST APIs
- Pandas + SQLAlchemy for ETL

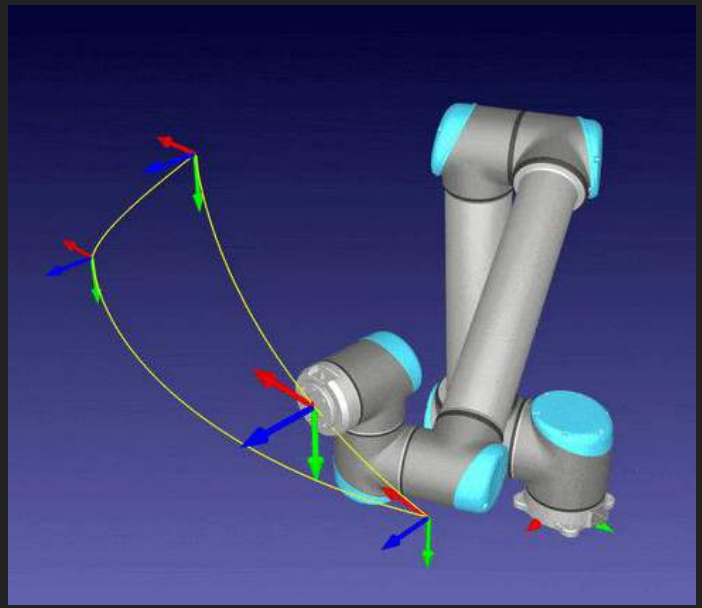
Print Farm

nadeau innovations


A
KUKA
KUKA
ÉTS
Ph.D.
B
C

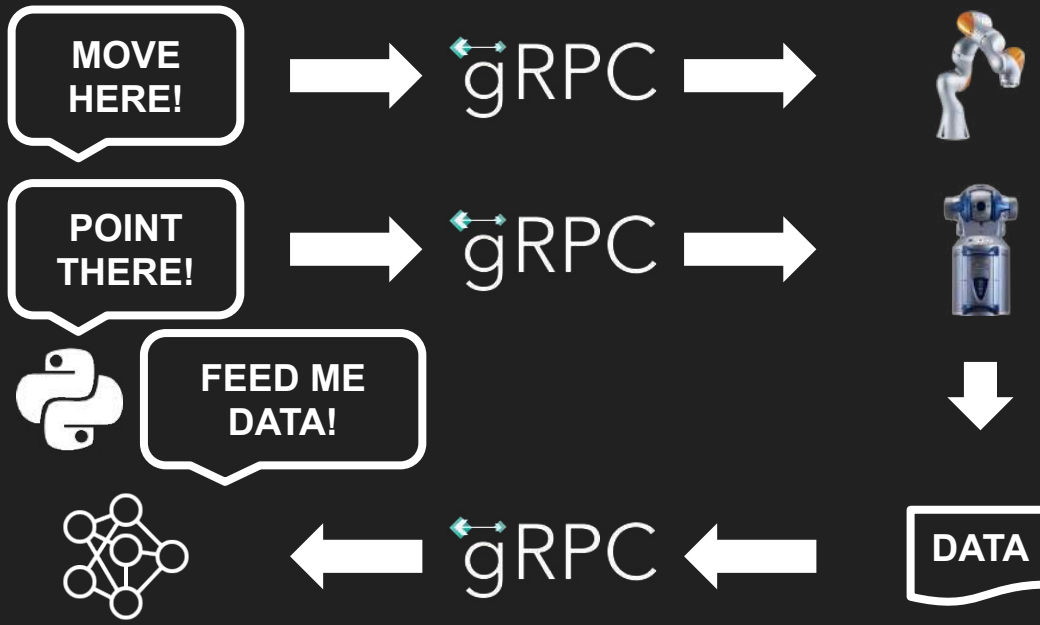
Offline Programming





nadeau innovations

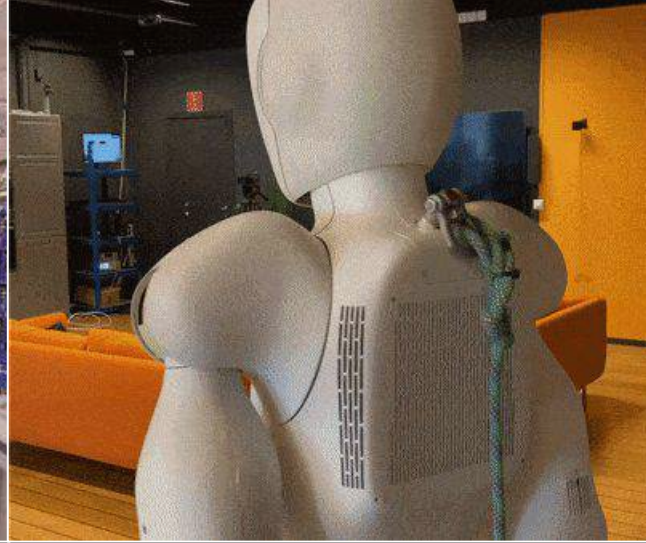
Autonomous Robot Data Collection and Training



nadeau innovations

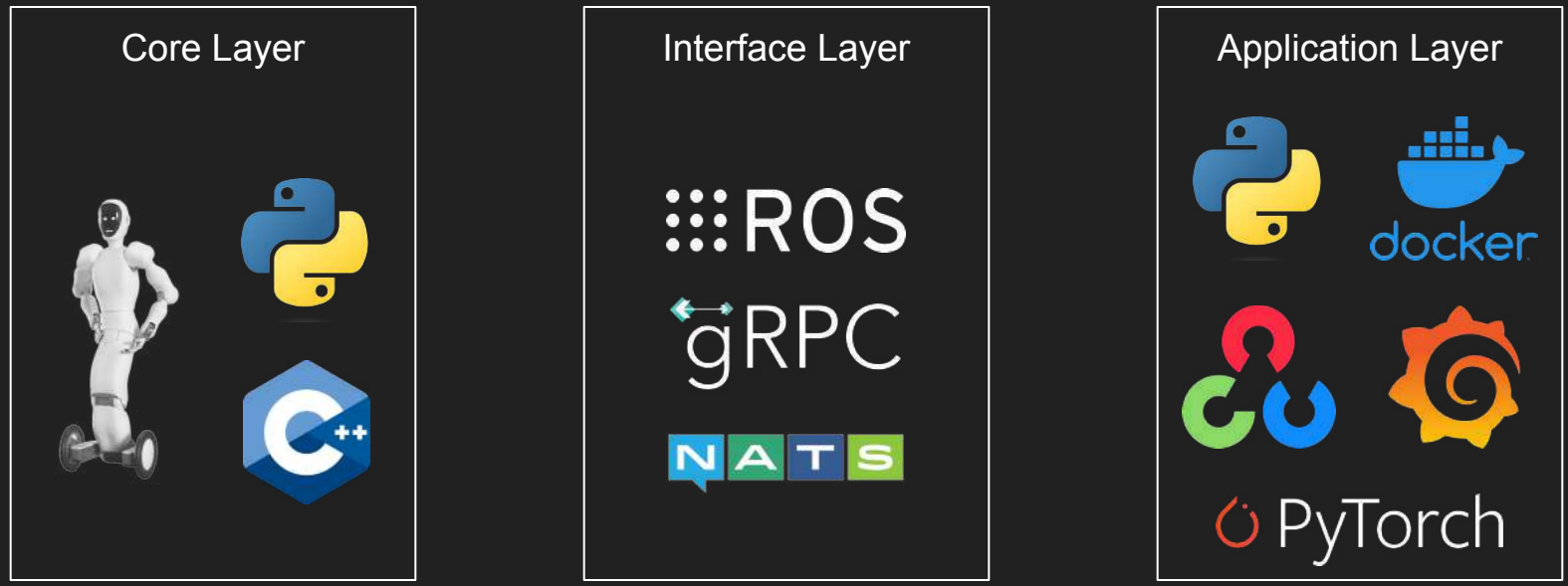

Halodi Robotics





VR Teleoperation and Autonomous Navigation and Interaction

Architecture



nadeau innovations

Industry 4.0 Demo at IPACK-IMA 2022 in Milan

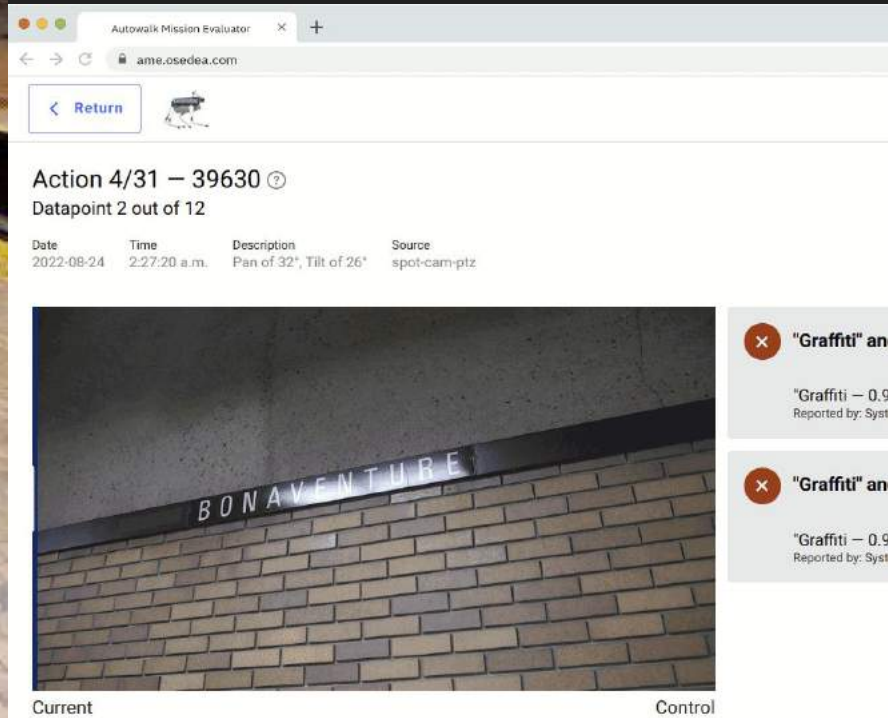
nadeau innovations

Nadeau Innovations

Osedea

MTL BLOG

ICI EXPLORA



nadeau innovations

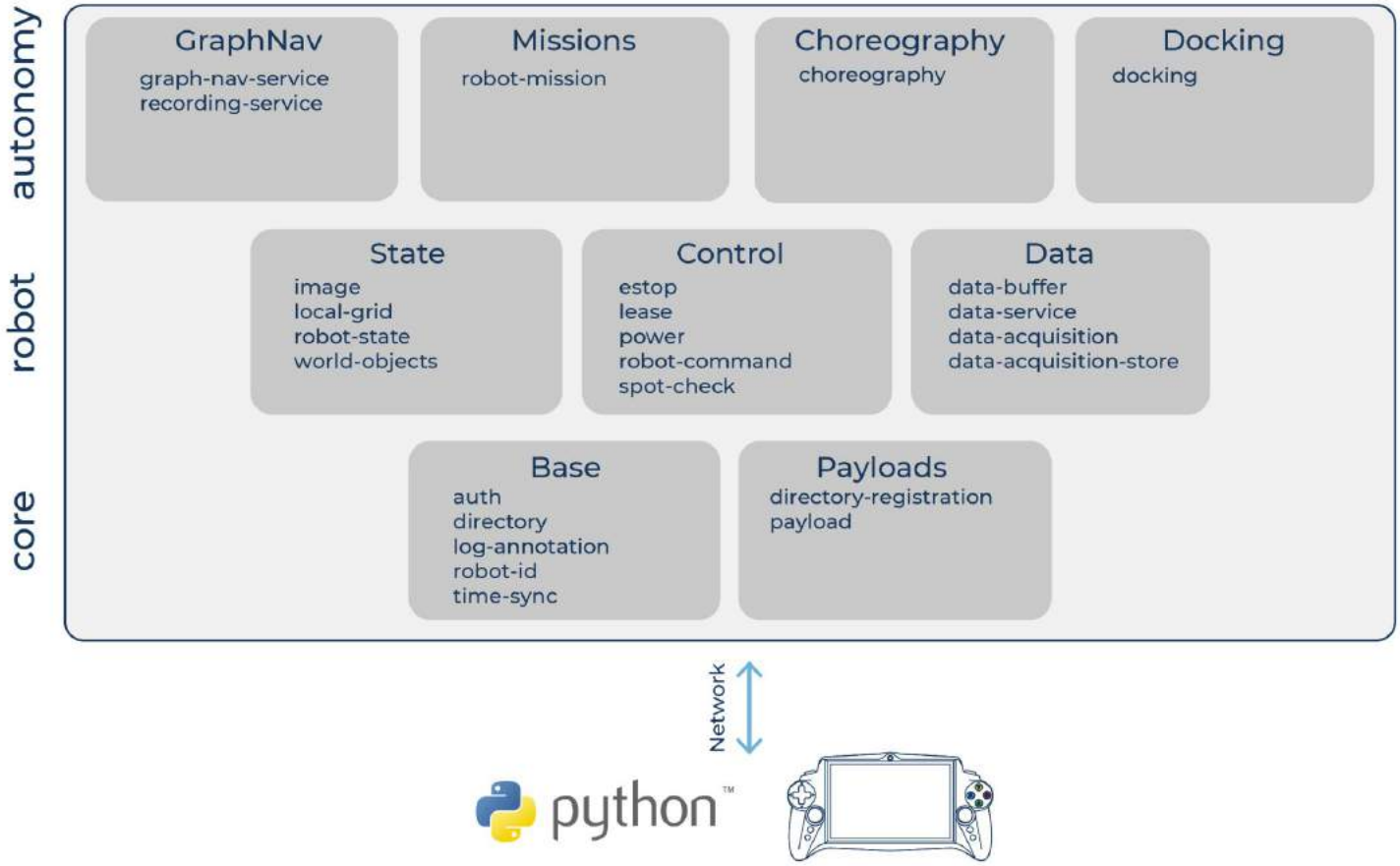
Spot Development

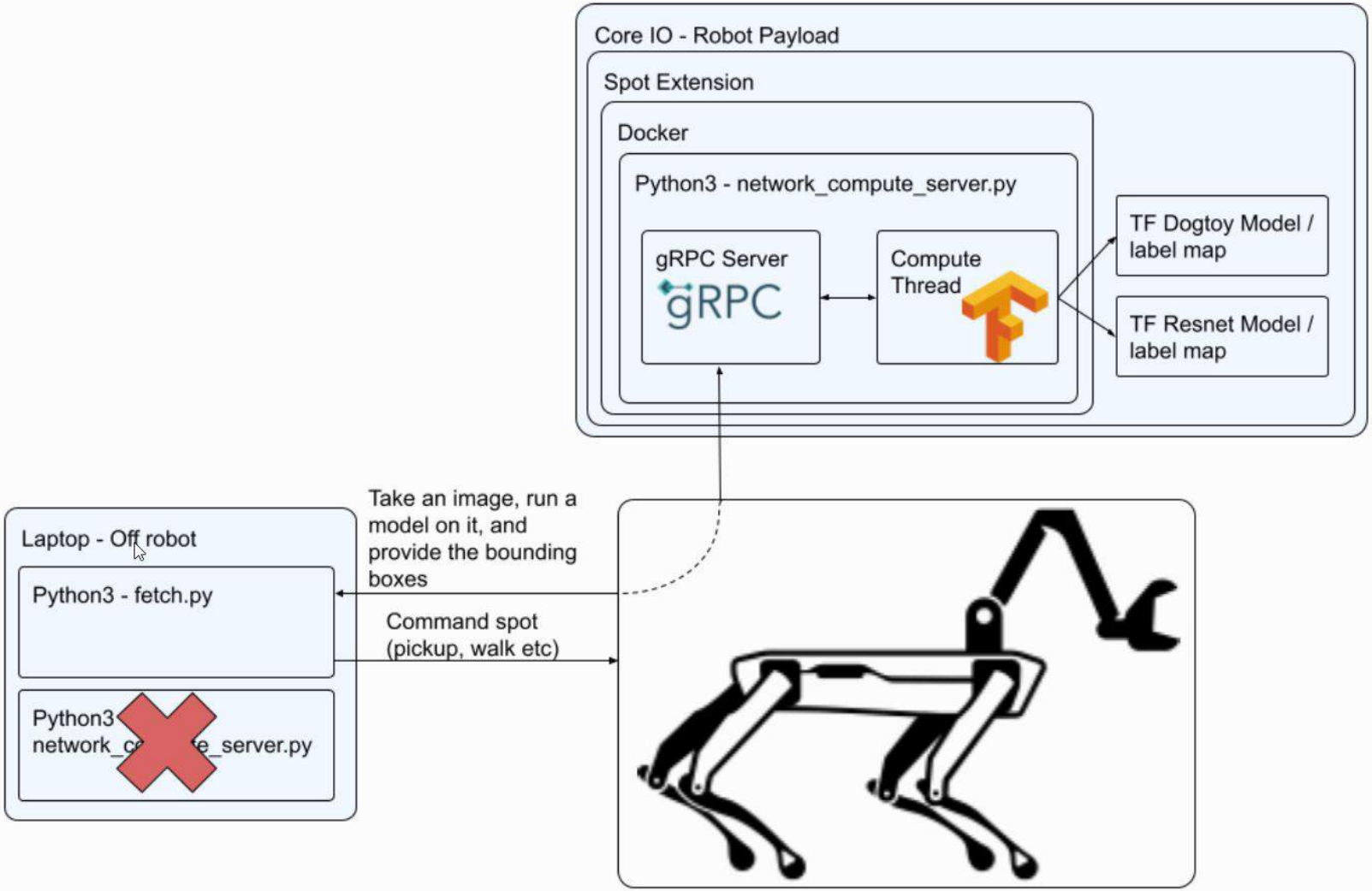
nadeau innovations

```python
import time

import bosdyn.client
import bosdyn.client.util
from bosdyn.client.image import ImageClient
from bosdyn.client.robot_command import (RobotCommandBuilder,
                                         RobotCommandClient, blocking_stand)

config = {}

# The Boston Dynamics Python library uses Python's logging module
bosdyn.client.util.setup_logging(config.verbose)

# The SDK object is the primary entry point to the Boston Dynamics API.
sdk = bosdyn.client.create_standard_sdk('HelloSpotClient')

# A Robot object represents a single robot.
robot = sdk.create_robot(config.hostname)

# Clients need to authenticate to a robot before being able to use it.
bosdyn.client.util.authenticate(robot)

# Establish time sync with the robot. This kicks off a background thread to establish time sync.
robot.time_sync.wait_for_sync()

# Verify the robot is not estopped and that an external application has registered
assert not robot.is_estopped(), "Robot is estopped. Please use an external E-Stop client, " \
                                "such as the estop SDK example, to configure E-Stop."
```

```python
# Only one client at a time can operate a robot.
lease_client = robot.ensure_client(bosdyn.client.lease.LeaseClient.default_service_name)
with bosdyn.client.lease.LeaseKeepAlive(lease_client, must_acquire=True, return_at_exit=True):
    # Now, we are ready to power on the robot.
    robot.logger.info("Powering on robot... This may take several seconds.")
    robot.power_on(timeout_sec=20)
    assert robot.is_powered_on(), "Robot power on failed."
    robot.logger.info("Robot powered on.")

    # Tell the robot to stand up.
    robot.logger.info("Commanding robot to stand...")
    command_client = robot.ensure_client(RobotCommandClient.default_service_name)
    blocking_stand(command_client, timeout_sec=10)
    robot.logger.info("Robot standing.")
    time.sleep(3)

    # Tell the robot to stand in a twisted position.
    footprint_R_body = bosdyn.geometry.EulerZXY(yaw=0.4, roll=0.0, pitch=0.0)
    cmd = RobotCommandBuilder.synchro_stand_command(footprint_R_body=footprint_R_body)
    command_client.robot_command(cmd)
    robot.logger.info("Robot standing twisted.")
    time.sleep(3)

    # Capture an image. Spot has five sensors around the body.
    image_client = robot.ensure_client(ImageClient.default_service_name)
    image_response = image_client.get_image_from_sources(['frontleft_fisheye_image'])

    # Power the robot off.
    robot.power_off(cut_immediately=False, timeout_sec=20)
    assert not robot.is_powered_on(), "Robot power off failed."
    robot.logger.info("Robot safely powered off.")
```

Spot


autonomy
robot
core
GraphNav
graph-nav-service
recording-service
Missions
robot-mission
Choreography
choreography
Docking
docking
State
image
local-grid
robot-state
world-objects
Control
estop
lease
power
robot-command
spot-check
Data
data-buffer
data-service
data-acquisition
data-acquisition-store
Base
auth
directory
log-annotation
robot-id
time-sync
Payloads
directory-registration
payload
Network
python™


Core IO - Robot Payload
Spot Extension
Docker
Python3 - network_compute_server.py
gRPC Server
gRPC
Compute Thread
TF Dogtoy Model / label map
TF Resnet Model / label map
Take an image, run a model on it, and provide the bounding boxes
Command spot (pickup, walk etc)
Laptop - Off robot
Python3 - fetch.py
Python3
network_c
e_server.py

```
# Use a base image provided by nvidia that already contains tensorflow 2.7
FROM nvcr.io/nvidia/l4t-tensorflow:r32.7.1-tf2.7-py3

# Do some basic apt and pip updating
RUN apt-get update && \
    apt-get install -y --no-install-recommends python3-pip && \
    apt-get clean

# Copy over the python requirements file and our prebuilt models API library
COPY docker-requirements.txt ./
COPY models-with-protos models-with-protos

# Install the python requirements
RUN python3 -m pip install pip==21.3.1 setuptools==59.6.0 wheel==0.37.1 && \
    python3 -m pip install -r docker-requirements.txt --find-links .

# Copy over our main script
COPY network_compute_server.py /app/
WORKDIR /app

# Set our script as the main entrypoint for the container
ENTRYPOINT ["python3", "network_compute_server.py"]
```


What’s next?

Nicholas Nadeau,
Ph.D., P.Eng.

@EngNadeau

linkedin.com/in/EngNadeau

NadeauInnovations.com

nadeau innovations