



[www.chameleoncloud.org](http://www.chameleoncloud.org)

# CHI@Edge: Enabling Edge to Cloud Experimentation

**Michael Sherman**

*shermanm@uchicago.edu*

*Feb 17th, 2026*



# Chameleon: An Edge to Cloud Testbed



- ▶ Chameleons like to change – testbed that adapts to your experimental needs
  - ▶ From bare metal reconfigurability/isolation -- KVM cloud – to containers for edge (CHI@Edge)
  - ▶ Capabilities: power on/off, reboot, custom kernel boot, serial console access, etc.
- ▶ From large to small – diversity and scale in hardware:
  - ▶ Supercomputing datacenters (UC/ALCF, TACC, NCAR) over 100G network – to edge devices
  - ▶ Diverse: FPGAs, GPUs, NVMe, NVDIMMs, Edge devices via CHI@Edge, etc.
- ▶ Based on mainstream open source – proud to be cheap!
  - ▶ 50% leveraging and influencing **OpenStack** + 50% “special sauce” (incl. fed id)
- ▶ Promoting digital artifact sharing
  - ▶ Integration with **Jupyter** for non-transactional experiment packaging
  - ▶ **Trovi** for experiment sharing and discovery, **Chameleon Daypass** for access sharing
  - ▶ Reproducibility and education: digital sharing killer apps!





800+  
Papers  
published

1,200+  
Unique  
projects

12,000+  
Users



# Chameleon Hardware

## CHI@UC

Skylake, CascadeLake, IceLake, AMD Zen3, and GigaO  
GPUs (A100, RTX6000), FPGAs (Xilinx Alveo U280)  
Storage (NVMe SSD, NVDIMM)  
Network (25G Ethernet, 200G InfiniBand)

## CHI@Edge

Raspberry Pi 4, 5, Jetson Nano, Jetson Xavier NX, AGX Orin  
**+ User-added devices**

## CHI@TACC

Haswell, SkyLake, CascadeLake, Fugaku (ARM64 + HBM),  
AMD Zen3, and LIQID/GigaO disaggregated hardware  
GPU (H100, A100, AMD MI100),  
Storage (NVMe SSD)  
Network (25G Ethernet, 200G InfiniBand)

Storage  
~0.5 PB

Network  
100Gbps public  
network

Storage  
~3.5PB

Commercial Clouds  
via **CloudBank**

**FABRIC, FAB**  
and other partners

**Chameleon  
Volunteer Sites**



**CHI@NCAR**



Chicago  
Austin

# Experiment Workflow

## Create experimental environment

discover  
resources

- Fine-grained
- Complete
- Up-to-date
- Versioned
- Verifiable

allocate  
resources

- Allocatable resources:  
nodes, VLANs, IPs
- Advance reservations  
and on-demand
- Non-fungible
- Isolation

configure  
environment

- Bare metal, KVM,  
containers
- Image catalog
- Snapshotting
- Network stitching
- Orchestration

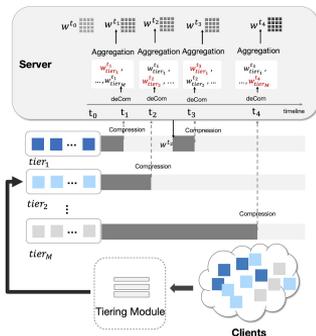
Run  
experiment  
body

analyze and  
present  
results

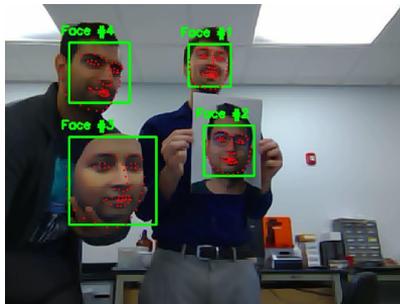
*Authentication via federated identity, accessed via GUI, CLI, and python-chi*

*Paper: "Lessons Learned from the Chameleon Testbed", USENIX ATC 2020*

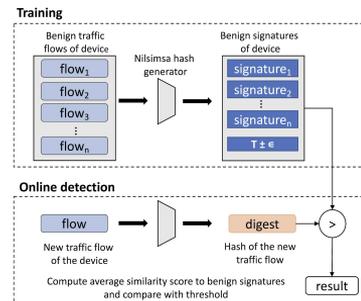
# From Cloud to Edge with Chameleon



*federated learning*



*biometrics*



*network traffic fingerprinting for IoT devices*

- ▶ Increasingly more Chameleon project applications working on IoT/edge
- ▶ Simulation/emulation don't always provide the answer: What are the impacts of this approach on power management on edge device? How will the performance transfer to edge? Can we measure the impact of distribution/networking for edge/cloud applications?
- ▶ **Goal: "realistic edge to cloud experiments from one Jupyter notebook"**

# Programmable with CHI@Edge



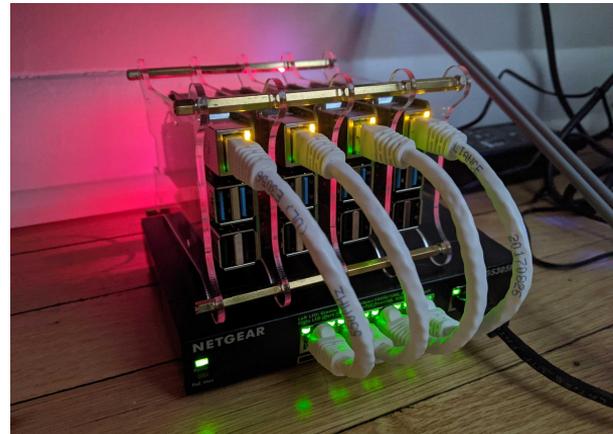
A lot like a cloud!  
All the features we know  
and love – but for edge!

Not at all like a cloud!  
Location, location, location!  
IoT: cameras, actuators, SDRs!  
Not server-class!  
And many other challenges!



- ▶ CHI@Edge: all the features you love in CHI, plus:
  - ▶ Reconfiguration through non-prescriptive **container deployment** via OpenStack interfaces (using K3 under the covers)
  - ▶ Support for “standard” **IoT peripherals** (camera, GPIO, serial, etc.) + support for your own peripherals
  - ▶ **Bring Your Own Device (BYOD): Mixed ownership** model via an SDK with devices, virtual site, and **restricted sharing** – building on OpenBalena

*Paper: “CHI@Edge: Supporting Experimentation in the Edge to Cloud Continuum” PEARC 2025*



# CHI@Edge Hardware

## Current Devices

- Raspberry Pi 4
- Raspberry Pi 5
- Jetson Nano
- Jetson Xavier NX

## Coming Soon

- Jetson Xavier AGX Orin
- Jetson Orin Nano

## Peripherals

- [Raspberry Pi Sense Hat](#)
- [Waveshare Sense Hat B](#)

## Camera Support

- [Pi Camera 2.1](#)
- [Pi Camera 3](#)
- Wide Angle OV5647
- Anything “libcamera”

# Core workflow and usage

- Updated Docs: <https://chameleoncloud.gitbook.io/chi-edge>
- Device Types:  
<https://chameleoncloud.gitbook.io/chi-edge/hardware-info/device-type>
- Peripherals and Usage:  
<https://chameleoncloud.gitbook.io/chi-edge/hardware-info/peripheral-types>
- Live Hardware browser:  
<https://www.chameleoncloud.org/edge/hardware/>

# Demos and Artifact Examples

- Camera Peripherals:  
<https://trovi.chameleoncloud.org/dashboard/artifacts/7d35f884-68d8-4b91-b42b-207717c9b742>
- Sense Hats and GPIO:  
<https://trovi.chameleoncloud.org/dashboard/artifacts/1b8cdcba-e10c-4cae-8b1e-145037bc4cda>
- SSH and interactivity:  
<https://trovi.chameleoncloud.org/dashboard/artifacts/4b08e80c-df0a-4866-bf29-19b5ad6a2950>

# Best Practices: Interactivity

- Keep “state” external to your container
  - run “exec” from a notebook
  - Execute scripts via ssh
- Verify assumptions
  - Run ls, are the right files there?
  - Using a peripheral, does it return sane results?
- Installing dependencies is “expensive”, bake it into your source image.

# Best Practices: Container Images

- Large images can take a long time to download
- Docker best practices work here too
  - Only install what you need
  - Multi-stage builds
- Use our images as a base, or as examples

```
FROM python:3.13.5-bookworm as builder
```

```
WORKDIR /app
COPY src/requirements.txt .
RUN pip wheel \
    --no-cache-dir \
    --no-deps \
    --wheel-dir /app/wheels \
    -r requirements.txt
```

```
# Final image
FROM python:3.13.5-slim-bookworm
```

```
WORKDIR /app
COPY --from=builder /app/wheels /wheels
RUN pip install --no-cache /wheels/*
```

```
COPY src/ .
ENTRYPOINT ["python"]
CMD ["src/main.py" ]
```

# Adding your own Peripherals

## Currently:

1. Find out what it needs from `/dev`
2. Make a PR for a new device profile

## Examples:

- [Container Artifact Source](#)
- [SenseHat Artifact Source](#)

**In Progress:** Making this self-service.

We want your feedback!

Big questions:

- What kind of peripherals?
- What dependencies does it have?
- How do we know they works?



*We're here to change*

[www.chameleoncloud.org](http://www.chameleoncloud.org)