

AUTOMATED GRAPHICAL TESTING ON REAL HARDWARE: ADVENTURES WITH OPENQA

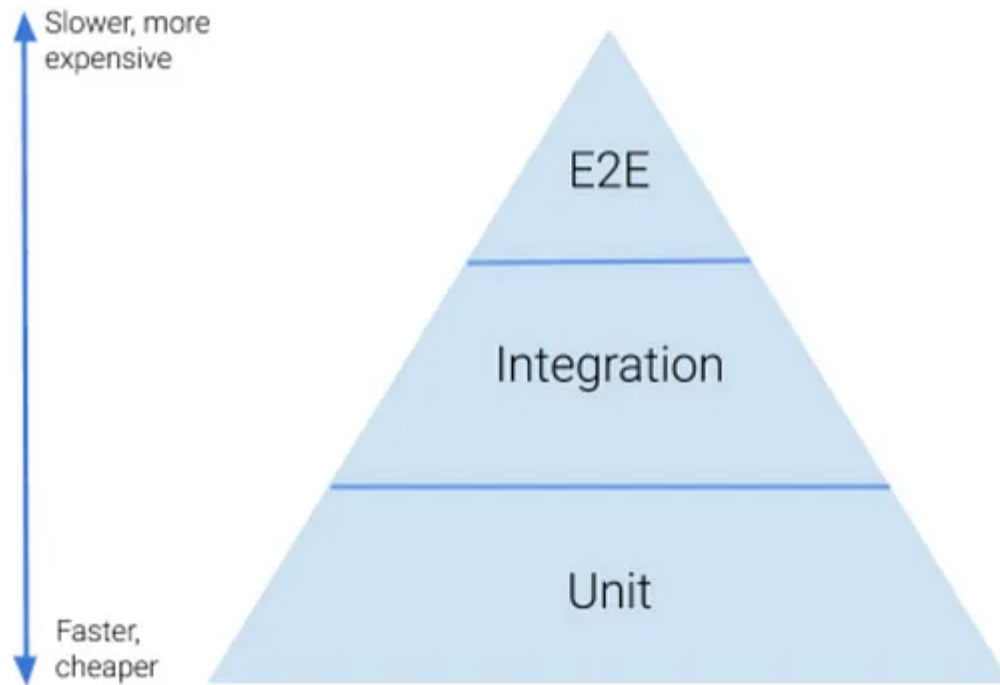
Sam Thursfield
XDC 2023



HELLO

I am

- senior software developer @ Codethink
- foundation member and maintainer @ GNOME



2 UNIT TESTS



NO INTEGRATION TESTS

OPENQA

Automated end-to-end testing of...

- **Desktop** operating systems
- **Phone** operating systems
- **Car** operating systems
- ...basically **anything with a screen**

THE PROJECT

- Open source (GPL-2.0-or-later)
- Community-driven development
- Paid maintainer team (funded by SUSE)
- "Continuous release" model (no "stable" version)

THE TOOL

- Strong support for **screenshot testing**
 - Fuzzy region matching (using openCV)
 - Graphical UI for updating screenshots
- Multiple **backends** for virtual and physical hardware
- Hackable Perl codebase!

EXAMPLE: GNOME OS TESTS

Let's try and run them locally...

```
rm -r ./out; env ssam_openqa run --tests-path . \  
  --hdd-path ./gnome_os_disk.latest.20230831.img \  
  --iso-path ./gnome_os_installer_525758.iso \  
  -o ./out
```


RUNNING ISOTOVIDEO CONTAINER

ssam_openqa is a CLI helper tool.

It wraps long Podman commands:

```
podman run --name ssam_openqa_gnome_apps \
  --privileged --detach \
  --volume=$(pwd)/gnome_os_disk.latest.20230831.img:/disk.img \
  --volume=$(pwd):/tests \
  --volume=$(pwd)/out/gnome_apps:/shared \
  --entrypoint isotovideo \
  --publish 5990 \
  --publish 20013 \
  -- \
  registry.opensuse.org/devel/openqa/containers15.4/openqa_worker:latest \
  --workdir=/shared ARCH=x86_64 \
  ASSETDIR=/var/lib/openqa/share/factory/ \
  BACKEND=qemu \
  ...
```

os-autoinst is also packaged in distros - but beware "rolling release" versioning.

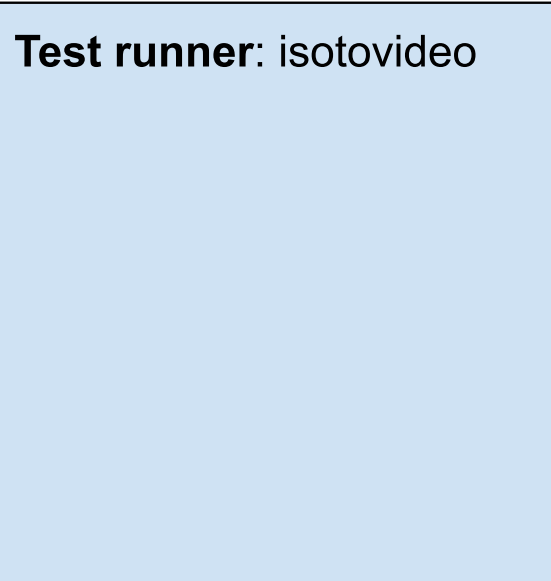
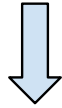
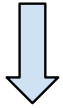
HOW IT WORKS

Assets: OS image, ... **Tests** and **needles**

HOW IT WORKS

Assets: OS image, ...

Tests and needles

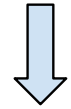


Test runner: isotovideo

HOW IT WORKS

Assets: OS image, ...

Tests and needles

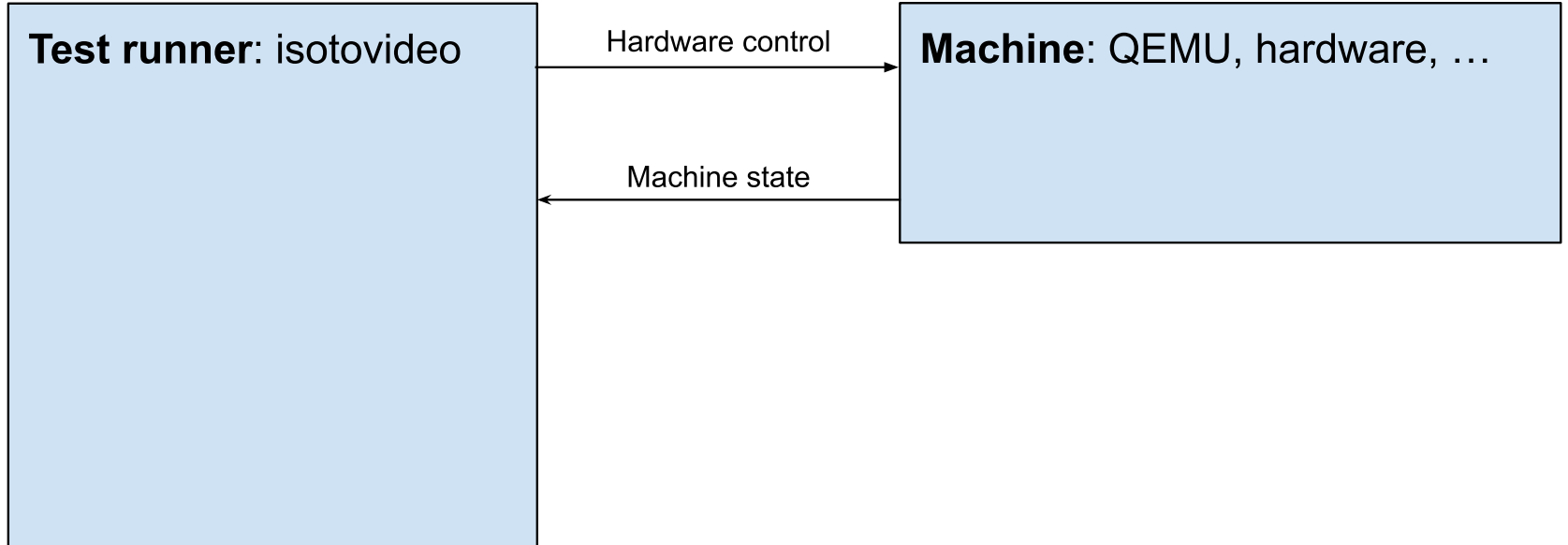


Test runner: isotovideo

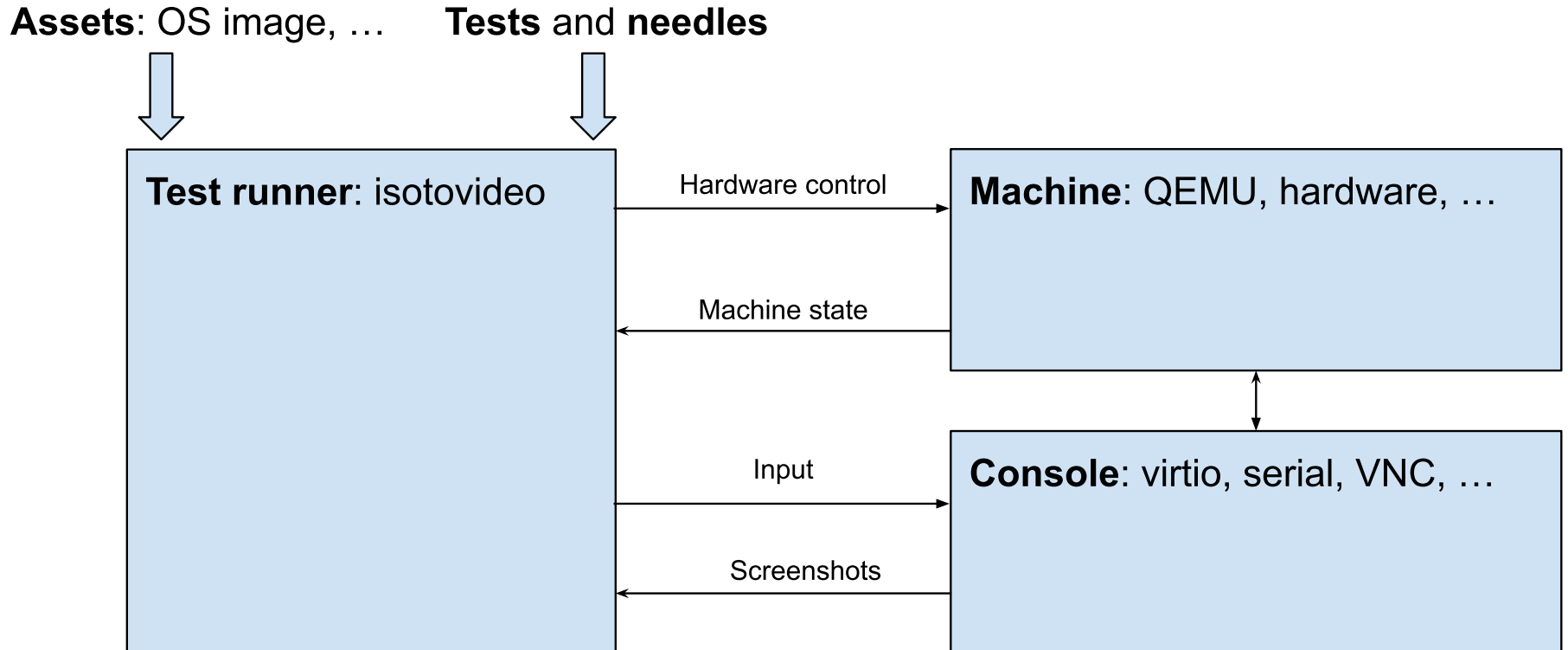
Hardware control

Machine: QEMU, hardware, ...

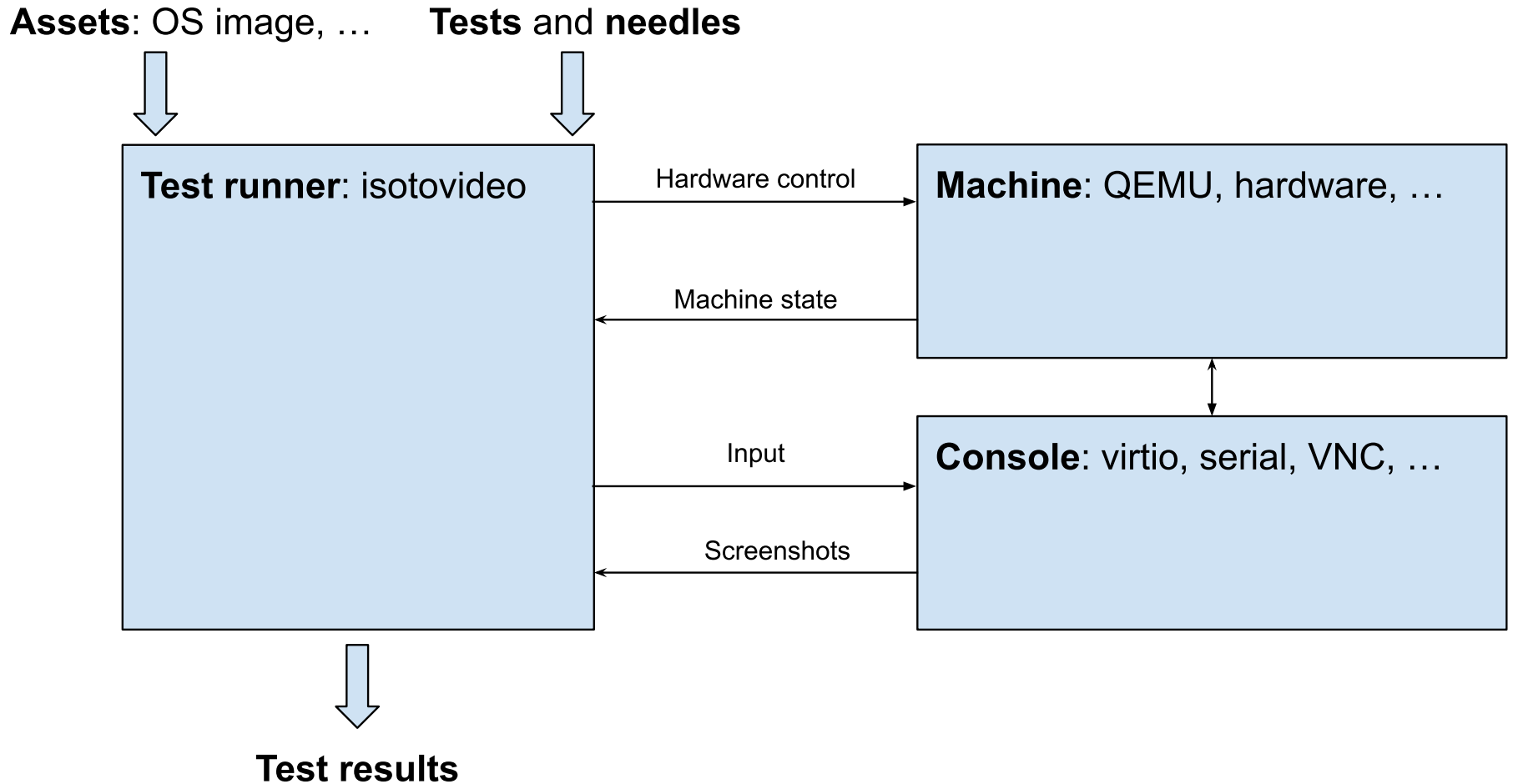
Machine state



HOW IT WORKS



HOW IT WORKS



INTEGRATING INTO CI

Two options:

- Permanent workers, managed by **openQA server**
- Transient workers, e.g. on a **Gitlab CI runner**

openSUSE use openQA to manage workers.

GNOME uses transient runners on [Gitlab CI](#).

THE OPENQA WEB UI

Let's see this online!

Screenshot tests will always have false positives.

openQA deals with this in 4 ways:

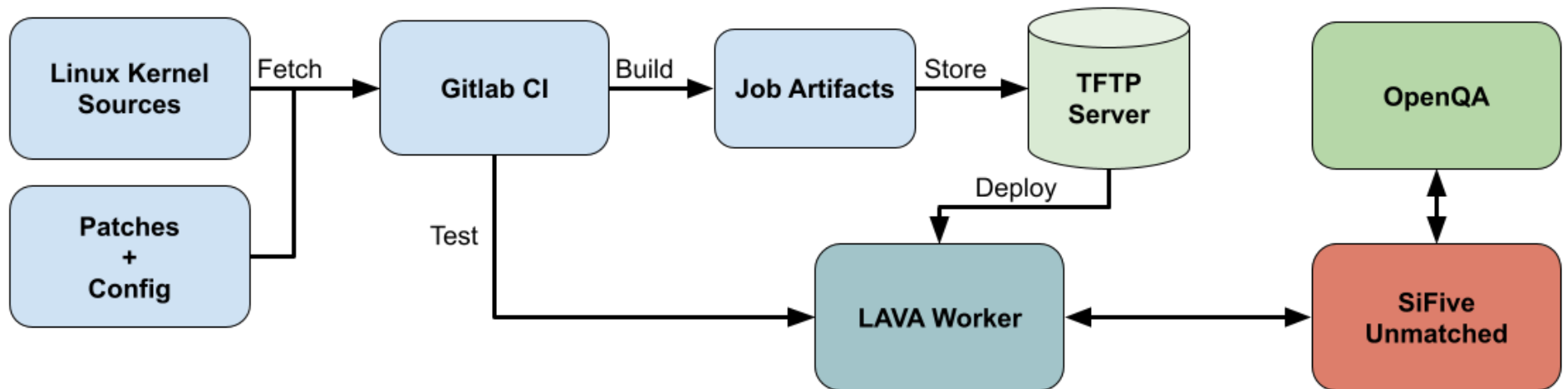
1. Search within the screen
2. Similarity threshold (90-100%)
3. Exclude zones
4. Web UI for needle updates.

BEYOND QEMU: TESTING ON HARDWARE



"Permenant worker" and "transient worker" approaches are possible.

KERNEL TESTING AT CODETHINK



EXAMPLE OF LAVA + OPENQA

See: <http://openqa.qa.codethink.co.uk/>

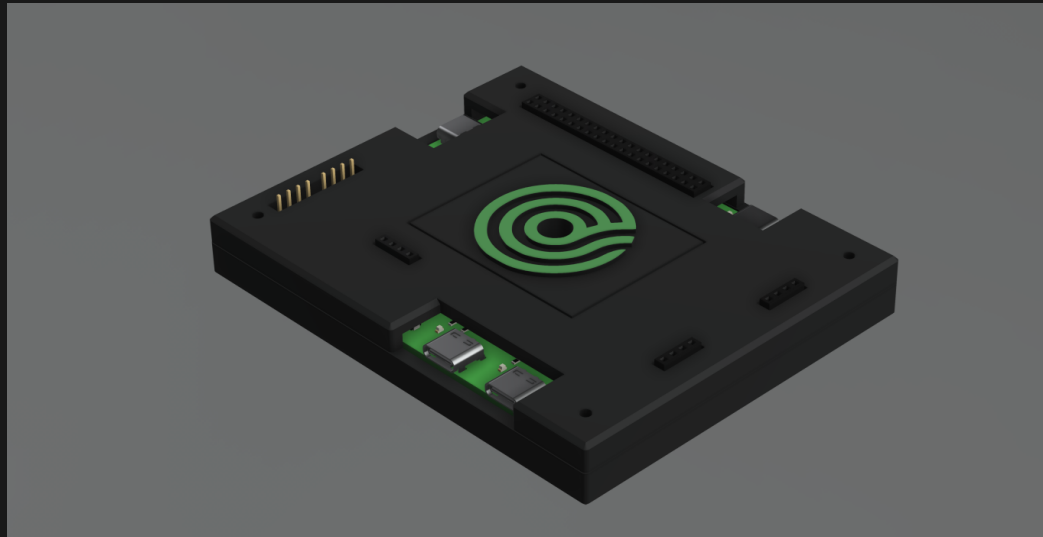
Testing on automotive hardware

How do you remote control a car IVI system?

- ~~virtio devices~~
- ~~VNC~~
- Q.A.D.: lightweight "remote control" daemon

HARDWARE TOOLS

USB-C switcher with computer control



For tests involving phones & USB media

Open hardware, see: <https://gitlab.com/CodethinkLabs/usb-switch>

HARDWARE TOOLS



What to do about the mess??

HARDWARE TOOLS

Testing in a Box



Hardware: *Host PC, serial, CAN emulator, USB Switch + Hub, Bluetooth/WiFi, HID emulation, ...*

Software: *Gitlab + Gitlab CI, openQA worker, ...*

Open hardware, see: <https://gitlab.com/CodethinkLabs/testing-in-a-box>

CODETHINK IS HIRING

openQA: <https://openqa.qa/>

GNOME tests: <https://gitlab.gnome.org/gnome/openqa-tests/>

Codethink projects:

- Code: <https://gitlab.com/CodethinkLabs/>
- Chat: [#codethinklabs:matrix.org](https://matrix.org/#/codethinklabs:matrix.org)

Sam Thursfield
XDC 2023



