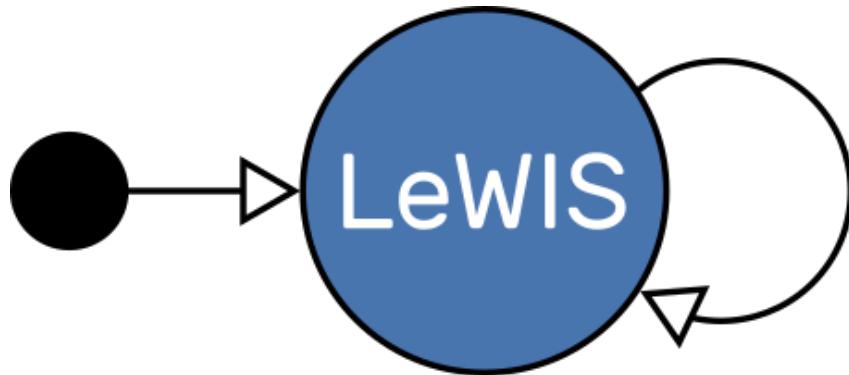


Introduction to...



A Stateful Device Emulation Framework

Owen Arnold

Freddie Akeroyd

Matt Clarke

Michael Hart

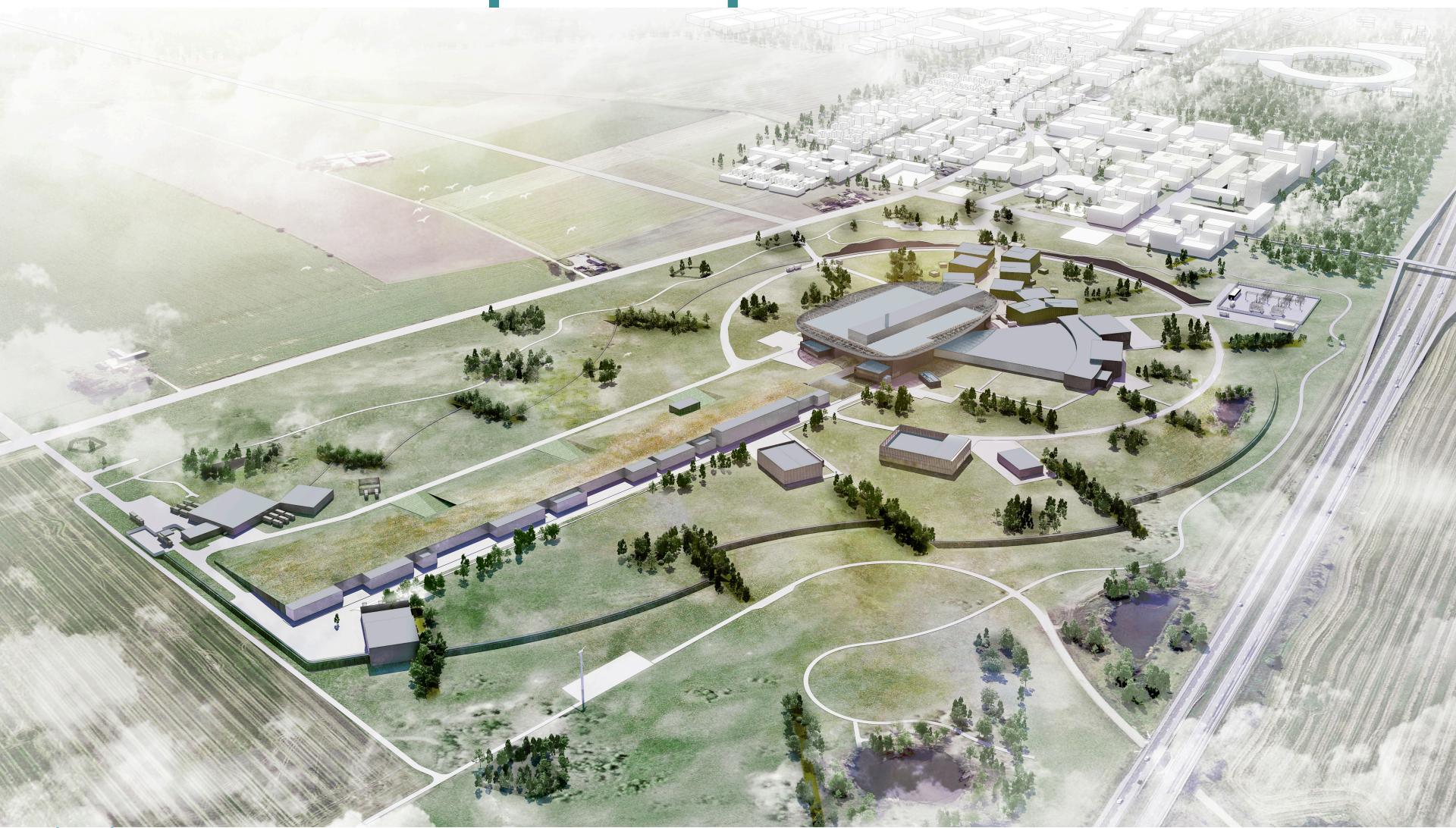
Michael Wedel

Contents

- Background and project drivers
- How LeWIS works
- Features of LeWIS
- Real use-cases from IBEX
- Future plans
- How to use and contribute
- Demos

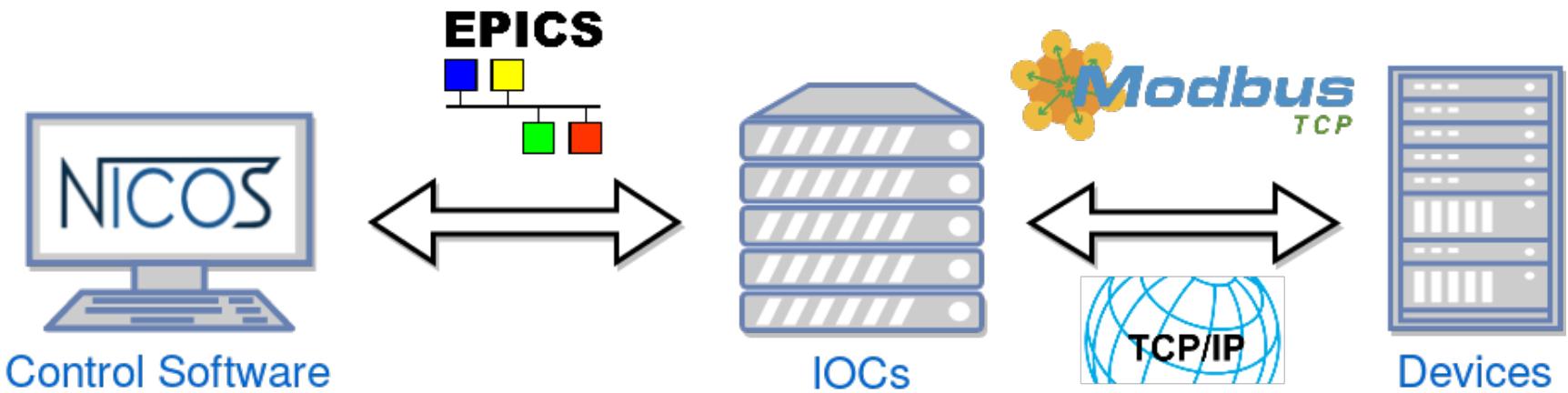


The European Spallation Source



Background

- Need to develop user-facing control software



ESS Artist Rendition



Science & Technology
Facilities Council

ESS Reality



Background

- Need to develop user-facing control software
- BUT....
 - Facility does not exist yet
 - Hardware not available yet
 - EPICS IOCs not available yet
 - Cannot test our work



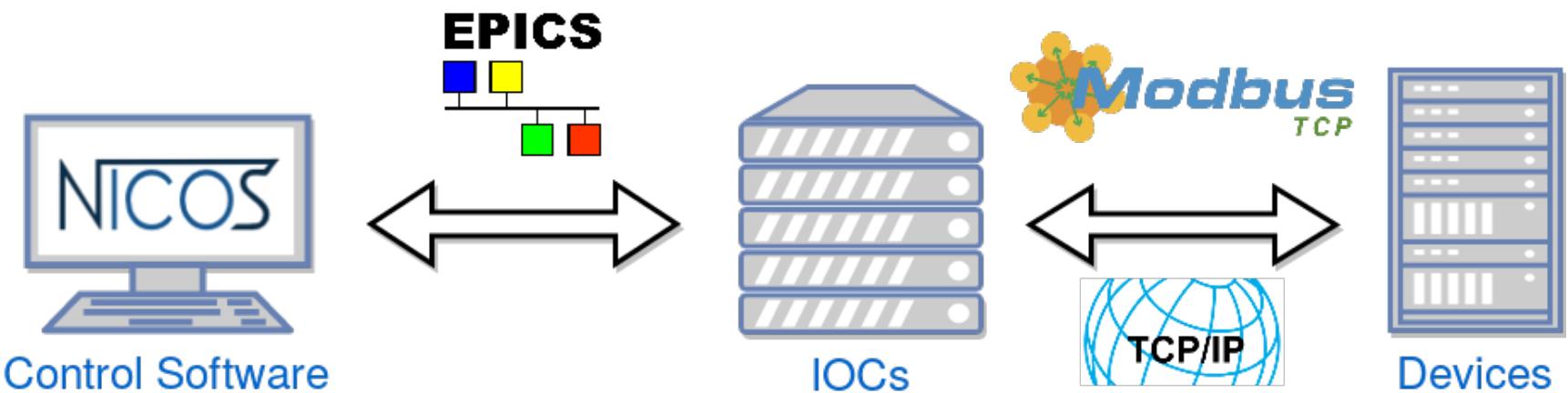
Unacceptable Project Risk



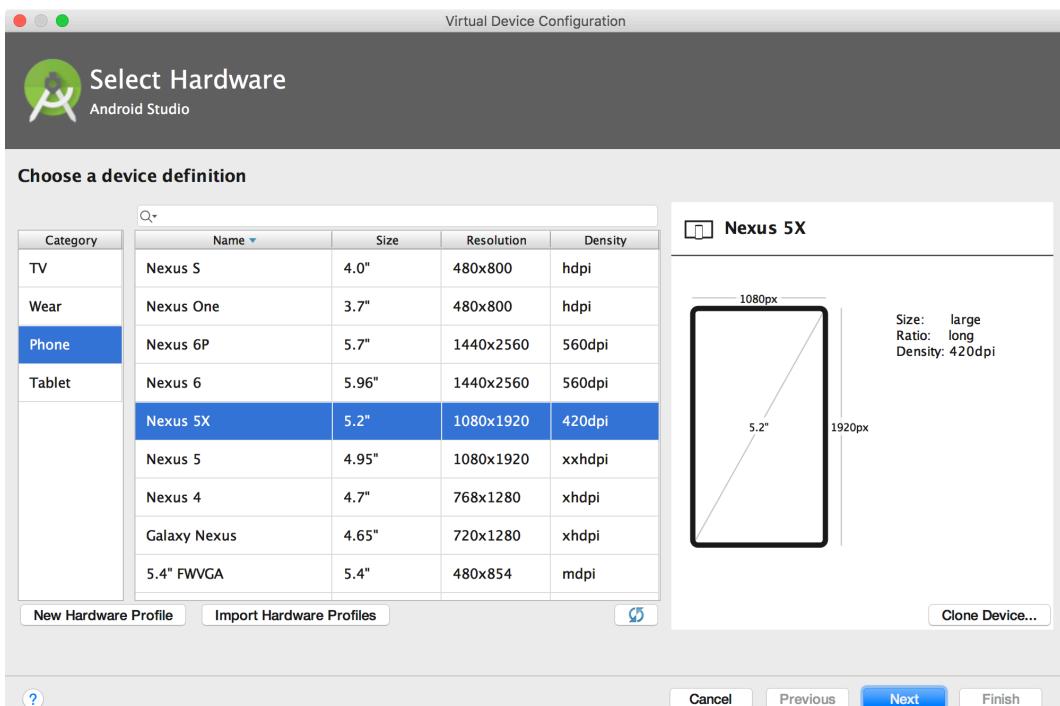
Science & Technology
Facilities Council

Solution

- Replace Devices + IOCs with software emulator



Emulation In Other Industries

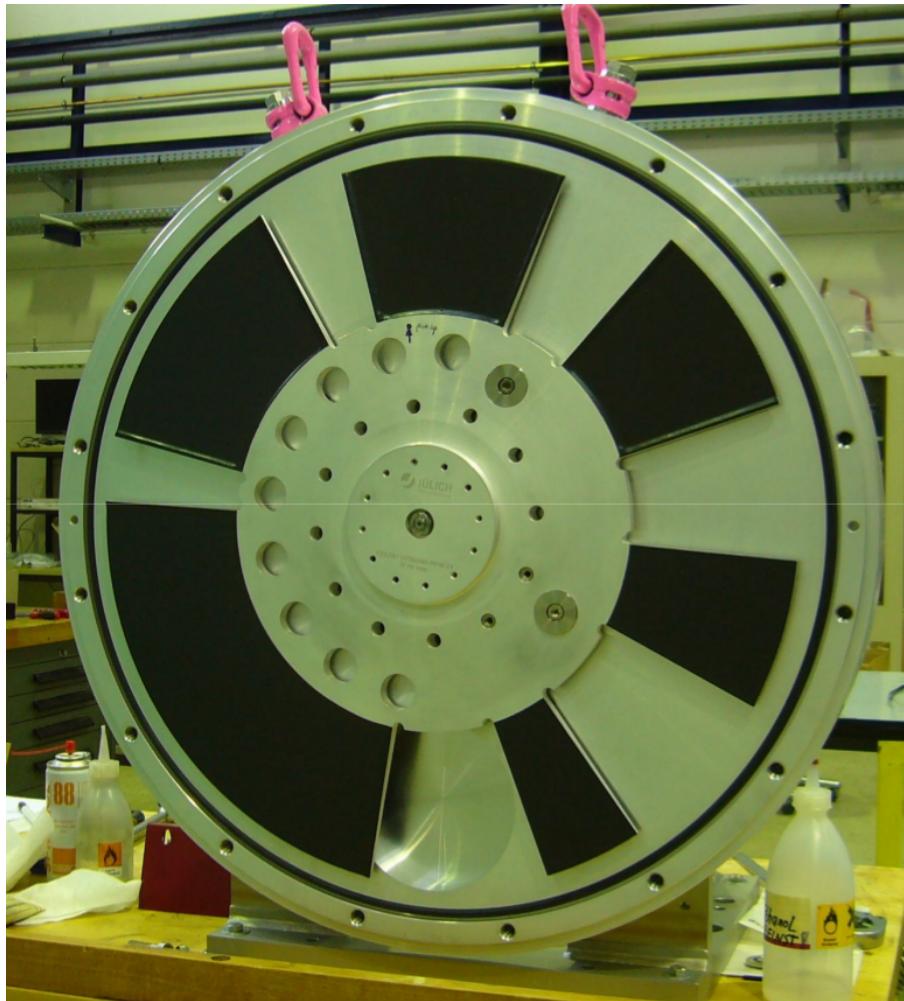


Using Emulators We Could ...

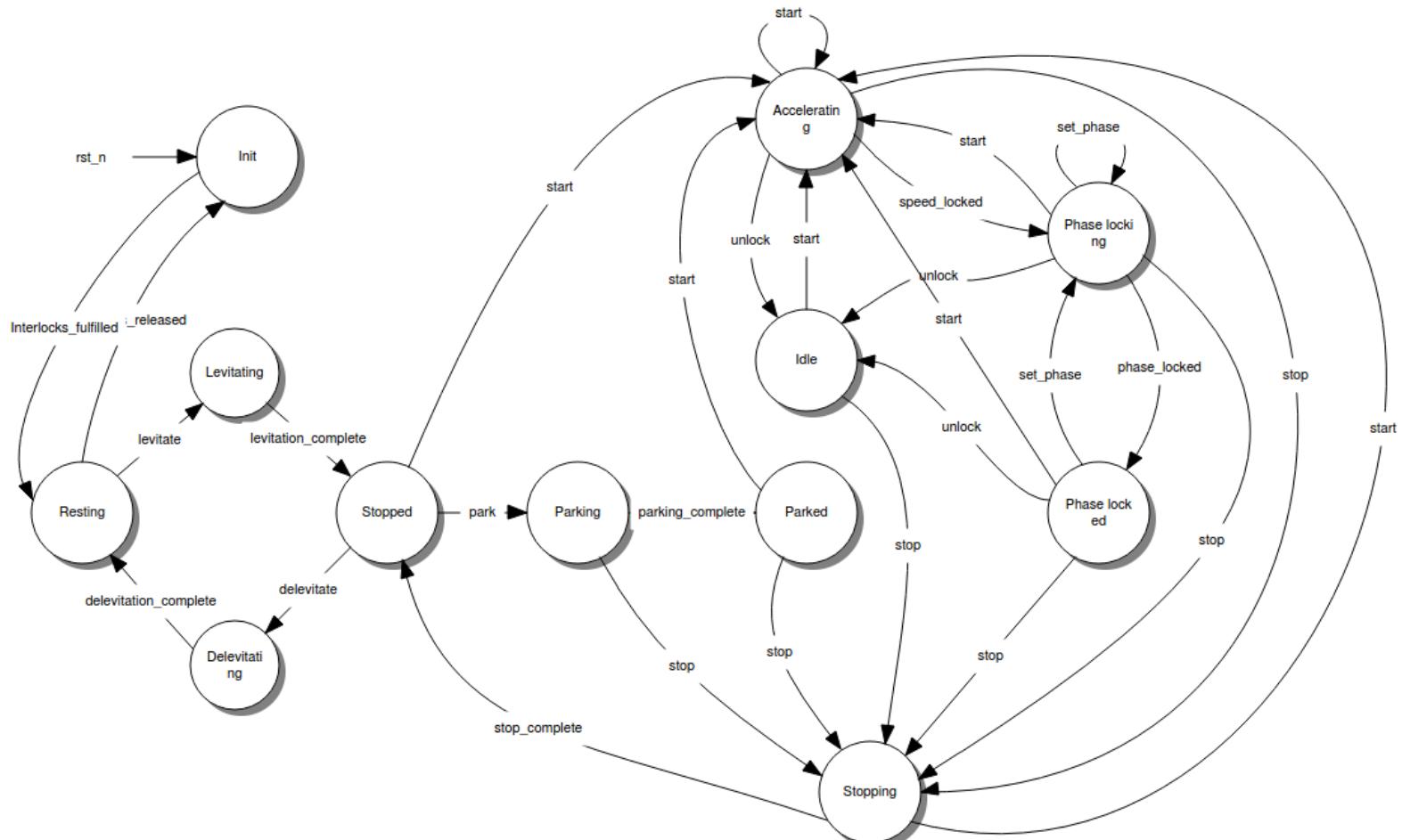
- Development and testing of user-facing software without access to hardware
- Testing edge conditions without stressing or jeopardizing hardware
- Automate test coverage
- Perform dry runs to test or time user scripts in an accelerated time frame.



Real Devices: Choppers



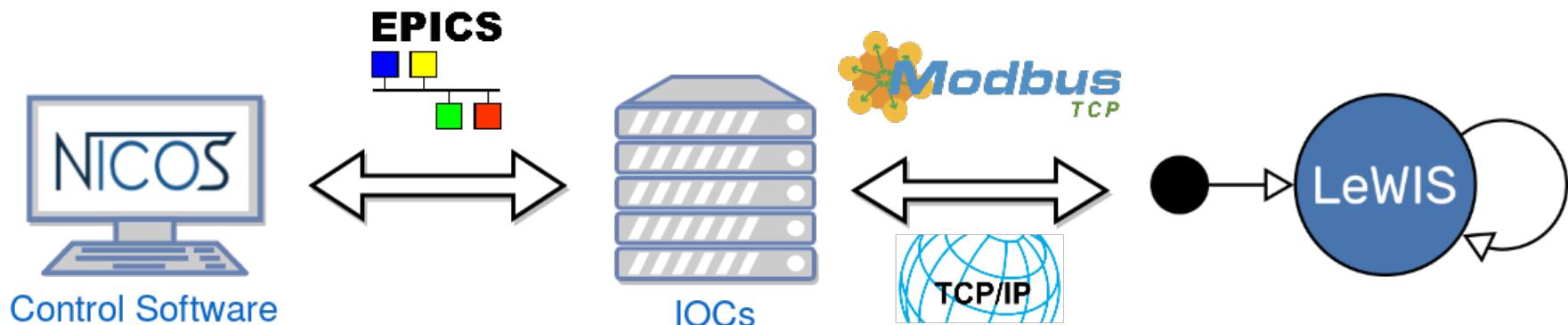
Chopper State Diagram



Read-back is not enough

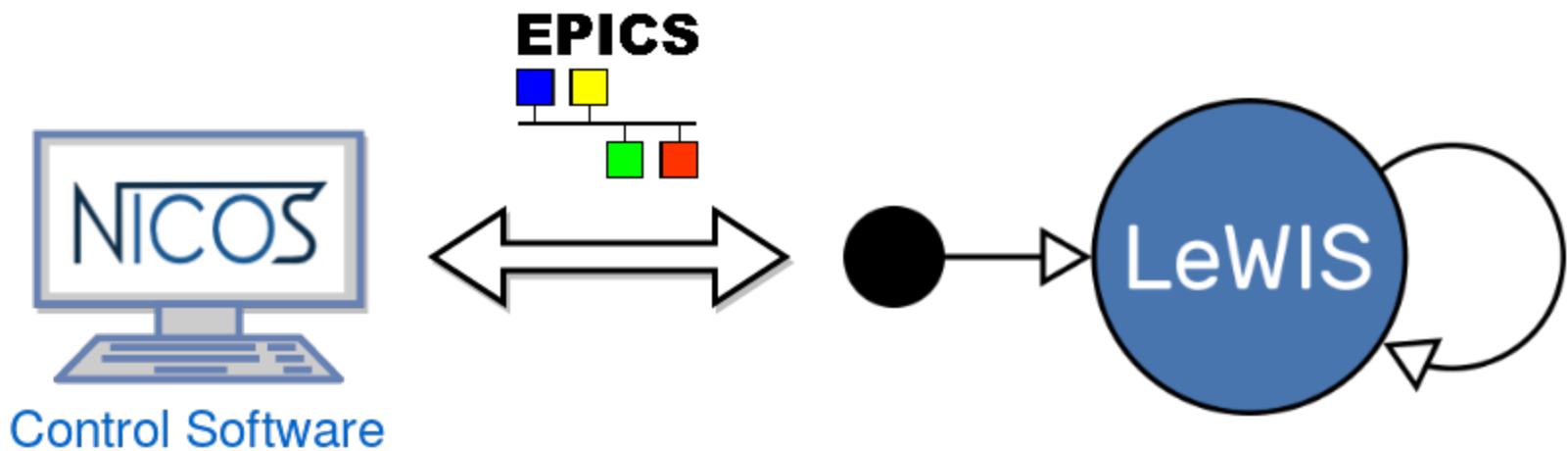
Replace Device

- Replace devices with emulator



Replace IOC

- Replace devices + IOCs with emulator

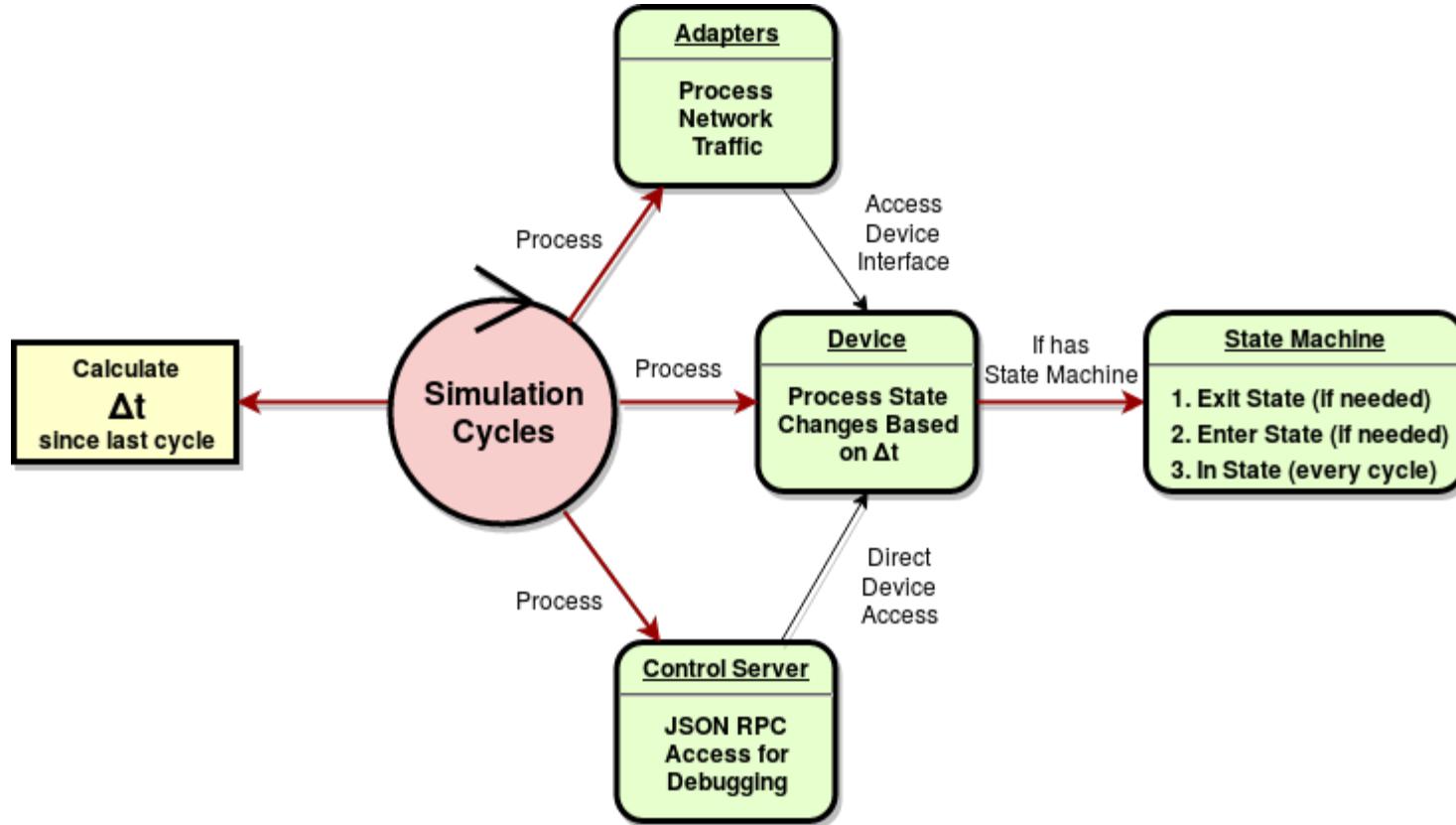


Lewis Overview

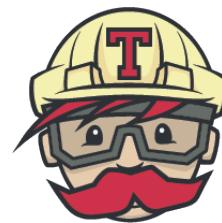
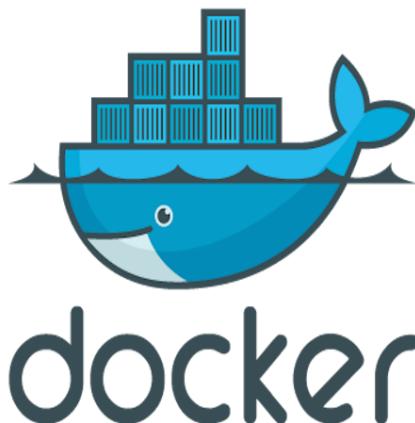
LEt's Write Intricate Simulators

- Python framework to unify common tasks
- State machine to model device states
- Flexible interfaces: IOC or Device-level
- Cycle-driven: Deterministic and controllable time-flow
- JSON-RPC to bypass normal protocol
- Available as Docker image

Lewis Cycle-Driven Design



Lewis Technologies



Current State

- Version 1.0.3 released March 24th 2017
- Helped progress Chopper design at ESS
- In use by IBEX team here at ISIS
 - Device-level emulators. Writing emulators exposed bugs in IOCS!
 - Test IOC and GUI behaviour

IBEX IOC Testing

- Would like to test IOC behaviour
 - Does setting PVs do what you intended
- And re-test it again if it/dependencies change
 - but hardware might not be available then
- Solution: Use a device emulator



ISIS IOC Testing

- Run automatically via a Jenkins job
- Device emulators written using LeWIS
- Tests written using Python unittest framework
- ISIS IOC Test framework then:
 - Defines some “under test” macros for the IOC
 - Starts the IOC
 - Testing framework allocates a TCP port and starts LeWIS
 - Framework runs tests under device simulation
 - Also reruns (subset) under IOC record simulation



https://epics-jenkins.isis.rl.ac.uk/job/System_Tests_IOCs/11/testReport/tests.julabo/JulaboTests/

System_Tests_IOCs #11 Julab... x Computer Name: **Windows-10-0000000000000000**

File Edit View Favorites Tools Help

x Google Search Share More >

Suggested Sites Web Slice Gallery

Find: plus Previous Next Options

Sign In

Jenkins System_Tests_IOCs #11 Test Results tests.julabo JulaboTests

2 search Akeroyd, Freddie (STFC,RAL,ISIS) | log out

ENABLE AUTO REFRESH

 Jenkins

Back to Project Status Changes Console Output Edit Build Information History Git Build Data No Tags Test Result Previous Build

Test Result : JulaboTests

0 failures (± 0), 7 skipped (± 0)

22 tests (± 0) Took 48 sec 

All Tests

Test name	Duration	Status
test_set_new_temperature_sets_setpoint_readback_correctly	10 sec	Passed
test_set_new_temperature_sets_setpoint_readback_correctly	10 sec	Passed
test_set_new_temperature_with_circulate_off_means_temperature_remains_unchanged	0 ms	Skipped
test_set_new_temperature_with_circulate_off_means_temperature_remains_unchanged	2.2 sec	Passed
test_set_new_temperature_with_circulate_on_changes_temperature	3.3 sec	Passed
test_set_new_temperature_with_circulate_on_changes_temperature	3.3 sec	Passed
test_setting_external_PID_above_limit_does_nothing	5 sec	Passed
test_setting_external_PID_above_limit_does_nothing	0 ms	Skipped
test_setting_external_PID_below_limit_does_nothing	4.4 sec	Passed
test_setting_external_PID_below_limit_does_nothing	0 ms	Skipped
test_setting_external_PID_sets_values_correctly	4.4 sec	Passed
test_setting_external_PID_sets_values_correctly	5 sec	Passed
test_setting_internal_PID_above_limit_does_nothing	0 ms	Skipped
test_setting_internal_PID_above_limit_does_nothing	5 sec	Passed
test_setting_internal_PID_below_limit_does_nothing	0 ms	Skipped
test_setting_internal_PID_below_limit_does_nothing	4.4 sec	Passed
test_setting_internal_PID_sets_values_correctly	5 sec	Passed
test_setting_internal_PID_sets_values_correctly	4.4 sec	Passed
test_setting_temperature_above_high_limit_does_not_set_value	0 ms	Skipped

Future Plans

- Release 1.1 scheduled for June 2017
 - Many-One Interfaces to Device mappings
 - Suspend Resume of Devices
 - Maintaining devices in separate package from LeWIS itself

Where to find Lewis

- GitHub
 - <https://github.com/DMSC-Instrument-Data/lewis>
- DockerHub
 - <https://hub.docker.com/r/dmscid/lewis/>
- PyPI
 - <https://pypi.python.org/pypi/lewis>
- Install!
 - `$ pip install lewis`
 - `$ docker pull dmscid/lewis`

Questions

