The EPIC(S) Battle between Individualism and Collectivism

S. Rose, W. Lewis, A. Olsson, J. Liu, European Spallation Source

1. What problem are we trying to solve

2. e3 Definitions

3. Standardised Interface
for
Epics Modules

4. e3 Environment

What is e3 Trying to Do

- Lots of code from many sources
- Trying to keep it all up-to-date
- Trying to keep track of what is being used, by whom
- Maintaining site configuration challenging
- Fix so-called "dependency hell"
- How to ensure that everything is under version control and easy to deploy

e3 Definitions

An IOC:

- Standard executable entrypoint (iocsh.bash, loads softIocPVA)
- Runs a configured startup script st.cmd
- Dynamically loads all necessary functionality
- Generated by external configuration management tool

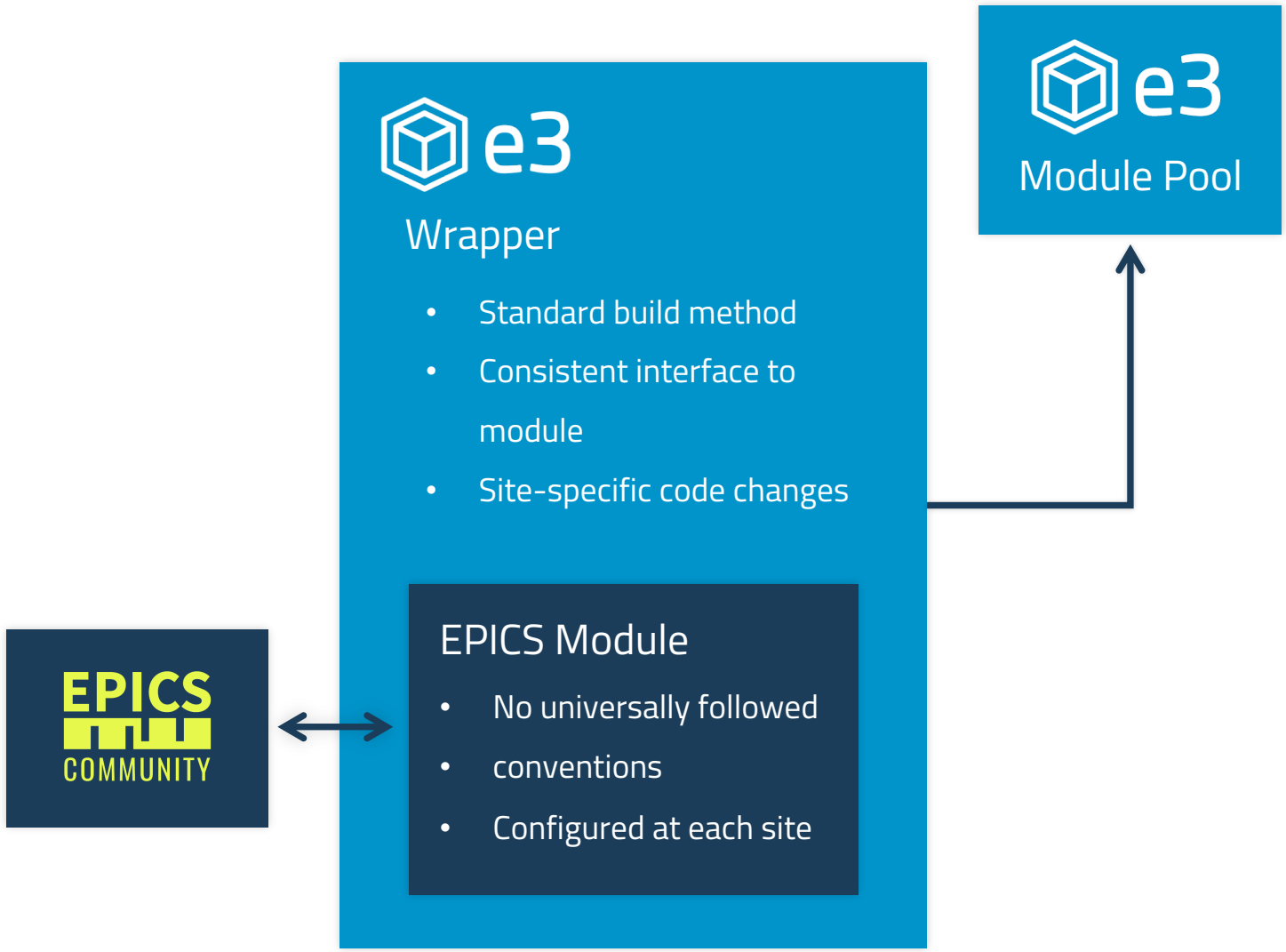
```
$ iocsh.bash st.cmd
```

```
require module1
require module2

epicsEnvSet(PARAM, value1)
iocshLoad($(module1_DIR)/config.iocsh, "PARAM=$(PARAM)")

epicsEnvSet(PARAM, value2)
iocshLoad($(module1_DIR)/config.iocsh, "PARAM=$(PARAM)")

iocshLoad($(module2_DIR)/module_2.iocsh)
```

API:

- Restricted interface to simplify running IOCs
- Exposes parameters to control system configuration management utilities
- Configuration is done by system expert, not necessarily the IOC integrator
- No need to explore inner workings of all modules

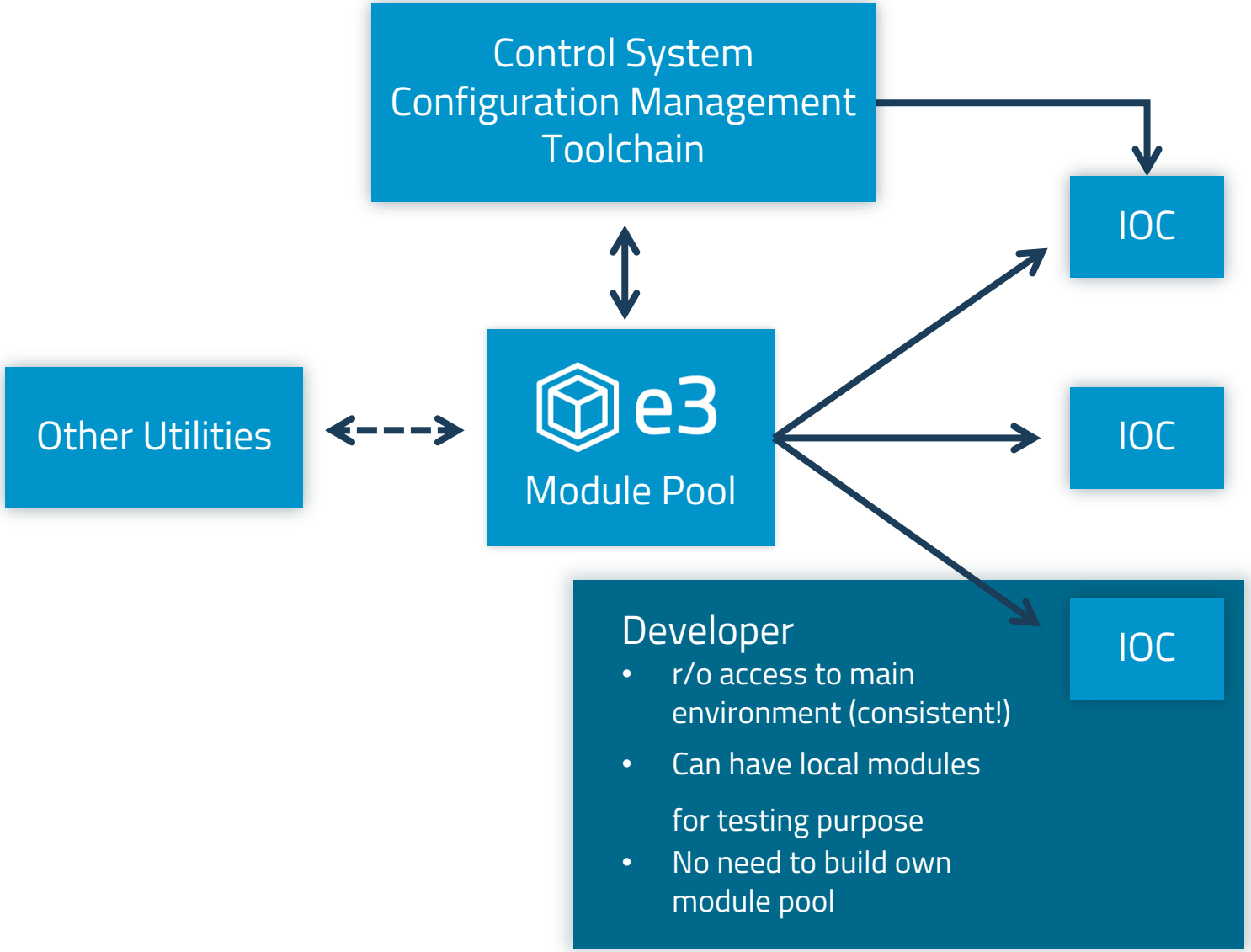
Standardised Interface for Epics Modules



e3

- Provides a standard interface between EPICS modules and IOCs
- Version controlled site-specific changes (db, sources, snippets, anything!) without needing to push to community repositories
- Limited pool of modules (avoids combinatorial explosion)
- Guaranteed consistent environment
- Focus on "high-level" dependencies: only need to specify StreamDevice version, no need to specify Asyn as well

e3 Environment



- Single pool of modules maintained by a dedicated team
- Regular review as a part of deployment in order to ensure high code quality
- Supports multiple version of EPICS base
- Supports multiple consistent environments