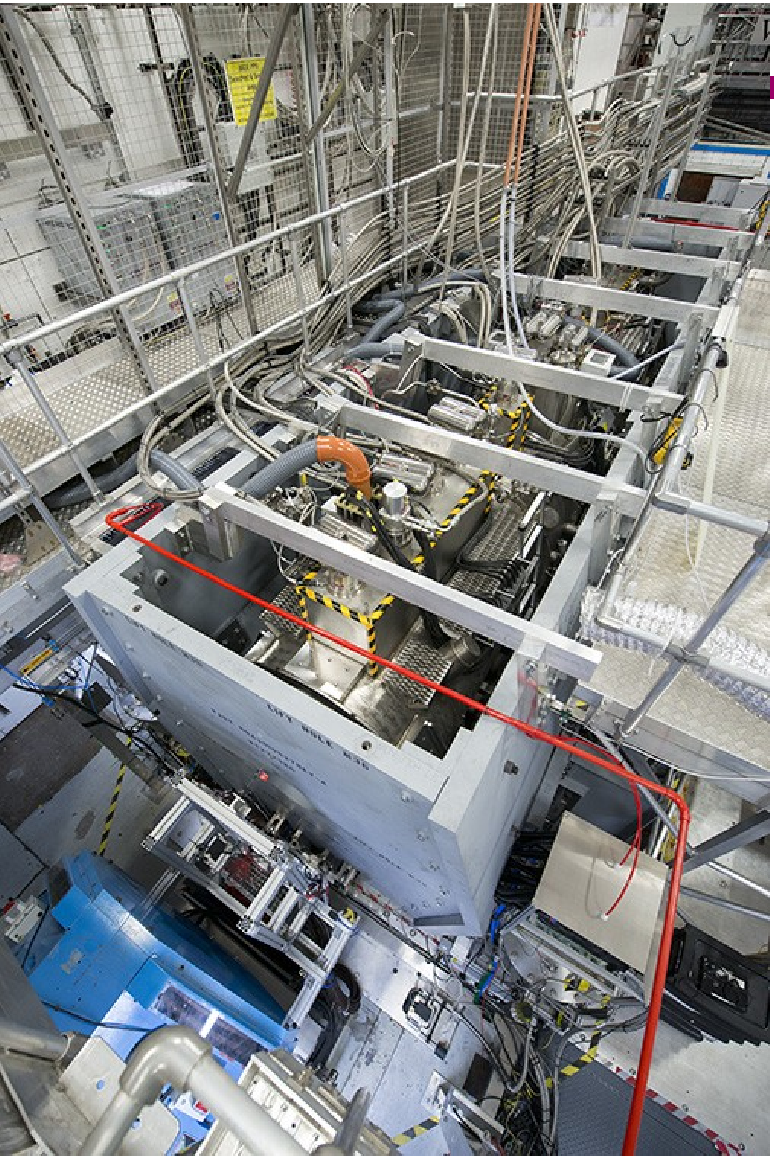
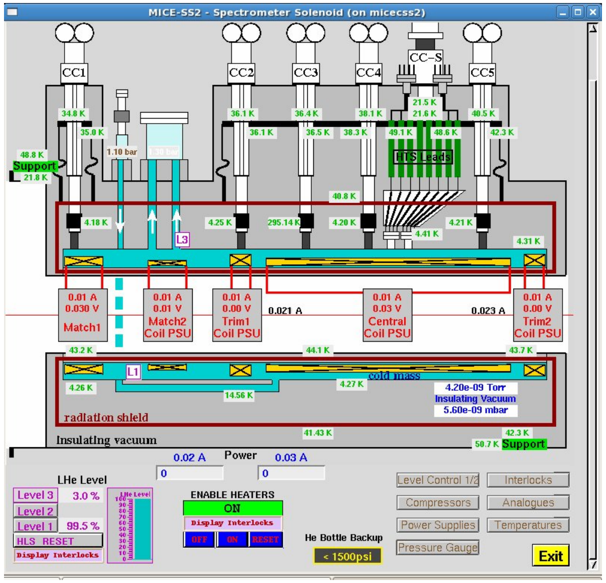


Acnet Device Record



Pierrick Hanlet

20 February 2020

Outline

I Introduction

II Simple View of EPICS

III Standard Deployment of EPICS

IV Acnet Device Record

V Outstanding Issues

VI Conclusions

Outline

I Introduction

II Simple View of EPICS

III Standard Deployment of EPICS

IV Acnet Device Record

V Outstanding Issues

VI Conclusions

Introduction

- So, should we completely switch to EPICS?
  - ACNET is a mature system
  - Change is too dramatic for those embedded in ACNET

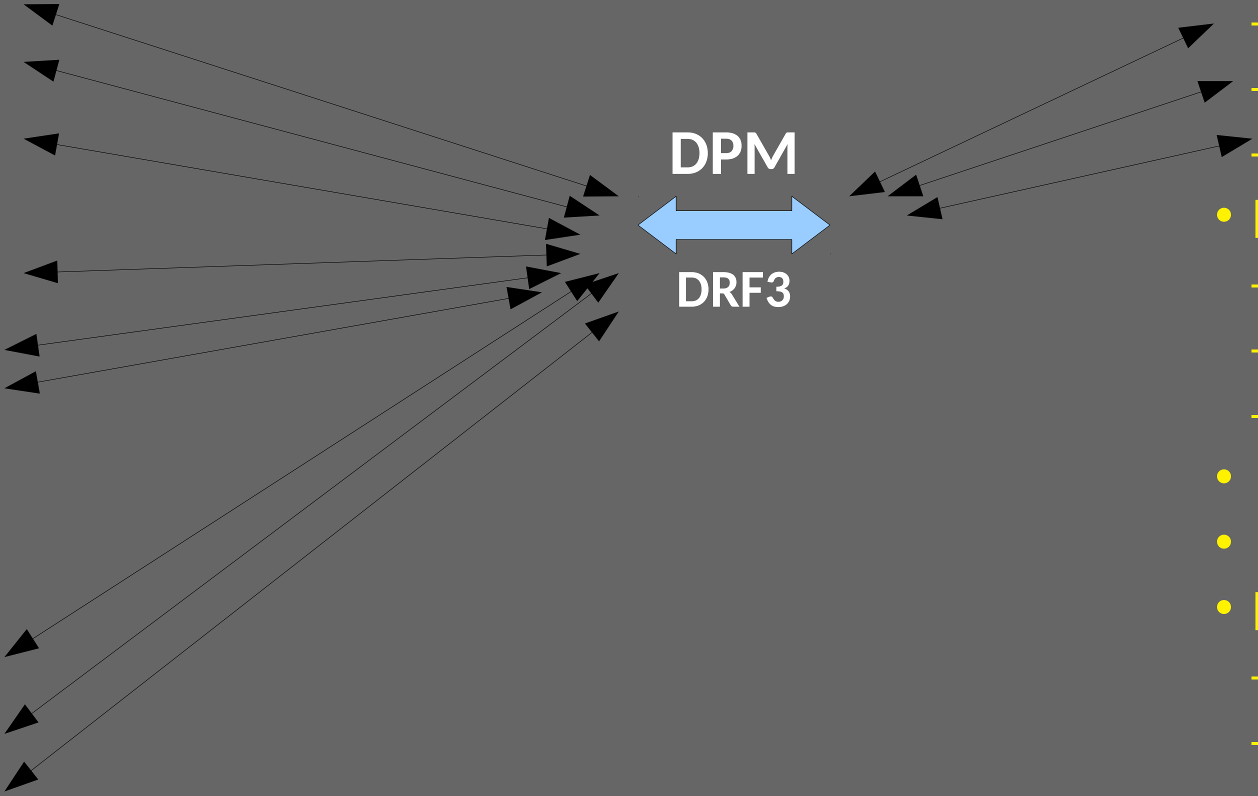
- However, most use of controls is client-side
- Keep familiar ACNET consoles and applications
- Feed EPICS IOCs to ACNET
  - Accomplished with DRF3 in the DPMs
  - Operators still see ACNET consoles
  - Can also have EPICS GUIs (edm and/or Phoebus)
- New front ends can come online as EPICS IOCs
  - When hardware is upgraded to linux-based system, IOC will only need to be recompiled for new platform, but not re-written

Introduction

- The tough part is the mapping:

IOC

- PV
  - field
  - field
  - field
- PV
  - field
  - field
  - field
-
-
- PV
  - field
  - field
  - field


DPM

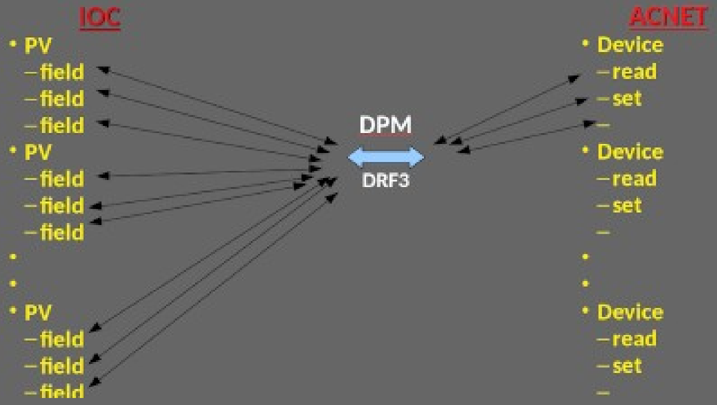
DRF3

ACNET

- Device
  - read
  - set
  - control
- Device
  - read
  - set
  - digital alarms
-
-
- Device
  - read
  - set
  -

Introduction

- The Acnet Device Record (with associated work with DPM) provides:
  - replication of existing acnet structure
  - an EPICS support module – easy modification of existing/new IOCs
  - allow for clients to be agnostic to FE type
  - simple rebuild of IOCs a la EPICS
  - configuration file deployment (after rebuild) does mapping

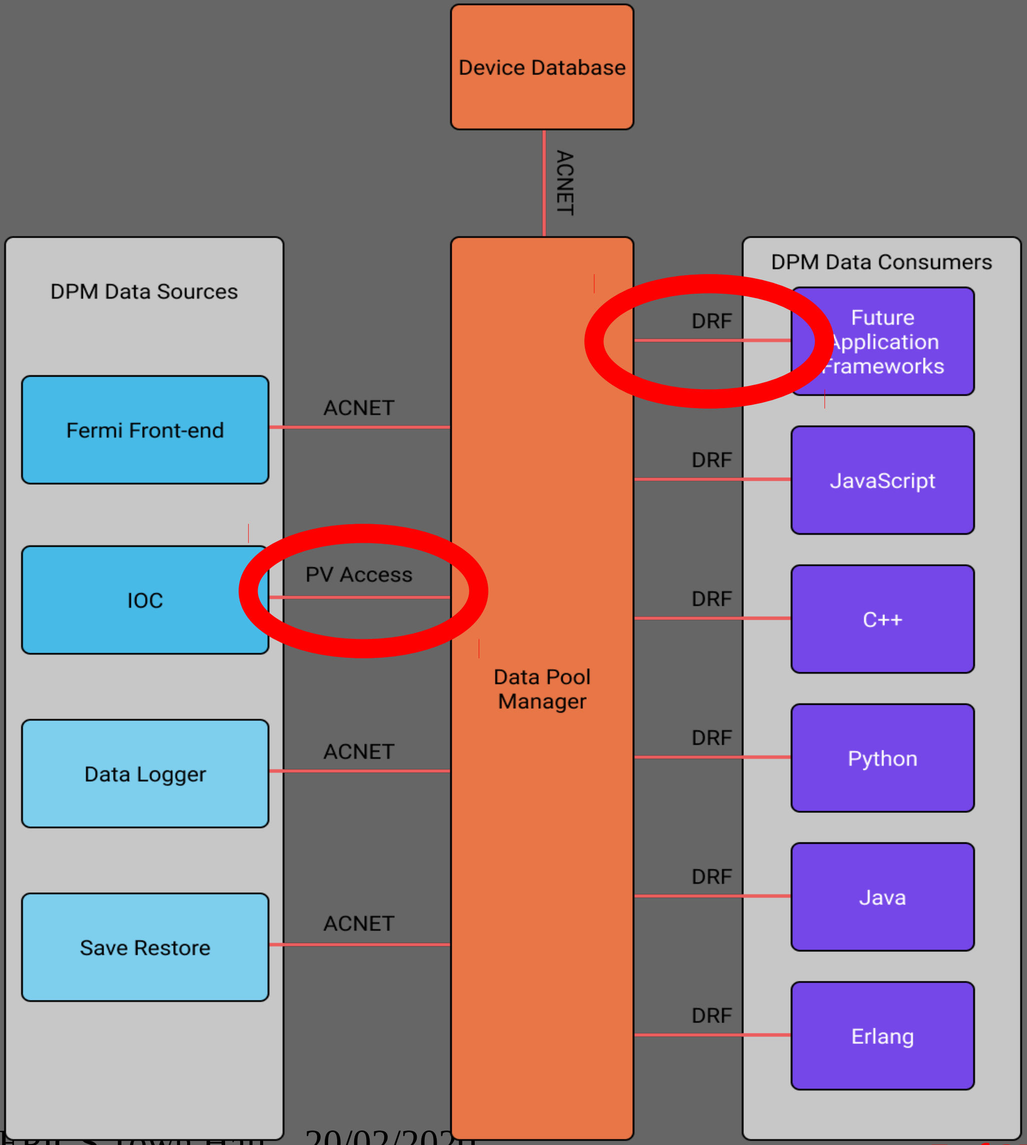


- Use EPICS-7 to:
  - Create Acnet Device Record C++ structure
  - Take advantage of v7 network protocol; i.e. high speed and bandwidth

Introduction

The ACNET Device Record allows IOC developers to mirror the device database structure on an IOC. This record gives developers the flexibility to use familiar IOC tools and conventions as opposed to one-to-one mapping PVs to device properties in the device database.

Thus familiar ACNET applications become agnostic to the front end type.



Outline

I Introduction

II Simple View of EPICS

III Standard Deployment of EPICS

IV Acnet Device Record

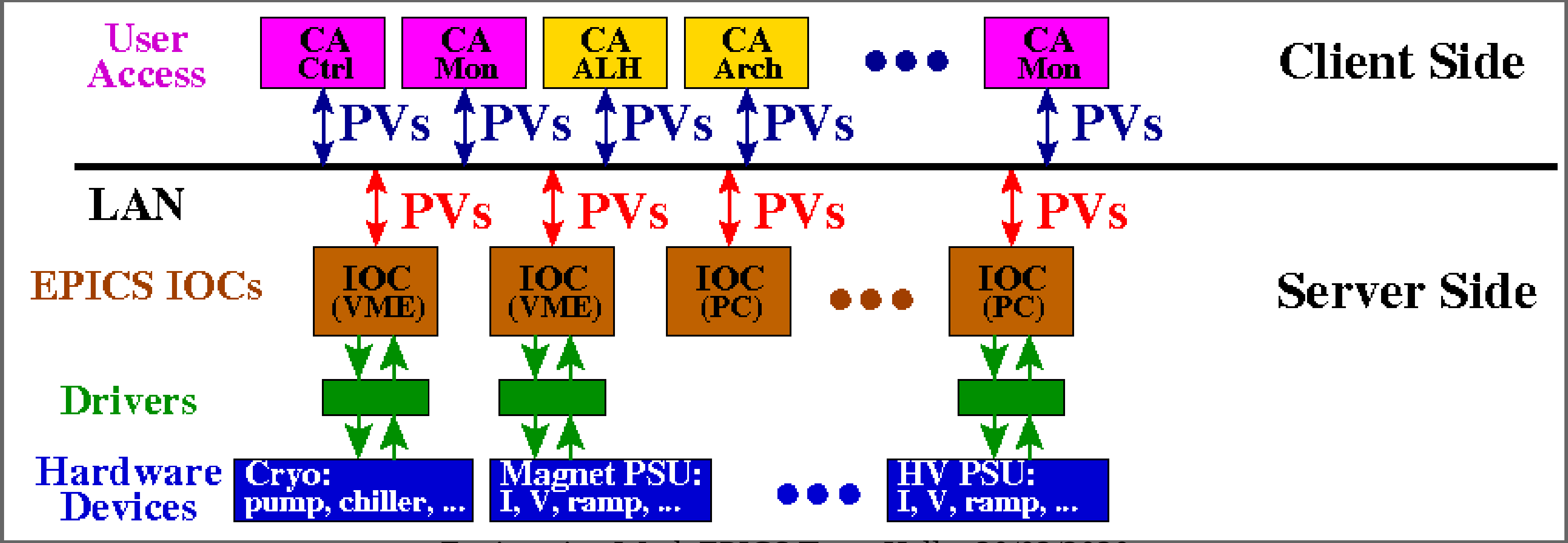
V Outstanding Issues

VI Conclusions

A Simple View of EPICS

Experimental Physics & Industrial Control Systems

- HW+Drivers connect to LAN via IOCs (Input/Output Controllers)
- IOC is controller (VME, PC, ...) and its software
- IOCs create PVs (process variables) to represent HW parameters
- PVs further described with native fields
- PVs available on LAN to other IOCs or clients
- Client Side are different GUIs, alarms, archives, displays, ...



A Simple View of EPICS

- EPICS “base”: the main core of EPICS, comprising the build system and tools, common and OS-specific interface libraries, Channel Access and PV Access client and server libraries, static and run-time database access routines, the database processing code, and standard record, device and driver support.

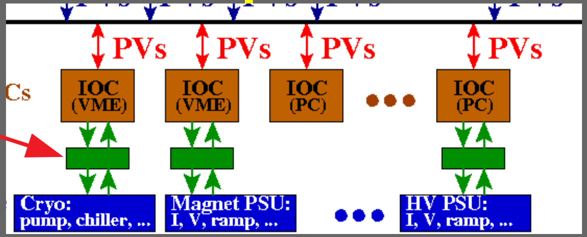
- For Modernization and PIP-II, writing EPICS-based front ends means that we can start now with the software because it will be transferable to any new platform that we will choose and use in the future

A Simple View of EPICS

- Many "record" types: e.g.
  - data types (binary, float, multi-bit, waveform, structures)
  - calculations
  - functional (heartbeat, computer status, busy, ...)
  - sequences
  - aSub: C/C++ function inputs PVs, processes, and outputs PVs
- Record types have native "fields" to further describe record: e.g.
  - alarm
  - units
  - operating limits
  - scanning rates: time based, I/O interrupt, passive
  - fanouts and forward links
  - description
  - dozens of fields; these are defined in xxx.dbd files
- Records are used to create PVs
- PVs are used to create IOCs

A Simple View of EPICS

- Support modules are analogous to modules one might add to a kernel; i.e. they aren’t required for a stable OS, but are required for a server’s specific function.
- Many support modules for HW already written
- Some support modules are effectively drivers for specific hardware
  - user written
  - found on EPICS website
  - ask EPICS user community



- One of the larger packages which contains many support modules is synApps

- ***The Acnet Device Record is a new record type***

Outline

I Introduction

II Simple View of EPICS

III Standard Deployment of EPICS

IV Acnet Device Record

V Outstanding Issues

VI Conclusions

Standard Deployment of EPICS

- **Production:**

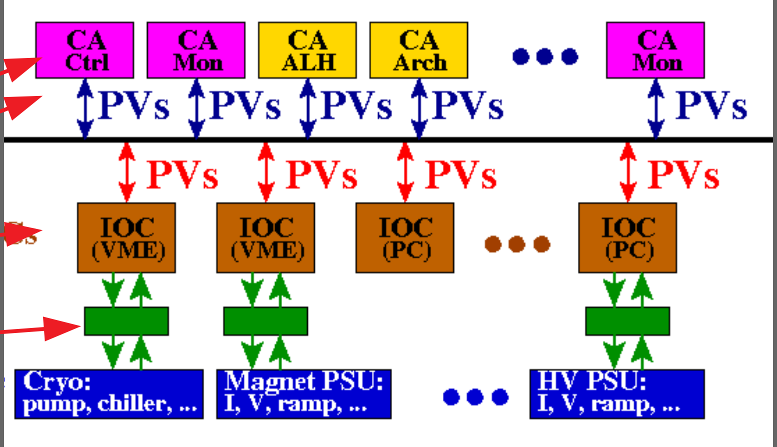
/usr/local/epics/Extensions

/usr/local/epics/Config

/usr/local/epics/Support

/home/epics/epicsPRO/iocTops

/usr/local/epics/base



- **Development**

/home/epics/epicsDEV/base	(build tools and core)
/Support	(modules: asyn, seq, ipac...)
/iocTops	(IOCs – built from base+Support)
/Extensions	(user applications)
/Config	(configuration files)
/Software	(external libraries)

https://cdcvs.fnal.gov/redmine/projects/ad-controls-dept/wiki/Building_an_EPICS_IOC_on_a_clx_Machine

Outline

I Introduction

II Simple View of EPICS

III Standard Deployment of EPICS

IV Acnet Device Record

V Outstanding Issues

VI Conclusions

Acnet Device Record

The Acnet Device Record: a new, custom EPICS module, implying:

- can build into an existing IOC with boilerplate modifications to existing IOC
- can create new IOC which reads PVs from existing IOC

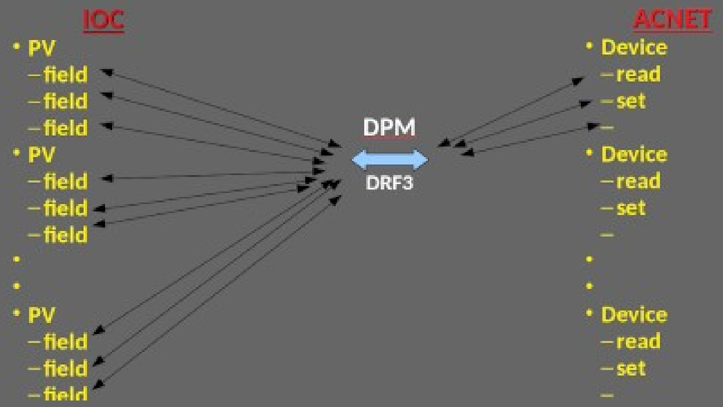
Use aSub record:

- Allows flat v3 style db file
- EPICS v7 does not have db files
- This is configurable from macro substitutions at build or run time

Acnet Device Record

One fundamental difference between ACNET and EPICS is:

- ACNET uses a paradigm in which the client drives all aspects of data collection
- EPICS uses a paradigm in which the IOC
- An ACNET device has several standard properties:
  - reading
  - setting
  - control
  - basic status
  - extended status
  - digital alarms
  - analog alarms



Acnet Device Record

With Acnet Device Record (ADR) as a "support module", building it Into the IOC requires 2 changes:

1)Create ACNET device – as usual

2)IOC build changes (boiler plate):

- ./configure/RELEASE – add line for acnetPV
- ./myIOCApp/src/Makefile – add lines for acnetPV library and dbd file

3)Next copy template substitution file and add desired PVs

4)Run make in your iocTops/myIOC directory. IOC now includes ADR

5)Add lines in IOC startup script: one for each acnet device

6)Run IOC

- New structured PVs now exist on the network and can be picked up by DPM

https://cdcvs.fnal.gov/redmine/projects/ad-controls-dept/wiki/EPICS_AcnetDeviceRecord

Acnet Device Record

Mapping documentation:

```
##************************************************************************##
##*                                                                      *##
##* Use this template to allow substitution files to fill PVs:           *##
##*                                                                      *##
##* di:   device index          (filled from config DB)                  *##
##* ssdn: SSDN of ACNET device  (filled from config DB)                  *##
##* node: ACNET trunk/node      (filled from config DB)                  *##
##* name: name of ACNET device  (filled from config DB)                  *##
##* bdev/bsta/bpp:              ACNET basic status                       *##
##* edev/esta/epp:              ACNET extended status                    *##
##* ddev/dsta/dpp:              digital alarm block                      *##
##* adev/asta/app:              analog alarm block                       *##
##* rdev/read/rpp/rtype:        reading value                            *##
##* wdev/wrd/wpp/wnelm/wtype:   reading waveform value                   *##
##* cdev/ctrl/cpp:              control bits                             *##
##* dev/set/pp:                 setting value                            *##
##*                                                                      *##
##* At present both PV description and engineering units are read from   *##
##* from the v3 record and written to the ADR                            *##
##*                                                                      *##
##* Where  Xdev/xxxx/Xpp (above) is:                                     *##
##*        xdev-> IOC family                                             *##
##*        xxxx-> PV name                                                *##
##*        Xpp -> CP (different IOC) or PP (same IOC)                    *##
##*                                                                      *##
##* When there are no input PVs, xxxx should be "Xdummy"                 *##
##*                                                                      *##
##* These macros are defined in the IOC substitution file:               *##
##* AcnetDevices.substitutions found in xxx/xxxApp/Db                    *##
##*                                                                      *##
##************************************************************************##
```

Acnet Device Record

Example substitution file:

```
global  {dev=PIP2-RF-FG-01,ndev=PIP2-RF-FG-03}
file acnetPV.template
{
    {name="E:HSTFG3",
     bdev=$(ndev),
     bsta="bdummy",
     bpp="PP",
     edev=$(ndev),
     esta="edummy",
     epp="PP",
     ddev=$(ndev),
     dsta="ddummy",
     dpp="PP",
     adev=$(ndev),
     asta="adummy",
     app="PP",
     rdev=$(dev),
     read="R3:V",
     rpp="CP",
     rtype="DOUBLE",
     wdev=$(ndev),
     wrd="wdummy",
     wpp="PP",
     wnelm="1",
     wtype="DOUBLE",
     cdev=$(ndev),
     ctrl="cdummy",
     cpp="PP",
     sdev=$(ndev),
     set="sdummy",
     spp="PP"}
}
```

Acnet Device Record

PV naming convention:

```
##************************************************************************##
##*                                                                      *##
##* PV names should have the form: ACC/EXP-SYS-SUB-NUM:VAR               *##
##* where:                                                               *##
##*       ACC/EXP is ACC=accelerator (e.g. LIN,BOO,MI )                  *##
##*               or EXP=experiment  (e.g. NUMI)                         *##
##*       SYS=system (e.g. VAC for vacuum)                               *##
##*       SUB=sub-system (e.g. GAUGE,PUMP,VALVE)                         *##
##*       NUM=element number (e.g. 01 for pump 1)                        *##
##*       VAR=variable being read or written (e.g. PRES for pressure)    *##
##* These names will eventually be standardized                          *##
##*                                                                      *##
##************************************************************************##
```

The ADR should have the same name as the ACNET device

Two such IOCs are presently deployed

Outline

I Introduction

II Simple View of EPICS

III Standard Deployment of EPICS

IV Acnet Device Record

V Outstanding Issues

VI Conclusions

Outstanding Issues

- Time stamps
  - comes for free with EPICS
- Settings
  - presently missing from DPM (work in progress)
  - authentication will use existing system
- Event driven data
  - biggest challenge: presently thinking to send hardware bit-mask and process PV using I/O Interrupt; read all pertinent events and load them into waveform. Specific events can be distributed either at IOC level or DPM
- Security – access control
  - standard ascii file can be loaded at IOC start
- Use of EPICS tools on client side
  - Phoebus is not tied to EPICS, but would require a new API
- IOCs on different platforms
  - plan to standardize build using Buildroot

EPICS-ACNET Integration – Early Efforts

- EPICS-ACNET bridge
  - Enables seeing EPICS PVs reflected into an ACNET Device
  - A Channel Access client built into an ACNET front-end
  - Implemented in Erlang
  - Implements low level Channel Access protocol (no libraries)
  - No Support for PV Access
  - Implemented by always adding an ACNET Device for any PV desired
    - No direct ACNET client access to a PV by PV name alone
    - Database's Foreign Device Mapping field indicates PV name
    - Device – PV mapping discovered from the database at Front-end startup (can be re-initialized with additions)
- Moving this functionality to the DPM gives more universal access to PVs from ACNET clients
  - But current DPM only supports PV Access, no CA from pure v.3 IOCs

.

ACNET-EPICS Integration

- Allow EPICS clients to transparently access ACNET devices.
- Implement a CA server which responds to EPICS CA Search requests for ACNET Devices
- Checks for target of search request matching the name of an ACNET Database Device
- When it matches:
  - opens a CA circuit with the requestor
  - Starts a DPM client reading of the corresponding Acnet device
  - Feeds scaled Acnet responses to the EPICS CA requestor
- I have an initial version of this implemented in Erlang
  - EPICS can potentially request many datatypes in the response (float, int16, int32,…) but for now I only support two (float, and float with extra info like units, timestamp)
  - It's a fair amount of work to implement all possible datatypes and I'm not sure if it is worth it
- No support yet for PV Access
- Not integrated into any infrastructure yet.

Outline

I Introduction

II Simple View of EPICS

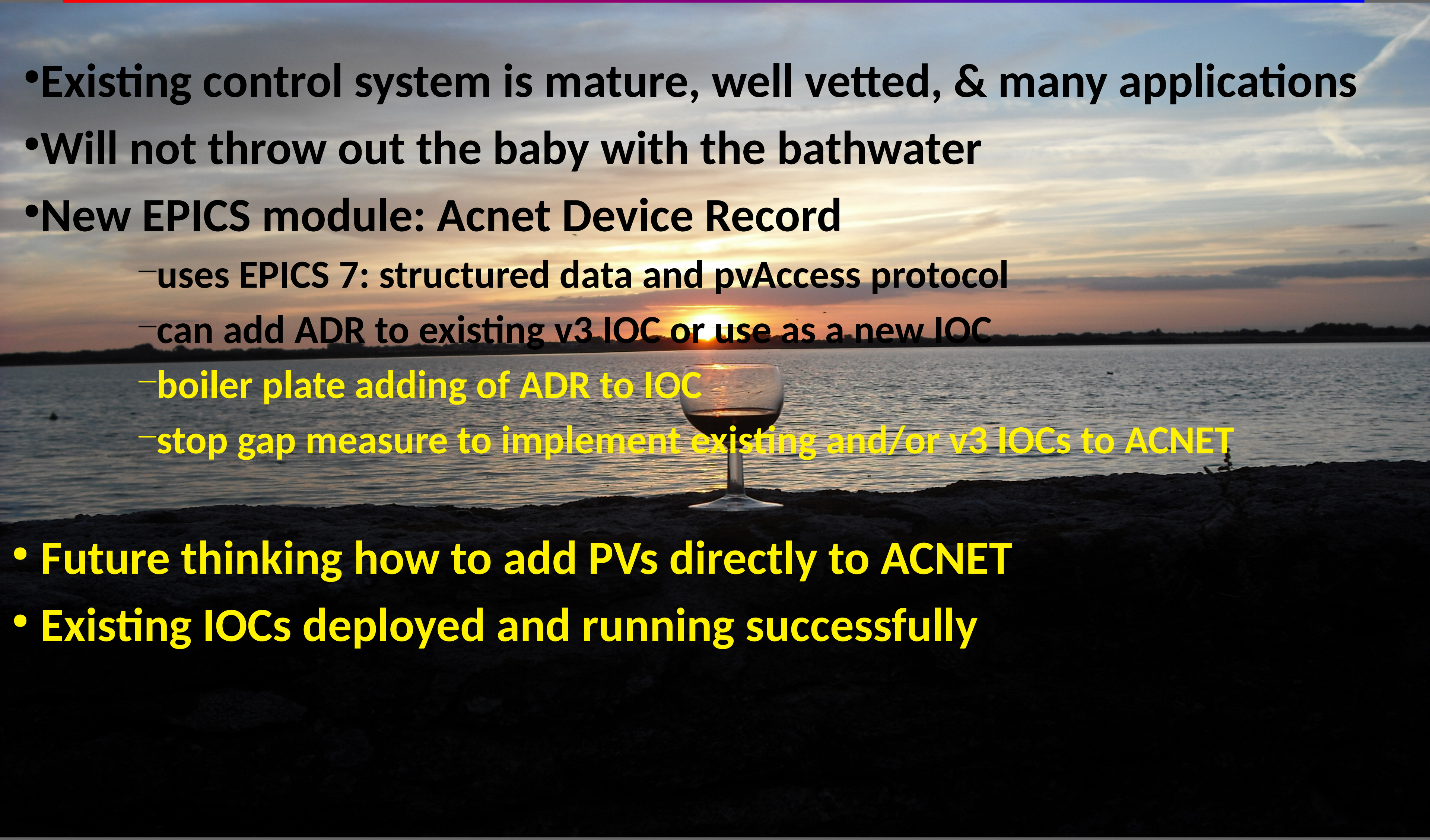
III Standard Deployment of EPICS

IV Acnet Device Record

V Outstanding Issues

VI Conclusions

Conclusions



- Existing control system is mature, well vetted, & many applications
- Will not throw out the baby with the bathwater
- New EPICS module: Acnet Device Record
  - uses EPICS 7: structured data and pvAccess protocol
  - can add ADR to existing v3 IOC or use as a new IOC
  - boiler plate adding of ADR to IOC
  - stop gap measure to implement existing and/or v3 IOCs to ACNET

- Future thinking how to add PVs directly to ACNET
- Existing IOCs deployed and running successfully