Configuration and Deployment of production-level services for CODAC operation at ITER

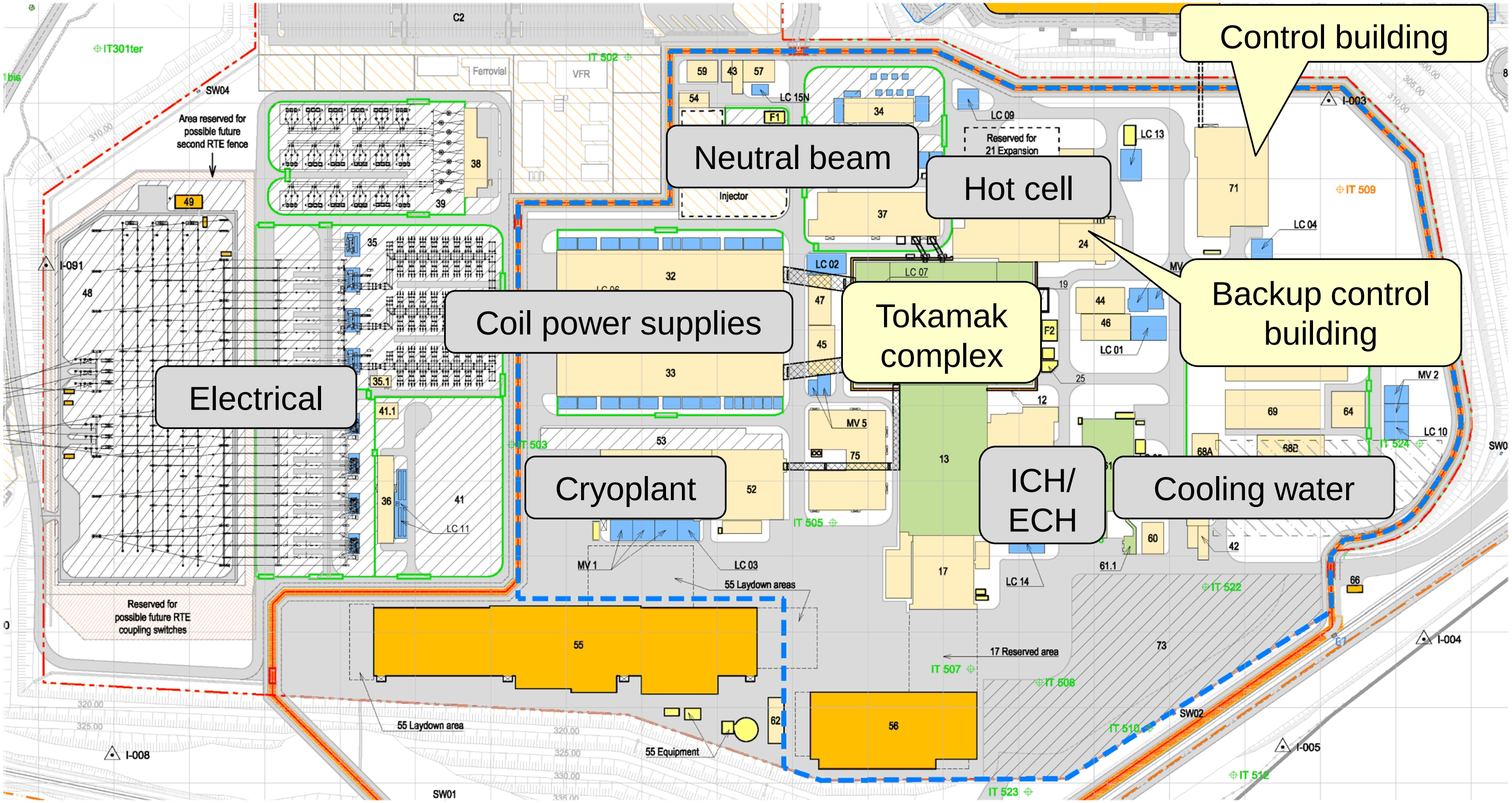
A.Bustos, B.Bauvir, R.Lange

ITER Organization

Disclaimer: The views and opinions expressed herein do not necessarily reflect those of the ITER Organization

Motivation

Infrastructure: Systems Requiring Control



Control building

Neutral beam

Hot cell

Coil power supplies

Tokamak complex

Backup control building

Electrical

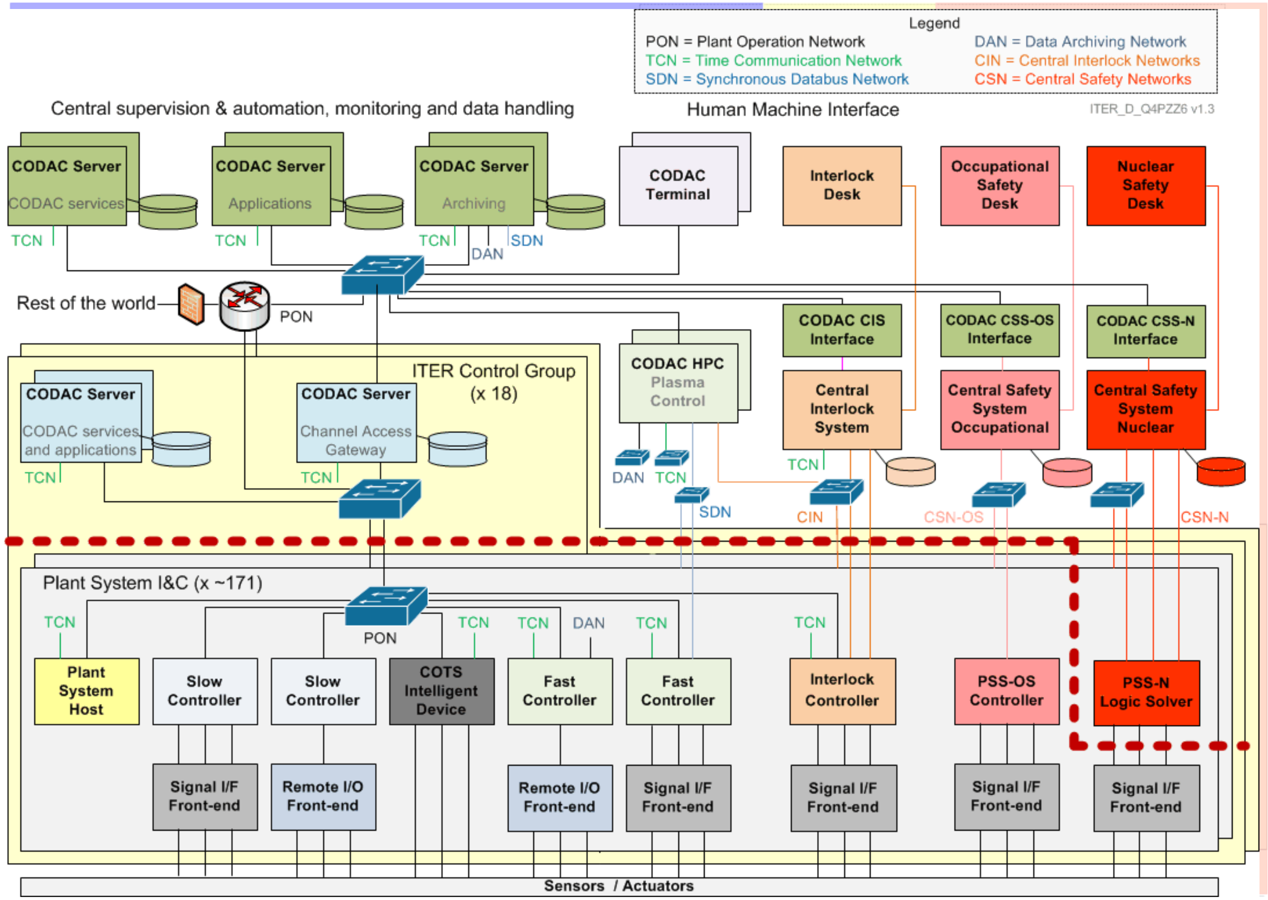
Cryoplant

ICH/ ECH

Cooling water

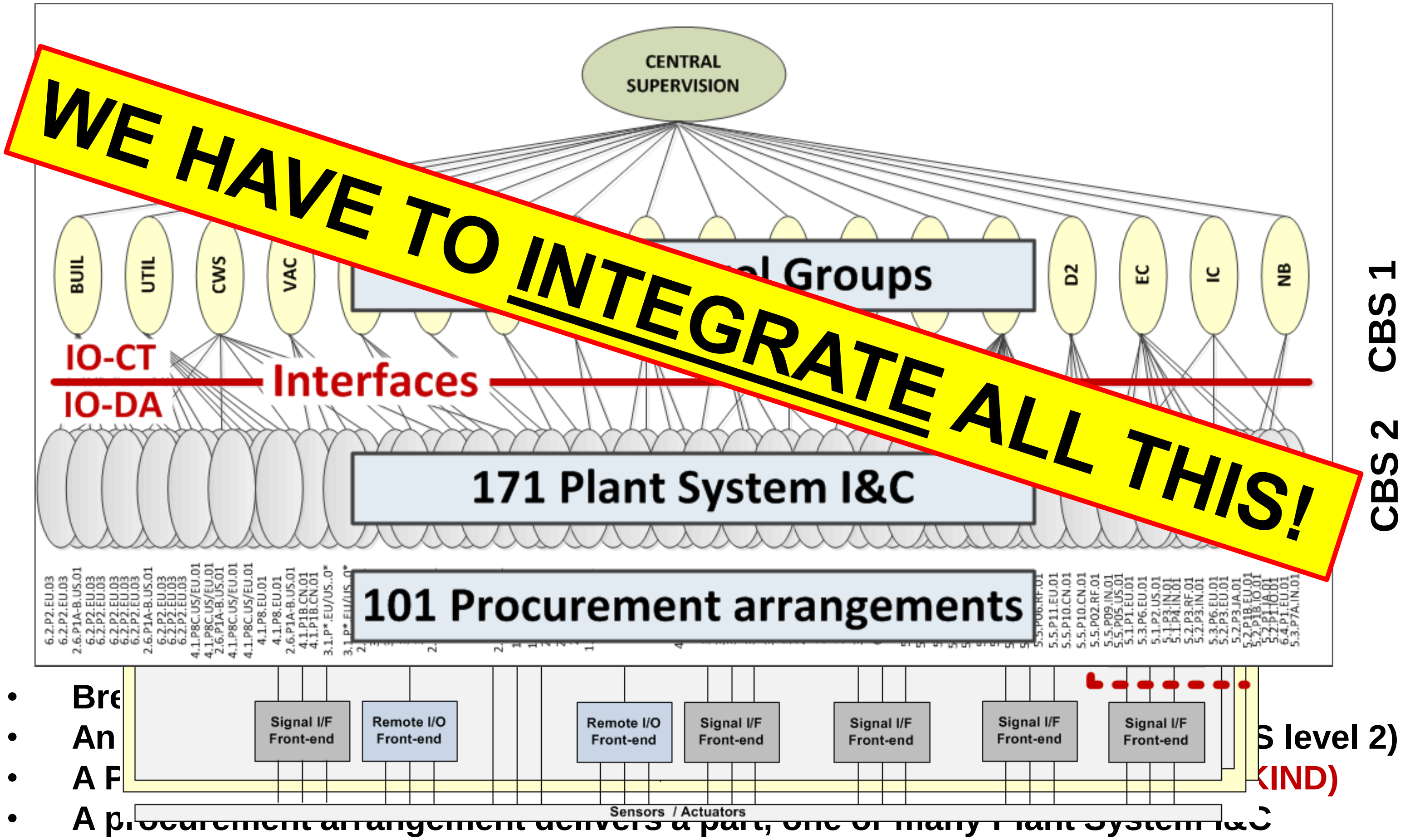
Motivation

Architecture Recap: ITER Control System Architecture



Motivation

Architecture Recap



- Bre
- An ... S level 2)
- A F ... KIND)
- A procurement arrangement delivers a part, one of many Plant System I&C

CBS = Control Breakdown Structure

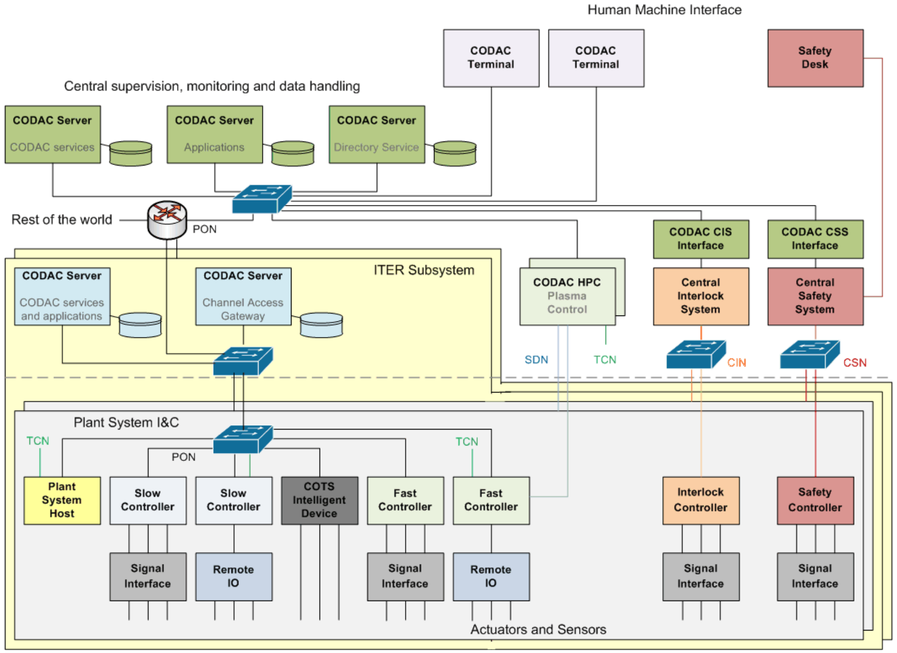
Motivation

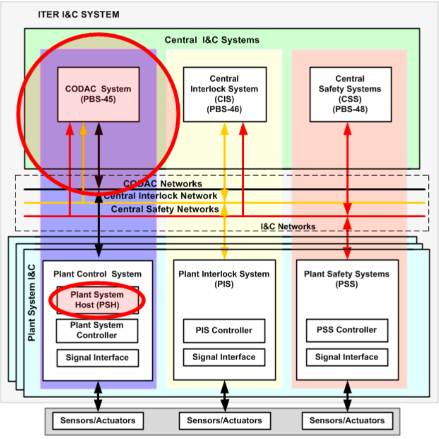
- ❑ Most of ITER Applications are built outside ITER (ITER shall provide standardization)
- ❑ 171 Plant System I&C to be integrated

- **Follow STRICT HW and SW standards**
- **Follow STRICT Software Configuration Control**
    - minimal human intervention in deployments
    - capability for disaster recovery
    - capability to inspect configuration version in production
    - capability for reinstallation of a known/verified configuration

- **Follow STRICT deployment procedures and reduced numbers of actors involved**
- **Support incremental changes during integration and test**

Scope





CODAC provides centralized services to Plant Systems during installation, testing, integration and operation.

- CODAC Services
  - BOY Operator Interface
  - BEAST Alarm System
  - BEAUTY Archiving System
  - OLOG Electronic Logbook
- CODAC Machines
  - Infra. Servers and Switches
  - Operator Terminals
  - Central Servers and DB
  - Fast Controllers
  - Plant System Hosts

CCS Technologies

Integration strategy – CODAC Core System

- The selected operating system is **Red Hat Enterprise Linux v7.4** for the x86-64 architecture (RHEL x86_64) and, optionally, real-time extension MRG-R
- The infrastructure layer is **EPICS v7.02** (Experimental Physics and Industrial Control System) @ CCSv6.1
- The CODAC services layer is Eclipse based **Control System Studio v4.6.2** including HMI, alarming, archiving etc.
- ITER specific software such as configuration (self-description data), state handling, drivers, networking, etc.
- Distribution over Internet to registered user organization

I&C Project Structure

Software project structure adheres to Maven recommended structure.
`src/main/epics` is allocated for EPICS projects

```
{unit_name}
├── pom.xml (project information)
├── doc/ (Directory for documentation)
└── src/ (Directory for source files)
  ├── main/ (Directory for source files of product)
  │ ├── beast/ (CSS alarm configuration files)
  │ ├── beauty/ (CSS archive configuration files)
  │ ├── boy/ (CSS boy opi files)
  │ ├── c++/ (C++ application sources)
  │ ├── databrowser/ (databrowser files)
  │ ├── epics/ (EPICS application sources)
  │ ├── java/ (Java application/library sources)
  │ ├── plc/ (PLC program sources)
  │ ├── python/ (Python application/library sources)
  │ ├── resource/ (application/library resources)
  │ └── scan/ (CSS scan command files)
  ├── test/ (Directory for source files of test)
  │   ├── ...
  └── target/ (Build results)
      ├── ...
```

CCS Approach to Packaging

□ Red Hat Linux as a base platform

□ RPM as a packaging tool, with YUM as a distribution mechanism

□ Build system is controlled by Apache Maven (clean, compile, …)

□ Maven is also used to automate frequent I&C tasks (start / test / stop sequence).

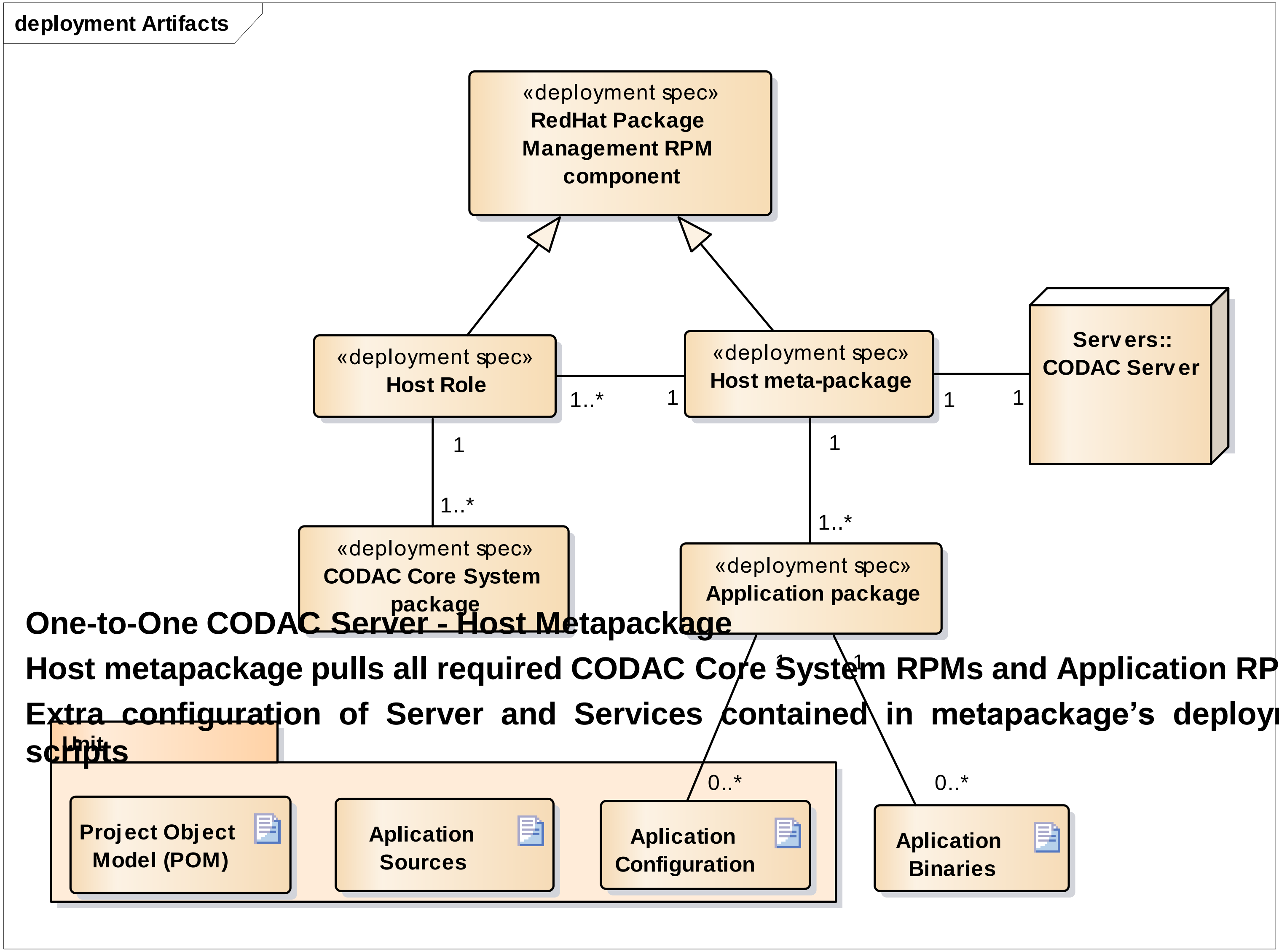
Maven™

yum
yellowdog updater modified

redhat®

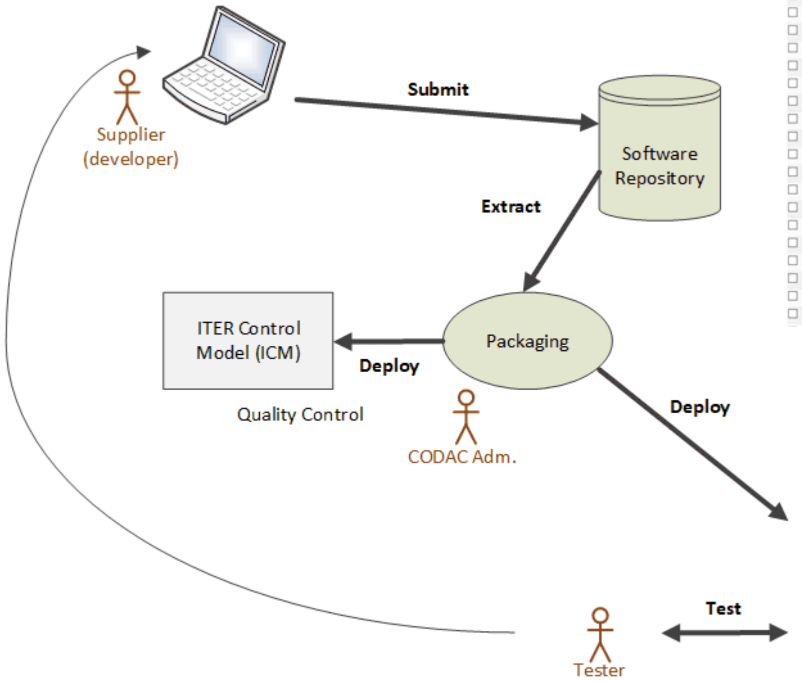
rpm

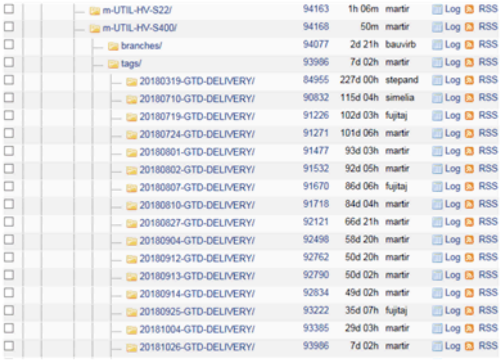
Software Units and RPM packages



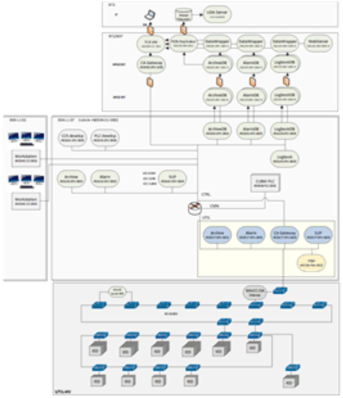
- One-to-One CODAC Server - Host Metapackage
- Host metapackage pulls all required CODAC Core System RPMs and Application RPMs
- Extra configuration of Server and Services contained in metapackage’s deployment scripts

SW Configuration Control Workflow

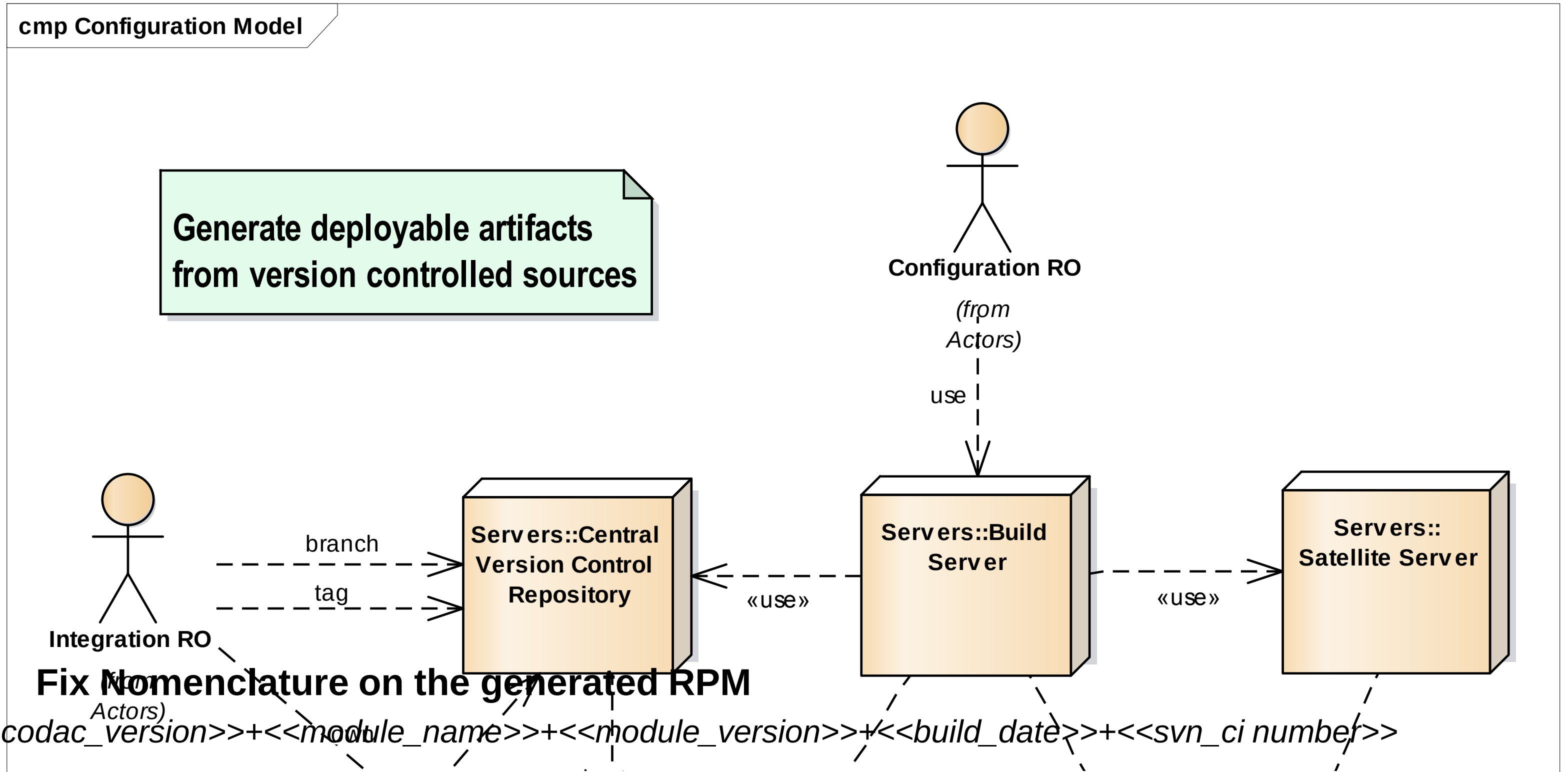




□	m-UTIL-HV-S22/	94163	1h 06m	martir	Log RSS
□	m-UTIL-HV-S400/	94168	50m	martir	Log RSS
□	branches/	94077	2d 21h	bauvirb	Log RSS
□	tags/	93986	7d 02h	martir	Log RSS
□	20180319-GTD-DELIVERY/	84955	227d 00h	stepand	Log RSS
□	20180710-GTD-DELIVERY/	90832	115d 04h	simelia	Log RSS
□	20180719-GTD-DELIVERY/	91226	102d 03h	fujitaj	Log RSS
□	20180724-GTD-DELIVERY/	91271	101d 06h	martir	Log RSS
□	20180801-GTD-DELIVERY/	91477	93d 03h	martir	Log RSS
□	20180802-GTD-DELIVERY/	91532	92d 05h	martir	Log RSS
□	20180807-GTD-DELIVERY/	91670	86d 06h	fujitaj	Log RSS
□	20180810-GTD-DELIVERY/	91718	84d 04h	martir	Log RSS
□	20180827-GTD-DELIVERY/	92121	66d 21h	martir	Log RSS
□	20180904-GTD-DELIVERY/	92498	58d 20h	martir	Log RSS
□	20180912-GTD-DELIVERY/	92762	50d 20h	martir	Log RSS
□	20180913-GTD-DELIVERY/	92790	50d 02h	martir	Log RSS
□	20180914-GTD-DELIVERY/	92834	49d 02h	martir	Log RSS
□	20180925-GTD-DELIVERY/	93222	35d 07h	fujitaj	Log RSS
□	20181004-GTD-DELIVERY/	93385	29d 03h	martir	Log RSS
□	20181026-GTD-DELIVERY/	93986	7d 02h	martir	Log RSS

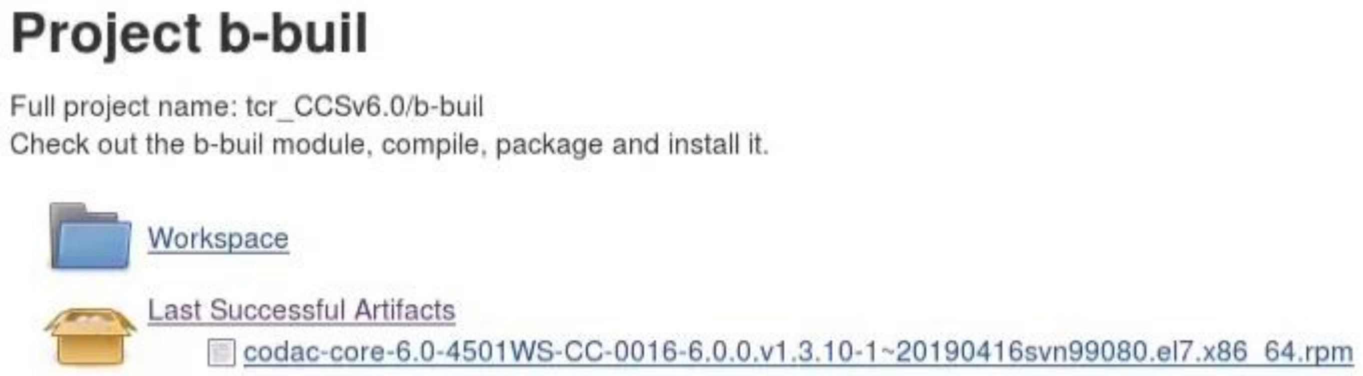


Packaging I&C Integration Modules


cmp Configuration Model
Generate deployable artifacts from version controlled sources
Configuration RO
(from Actors)
use
Integration RO
(from Actors)
branch
tag
Servers::Central Version Control Repository
«use»
Servers::Build Server
«use»
Servers:: Satellite Server

- **Fix Nomenclature on the generated RPM**

<<codac_version>>+<<module_name>>+<<module_version>>+<<build_date>>+<<svn_ci number>>



Project b-buil

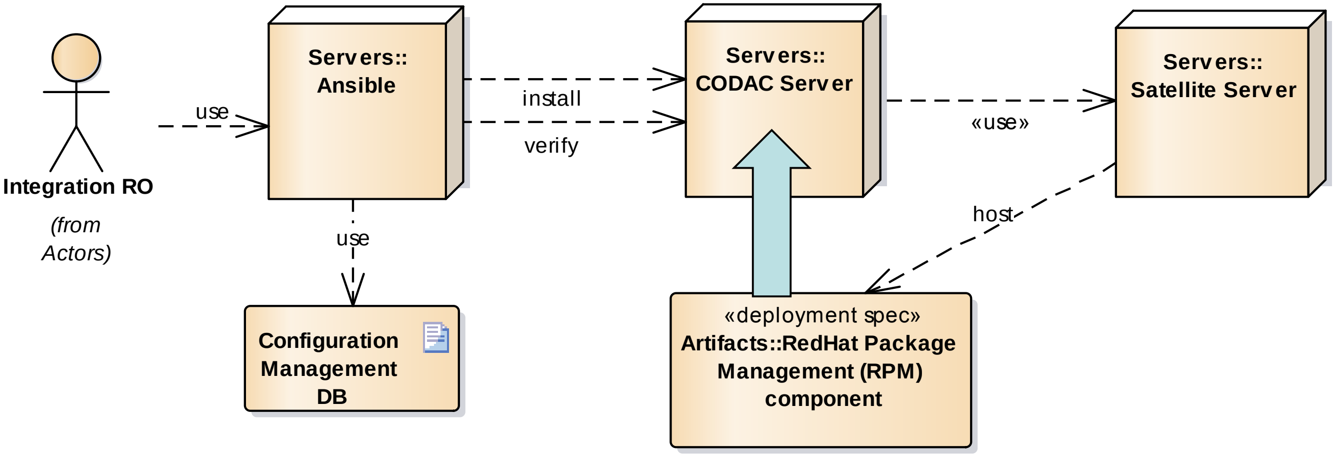
Full project name: tcr_CCSv6.0/b-buil
Check out the b-buil module, compile, package and install it.

Workspace

Last Successful Artifacts

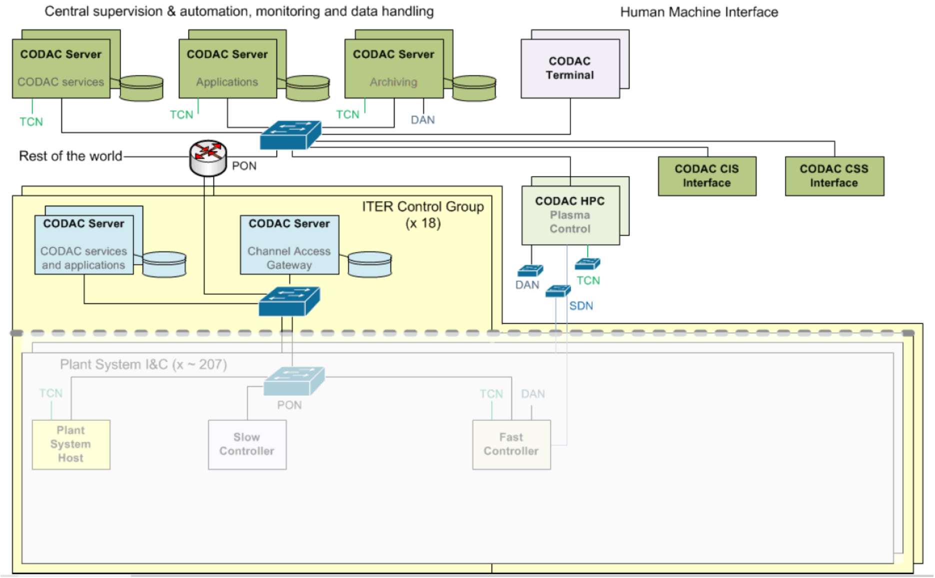
codac-core-6.0-4501WS-CC-0016-6.0.0.v1.3.10-1~20190416svn99080.el7.x86_64.rpm

Deployment Artifacts



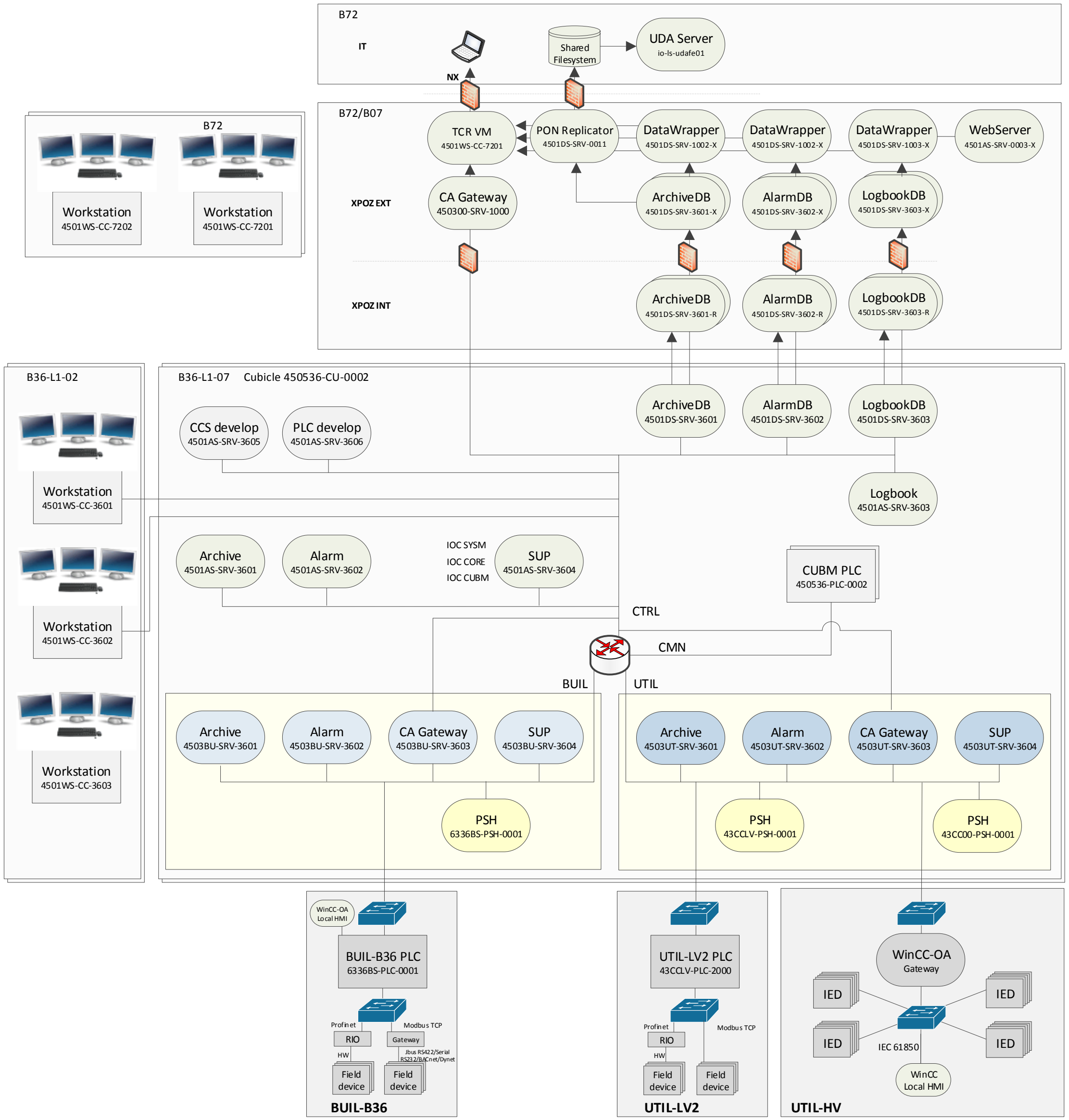
Deployment Modules: Pre-Production

ITER Control System Model

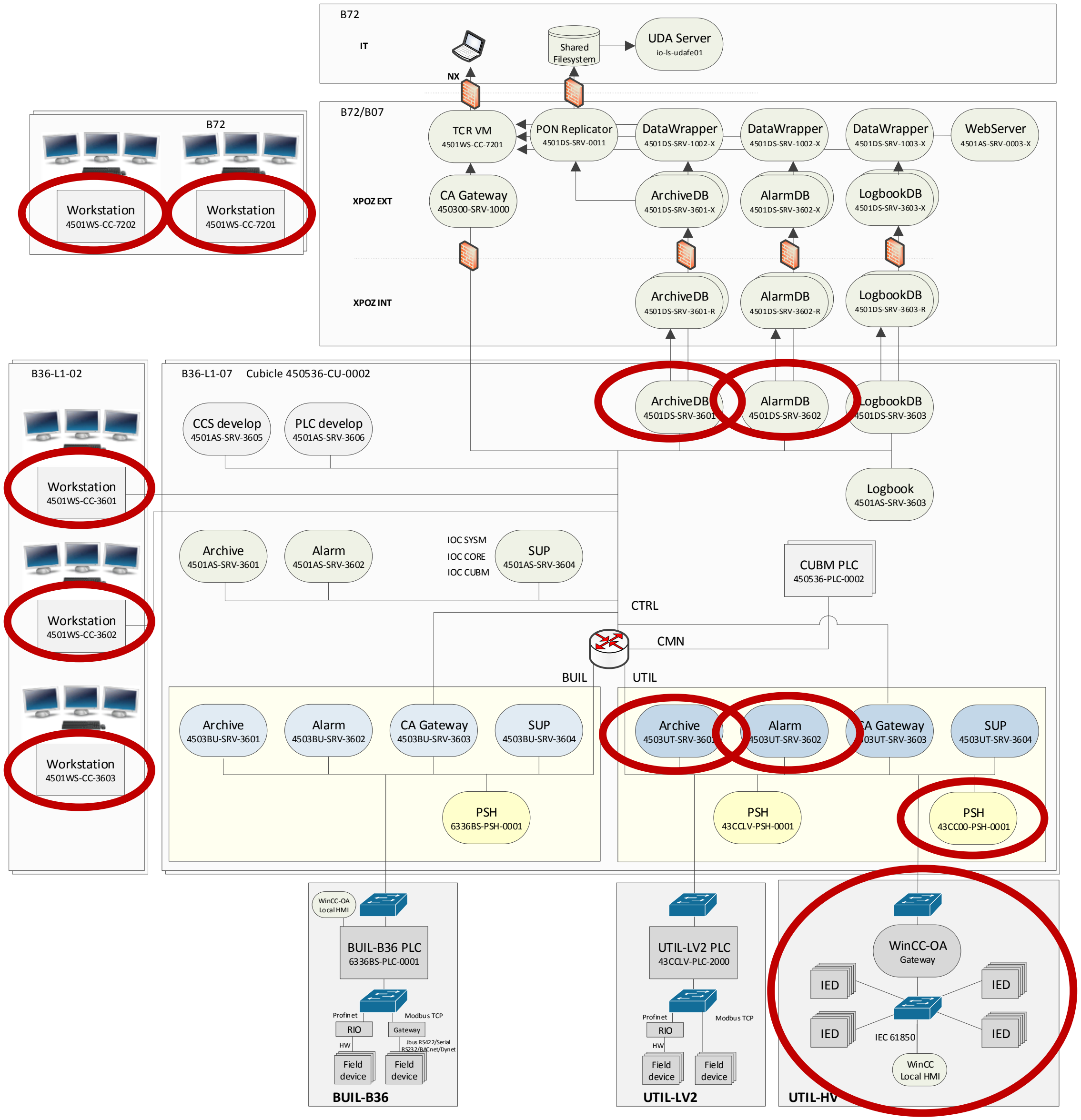


- ❑ Validate CODAC Scalability
- ❑ Provide isolate test environment
- ❑ Provide Simulators for testing CODAC Tech. & PS I&C Modules
- ❑ Provide environment to Operator Training

Deployment: Production TCRs



Deployment: Production TCRs



i.e. New Version of PS I&C module for UTIL-HV-S400!

Conclusion

- ❑ CCS is based on fix releases (patch or new release needed for updates) vs PS I&C SW Support incremental changes during integration and test.

- ❑ Deployment in production only configuration control versioned resources

- ❑ Support incremental changes (~30 versions of m-UTIL-HV-S400/ integrated in 7 months)

- ❑ 11 TCRs in 6 Buildings to come

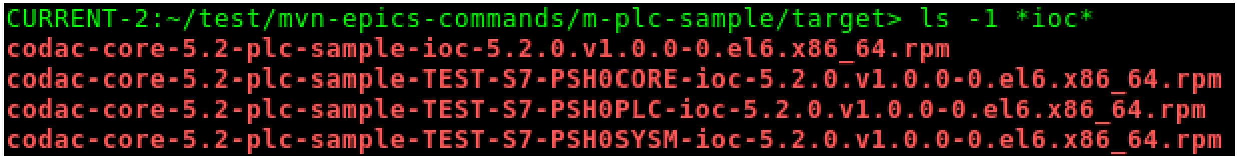
EXTRA Support Slides

courtesy Denis Stepanov

Packaging Instructions for IOCs

pom.xml snippet:

```
<build>
 <plugins>
  <plugin>
   <groupId>org.apache.maven.plugins</groupId>
   <artifactId>maven-iter-plugin</artifactId>
   <extensions>true</extensions>
   <configuration>
    <packaging>
     <package name="TEST-S7-PSH0CORE-ioc">
      <include type="ioc" name="TEST-S7-PSH0CORE" runlevels="345" />
     </package>
     <package name="TEST-S7-PSH0PLC-ioc">
      <include type="ioc" name="TEST-S7-PSH0PLC" runlevels="345" />
     </package>
     <package name="TEST-S7-PSH0SYSM-ioc">
      <include type="ioc" name="TEST-S7-PSH0SYSM" runlevels="345" />
     </package>
     <package name="ioc">
      <dependency version="current">%{codac_rpm_prefix}-${project.artifactId}-TEST-S7-PSH0CORE-ioc</dependency>
      <dependency version="current">%{codac_rpm_prefix}-${project.artifactId}-TEST-S7-PSH0PLC-ioc</dependency>
      <dependency version="current">%{codac_rpm_prefix}-${project.artifactId}-TEST-S7-PSH0SYSM-ioc</dependency>
      <include type="script" scriptType="initd" file="plc-sample-iocs" />
     </package>
    </packaging>
   </configuration>
  </plugin>
 </plugins>
</build>
```



```
CURRENT-2:~/test/mvn-epics-commands/m-plc-sample/target> ls -1 *ioc*
codac-core-5.2-plc-sample-ioc-5.2.0.v1.0.0-0.el6.x86_64.rpm
codac-core-5.2-plc-sample-TEST-S7-PSH0CORE-ioc-5.2.0.v1.0.0-0.el6.x86_64.rpm
codac-core-5.2-plc-sample-TEST-S7-PSH0PLC-ioc-5.2.0.v1.0.0-0.el6.x86_64.rpm
codac-core-5.2-plc-sample-TEST-S7-PSH0SYSM-ioc-5.2.0.v1.0.0-0.el6.x86_64.rpm
```

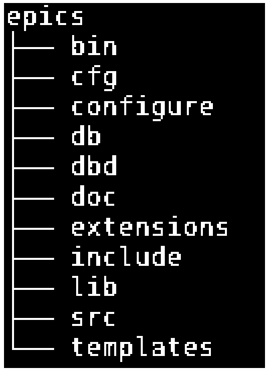
EPICS Base Packaging

ITER made a substantial effort to package EPICS itself in accordance with Red Hat Linux practices. This results in approx. 90 RPMs covering the following areas:

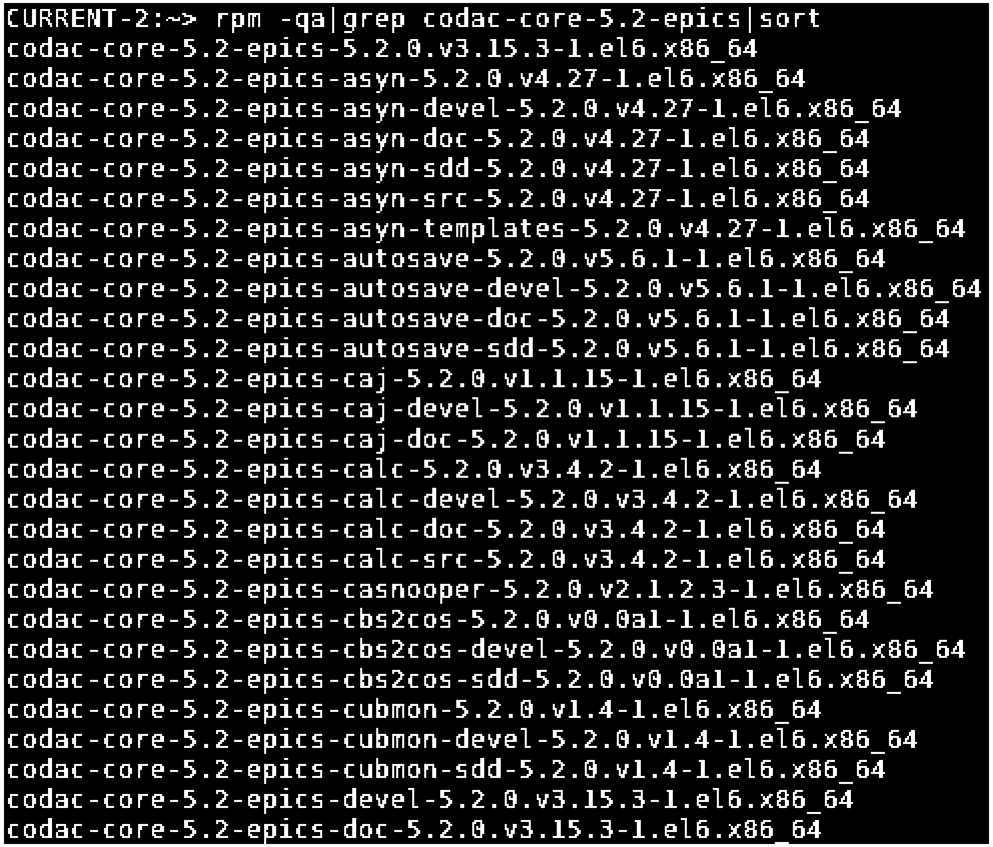
- EPICS base
- EPICS extensions and tools
- EPICS device support modules

Whenever possible, runtime and devel packages are separated.

ITER-specific EPICS packages are clearly separated too.



```
epics
├── bin
├── cfg
├── configure
├── db
├── dbd
├── doc
├── extensions
├── include
├── lib
├── src
└── templates
```



```
CURRENT-2:~> rpm -qa|grep codac-core-5.2-epics|sort
codac-core-5.2-epics-5.2.0.v3.15.3-1.el6.x86_64
codac-core-5.2-epics-asyn-5.2.0.v4.27-1.el6.x86_64
codac-core-5.2-epics-asyn-devel-5.2.0.v4.27-1.el6.x86_64
codac-core-5.2-epics-asyn-doc-5.2.0.v4.27-1.el6.x86_64
codac-core-5.2-epics-asyn-sdd-5.2.0.v4.27-1.el6.x86_64
codac-core-5.2-epics-asyn-src-5.2.0.v4.27-1.el6.x86_64
codac-core-5.2-epics-asyn-templates-5.2.0.v4.27-1.el6.x86_64
codac-core-5.2-epics-autosave-5.2.0.v5.6.1-1.el6.x86_64
codac-core-5.2-epics-autosave-devel-5.2.0.v5.6.1-1.el6.x86_64
codac-core-5.2-epics-autosave-doc-5.2.0.v5.6.1-1.el6.x86_64
codac-core-5.2-epics-autosave-sdd-5.2.0.v5.6.1-1.el6.x86_64
codac-core-5.2-epics-caj-5.2.0.v1.1.15-1.el6.x86_64
codac-core-5.2-epics-caj-devel-5.2.0.v1.1.15-1.el6.x86_64
codac-core-5.2-epics-caj-doc-5.2.0.v1.1.15-1.el6.x86_64
codac-core-5.2-epics-calc-5.2.0.v3.4.2-1.el6.x86_64
codac-core-5.2-epics-calc-devel-5.2.0.v3.4.2-1.el6.x86_64
codac-core-5.2-epics-calc-doc-5.2.0.v3.4.2-1.el6.x86_64
codac-core-5.2-epics-calc-src-5.2.0.v3.4.2-1.el6.x86_64
codac-core-5.2-epics-casnooper-5.2.0.v2.1.2.3-1.el6.x86_64
codac-core-5.2-epics-cbs2cos-5.2.0.v0.0a1-1.el6.x86_64
codac-core-5.2-epics-cbs2cos-devel-5.2.0.v0.0a1-1.el6.x86_64
codac-core-5.2-epics-cbs2cos-sdd-5.2.0.v0.0a1-1.el6.x86_64
codac-core-5.2-epics-cubmon-5.2.0.v1.4-1.el6.x86_64
codac-core-5.2-epics-cubmon-devel-5.2.0.v1.4-1.el6.x86_64
codac-core-5.2-epics-cubmon-sdd-5.2.0.v1.4-1.el6.x86_64
codac-core-5.2-epics-devel-5.2.0.v3.15.3-1.el6.x86_64
codac-core-5.2-epics-doc-5.2.0.v3.15.3-1.el6.x86_64
```

EPICS and CODAC profiles

- It is possible to have several CODAC versions installed on one machine, hence, several EPICS versions too.
- CODAC has profiles for different types of control system machines, which dictate which part of EPICS has to be installed

```
<meta name="ct">
  <summary>CODAC Core System ${codac.version} CODAC
Terminal</summary>
 <provides codac="true">system-role</provides>
  <!-- client/user applications -->
  <requires codac="true">css</requires>
  <requires codac="true">css-jms-send</requires>
  <requires codac="true">css-opi</requires>
  <requires codac="true">epics</requires>
  <requires codac="true">python-cachannel</requires>
  <requires codac="true">python-pyepics</requires>
  <requires codac="true">system-tests</requires>
  <requires codac="true">psps-fatsat-editor</requires>
</meta>
```

```
<meta name="pcf">
  <summary>CODAC Core System ${codac.version} PCF
Operation</summary>
  <provides codac="true">system-role</provides>
  ...
  <!-- epics modules -->
  <requires codac="true">epics-irio</requires>
  <requires codac="true">epics-mcoreutils</requires>
  <requires codac="true">epics-nisync</requires>
  <requires codac="true">epics-nisync-general-time</requires>
  <requires codac="true">epics-pxi6259</requires>
  <requires codac="true">epics-pxie6368</requires>
  <requires codac="true">epics-pxi6528</requires>
<!-- libraries -->
  <requires codac="true">irio</requires>
  <requires codac="true">dan-daq</requires>
  <requires codac="true">log-lib</requires>
  ...
</meta>
```