



ITER CODAC Development Environment – CODAC Core System



Rok Sabjan, Cosylab

PCaPAC 2010, Saskatoon, Canada

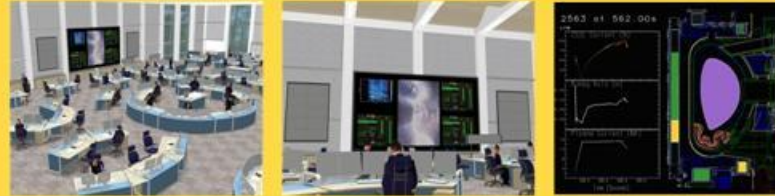


Acknowledgement

- ITER: Lana Abadie, Franck Di Maio, Kirti Mahajan, Petri Makijarvi, Denis Stepanov , Nadine Utzel, Anders Wallander
- Cosylab: Matjaz Kobal, Csaba Gajo, Janez Golob, Takashi Nakamoto, Klemen Strnisa, Anze Zagar, Klemen Zagar

Outline

- Motivation
- Explain Core System
- Overview of Development Environment
- Conclusions



Central Instrumentation and Control System

IN-FUND

CONTROL

INTERLOCK

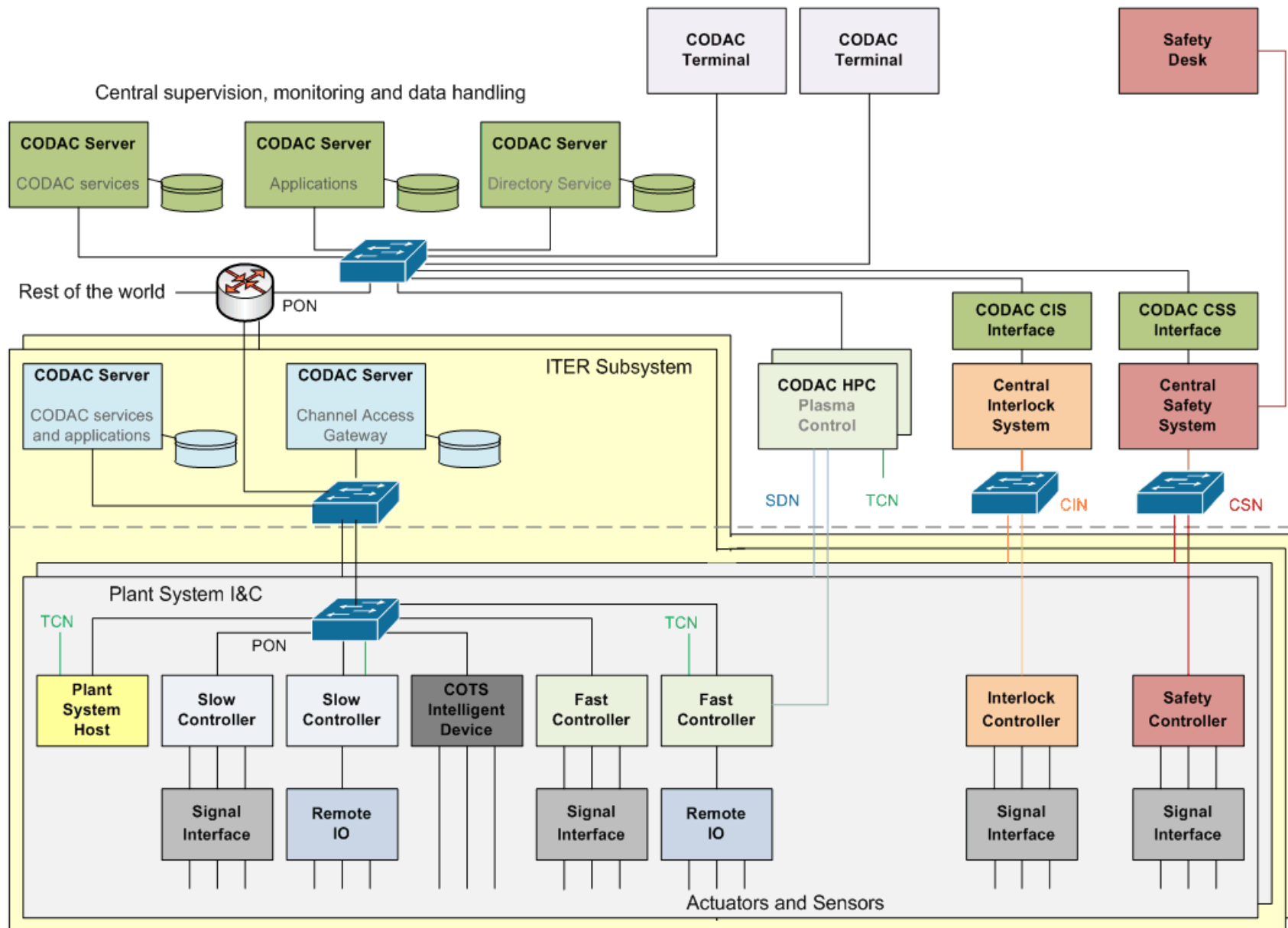
SAFETY

Plant System Instrumentation and Control

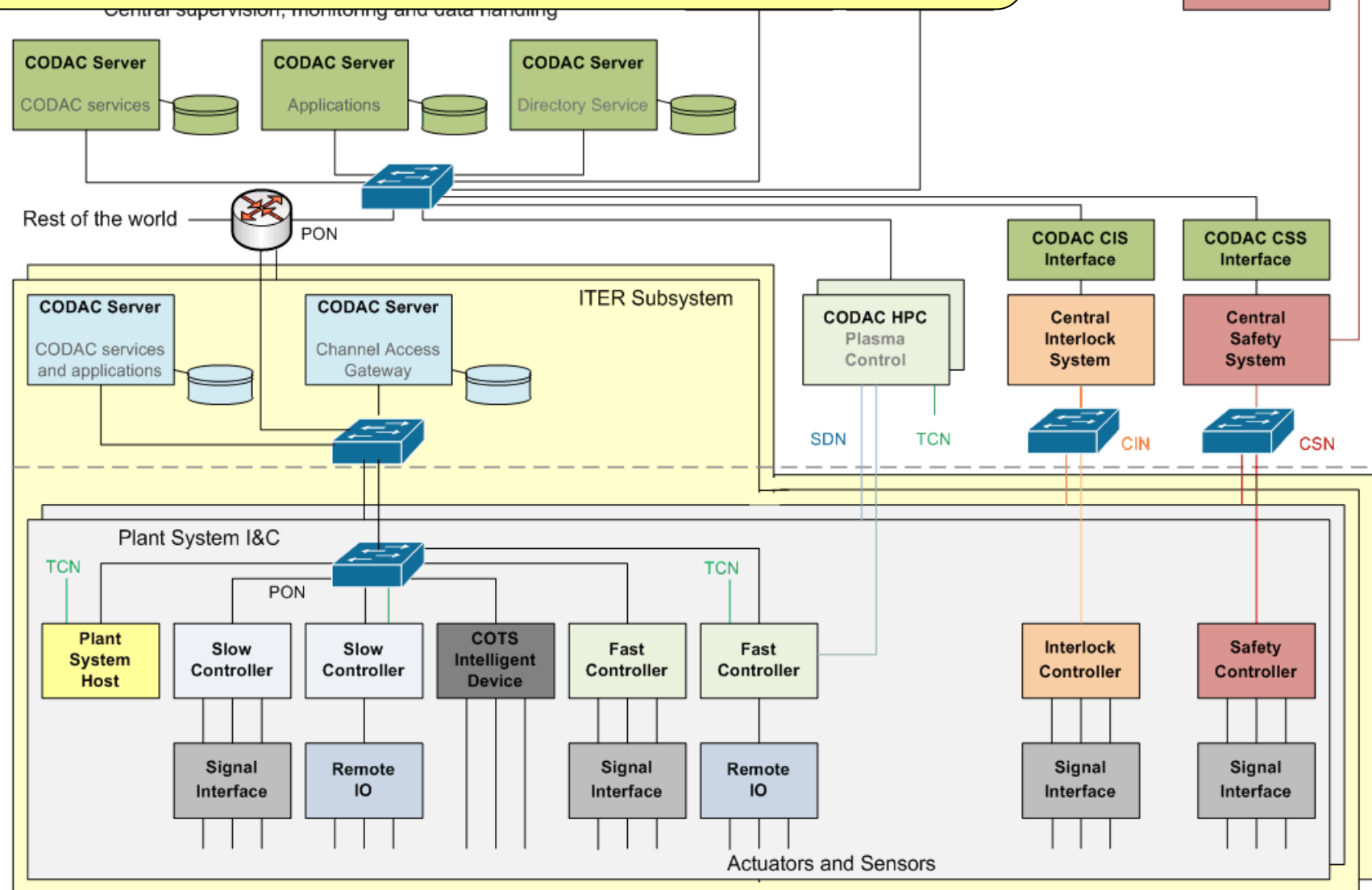
IN-KIND

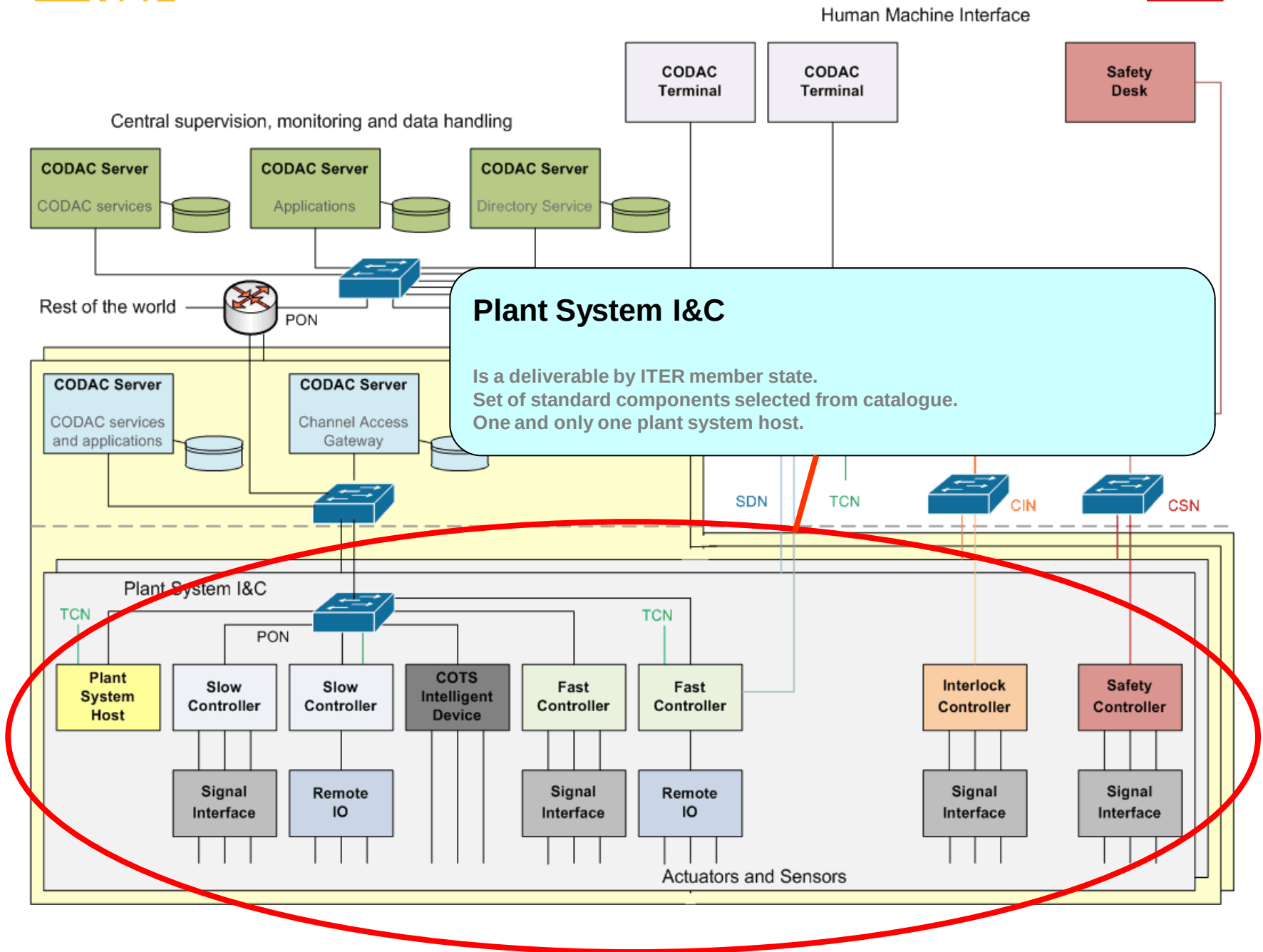


Human Machine Interface

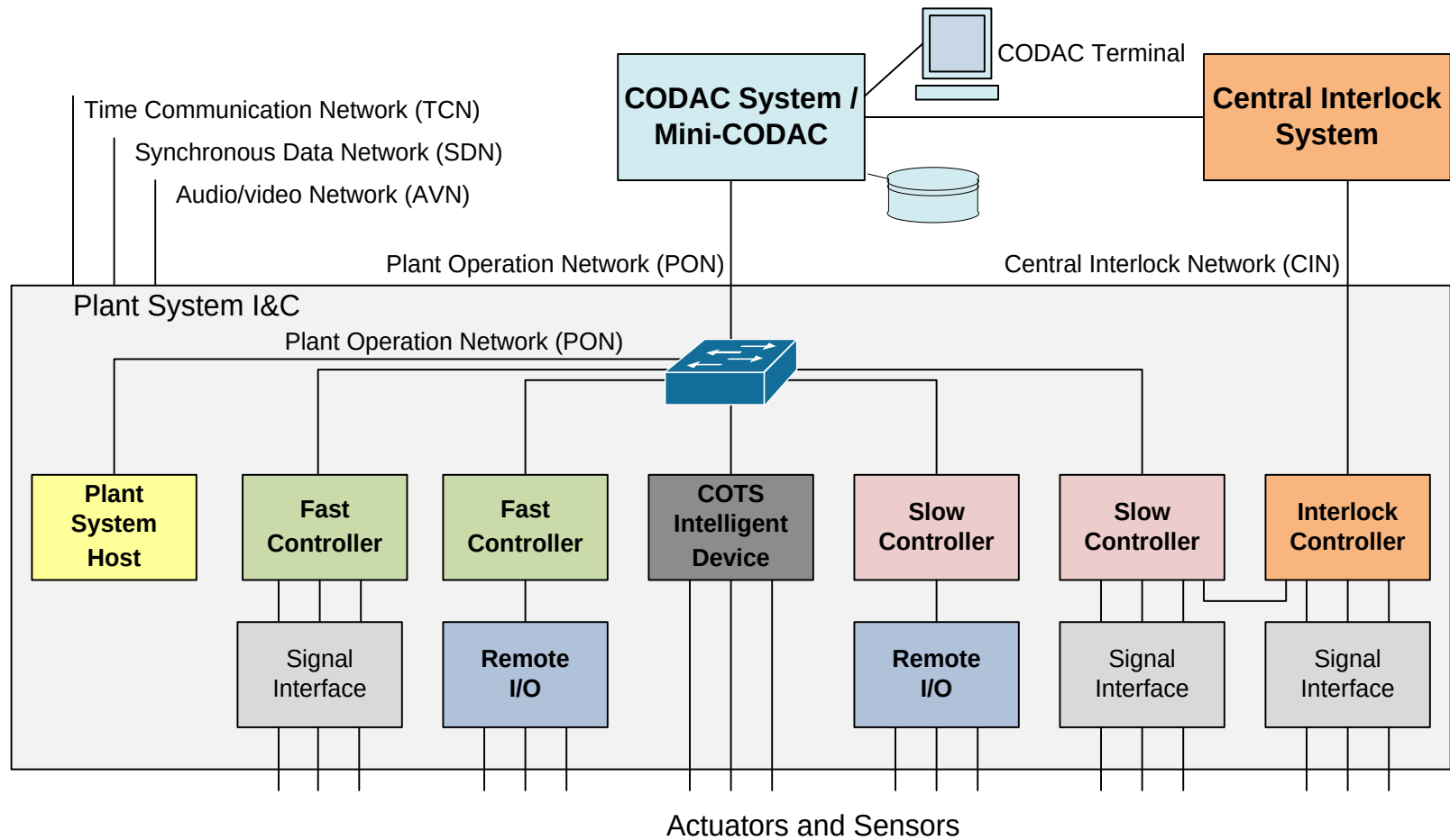


Finite set of “Lego blocks”, which can be selected and connected as required

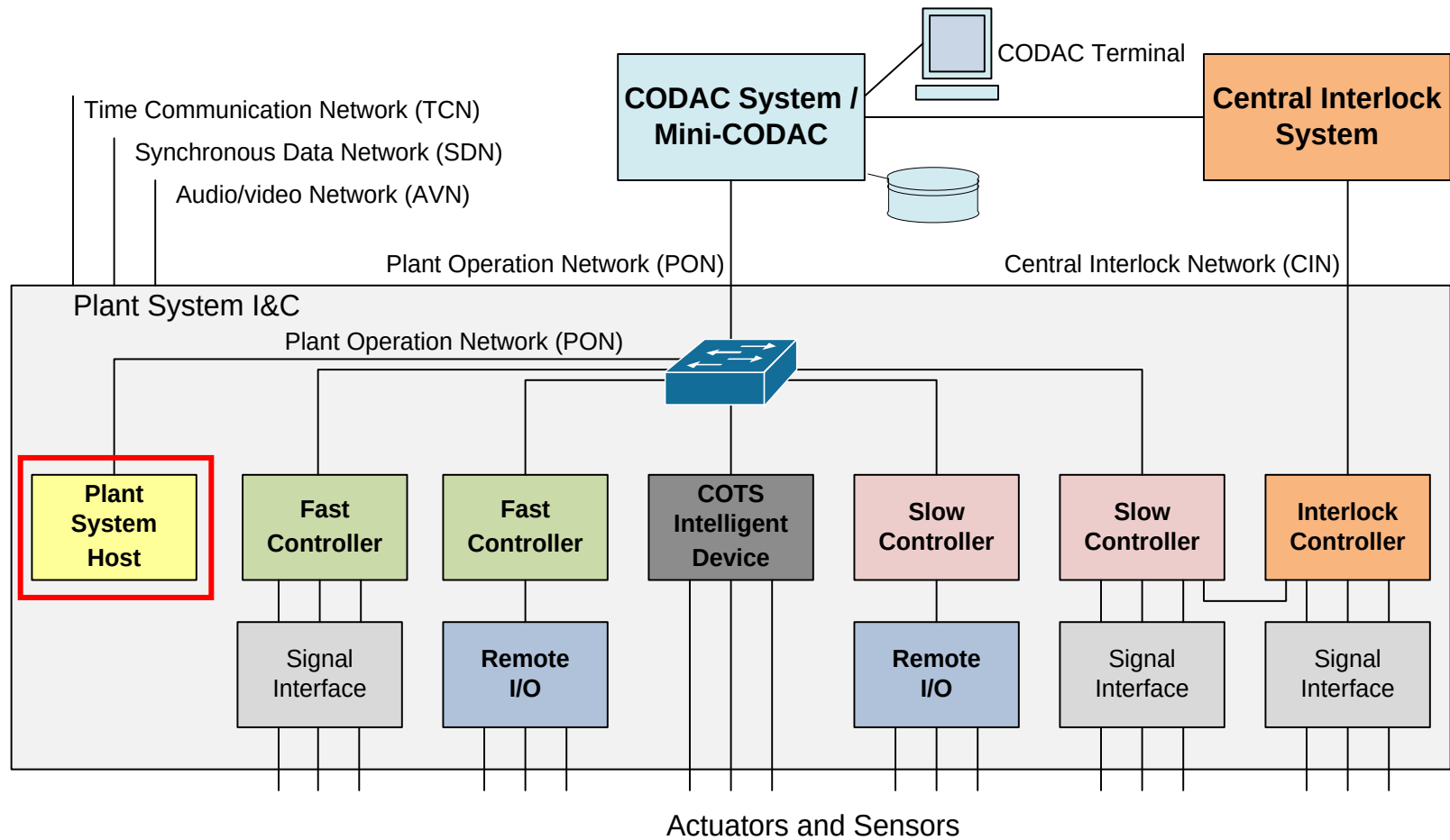




Plant System Instrumentation & Control (I&C) Architecture

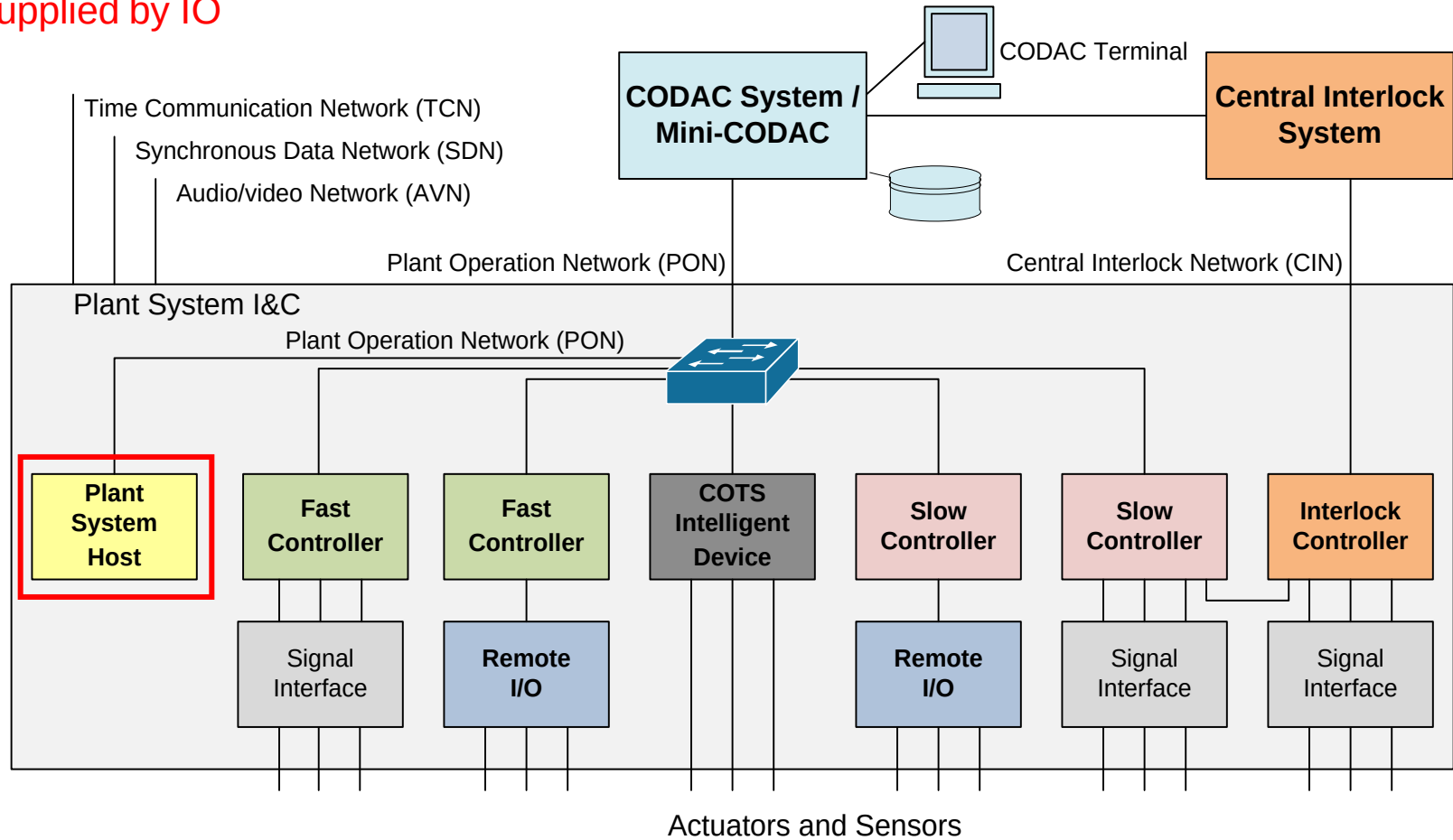


Plant System Instrumentation & Control (I&C) Architecture



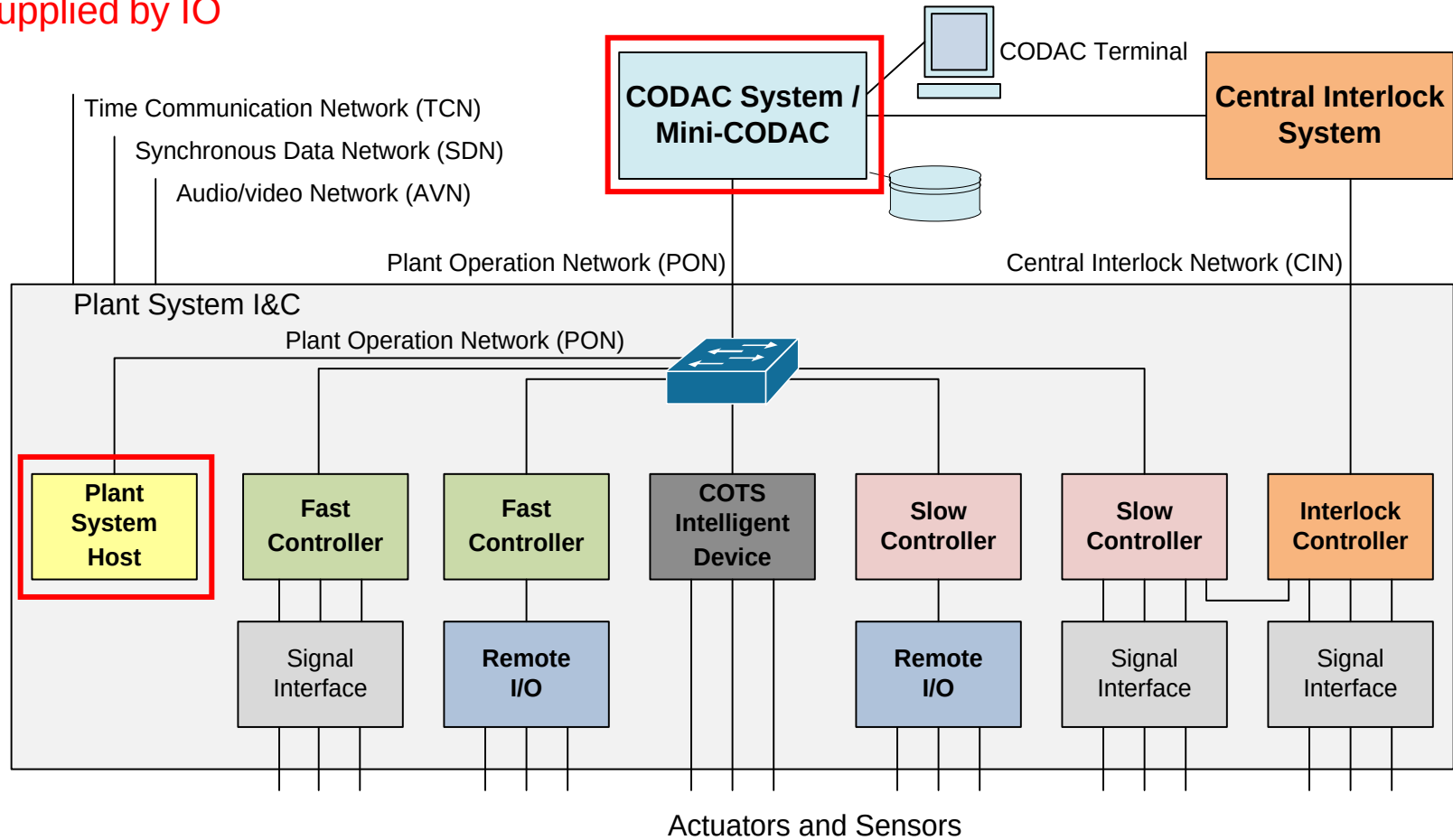
Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO



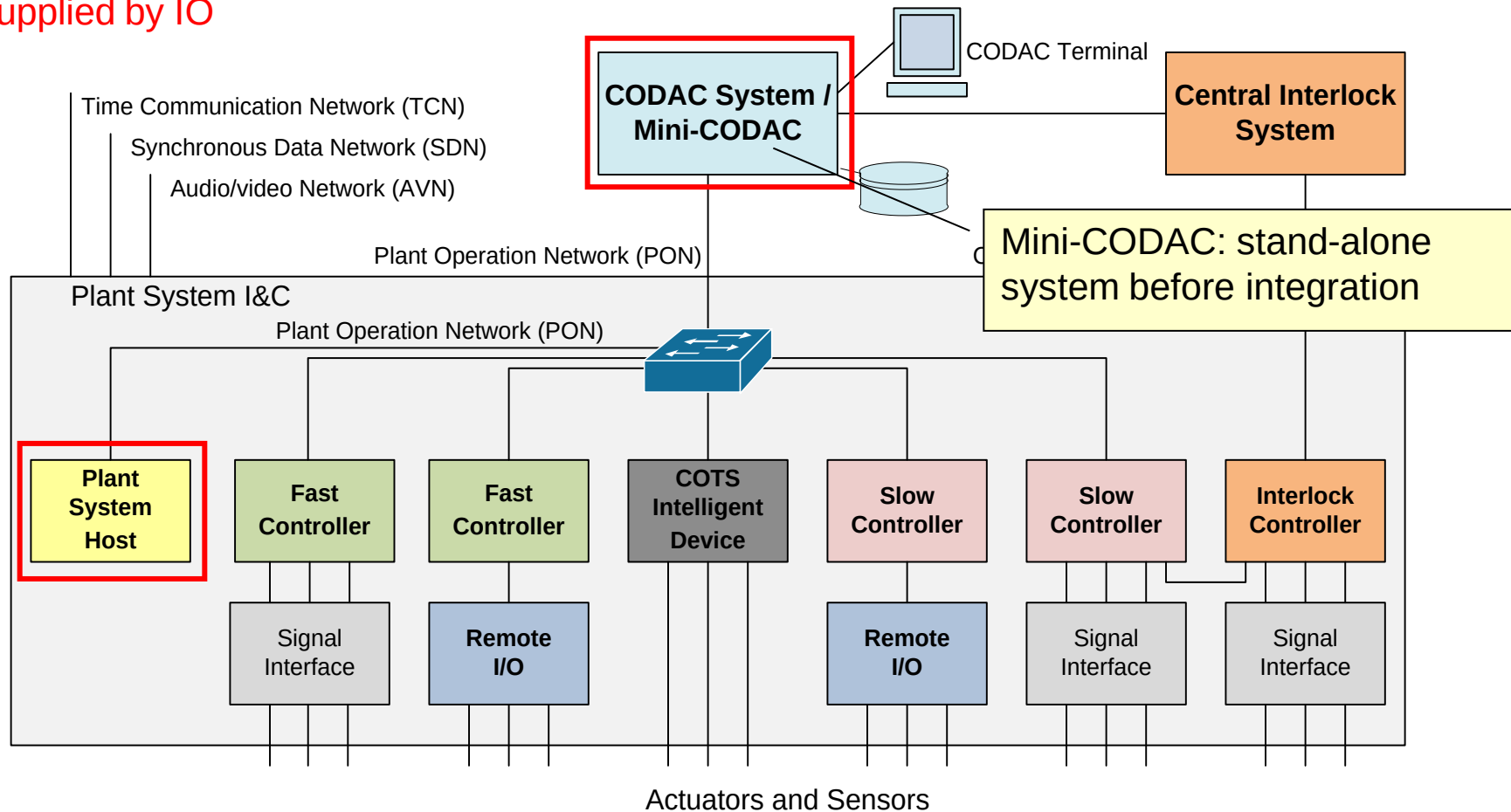
Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO



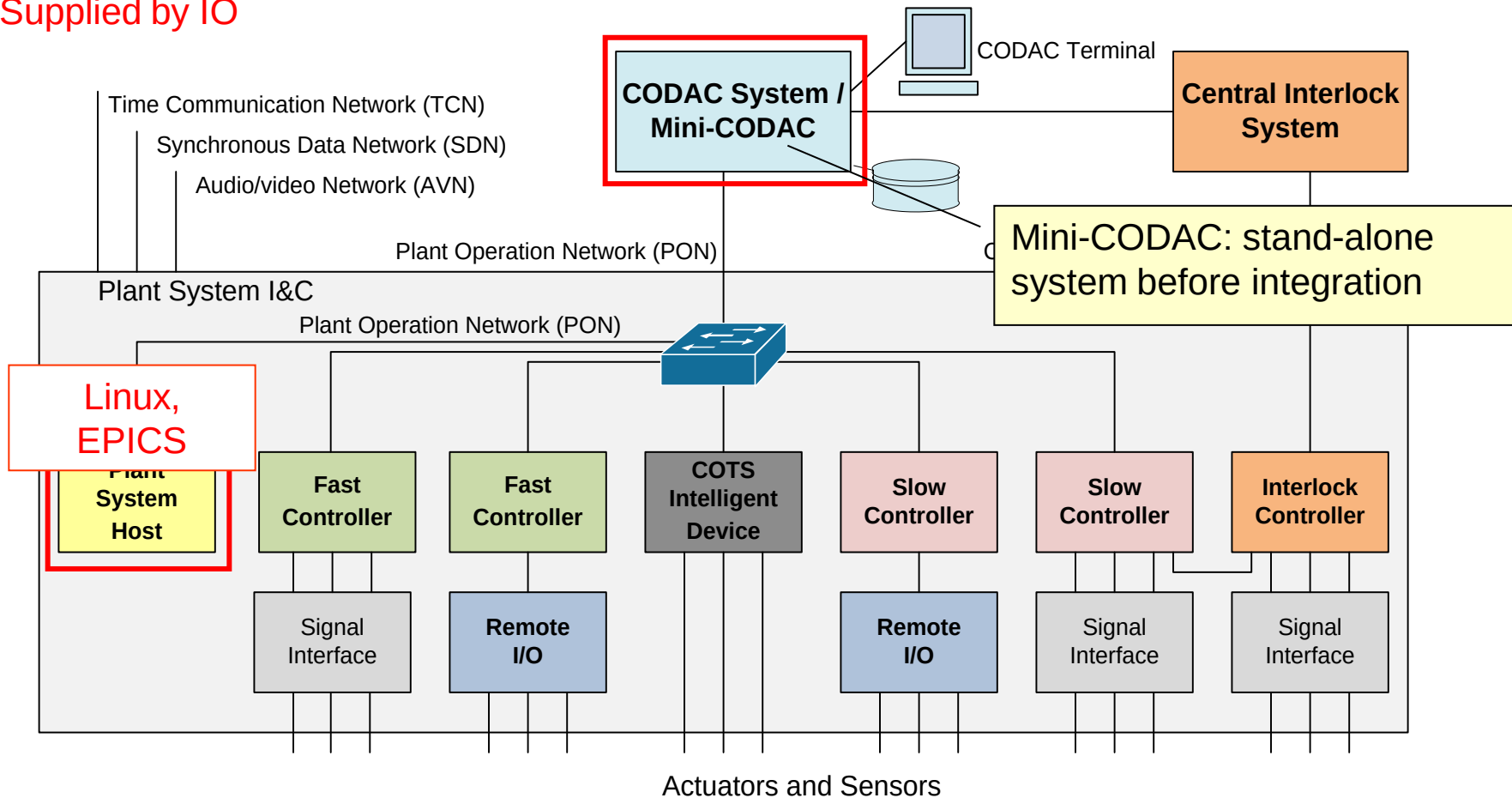
Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO



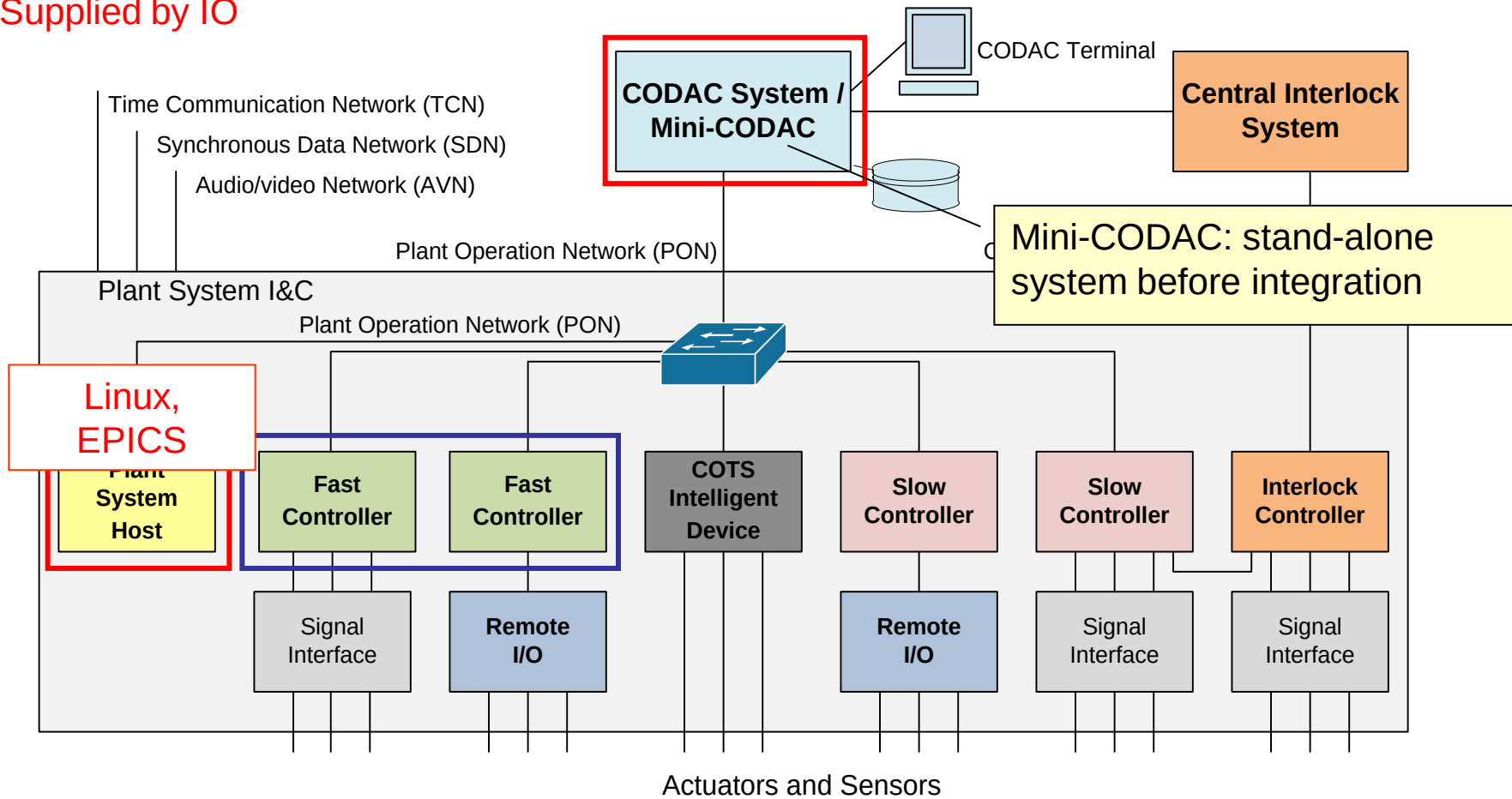
Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO



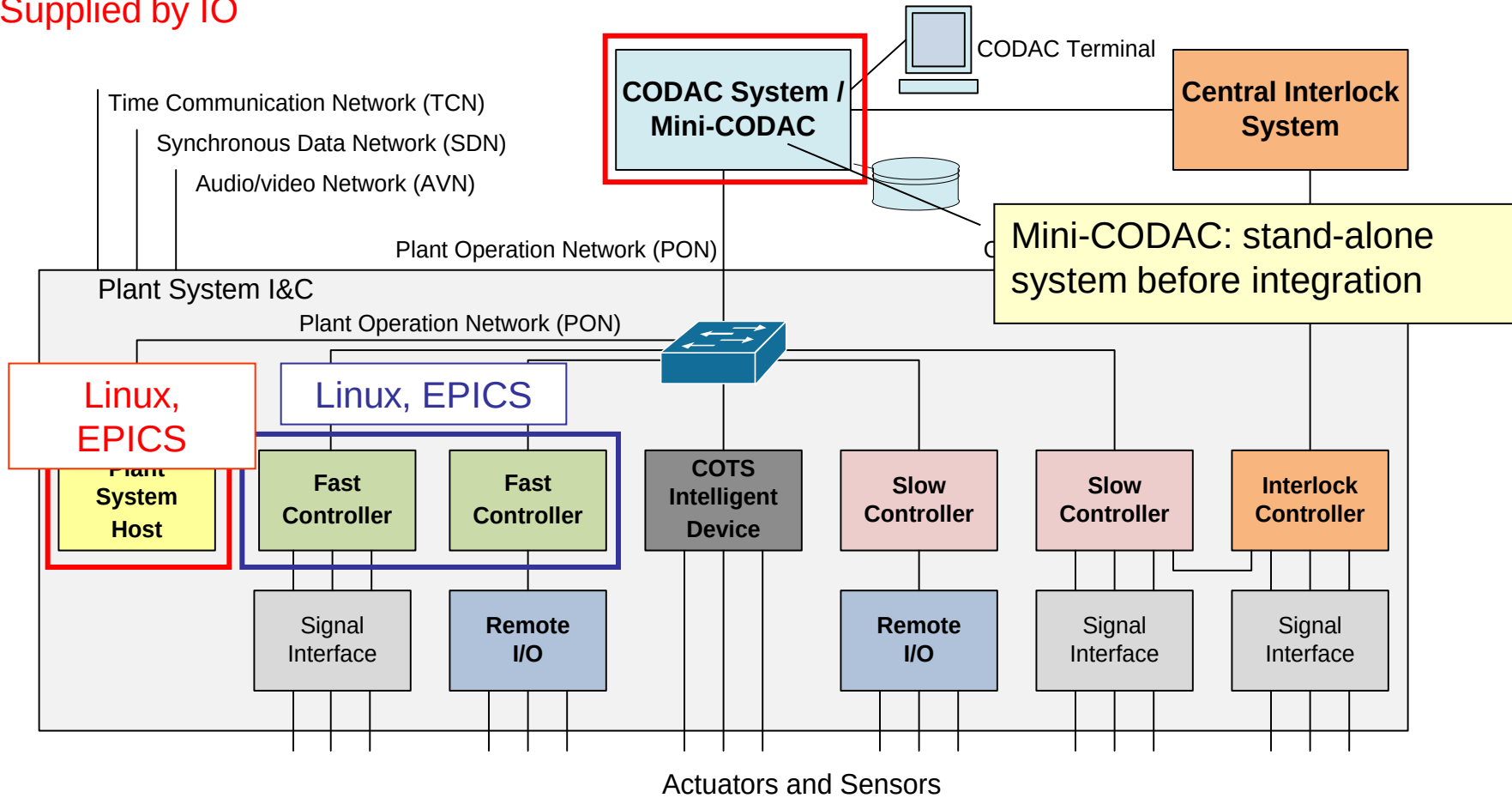
Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO



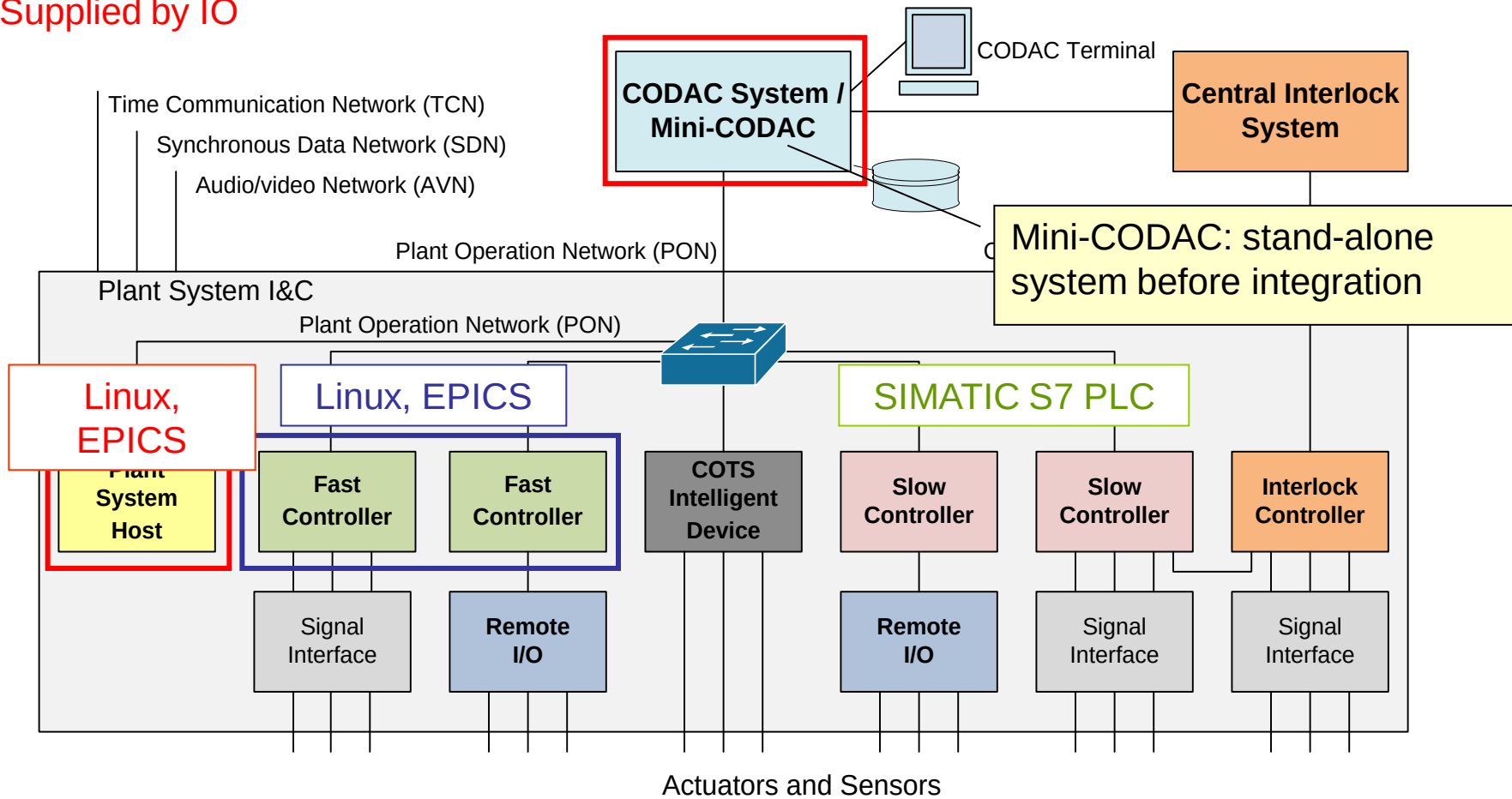
Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO



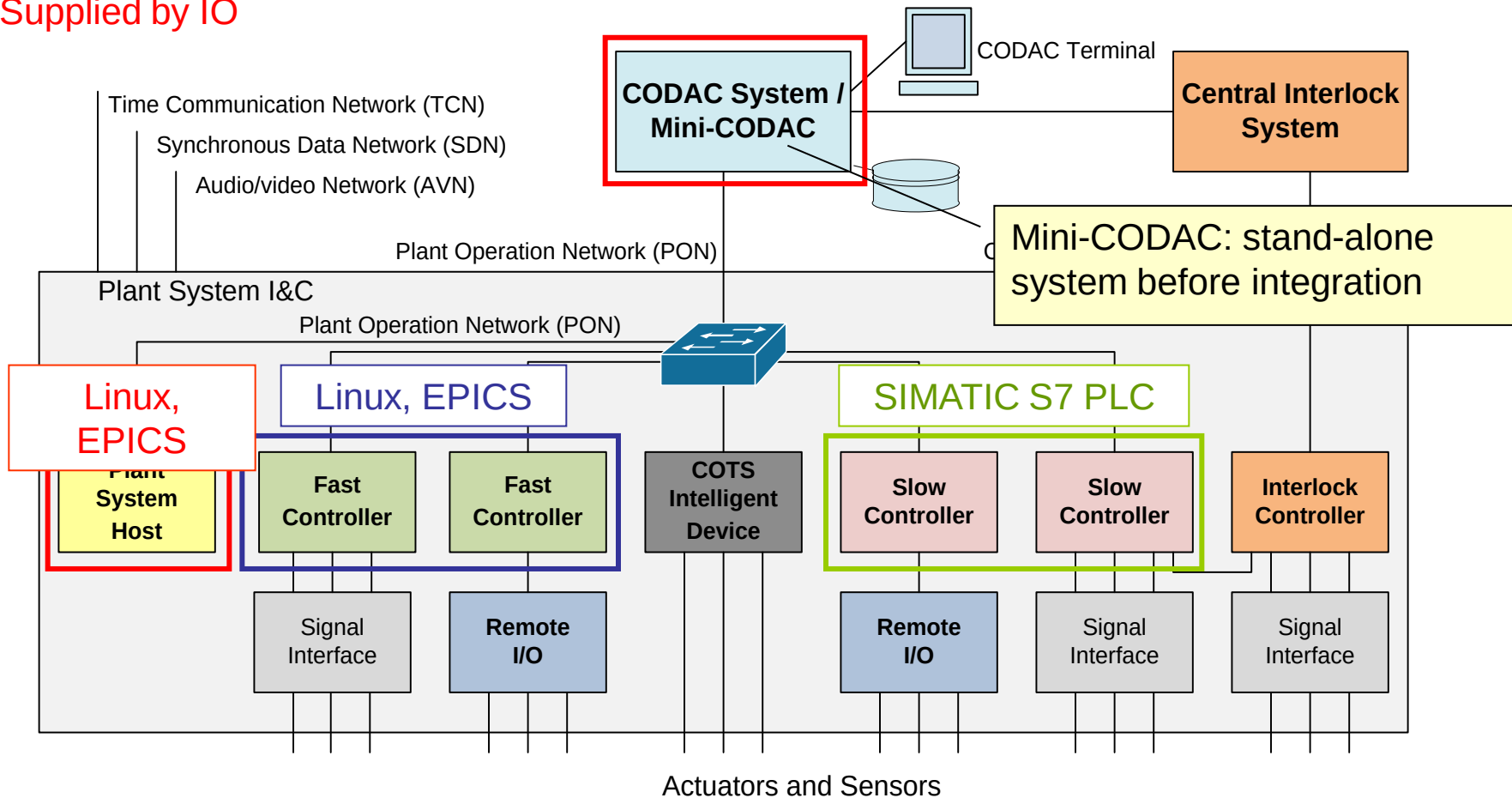
Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO



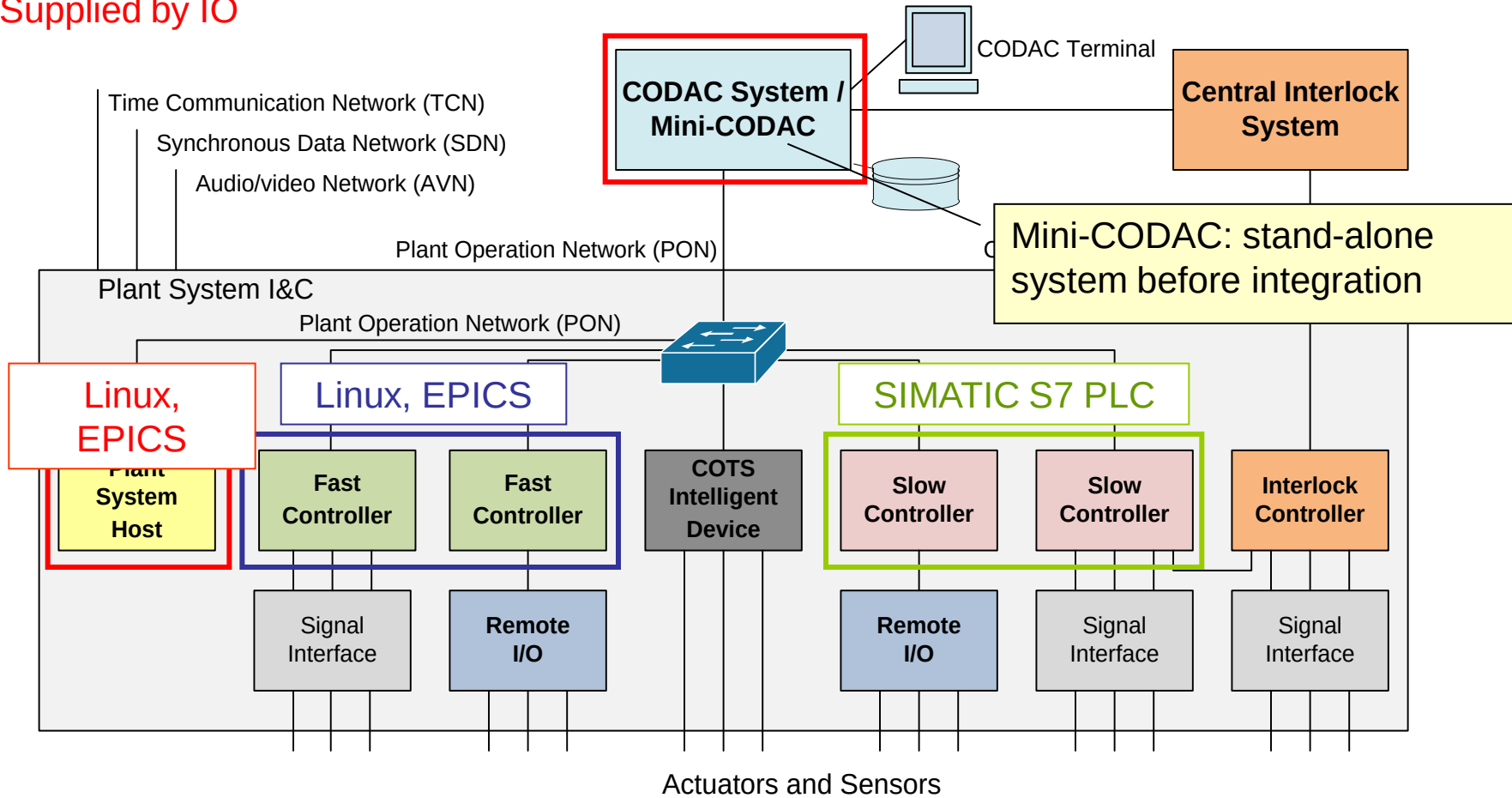
Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO



Plant System Instrumentation & Control (I&C) Architecture

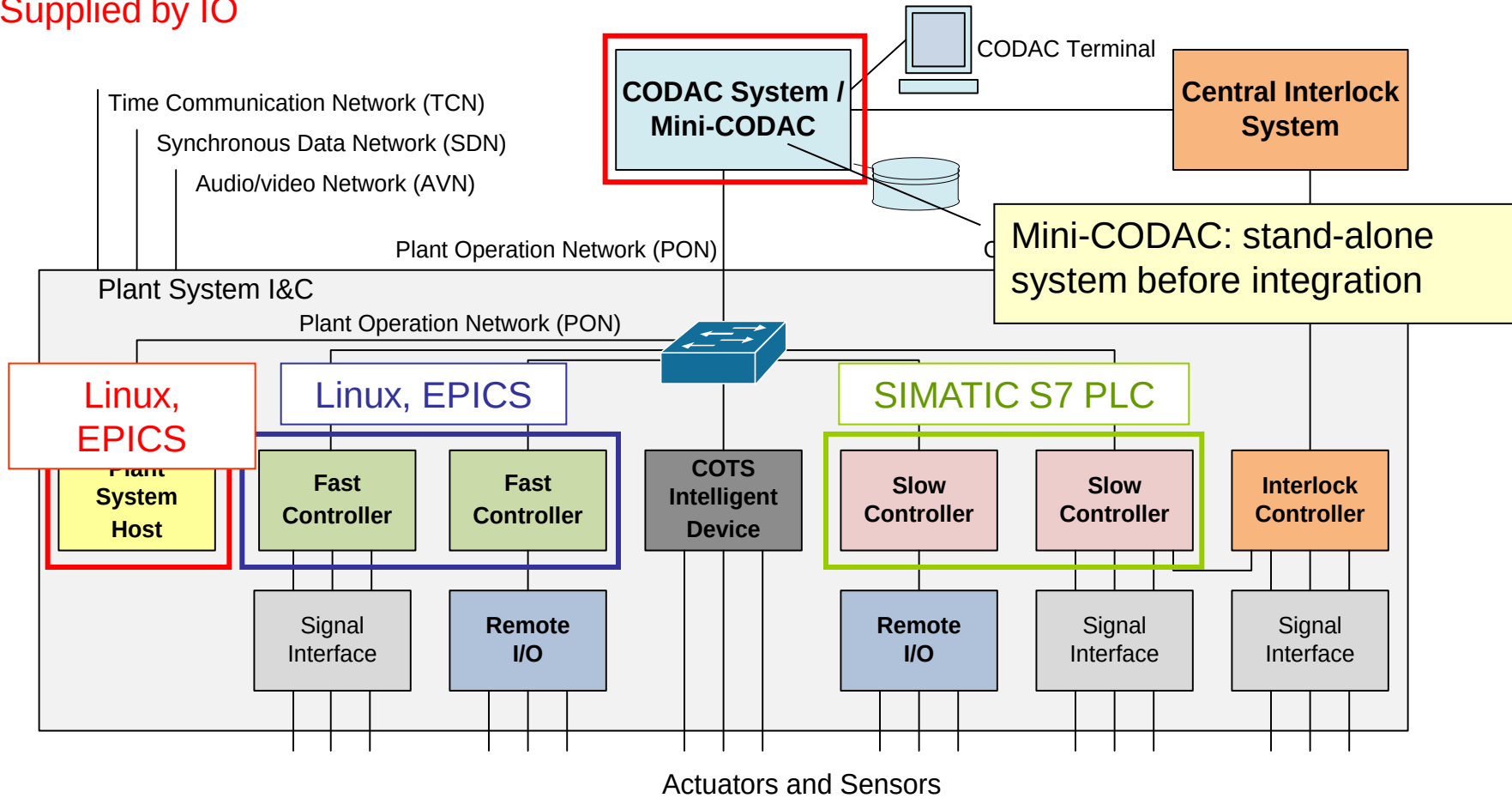
Supplied by IO



Complete software

Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO

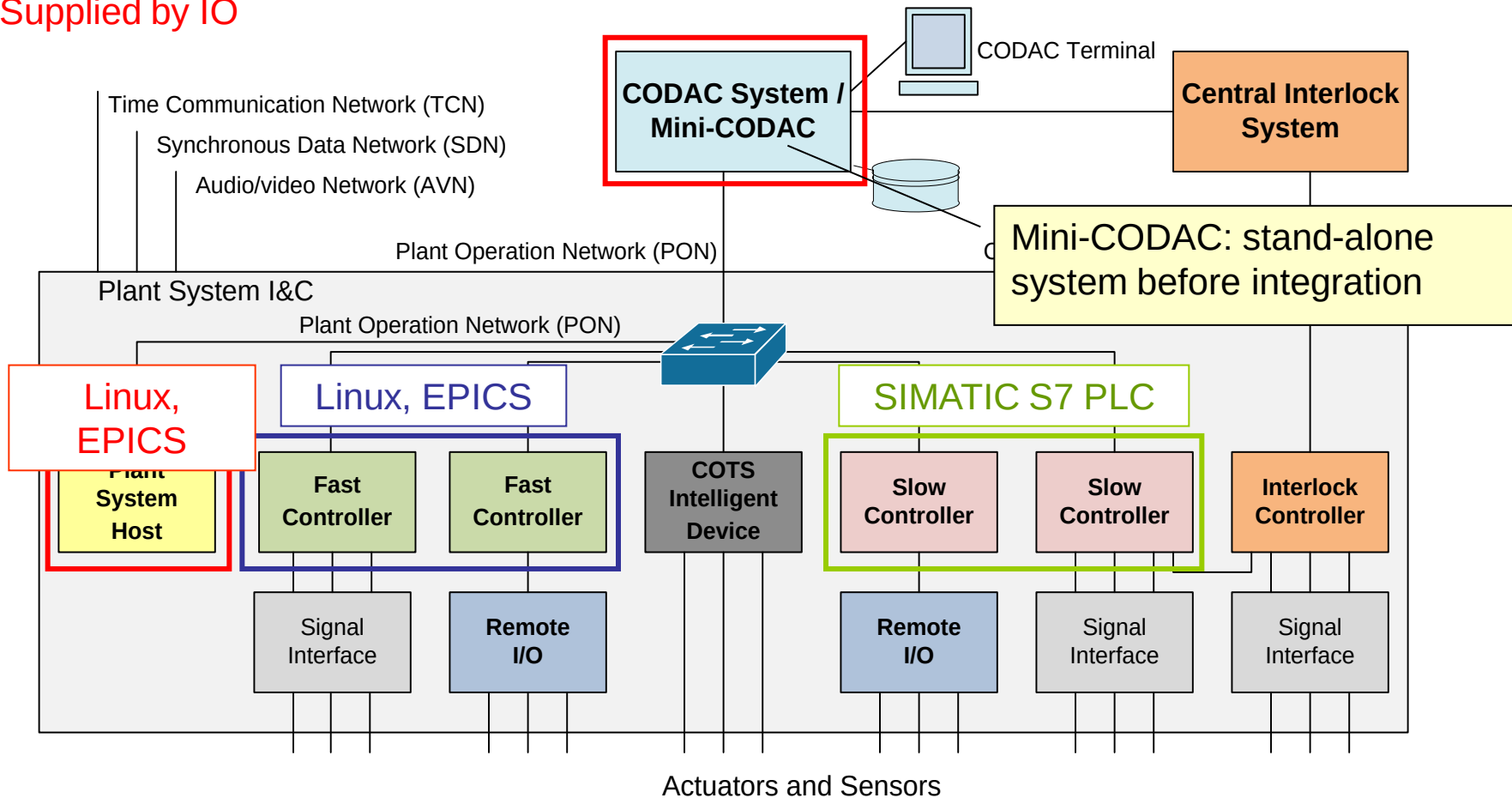


Complete software

Development tools and standard components

Plant System Instrumentation & Control (I&C) Architecture

Supplied by IO



Complete software

Development tools and standard components

Standard components for integration

Motivation

- I&C development must follow standards (Plant Control Design Handbook).
- A long development period with many distributed partners.
- Partners not familiar with ITER control system.
- ITER procedures should be easy to adopt.

Core System Releases Roadmap

2010/Q1	Version 1 <i>Preliminary</i>	Integration of PLCs EPICS distribution Self-description tools (preliminary) Examples
2011/Q1	Version 2 <i>(1st) Stable for developments</i>	Extensions for fast controllers Introduction of new Mini-CODAC tools APIs frozen
2012/Q1	Version 3 <i>(1st) Stable for tests</i>	New Mini-CODAC tools (CSS) Robustness

Released with Version 1

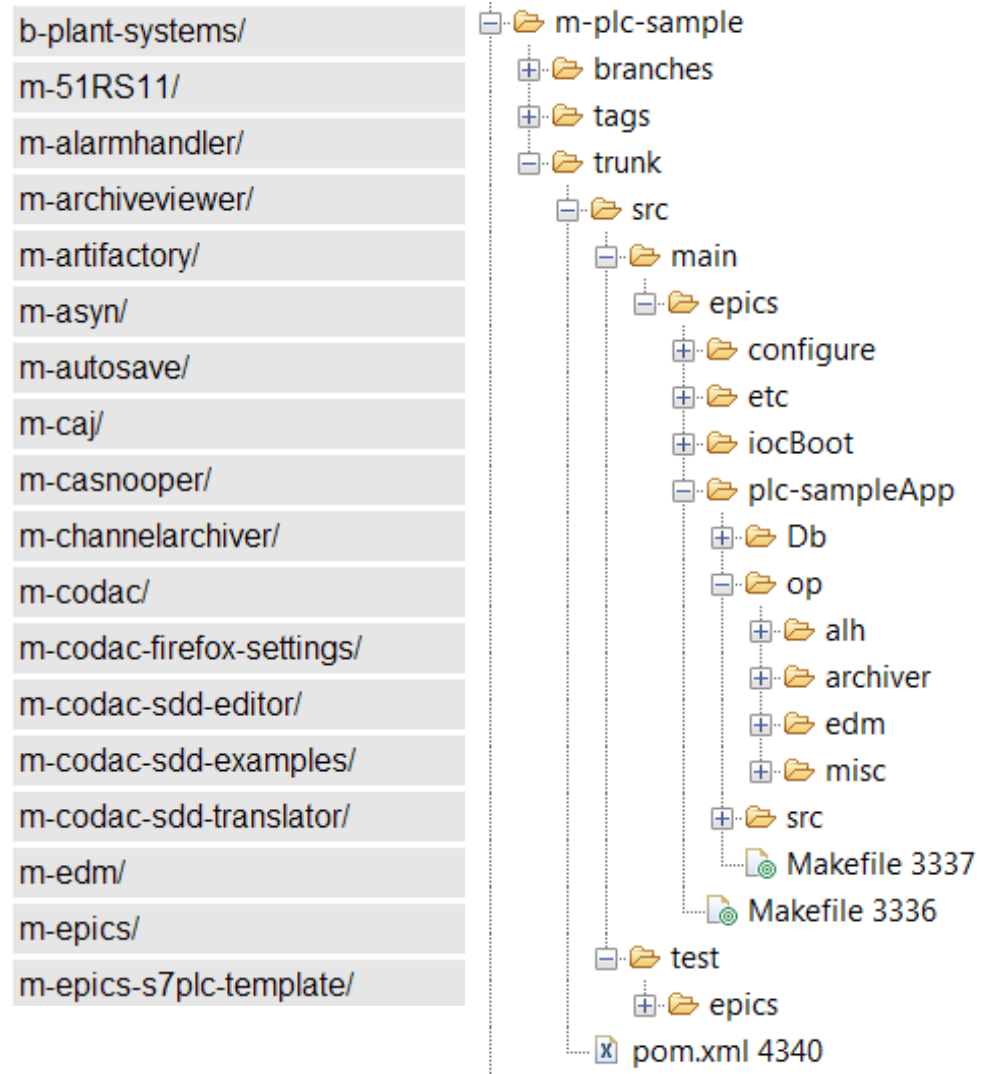
- A common development environment.
- Tools to enforce strict development procedures (SDD).
- Based on EPICS and Red Hat Enterprise Linux.
- Automatic installation via RHN Satellite Server.
- Two samples (PLC demo, Ion Cyclotron Heating (ICH) demo)
- Documentation, training and support.

Overview - Scope

- File organization.
- Development tools.
- Unit and integration testing.
- Automatic system build and testing.
- Packaging and distribution procedures.
- Issue tracking.

Overview – File organization

- Functional units.
- SVN
- Prescribed directory structure (Maven)
- src/, doc/, target/



Overview – Development tools

SDD editor



Self Description Data Editor v1.0.0

Plant System property configuration
Plant system I&C configuration
List of components
Help

Plant system I&C name : 4511RT

PBS-L1: 45-CODAC PPPP (4 charac.) 11RT

Component types and their properties

✗ Name (PPPPPP-TTT):4511RT-TSTPPPP: 11RT TTT: TST

List of its properties

+ New property

✗ + Duplicate

Name: AIU8-JT-CRC Description: unsigned char Enable Archiving: Yes Deployment Unit: slow controller

Archiving sampling frequency (msec): 100 Archiving sampling mode: Monitor (event-driven) Is a signal: true

Type: continuous Category: State Property Storage Specifier: volatile

Data Type: uint8 High oper.: Low oper.: Arch. Dead band: Unit: Precision: 2

HIHI: 255 HIGH: 255 LOW: 0 LOLO: 0 Static value:

- Hide property

Overview– Automatic testing

- Reduce QA effort.
- Problems are easily spotted.
- Should be part of every unit.
- Automatic tests are part of the build process.

Overview– Automatic testing

```

codac-dev@localhost:~/s-codac
File Edit View Terminal Tabs Help
RT  X codac-dev... X codac-dev... X codac-dev... X TUNEL X golobj@co... X
Testing m-extensionstop [OK]
Testing m-alarmhandler [OK]
Testing m-archiveviewer [FAILURE]
    Exit code: 1.
    Examine test.log and m-archiveviewer/test.log file for details.
Testing m-sequencer [OK]
Testing m-asyn [OK]
Testing m-autosave [OK]
Testing m-caj [OK]
Testing m-casnooper [OK]
Testing m-channelarchiver [FAILURE]
    Exit code: 1.
    Examine test.log and m-channelarchiver/test.log file for details.
Testing m-edm [OK]
Testing m-jca [OK]
Testing m-msi [OK]
Testing m-s7plc [OK]
Testing m-visualdct [FAILURE]
    Exit code: 1.
    Examine test.log and m-visualdct/test.log file for details.
Testing m-wiresharkCA [OK]
[codac-dev@localhost s-codac]$

```

Overview– Automatic testing

```

codac-dev@localhost: ~/s-codac
File Edit View Terminal Tabs Help
RT  X codac-dev... X codac-dev... X codac-dev... X TUNEL X golobj@co... X

Testing m-extensionstop [OK]
Testing m-alarmhandler [OK]
Testing m-archiveviewer [FAILURE]
    Exit code: 1.
    Examine test log and m-archiveviewer/test.log file for details.
Testing m-sequencer [codac-dev@localhost m-epics]$ sh test.sh
Testing m-asyn /home/codac-dev/s-codac/m-epics/test.sh
Testing m-autosave
Testing m-caj Executing test_install_dir... [OK]
Testing m-casnooper Executing test_executables... [OK]
Testing m-channelarch Executing test_create_test_app... [OK]
    Exit code: 1 Executing test_test_app... [OK]
    Examine test
Summary:
-----
Tests ok: 4/4
Tests failed: 0/4
Testing m-edm [OK]
Testing m-jca [FAILURE]
Testing m-msi
Testing m-s7plc [OK]
Testing m-visualdct [FAILURE]
    Exit code: 1.
    Examine test.log and m-visualdct/test.log file for details.
Testing m-wiresharkCA [OK]
[codac-dev@localhost s-codac]$

```

Overview– Automatic builds

- Automatic build from sources in SVN.
- Reduces build effort and removes human errors.
- Easy to spot problems.

Overview– Automatic builds

-  [New Job](#)
-  [Manage Hudson](#)
-  [People](#)
-  [Build History](#)
-  [Edit View](#)
-  [Delete View](#)
-  [Project Relationship](#)
-  [Check File Fingerprint](#)

Build Queue	
No builds in the queue.	
Build Executor Status	
#	Status
1	Idle

All	Branch rt53045_20091209	trunk	+			
S	W	Job ↓	Last Success	Last Failure	Last Duration	
		trunk-check-website-url	N/A	N/A	N/A	
		trunk-clean-mvn-repository	21 hr (#15)	N/A	1.6 sec	
		trunk-m-alarmhandler	44 min (#57)	N/A	17 sec	
		trunk-m-archiveviewer	44 min (#214)	N/A	19 sec	
		trunk-m-artifactory	N/A	N/A	N/A	
		trunk-m-asyn	37 min (#57)	5 hr 14 min (#54)	1 min 5 sec	
		trunk-m-autosave	44 min (#48)	N/A	9 sec	
		trunk-m-caj	2 hr 40 min (#32)	7 hr 12 min (#29)	5.8 sec	
		trunk-m-casnooper	43 min (#50)	N/A	8.3 sec	
		trunk-m-channelarchiver	43 min (#84)	4 hr 58 min (#79)	2 min 5 sec	
		trunk-m-edm	41 min (#69)	6 hr 0 min (#63)	3 min 53 sec	
		trunk-m-epics	47 min (#49)	5 hr 26 min (#44)	2 min 14 sec	
		trunk-m-extensionstop	44 min (#42)	6 hr 0 min (#37)	6.1 sec	
		trunk-m-ich-core-demo	31 min (#56)	45 min (#55)	1 min 36 sec	
		trunk-m-ica	29 min (#31)	7 hr 17 min (#28)	5.8 sec	
		trunk-m-maven	44 min (#32)	9 days 23 hr (#11)	5.6 sec	

ENABLE AUTO REFRESH

 [add description](#)

Overview - Packaging

- All code packaged in RPMs.
- RHN satellite is used for package distribution.

Overview - Packaging

[English \(change\)](#)
[knowledgebase](#)
[Documentation](#)

USER: rhn-admin

ORGANIZATION: ITER Organization

Preferences

Sign Out

Systems

Search

Overview

Systems

Errata

Channels

Configuration

Schedule

Users

Admin

Help

1 SYSTEM SELECTED

MANAGE

CLEAR

Overview

Your Account

Your Preferences

Locale Preferences

Subscription Management

Organization Trusts

Overview

Tasks

Inactive Systems

Most Critical Systems

Overview Legend

OK

Critical

Warning

Unknown

Locked

Kickstarting

Pending Actions

Failed Actions

Completed Actions

Manage Entitlements and Subscriptions:

My Organization | RHN Satellite-Wide

Register Systems

Manage Activation Keys

Manage Kickstarts

Manage Configuration Files

Manage RHN Satellite Organizations

Configure RHN Satellite

ws3

1 Day(s)

fc1

2 Week(s)

fc1

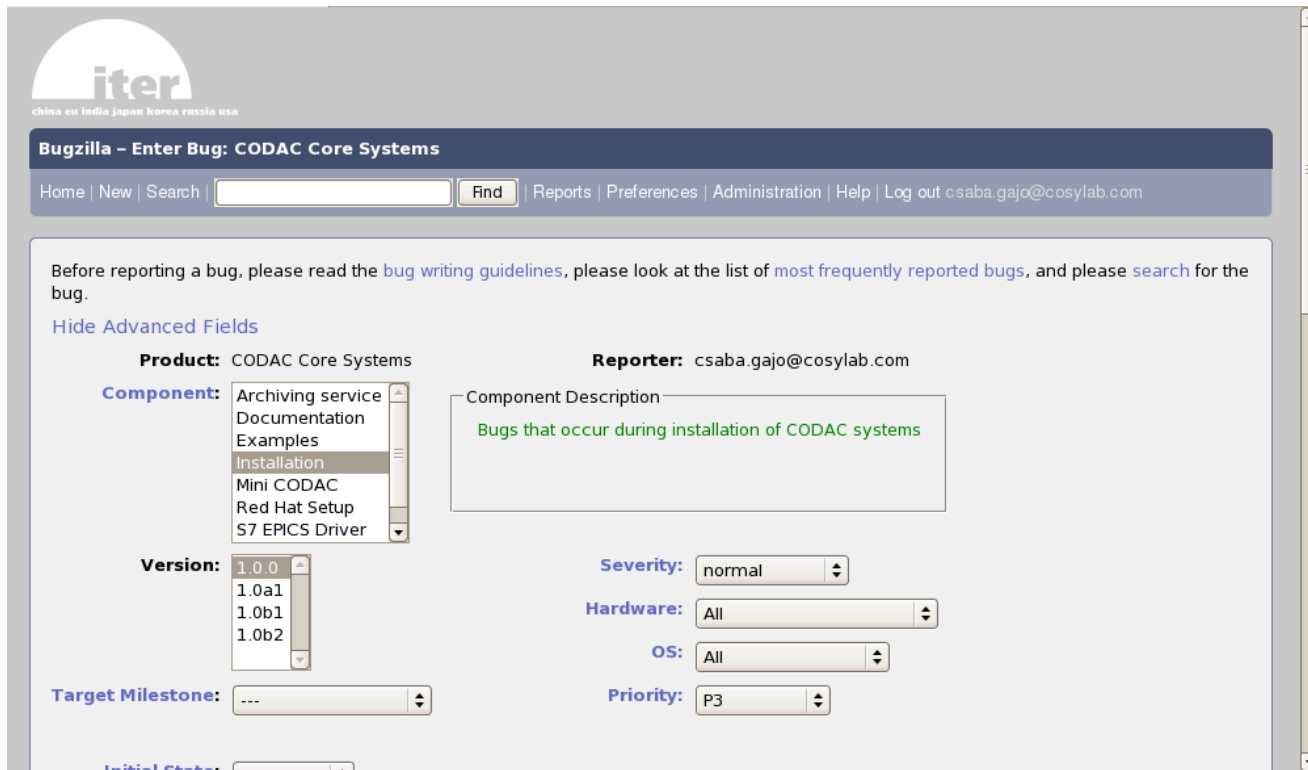
2 Week(s)

View All Inactive Systems (3)

System Name	All Updates	Security Errata	Bugfix Errata	Enhancement Errata
psh2	299	102	172	25
unknown	299	102	172	25
unknown	299	102	172	25
ws3	299	102	172	25

Overview– Issue tracking

- Bugzilla is used to track bugs.
- Bugs can be reported by all users.



Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#), and please [search](#) for the bug.

[Hide Advanced Fields](#)

Product: CODAC Core Systems

Reporter: csaba.gajo@cosylab.com

Component: Archiving service
Documentation
Examples
Installation
Mini CODAC
Red Hat Setup
S7 EPICS Driver

Version: 1.0.0
1.0a1
1.0b1
1.0b2

Target Milestone: ...

Component Description:
Bugs that occur during installation of CODAC systems

Severity: normal

Hardware: All

OS: All

Priority: P3

Initial State: ...

Overview– Web page

- All CODAC information found in one place.
- CODAC and EPICS documentation.



the way to new energy

Links to infrastructure

ITER CODAC Development

CODAC IDM Home Page | CODAC Team Site | Contact Web Master

Last Published: 2010-02-05 | Version: 1.0b2

Documentation

- Tutorials
- Manuals
- FAQ
- Release Notes
- Roadmap
- Dependencies

Support

- Remote Access
- Report a Bug
- Development Process
- Infrastructure

Download

- CODAC Development Environment

Project Documentation

- Project Information

CODAC Manuals

\$Id: manuals.apt 3801 2010-02-04 15:39:45Z golobj \$

Document Status				
Document	Owner	Status	Next action - CSL	Next action - ITER
CODAC web page	kzagar	First proposal		Make review
CODAC Core Systems Overview	fdimaio	Draft		Finalize
CODAC Installation Manual	mkobal	Passed review 1		Make review
CODAC Developer Manual	mkobal	Passed review 1		Make review
CODAC Software Development Environment	jkamenik	CSL review	Update	Provide licensing policy
CODAC Development Environment Overview Presentation	kzagar	Ready for review 1		Make review
CODAC Versioning System	azagar	CSL review	Make review	
ITER Plugin For Maven	cgajo	CSL review	Make review	
Self Description Editor Manual	dstepanov	Draft		Finalize
Directory Structure of Software Repository	mkobal	Ready for review 1		Make review
Installed System Directory Structure	mkobal	Ready for review 2		Make review
ICH User Manual	nutzel	First proposal	Review	Review
ICH Developer Manual	nutzel	First proposal	Review	Review
PLC Sample Application Guide	kmahajan	First proposal	Review	Review
CODAC Core Support And Development Process	kzaqar	CSL review	Make review	

Core System Releases Roadmap

2010/Q1	Version 1 <i>Preliminary</i>	Integration of PLCs EPICS distribution Self-description tools (preliminary) Examples
2011/Q1	Version 2 <i>(1st) Stable for developments</i>	Extensions for fast controllers Introduction of new Mini-CODAC tools APIs frozen
2012/Q1	Version 3 <i>(1st) Stable for tests</i>	New Mini-CODAC tools (CSS) Robustness

Conclusions

- Standardization of EPICS development and deployment
 - Simplified installation for Red Hat systems (tarball, RPMs?).
 - Reduced learning curve (tools and documentation)
- Current version (1) is preliminary: focus on development infrastructure
 - All cases are not yet covered.
 - Basic sample applications and preliminary SDD tools
- Will be improved with newer versions

**Come and try out CODAC Core System v1:
it will make ITER and EPICS better**

<http://www.iter.org/ORG/TEAM/CHD/CID/CODAC/Pages/CoreSystem.aspx>

Thank You for Your Attention