

# Intro to CS 3.0

- The CS Framework...
- Ideas behind CS
- Usage
- News and project management
- What about HITRAP?

# Definition of the CS framework

Ansatz:

Back-end (SCADA, GUI, ...)

Middleware (communication, ...)

Front-end (devices, drivers, ...)

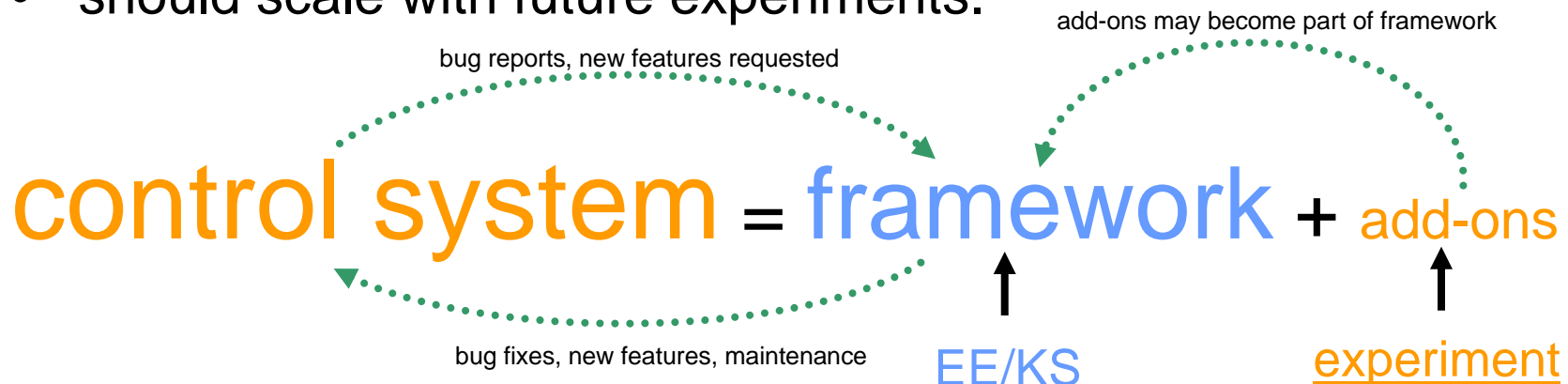
"Three-layer  
architecture"

- optional usage of sequencers (= sequences of commands)
- optional usage of state machines (= states and actions)
- optional GUIs on the device level
- optional interfaces to control systems of accelerator and DAQ
- standardization via a framework, here: CS framework

⇒ **Main emphasis: device control, not process control**

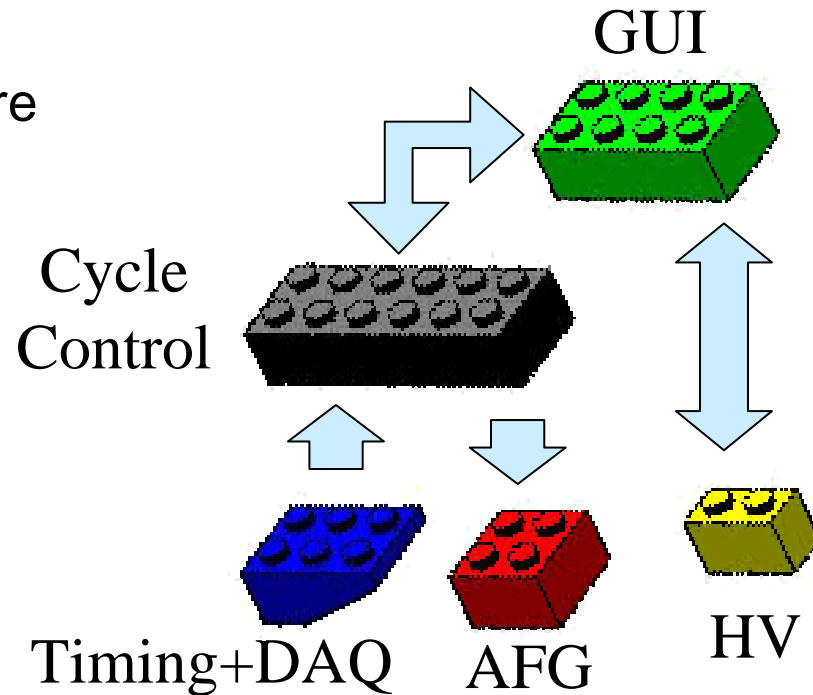
# Definition of the CS framework

- provides features that are commonly needed by many experiments.
- can be maintained by a dedicated and central group.
- allows for exchanging software and know-how.
- saves man power.
- should scale with future experiments.

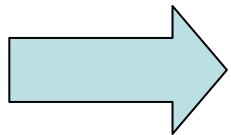


# Idea behind the CS framework

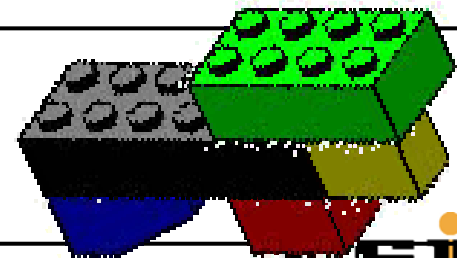
- Distributed, individual objects are responsible for subtasks, as:
  - User interface
  - Cycle control
  - Acquisition
  - Devices



- No intrinsic bottleneck: Everything can talk to everything



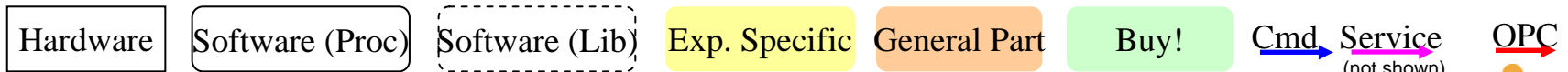
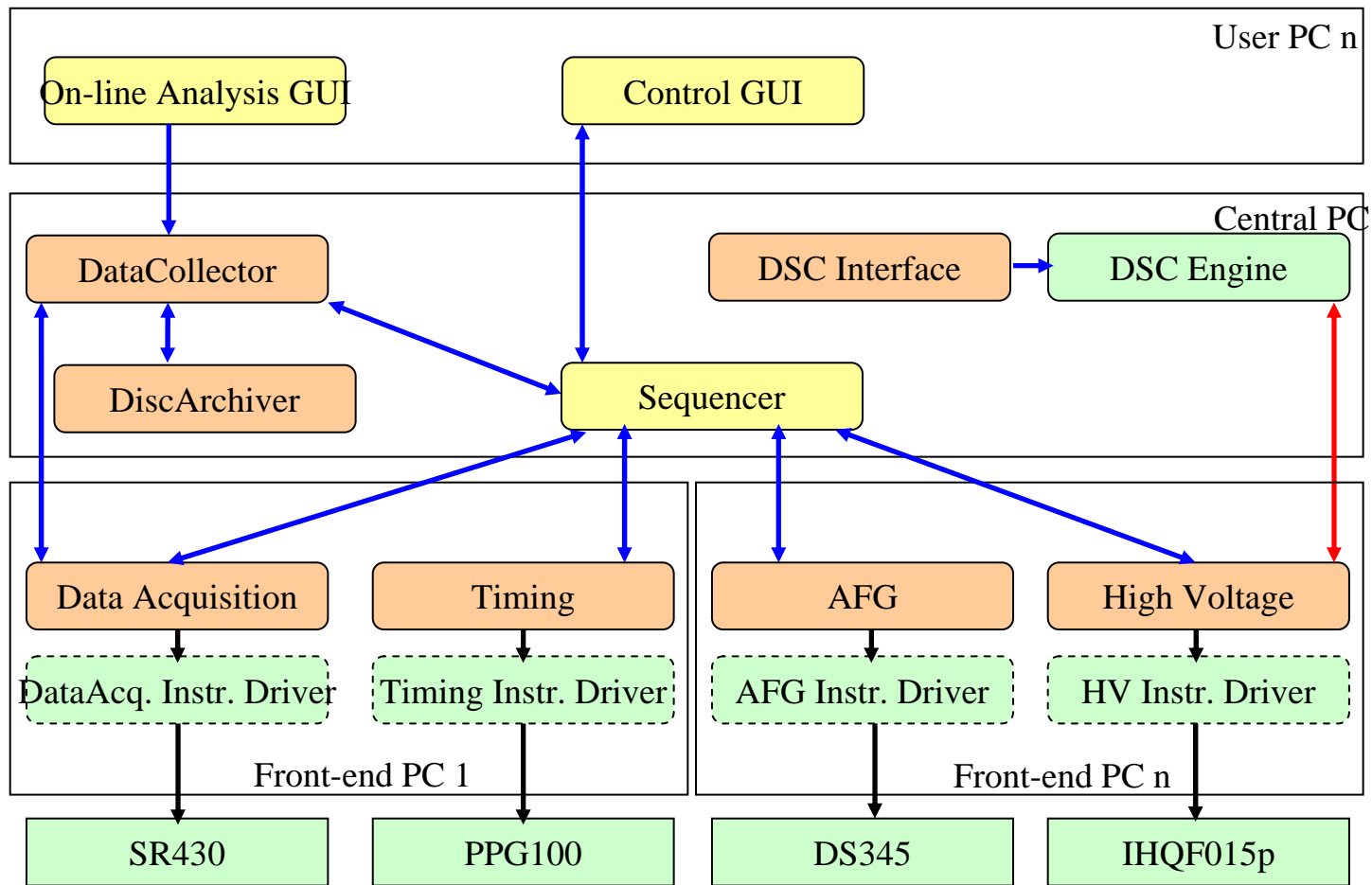
Using 'standard' pieces:  
Flexible system, that can even during  
runtime be adapted to the experiment



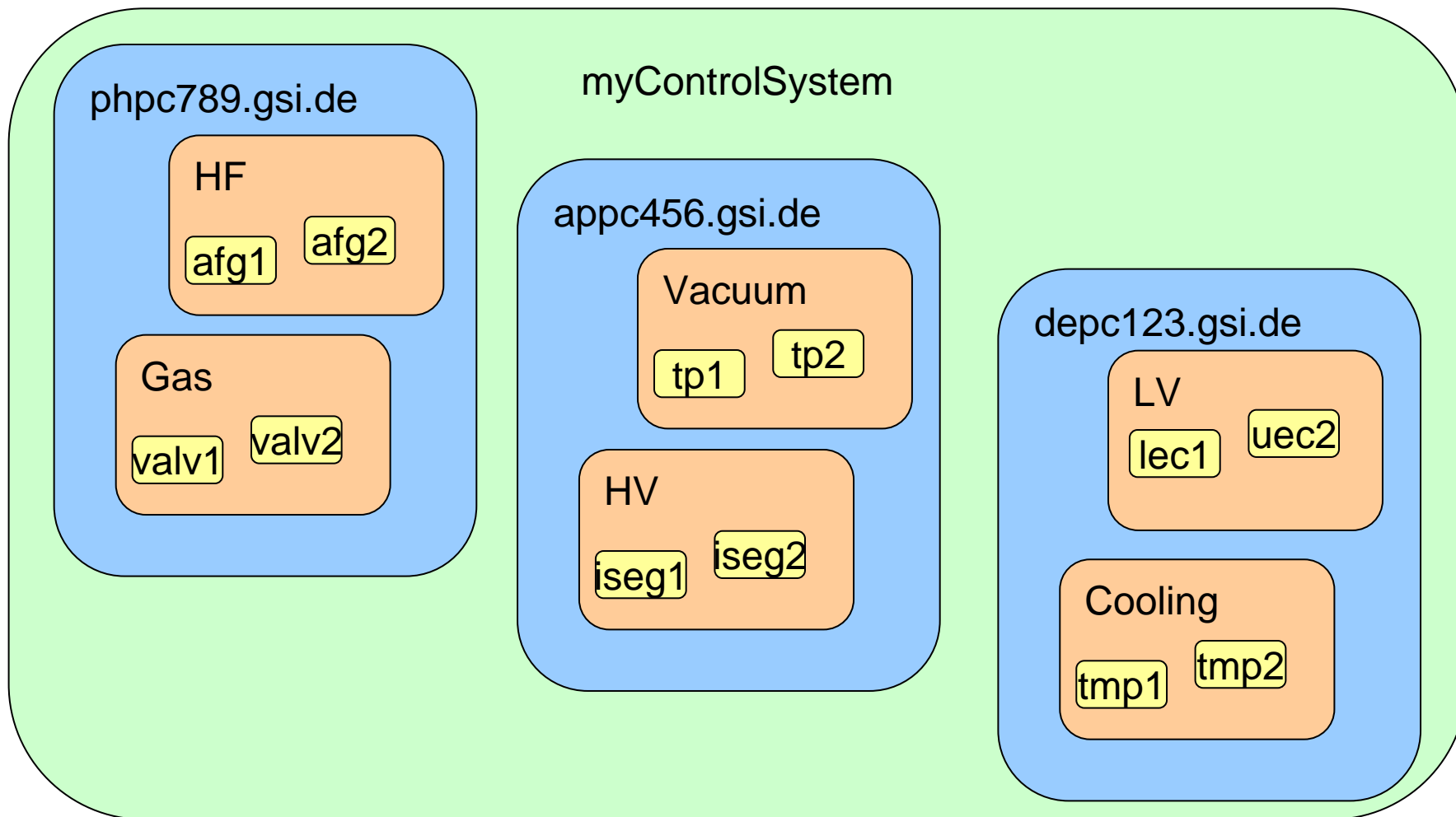
# Cooking recipe

- **One development tool** ⇒ LabVIEW
  - Fast learning curve
  - Multi-Threading
  - Synchronization via events, semaphores, ...
  - ...
- **Object oriented approach** ⇒ CS framework
  - Create objects (processes) on the fly
  - Create high level classes by inheriting from base classes
  - Encapsulate data and their treatment “information hiding”
  - ...
- **SCADA functionality** (alarming, trending, ...) ⇒ LabVIEW DSC module
- **Distributed system on many nodes** ⇒ DIM ([www.cern.ch/dim](http://www.cern.ch/dim))
  - Event driven communication for everything
  - Scalability
  - Remote access
  - ...

# Example for a simple control system



# Distribution



domain

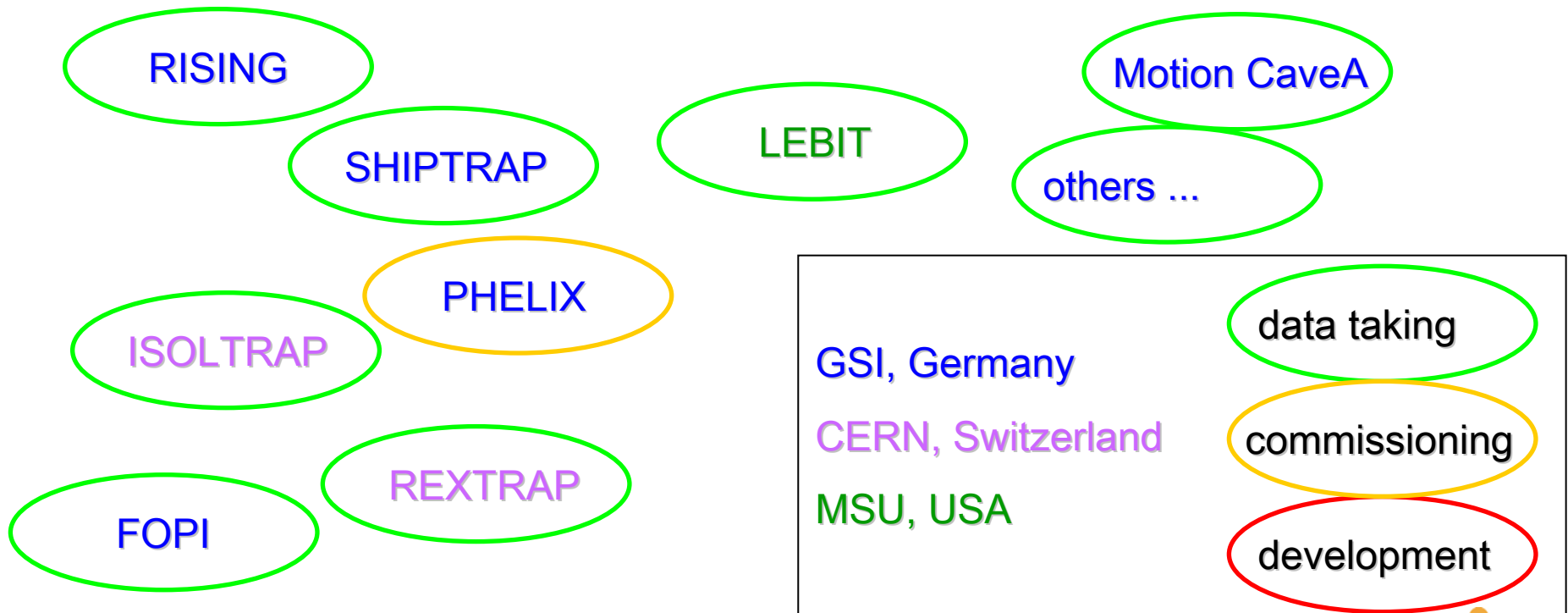
node

system

object

# Usage of the CS framework today

- experiments requiring high flexibility
- experiments with a large variety of hardware types
- experiments with up to 10,000 (1M possible) process variables





# (New) Features for CS 3.00

- Support for LabVIEW 7.1.
- DIM ([www.cern.ch/dim](http://www.cern.ch/dim)) as communication layer
- real publisher-subscriber mechanism
- CSAccessSystem: locking access to (groups of) objects
- DomainManagementSystem: distributed process management (Windows, Linux)
- Source code control via SubVersion
- <http://wiki.gsi.de/cgi-bin/view/CSframework/WebHome>
  - documentation
  - HOW-TOs
  - FAQs
  - ...
- <https://sourceforge.net/projects/cs-framework/>
  - downloads
  - bug reports
  - feature requests
  - ...

# HITRAP Control System

- CS as basis?
- Attempt for project management: <http://wiki.gsi.de/cgi-bin/view/CSframework/HITRAPControlSystem>