

Control Systems

Alessandro Stecchi & Giampiero Di Pirro

INFN-LNF • Accelerator Division





Video courtesy of SpaceX



SPEED
445
KM/H

ALTITUDE
0.4
KM

STAGE 1 TELEMETRY

ENTRY

LANDING

SECO-1

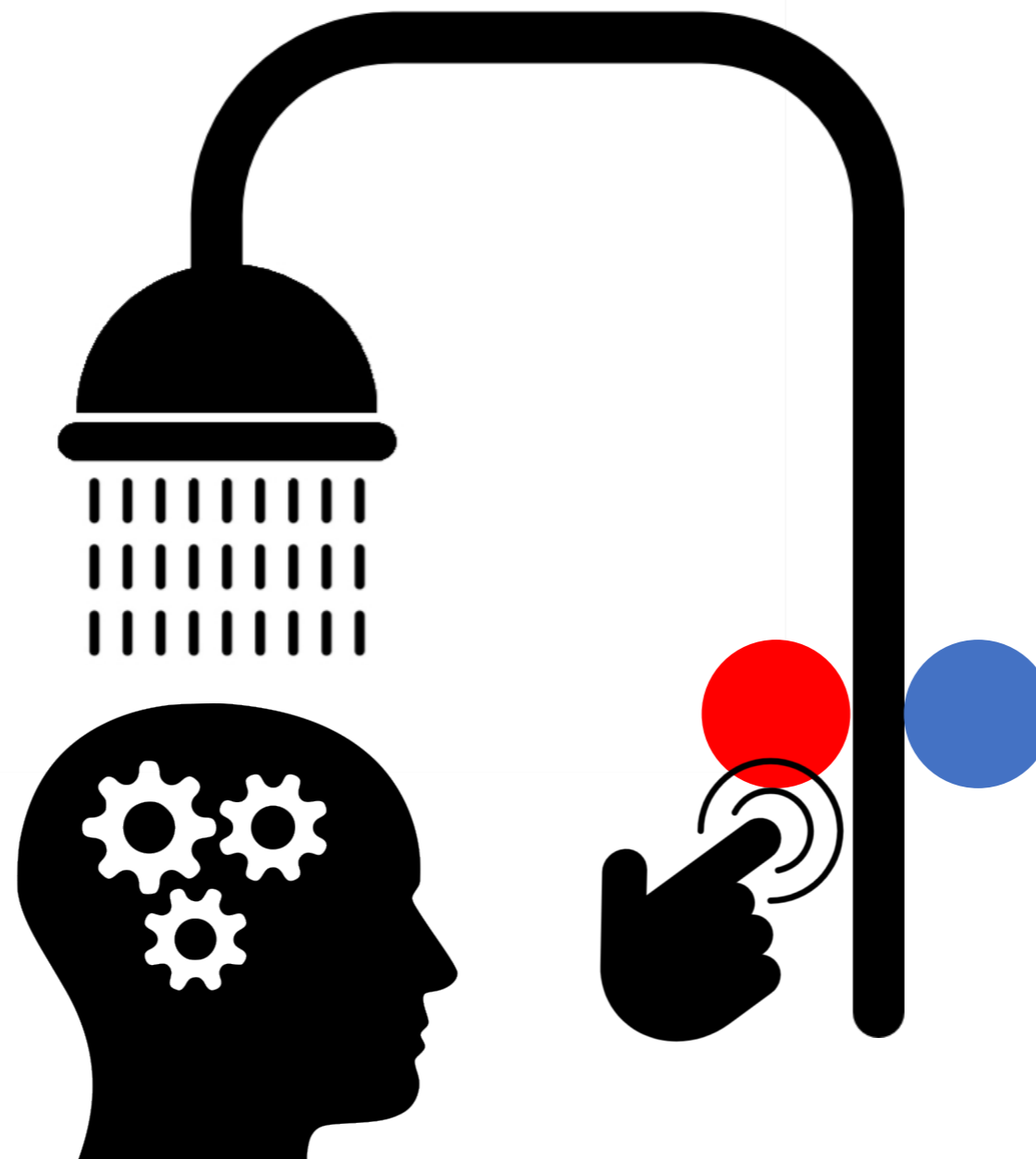
T+ 00:08:30
STARLINK

SPEED
23908
KM/H

ALTITUDE
211
KM

STAGE 2 TELEMETRY

A Control System is a system that allows the **output variables** of a **process** to be varied (or kept constant) according to **predetermined laws**.



Open loop

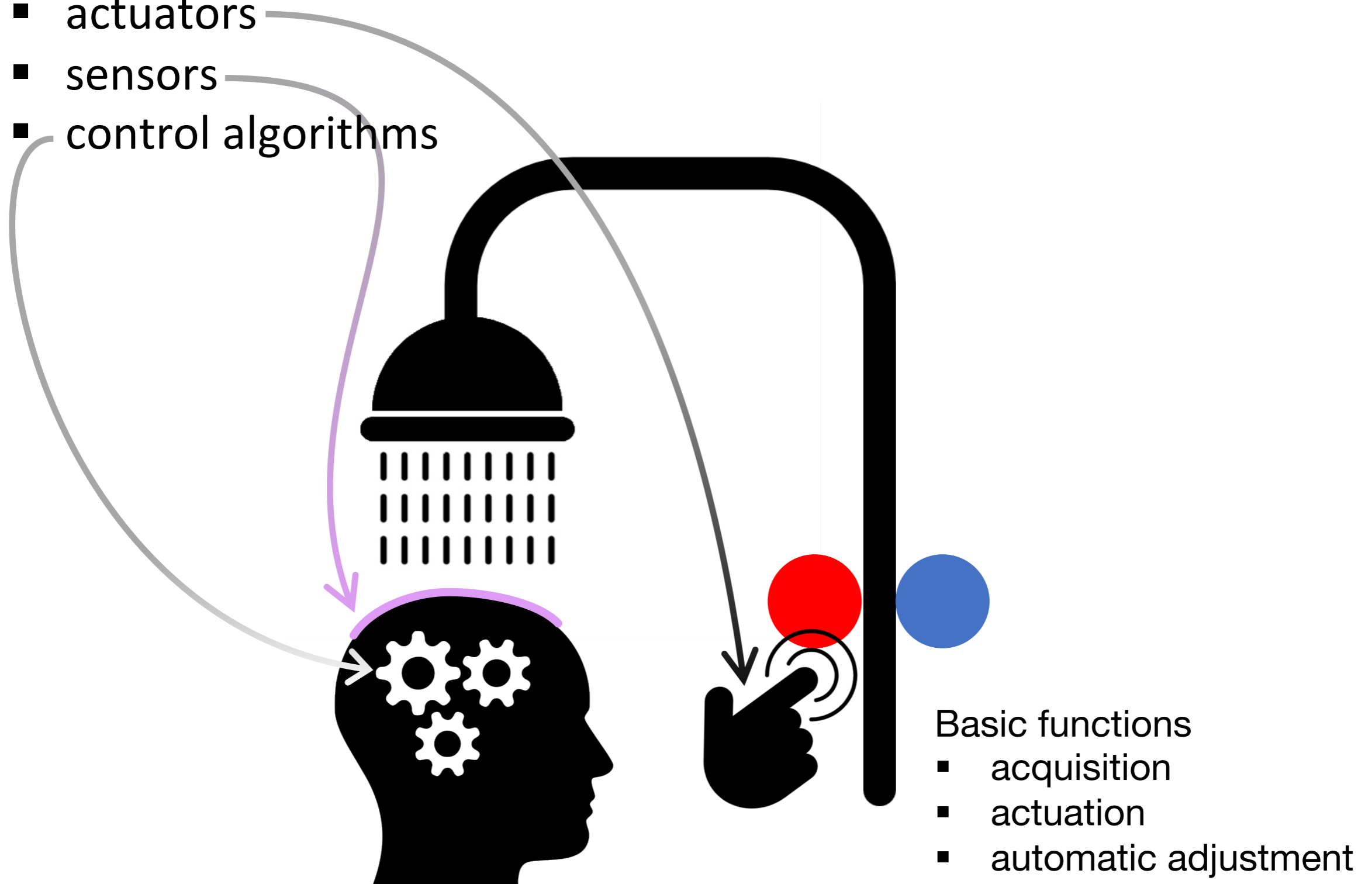
Action is based on an algorithm and initial **static information**.

Closed loop

Action is based on an algorithm and **dynamic information** continuously read back from the controlled system.

Key elements of a Control System:

- actuators
- sensors
- control algorithms



Fields of application of control systems

Civil, Social

- home automation
- automotive, aeronautics
- building climate control (ESCO: Energy Service Company)
- agriculture
- medical equipment, radiotherapy, diagnostics
- telecommunications

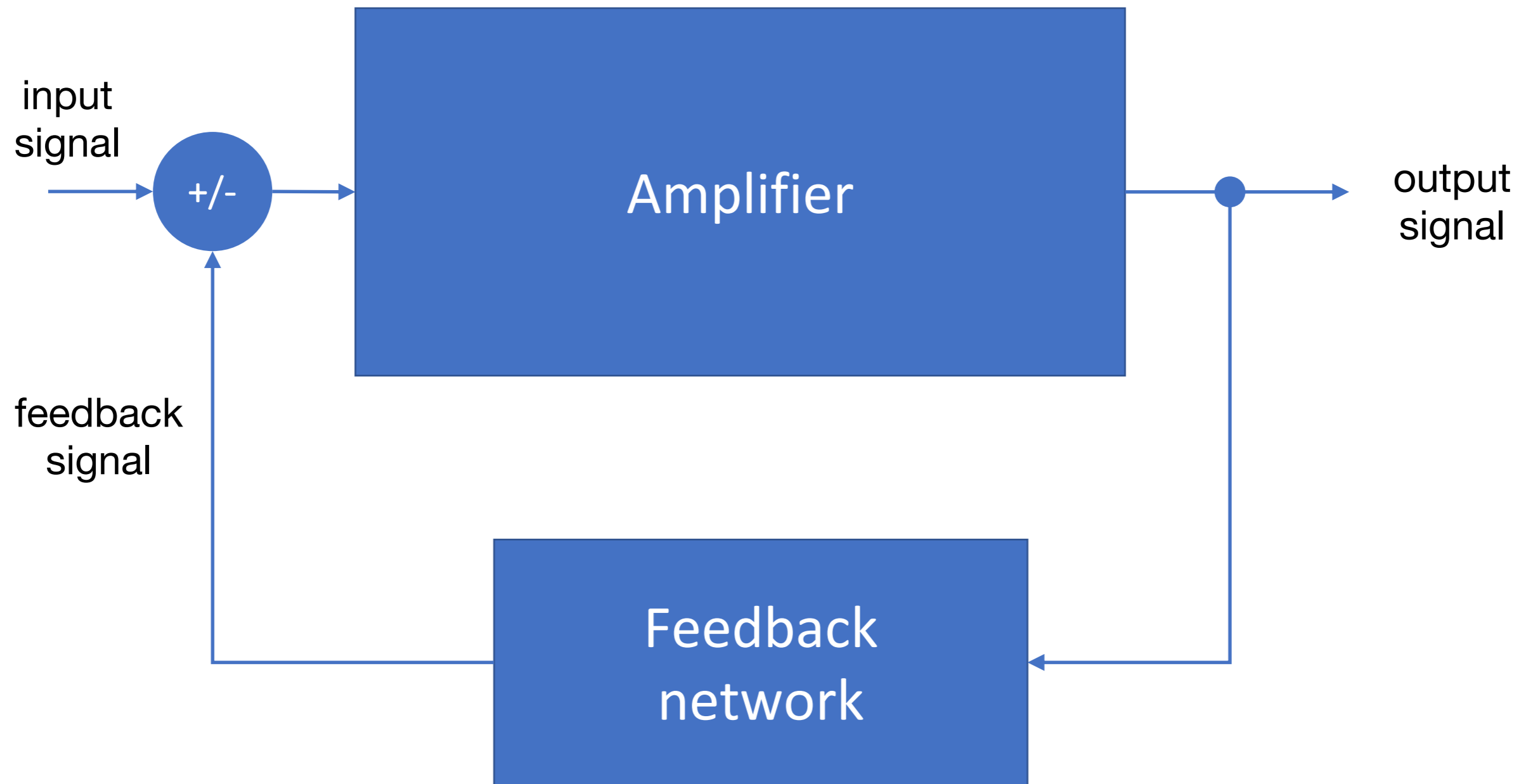
Industrial

- production lines (robots)
- prognostics
- power plants (thermal, electrical, nuclear)

Scientific

- unmanaged rovers, space probes
- astronomy
- experimental apparatus, detectors, particle accelerators

What physicists and engineers usually mean by *Control System*



Controls adopting "CPUs"

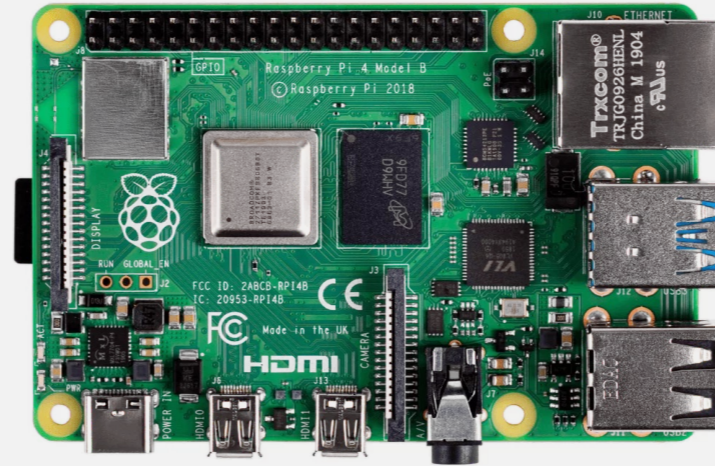


Computers $t \sim 1-10$ ms

PC



SBC: Single Board Computer



Embedded processor



FPGA: Field Programmable Gate Array
 $t \sim$ ns



Virtual Machines



uP: microprocessor
 $t \sim 1$ ms



ASIC: Application-Specific Integrated Circuit

Fields of application of our control systems

Civil, Social

- home automation
- automotive, aeronautics
- building climate control (ESCO: Energy Service Company)
- agriculture
- medical equipment, radiotherapy, diagnostics
- telecommunications

Industrial

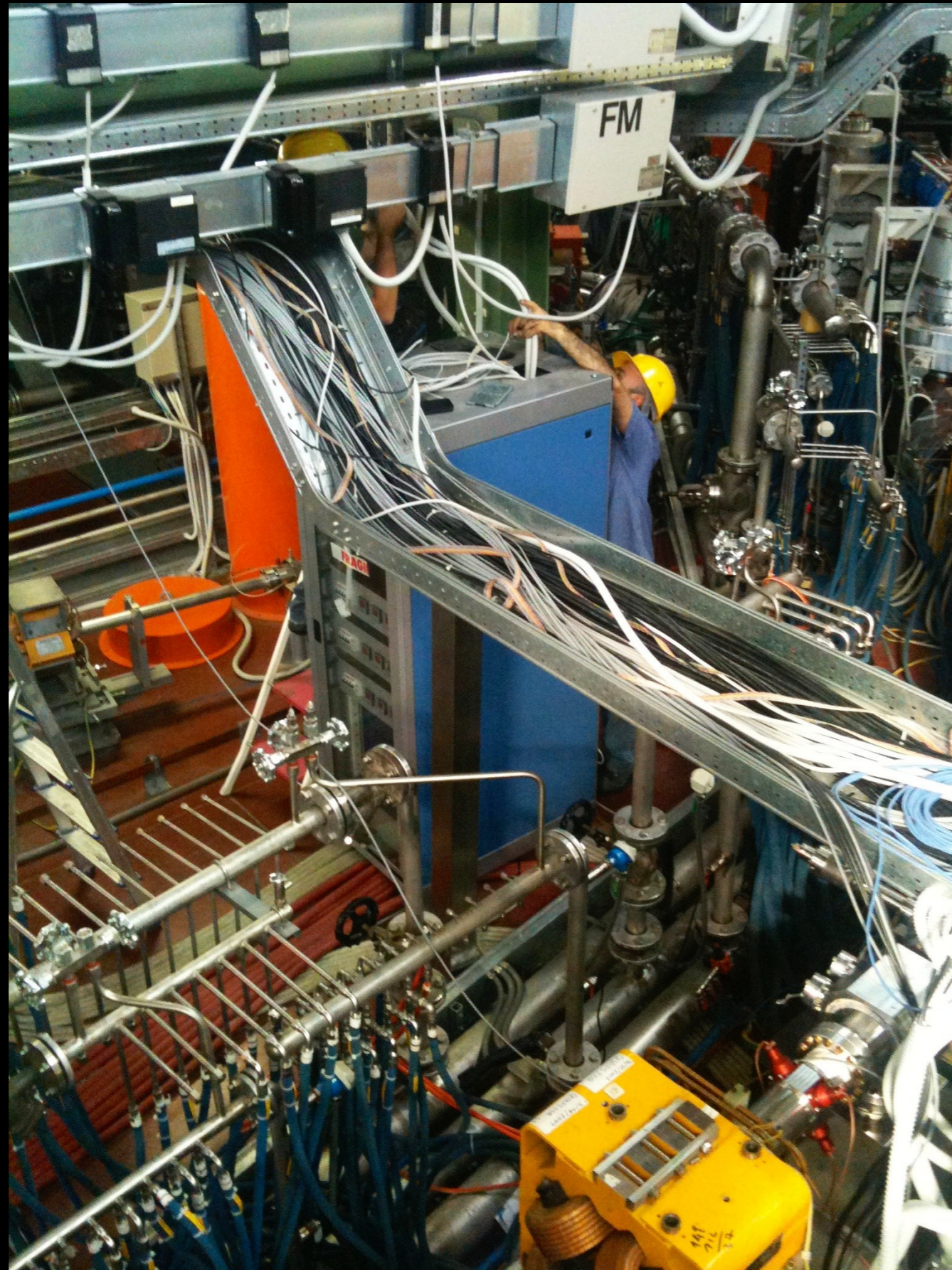
- production lines (robots)
- prognostics
- thermal, electrical, nuclear power plants

Scientific

- automatic rovers, space probes
- astronomy
- **experimental apparatus, detectors, particle accelerators**

Control Systems – of complex plants – employing computers (*and IT in general*)

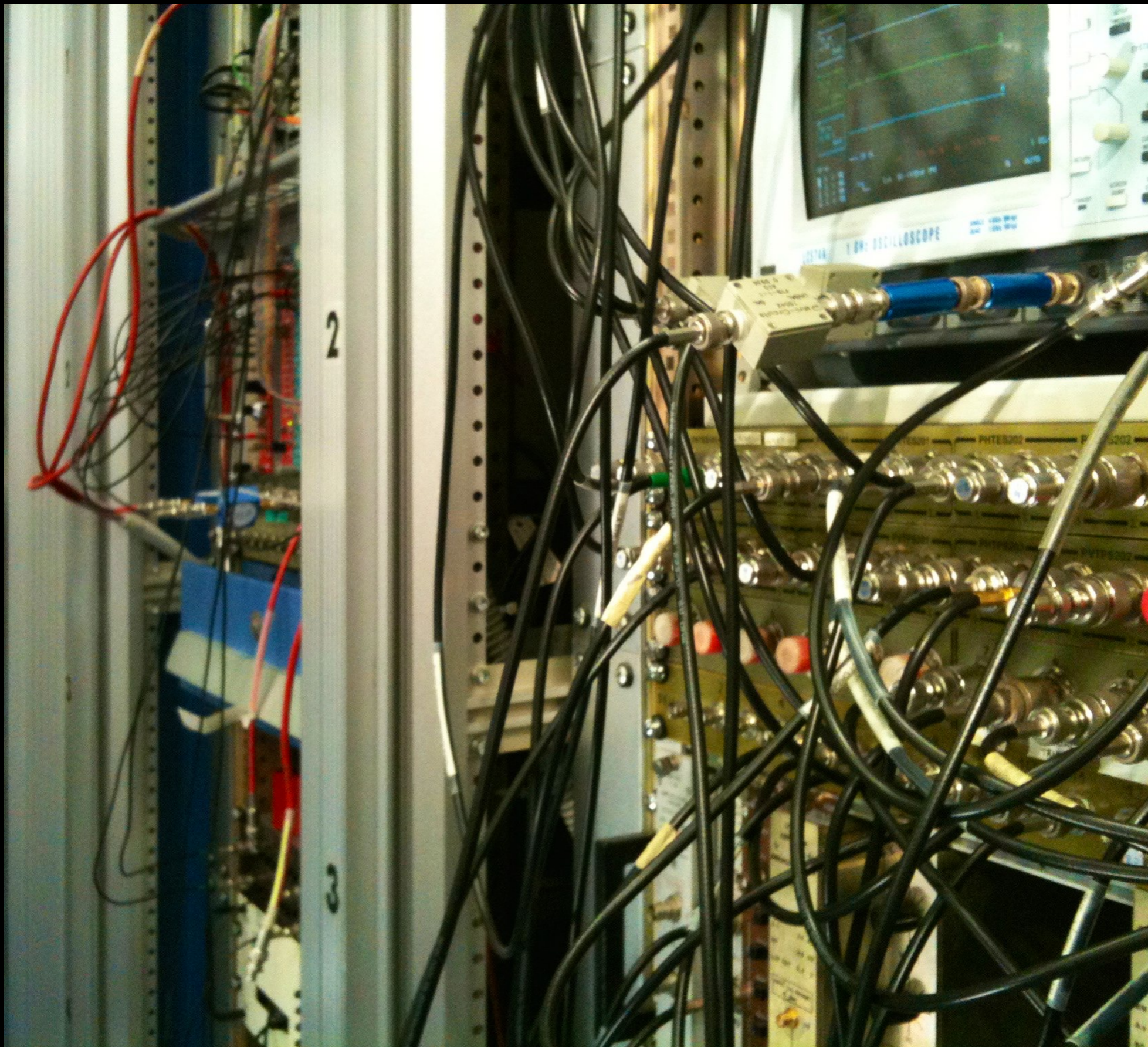
plant



plant → sensors/actuators



plant → sensors/actuators → **front-end**



plant → sensors/actuators → front-end → **control room**



Design and Implementation

Definition of specifications and objectives to achieve

- performance
- features
- reliability
- assessment of complexity and load estimation (processing power, throughput, latency, storage)

System modeling

- design of the control framework or adoption of an existing one (e.g. EPICS, TANGO,...)

Development

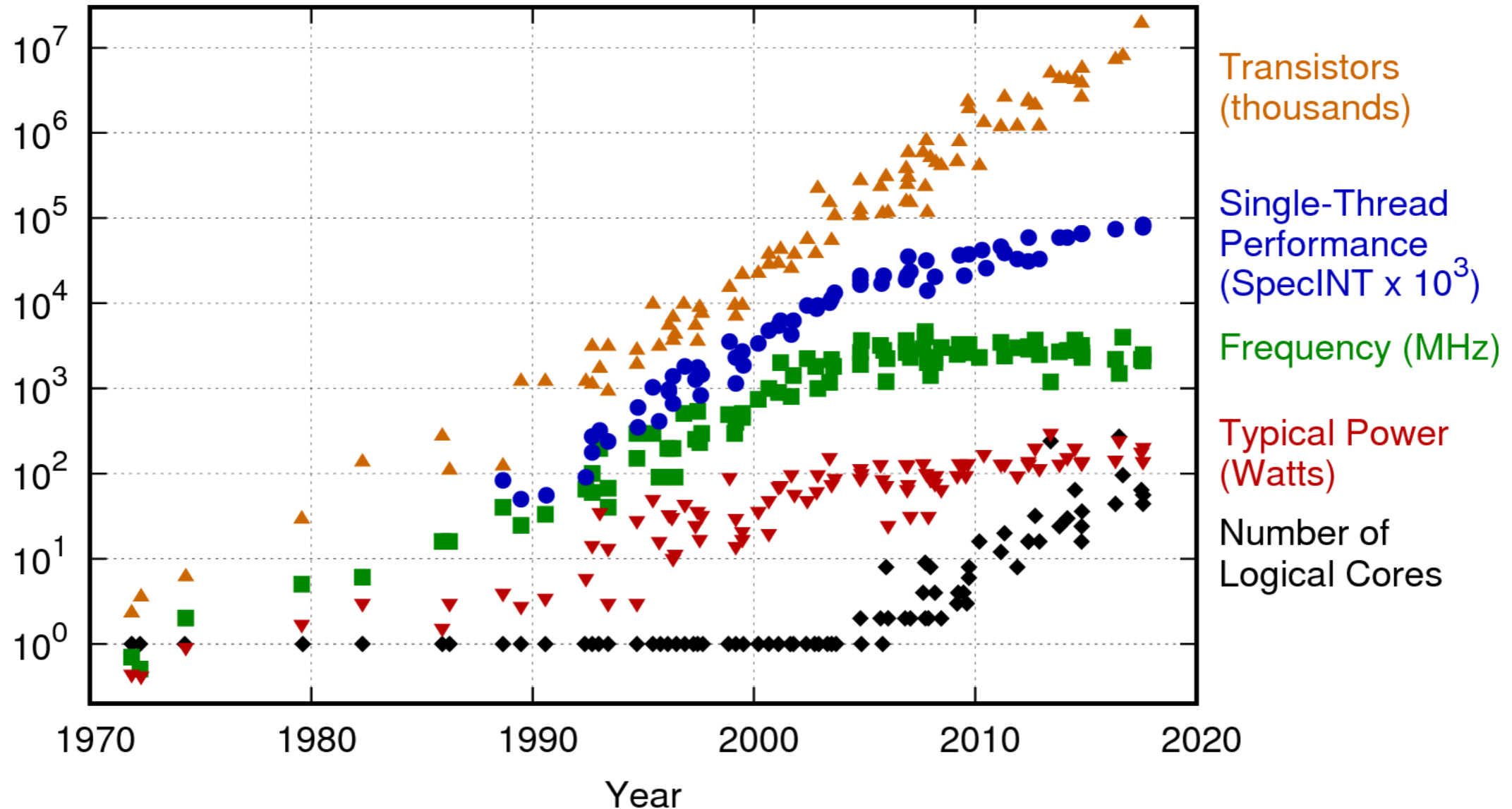
- OOD, OOP (properties, methods)
- software development
- simulation
- engineering
- validation

General characteristics

- **open** to other systems (experiments, DAQ, machine protection, service plants);
- **easy** for the users (ergonomic UI, data import/export, custom software interfacing);
- **scalable** according to plant size and required performance;
- **reliable** (no SPF, redundancy);
- **flexible** with respect to technological updates.

Technological growth

42 Years of Microprocessor Trend Data



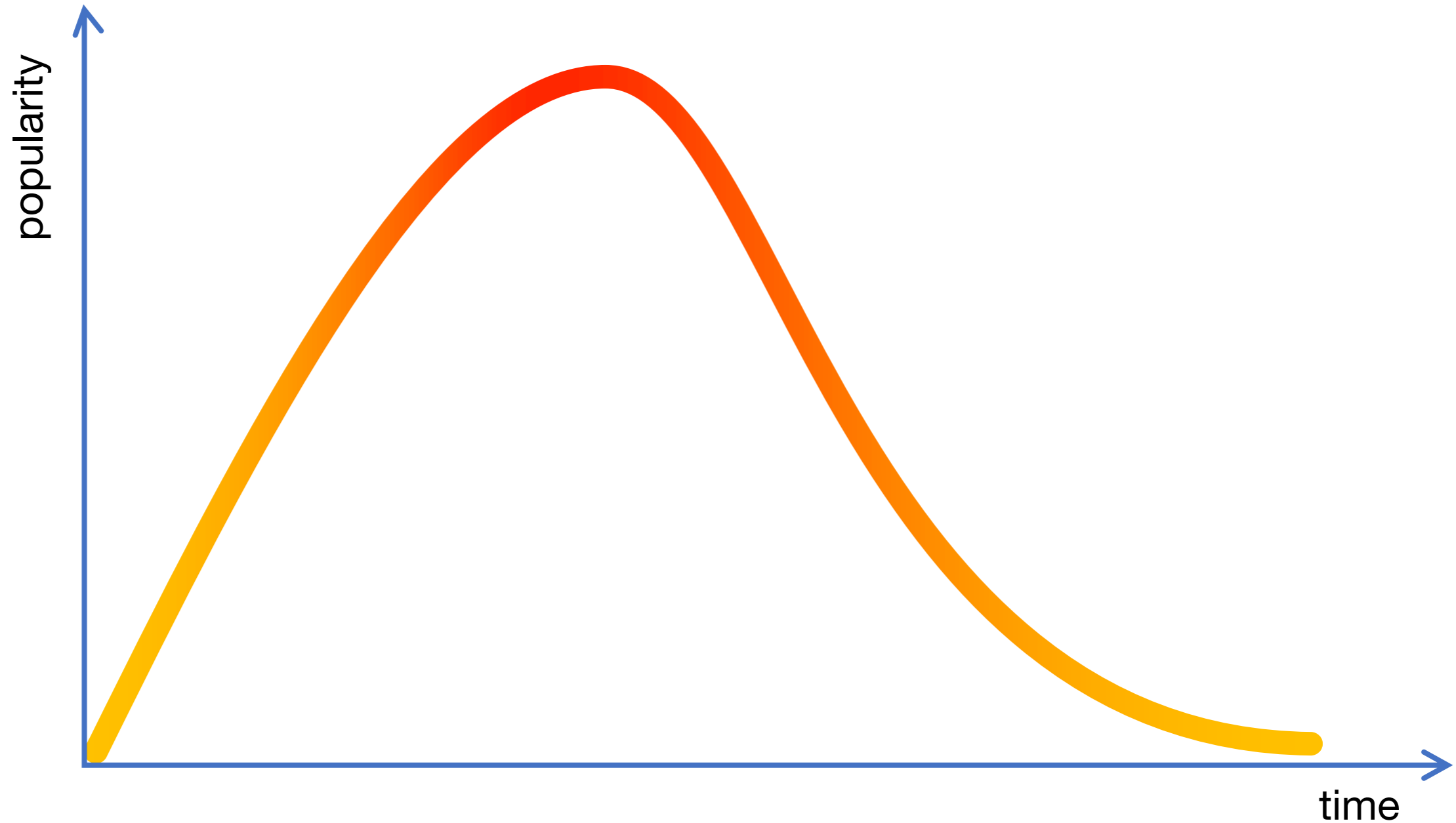
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Moore's law

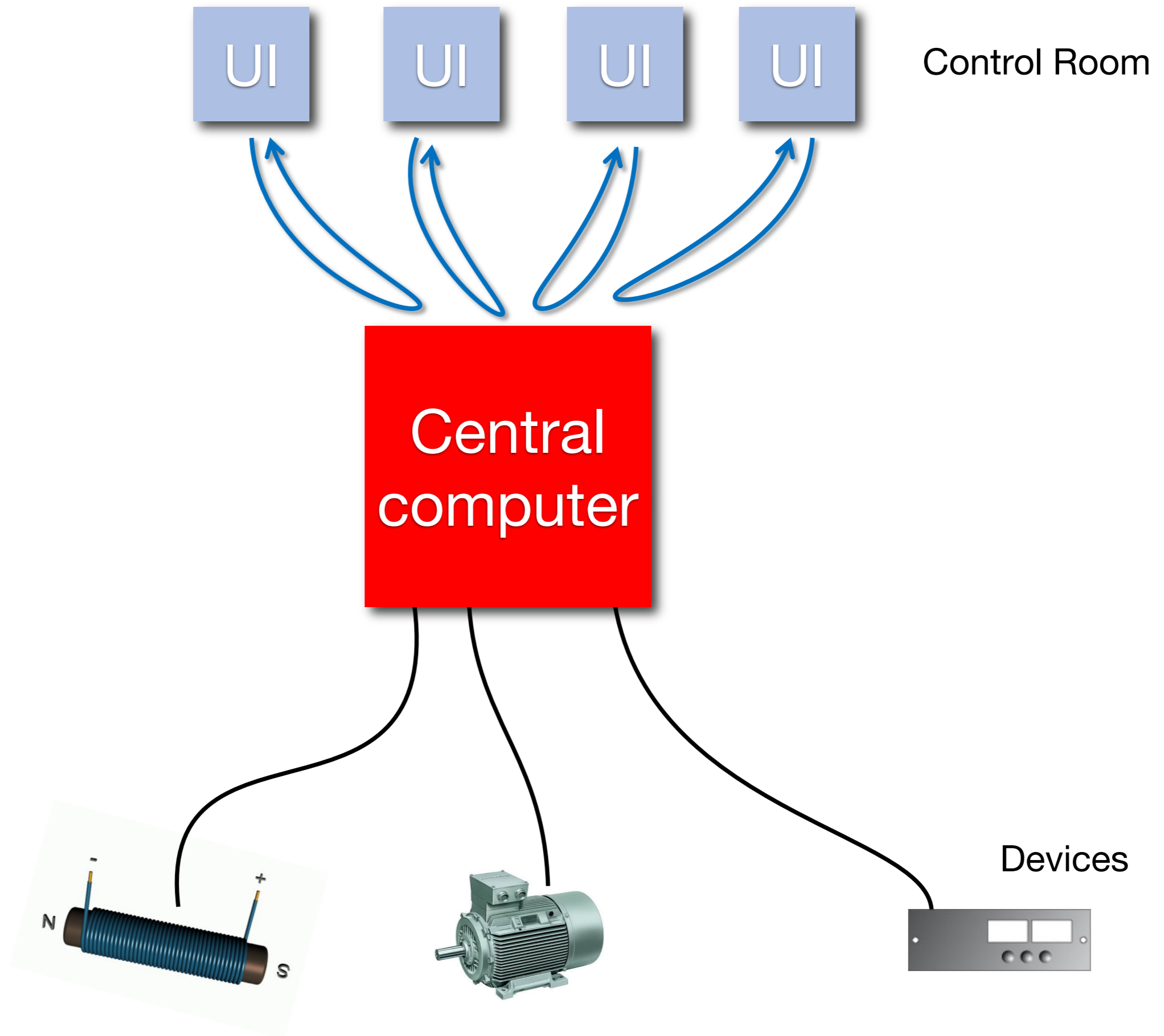
the number of transistors in processors doubles every 18 months

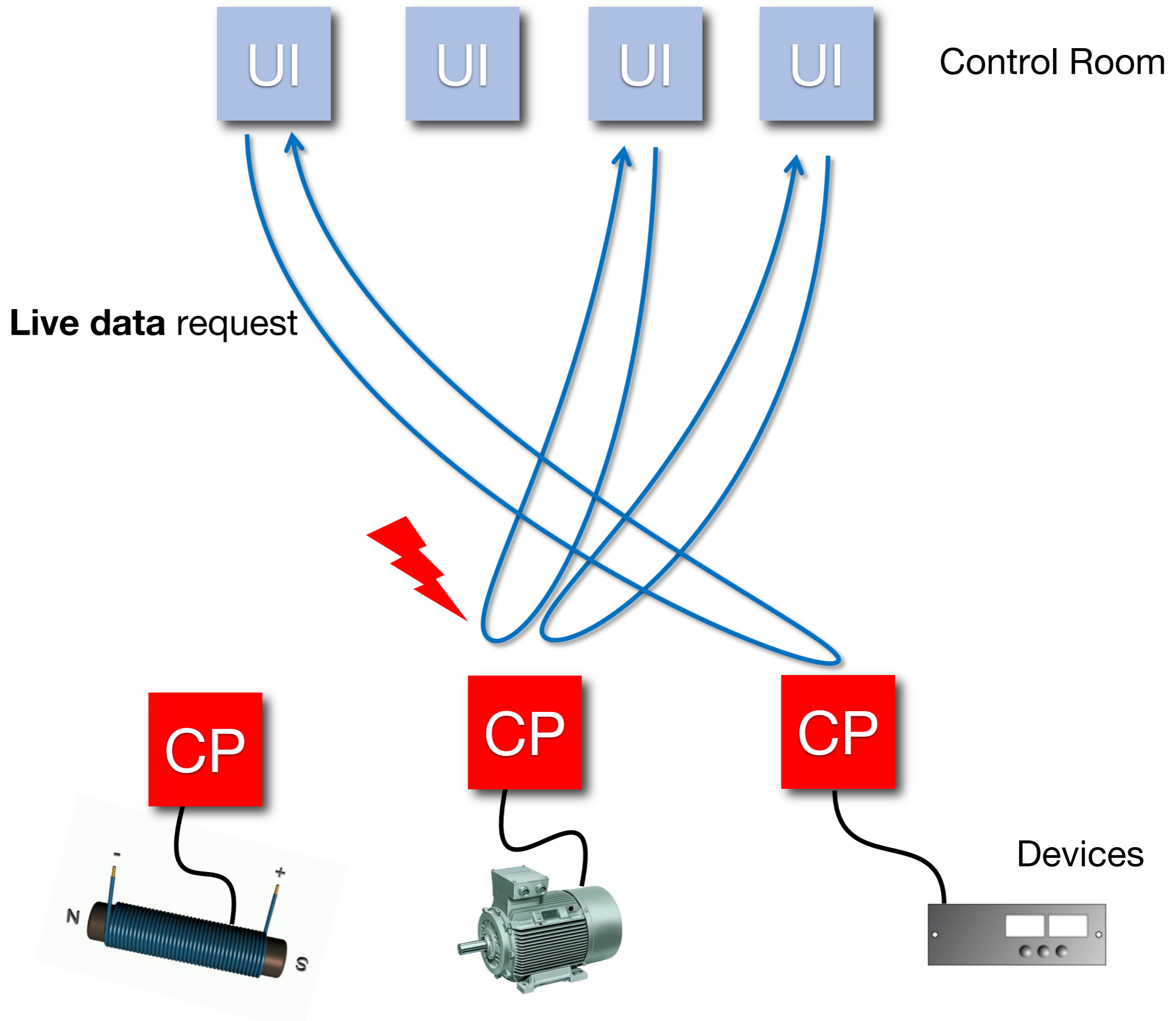
-> limits on reducing the size of transistors below which unwanted 'parasitic' quantum effects would occur in electronic circuits.

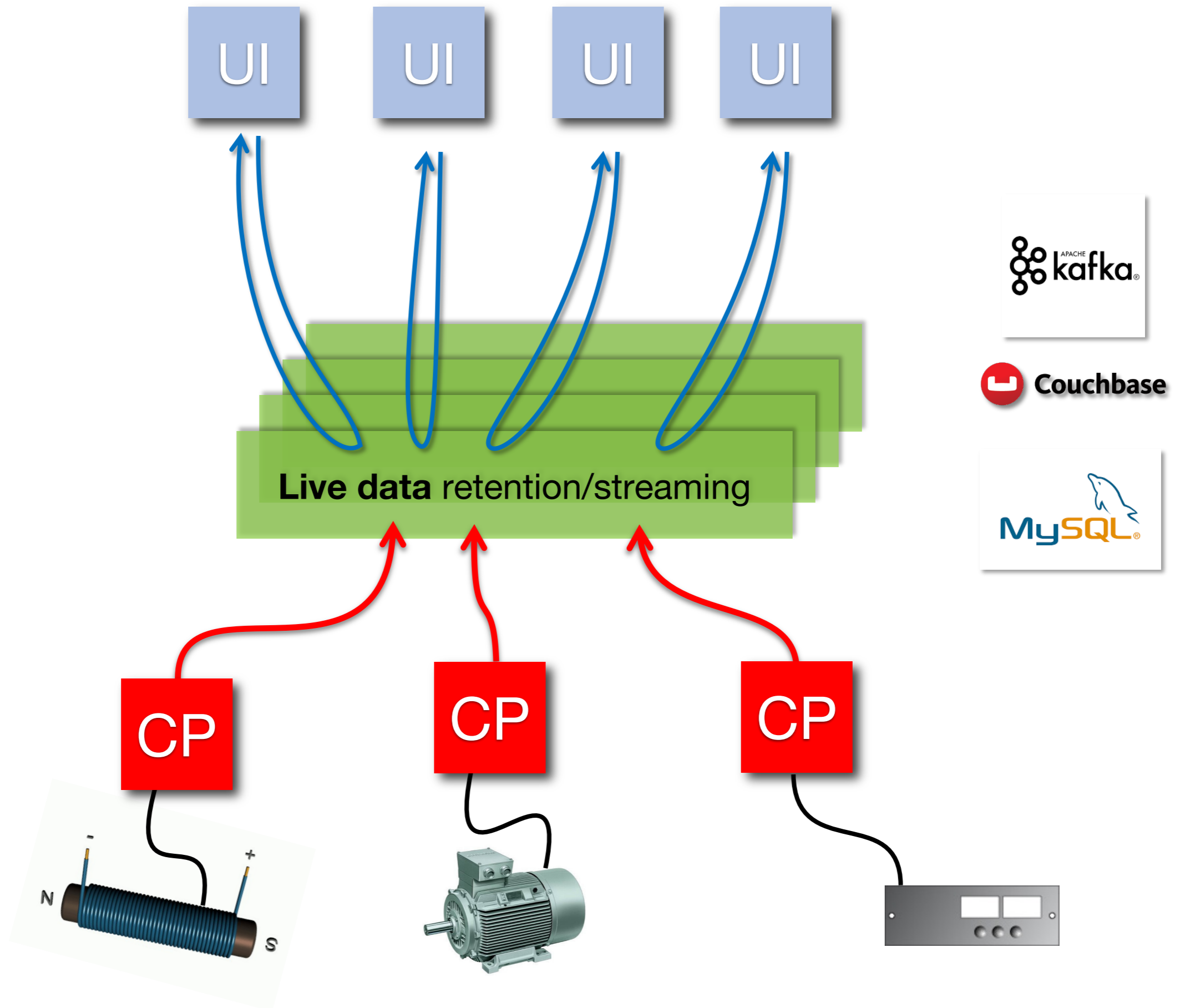
Obsolescence of standards

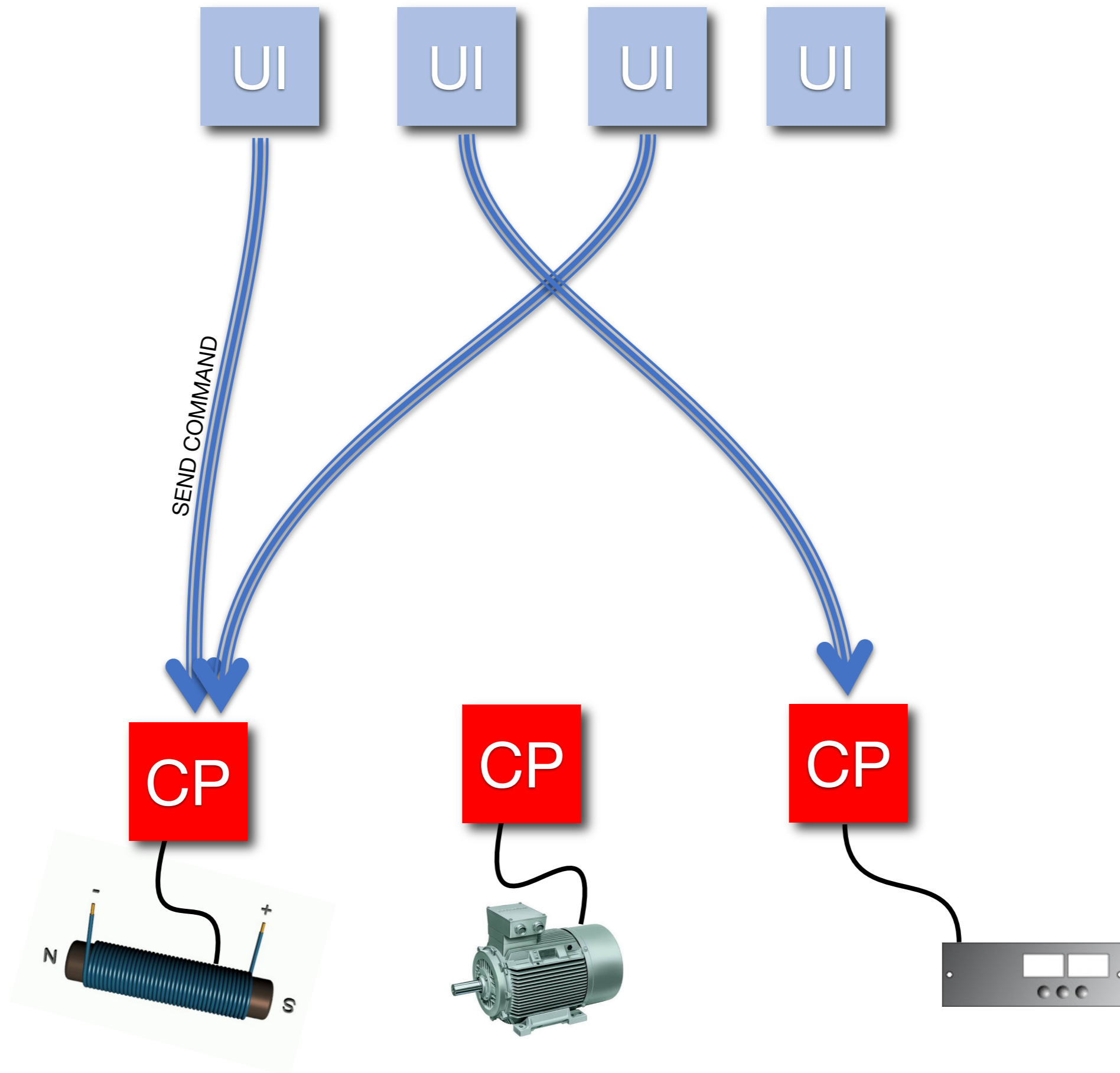


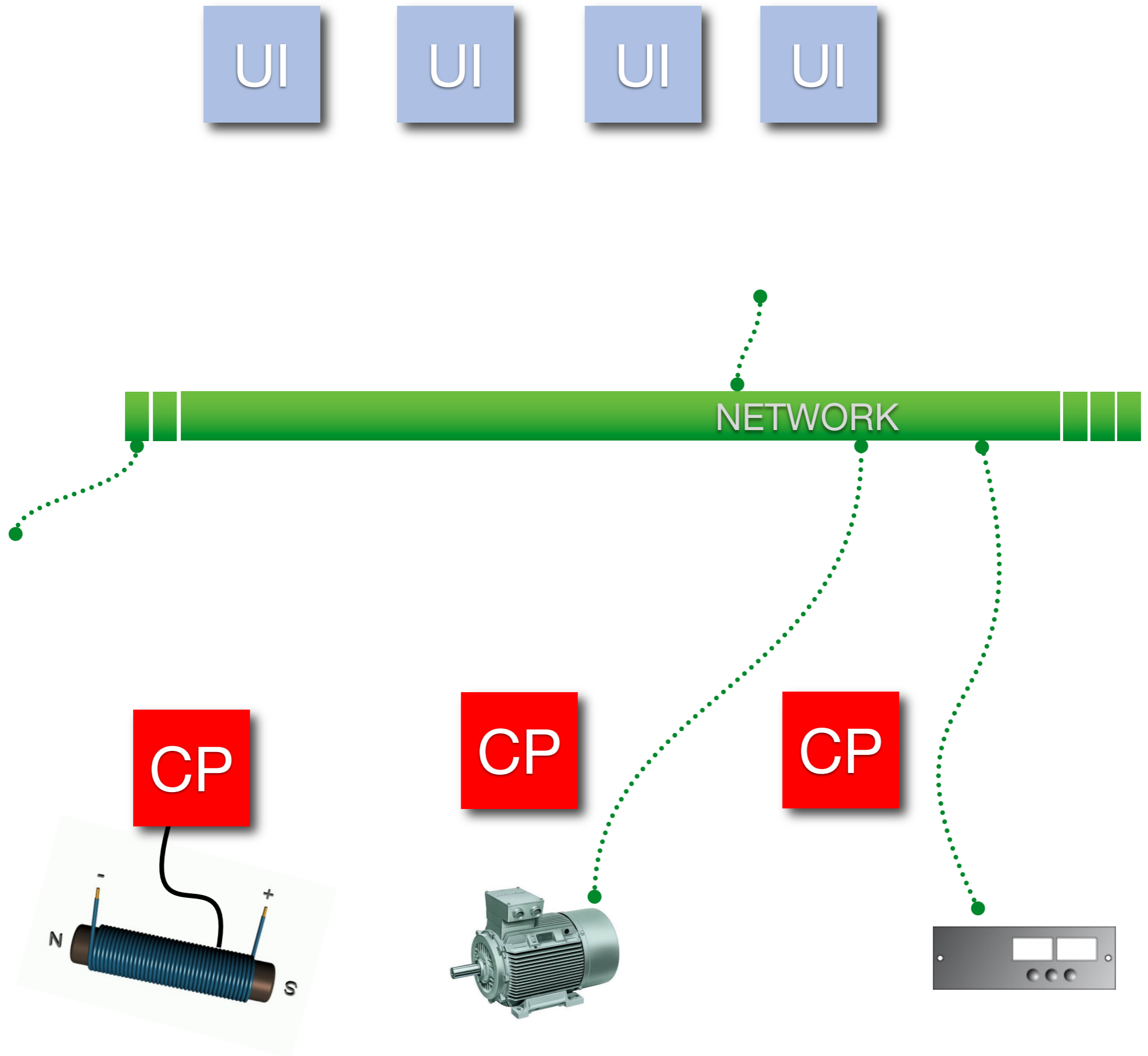
The lesson is: a system has to be as **independent** as possible from hardware standards and specific software products.





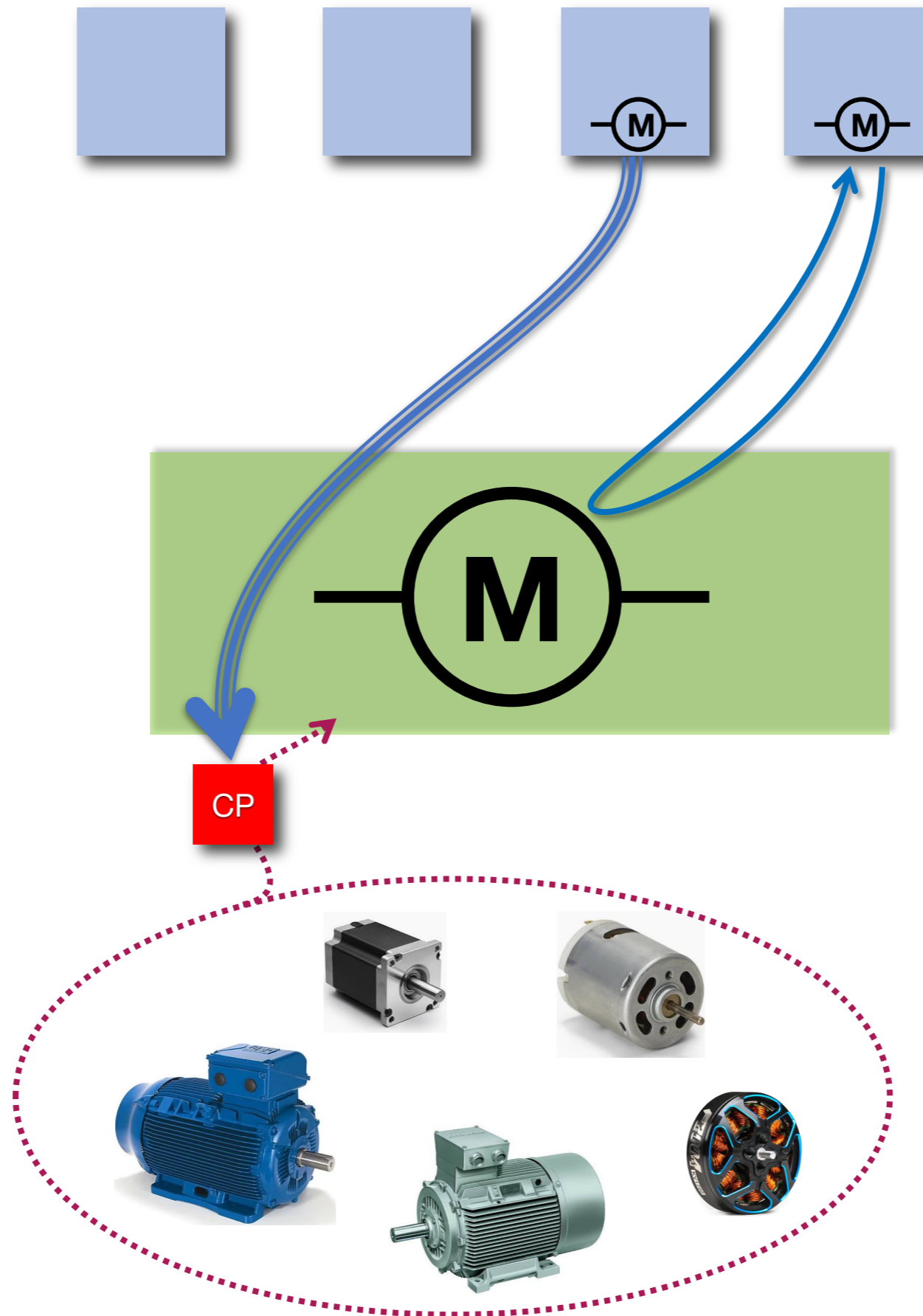




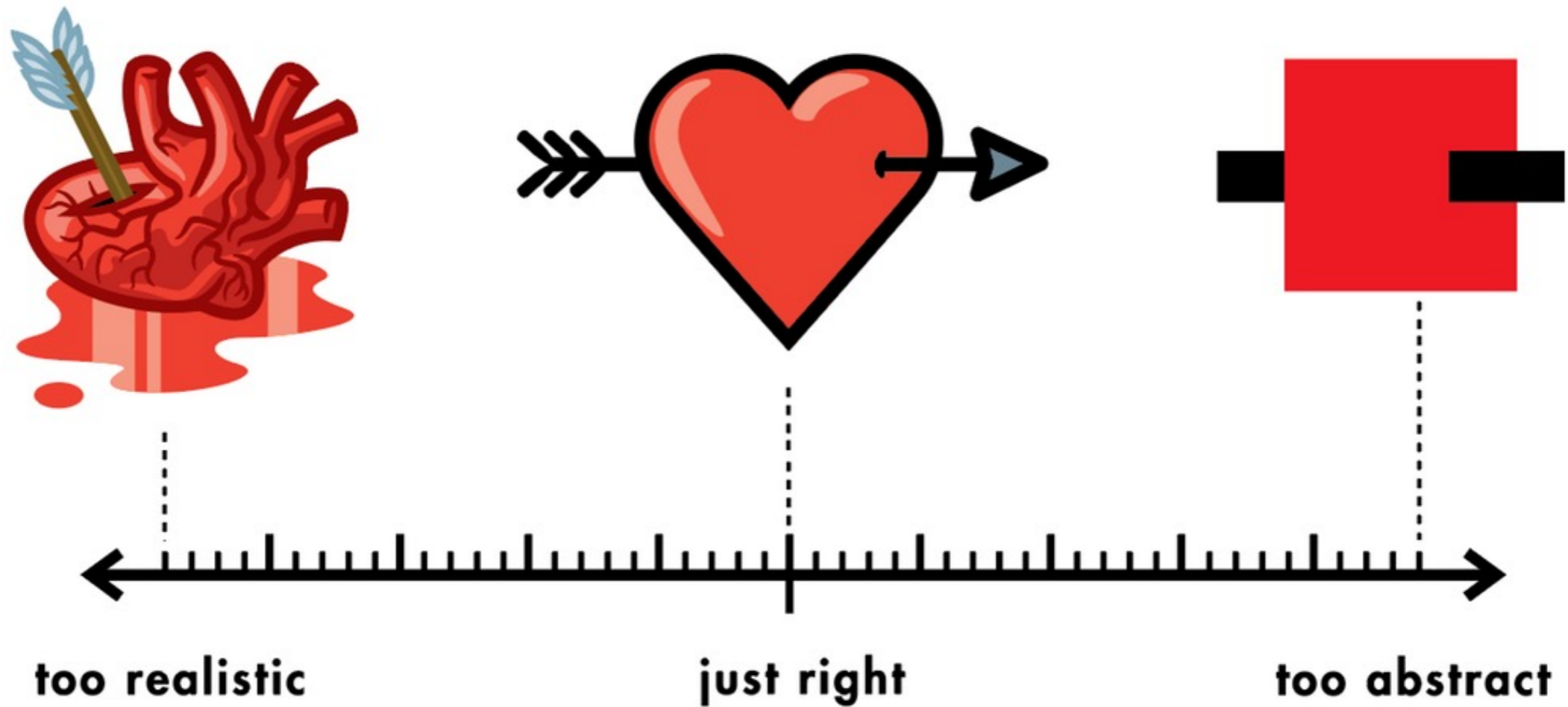


Abstraction





THE ABSTRACT-O-METER

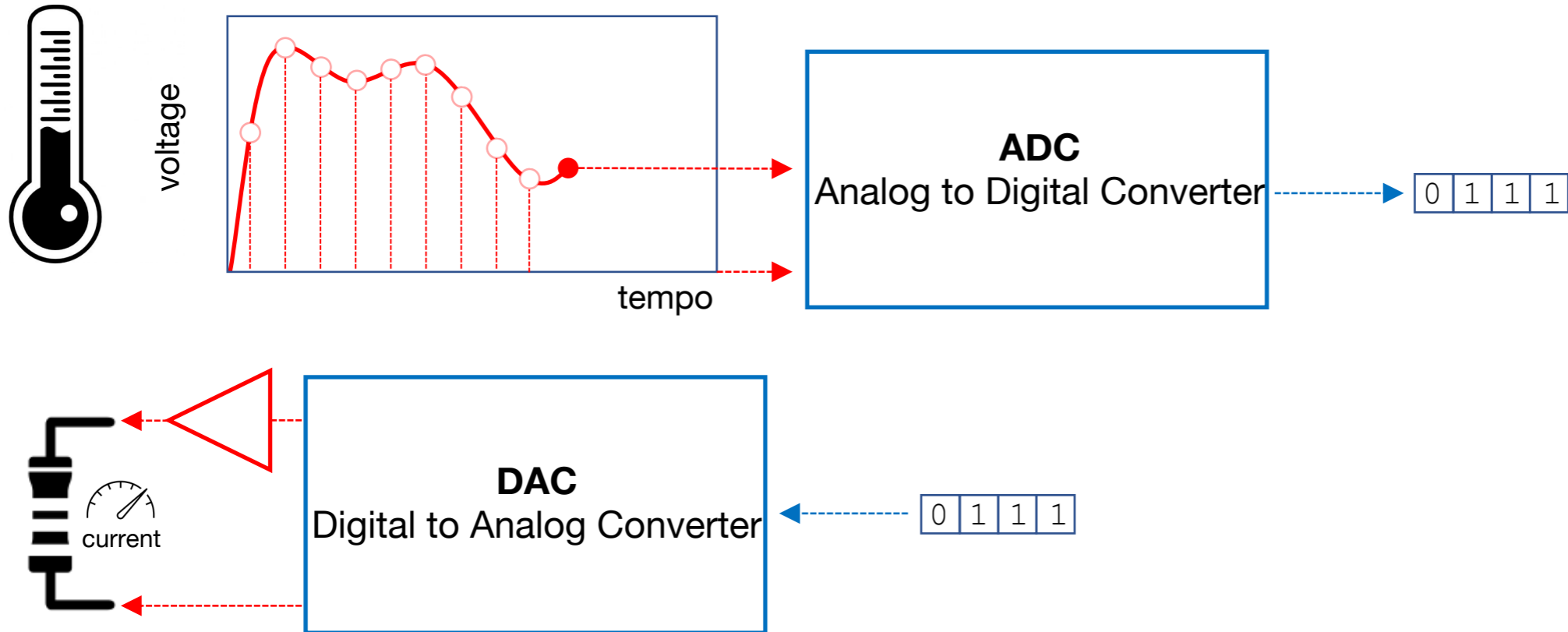


Features of a Control System

- acquisition/actuation
- communication
- storage
- presentation
- analysis / correlation
- historization

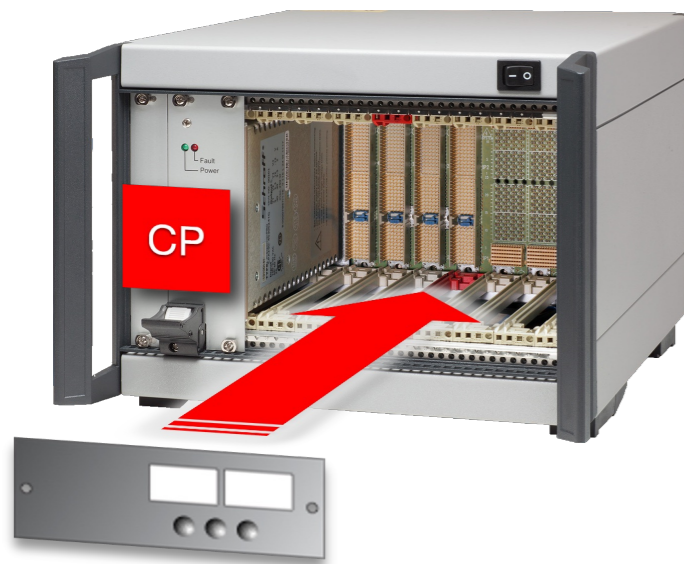
- **acquisition/actuation**

- communication
- storage
- presentation
- analysis / correlation
- historization



- acquisition/actuation
- **communication**
- storage
- presentation
- analysis / correlation
- historization

Embedded devices



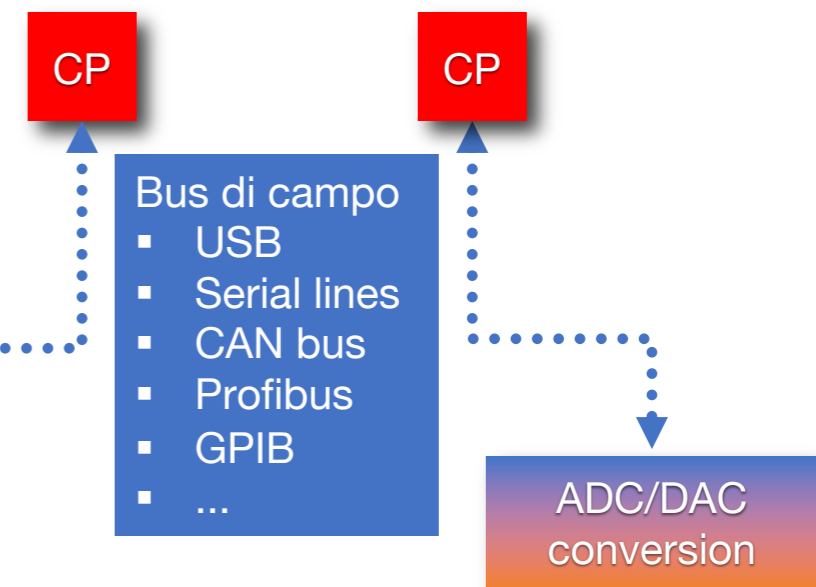
Intelligent devices



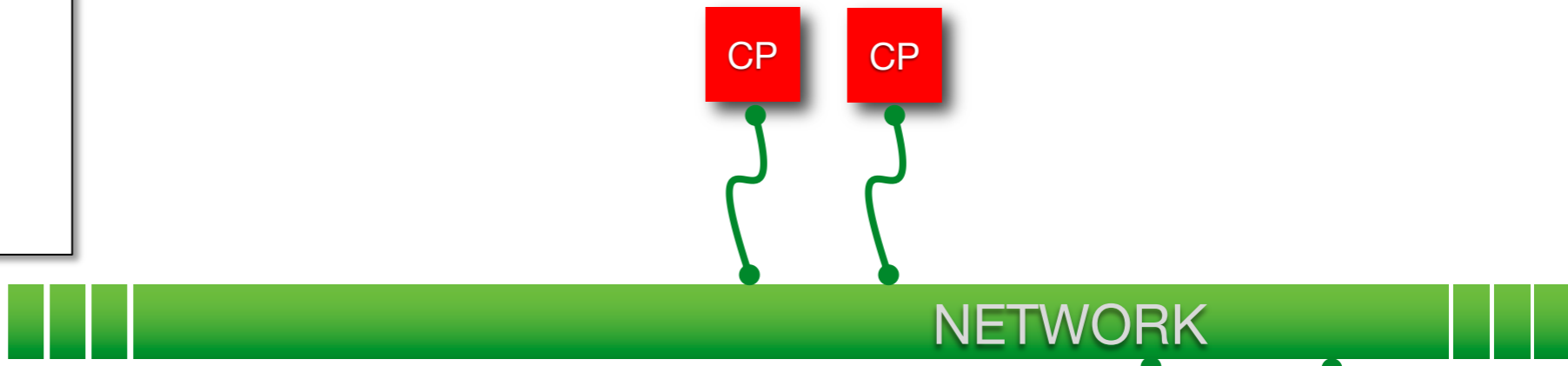
Digital devices



Analog devices



- acquisition/actuation
- **communication**
- storage
- presentation
- analysis / correlation
- historization



Embedded devices



Intelligent devices



Digital devices



Analog devices

- acquisition/actuation
- communication
- **storage**
- presentation
- analysis / correlation
- historization

average size (@DAFNE)
~ 650 byte

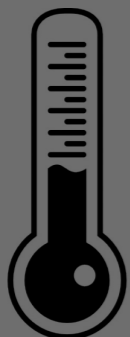
```
{
float64 temperature_read;
char    units;
boolean status;
....
}
```

```
{
float64 current_read;
float64 current_set;
char    units;
boolean error;
....
}
```

```
{
float64 mag_field_read;
float64 mag_field_set;
int     polarity;
....
}
```

```
{
int     aperture;
int     time;
int     n_pixels_h;
int     n_pixels_v;
boolean status;
....
}
```

CP



CP



CP



CP



650 byte

Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura,
ché la diritta via era smarrita.

Ahi quanto a dir qual era è cosa dura
esta selva selvaggia e aspra e forte
che nel pensier rinova la paura!

Tant'è amara che poco è più morte;
ma per trattar del ben ch'i' vi trovai,
dirò de l'altre cose ch'i' v'ho scorte.

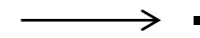
Io non so ben ridir com'i' v'intrai,
tant'era pien di sonno a quel punto
che la verace via abbandonai.

Ma poi ch'i' fui al piè d'un colle giunto,
là dove terminava quella valle
che m'avea di paura il cor compunto

guardai in alto e vidi le sue spalle
vestite già de' raggi del pianeta
che mena dritto altrui per ogni calle.



Dante Alighieri (1265–1321)



The Divin Comedy
~0.5 Mbyte

DAFNE ~1.4 Mbyte

Relational database →



```
{  
float64 temperature_read;  
char    units;  
boolean status;  
.....  
}
```



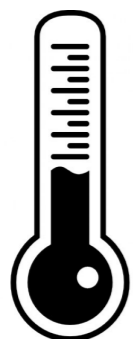
```
{  
float64 current_read;  
float64 current_set;  
char    units;  
boolean error;  
.....  
}
```



```
{  
float64 mag_field_read;  
float64 mag_field_set;  
int     polarity;  
.....  
}
```



```
{  
int     aperture;  
int     time;  
int     n_pixels_h;  
int     n_pixels_v;  
boolean status;  
.....  
}
```



Relational database →



```
{  
float64 temperature_read;  
char    units;  
boolean status;  
.....  
}
```



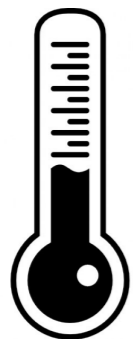
```
{  
float64 current_read;  
float64 current_set;  
char    units;  
boolean error;  
.....  
}
```



```
{  
float64 mag_field_read;  
float64 mag_field_set;  
int     polarity;  
.....  
}
```



```
{  
int     aperture;  
int     time;  
int     n_pixels_h;  
int     n_pixels_v;  
boolean status;  
.....  
}
```



Relational database →

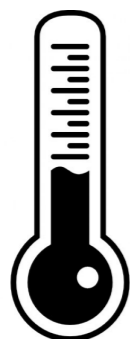


```
{  
float64 temperature_read;  
char    units;  
boolean status;  
.....  
}
```

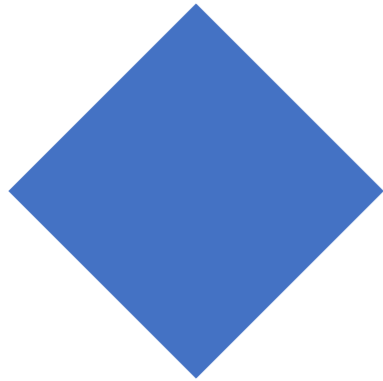
```
{  
float64 current_read;  
float64 current_set;  
char    units;  
boolean error;  
.....  
}
```

```
{  
float64 mag_field_read;  
float64 mag_field_set;  
float64 orientation;  
int.    polarity;  
}...  
}
```

```
{  
int    aperture;  
int    time;  
int    n_pixels_h;  
int    n_pixels_v;  
boolean status;  
.....  
}
```



Serialization: *self-describing* data vs *formatted* data



```
BEGIN OF DATA  
  here it is described a rhombus;  
  each side measures 100 mm;  
  the orientation of the diagonal is 0 deg;  
  blah blah...;  
END OF DATA
```

Most popular *self-describing* data formats

XML

```
<note>
  <to>Andrew</to>
  <from>Jane</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

YAML

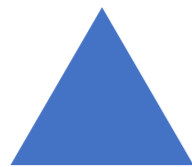
```
note:
  to: Andrew
  from: Jane
  heading: Reminder
  body: "Don't forget me this weekend!"
```

JSON

```
{
  "note":{
    "to":"Andrew",
    "from":"Jane",
    "heading":"Reminder",
    "body":"Don't forget me this weekend!"
  }
}
```


Non relational database

Key	Value



my_data

`0x16\0x00\0x00\0x00\0x02hello\0x00\0x06\0x00\0x00\0x00world\0x00\0x00`



my_other_data

`.... another descriptive sequence`

e.g. BSON serialization

`\0x16\0x00\0x00\0x00
\0x02
hello\0x00
\0x06\0x00\0x00\0x00world\0x00
\0x00`

total document size
0x02 = field data type is string
field name
field value (size of value, value, terminator)
type EOO ('end of object')

- acquisition/actuation
- communication
- storage
- **presentation**
- analysis / correlation
- historization

User Interface

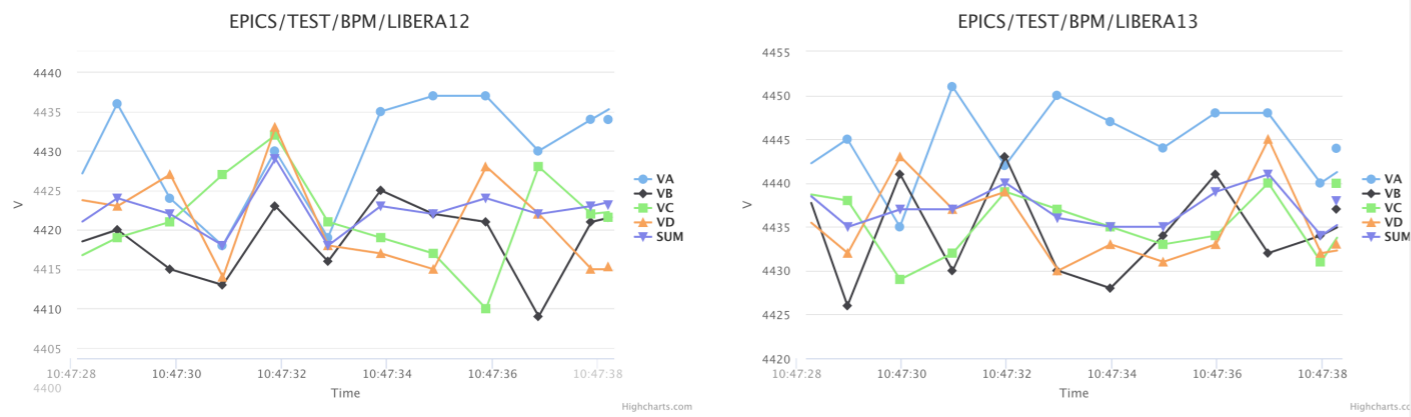
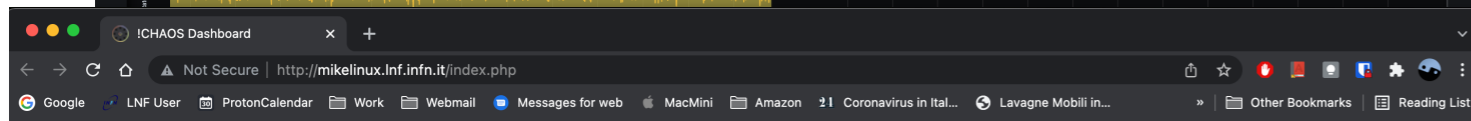
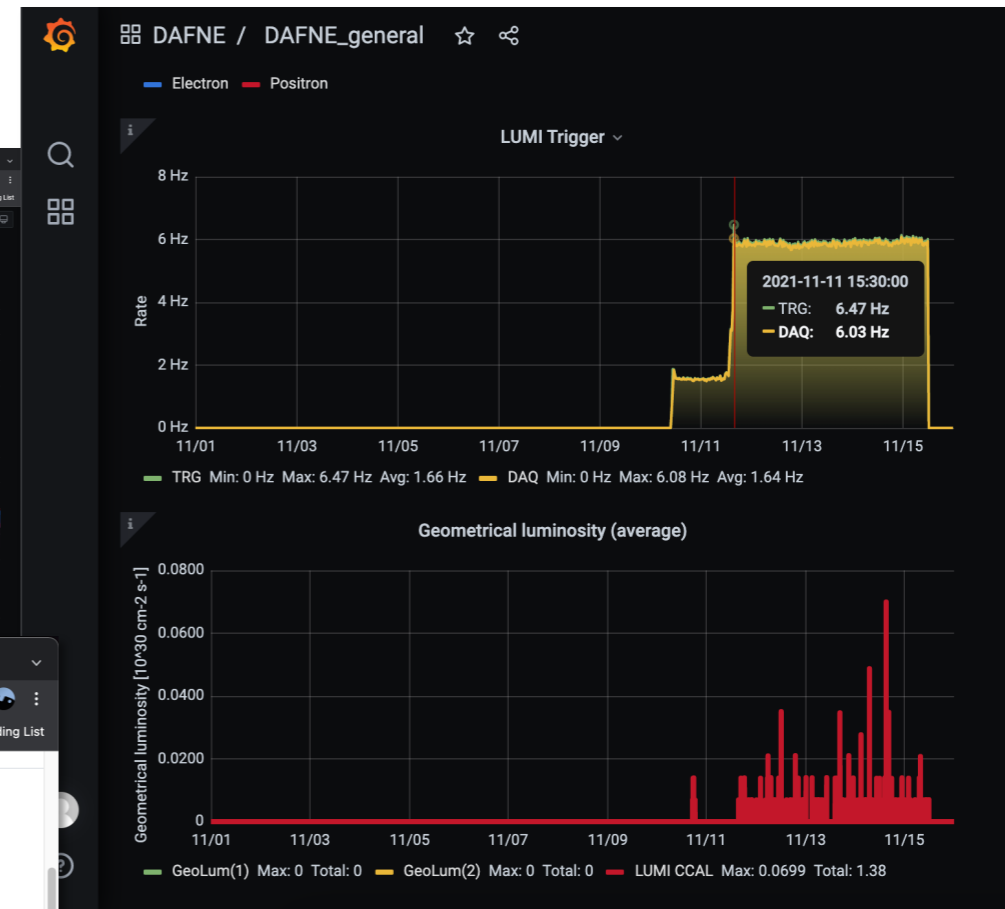
The screenshot displays a complex LabVIEW interface for a particle accelerator control system. The interface is organized into several functional areas:

- Left Panel (Project Tree):** Shows a hierarchical view of the project files, including folders for 'My' and 'Project'.
- Central Panel (Magnet Parameters Table):** A table listing various magnet elements with their current readout and target setting values.

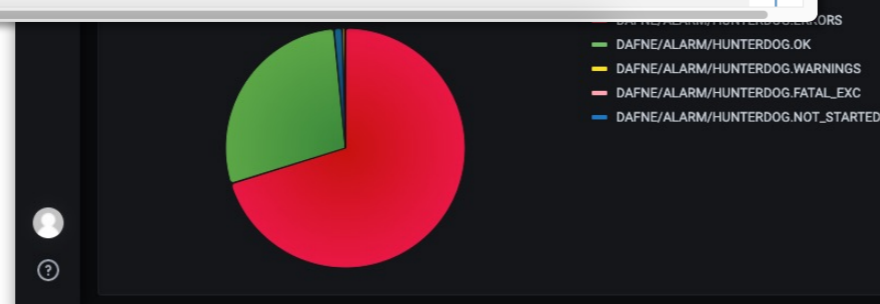
ELEMENT	READOUT	SETTING	SAVED	FLAGS
QUATM001	21.521	21.500		⚡ +
CHHTM001	0.000	-0.004		⚡
CVVTM001	-0.009	-1.900		⚡
QUATM002	67.324	67.500		⚡ -
CHHTM002	0.000	-0.001		⚡
CVVTM002	-0.018	-2.000		⚡
QUATM003	50.964	51.000		⚡ +
CHHTM003	0.001	-0.010		⚡
CVVTM003	-0.005	-0.004		⚡
QUATM004	13.022	13.000		⚡ -
DHPTB101	294.516	295.000		⚡ +
QUATB101	29.813	30.000		⚡ -
QUATB102	29.941	30.000		⚡ +
DHSTB001	320.451	320.400		⚡ +
QUATB001	29.953	30.000		⚡ -
CHHTB001	-0.007	-0.017		⚡
CVVTB001	0.005	0.000		⚡
QUATB002	34.943	35.000		⚡ +
CHHTB002	-0.010	0.010		⚡
CVVTB002	-0.012	0.000		⚡
DHPTB102	0.500	0.500		⚡
CHHTT008	-0.010	-0.005		⚡
CVVTT008	4.990	5.000		⚡
QUATB003	49.966	50.000		⚡ -
- Top-Right Panel (Status):** Displays real-time data including 'Time window zoom' (8x), 'DCTEL002', and current/voltage readouts (0.4 mA, 0.4 mA, -0.31 mA).
- Main Plot Area (ChargeMonitor.vi):** Shows a signal waveform plot with a 'Pause Acq' button and 'oversampling' indicator.
- Bottom-Right Panel (Main Ring Schematic):** A schematic diagram of the main ring with various components labeled (TM, TS, TB, TP, TE, TL, TR).
- Bottom Panel (Monitors):** Two bar charts showing 'Monitors' for 'LAC' and 'AMR' in units of [nC].

- acquisition/actuation
- communication
- storage
- **presentation**
- analysis / correlation

User Interface

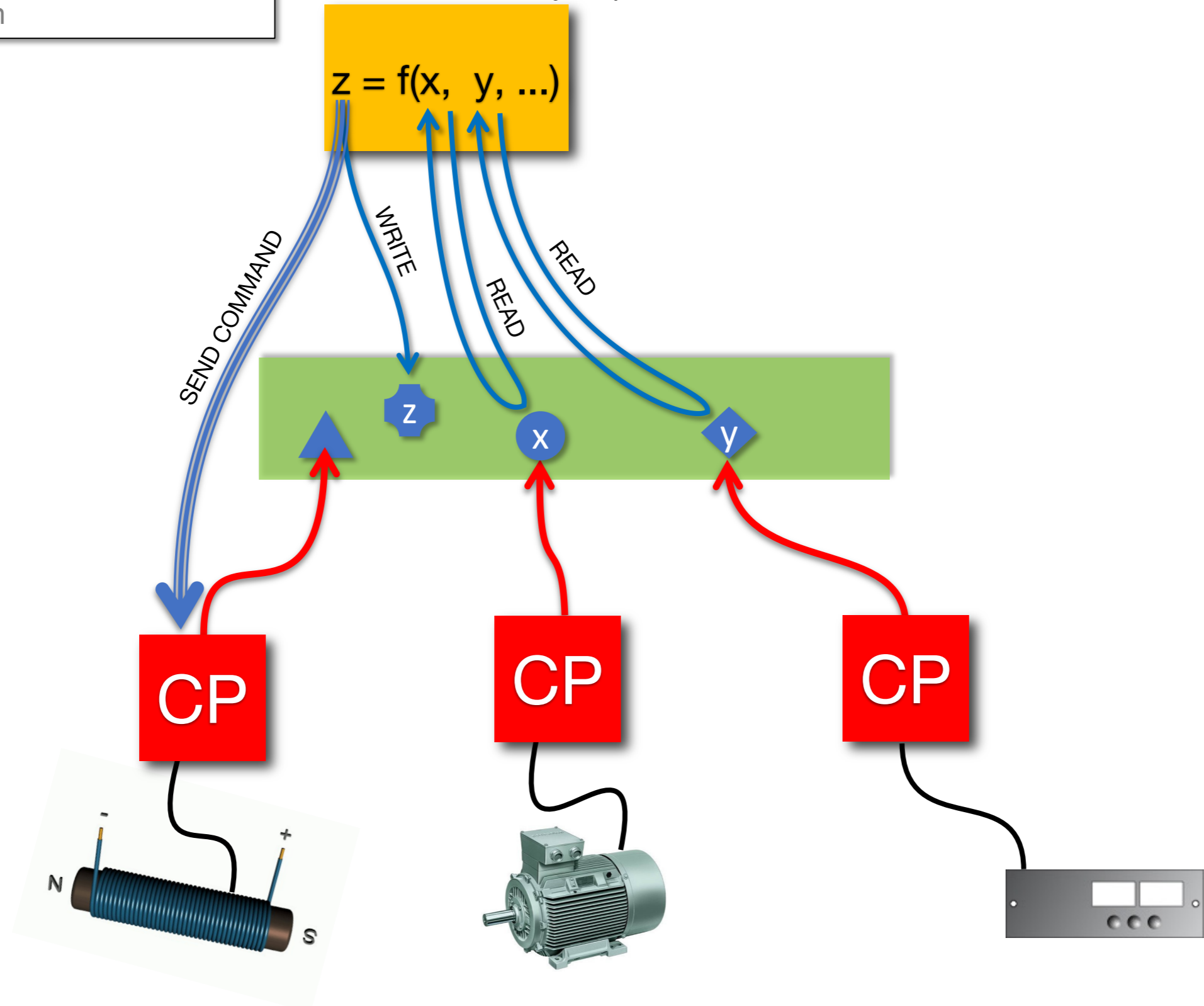


Sel	Element	Status	X	Y	VA	VB	VC	VD	SUM	Samples/Trigger	Alarms dev/cu	
<input type="checkbox"/>	EPICS/TEST/BPM/LIBERA01	▶	ψ	0.000	0.000	667	862	1166	1093	3788	0	No Trigger
<input type="checkbox"/>	EPICS/TEST/BPM/LIBERA02	▶	ψ	0.000	0.000	777	508	950	873	3110	0	No Trigger
<input type="checkbox"/>	EPICS/TEST/BPM/LIBERA03	▶	ψ	0.000	0.000	591	481	705	650	2428	0	No Trigger
<input type="checkbox"/>	EPICS/TEST/BPM/LIBERA05	▶	ψ	0.000	0.000	554	446	640	515	2156	0	No Trigger
<input type="checkbox"/>	EPICS/TEST/BPM/LIBERA06	▶	ψ	0.000	0.000	624	732	762	1920	4039	0	No Trigger

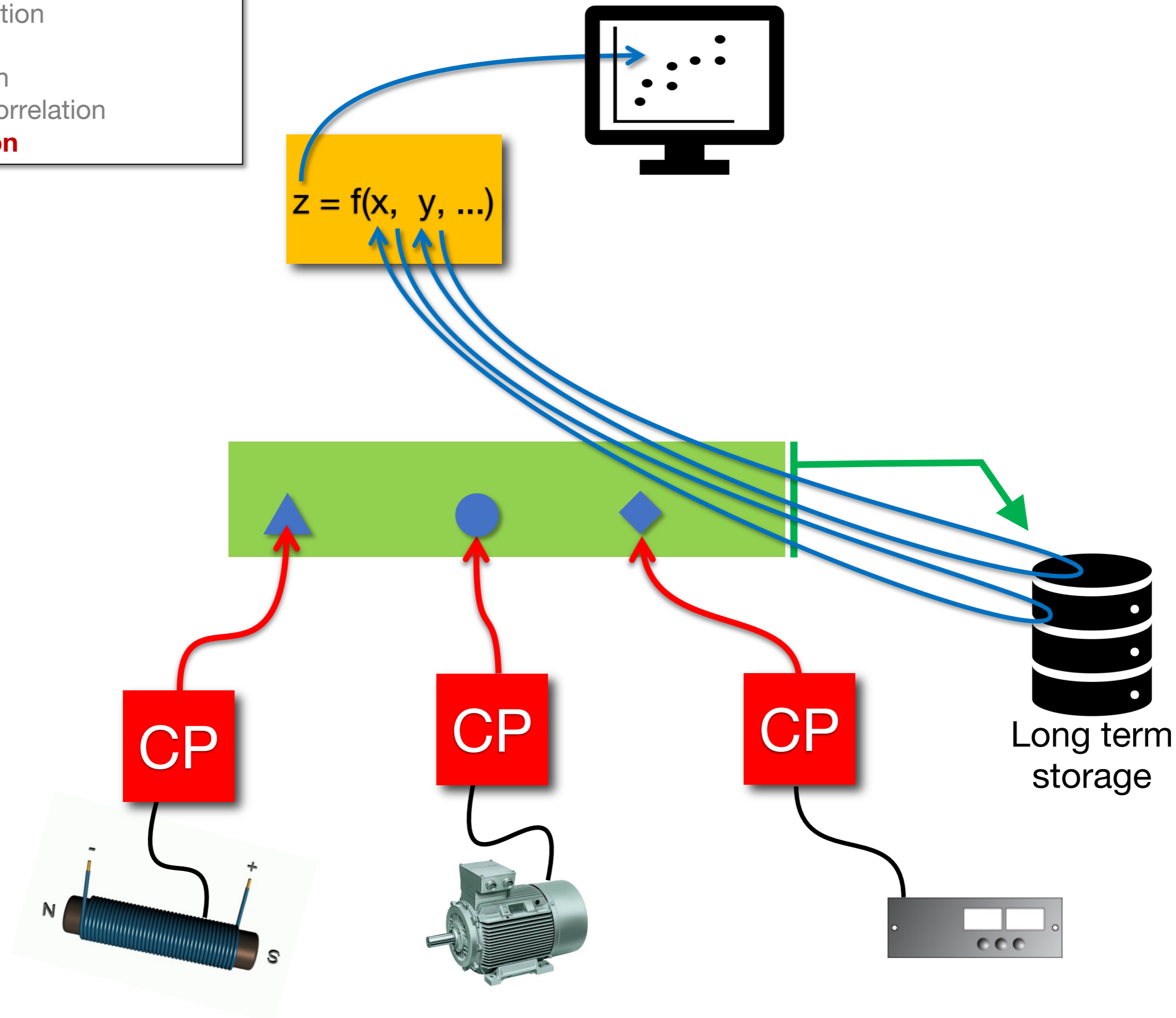


- acquisition/actuation
- communication
- storage
- presentation
- **analysis / correlation**
- historization

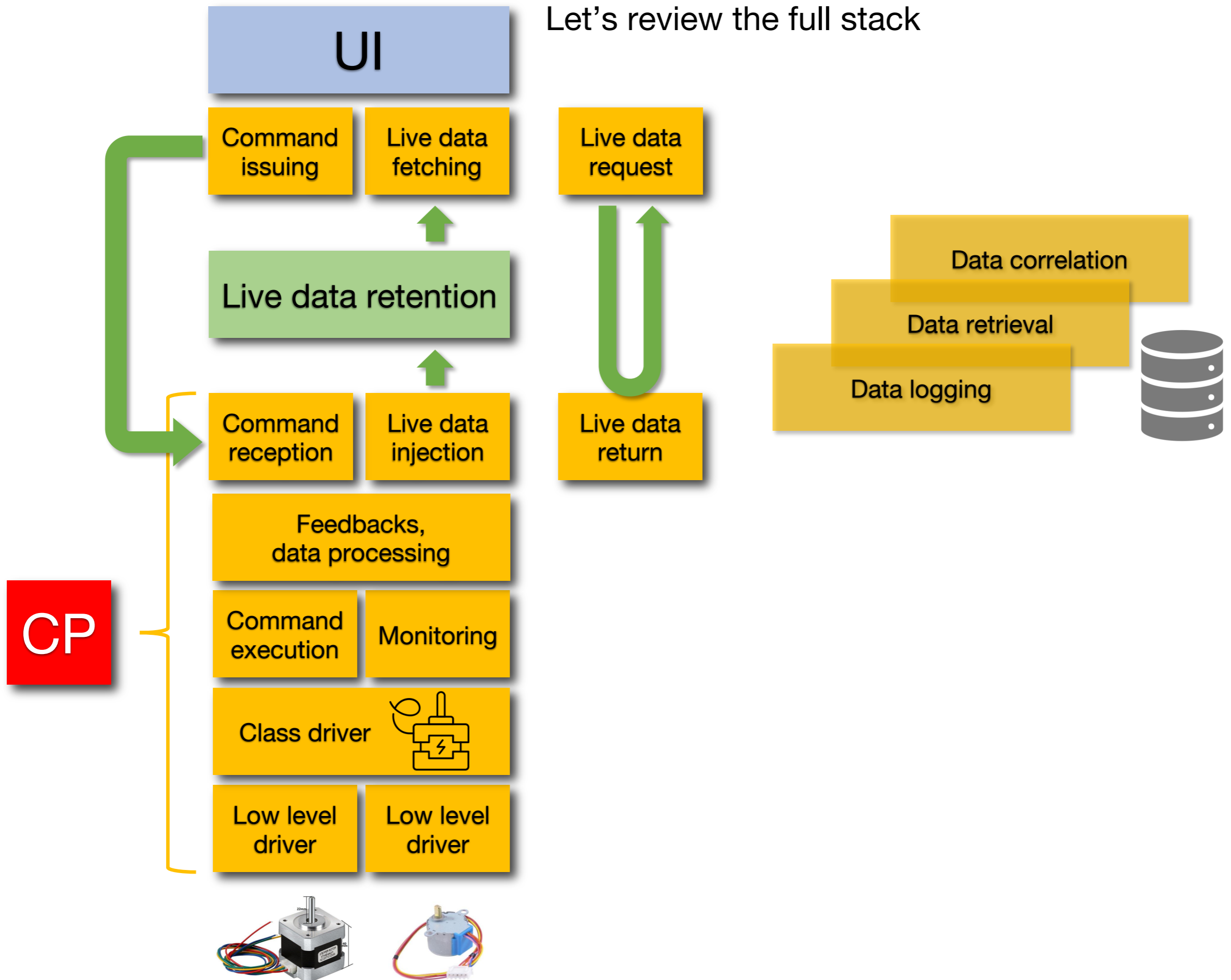
Correlation and analysis processes



- acquisition / actuation
- communication
- storage
- presentation
- analysis / correlation
- **historization**



Let's review the full stack



Golden rules

- Analysis → Design → Development → Qualification
- Do abstract physical devices and create classes
- **KISS** (Keep It Simple)

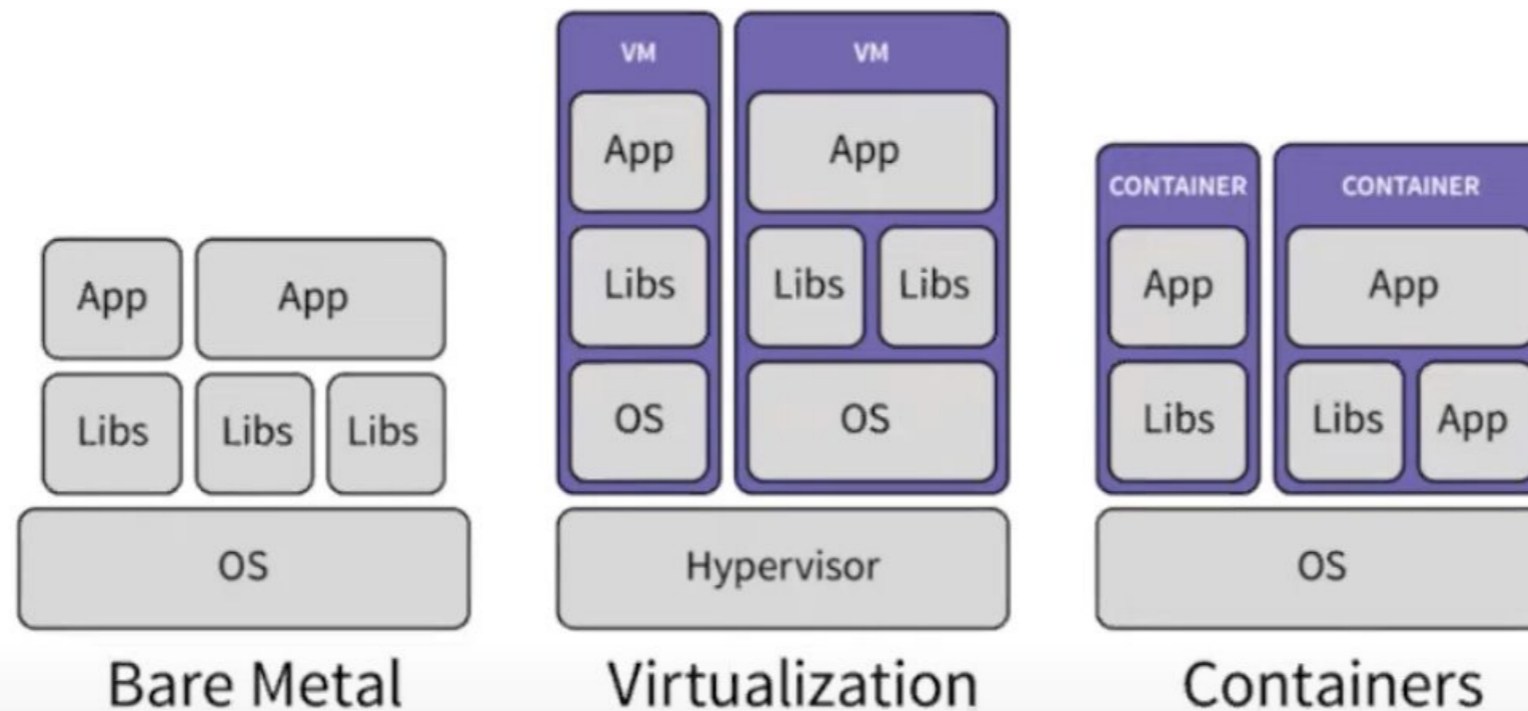
IT good methods & practices

- Development: [Agile approach](#), CI/CD, DevOps
- [Documentation](#), Asset Management
- [Project tracking](#)

The screenshot displays the JIRA Software interface for a project named 'Teams in Space'. The main view is a 'TIS-70 Scrum Board' with three columns: '12 To do', '2 In progress', and '3 Done'. The board is organized into two sections: 'TIS Developer Love' (3 issues) and 'Everything Else' (21 issues). Each issue card includes a title, a description, a priority indicator, a 'SeeSpaceEZ plus' button, and a user profile picture. The 'To do' column contains issues TIS-37, TIS-68, TIS-20, and TIS-12. The 'In progress' column contains issues TIS-10 and TIS-17. The 'Done' column contains issues TIS-8 and TIS-56. A third issue, TIS-45, is partially visible in the 'Done' column. The left sidebar shows navigation options like 'Backlog', 'Agile board', and 'Releases', along with project shortcuts.

IT interesting topics

- Big data
- AI
- Virtualization, Dockers, Orchestrators

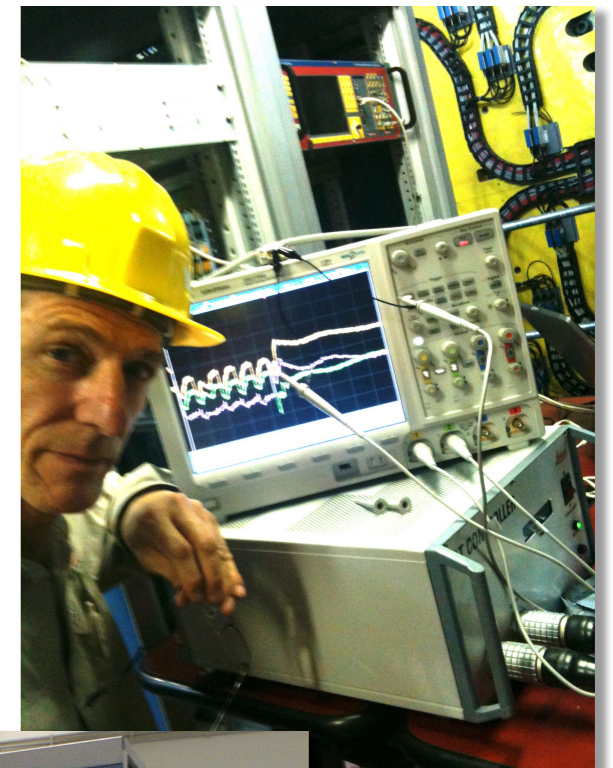


Containers, like VMs, isolate an application and its dependencies into a self-contained unit that can run anywhere.

Courtesy of Giles Knap (Diamond)

Professionals

- Data scientist
- Software Developer, Software Engineer, DevOps
- System Manager
- Network Manager
- HW system Integrator



Infrastructure @LNF

PRODUCTION CLUSTER

SERVICE CLUSTER

STORAGE 1/2 PB

SPARES

SERVICE CLUSTER

DEVELOPMENT CLUSTER

PRE-PRODUCTION CLUSTER

- 376 cores
- 824 GB RAM
- 500 TB HD

Today's exercise

