Si Automation

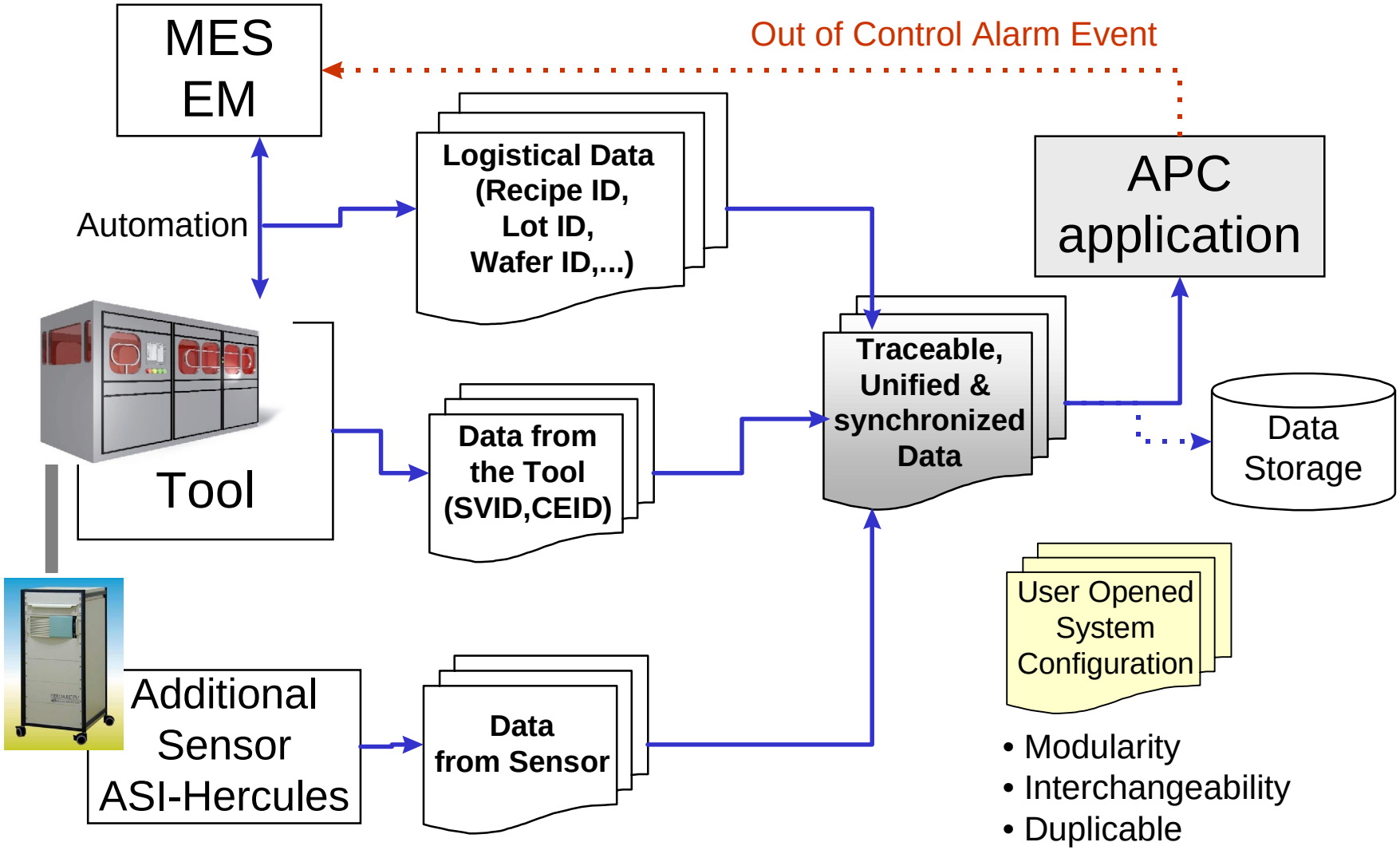
Sensor Integration & Data collection Project

Summary

- Project Goals to be achieved
- Project Issues to be solved
- How to answer in a Standard Approach ?
- Splitting the project into interoperable software modules
- Which Hardware Architecture ?
- Application Implementation
- Conclusions: main improvement to expand capabilities

Si Automation

Project Goals to be achieved



Si Automation

Project Issues to be solved

Additional sensor integration :

- **Hardware** interface : where to connect ? What kind ?
- **Software** interface : where to locate ? What protocol ?

Data Collection and unification :

- How to Collect data from the **tool** ?
- How to Collect data from the **sensor** ?
- How to Get **run context**, logistical data from automation (recipe, lot & wafer ID), in order to select appropriate **control strategy** and ensure **trace ability** ?
- How to perform **data synchronization and unification** ?

Si Automation

Project Issues to be solved (cont.)

Application/Automation contention :

- how to **share** the tool communication line with Automation ?
- how to deal with the Fab Automation Architecture (to get tool context and tool data) without **disturbance** ?

Data Transfer & storage :

- how to **store** & **transfer** data to APC analysis software ?

Application :

- How to leave the **end user** free to configure, modify and duplicate the whole or part of the application ?

Si Automation

How to answer in a Standard Approach ?

- How to make this project a **re-usable** reference, using **SEMI Standards and SEMATECH guidelines** ?

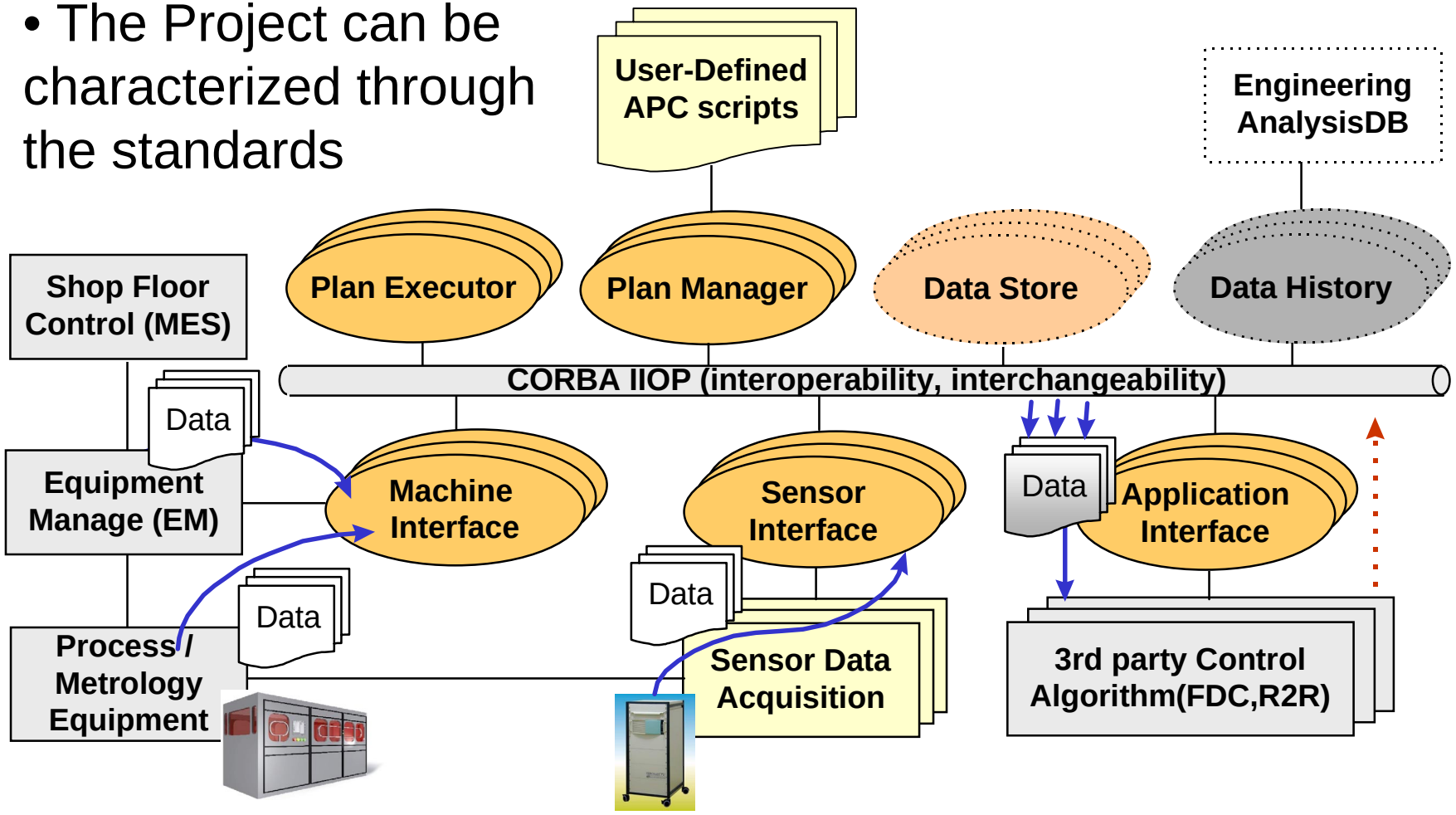
Main applicable standards references :

- **SEMI E93 APC CIM Frameworks** - Provisional Specification for CIM Framework Advanced Process Control Component
- **SEMI E97 CIM Frameworks** - Provisional Specification for CIM Framework Global Declarations and Abstract Interfaces.
- **SEMATECH #99053735A** - APC Framework initiative
- **SEMATECH #00094004A** - Automatic Data Collection Baseline Requirements: Level 1 and Level 2 Events and Variables.
- **SEMI E54** - Sensor/Actuator Network
- **SEMATECH #94102564ATR** - Sensor/Actuator Bus
- **SEMATECH #94102567ASTD** - Device Interoperability Guideline for Sensors, Actuators, and Controllers.

How to answer in a Standard Approach ? (cont.)

Project View in the CIM framework and APC Framework Context

- The Project can be characterized through the standards



Si Automation

How to answer in a Standard Approach ? (cont.)

Some CIM & APC Framework Issues

The Proposed Standards approach :

- **Split** the project into reusable & **interoperable** software **modules** or components.
- **CORBA** based object interface to achieve the interoperability and the capability to "**Plug & Play**" any **functional compliant** components or devices, from any vendor.

Current Barriers & Remaining Issues :

- **Poor** CORBA **implementation** in today's Fabs
- Needs CORBA **gateways** (IDL) to be **developed** for most of today's software tools and most of devices. A big effort.
- No clear standard recommendation about the **Hardware architecture**, regarding the **performance** impact : Network traffic, global system speed and reactivity...

Si Automation

How to answer in a Standard Approach ? (cont.)

Solutions for this project :

- Some of software components will **interact directly** without the needs of CORBA, keeping the modularity at a high level leaving possible future Implementation of CORBA interface.
- Keep the standards as a **strong guideline** and comply with the proposed object descriptors **naming conventions** to build the solution
- Complete with **field experience** and propose **complementary** solutions, if no standard coverage.
- Develop Software modules with **standardized interfaces** minimizing the re-coding impact (limited to specifics driver or gateway) in case of device interchange or upgrade.
- suggest a generic approach of the **Hardware Architecture**.
-

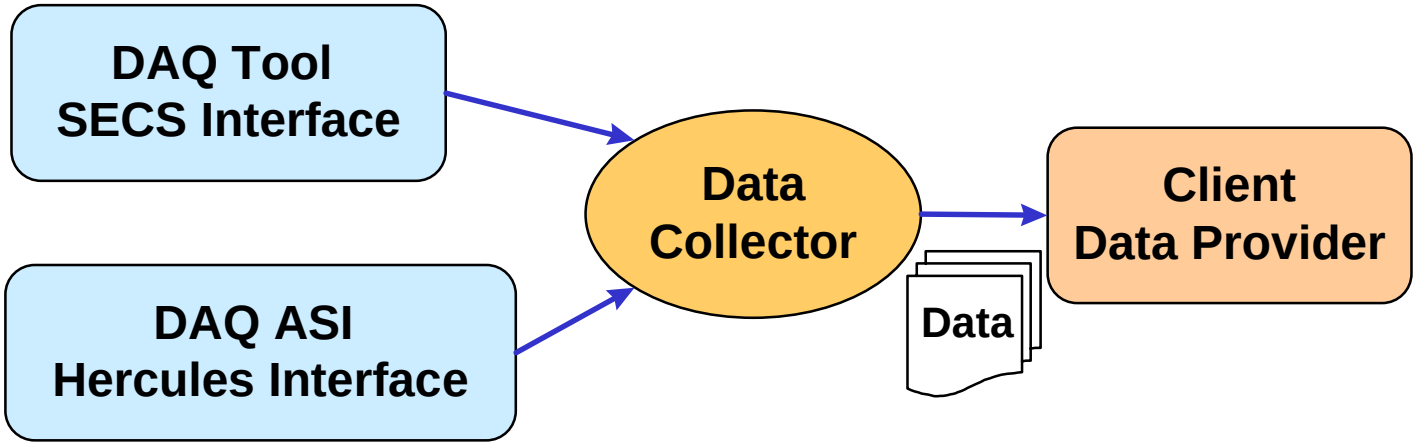
Si Automation

Splitting the project into interoperable software modules

Data Collector component :

Part of the Plan Executor and Plan Manager, The first project brick is the **Data Collector** component, whose the design specification was **formally defined**, by compiling and complete the information found in related standards :

- The **Data Collector** is the name given to the **component** dedicated to:
  - **Collect and unify data from** one or several entities called **Data AcQuisition points** (**DAQ**),
  - **Provide** these same **data requested by** another or several others entities called "**Clients**", for data storage and/or analysis further operations.

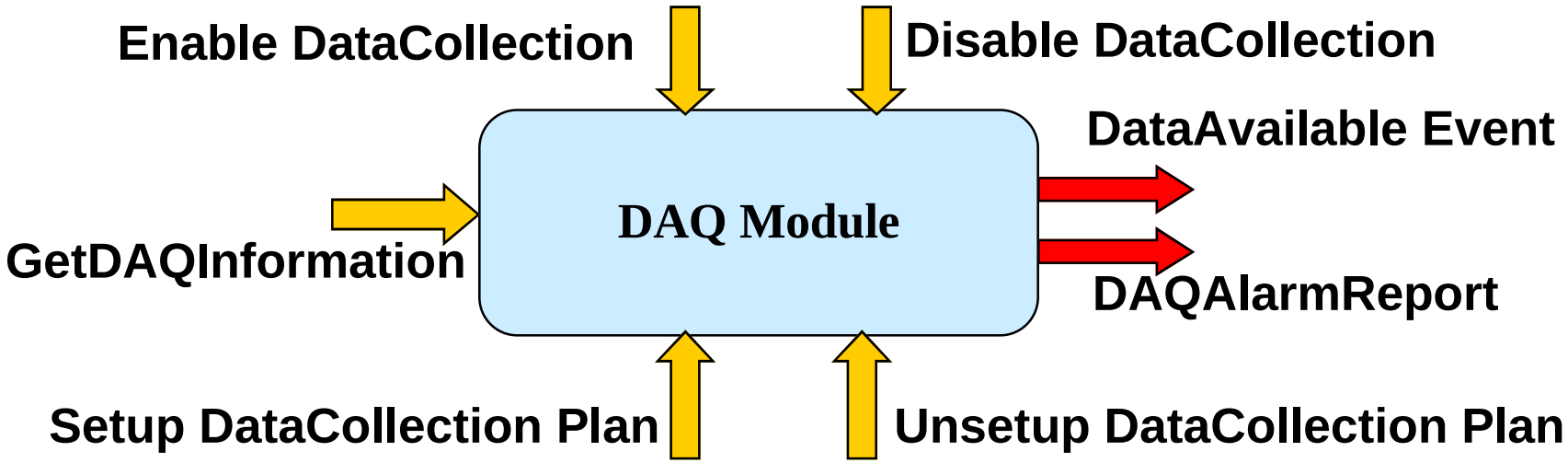


Automation

Splitting the project into interoperable modules (cont.)

Data Collector - A DAQ is an entity able to:

- **Publish** a list of data names available for data collection,
- **Receive** a list of data name for the data to collect and a data sampling period,
- **Receive** a specified frequency for data sampling,
- **Collect data** on a specific system, with a default or specified data sampling frequency,
- **Provide** data samples,
- **Start / Stop** the data sampling when receiving a start / stop command

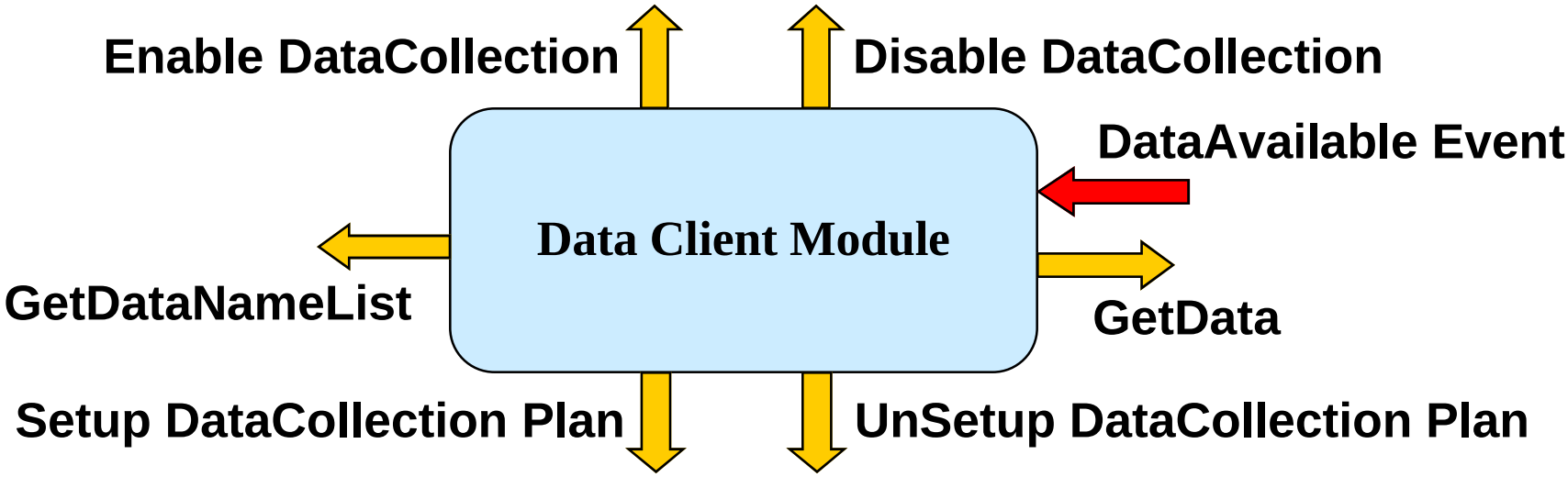


Si Automation

Splitting the project into interoperable modules (cont.)

Data Collector : A Data Client is an entity able to:

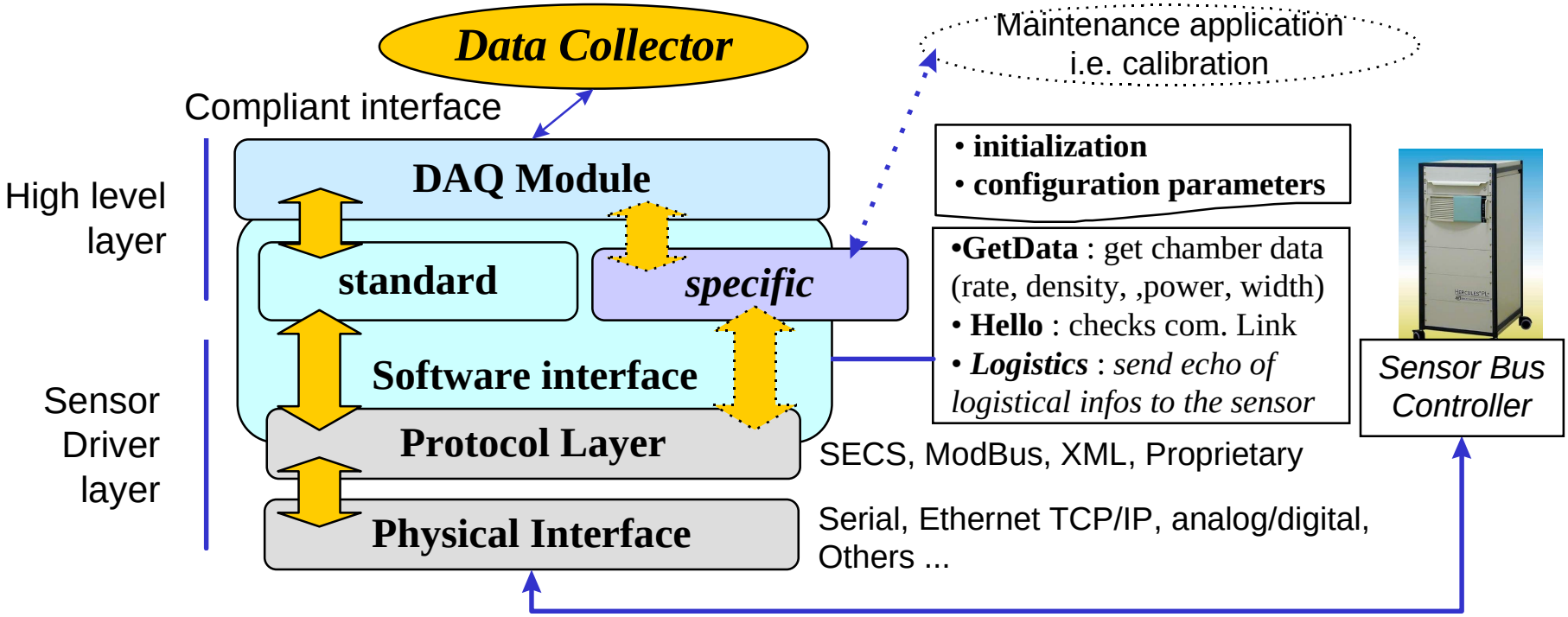
- **Get** the list of all data name available for data collection,
- **Setup** a DataCollection Plan,
- **Enable** the DataCollection,
- **Disable** the DataCollection,
- **Receive** a list of data value, matching with the list of data name (this list of data value can be requested or received on event).



Splitting the project into interoperable modules (cont.)

Sensor Integration - DAQ applied to Hercules interface :

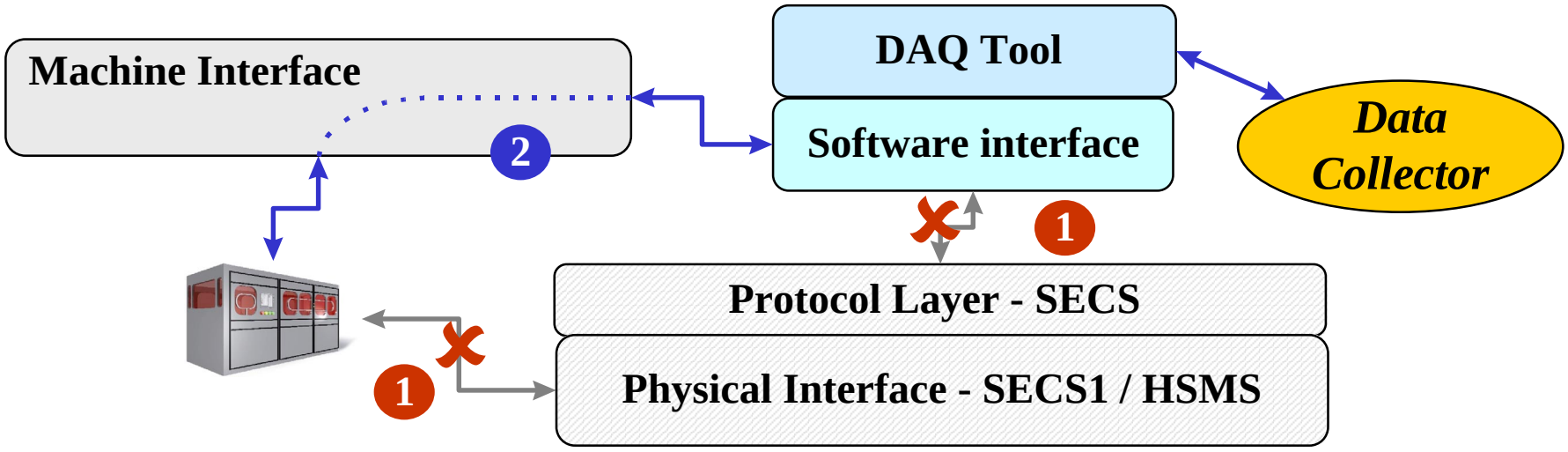
- **Sensor integration driver** : effort limited to the protocol layer implementation using standard DAQ interface at upper layer
- **Serial link** hardware interface used for this project.
- Another alternative will be to implement the Hercules **HSMS interface** : only the lower layer of the interface have to be modified without another coding impact on the application



Splitting the project into interoperable modules (cont.)

Tool Data Collection - DAQ applied to the tool :

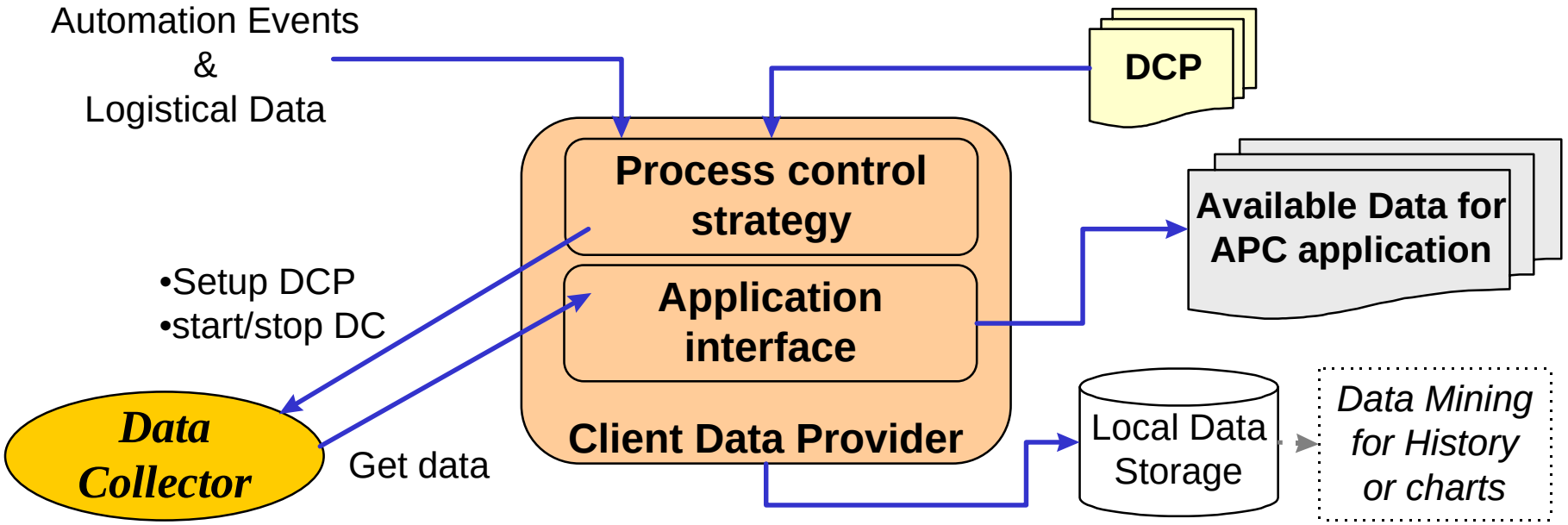
- The retained method to Collect Data from the tool, performed by the DAQ Tool, is to use **SECS Status Variable (SVID) requests** through serial line (or HSMS port, upon configuration)
- because of the **single SECS line** (serial port or HSMS TCP/IP Socket), already used by the Automation, it's not possible to connect the DAQ directly to the tool. (solution ❶)
- The software interface should be redirected to a new software component, called **Machine Interface**, able to **share the line with the Automation** (solution ❷). This component will be further discussed.



Splitting the project into interoperable modules (cont.)

Client Data Provider :

- Play **several roles** in Plan Execution/Management, Data Store and application interface
- **Process control strategy** : setup **Data Collection Plan** (DCP) in **synchronization** with automation events
- **Unify** and **format data** ready to use by APC **application** (CSV files)
- **Store** the whole data context in a local data base, part of Fab **distributed data base** concept

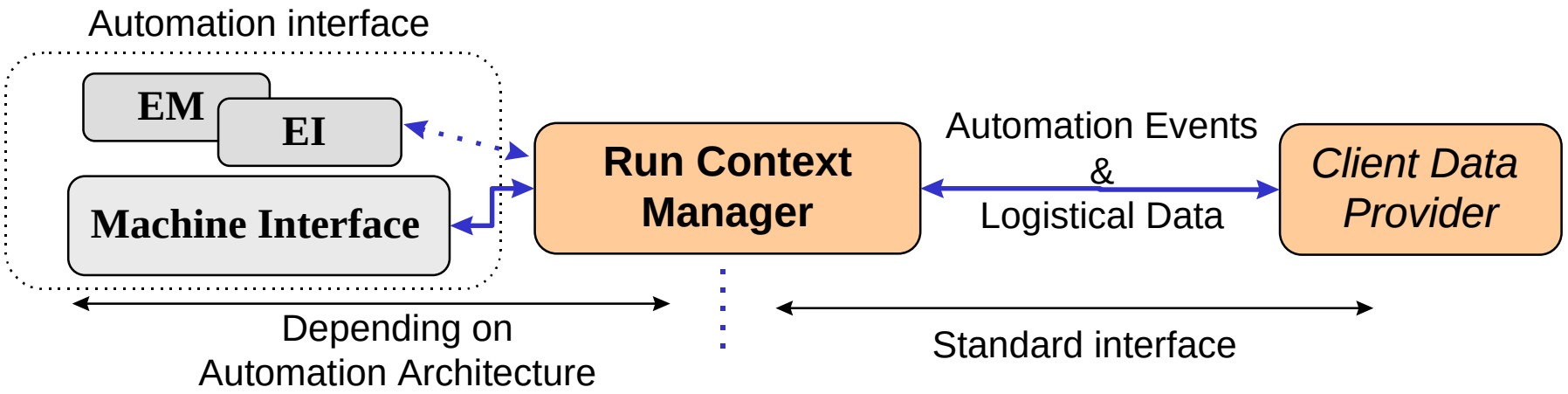


Si Automation

Splitting the project into interoperable modules (cont.)

Run Context Manager :

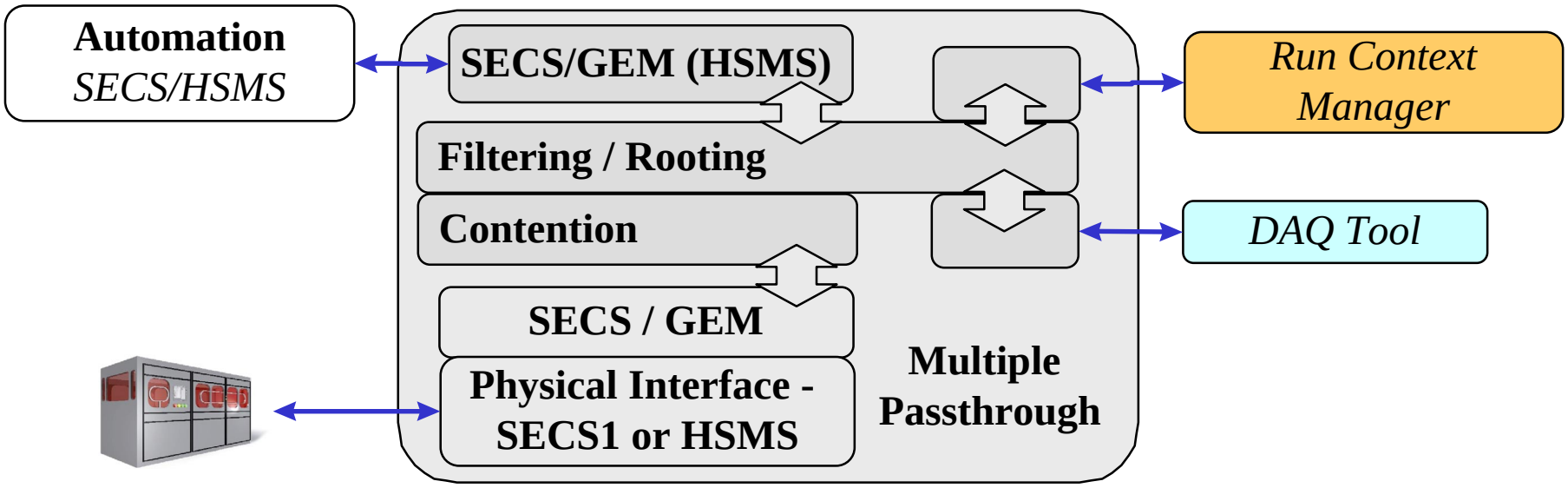
• As an **interface** with the **Automation**, Its role is to **provide**, in a **standard way**, **context information** : logistical data (i.e. Process ID, Batch/Lot/wafer ID,...) and synchronization events (i.e. start & stop)
• Depending on the Fab automation **architecture**, there is **many way to get these information** : i.e. from Equipment Manager or interface (EM or EI), from the equipment through **the SECS/GEM line**. This last case, chosen for this project, is more complex, but adaptable to any kind of architecture. It requires the **Machine Interface** module.
• This modular approach will allow to change the automation interface without major application changes.



Splitting the project into interoperable modules (cont.)

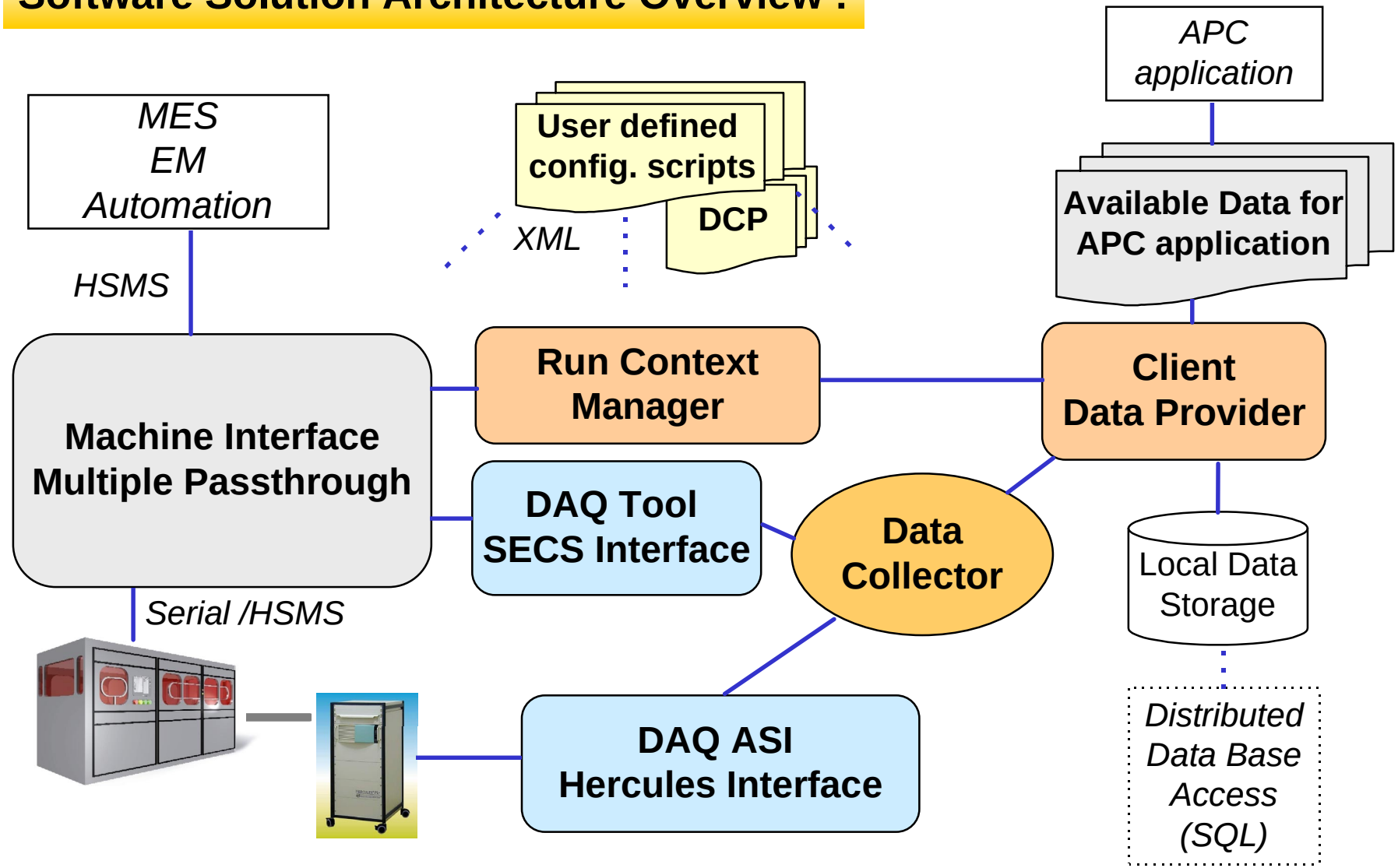
Machine Interface : Multiple Passthrough

- Based on HSMS/SECS Passthrough, the **role** of this component is **multiple**.
- Give an access to the **SECS/GEM tool interface** for data acquisition
- Give **Automation** logistical information (process ID, Lot ID, Wafer ID...)
- Give **Process** information & **event** to **synchronize** data collection (start / stop process, recipe step,...)
- **Share** the line between Automation and Data Collection (Priority given to Automation in case of access conflict)



Splitting the project into interoperable modules (cont.)

Software Solution Architecture Overview :



Automation

Which Hardware Architecture ?

Hardware solution : a Tool integration platform

• Through the project software implementation needs, we suggest to define some draft **requirements** for a standardized “tool integration platform” (hardware and basic software layer) :

• **Distributed** : dedicated to the tool, allowing local storage and treatment increasing reactivity and decreasing network traffic

• **Connectivity** : allowing tool, peripherals, sensors/actuators and metrology integration, embedding basic Hardware interface & software protocols.

• **Basic interoperable architecture** allowing intra or inter platform software components interactions (i.e. COM for Windows NT, TCP/IP). Ready for CORBA Object interface capability for external interoperability.

• **Open architecture, interchangeability** : Capability to install standard compliant tool/sensor drivers or other third party software

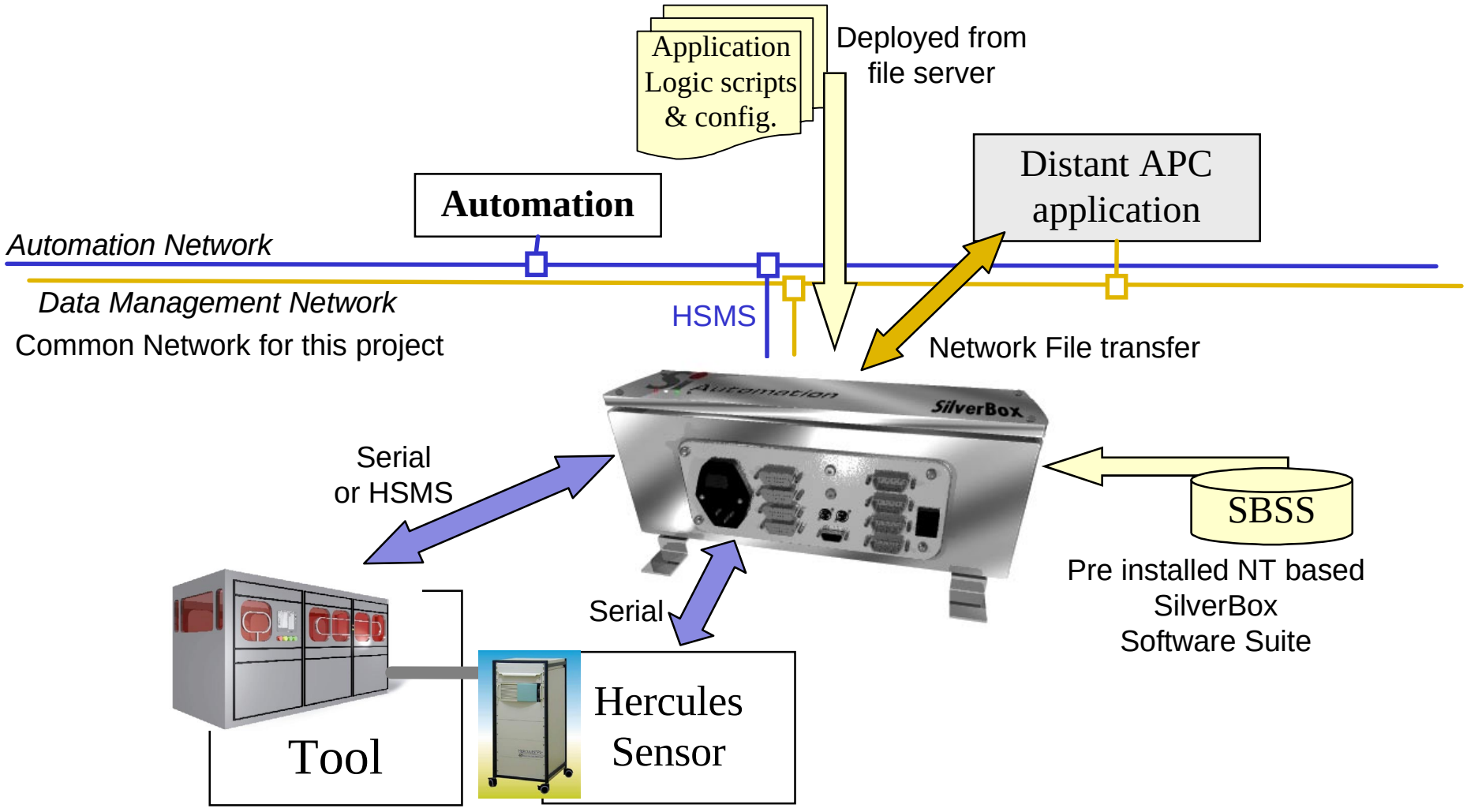
• **Logic Script** allowing end user configuration (i.e. XML format)

• Hardware unification, administration tools and installation rules allowing **fast deployment** and **maintenance**

Si Automation Application Implementation

Final Application Diagram :

- Si Automation **SilverBox™** used as “Tool integration platform”



Si Automation

Conclusions : main improvement to expand capabilities

Tool provider - Highway for data collection :

• Give to the tool a **separate and dedicated** line for **data management** (i.e. using HSMS with different socket from the Automation one), allowing high data collection rate, without disturbing the automation exchanges and the process controller.

Sensor provider :

• Enabling sensor to become "**plug & play**" by using one or several well known kind of interface and protocol : Ethernet (HSMS, ModBus TCP-Open ModBus), RS232 based, USB or IEEE1394 (FireWire or i-Link), DeviceNet... , and providing software drivers compliant with "CIM/APC framework" and "tool integration platform" standards.

Si Automation

Conclusions : main improvement to expand capabilities

Solution providers :

• Provide software solution and sensor interfaces is not sufficient, the **integration platform** needs to be defined : unified standards around the flexible architecture of this platform could be envisaged, giving a start point for CIM/APC framework standards implementation. This unification will make solution duplication very easy.
• Si Automation's **SilverBox™** is a first version of this platform. *(See the next slide about a draft architecture view)*

IC manufacturers : the End Users

• want to uses the BOB solutions at a maximum level of interchangeability, configuration flexibility and performance. The idea of a tool integration platform fulfill this criteria, by opposition to the ideas to embed more and more software proprietary solution integrated to the tool itself, or build more and more centralized application leading to Fab network saturation.

Si Automation

Tool Integration Platform Architecture