

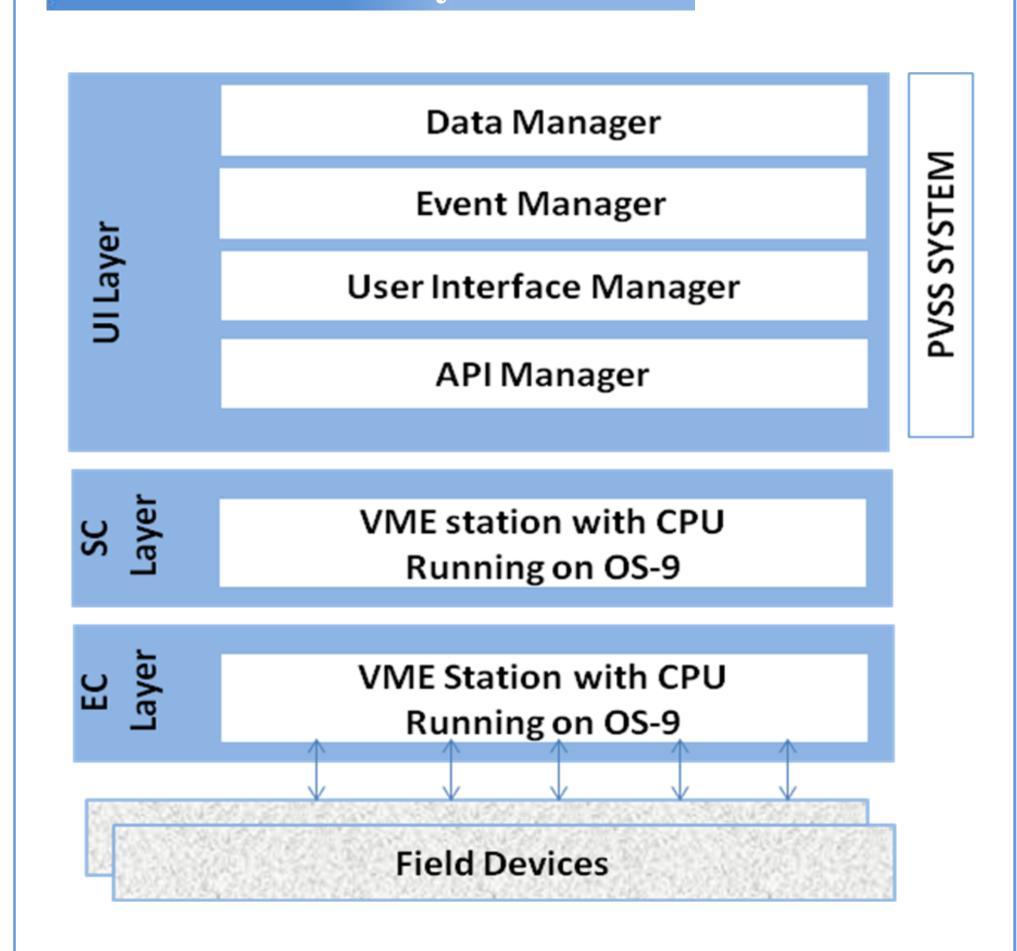
# API MANAGER IMPLEMENTATION AND ITS USE FOR INDUS ACCELERATOR CONTROL

Bhavna N. Merh, R. K. Agrawal, K. Barpande, P. Fatnani, C. P. Navathe Raja Ramanna Centre for Advanced Technology (RRCAT), Indore, India *Email: bhavna@rrcat.gov.in* 

#### **Abstract**

The control system software needed for operation of Indus accelerators is coupled to the underlying firmware and hardware of the control system by the Application Programming Interface (API) manager. In the three-layered architecture of Indus control system, PVSS-II SCADA is being used at the layer-1(L1) for control and monitoring of various sub-systems. The layer-2(L2) consists of VME bus based system. The API manager plays a crucial role in interfacing the L1 and L2 of the control system. It has to interact with both the PVSS database and the L2. In order to access the PVSS database it uses the PVSS API, a C++ class library, whereas in order to access the L2 custom functions have been built. Several other custom functionalities have also been implemented. The paper presents the important aspects of the API manager like its implementation, its interface mechanism to the lower layer and features like configurability, reusable classes, multithreading capability etc.

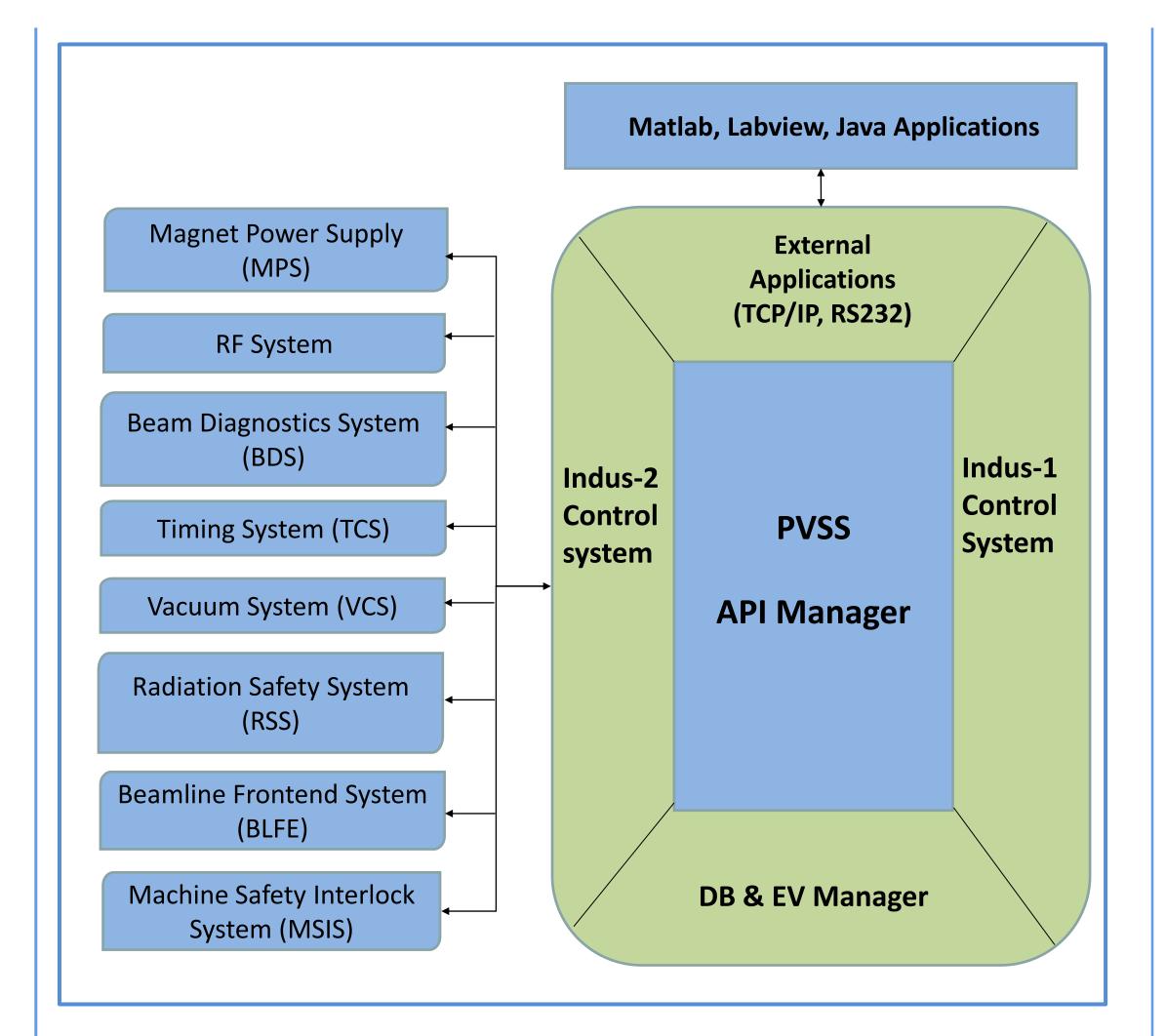
# **Indus-2 Control System**



- PVSS (Process visualization and control system) SCADA system works at user Interface (UI or L1) Layer.
- The lower layers are Supervisory Control (SC or L2) and Equipment Controller (EC or L3) layers.
- The API manager is interfaced to the SC layer over Ethernet.

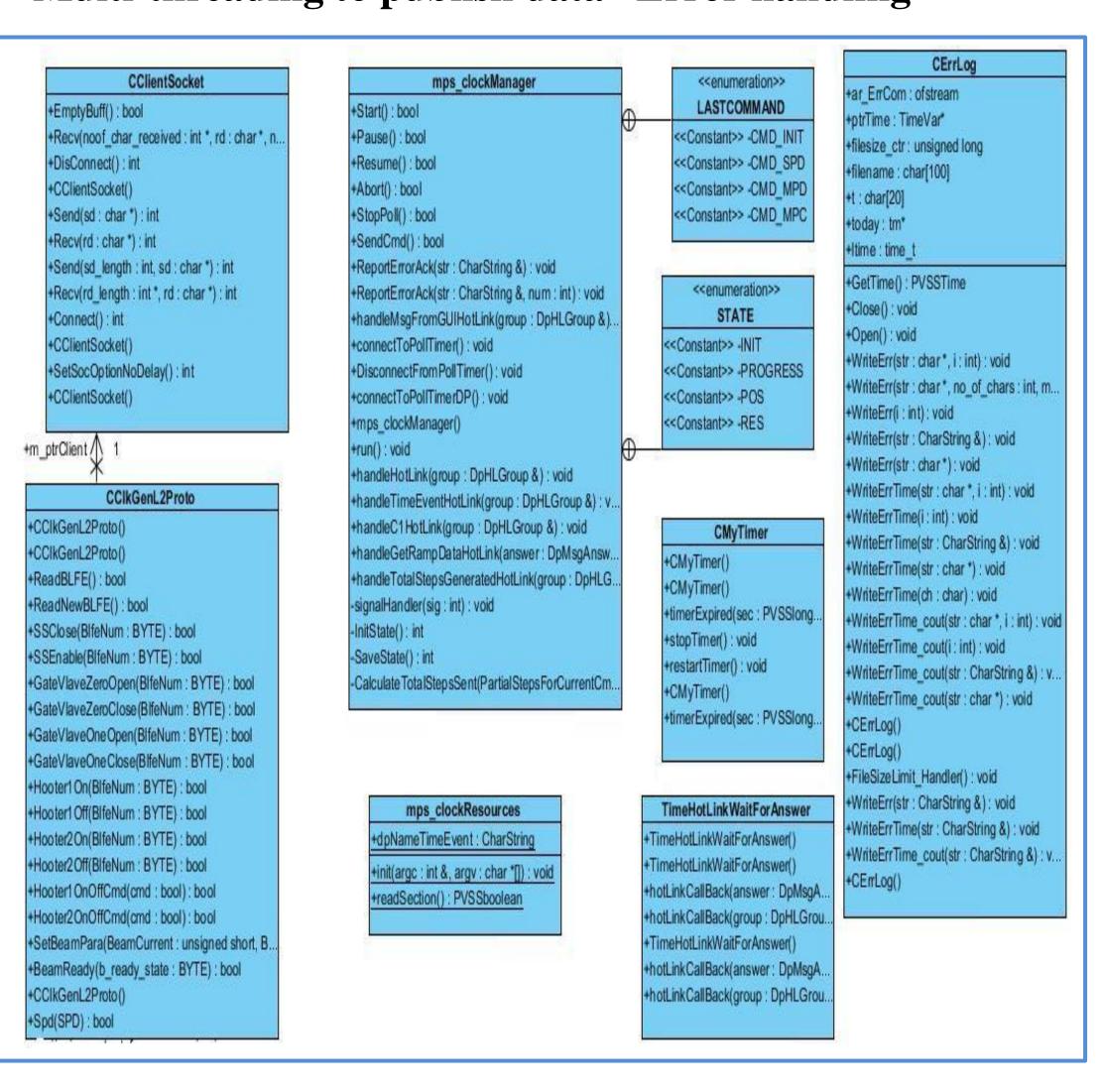
# **API Manager for Indus Control**

- A C++ application interface
- It allows the developer to implement his own custom functions together with full access to PVSS database.



#### **API Manager Features**

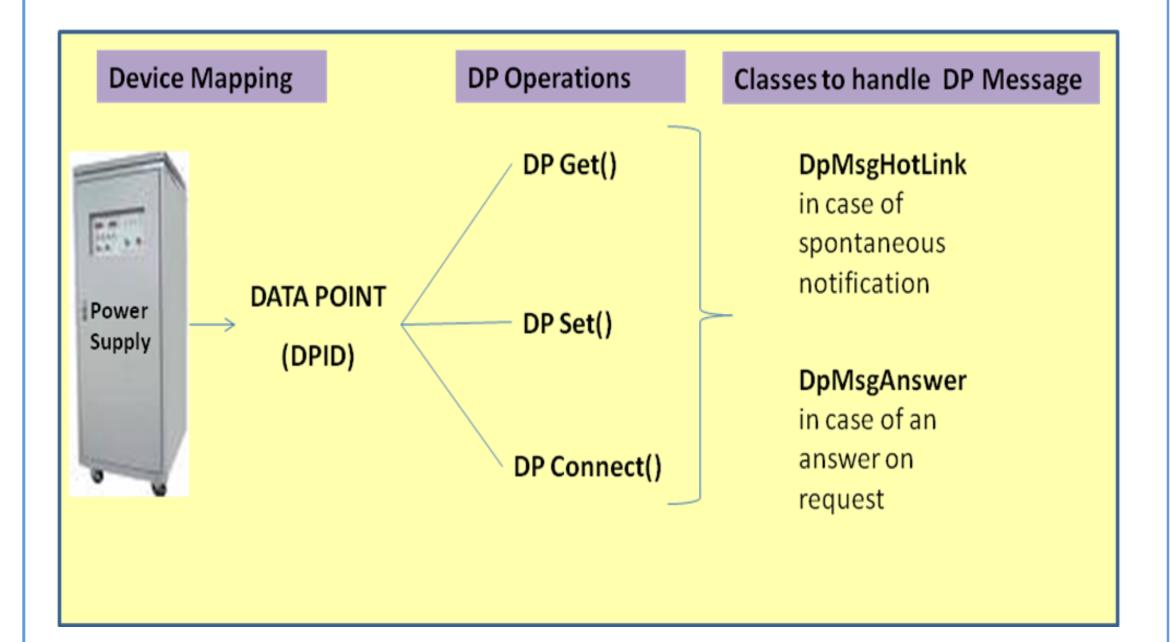
- Object oriented approach
- State Based
- Easily configurable
- Data Refresh on request
- Multi-threading to publish data Error handling



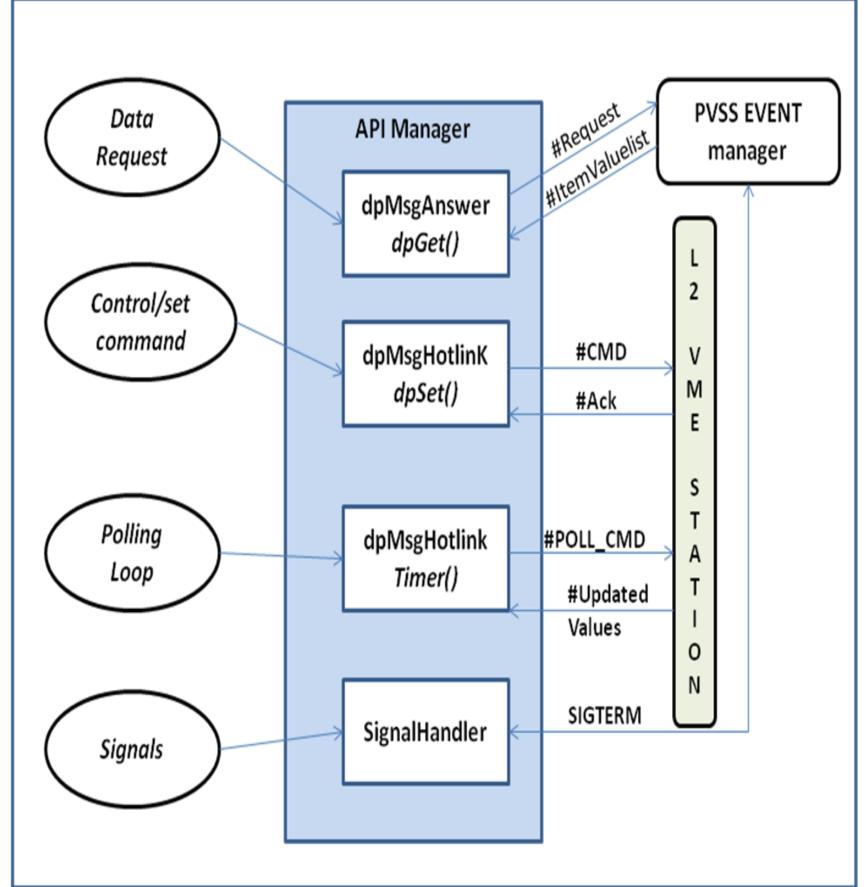
# Message Handling in API

- Each message is composed of group(s) (DpMsgAnswer, DpHLGroup class) of DpVCItem (Data Point(DP) value changes)[3].
- In order to receive notification for any DP event HotlinkWaitForAnswer is specified as:

Static PVSSboolean dpConnect(const DpIdentifier &dpId, WaitForAnswer \*wait, PVSSboolean del = PVSS\_TRUE);



### **API Manager Functions**



#### **Initialization Routines**

- Load device configuration information
- Interface to PVSS DB & EV
- Establish connection to SC Layer

#### **Periodic Routines**

Polling

Conversion

Data filtering

• Control /Set commands

• Data transformation • Event log

System Name	No. of Devices (approx.)	Total DP handled by API (approx.)
I2- MPS	174	6960
I2-RF	25	320
I2-VCS	220	1350
I2-TCS	10	145
I2-BDS	80	235
I2-RSSS	81	378
I2-BLFE	270	756
I2-MSIS	-	165
I1-MPS	97	112

## Conclusion

The Indus-2 API managers have been developed and commissioned in 2005. Since then augmentations and new additions have been made. All features and functionalities mentioned in the paper have been implemented all through these years. API managers have been running with no crash event being reported. The load of the over all system has been nearly constant and lies between 17-21% with API manager load is maximum 2%.

## References

- [1] PVSS-II is a SCADA package from ETM, Austria
- [2] PVSS Driver Development by ETM
- [3] H. Milcent, P. Burkimsher, W. Salter, "How Do I Write a Driver? Ctrl Managers, API Managers and Drivers", Release 2.1, 2003
- [4] PVSS-II API Documentation by ETM