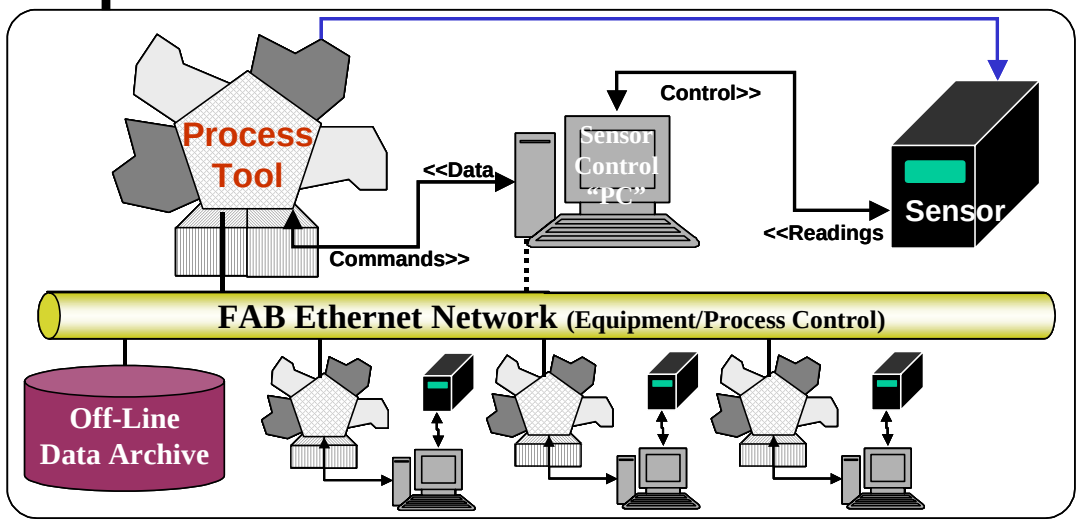
Migration of Intelligent Autonomy into Embedded Instrumentation for the Optimization of Process Monitoring and Control

AEC /APC Symposium September 2005

Rick Daignault

Environmental Conditions

- **Many process tool installations require industrial computers (PCs) for sensor control and monitoring**
  - **Cost and maintenance burden for OEM and FAB**
  - **More potential failure points**
  - **Degraded communication integrity:**
    - Added latency,
    - Cabling



- **PCs typically use Windows® Operating System**
  - **Susceptible to viruses and hacking when networked**
  - **Not real-time performance**

Marketplace

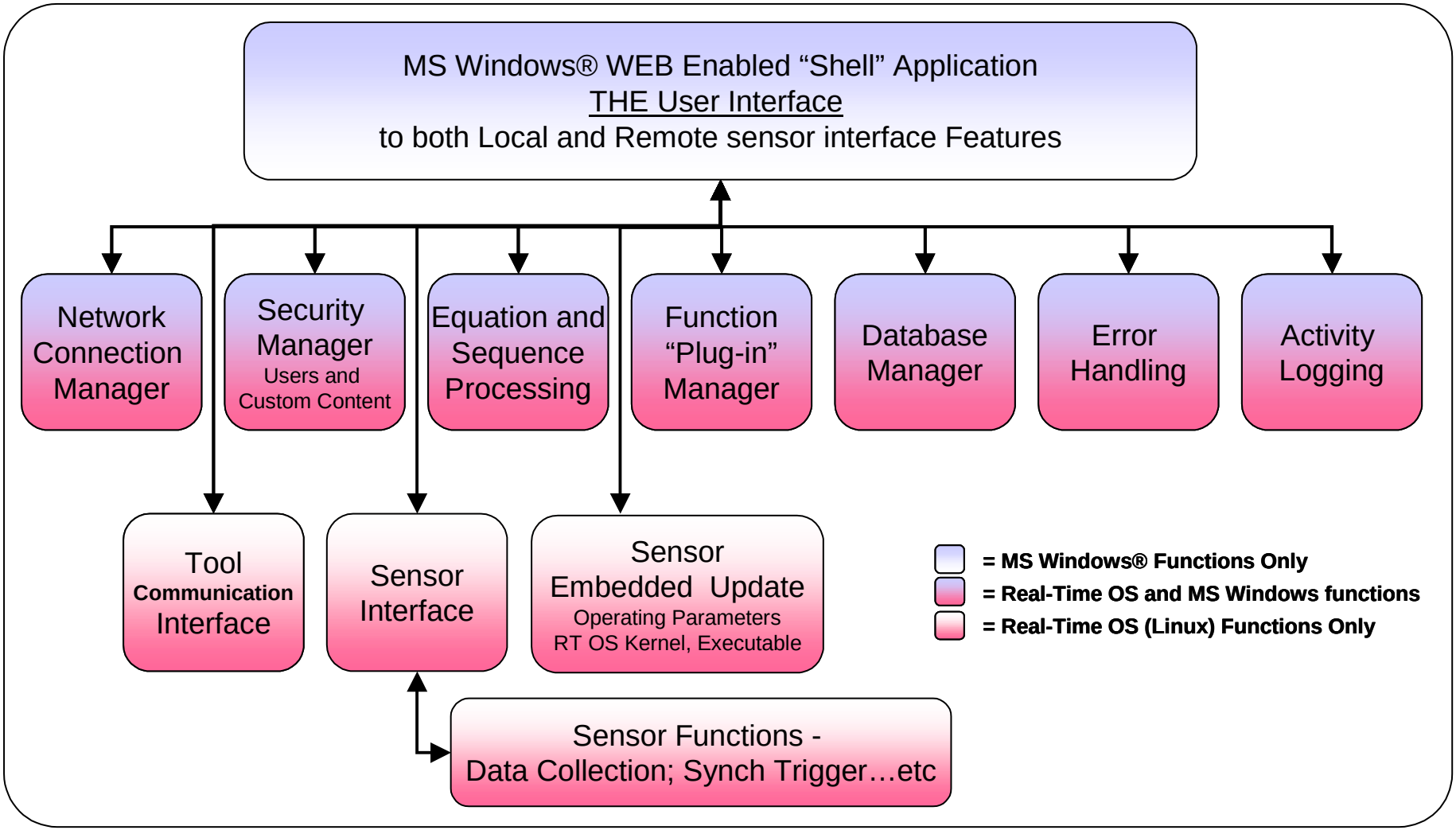
- **Specific markets require customized PC solutions**
  - **Specific PC Vendors required in some cases.**
  - **Versions of Windows® OS “Frozen”.**
  - **Windows sensor application software interface “productized”.**
  - **Multiple versions of PC application software must be maintained.**
  - **Many existing applications reaching limits of enhancements and customization.**
  - **Manufacturing and labeling of PC components vary by market.**

Migration Solution

- **Eliminate the industrial PC and Windows® OS**
- **Separate real-time functions from User Interface(UI)**
  - **Adopt a truly deterministic / real-time embedded OS and embed the real-time (RT) functions in the Sensor**
  - **Allow any PC platform to act as an "as needed" UI**
- **"Modularize" software elements (RT and UI)**
  - **Create automated method for adding / removing functional components**
  - **Allow specific marketplace users to perform their own customization**
- **Adopt Industry and Semi Standards**
  - **Communication; Security; Diagnostics…etc**

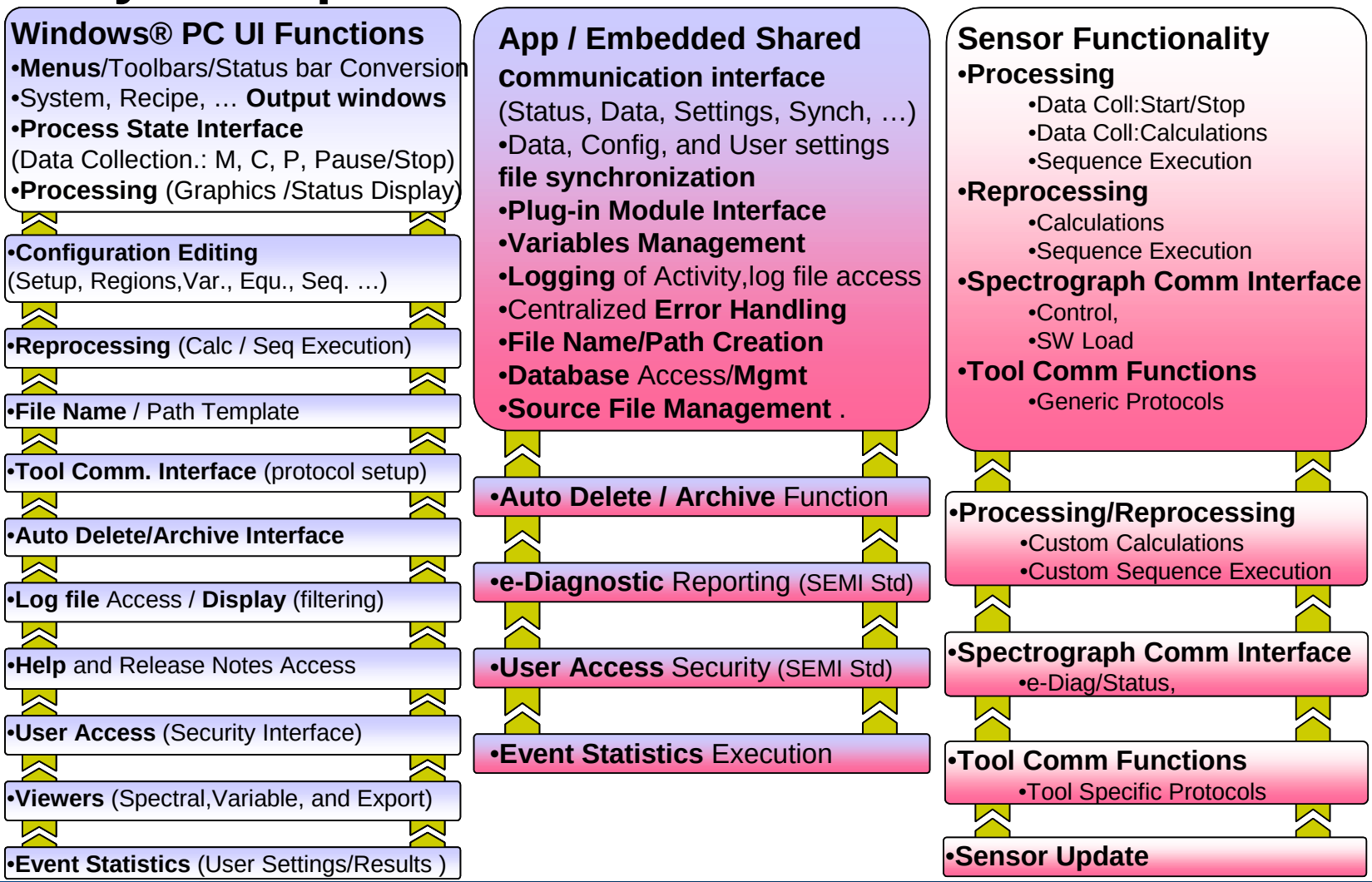
Functional Separation

- **Component Foundation**



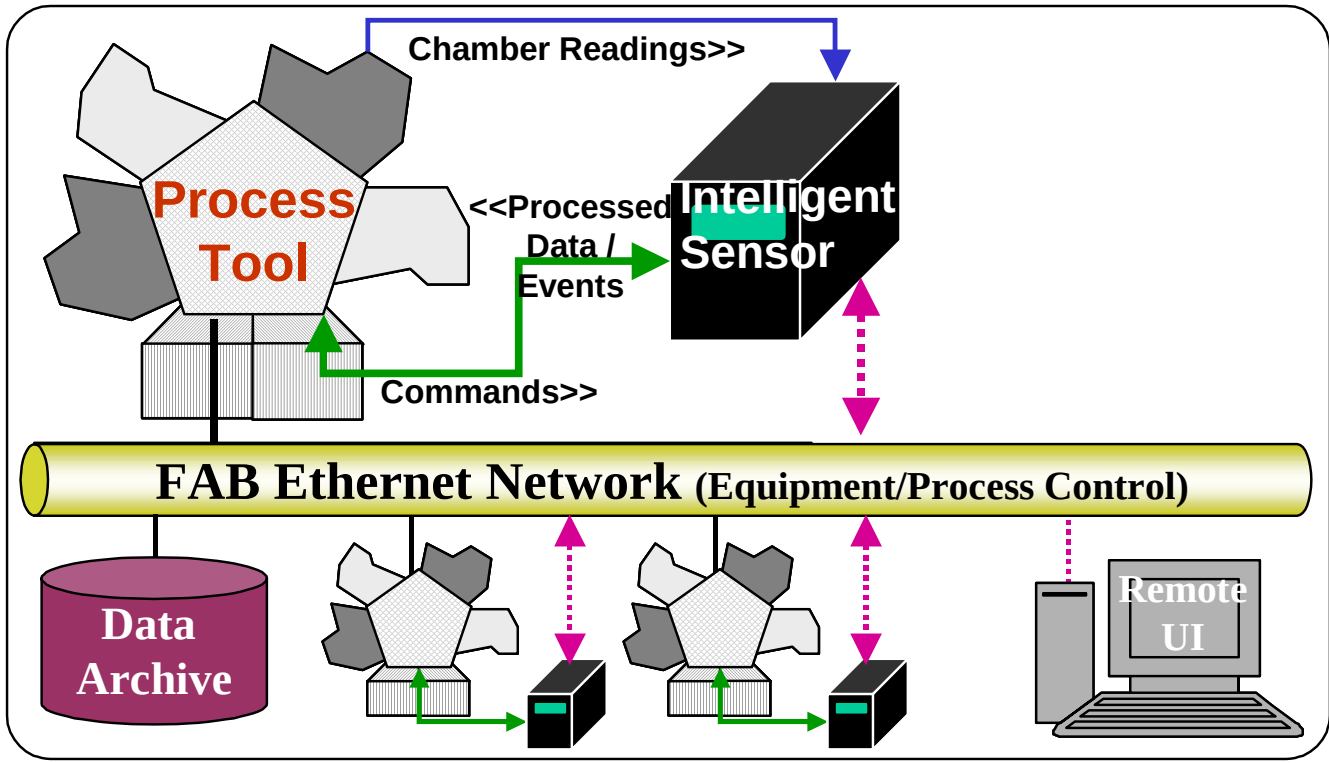
Modularization

- **Verity Example**



Post Migration / Connectivity

- **Permanent Connection - Green**
  - **Tool control to Sensor**
  - **Processed Data and Events from Sensor**
- **Temporary Connection - Purple**
  - Status Monitoring
  - Setup
  - Data download
  - SW Update



UI Operating Modes - On-Line / Off-Line

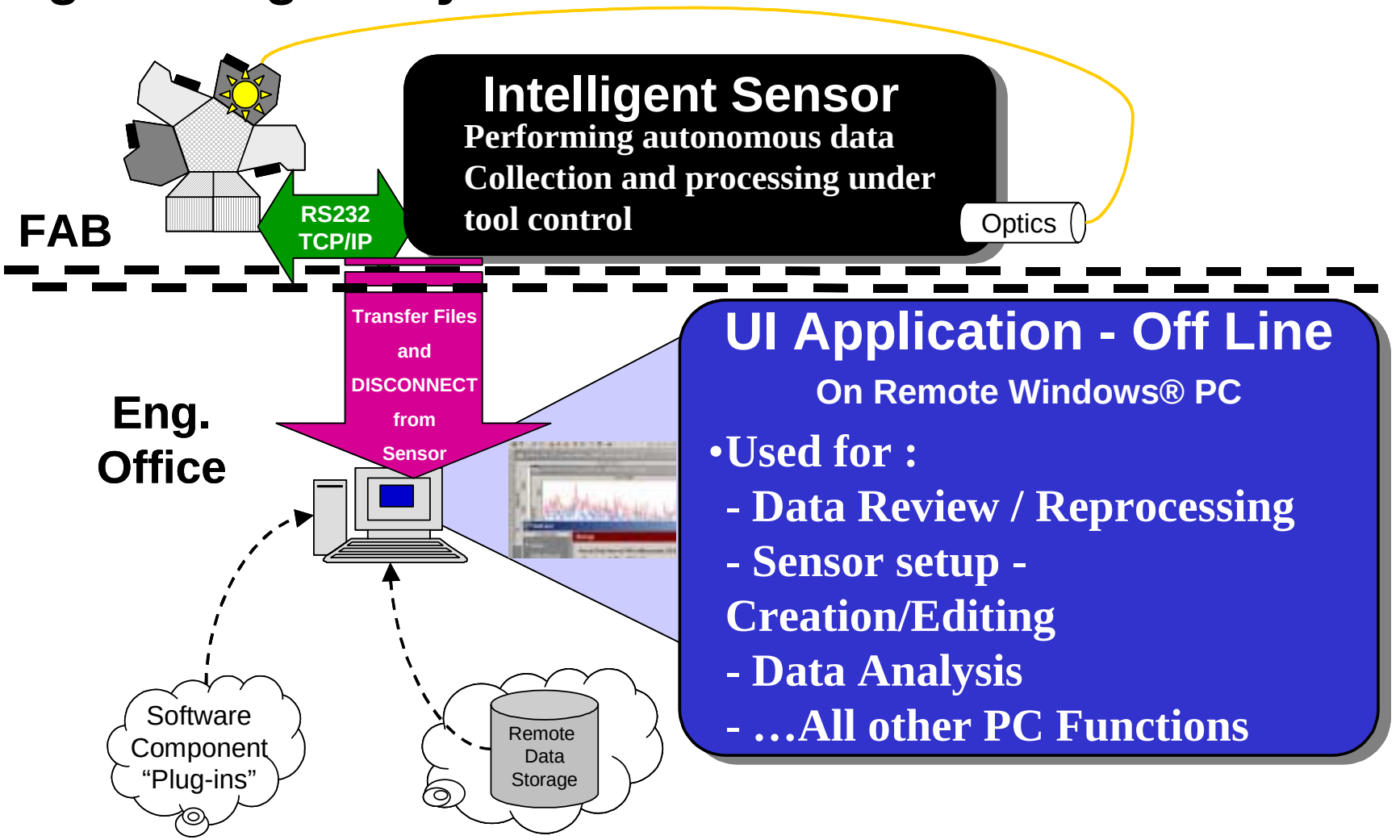
- **On-Line**
  - **The User Interface (UI and its Host PC) has a current network connection to an Intelligent Sensor.**
  - **When ON-Line, Data and other files are stored / referenced based on Sensor *Internal* Storage locations.**
- **Off-Line**
  - **The User Interface (UI and its Host PC) has NO connection to any sensor.**
  - **When OFF-Line, Data and other files are stored / referenced based on either:**
    1. *User settings, or*
    2. *The installation directory of the Hosting PC*.

UI On-Line

- **Real-Time (Deterministic)**
  - Applies ONLY to the Operating System and Embedded Software processing data.
  - Does NOT apply to the User Interface functions
- **Live (Approaching Real-Time / Not Deterministic)**
  - UI Updates (Sensor graphs) and other Information MAY / MAY NOT be deterministic due to:
    A. Priority of data processing and Endpoint Detection;
    B. UI message transfer time; and
    C. The OS Hosting the User Interface (MS Windows®).
  - Thus: “Live” Data display update =
    - Real-Time + Latency(A) + Latency(B) + Latency(C)

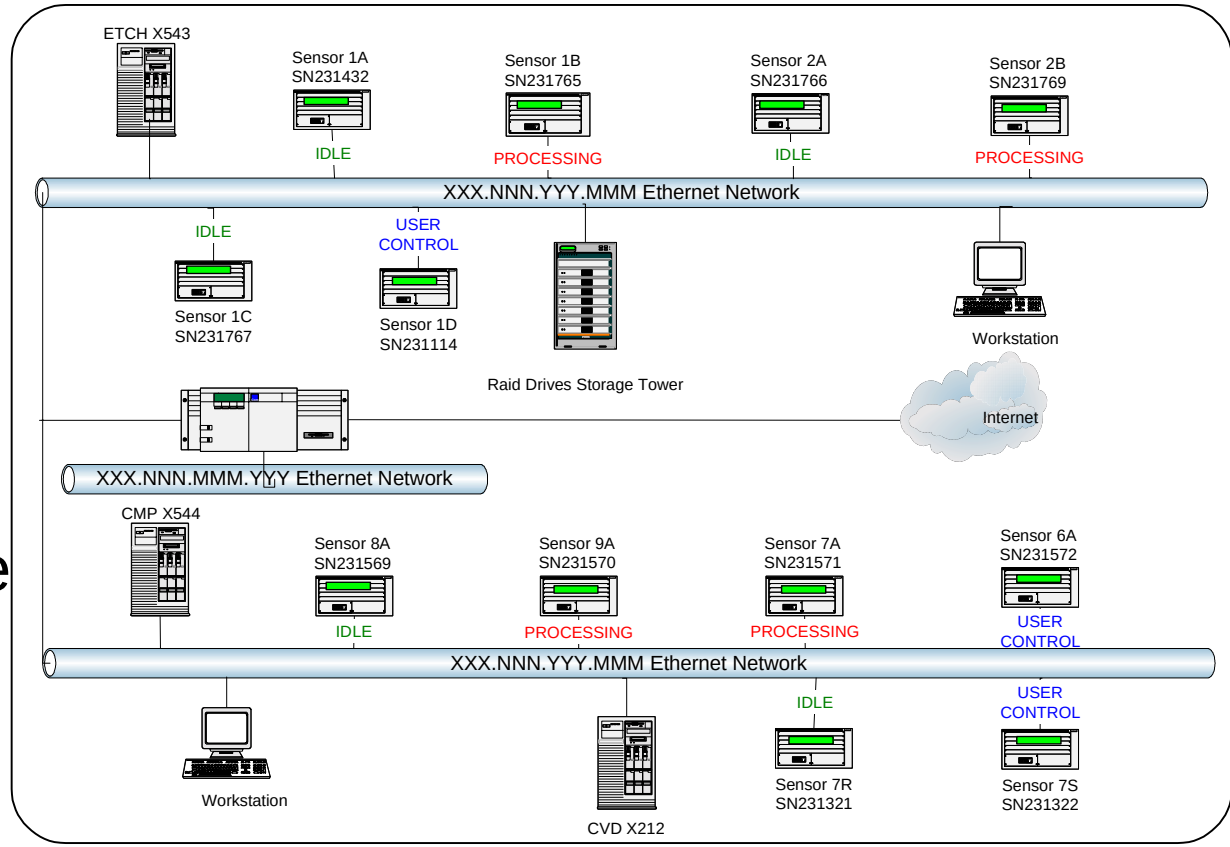
UI Off-Line

- **Engineering Analysis**



Administration

- **Sensor "Surveying" - Any Single Remote PC can be used for:**
  - **Viewing statistics and connection status for multiple sensors**
  - **Comparing**
    - SW Versions
    - Recipes
    - Error Status
  - **Cloning**
    - Group Update



Implementation and SEMI Standards

- **Use of Industry leading tools and frameworks**
  - **UI: Microsoft Visual Studio, Embedded Linux: eclipse**
  - **Open standard protocols : SOAP/XML and HTTP**
- **Linux**
  - **Adopt / customize kernel for RT performance**
- **Interface “A”**
  - **Access and Communication Security**
  - **Protocol standards for remote UI message passing**
- **Adopt appropriate E54 Sensor standards**
  - **Tool and FAB communication message passing**
- **Assess 3522,3510,3509,3563 readiness /applicability**
  - **E-diagnostics support for “Surveyor” function**

Challenges

- **Retention of all the existing features and functions.**
  - **As performed by the PC while receiving readings from the sensor during process tool operation**
    - Example: reprocessing of collected data and analysis must produce consistent multi-platform results both on-line and off-line.
- **Management of multiple remote connections by a single UI PC.**
  - **Network Interrogation and Sensor detection.**
- **Managing the “Live” UI update rate.**
  - **Keeping the rate reasonable for users while not impacting real-time processing.**
- **Function Module update in the field.**

Conclusions / Benefits

- **PC Elimination and Function Migration**
  - **Reduces cost for hardware and maintenance**
  - **Improves system reliability and integrity by reducing number of wired communication paths**
- **Use of customized real-time OS (Linux)**
  - **True real-time performance and determinism**
  - **Hacking and Virus attacks reduced or eliminated**
- **Modular Software Approach**
  - **Increases customization options**
  - **Isolates functional failures for faster diagnosis & repair**
  - **Allows sharing of code between platforms**
  - **Improves performance**

Recognition

- **The following have provided valuable contributions to this presentation:**
  - **Tim Michals**
  - **Steve Hartmann**