

---

# Agent Services

**Alex Kirn**

**akirn@eos.hitc.com**

---

**16 April 1996**

# Agent Services Outline



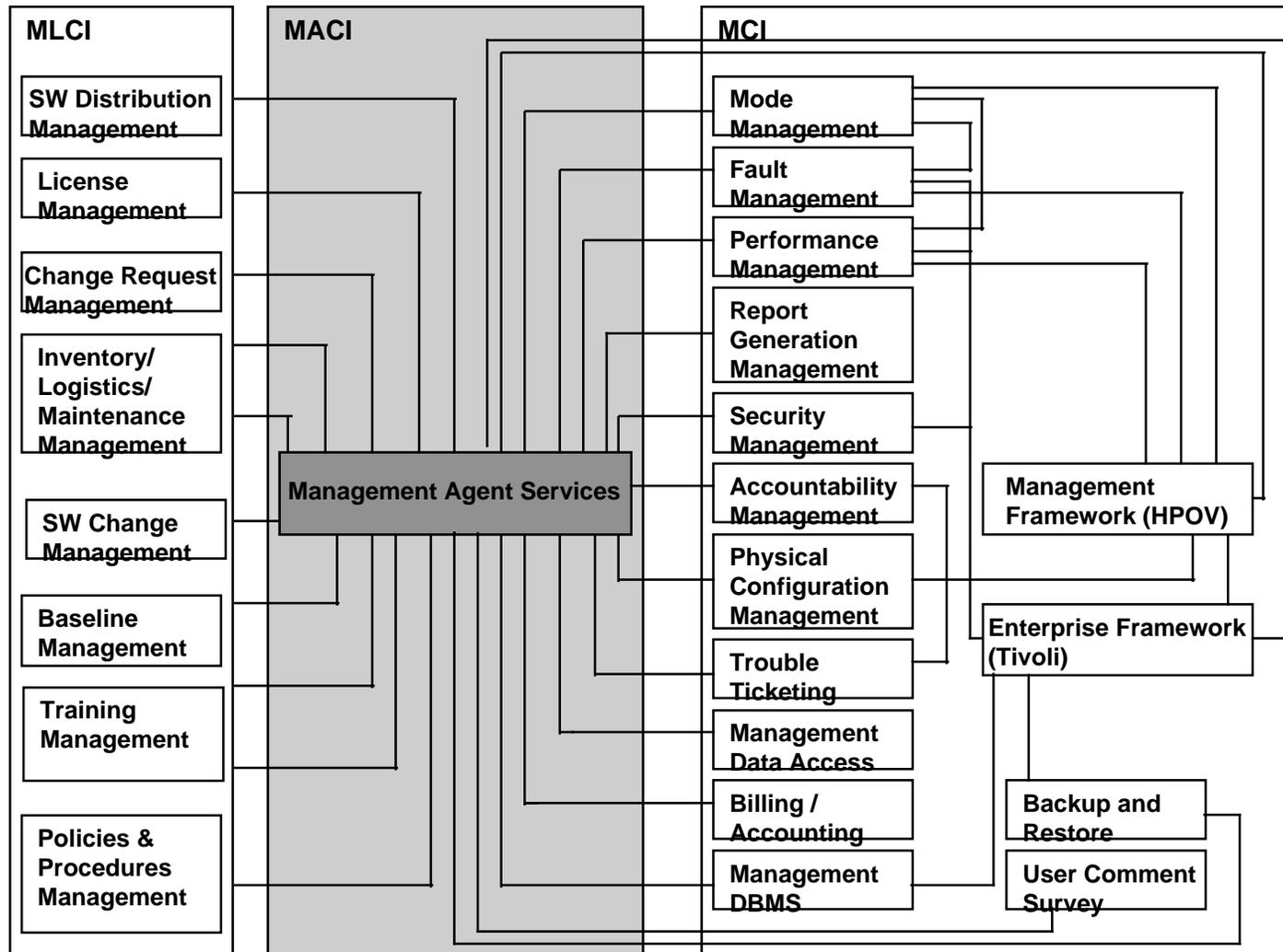
- Driving Requirements
- Agent Release B Enhancements
- Software Design
- Physical Design
- Object Model
- Event Trace Diagrams
  - Agent Startup
  - Suspend Execution

# Driving Requirements



- **Agent Services shall provide an interface to incorporate Life-Cycle services**
- **Agent Services shall have the capability to identify and distinguish managed applications based on their mode**
- **Agent Services shall incorporate the mode into all metrics collected**

# Agent Services Software Architecture Overview



# Release B Enhancements



- **Mode Management Capability**

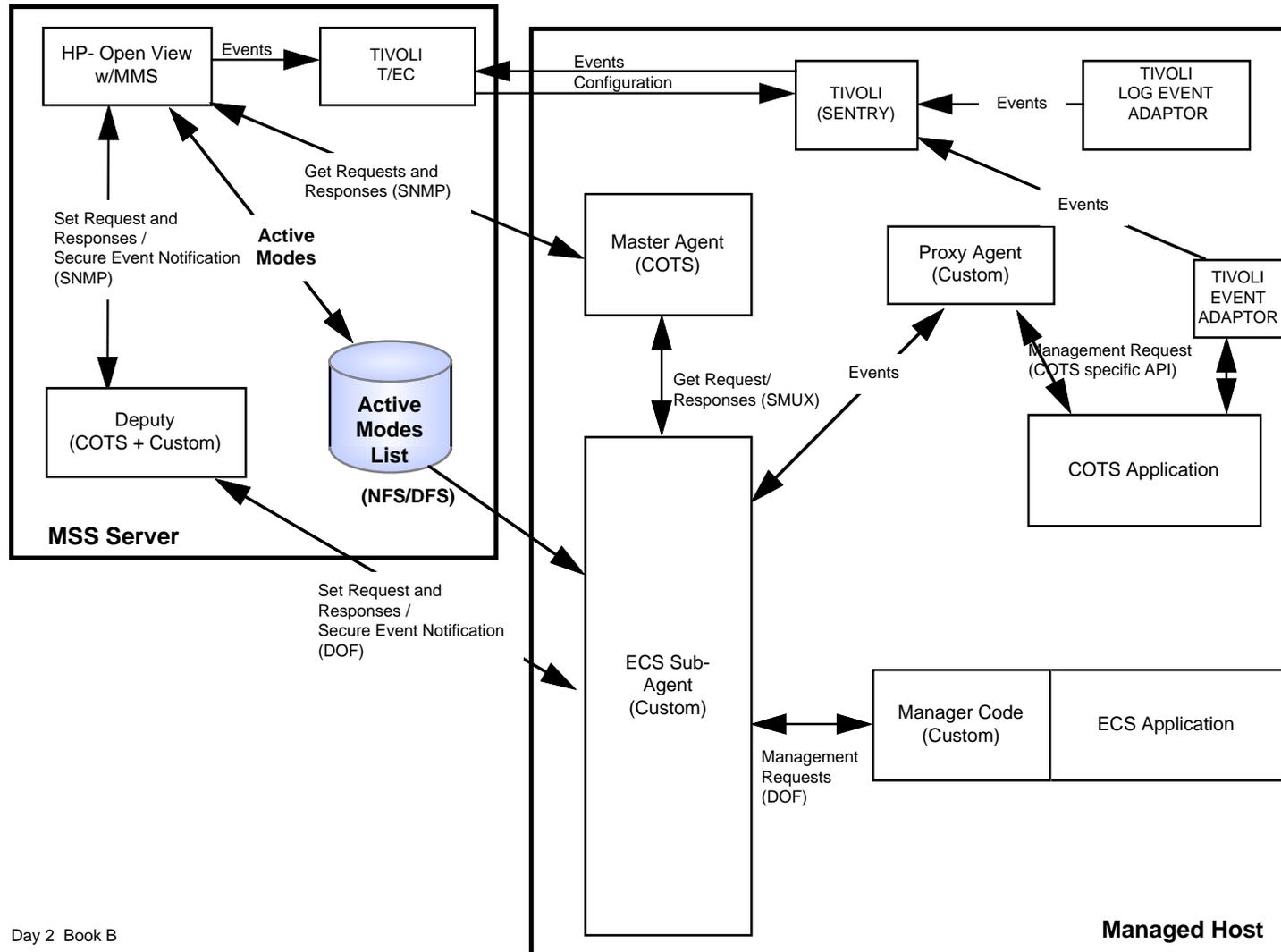
- Enable Agents to Discover mode specific applications
- Enable Mode distinction of metrics collected
- The Subagent is mode independent and will register under the “shared” branch in the CDS
- Each application will know its mode ( the mode is passed in as a command line argument and is maintained in the process framework)
- Mode is an attribute in the MIB, and can be accessed via a get operation
- MsAgTeMdMg is an abstract class from which all mode managed table entries are inherited

# Release B Enhancements (cont.)

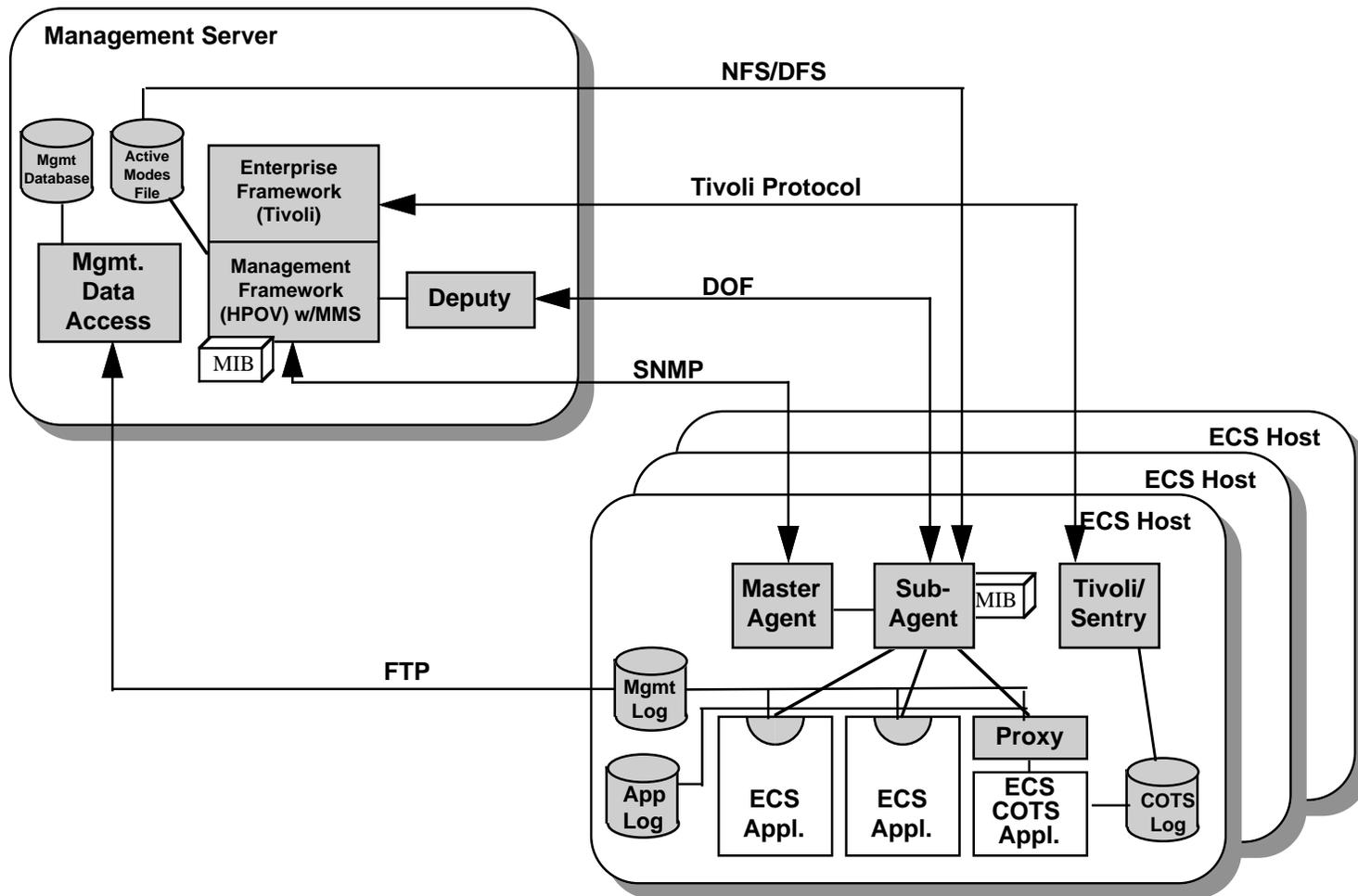


- **Enable Life-Cycle Services**
  - **Suspend**
    - **Stop Listening**
    - **Unregister the object with GSO**
    - **Unexport the object**
    - **Save binding information**
    - **Save the current status**
    - **Stop the current process**
  - **Resume**
    - **Start the selected process**
    - **Restore the status**
    - **Get binding information**
    - **Export the object**
    - **Register the object with GSO**
    - **Begin listening**
  - **Startup**
    - **Add mode specific startup and Simulated Time Capability**

# Agent Services Data Flow Overview



# Agent Services Physical Design



# Object Model



- The following object models will be reviewed

<u>Model Name</u>	<u>Diagram Name</u>	<u>Document Reference</u>
Agent Overview	MgtAgent_ObjectModel_GC	305 Vol29, Section 4.1.3
Agent Full View	MgtAgent_ObjectModel_Overview	305 Vol29, Section 4.1.3

# Affected Classes



- **Following classes are modified to incorporate Release B enhancements**
  - **EcAgManager**
  - **EcAgProxy**
  - **MsAgRegistry**
  - **MsAgDiscoverer**
  - **MsAgDeputyGate**
- **MsAgDiscoverer** : **Discovers installed applications/programs by looking for the application's configuration file in the cfg directory.**
- **MsAgRegistry** : **Discovers running processes**
- **The MsAgDiscoverer class includes the following two operations:**
  - DiscoverNow()** : **reports only changes in installed applications**
  - ReDiscover()** : **reports all installed applications**

# Affected Operations and Attributes



## Operations:

```
EcAgManager::Suspend( EcTAgMgmtLevel nLevel)
EcAgManager::Resume( EcTAgMgmtLevel nLevel)
EcAgManager::SuspendExec(pthread_addr_t)
EcAgManager::ResumeExec(pthread_addr_t)
MsAgDiscoverer::FindInstalled(RWOrdered &appVector, &
prgVector)
MsAgRegistry::Resume()
MsAgRegistry::Suspend()
MsAgRegistry::SuspendThreadExec()
MsAgRegistry::ResumeThreadExec()
MsAgRegistry::RestoreMgmtHandle()
MsAgRegistry::SaveMgmtHandle()
EcAgProxy::SuspendManager()
EcAgProxy::ResumeManager()
EcAgProxy::SaveSuspendedEntryVector()
MsAgDeputyGate::SuspendExecutable()
MsAgDeputyGate::ResumeExecutable()
```

## Attributes:

```
MsTAgSuspendParam
MsTAgResumeParam
```

## Files:

```
Active mode file (ASCII file - NFS/DFS mounted to manager host)
```

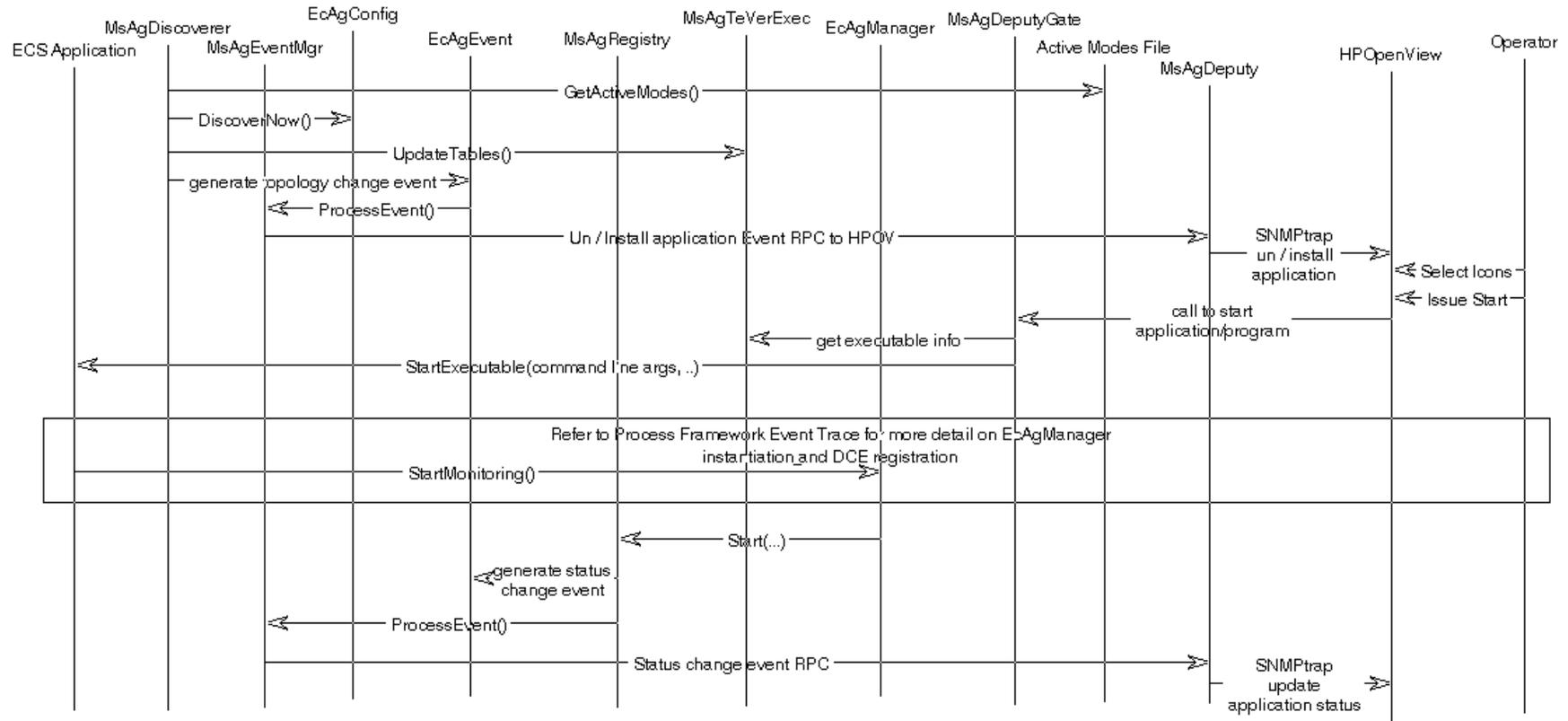
# Dynamic Model



- The following event trace will be reviewed

<u>Model Name</u>	<u>Diagram Name</u>	<u>Document Reference</u>
Agent Startup	MSS_Agent_Startup_EventTrace	contained within
Suspend Execution	MSS_Agent_Suspend_EventTrace	contained within

# Agent Startup Event Trace



# Agent Startup Event Trace Description



- **Givens:**
  - Every application and program has a corresponding .acfg and .pcfg configuration file in the configuration directory.
    - `../cfg/ops/`
    - `../cfg/ts1/`
    - `../cfg/ts2/`
    - ...
  - The subagent calls the DiscoverNow() operation upon startup. In addition, the DiscoverNow() operation is called periodically to detect changes in the installed applications. Both the DiscoverNow() and ReDiscover() operations can be manually invoked.
  - The subagent checks the applicable cfg subdirectories based on the “active modes” file.
  - The Agent Startup event trace starts after the system has been activated to recognize the desired modes via the MMS ActivateMode() operation. An event trace that addresses mode activation is detailed in the Mode Management Design Presentation.

# Agent Startup Event Trace Description (cont.)



- **Event Trace Overview**

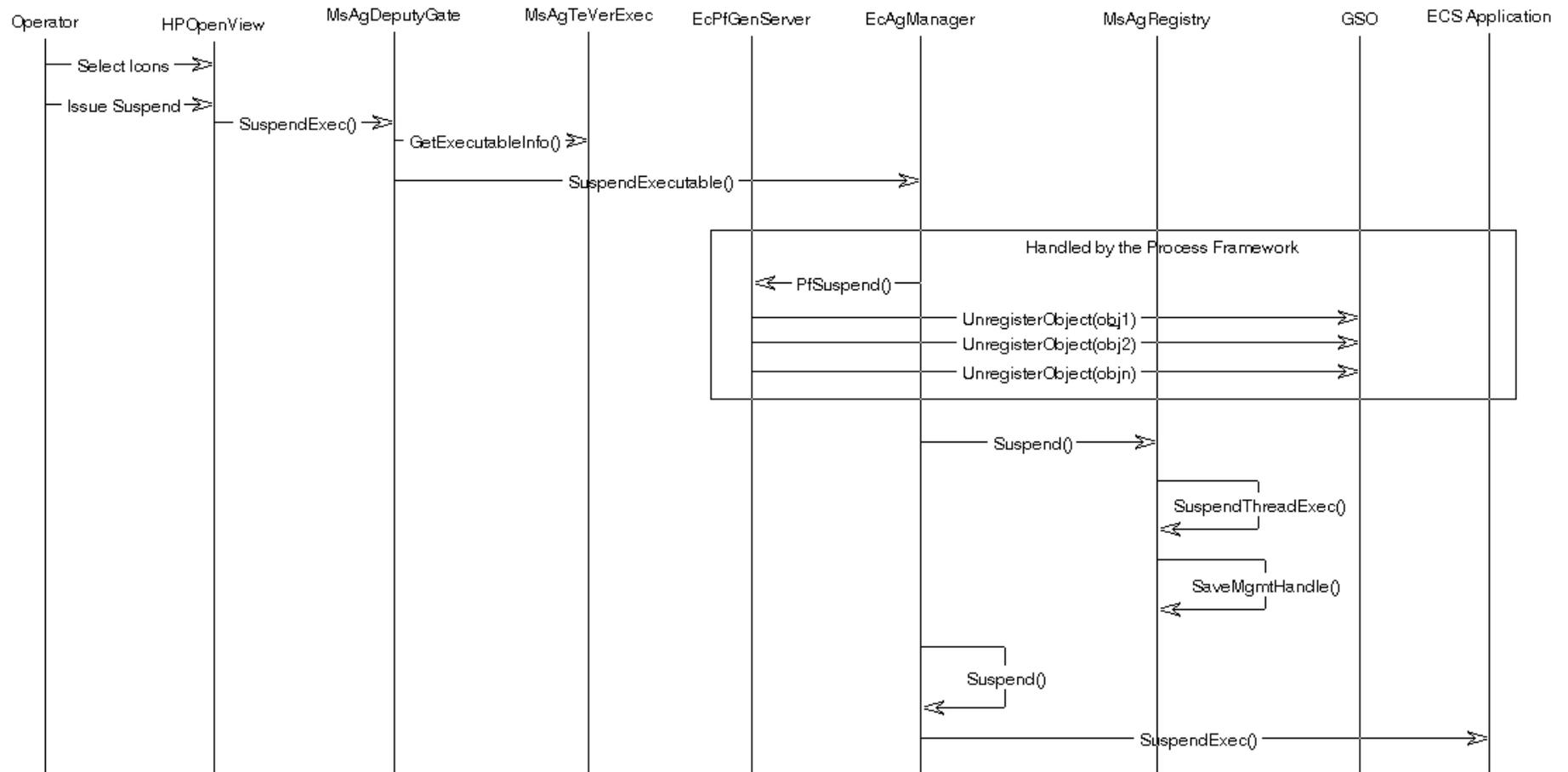
- **When the subAgent starts up it checks (via DiscoverNow() ) if the installed applications match the last known installed applications as listed in it's internal tables. These internal tables are represented by the MsAgTeStApp and MsAgStProg classes.**
- **If any changes are detected a topology change event is generated.**
- **An RPC is sent to HP OpenView to add/remove the application/ program from the applicable session(s) and submap(s).**
- **The Operator starts the desired subsystems/applications by highlighting the applicable icon(s) and selecting start. Since the icons (symbols), which are linked to specific applications/ programs (objects), are already tied to a given mode, the operator does not need to enter the mode identifier. The mode has been previously established.**

# Agent Startup Description (cont.)



- **Event Trace OverView (cont.)**
  - HP OpenView issues an **SNMPset()** to the **EcAgDeputyGateClient** to start the executables. The **EcAgDeputyGateClient** sends an **RPC** call to the **EcAgDeputyGate** who calls the **StartExecutable()** operation. (This process is repeated based on the number of icons selected.) The startup parameters passed are the **nTableID**, **nRowIndex**, **commandLine**, **nInstanceID**. The mode and, if applicable, the **deltaTime** is specified in the **commandLine**.
  - The application(s)/program(s) are started within the given mode.
    - The **EcAgManager** does the following
      - Registers the metrics
      - Calls **PfStart()** to start the **Process Framework (PF)** (the PF event trace is detailed in the **PFDesign Inspection**).
      - Call **StartMonitoring()**. This calls the **EcAgRegistry** class to check for running processes. When a process is detected an event is sent to **HPOV** to change the status of the icon to **executing**.

# Suspend Event Trace



# Suspend Event Trace Description



- The operator selects executable's symbol from within HPOV and issues a suspend command.
- The MMS application within HPOV will issue a SuspendExec() call to the MsAgDeputyGate.
- MsAgDeputyGate gets the executable's information from the SubAgent's internal tables via the MsAgTeVerExec class.
- MsAgDeputyGate issues a SuspendExecutable() call to the EcAgManager.
- The EcAgManager calls the Process Framework's EcPfManagedServer class PfSuspend() operation.
- The Process Framework handles the DCE associated functions and unregisters the associated objects from the GSO.

# Suspend Event Trace Description (cont.)



- **EcAgManger calls the MsAgRegistry classes Suspend() member function to tell the subagent to stop monitoring the process.**
- **The EcAgManager then issues a SuspendExec() call to the application. (Application processing state information will be save at the application level when it receives the Suspend call.)**