**<slide 1 - Introduction>**

In this presentation we describe what we set out to achieve in building our multi-platform system management system: “Sysgem Enterprise Manager” (SEM). We outline its architecture, the security considerations during its design and we give examples of its use with VMS systems.

First the objectives we set ourselves at the outset of the project…

**<slide 2 – Objectives - Delegation>**

Delegation! Have you ever known a system manager with time to spare? One of the primary objectives of Sysgem Enterprise Manager was to provide a secure system whereby routine tasks can be offloaded from busy, technical (and therefore expensive), system managers and assign those tasks to other (less-technically experienced) individuals in the organization such as help desk operators.

Since the primary users of SEM was to be help desk operators, user administrators, security managers, etc. we wanted the UI to be intuitive, non-technical and be available on users’ desktops. We chose a GUI interface on the Windows platform. [One may ask – why not a web-browser interface? And indeed that would have perhaps have been a better interface for general use, but actually we considered that it would have been very much more difficult to make it customizable – which was another of our objectives.]

**<slide 3 – Objectives - Authorization>**

In the SEM architecture, we created a central software component that we call the ‘SEM Authorization Server’, which is:

* the focal point for SEM security
* used for the authorization of SEM users
* used for the management of SEM software
* used for the signing of messages before transmission to SEM software on remote managed servers
* the default location for the audit trail and other SEM databases

**<slide 4 – Authorization of SEM Users (illustration)>**

Users log into SEM using software on their Windows workstation and have their SEM credentials validated by the Authorization Server. A SEM user account held at the Authorization Centre holds a profile (for permissions) that determines the level of access for individual users.

It is possible to set the level of access to a very fine degree of granularity. For instance:

* All help-desk menu options can be hidden or revealed according to the profile settings
* Access to nominated groups and/or users can be granted or denied
* Communication with certain target servers can be approved or disallowed

**<slide 4a – Software Components (illustration)>**

SEM software runs on:

* the users workstations (the ‘SEM Client’ component). This component provides all the graphical user interface to the user.
* the central Windows Server (the ‘SEM Authorization Server’ component). This component is the central secure location for validation of SEM user logins, a repository for user permissions, all script and UI libraries, Audit and other databases, SEM certificate authority.
* the target servers being managed (the ‘SEM Agent’ component) used as a ‘script processing engine’ – communicates with the SEM Client, receives and executes scripts sent by the client and returns output to it.

**<slide 5 – Objectives - Software Management>**

All scripts that run on target machines are held in one location – at the SEM Authorization Server. Updates to scripts and the customization of scripts are concentrated in this one central location.

Scripts appropriate to the transaction type are transmitted with each communication messages from the Authorization Server to the remote managed servers. So software upgrades are self-propagating from the central Authorization Server to all remote managed systems.

**<slide 6 – Objectives - *Message Signing>***

For security reasons, all transaction messages used in the SEM network are first ‘signed’ by the Authorization Server.

Unsigned or incorrectly signed messages will be rejected by recipient software components.

**<slide 7 – *Message Signing (illustration)>***

So for example a transaction to update a user account on one or more a target VMS servers is first sent from the SEM Client to the Authorization Server for ‘signing’.

The ‘signed’ message, together with the appropriate processing script is returned to the client and forwarded to the agents running on the target VMS servers. The input data plus the scripts required to process the transaction are all included in the signed ‘envelope’ to ensure that no interference to the input data or the scripts is possible.

**<slide 8 – Objectives – *SEM Databases>***

SEM databases such as the SEM Audit Trail Database can be located in any location as they are accessed via ‘Open Database Connectivity’ (ODBC) but by default they will be centralised at the Authorization Server and it is here that the software calls are made to write to the databases.

**<slide 9 – Objectives – *Secure Network>***

An installation option allows SSL/TLS to be used for network security or if a customer prefers, e.g. for legacy systems where SSL is not available or is not convenient to be used, an alternative communication security mechanism is made available by SEM

**<slide 10 & slide 10a – *Secure Network (illustration)>***

If SSL is being used then all communication connections between the SEM software components are transmitted via SSL.

**<slide 11 – Objectives – *Audit Trail>***An Audit database is maintained at the SEM Authorization Server that gives all information available to us regarding all Help Desk (user account admin) transactions. For instance it records the following:

* + Date and time of the transaction
	+ The identity of the SEM user account used to conduct the transaction
	+ The identity of the target server and the target object (e.g. a user account ID)
	+ The nature of the transaction (create/modify/delete/enable/disable/etc.)
	+ The details of the input submitted (except for passwords)
	+ If the transaction is an account modify transaction then the before and after states of the fields modified
	+ The identity of the workstation used by the SEM user
	+ The identity of the Windows user account (used by the SEM user) to log into Windows on the workstation

**<slide 12 – Objectives – *Target Platforms>***

Not only are VMS severs managed / monitored, but so too are Linux, UNIX, Windows, OS400.

**<slide 13 – Objectives – *Easily Customized>***

An important objective for the project was to allow customers to view and if required to modify all scripts that run on target servers and all input forms - to give a completely customised view or set of actions.

In this respect, the entire system was to have an ‘open’ architecture and, importantly, procedures put into place so that any customer changes introduced into the system can be carried forward and preserved into the next version should the standard UI and scripts, from Sysgem, be modified between releases. This protects the customers’ investment in making those changes to the product.