**<slide 1 - Introduction>**

Good afternoon, I am Shoshana Huss from Sysgem AG. Sysgem is a Swiss-based software development company and I work in marketing in our German office.

The subject of our presentation this afternoon is “Securely Managing VMS from a Windows Environment”. During this presentation I will describe the objectives that we set out to achieve when we first started to design our multi-platform product for System Management.

**<slide 2 – Sysgem Enterprise Manager**

We named the product: “Sysgem Enterprise Manager” and we abbreviate it to “SEM”.

Later, my colleague Mike Schofield from technical support in our UK office will demonstrate some of the topics we have discussed. We will outline the SEM architecture, the security considerations we made during its design and we will give examples of its use with VMS systems.

The first part of the presentation will last around 10 minutes. The second part will last about 30 minutes and then we’ll be happy to take any questions.

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

**<slide 3 – 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 were 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. Since Windows is the prevalent operating system for workstations in a commercial environment, we chose a GUI interface on the Windows platform.

**<slide 4 – Objectives - Authorization>**

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.
* It is used for the authorization of SEM users
* and for the management of SEM software.
* It is also used for the signing of messages before transmission to SEM software on remote managed servers
* and it is the default location for the audit trail and other SEM databases.

**<slide 5 – 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 Server 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 5a – Software Components (illustration)>**

Turning our attention to the software components that run in these locations:

* At the users’ workstations, we have the ‘SEM Client’ component. This component provides all the graphical user interface to the user.
* At the central Windows Server, we have 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
	+ and the SEM certificate authority.
* At each target server being managed, we have the ‘SEM Agent’ component.

It is kind of a ‘script processing engine’: it communicates with the SEM Client, it receives and executes scripts sent by the client and it returns output to the client.

**<slide 6 – Objectives - Software Management>**

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 every communication message 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 7 – Objectives - *Message Signing>***

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 8 – *Message Signing (illustration)>***

So for example, a transaction to update a user account on one or more 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 servers. The input data plus the scripts required to process the transaction are all signed to ensure that no interference is possible.

**<slide 9 – Objectives – *SEM Databases>***

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

**<slide 10 – Objectives – Secure Network Protocol>**

Secure Network Protocol.

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

**<slide 11 & slide 11a – Secure Network (illustration)>**

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

**<slide 12 – Objectives – *Audit Trail>***

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:

* + The date and time of the transaction
	+ The identity of the SEM user who conducted the transaction
	+ The identity of the target server and the target object (e.g. a user account ID)
	+ The nature of the transaction, such as:
	+ create
	+ modify
	+ delete
	+ enable
	+ disable/etc.
	+ The details of the input submitted, except for passwords, of course. We never record typed passwords
	+ If the transaction is an *account modify transaction*, then the before and after states of the fields modified are shown. We show the content of the field before it was modified and we show the content of the field after it was modified.
	+ The identity of the Windows user account (used by the SEM user) to log into Windows on the workstation
	+ and the identity of the workstation used by the SEM user

SEM provides an audit browser which can be used to search for specific auditable events and export audit records into a number of different formats, including HTML, Word documents, Excel spreadsheets, etc.

In fact, any report exported from SEM can be saved in multiple formats.

**<slide 13 – Objectives – *Target Platforms>***

Target Platforms.

Not only are VMS servers managed / monitored, but so too are

* Linux,
* UNIX,
* Windows,
* and OS400.

Since this boot camp is an **OpenVMS** boot camp, our demonstration will predominantly use the VMS platform.

If someone wants to see the product in action on one of the other platforms, or indeed any other feature that we do not have time to show, please come and see us at our stand.

**<slide 14 – Objectives – *Easily Customized>***

Finally, customization.

An important objective for the project was to allow customers to view and, if required, to modify all scripts that run on target servers.

Also, to be able to change all input forms in the user interface, allowing a completely customized view of the product and of the actions it takes.

Accordingly, the objective was for the entire system to have an

**‘open’ architecture**.

It was also an important objective to allow customers to write their own applications that run within the SEM framework. SEM provides the tools for building those applications by interfacing to the SEM framework.

Equally important was to protect a customer’s investment after he made changes. SEM is allowing any changes, that are made, to be automatically carried forward between versions, when software from Sysgem is upgraded.

Mike will demonstrate these techniques later.

**<slide 15 – Sysgem Enterprise Manager (SEM) *>***

Our ability to delegate routine tasks away from the overworked and expensive System Manager to the Help Desk saves money on man power resources. Additionally, the product is easily maintained using the centralized architecture of SEM. In most cases, an upgrade of the Sysgem software does not require an upgrade to the multiple distributed agents – only the central servers.

If anyone has any questions or needs further information after this session, we have a sponsor’s table open at Ashwood Court just outside this conference room.

I will now hand over to Mike Schofield who will demonstrate some of the features of Sysgem Enterprise Manager.