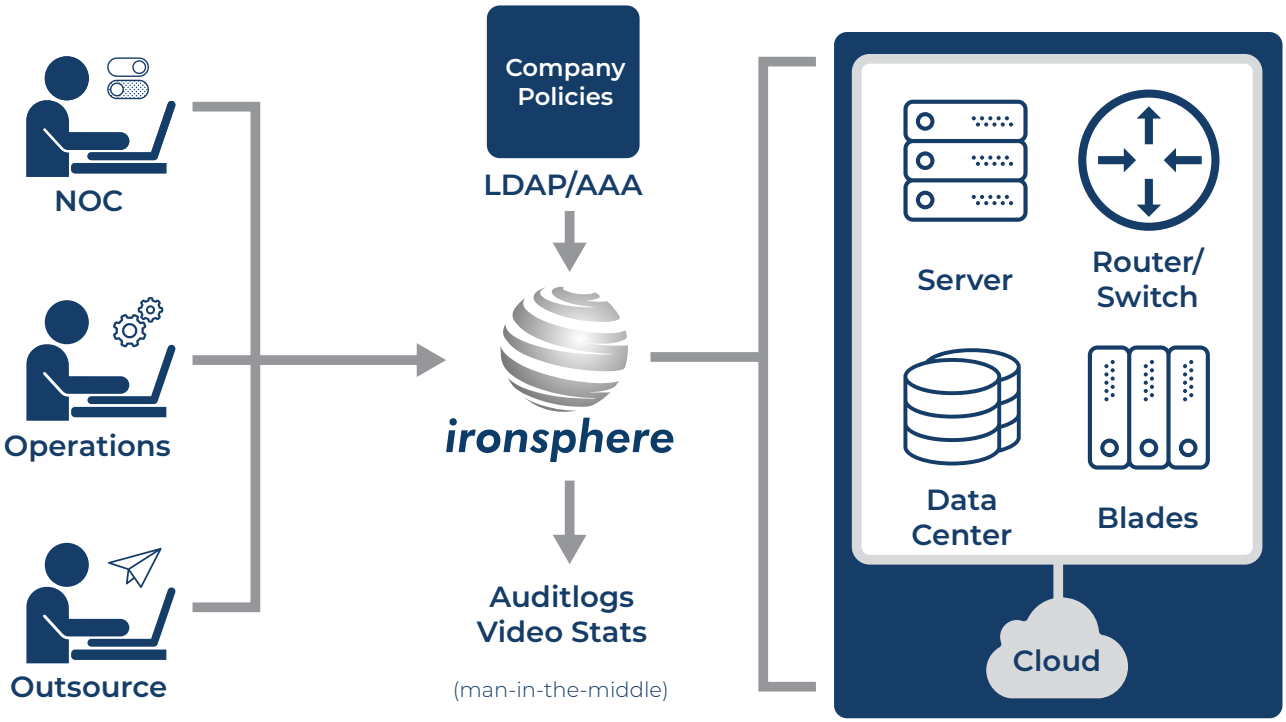
Session Manager

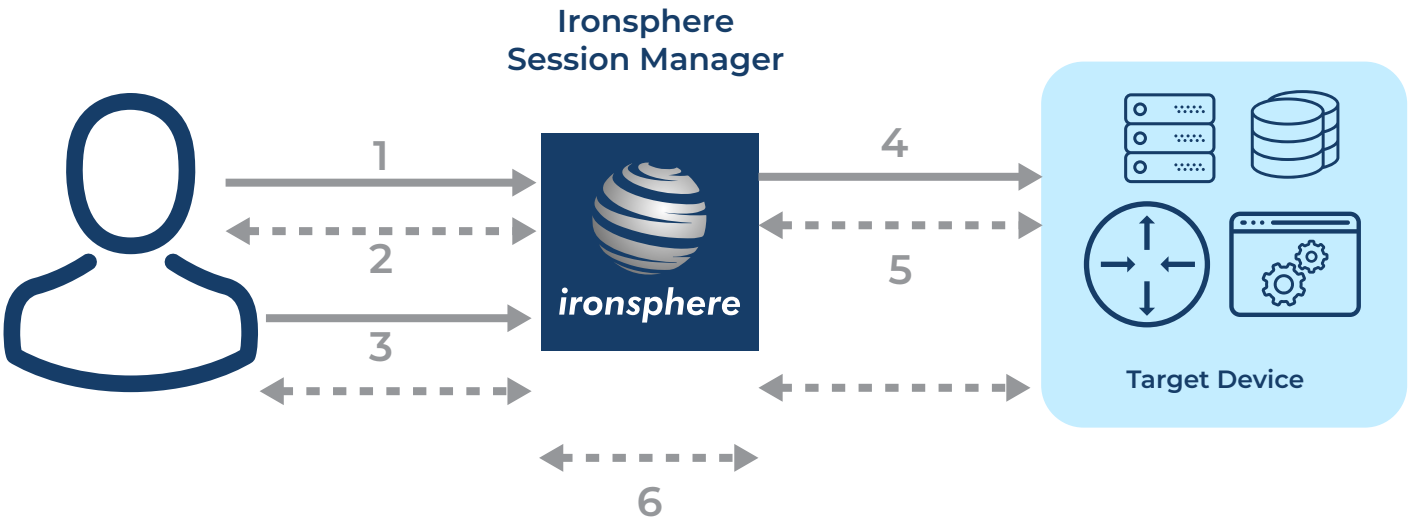
In any given IT and Network Infrastructure, there are thousands of servers/devices and thousands of users (employees, contractors, etc.) who connect to them on a daily basis. So, there are tens of thousands of connections established between users and servers/devices every day, which is very complex and probably unmanageable from a security point of view. Not every connection (between user and device/servers) is at the same level of importance. For instance, there are users (employees) which you trust to connect to servers/devices that do not manage critical enterprise assets. For such connection types it may be enough to just log which user connected to which device/server. However, some connections are extremely sensitive, like 3rd party technical support workers who access the most critical network/IT resources; in such cases, you want to ensure that you have "full visibility" and "full control" during such connections. Ironsphere's Session Manager is the solution.



Ironsphere's Session Manager transparently sits in the middle of sessions and does not require an agent to be installed on User PCs or target servers/applications.

Session Manager supports Command Line Interfaces (SSH, Telnet), Remote Desktop Connections (RDP/VNC), Web sessions (HTTP/S), File transfer (SFTP), Database connections (SQL).

How Session Manager Works



Step 1: User initiates a CLI session towards Ironsphere with his/her own username and password.

Step 2: A CLI session between User and Ironsphere is established. Ironsphere displays the list of devices that the user has permission to access.

Step 3: User selects the Target Device he/she wants to connect to from the list.

Step 4: Ironsphere initiates a CLI session towards the Target Device with a username/password.

Step 5: A CLI session is established between Ironsphere and the Target Device.

Step 6: Two separate CLI sessions (User<->Ironsphere and Ironsphere<->Target Device) are connected back-to-back by the Session Manager. Ironsphere is the man-in-the-middle for the entire duration of the session and has "full control" and "full visibility" of the session. When the User enters a command on his/her CLI screen, Ironsphere receives it, "processes" it and decides whether to forward the command to the Target Device or reject it.

Features & Benefits

- Full visibility. Detailed audit logs. All commands, either failed or successful, are logged. Indisputable logging of which user attempted to run which command on which device and when.
- "Separation of duties" and "least privilege" practices are achieved, regardless of the role/profile capabilities of the Target Device. Any custom policies (allowed command sets, blocked command sets) can be defined and applied to any user group, ensuring that only the "required set of commands" can be executed by a user in order to fulfill his tasks, restricting standard user accounts from having over-privileged access.
- Context-aware policy. For example, do not allow "delete" command to run at the device level (higher/outer level of the command tree), but allow it to run at the port level (lower/inner level of the command tree).
- 2-factor authorization. It is possible to define that certain commands (e.g. shutdown command) require an approval from a second person to run, i.e., when a user enters a "shutdown" command, his supervisor receives an email, and if he/she clicks the "approve" link then the command is executed, otherwise it is rejected.
- Eliminates weak passwords and/or non-expiry passwords.
- Enables the definition of time-based access limitations, based on time of day, day of the week, maintenance window hours, etc.
- Disables inactive privileged accounts.
- Session recording and video playback for forensic analysis.
- Dual control (referred to as "four eyes" or "second eye"). When a user is connected to a device, a supervisor can monitor the session in real-time and can also take/release the control of the session. This is particularly useful when real time monitoring is required for emergency accounts or to monitor someone who is in training.
- Single sign-on. The user connects to Ironsphere with his/her username/password and selects any allowed device to connect. The user does not need to use/know separate username/passwords to connect to different devices/servers.
- Helps to eliminate password sharing and shared account usage. Users always log in with their own username/password, even if a shared account is used to connect to the device. For example, I connect to Ironsphere with username=Frank and then select a device to connect. Ironsphere establishes a session towards the target device, but may be using username=admin. As a user, I never see/know the real username/password used to connect to the target device, all I know is my own username/password.
- Makes sure it is the real user connecting to the target device, indisputably.
- Auto lock user account when an employee terminates employment (integration with enterprise Active Directory or LDAP is required).