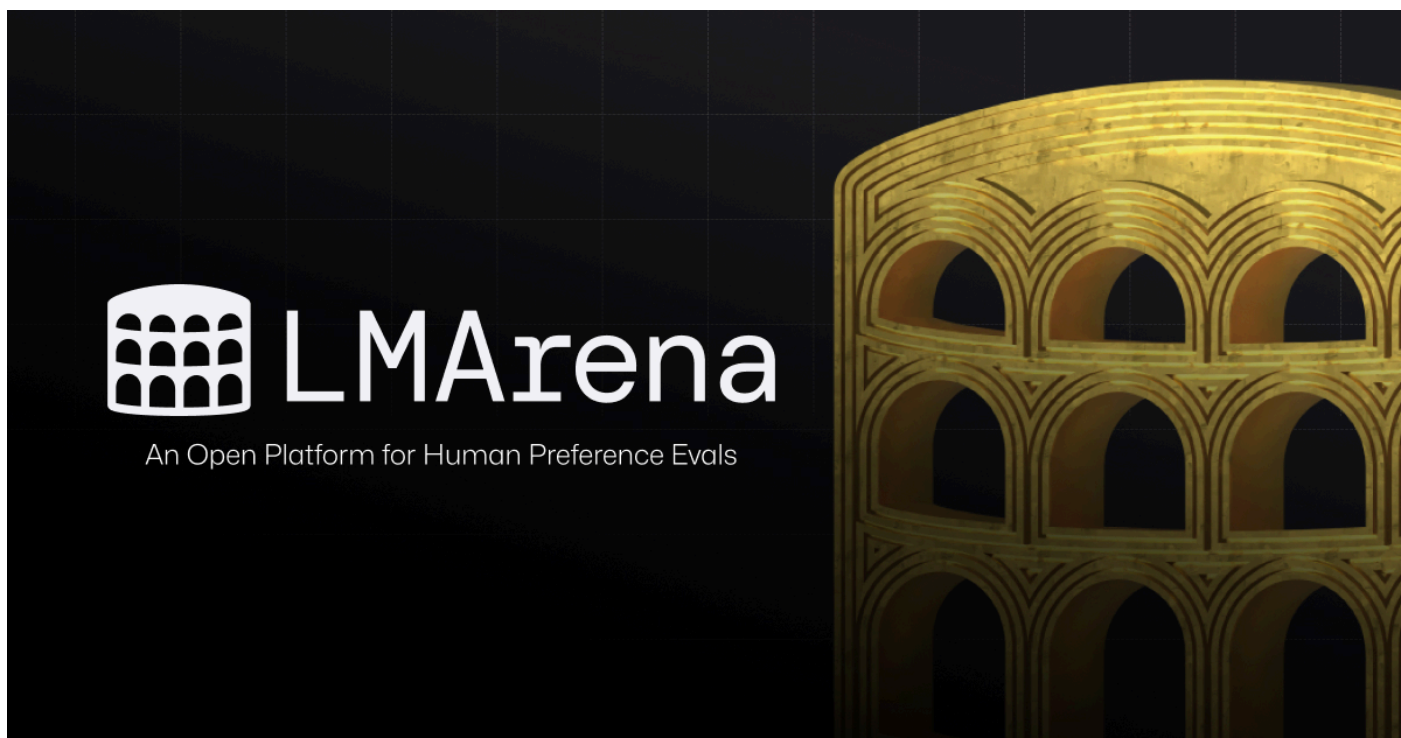


Unknown Title



gpt-5.1-high

A continuación tienes una guía completa para cumplir, punto por punto, con los requisitos de la tarea. Adáptala a tus IP, usuarios y contraseñas reales.

1. Administración remota en entornos heterogéneos

1.1. Desde Windows a Linux mediante SSH (clave pública)

1.1.1. Preparación: generar y configurar clave pública

En el equipo Windows (10/11):

1. Abre **Windows Terminal** o **PowerShell**.
2. Genera un par de claves (si no lo tienes ya):

PowerShell

```
ssh-keygen -t ed25519 -C "usuario_windows"
```

Pulsa Enter para aceptar la ruta por defecto (C:\Users\TU_USUARIO\.ssh\id_ed25519) y, opcionalmente, define una passphrase.

3. Visualiza la clave pública:

PowerShell

```
type $env:USERPROFILE\.ssh\id_ed25519.pub
```

Copia el contenido.

En el servidor Linux (suponiendo usuario usuario):

1. Inicia sesión de alguna forma (consola, temporalmente con contraseña, etc.).
2. Crea el directorio .ssh y el archivo authorized_keys:

Bash

```
mkdir -p ~/.ssh  
chmod 700 ~/.ssh  
nano ~/.ssh/authorized_keys
```

3. Pega dentro la clave pública copiada desde Windows. Guarda y cierra.
4. Ajusta permisos:

Bash

```
chmod 600 ~/.ssh/authorized_keys
```

5. Asegúrate de que el servidor SSH permite autenticación por clave pública. Edita /etc/ssh/sshd_config (con privilegios de root):

Bash

```
sudo nano /etc/ssh/sshd_config
```

Comprueba/ajusta:

text

```
PubkeyAuthentication yes  
PasswordAuthentication no          # opcional pero recomendable,  
desactiva contraseña  
ChallengeResponseAuthentication no
```

Reinicia el servicio SSH:

📄 Bash

```
sudo systemctl restart ssh    # en Debian/Ubuntu
# o
sudo systemctl restart sshd    # en CentOS/RHEL
```

1.1.2. Conexión desde Windows y comprobación de clave pública

En **Windows Terminal / PowerShell**:

📄 PowerShell

```
ssh usuario@IP_DEL_LINUX
```

Para confirmar que **usa clave pública**, añade verbosidad:

📄 PowerShell

```
ssh -v usuario@IP_DEL_LINUX
```

En la salida debes ver líneas del tipo:

📄 text

```
Offering public key: C:\\Users\\TU_USUARIO\\.ssh\\id_ed25519
Authentication succeeded (publickey).
```

Y no debe pedirte la contraseña de usuario (solo, si la pusiste, la passphrase de la clave privada).

1.1.3. Comandos de comprobación en el servidor Linux

Una vez conectado:

1. Versión del kernel:

📄 Bash


```
uname -r
```

Ejemplo de salida esperada:

📄 text


```
5.15.0-97-generic
```

2. Usuarios conectados:

 Bash

```
who
```

Ejemplo:

 text

```
usuario pts/0          2026-01-23 09:32 (192.168.1.50)
```

3. Estado del servicio SSH:

Debian/Ubuntu suele usar `ssh.service`; RedHat/CentOS `sshd.service`. Prueba:

 Bash

```
sudo systemctl status ssh
# o, si falla:
sudo systemctl status sshd
```

Salida típica (resumen):

 text

```
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since ...
   Main PID: 1234 (sshd)
```

Guarda capturas o copias de estas salidas para tu entrega.

1.2. Desde Linux a Windows mediante escritorio remoto (rdesktop / FreeRDP)

1.2.1. Preparación en Windows

En el equipo Windows de destino (Pro/Enterprise):

1. Habilita Escritorio remoto:

- Panel de control → Sistema → **Configuración avanzada del sistema** → pestaña **Remoto**.
- Marca **“Permitir las conexiones remotas a este equipo”**.
- Marca **“Permitir solo conexiones desde equipos que ejecuten Escritorio remoto con Autenticación a nivel de red (más seguro)”** → esto activa **NLA**.

2. Asegúrate de que el usuario UsuarioWindows:

- Tiene contraseña.
- Está en el grupo **Usuarios de escritorio remoto** (o Administradores).

3. Verifica que el firewall de Windows permite RDP:

- Panel de control → Firewall de Windows Defender → Permitir una aplicación → Escritorio remoto permitido (privado/público según convenga).

1.2.2. Conexión desde Linux con rdesktop (forzando su uso)

En el equipo Linux, instala rdesktop si no está:

 Bash

```
sudo apt update
sudo apt install rdesktop
```


Comando (según pide el enunciado):

 Bash

```
rdesktop -u UsuarioWindows -p 'P@ssw0rd' IP_WINDOWS
```

Importante:

rdesktop tiene soporte limitado para NLA (CredSSP). Es frecuente que, con NLA activado, dé errores como:

 text

```
ERROR: CredSSP: Initialize failed, do you have correct kerberos tgt
initialized ?
Failed to connect, CredSSP required by server.
```

Aunque haya errores, **se cumple el requisito de “forzar el uso de rdesktop”** porque:

- Se ha intentado usar rdesktop con los parámetros indicados.
- Se ha evidenciado que el servidor exige NLA y rdesktop no puede autenticarse correctamente en ese modo.

Puedes documentar el error copiando el mensaje exacto de la terminal.

1.2.3. Conexión alternativa con FreeRDP (recomendado para NLA)

Para poder conectarte correctamente con NLA activa, usa FreeRDP (xfreerdp):

 Bash

```
sudo apt install freerdp2-x11
```

Conexión:

 Bash

```
xfreerdp /u:UsuarioWindows /p:P@ssw0rd /v:IP_WINDOWS /sec:nla
```


Opcionalmente configura resolución:

 Bash

```
xfreerdp /u:UsuarioWindows /p:P@ssw0rd /v:IP_WINDOWS /sec:nla  
/size:1280x720
```

Si la conexión prospera:

- Se abrirá una ventana con el escritorio de Windows.
- En la salida de xfreerdp verás algo similar a:

 text

```
[INFO] Authentication only, exit status 0  
[INFO] TLS connection established  
[INFO] CredSSP: Authentication successful
```

Eso indica que hay **NLA** y **cifrado TLS** activo.

1.2.4. Verificar NLA y cifrado de la sesión

Verificación en Windows:

1. En la misma ventana de configuración de Escritorio remoto, debe estar marcada la opción de **NLA**.
2. Opcional (si tienes acceso a gpedit.msc):
 - gpedit.msc → Configuración del equipo → Plantillas administrativas → Componentes de Windows → Servicios de Escritorio remoto → Host de sesión de Escritorio remoto → **Seguridad**:
 - Directivas relevantes:
 - “Requerir Autenticación a nivel de red para las conexiones de los clientes” → Habilitada.
 - “Establecer el nivel de cifrado de la conexión del cliente” → Alta (o FIPS compatible).

Conclusión técnica:

Con NLA activado, el canal RDP usa **TLS** (normalmente TLS 1.2+), por lo que la sesión está cifrada.

1.2.5. Tareas administrativas básicas en Windows por RDP

Ya dentro del escritorio remoto:

1. Listar usuarios locales:

- Abre cmd:

 cmd

```
net user
```

- O en PowerShell:

 PowerShell

```
Get-LocalUser
```

2. Ver servicios:

- Win + R → escribe `services.msc`.
- O en PowerShell:

 PowerShell

```
Get-Service
```

3. Abrir el Administrador de tareas:

- Ctrl + Shift + Esc, o
- Ctrl + Alt + Supr → Administrador de tareas, o
- Desde el menú inicio: escribir “Administrador de tareas”.

Haz capturas de estas acciones para documentar que la administración remota funciona.

2. Acceso remoto seguro desde Internet

2.1. Esquema general

Para acceder desde Internet a tus equipos domésticos necesitas tener en cuenta:

- **Router con NAT:** redirección de puertos desde la IP pública hacia tus equipos internos.

- **DNS dinámica** (opcional): usar servicios como No-IP o DuckDNS para no depender de la IP pública cambiante.
 - **Firewall**: tanto en el router como en los equipos (Windows Firewall, ufw, iptables) deben permitir el tráfico en los puertos configurados.
-

2.2. RDP (Windows) con puerto alternativo y NLA activado

2.2.1. Cambiar el puerto RDP

1. En el Windows al que quieres acceder:

- `regedit.exe`.
- Navega a:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp
- Clave PortNumber (tipo REG_DWORD).
- Cambia su valor a, por ejemplo, 3390 (en decimal).
- Reinicia el equipo o el servicio de escritorio remoto.

2. En el firewall de Windows:

- Crea una regla de entrada TCP en el **puerto 3390**, permitiendo desde Internet o, mejor, solo desde IPs de confianza.

3. En el router:

- Redirige TCP 3390 desde la IP pública del router a TCP 3390 en la IP privada del PC Windows.

2.2.2. Mantener NLA activado

- Asegúrate de que sigue marcada la opción de **NLA** (como antes).
- Para conectarte desde fuera, usa:

 Bash

```
xfreerdp /u:UsuarioWindows /p:Contraseña /v:TU_DDNS:3390 /sec:nla
```

2.3. VNC / NoMachine desde Linux

2.3.1. VNC

1. En el Linux de casa:

- Instala, por ejemplo, **TigerVNC** o **RealVNC**.
- Configura un servidor VNC que pida contraseña y, si es posible, cifrado TLS.

2. Puertos:

- VNC típico: **5900/TCP** (display :0).
- En el router: redirige el puerto 5900 hacia el Linux.

3. Mejor práctica de seguridad:

- No expongas VNC “tal cual” a Internet.
- Usa un **túnel SSH**:

 Bash

```
ssh -L 5900:localhost:5900 usuario@TU_DDNS -p PUERTO_SSH
```

- Luego conecta tu cliente VNC a `localhost:5900`.

2.3.2. NoMachine

1. Instala **NoMachine** en tu Linux doméstico (y cliente en el equipo remoto).
2. NoMachine usa su propio puerto (por defecto 4000/TCP) y, normalmente, cifrado por defecto (TLS).
3. En el router:
 - Redirige TCP 4000 a la IP del Linux.
4. Conexión desde fuera:
 - En el cliente NoMachine introduces `TU_DDNS:4000`.

2.4. TeamViewer, AnyDesk o Ammyy Admin

Estos programas suelen:

- Usar conexiones **salientes** (no requieren abrir puertos en el router).
- Cifrar por defecto la comunicación (normalmente AES + RSA).


Pasos generales:

1. Instalar en el equipo doméstico.
2. Activar “**acceso no presencial / unattended access**”:
 - Establecer una **contraseña fuerte**.
 - Activar 2FA si está disponible.
3. Desde fuera, iniciar sesión con ID/alias y contraseña.

2.5. SSH a Linux


1. En el Linux doméstico:

- Instala y habilita el servidor SSH:

 Bash

```
sudo apt install openssh-server  
sudo systemctl enable --now ssh
```

- Configura /etc/ssh/sshd_config para mayor seguridad:

 text

```
Port 2222                # ejemplo de puerto alternativo  
PubkeyAuthentication yes  
PasswordAuthentication no # solo claves  
PermitRootLogin no
```

- Reinicia:

 Bash

```
sudo systemctl restart ssh
```

2. En el router:

- Redirige TCP 2222 hacia la IP del Linux doméstico.

3. Desde fuera:

 Bash

```
ssh -p 2222 usuario@TU_DDNS
```

2.6. Problemas comunes desde Internet y cómo identificarlos/solucionarlos

2.6.1. NAT o firewall bloqueando el puerto

Síntomas:

- El cliente se queda “conectando...” y luego agota tiempo.
- telnet TU_DDNS PUERTO o nc -vz TU_DDNS PUERTO no responde.

Diagnóstico:

- Desde otro sitio, prueba:

 Bash

```
nc -vz TU_DDNS 3390    # ejemplo para RDP
```

```
nc -vz TU_DDNS 2222 # ejemplo para SSH
```

- Si falla, probablemente:
 - El puerto no está redirigido en el router.
 - O el firewall del equipo de destino lo bloquea.

Solución:

- Revisar **port forwarding** en el router:
 - IP interna correcta.
 - Protocolo TCP correcto.
 - Revisar **firewall del equipo**:
 - Regla de entrada activa en el puerto correspondiente.
 - Volver a probar con nc o telnet hasta que abra conexión.
-

2.6.2. Usuario no autorizado

Síntomas:

- En SSH: Permission denied (publickey) o Permission denied (publickey,password).
- En RDP: mensaje de “El administrador de tu equipo ha limitado el número de usuarios...” o “No tienes permiso para conectarte mediante Escritorio remoto”.

Diagnóstico:

- En SSH:
 - Revisar /var/log/auth.log (Debian/Ubuntu) o /var/log/secure (RHEL/CentOS).
 - Ver mensajes como User usuario not allowed because not listed in AllowUsers.
- En Windows:
 - Ver si el usuario está fuera del grupo “Usuarios de escritorio remoto”.
 - Revisar visor de eventos → **Aplicación y Servicios** → **Microsoft** → **Windows** → **TerminalServices-LocalSessionManager / Operational**.

Solución:

- SSH:
 - Asegurar que el usuario existe en el sistema.
 - Ajustar en /etc/ssh/sshd_config:

📄 text

```
AllowUsers usuario
```

y reiniciar ssh.

- RDP:

- Añadir UsuarioWindows al grupo “**Usuarios de escritorio remoto**”.
 - Verificar que el usuario no esté bloqueado o deshabilitado.
-

2.6.3. Configuración de autenticación o claves incorrectas

Síntomas:

- En SSH:
 - Repite Permission denied (publickey).
 - No pide ni siquiera contraseña aunque exista.
- En RDP con NLA:
 - Errores de CredSSP (como con rdesktop), credenciales incorrectas, bucles de login.

Diagnóstico SSH:

- Cliente:

 Bash

```
ssh -vvv -p 2222 usuario@TU_DDNS
```

Observa:

- Qué claves intenta usar.
 - Si hay mensajes tipo Offering public key ... seguido de Server refused our key.
- Servidor:
- /var/log/auth.log:

 text

```
Authentication refused: bad ownership or modes for directory  
/home/usuario/.ssh
```

Solución SSH:

- Corregir permisos:

 Bash

```
chmod 700 ~/.ssh  
chmod 600 ~/.ssh/authorized_keys  
chown usuario:usuario ~/.ssh -R
```

- Verificar que la clave pública correcta está en authorized_keys.
- Asegurarse de que el tipo de clave (ed25519, rsa) está permitido en sshd_config.

Diagnóstico/solución RDP/NLA:

- Asegurarse de:
 - Usuario y contraseña correctos.
 - NLA habilitada en el servidor.
 - Cliente compatible (p.ej. FreeRDP con /sec:nla).
 - Si un cliente antiguo no soporta NLA, usar uno actualizado (FreeRDP / cliente oficial de Microsoft).
-

3. Seguridad avanzada

3.1. RDP: NLA, cifrado TLS, puerto alternativo

Comprobaciones:

1. NLA:

- Configuración del sistema → Acceso remoto → opción de “solo equipos con Autenticación de nivel de red” marcada.
- En gpedit.msc, la directiva “Requerir Autenticación a nivel de red” habilitada.

2. Cifrado TLS:

- Con NLA activo, RDP usa TLS por defecto.
- Puedes capturar tráfico con Wireshark y ver que el puerto 3390/TCP (o el que uses) comienza con handshake TLS.

3. Puerto alternativo:

- Ver en regedit el PortNumber (como se configuró en 2.2).

Limitación de acceso:

- **Windows Firewall** → regla de entrada de RDP → pestaña **Ámbito** → “Direcciones IP remotas”.
 - Añade solo las IP públicas de confianza (por ejemplo la de tu trabajo).
 - Control de usuarios:
 - Solo miembros del grupo “Usuarios de escritorio remoto”.
-

3.2. SSH: claves y cifrado de sesión, restricciones

Autenticación por clave pública / privada:

- En ~/.ssh/authorized_keys del servidor están las claves públicas autorizadas.
- En el cliente, las claves privadas en ~/.ssh/id_ed25519 (u otras).
- sshd_config:

📄 text

```
PubkeyAuthentication yes
PasswordAuthentication no
```

Cifrado de sesión:

- SSH negocia cifrados (AES, chacha20, etc.).
- Puedes verlos con:

 Bash

```
ssh -Q cipher
```

- Para forzar alguno:

 Bash

```
ssh -c aes256-gcm@openssh.com usuario@IP
```

Limitar acceso a usuarios y rangos IP:

- En /etc/ssh/sshd_config:

 text

```
AllowUsers usuario1 usuario2

Match Address 203.0.113.0/24
    PasswordAuthentication no
    PubkeyAuthentication yes

Match Address 0.0.0.0/0
    PasswordAuthentication no
```

- A nivel de firewall (ej. ufw en Ubuntu):

 Bash

```
sudo ufw allow from 203.0.113.10 to any port 2222 proto tcp
sudo ufw deny 2222/tcp
```

3.3. VNC / NoMachine / TeamViewer / AnyDesk

3.3.1. Cifrado de sesión y autenticación

- **VNC:**
 - Muchas implementaciones básicas **no cifran** por defecto.

- Lo recomendable:
 - Usar un servidor que soporte TLS (por ejemplo, RealVNC Enterprise, TigerVNC con VeNCrypt).
 - O encapsular VNC en un **túnel SSH**.
- Autenticación:
 - Contraseña VNC robusta.
 - Opcional: integración con cuentas del sistema.
- **NoMachine:**
 - Usa el protocolo NX, con **cifrado activado por defecto** (generalmente AES).
 - Autenticación:
 - Credenciales de usuario de sistema.
 - Puede configurarse 2FA en algunos escenarios.
- **TeamViewer / AnyDesk:**
 - Ambas soluciones cifran end-to-end (AES 256 + RSA, según documentación oficial).
 - Autenticación:
 - ID + contraseña.
 - Acceso no presencial con contraseña robusta + 2FA.
 - Posibilidad de listas blancas de equipos autorizados.

3.3.2. Limitar acceso a usuarios e IPs

- **VNC:**
 - Si está directamente abierto a Internet: usa firewall:

 Bash


```
sudo ufw allow from 203.0.113.10 to any port 5900 proto tcp
sudo ufw deny 5900/tcp
```
 - Protege con túnel SSH solo accesible desde IPs concretas.
- **NoMachine:**
 - Ajusta reglas del firewall igual que con SSH/RDP.
 - Autoriza solo cuentas de usuario necesarias.
- **TeamViewer / AnyDesk:**
 - Configura **listas blancas** (solo ciertos IDs/cuentas pueden conectar).
 - Activa **2FA** en la cuenta principal.
 - Desactiva acceso fácil (Easy Access) si no lo controlas bien.

Con estos pasos y comprobaciones puedes elaborar tu memoria:

- Evidenciando conexiones (comandos, capturas).
- Documentando errores (especialmente con rdesktop y NLA).
- Describiendo cómo aseguras y restringes el acceso en cada protocolo.

Message from gpt-5.1-high

Realiza la siguiente tarea con total precisión: "Administración remota en entornos heterogeneos

Requisitos de finalización

Administración remota en entornos heterogéneos

text

Desde un equipo Windows, conéctate a un servidor Linux mediante SSH (PuTTY o Windows Terminal con OpenSSH).

Conéctate al servidor Linux con un usuario autorizado (ssh usuario@ip_linux)

Comprueba que la conexión se establece usando autenticación por clave pública

Ejecuta comandos para comprobar el estado del sistema:

`uname -r` → versión del kernel

`who` → usuarios conectados

`systemctl status ssh` → estado del servicio SSH

Desde un equipo Linux, conéctate a un equipo Windows mediante escritorio remoto rdesktop (comandos) (rdesktop -u UsuarioWindows -p P@ssw0rd ip_windows)- Debes forzar el uso de rdesktop aunque haya errores. - Otra opción para conectarse al escritorio remoto mediante comandos es usar FreeeRDP.

Verifica que la autenticación NLA está activa y que la sesión está cifrada.

Realiza tareas administrativas básicas: listar usuarios, ver servicios, abrir el administrador de tareas.

Acceso remoto seguro desde Internet

text

Configura acceso remoto a tus equipos domésticos desde fuera de la red local usando:

RDP (Windows) con puerto alternativo y NLA activado

VNC / NoMachine desde Linux

TeamViewer, AnyDesk o Ammyy Admin

SSH a Linux

Documenta problemas comunes al acceder desde Internet y cómo los resolviste. Como identificarías y resolverías cada uno de estos problemas:

NAT o firewall bloqueando el puerto

Usuario no autorizado

Configuración de autenticación o claves incorrectas

Seguridad avanzada

📄text

Para cada protocolo o aplicación, comprueba y documenta:

RDP: NLA, cifrado TLS, puerto alternativo

SSH: autenticación por clave pública/privada, cifrado de sesión

VNC / NoMachine / TeamViewer / AnyDesk: cifrado de sesión y autenticación de usuario

Limita acceso solo a usuarios autorizados y, si es posible, a rangos de IP confiables.

"

Inputs are processed by third-party AI and responses may be inaccurate.

Your conversations and certain other personal information will be disclosed to the relevant AI providers and may otherwise be disclosed publicly to help support our community and advance AI research.

Do not submit to our Services any personal information or other sensitive information that you would not want to be shared publicly. By continuing to use our Services, you acknowledge and direct us to engage in such sharing.

