

# PROXY PATTERN



```
public interface Servicio {  
    int obtenPaga (String empleado);  
    void fijaPaga (String empleado, int paga);  
}
```



```
public class ServicioImpl implements Servicio {  
    ...  
}
```



Nuevo requisito:

Algunas partes de la aplicación deberán acceder en modo  
"securizado"

En modo "securizado" solo los administradores podrán modificar el  
sueldo y un usuario solo podrá consultar su propio sueldo



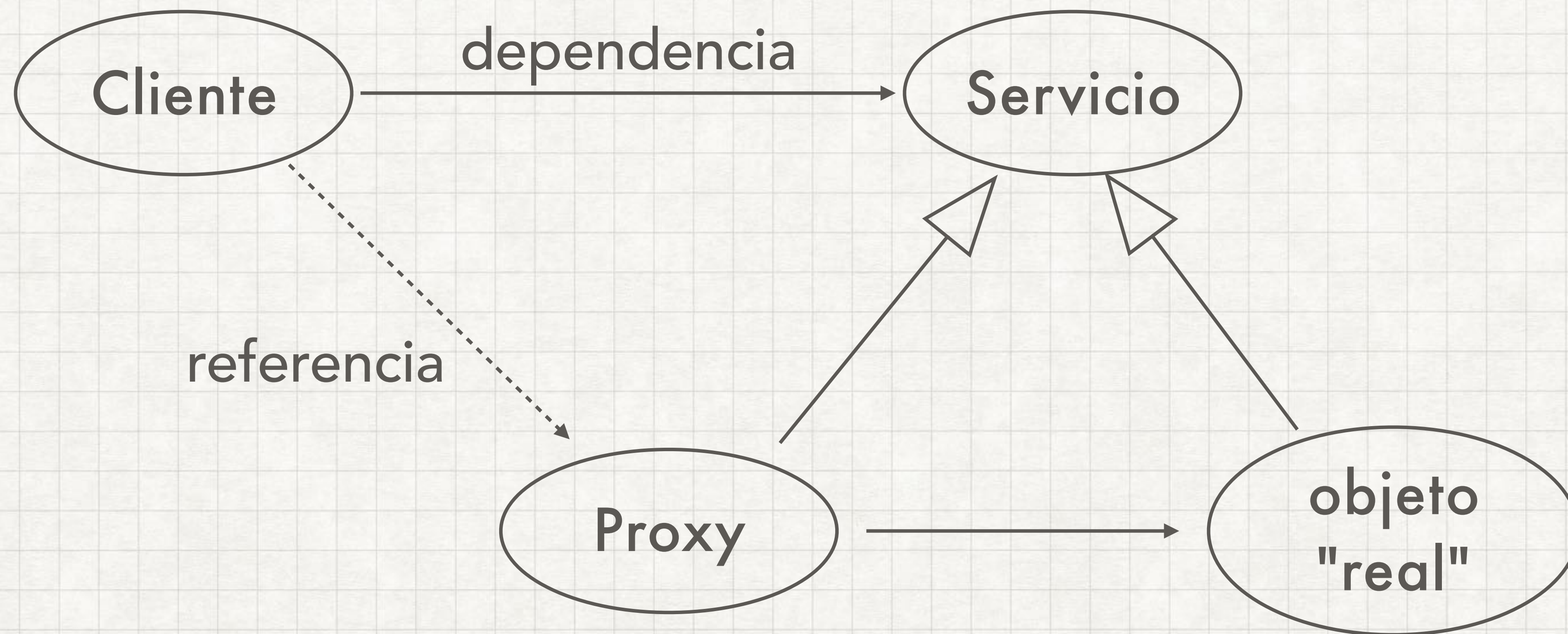
“

Provide a surrogate or placeholder for another object to control access to it.

— *GoF*

”







```
public class ServicioSegurizado implements Servicio {

    private final String usuario;
    private final Servicio componente;

    public ServicioSegurizado(String usuario, Servicio componente) {
        this.usuario = usuario;
        this.componente = componente;
    }

    @Override
    public int obtenPaga(String empleado) {
        if (!empleado.equals(usuario)) {
            throw new UnsupportedOperationException("error de seguridad");
        }
        return componente.obtenPaga(empleado);
    }

    @Override
    public void fijaPaga(String empleado, int paga) {
        if (!"admin".equals(usuario)) {
            throw new UnsupportedOperationException("error de seguridad");
        }
        componente.fijaPaga(empleado, paga);
    }
}
```



## Variante "*Virtual Proxy*"

Cuando la creación del objeto real es costosa

Retrasa al máximo la creación del objeto real



# PROXY PATTERN