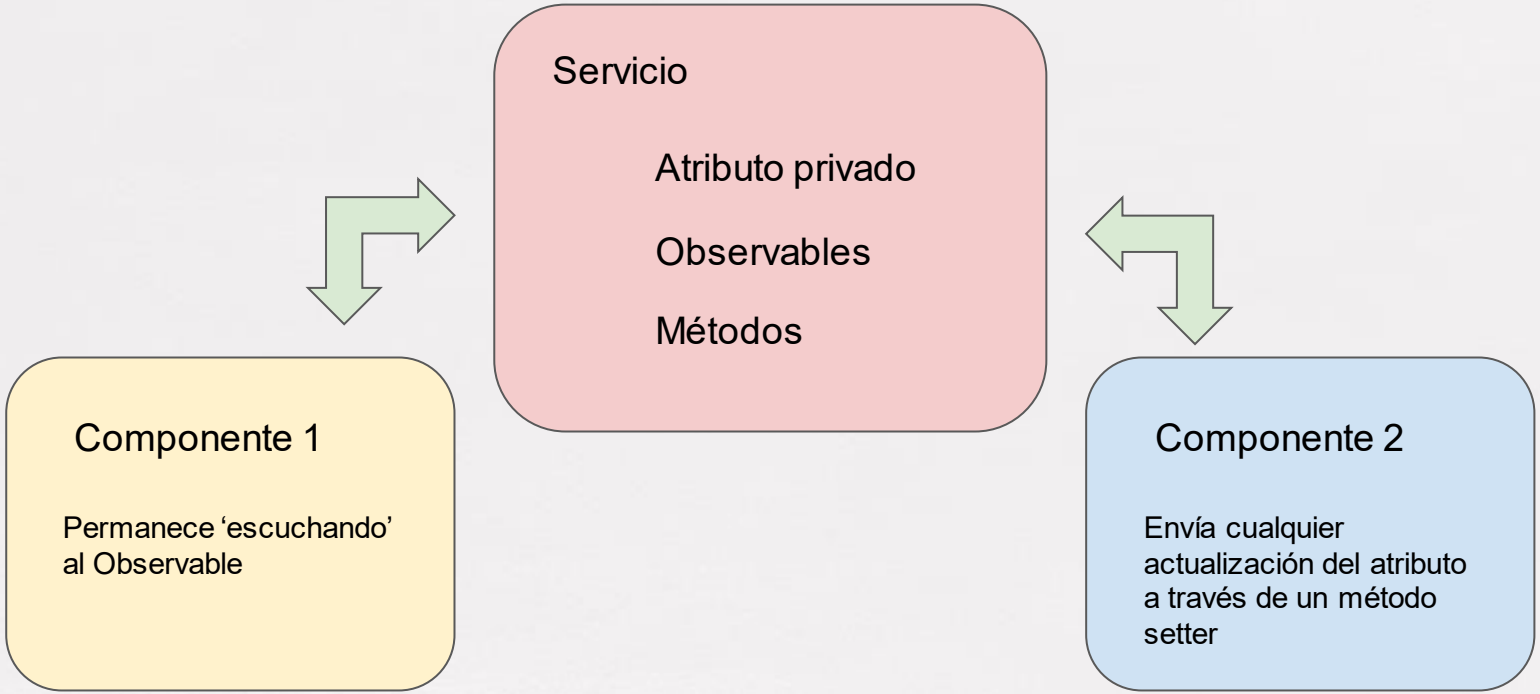
RELACIÓN ENTRE COMPONENTES

Relación mediante un servicio



A través de un servicio

Toda la comunicación entre ambos componentes se realiza a través de un servicio intermedio

SERVICIO

```
export class DataService {
  // Atributos
  private titulo: Subject<string> = new Subject<string>();

  constructor() { }

  // Observables
  public titulo$: Observable<string> = this.titulo.asObservable();

  // Métodos
  public setTitulo(titulo: string): void {
    this.titulo.next(titulo);
  }
}
```

Cada atributo que queramos compartir a través de un servicio deberá de tener estos tres objetos:

- Atributo privado
- Observable para subscribirnos
- Método setter que actualizará el valor del atributo privado

COMPONENTES 1 Y 2

Componente 1

Permanece 'escuchando' actualizaciones del Observable

```
ngOnInit(): void {

    this.dataService.titulo$.subscribe(
      (data: string) => {
        this.titulo = data;
      }
    )
  }
```

```
public actualizarTitulo(titulo: string): void {

    this.dataService.setTitulo(titulo);
  }
```

Componente 2

Actualiza el atributo a través del método

IMPORTANTE SABER

```
private titulo: Subject<string> = new Subject<string>();
```

```
private titulo: BehaviorSubject<string> = new BehaviorSubject('');
```

- Utilizaremos la clase ***Subject*** cuando los componentes estén en la misma vista, es decir relación Padre-Hijo o que ambos sean Hijos de otro componente

- Para relacionar componente que se encuentran en vistas diferentes usaremos la clase ***BehaviorSubjet***.