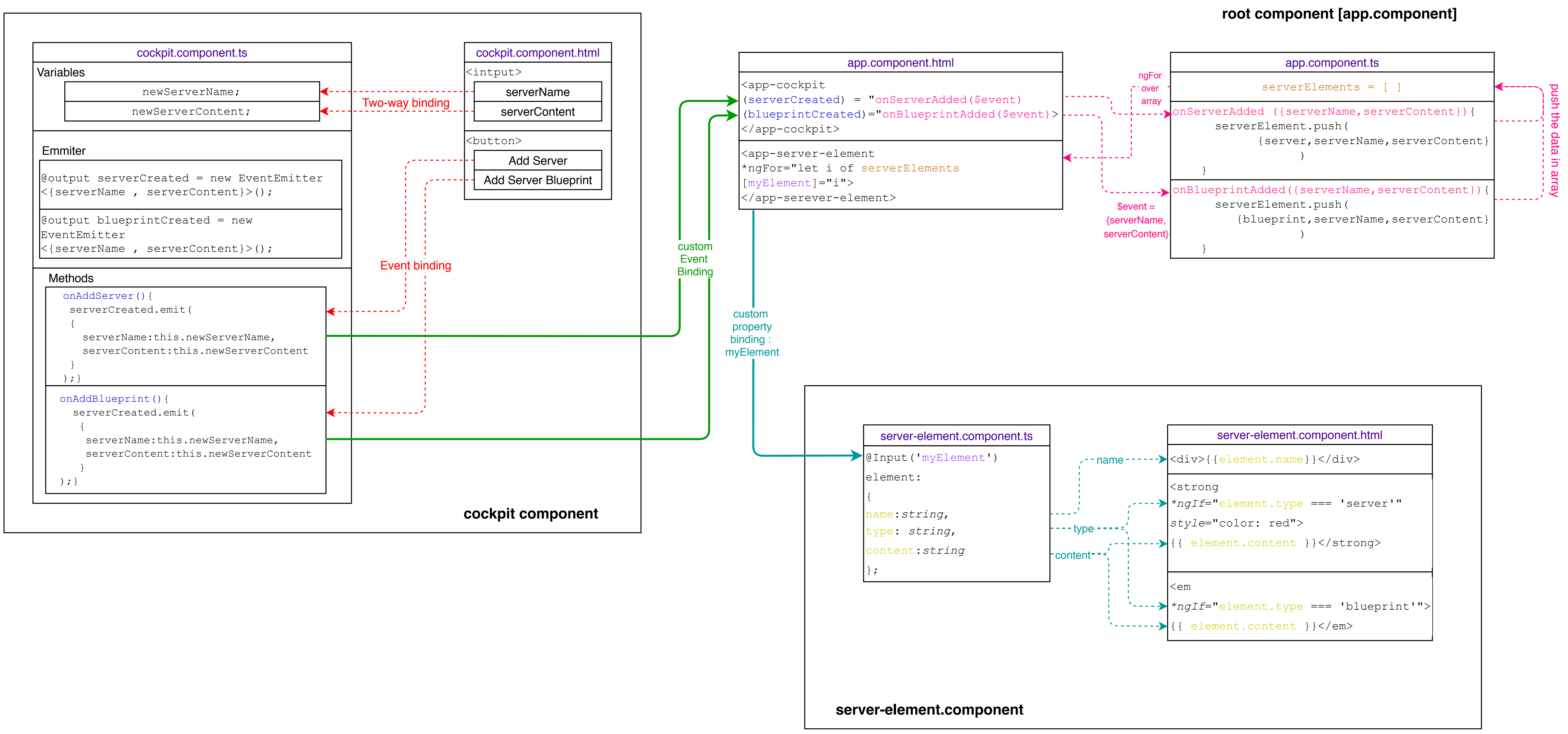

cockpit component

cockpit.component.ts

Variables

```
newServerName;
newServerContent;
```

Emmiter

```
@output serverCreated = new EventEmitter
<{serverName , serverContent}>();
```

```
@output blueprintCreated = new
EventEmitter
<{serverName , serverContent}>();
```

Methods

```
 onAddServer(){
  serverCreated.emit(
  {
    serverName:this.newServerName,
    serverContent:this.newServerContent
  }
 );}
```

```
onAddBlueprint(){
  serverCreated.emit(
   {
    serverName:this.newServerName,
    serverContent:this.newServerContent
   }
);}
```

cockpit.component.html

```
<intput>
```

serverName
serverContent

```
<button>
```

Add Server
Add Server Blueprint

root component [app.component]

app.component.html

```
<app-cockpit
(serverCreated) = "onServerAdded($event)
(blueprintCreated)="onBlueprintAdded($event)>
</app-cockpit>
```

```
<app-server-element
*ngFor="let i of serverElements
[myElement]="i">
</app-serever-element>
```

app.component.ts

```
serverElements = [ ]
```

```
onServerAdded ({serverName,serverContent}){
      serverElement.push(
            {server,serverName,serverContent}
                        )
     }
```

```
onBlueprintAdded({serverName,serverContent}){
      serverElement.push(
         {blueprint,serverName,serverContent}
                        )
     }
```

server-element.component

server-element.component.ts

```
@Input('myElement')
element:
{
name:string,
type: string,
content:string
};
```

server-element.component.html

```
<div>{{element.name}}</div>
```

```
<strong
*ngIf="element.type === 'server'"
style="color: red">
{{ element.content }}</strong>
```

```
<em
*ngIf="element.type === 'blueprint'">
{{ element.content }}</em>
```