

Brève introduction à



Au commencement,
les composants...



Recherche directe dans la liste

Urgent



À traiter



État

Nom du patient

Né(e) le



Type

Mise à jour



Identifiant du patient

Sexe



N° Télédossier

Création



Créer un télédossier

Colonnes



1 - 38 sur 38



<input type="checkbox"/>	État	Nom du patient	Né
<input type="checkbox"/> ➔	Réponse reçue	test1 test2	
<input type="checkbox"/>	Réponse envoyée	test1 test2	06
<input type="checkbox"/>	Pris en charge	test1 test2	06
<input type="checkbox"/>	Pris en charge par mon établissement	test test	10
<input type="checkbox"/>	Demande envoyée	test test	14
<input type="checkbox"/>	Pris en charge par mon établissement	MON dossier	18
<input type="checkbox"/>	Pris en charge	envoi nom avec plusieurs commu	07
<input type="checkbox"/>	Pris en charge	clem test	11
<input type="checkbox"/>	Pris en charge par mon établissement	envoi communauté test état	08
<input type="checkbox"/>	Pris en charge par mon établissement	envoi communauté test état	08
<input type="checkbox"/>	Pris en charge par mon établissement	ENVOI Test	24



Cliquer sur un télédossier pour afficher son contenu



Recherche directe dans la liste

Urgent



À traiter



État

Nom du patient

Né(e) le



Type

Mise à jour



Identifiant du patient

Sexe



N° Télédossier

Création



Créer un télédossier

Colonnes



1 - 38 sur 38



État

Nom du patient

Né



Réponse reçue

test1 test2



Réponse envoyée

test1 test2

06



Pris en charge

test1 test2

06



Pris en charge par mon établissement

test test

10



Demande envoyée

test test

14



Pris en charge par mon établissement

MON dossier

18



Pris en charge

envoi nom avec plusieurs commu

07



Pris en charge

clem test

11



Pris en charge par mon établissement

envoi communauté test état

08



Pris en charge par mon établissement

envoi communauté test état

08



Pris en charge par mon établissement

ENVOI Test

24



Cliquer sur un télédossier pour afficher son contenu



Recherche directe dans la liste

Urgent



À traiter



État

Nom du patient

Né(e) le



Type

Mise à jour



Identifiant du patient

Sexe



N° Télédossier

Création



Créer un télédossier

Colonnes



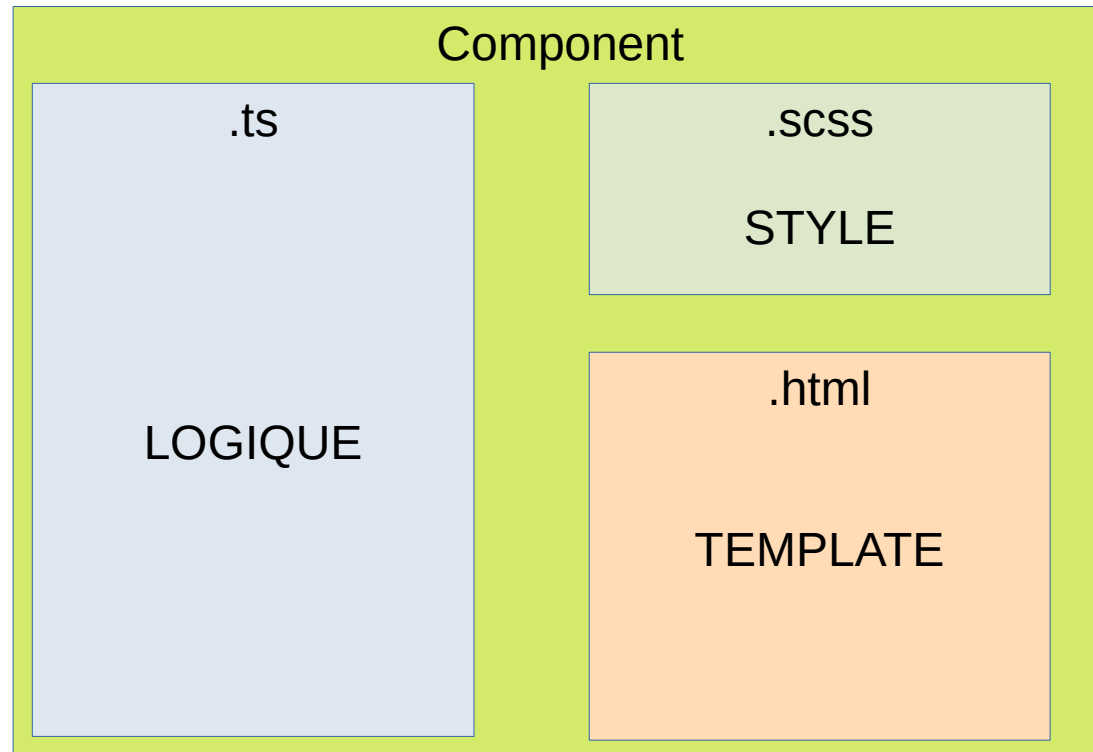
1 - 38 sur 38



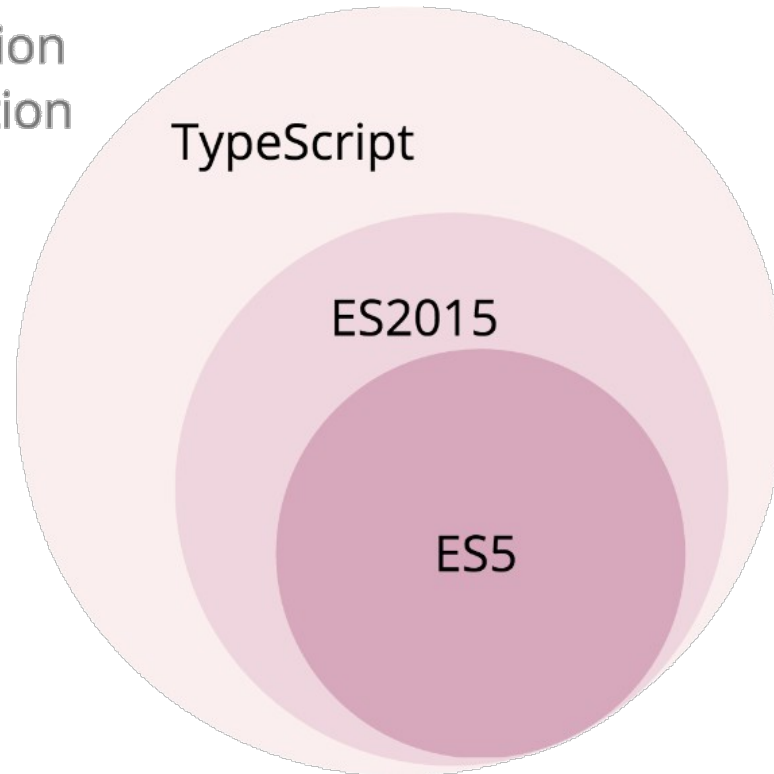
<input type="checkbox"/>	État	Nom du patient	Né
<input type="checkbox"/>	Réponse reçue	test1 test2	
<input type="checkbox"/>	Réponse envoyée	test1 test2	06
<input type="checkbox"/>	Pris en charge	test1 test2	06
<input type="checkbox"/>	Pris en charge par mon établissement	test test	10
<input type="checkbox"/>	Demande envoyée	test test	14
<input type="checkbox"/>	Pris en charge par mon établissement	MON dossier	18
<input type="checkbox"/>	Pris en charge	envoi nom avec plusieurs commu	07
<input type="checkbox"/>	Pris en charge	clem test	11
<input type="checkbox"/>	Pris en charge par mon établissement	envoi communauté test état	08
<input type="checkbox"/>	Pris en charge par mon établissement	envoi communauté test état	08
<input type="checkbox"/>	Pris en charge par mon établissement	ENVOI Test	24



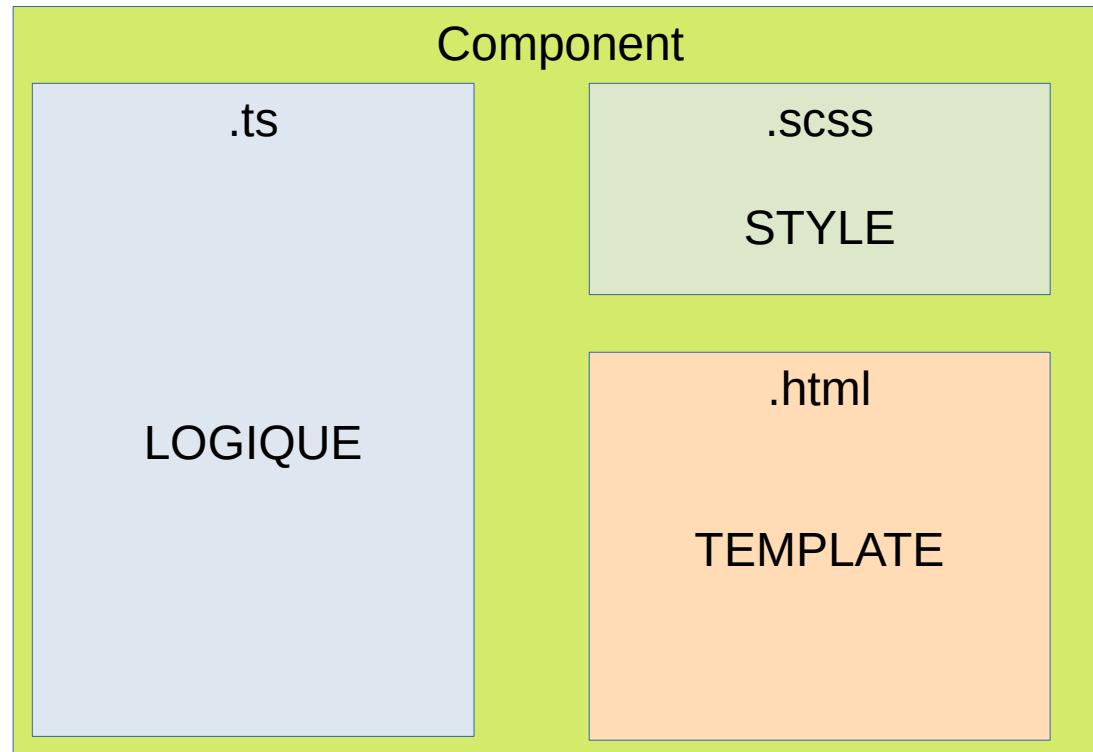
Cliquer sur un télédossier pour afficher son contenu

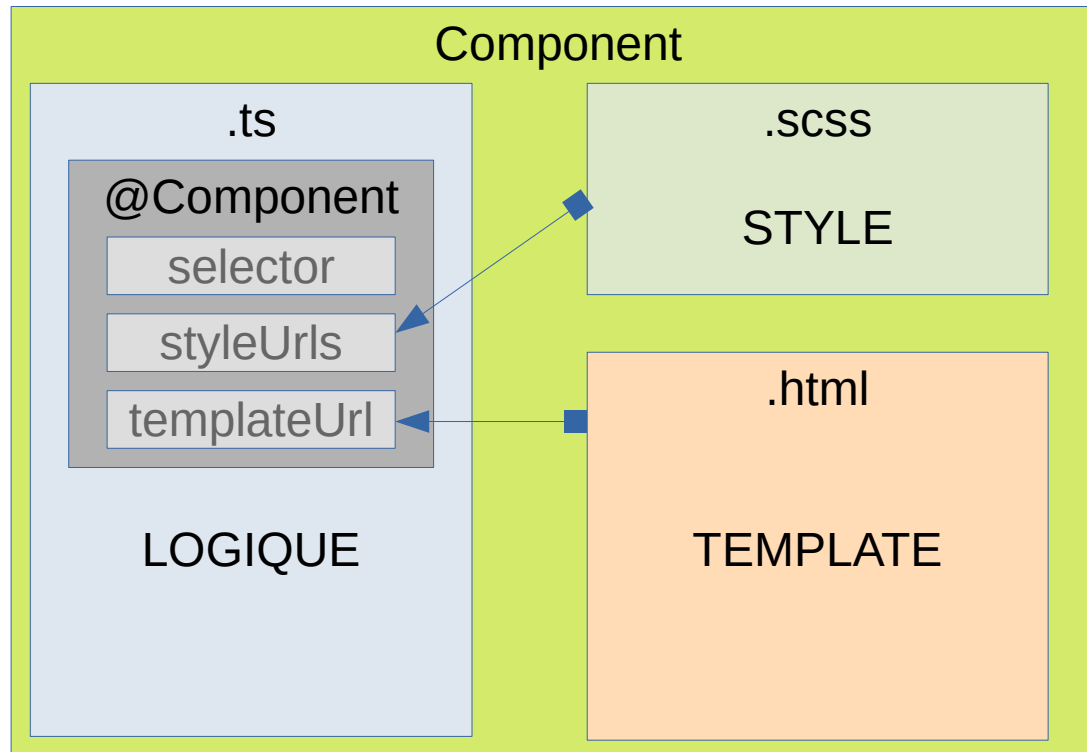


- Type Annotation
- Meta Annotation
- ...

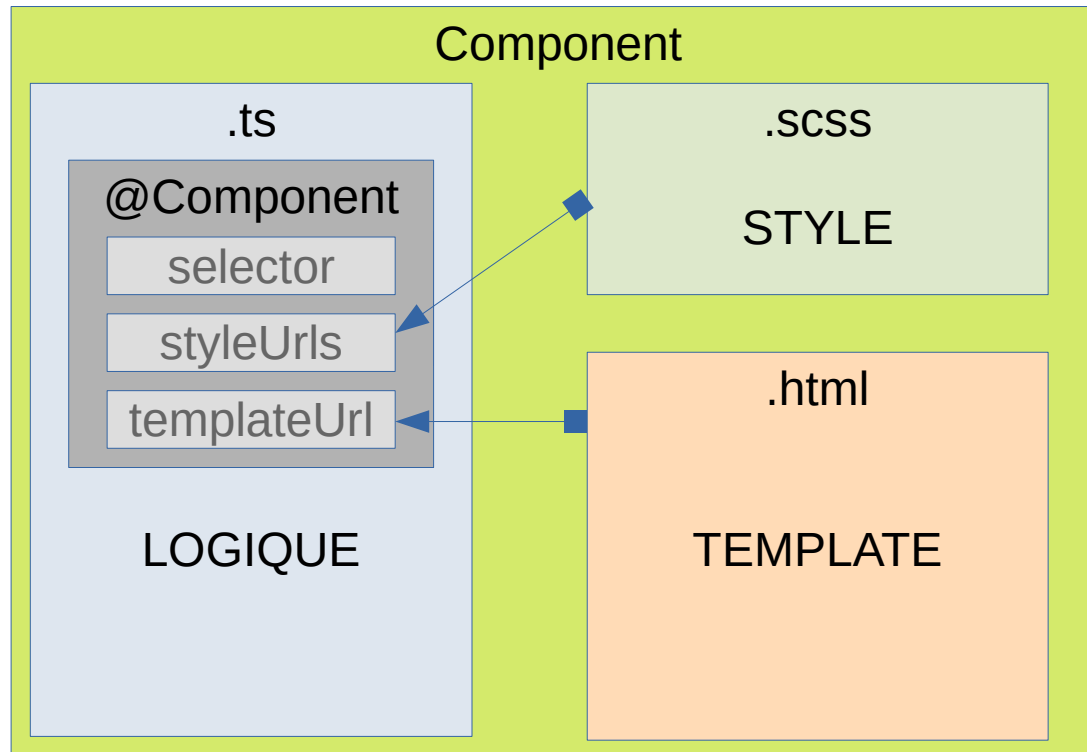


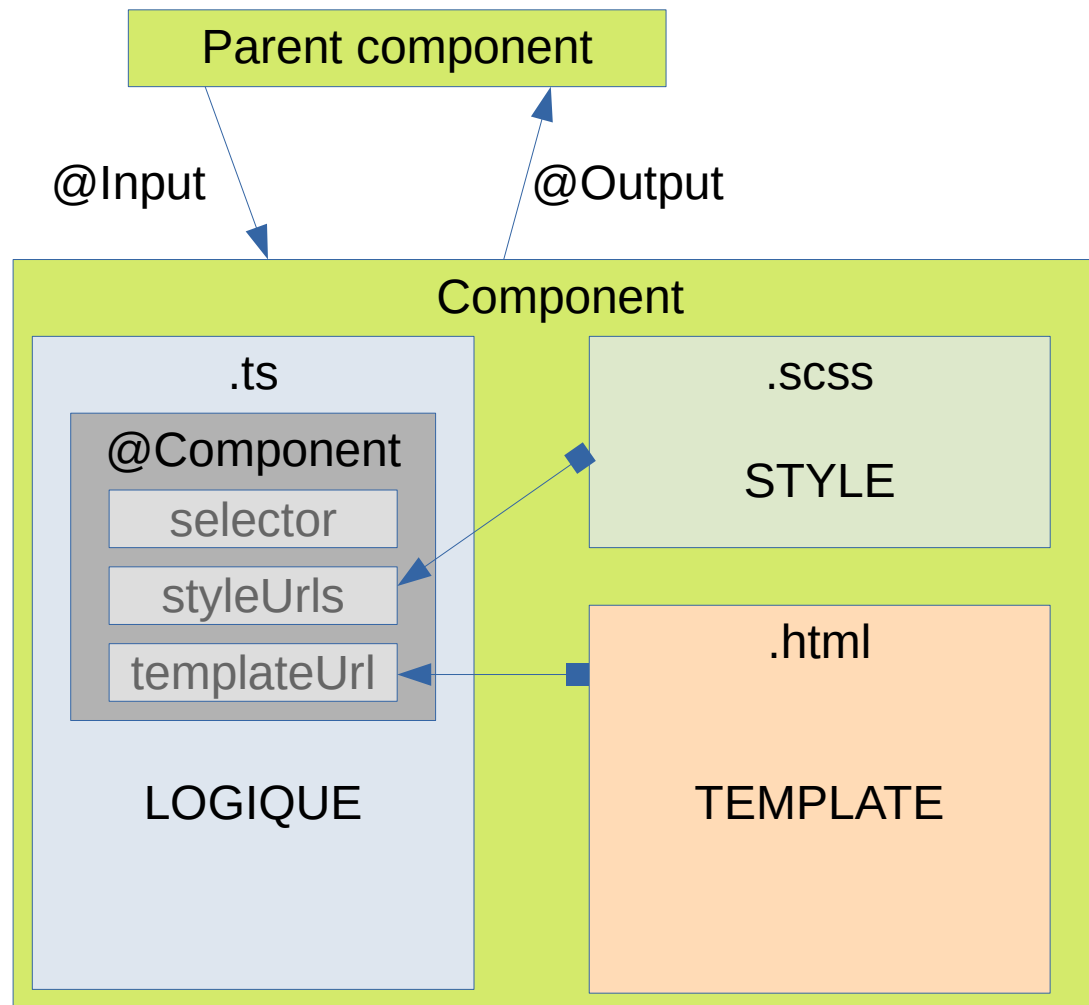
- Classe
- Modules
- ...





```
@Component({  
  selector: 'xc-xcases-list',  
  templateUrl: './xcases-list.component.html',  
  styleUrls: ['./xcases-list.component.scss']  
})  
export class XcasesListComponent implements OnInit {
```





```

<mat-checkbox
    (change)="masterToggle()"
    [checked]="selection?.hasValue() && isAllSelected()"
    [indeterminate]="selection?.hasValue() && !isAllSelected()"
    matTooltipPosition="above"
    matTooltip="{{('apxc.xcase.selectAll' | translate)}}">
</mat-checkbox>

```

```

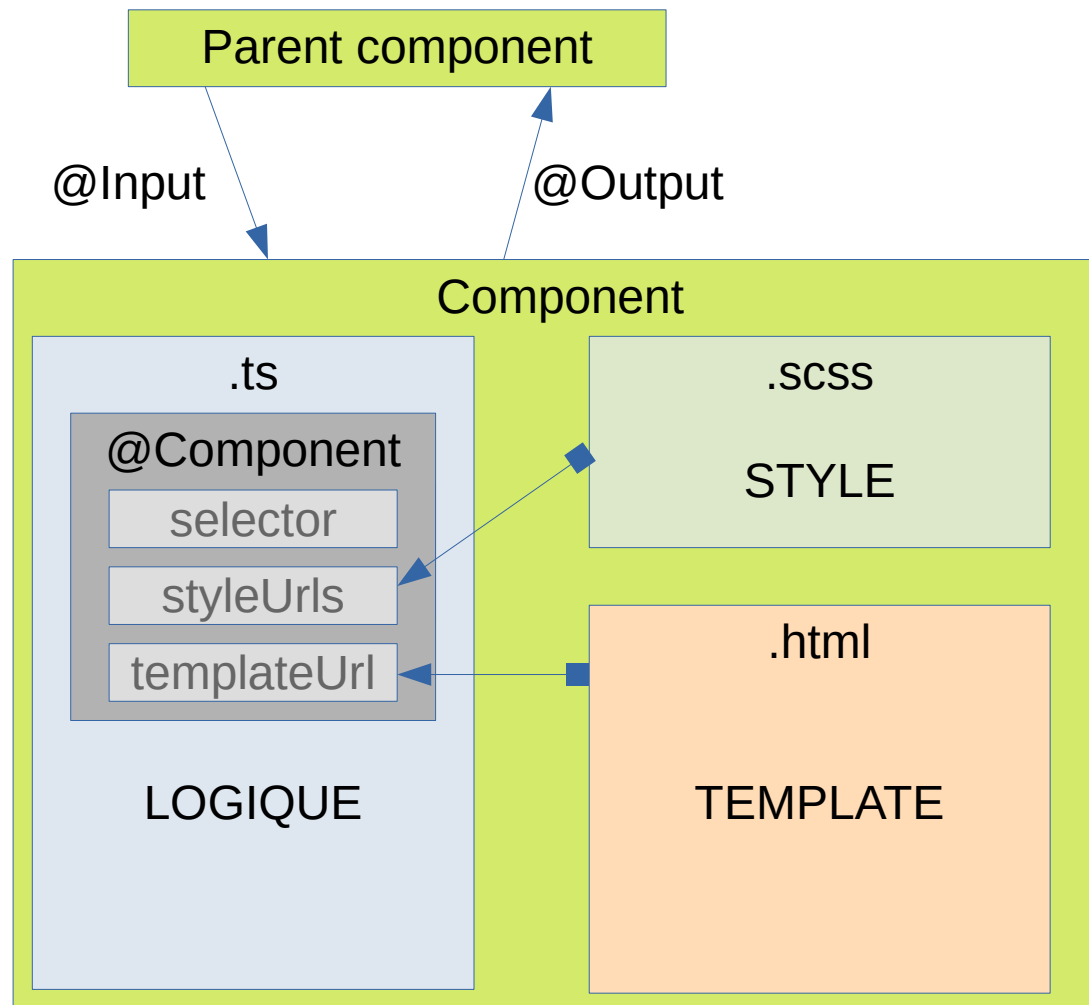
@Component({
  moduleId: module.id,
  selector: 'mat-checkbox',
  templateUrl: 'checkbox.html',
  styleUrls: ['checkbox.css'],
  exportAs: 'matCheckbox',
  host: {
    'class': 'mat-checkbox',
    '[id]': 'id',
    '[attr.tabindex]': 'null',
    '[class.mat-checkbox-indeterminate]': 'indeterminate',
    '[class.mat-checkbox-checked]': 'checked',
    '[class.mat-checkbox-disabled]': 'disabled',
    '[class.mat-checkbox-label-before]': 'labelPosition == "before"',
    '[class._mat-animation-noopable]': '_animationMode === "NoopAnimations"',
  },
  providers: [MAT_CHECKBOX_CONTROL_VALUE_ACCESSOR],
  inputs: ['disableRipple', 'color', 'tabIndex'],
  encapsulation: ViewEncapsulation.None,
  changeDetection: ChangeDetectionStrategy.OnPush
})
export class MatCheckbox extends _MatCheckboxMixinBase implements ControlValueAccessor,
  AfterViewChecked, OnDestroy, CanColor, CanDisable, HasTabIndex, CanDisableRipple {

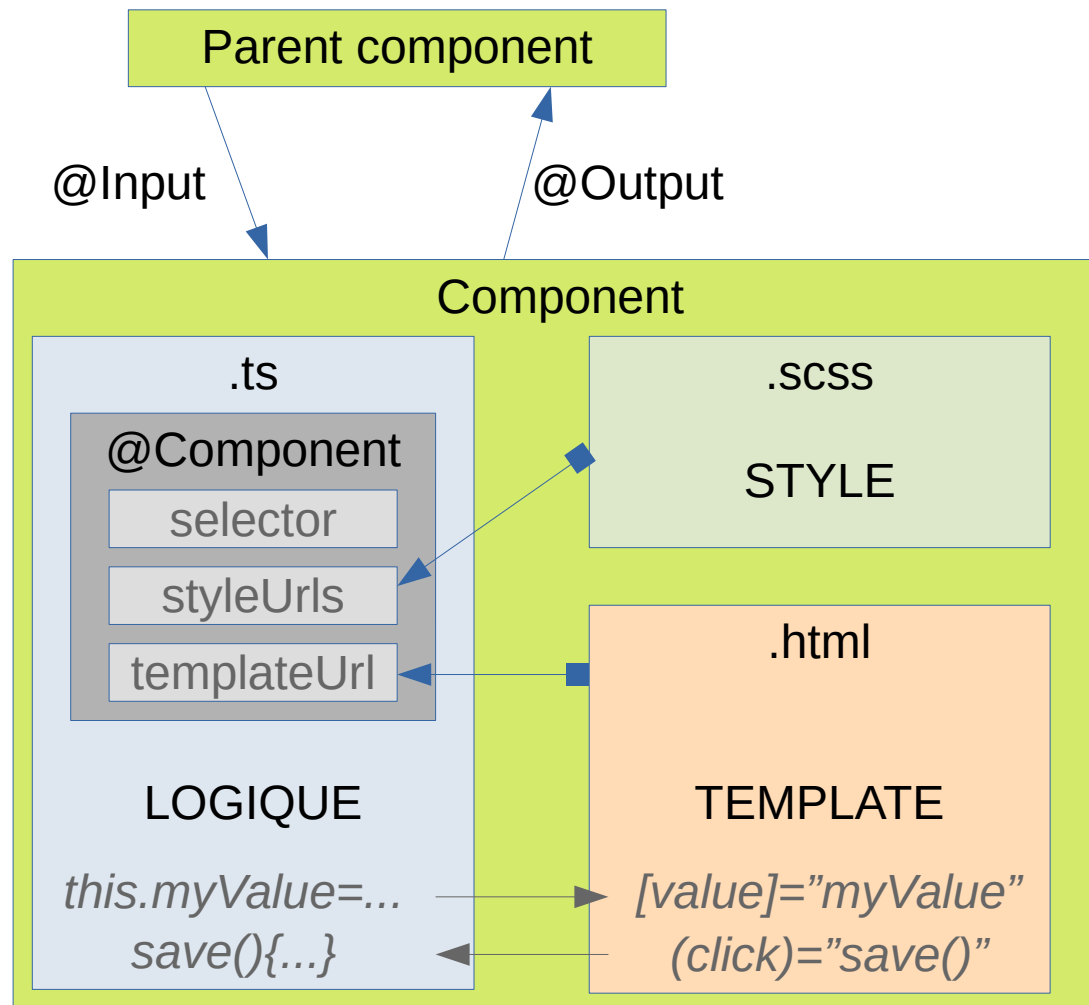
  /**
   * Whether the checkbox is checked.
   */
  @Input()
  get checked(): boolean { return this._checked; }
  set checked(value: boolean) {
    if (value !== this.checked) {
      this._checked = value;
      this._changeDetectorRef.markForCheck();
    }
  }
  private _checked: boolean = false;

  /** Event emitted when the checkbox's 'checked' value changes. */
  @Output() readonly change: EventEmitter<MatCheckboxChange> =
    new EventEmitter<MatCheckboxChange>();

  /** Event emitted when the checkbox's 'indeterminate' value changes. */
  @Output() readonly indeterminateChange: EventEmitter<boolean> = new EventEmitter<boolean>();
}

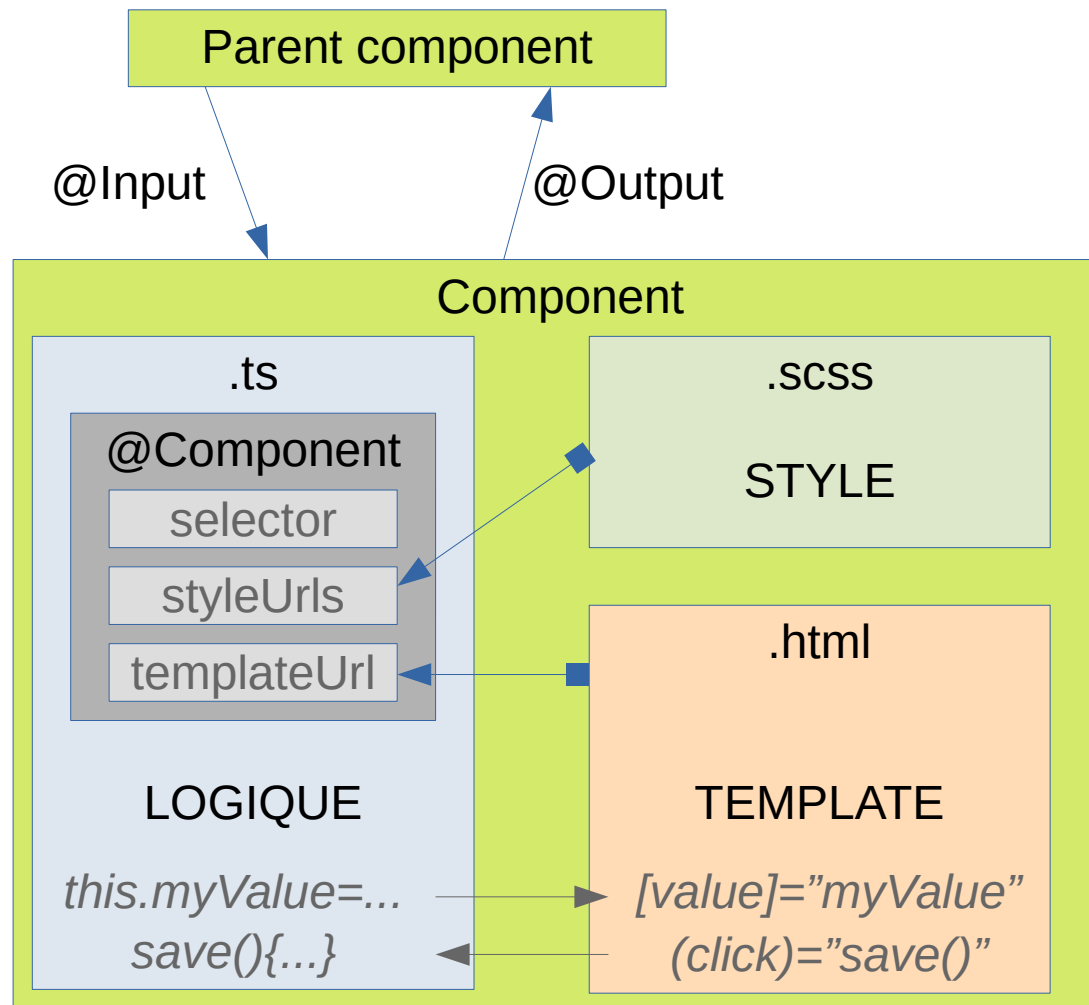
```

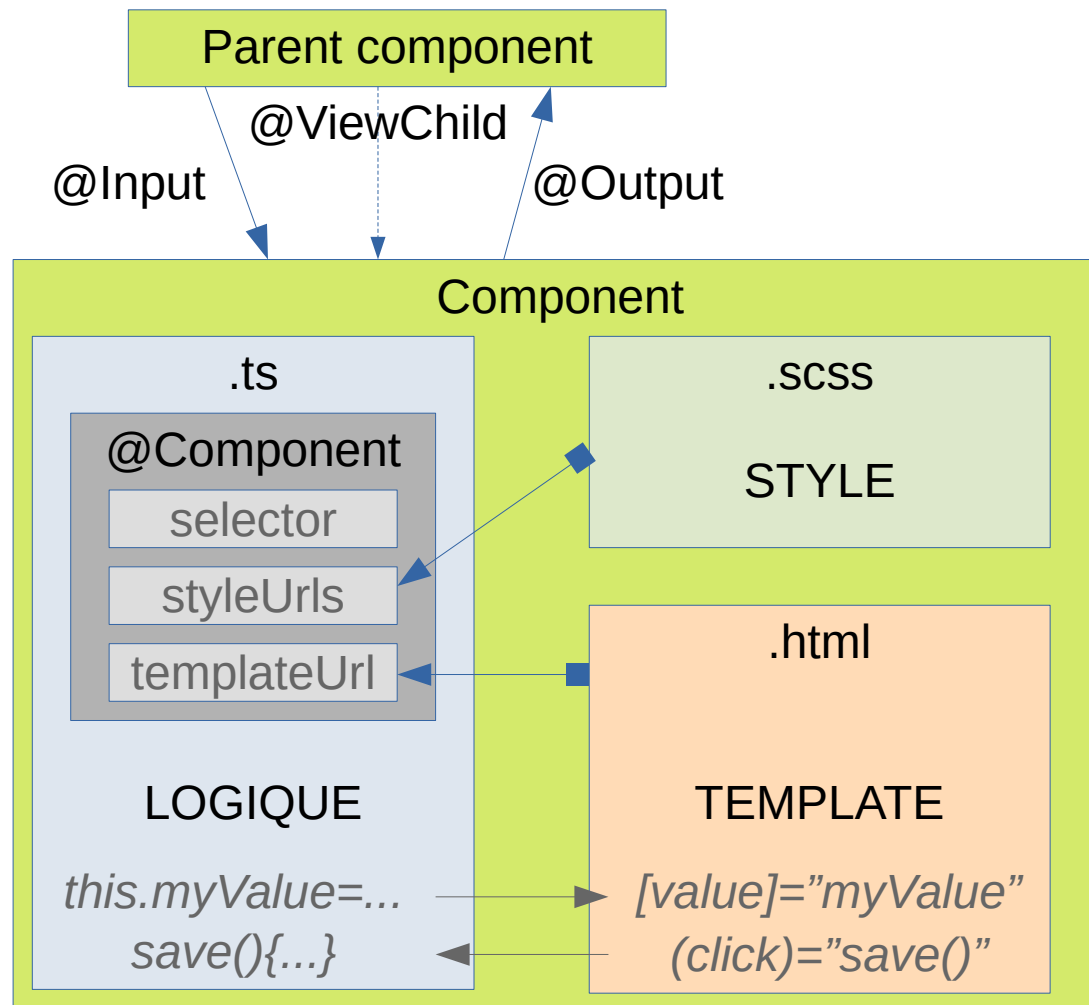




```
@Component({
  selector: 'xc-xcases-preview',
  templateUrl: './xcases-preview.component.html',
  styleUrls: ['./xcases-preview.component.scss']
})
export class XcasesPreviewComponent implements OnInit {
  public xcase: IXcase;
  public lang: string;
```

```
<div>
  <div class="xcases-priority-timer">
    {{xcase?.xcaseComputedDelay}}
  </div>
  <div class="preview-priority">
    {{xcase?.xcasePriority}}
  </div>
</div>
```



```

<ng-container *ngIf="isLoading">
  <div class="spinner-container">
    <mat-spinner [diameter]="80"></mat-spinner>
  </div>
</ng-container>
<mat-table
  [dataSource]="dataSourceStore|async"
  [trackBy]="trackById"
  matSort
  (matSortChange)="sortData($event)"
  matSortActive="{{getCurrentSort().sortBy}}"
  matSortDirection="{{getCurrentSort().orderBy}}"
  class="ac_table mat-table"
  cdkDropListGroup
  [ngClass]="{'no-display':isLoading}"
>
  ...

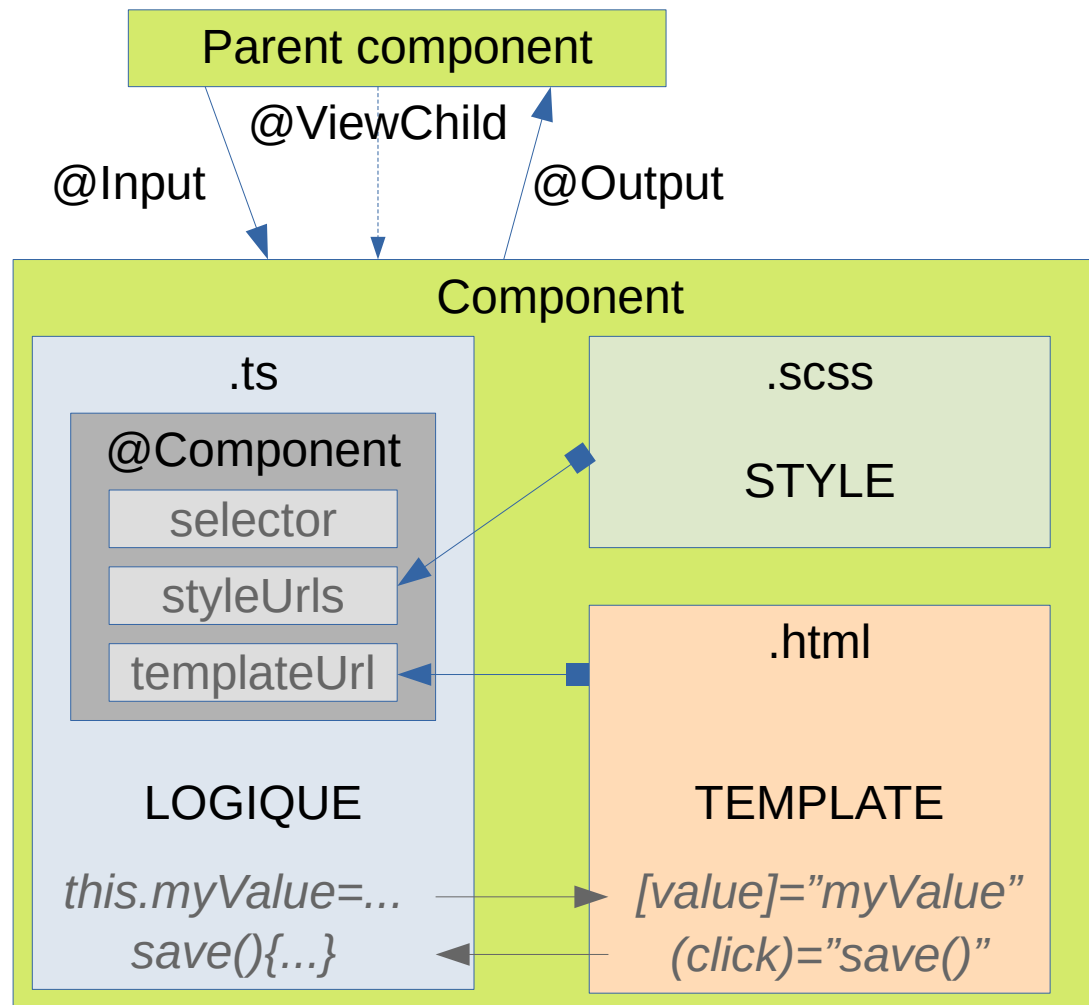
```

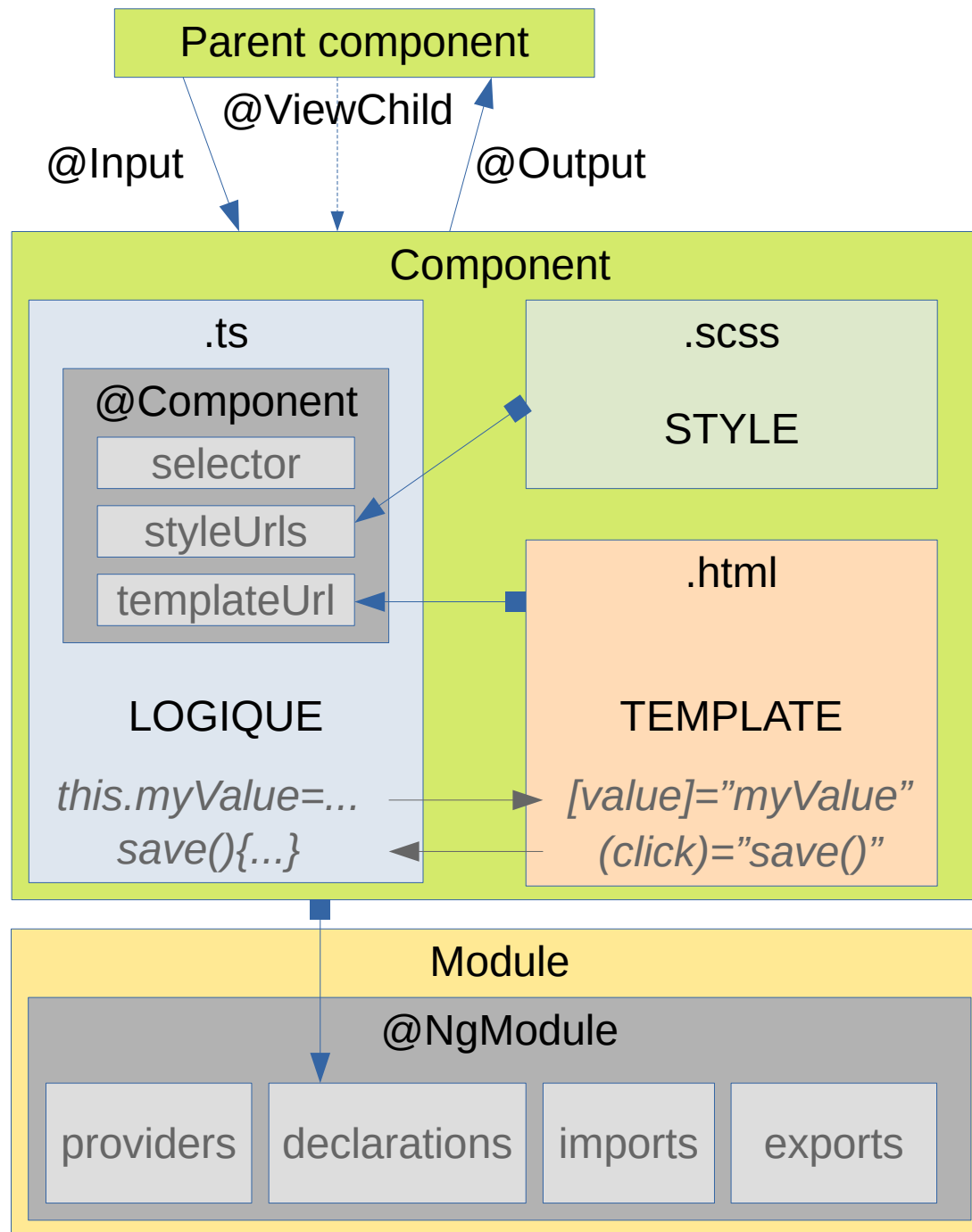
```

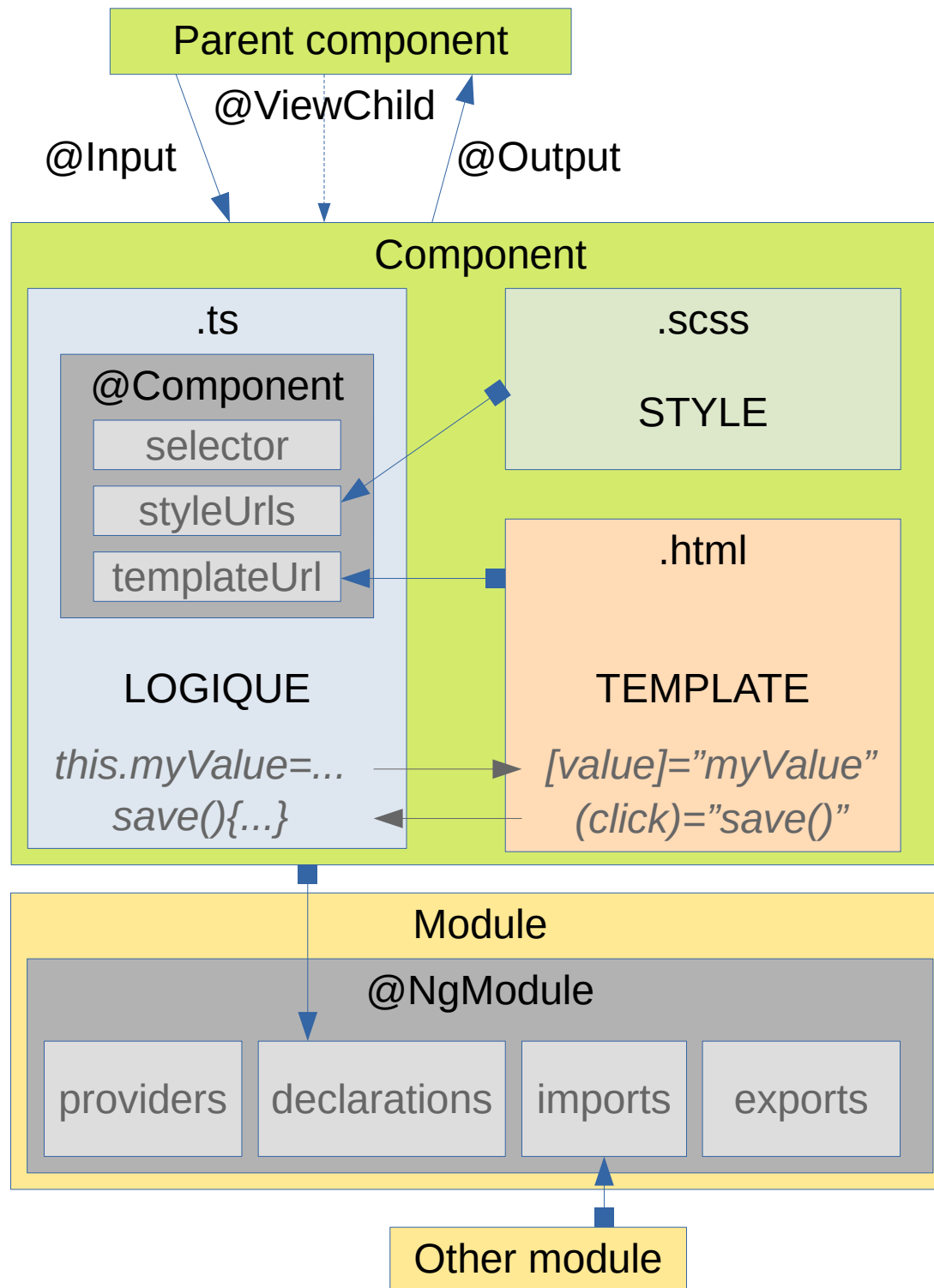
import { MatTable } from '@angular/material/table';
import { MatSort } from '@angular/material/sort';

@Component({
  selector: 'xc-xcases-list',
  templateUrl: './xcases-list.component.html',
  styleUrls: ['./xcases-list.component.scss']
})
export class XcasesListComponent implements OnInit {
  @ViewChild(MatSort, { static: false }) public sort: MatSort;
  @ViewChild(MatTable, { static: true }) private matTable: MatTable<any>;
  @ViewChild(MatTable, { read: ElementRef, static: true }) private matTableRef: ElementRef;

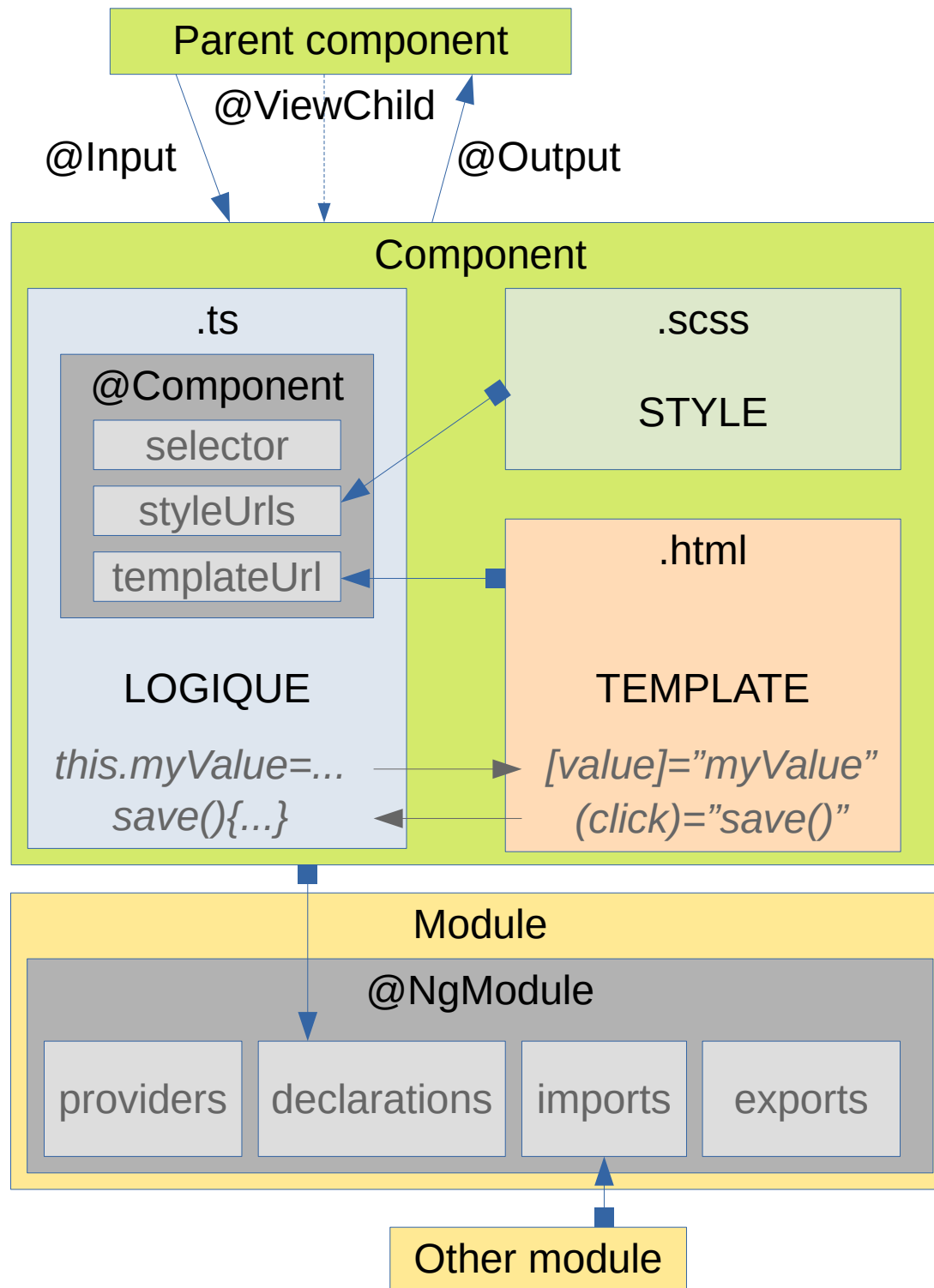
```

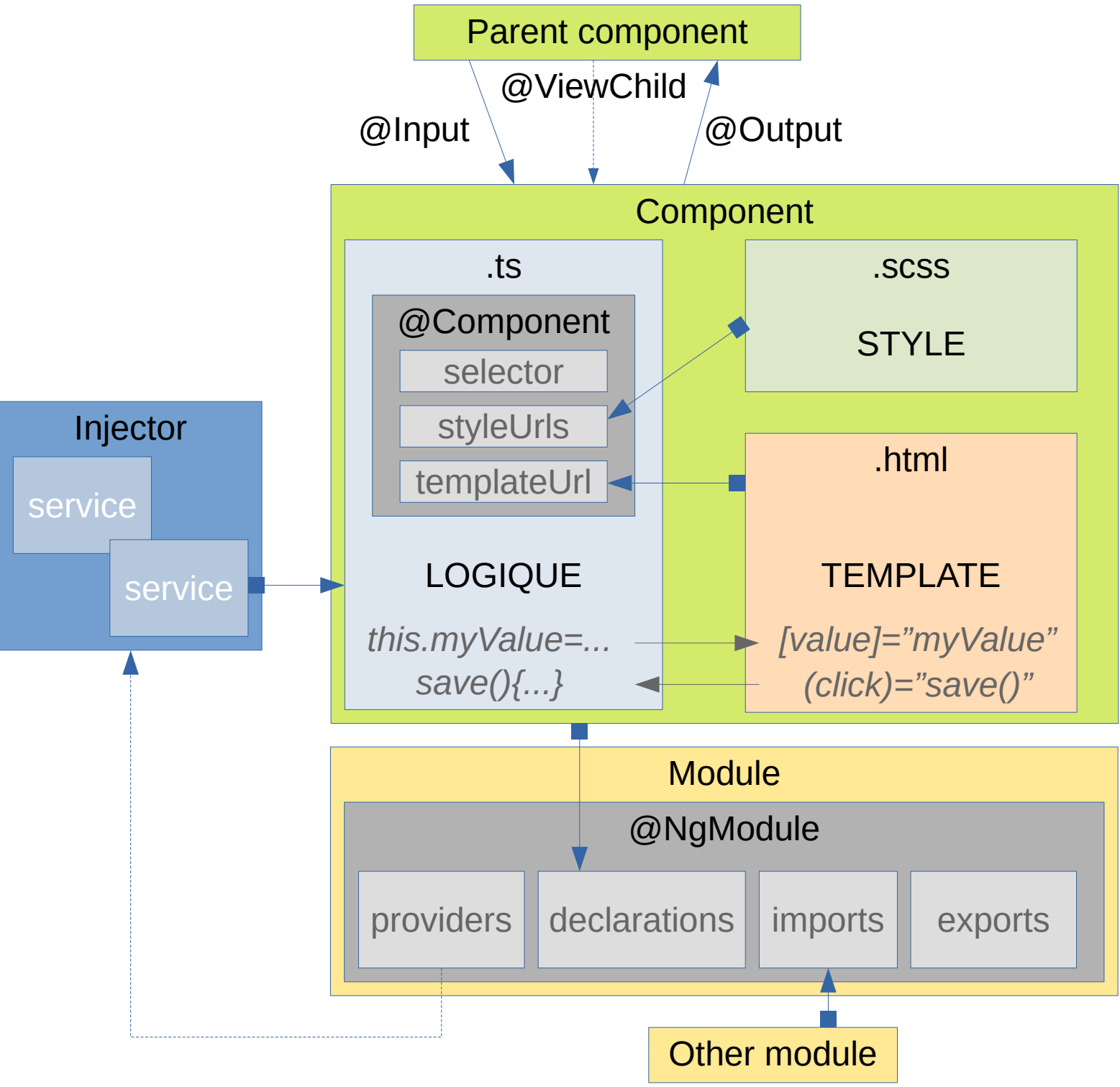


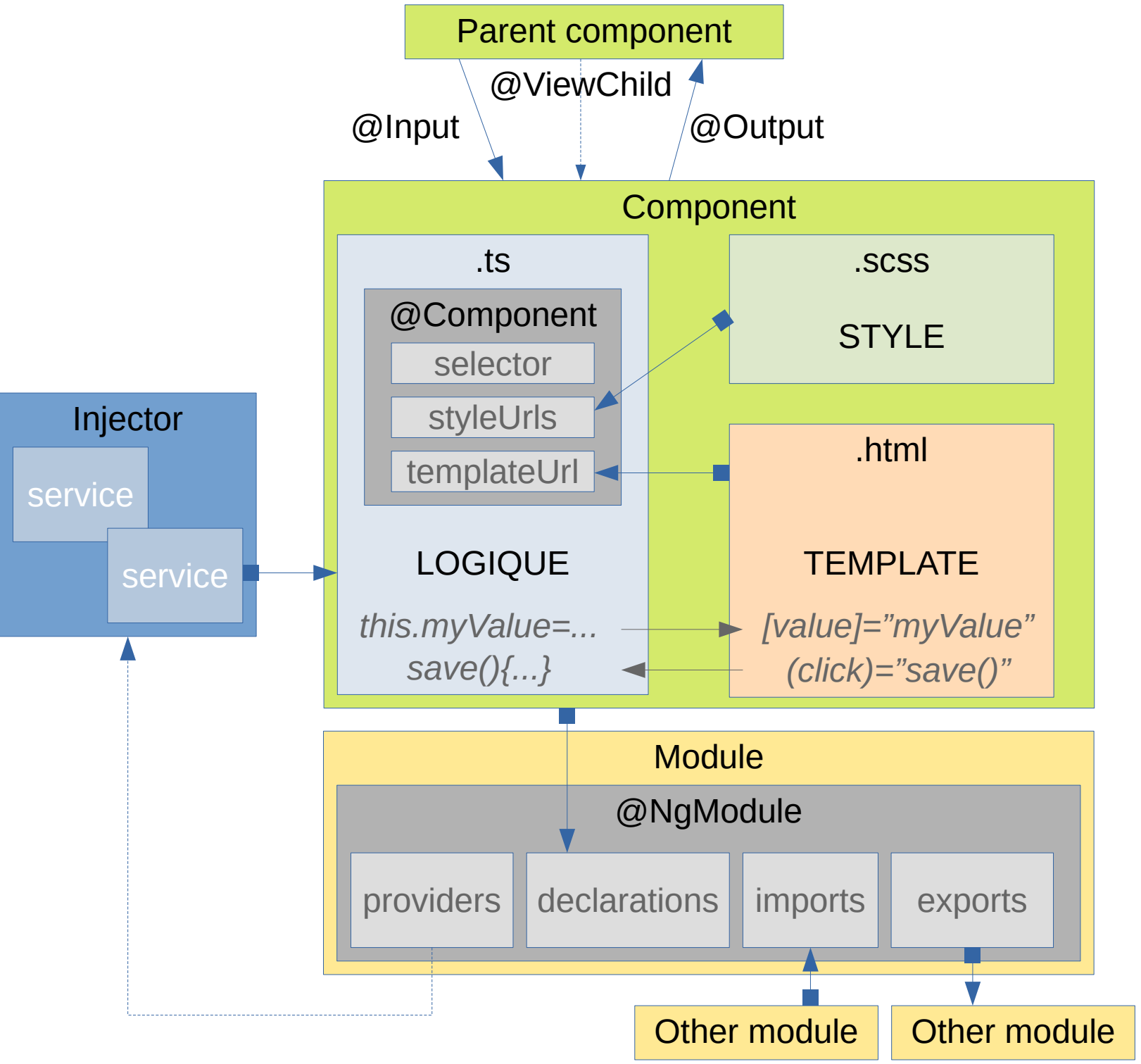




```
@NgModule({
  declarations: [
    XcasesActionsComponent,
    XcasesFiltersComponent,
    XcasesListComponent,
    XcasesPreviewComponent,
    XcasesIframeComponent,
    XcasesDialogComponent,
    XcaseSynthesisComponent,
    XcasePatientInfoComponent,
    XcaseStateComponent,
    XcaseSidePanelComponent,
    XcaseMainAreaComponent,
    XcasePriorityComponent,
    XcaseActionsComponent,
  ],
  imports: [
    ...COMMON_SHARED_IMPORTS,
    ApiModule,
    XcasesRoutingModule
  ]
})
export class XcasesModule {
}
```

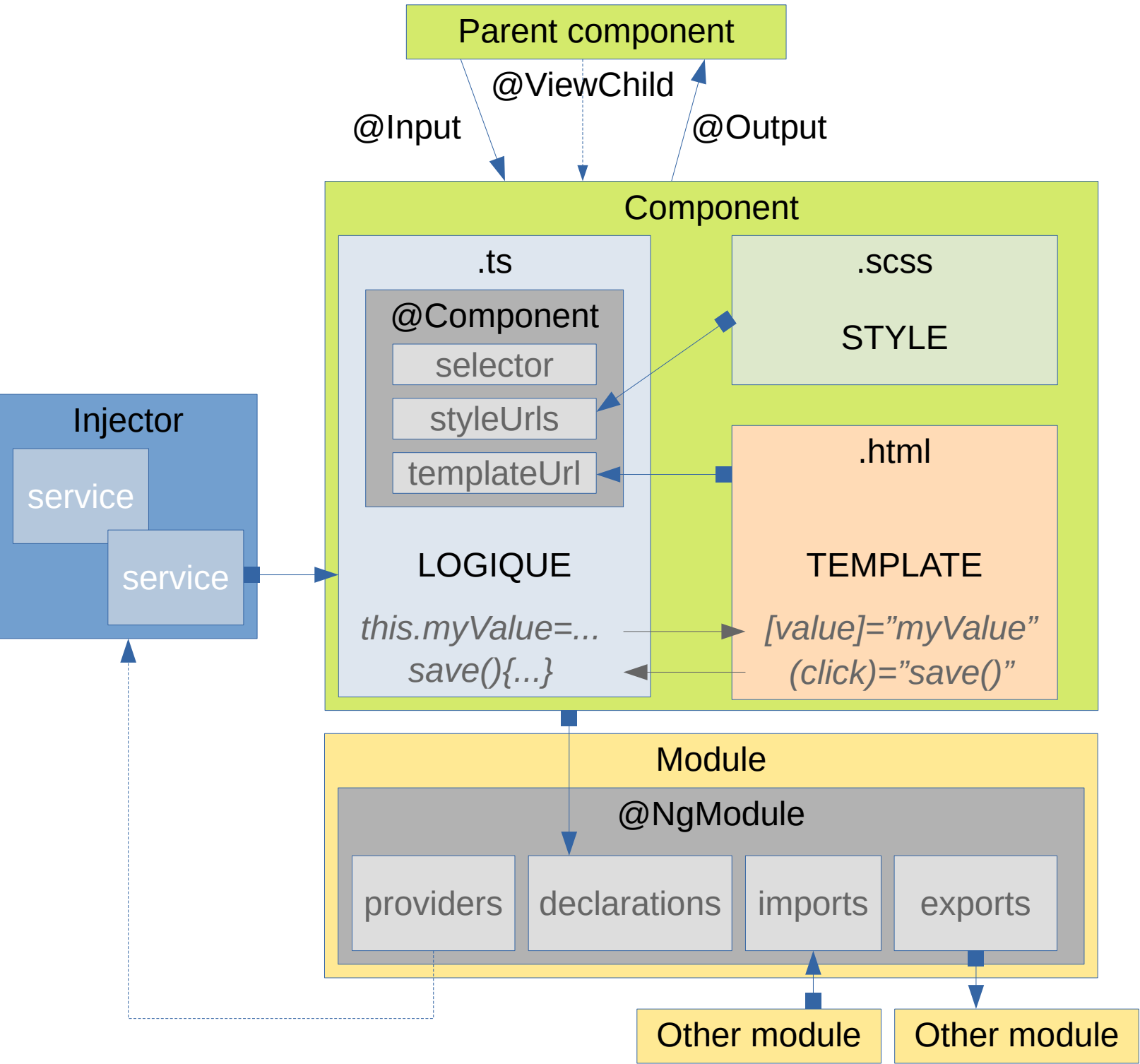


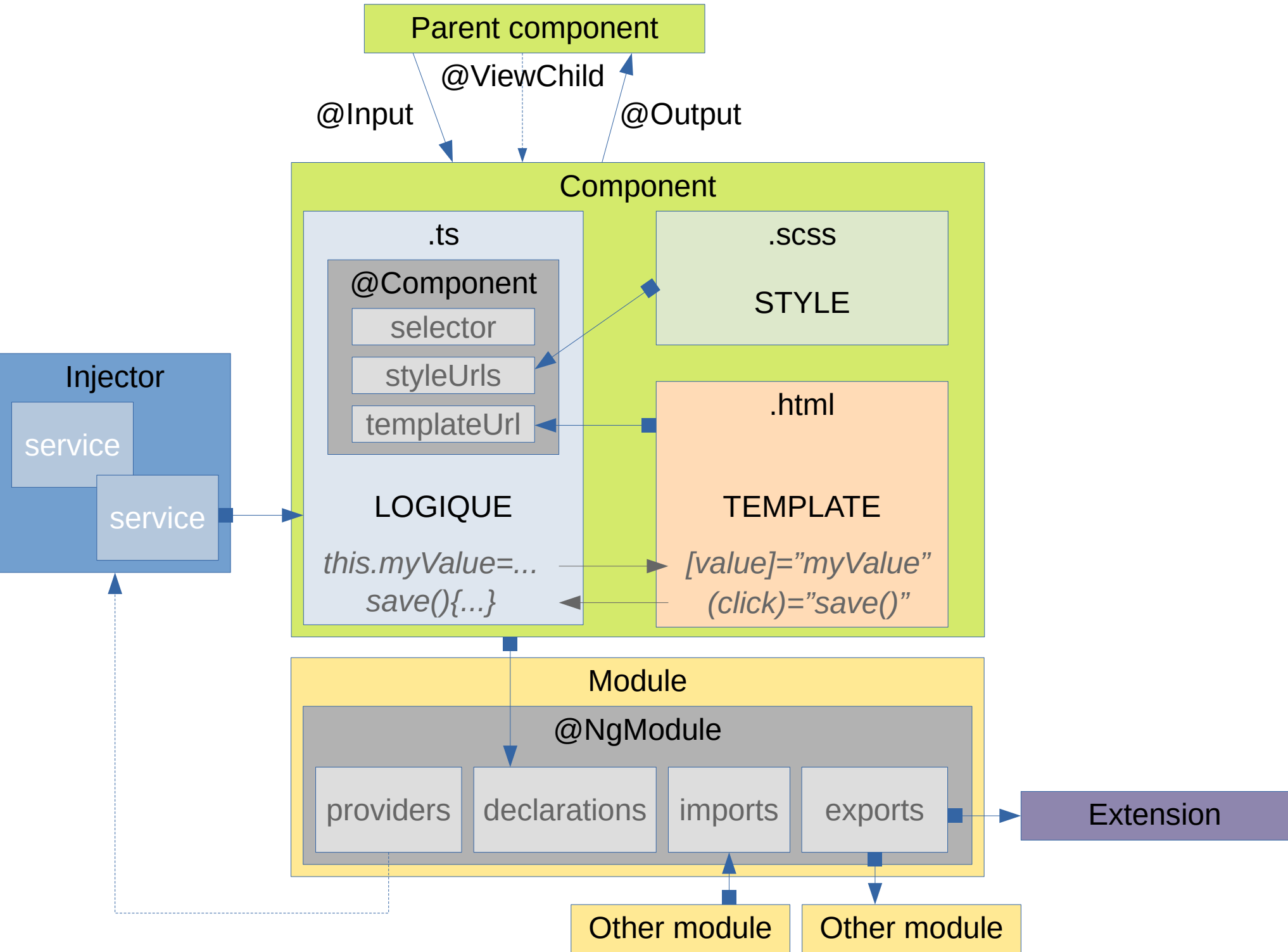


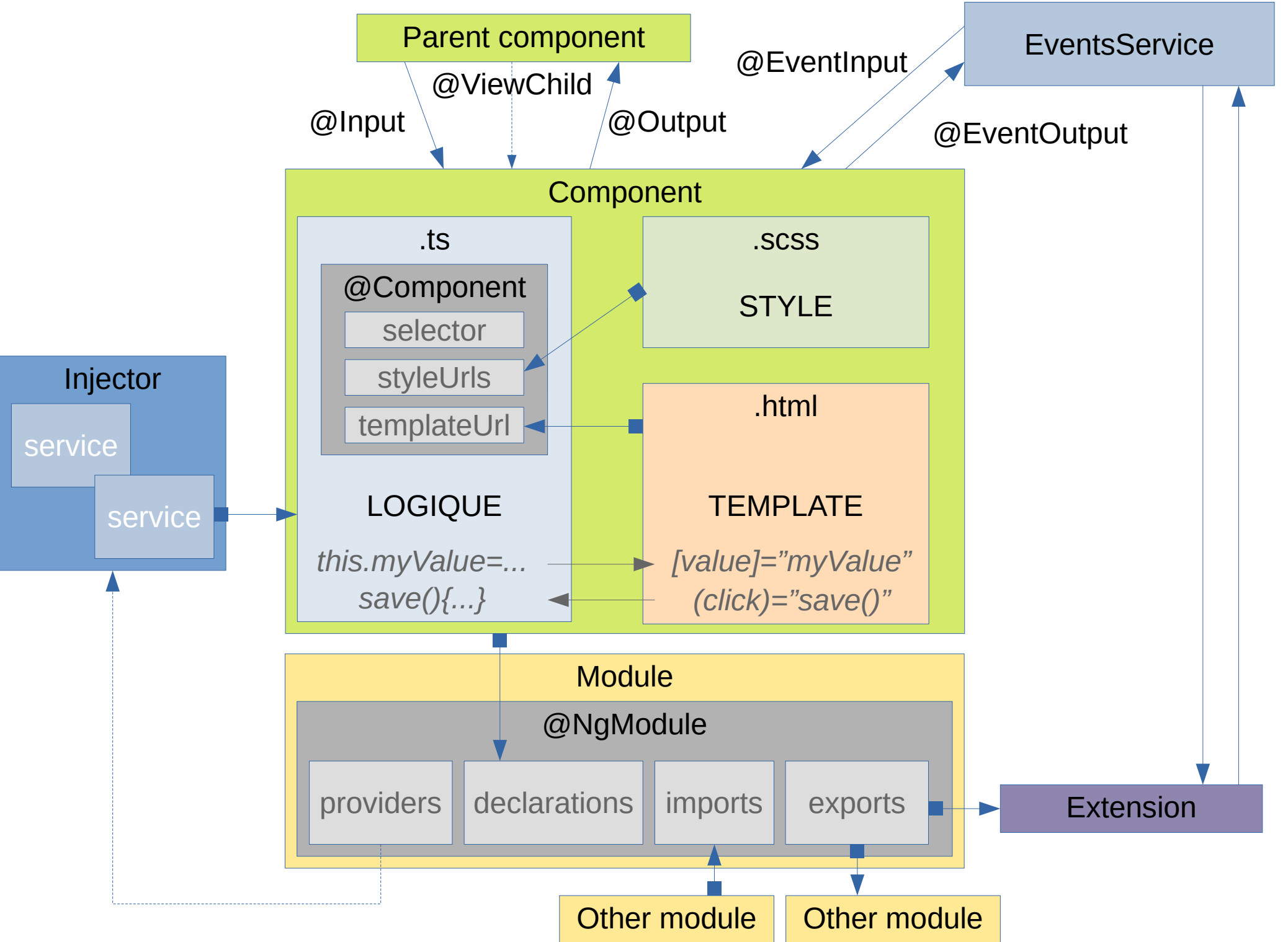


```
import { TranslateModule } from '@ngx-translate/core';
import { TranslateSharedService } from '../services/translate/translate.service';

@NgModule({
  exports: [TranslateModule],
  providers: [TranslateSharedService]
})
export class TranslateSharedModule {
  constructor() {}
}
```







```

@Component({
  selector: 'xc-xcase-actions',
  templateUrl: './xcase-actions.component.html',
  styleUrls: ['./xcase-actions.component.scss']
})
export class XcaseActionsComponent implements OnInit {

  /**
   * Listen legacy to know when user click en send folder
   */
  @EventInput('legacy.xcase_send_folder')
  onGetSendFolder() {
    this.openConfirm();
  }

  /**
   * Notify extension V5plus : need to reload xcase
   */
  @EventOutput('extension_V5plus.reload_xcase_requested')
  public notifExtensionToReloadXcaseView(newStep) {
    // Refresh xcase view
    const actionResult = {
      action: this.selectedAction,
      result: newStep
    };
    return actionResult;
  }
}

```

ET AUSSI ...

*nglf
&
pipe

```
<button mat-button
  (click)="openXcase(xcase)"
  *ngIf="xcasesService.actionIsPossible(xcasesService.XcasesRights.OPEN)"
  [disabled]="xcasesService.actionIsDisabled(xcasesService.XcasesRights.OPEN)"
>
  {{'apxc.xcase.open' | translate}}
</button>
```

*ngFor

```
<p class="subtitle2 default high my-2">{{'apxc.xcase.actors' | translate}}</p>
<div class="preview-actors" *ngFor="let actor of xcase?.xcaseActorsArray">
|   <div class="caption1 default medium">{{actor}}</div>
</div>
```

Lifecycle Hooks

constructor

ngOnChanges

ngOnInit

ngDoCheck

ngAfterContentInit

ngAfterContentChecked

ngAfterViewInit

ngAfterViewChecked

ngOnDestroy

Et les observables ?

Introduction

<https://www.coursepixel.com/blogs/rxjs-basics>

Quelques opérateurs

<https://reactive.how/>

<https://player.vimeo.com/video/239108249?title=0&autoplay=0>

<https://player.vimeo.com/video/240131697?title=0&autoplay=0>

API

<https://rxjs-dev.firebaseio.com/api>

