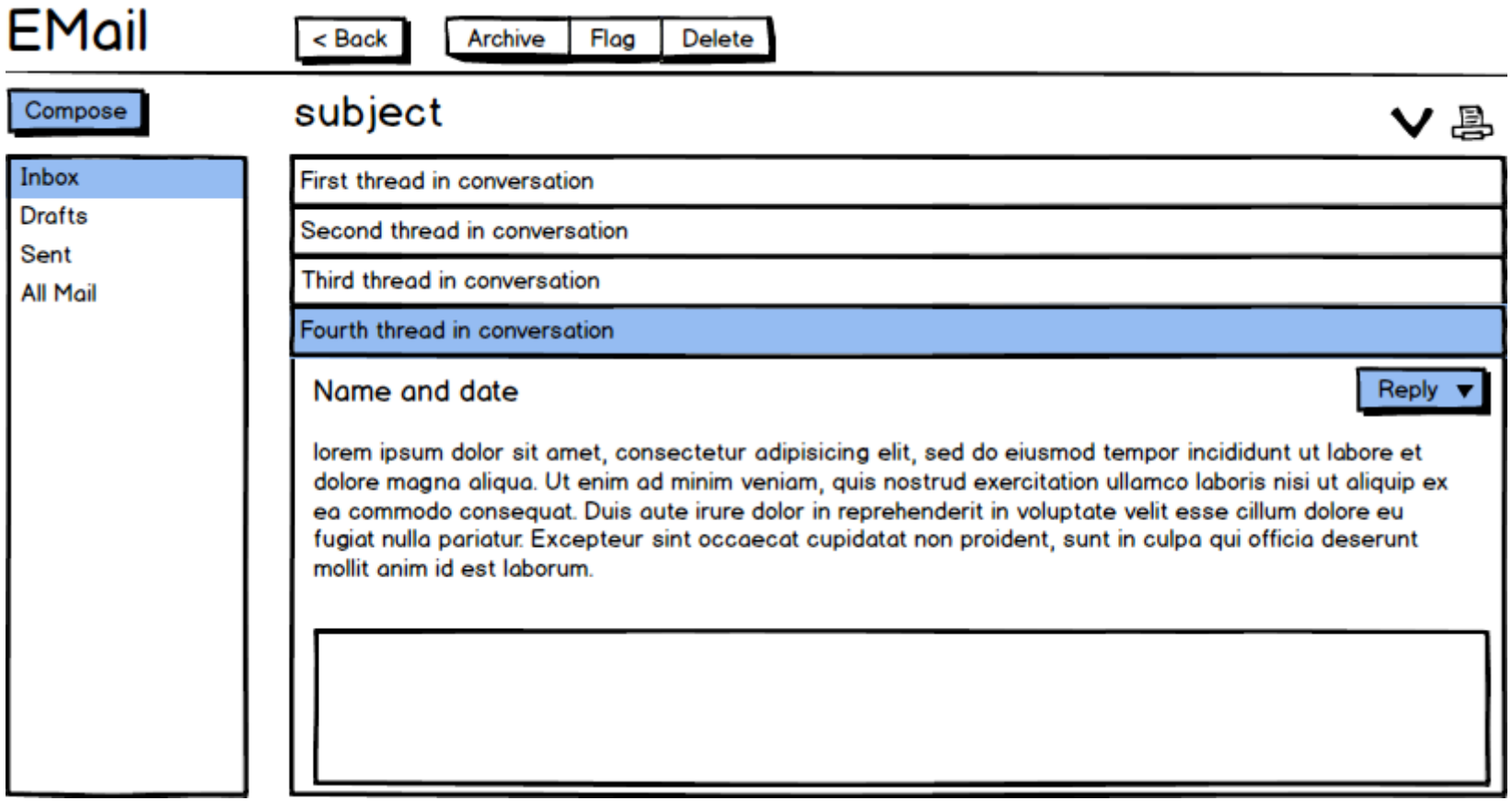
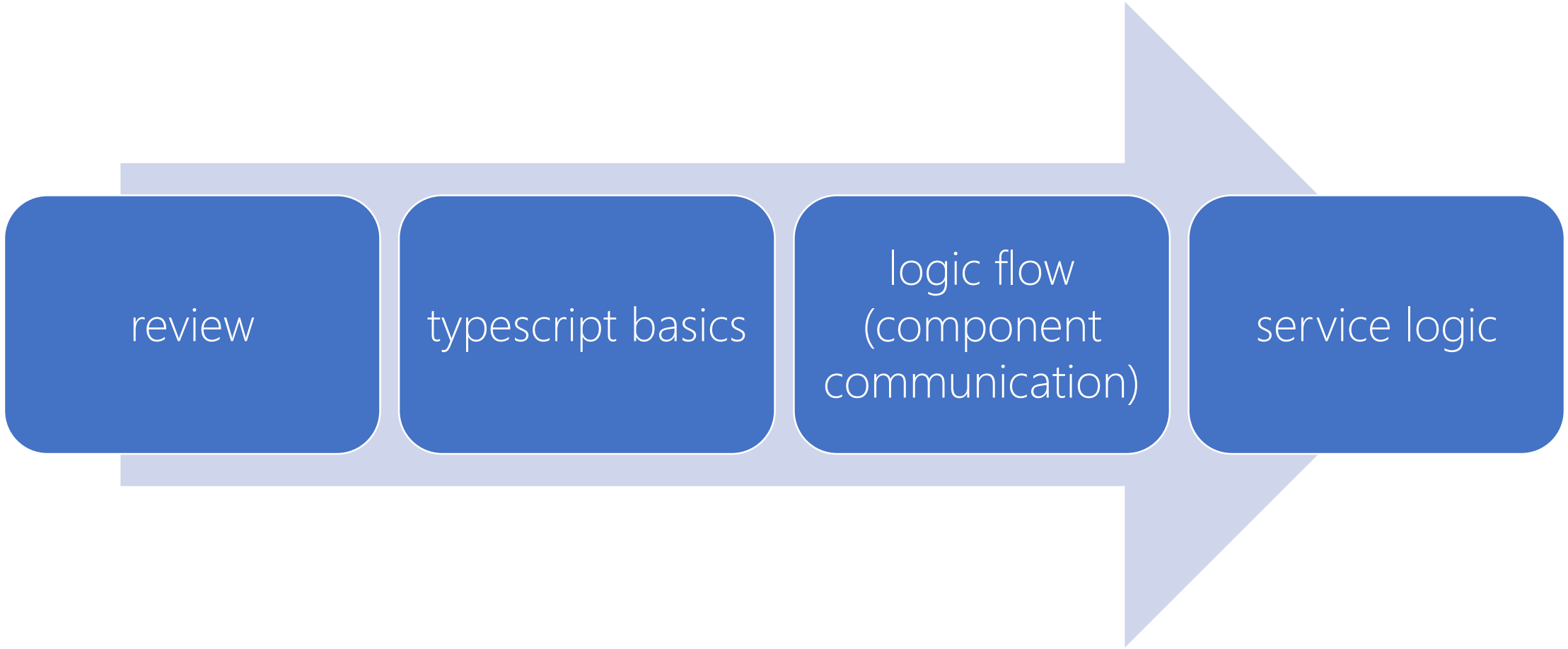
part 3

kamil mijacz


EMail
< Back
Archive
Flag
Delete
Compose
subject
Inbox
Drafts
Sent
All Mail
First thread in conversation
Second thread in conversation
Third thread in conversation
Fourth thread in conversation
Name and date
Reply
lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

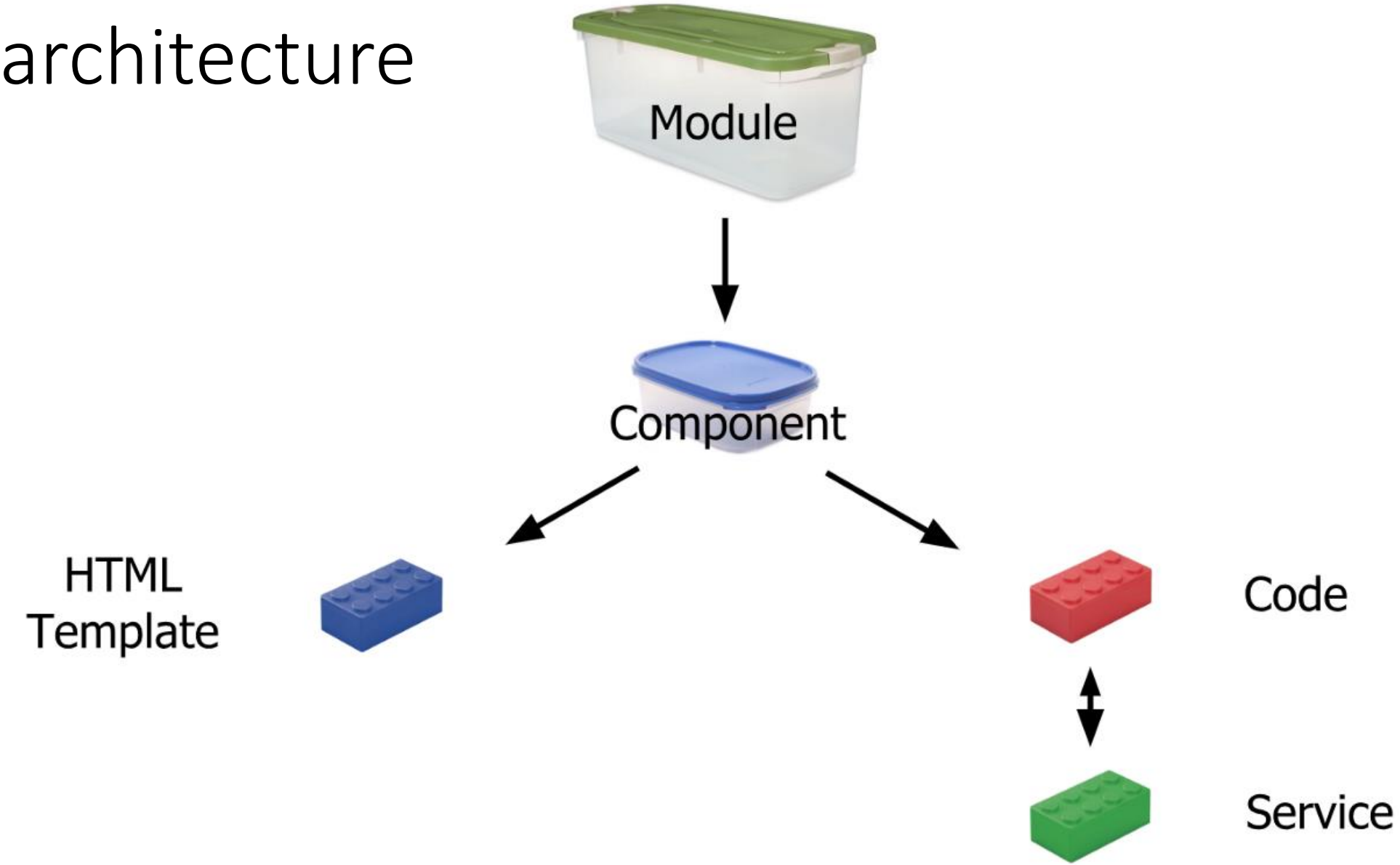

My Mail Client
Compose
Spam
Refresh
peter@demo-vp.com
Inbox
Draft
Sent
Trash
Subject
From
Date
Size
From
Subject
To
Reply
Forward
Mail content...

plan

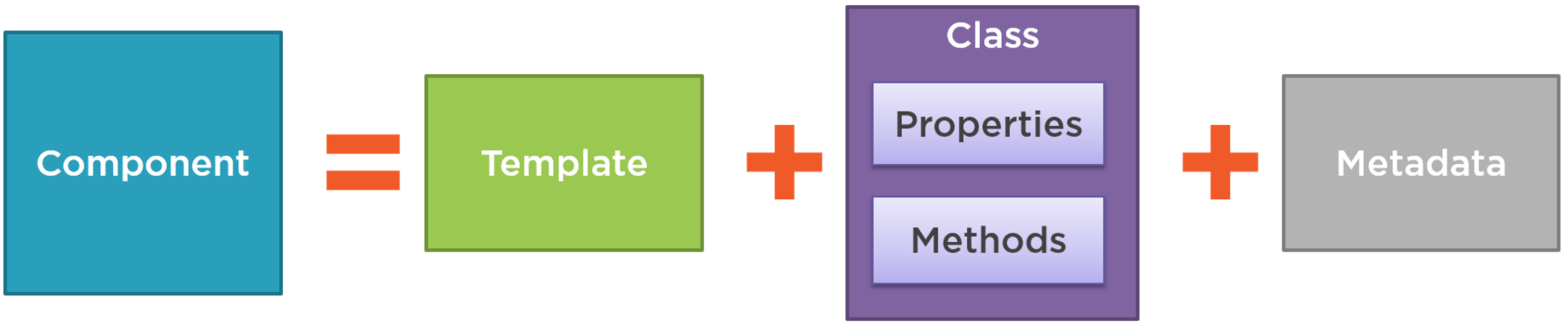

review
typescript basics
logic flow (component communication)
service logic


REVIEW

architecture



What Is a Component?

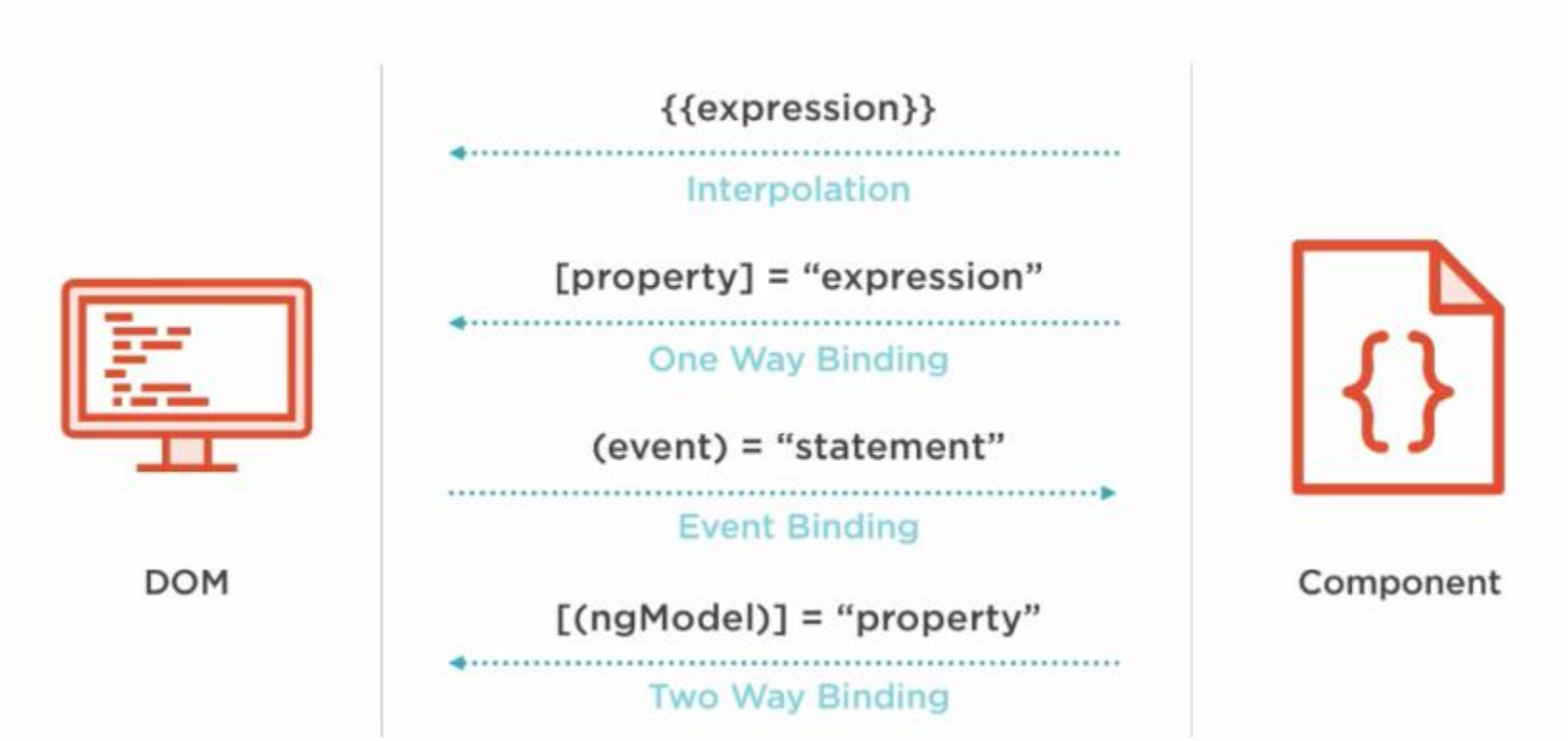


- View layout
- Created with HTML
- Includes binding and directives

- Code supporting the view
- Created with TypeScript
- Properties: data
- Methods: logic

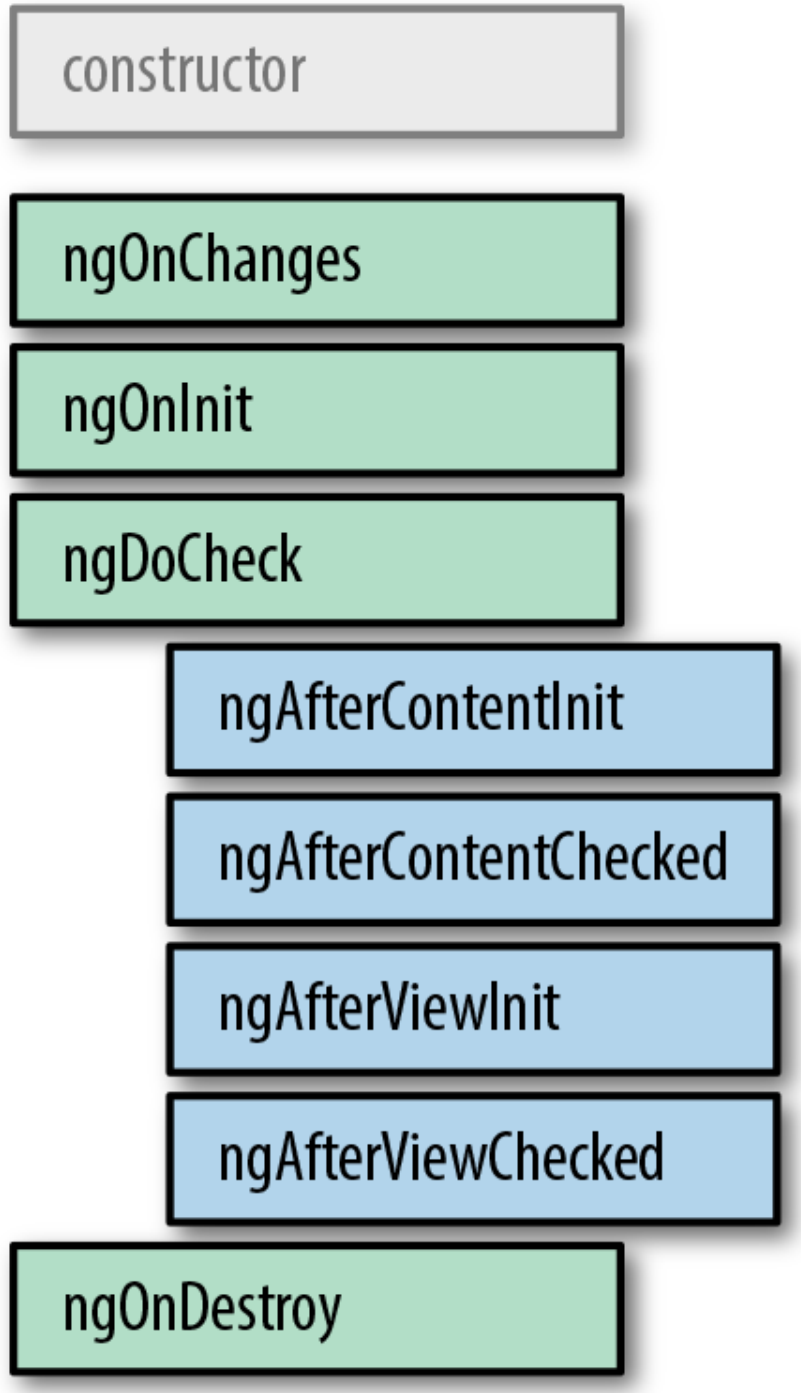
- Extra data for Angular
- Defined with a decorator

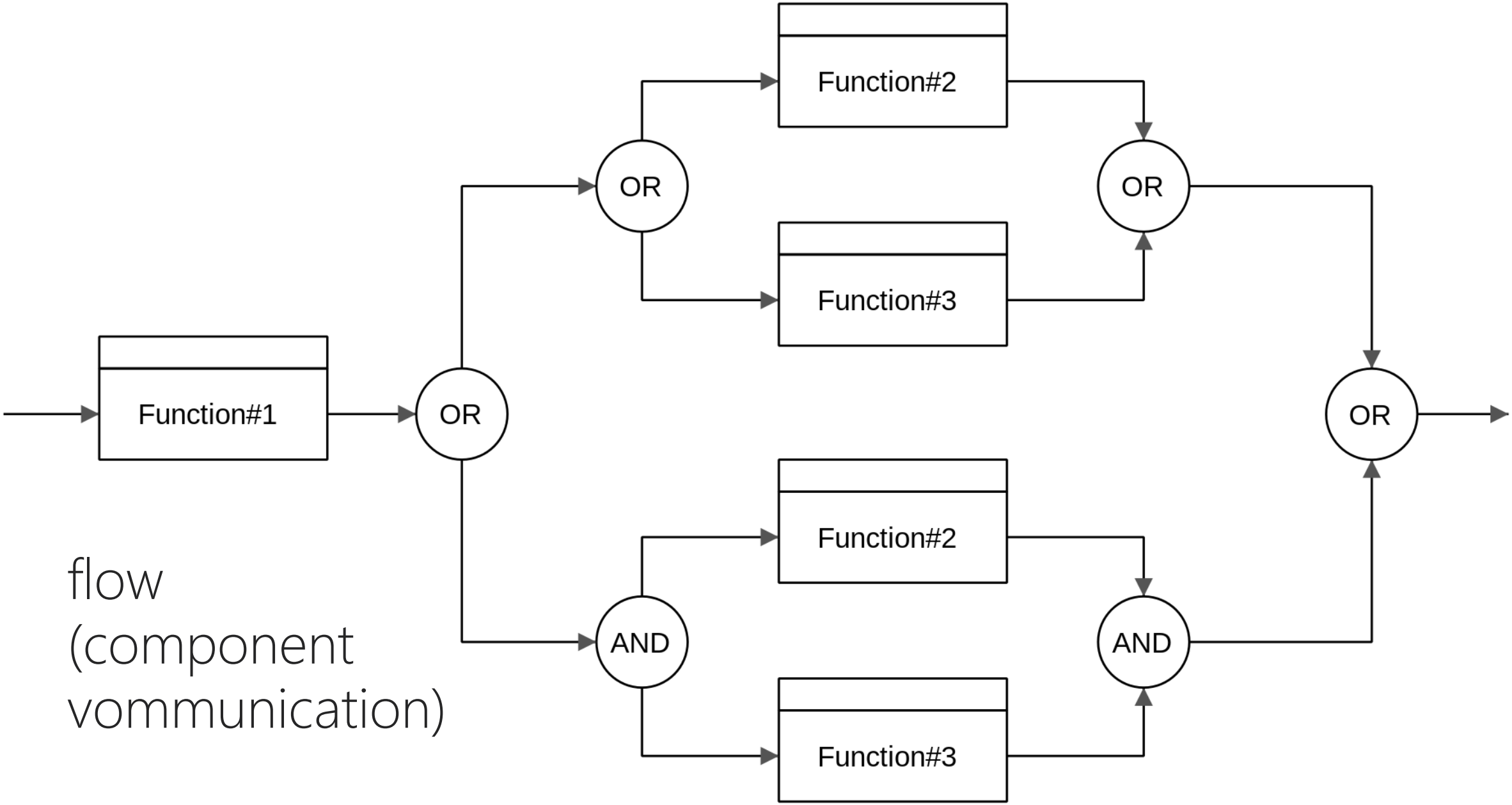
component data flows



component lifecycle

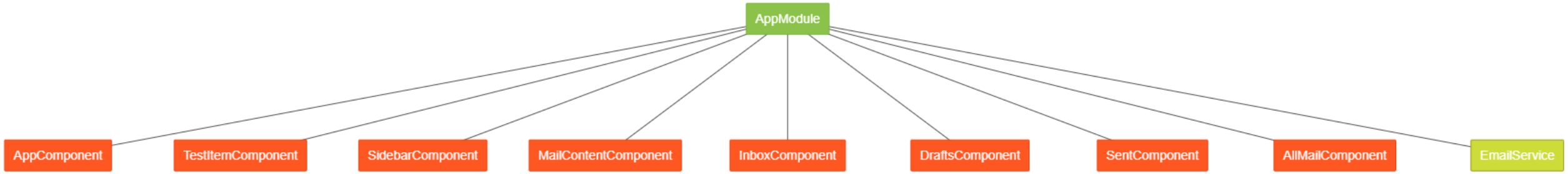
ngOnChanges	**input** property value changes
ngOnInit	Initialization step
ngDoCheck	every change detection cycle
ngOnDestroy	before destruction





flow
(component
vommunication)

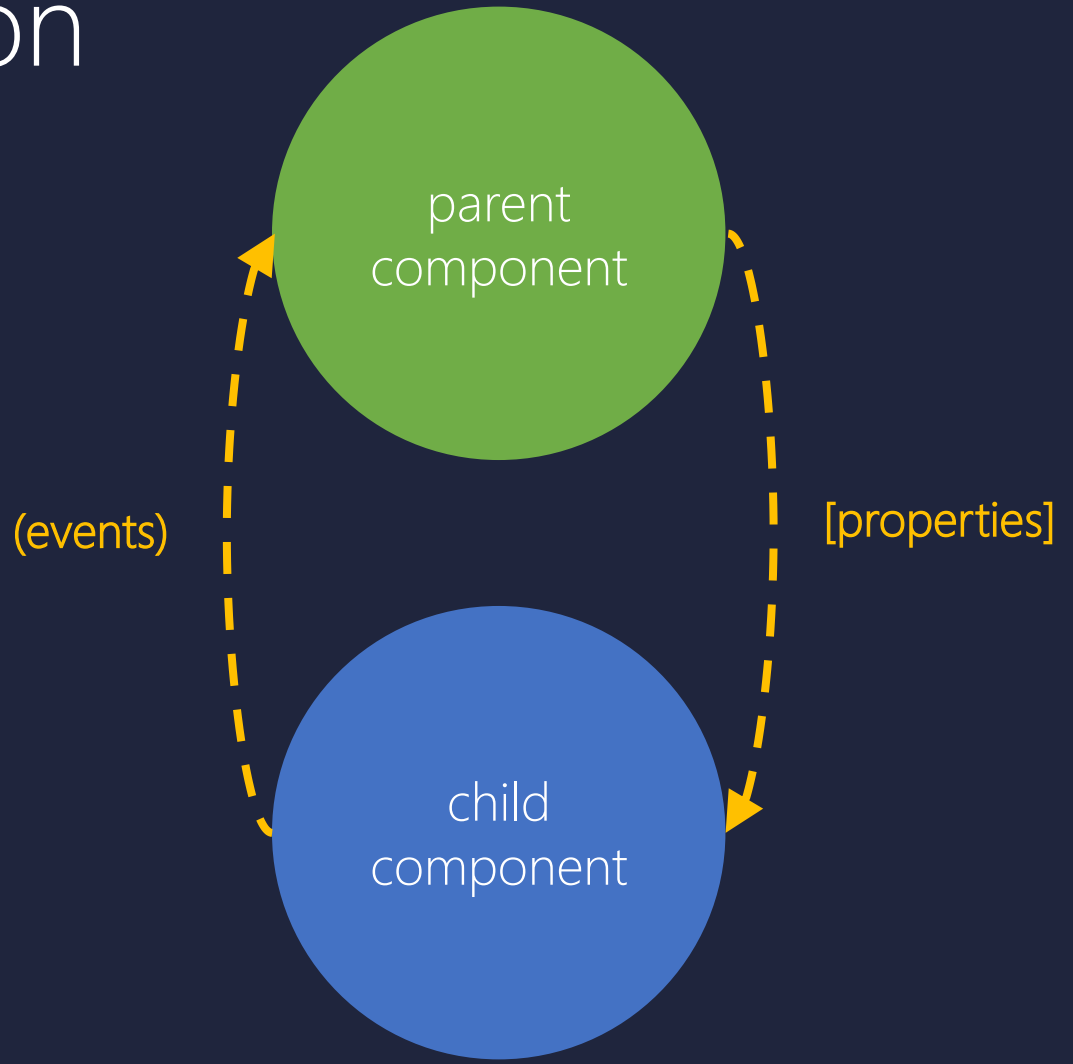
hierarchy of components




```
▼ AppComponent
    SidebarComponent
  ▼ MailContentComponent
    ▼ InboxComponent
        NgbAccordion
    router-outlet
```

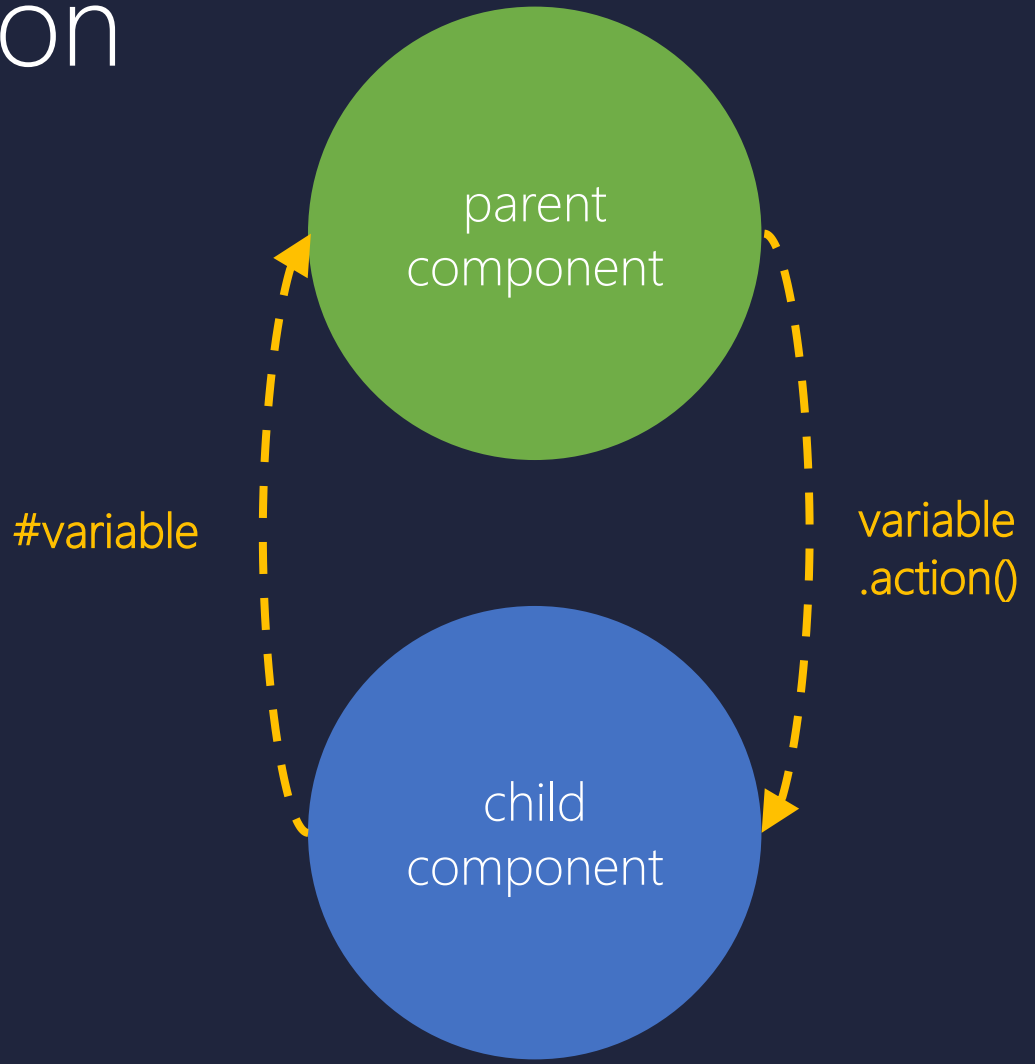
component communication

- passing data via properties
- emmiting custom events



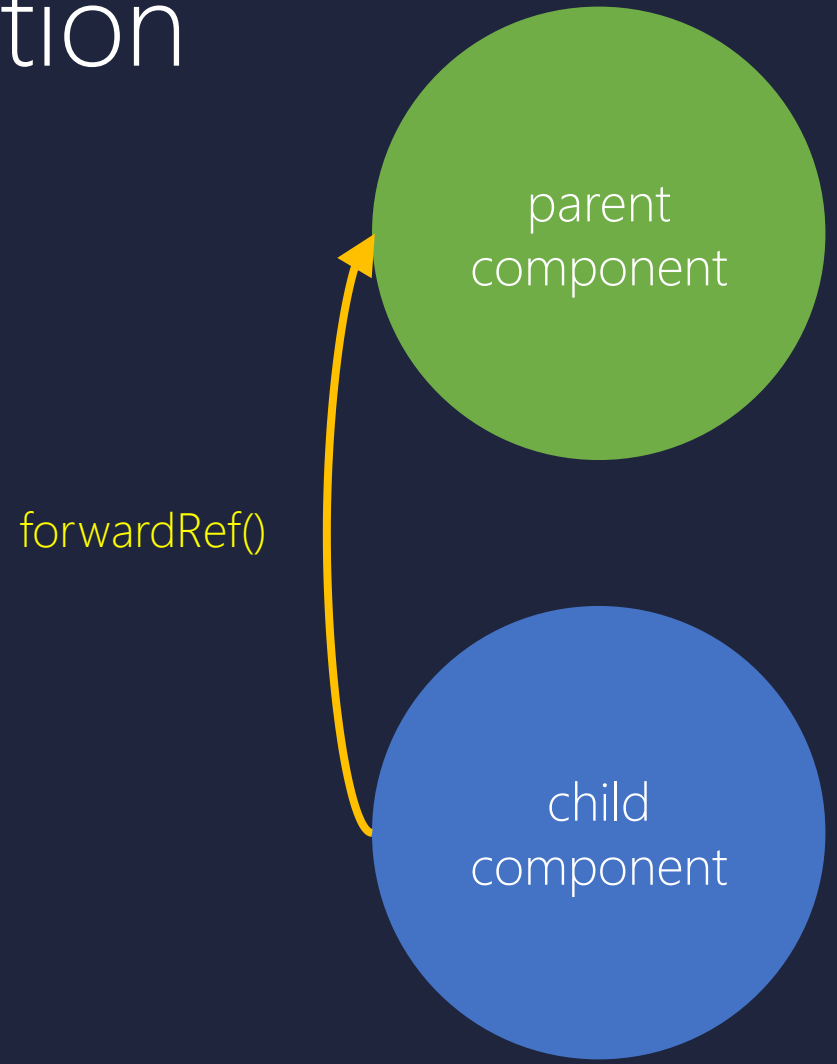
component communication

- referencing with a local variable
- querying child components
  - @ViewChildren(...)
  - @ContentChildren(...)



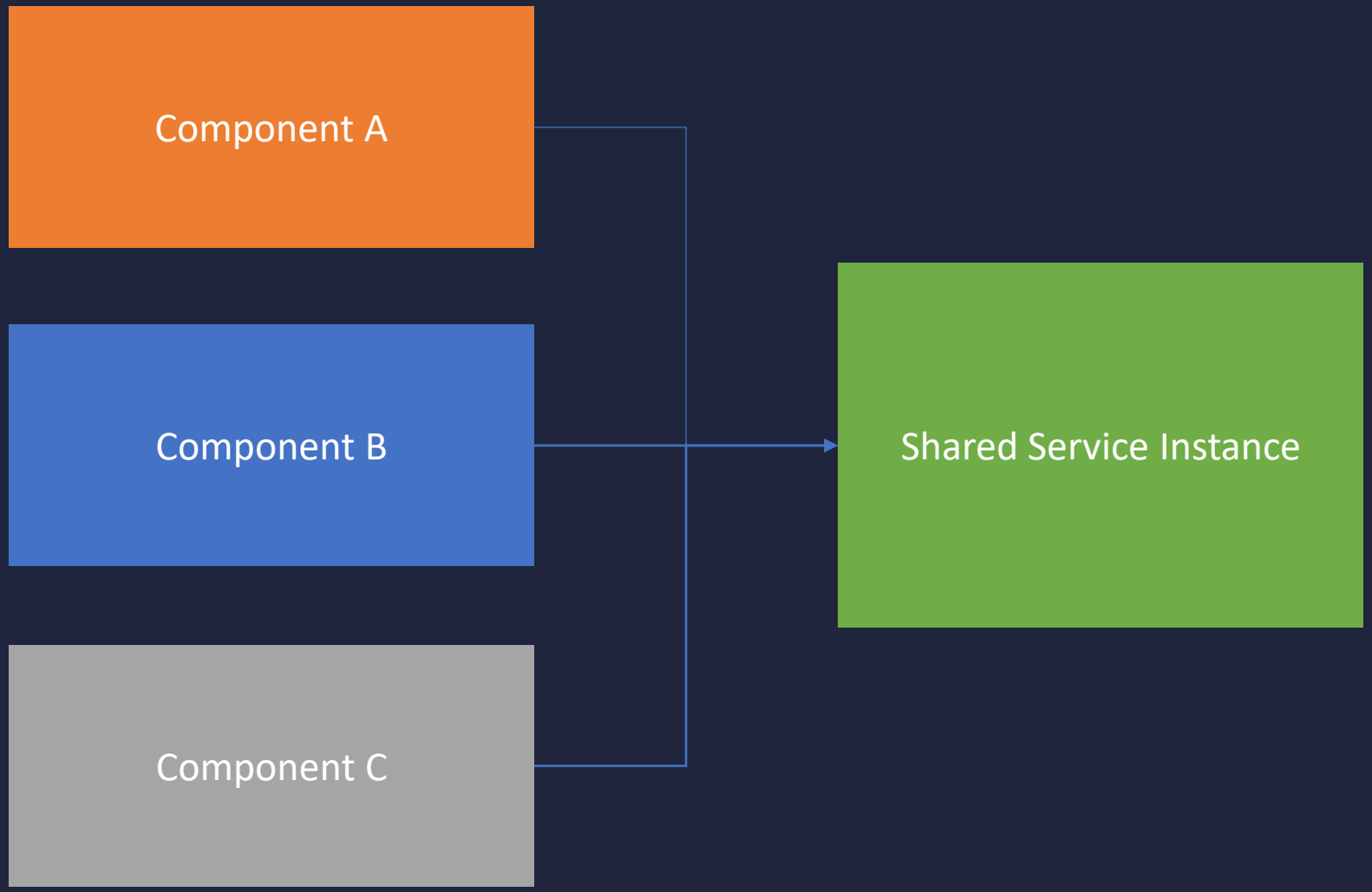
component communication

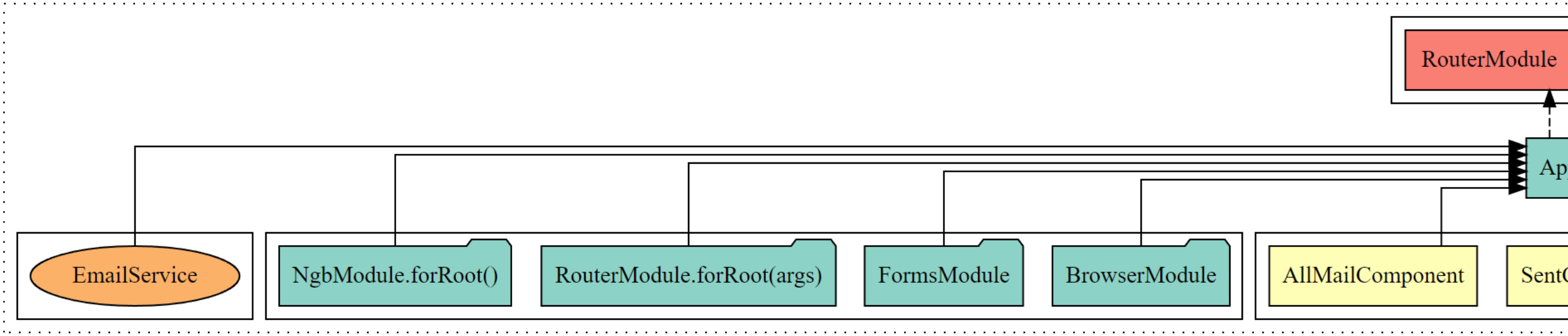
- dependency cycle with forwardRef



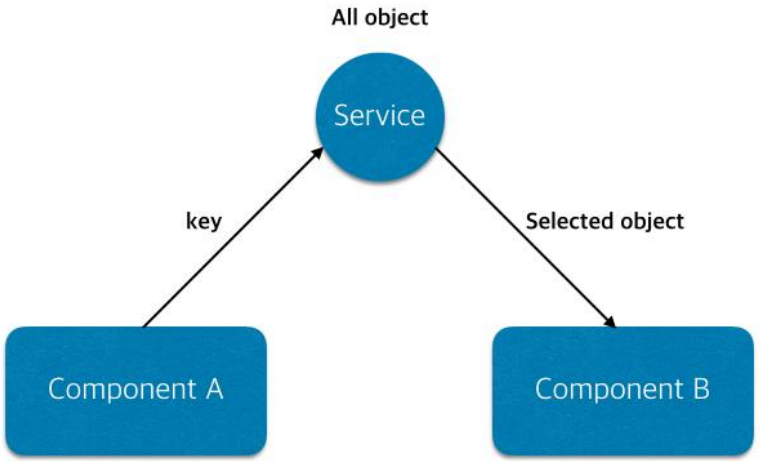
component communication

- via service



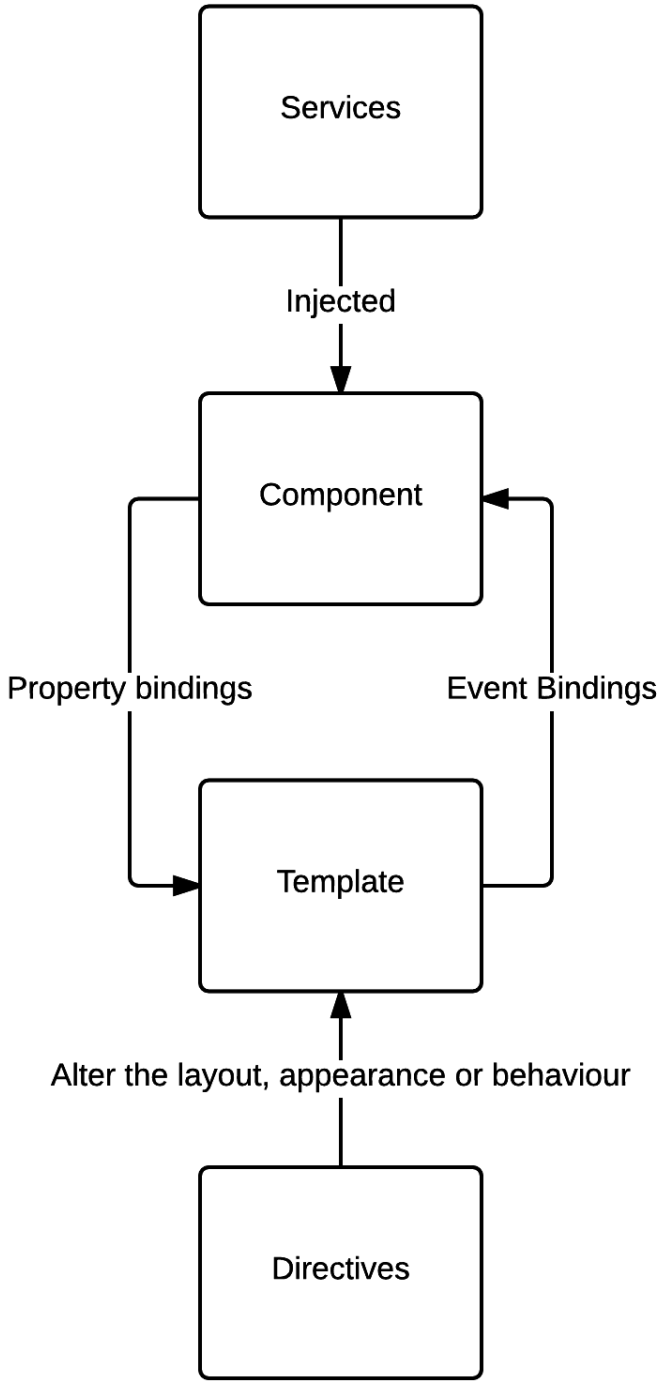

RouterModule
Ap
EmailService
NgbModule.forRoot()
RouterModule.forRoot(args)
FormsModule
BrowserModule
AllMailComponent
Sent
Leg
Declarations
Module

services


All object
Service
key
Selected object
Component A
Component B

services

- …are objects
- simplify the structure of applications
- with @Injectable decorator
- classes declare dependencies on services
- using constructor parameters



services

ng g service data

services

```
import { Injectable } from '@angular/core';

@Injectable()
export class DataService {
  constructor() { }
}
```

services

```
...
export class DataService {
    private counter: number;
    constructor() {
        this.counter = 0;
    }
    action() {
        this.counter++;
    }
}
```

services

```
...
import { DataService } from './data.service';

@NgModule({
    ...
    providers: [ DataService ],
    ...
)
```

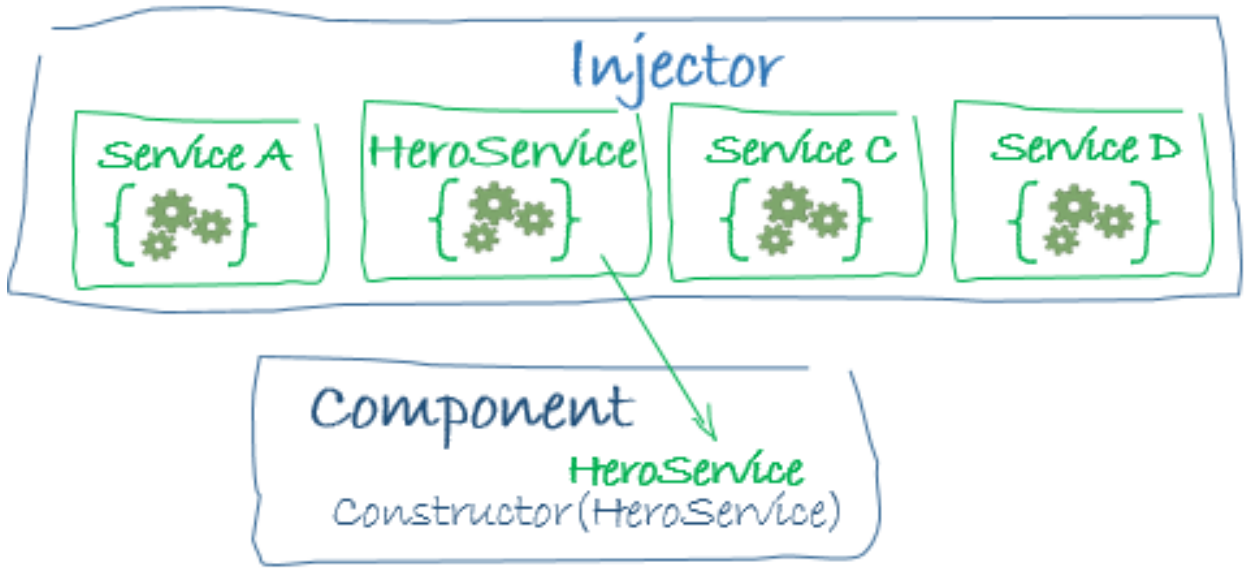
services

```
...
import { DataService } from '../data.service';

@Component(...)
export class TestComponent implements OnInit {

  constructor(private dataService : DataService,
              private testService: TestService) { }
  ...
}
```

service injection



https://github.com/Banndzior

#slack

kamil.mijacz@gmail.com

kamil.mijacz@softwarehut.com