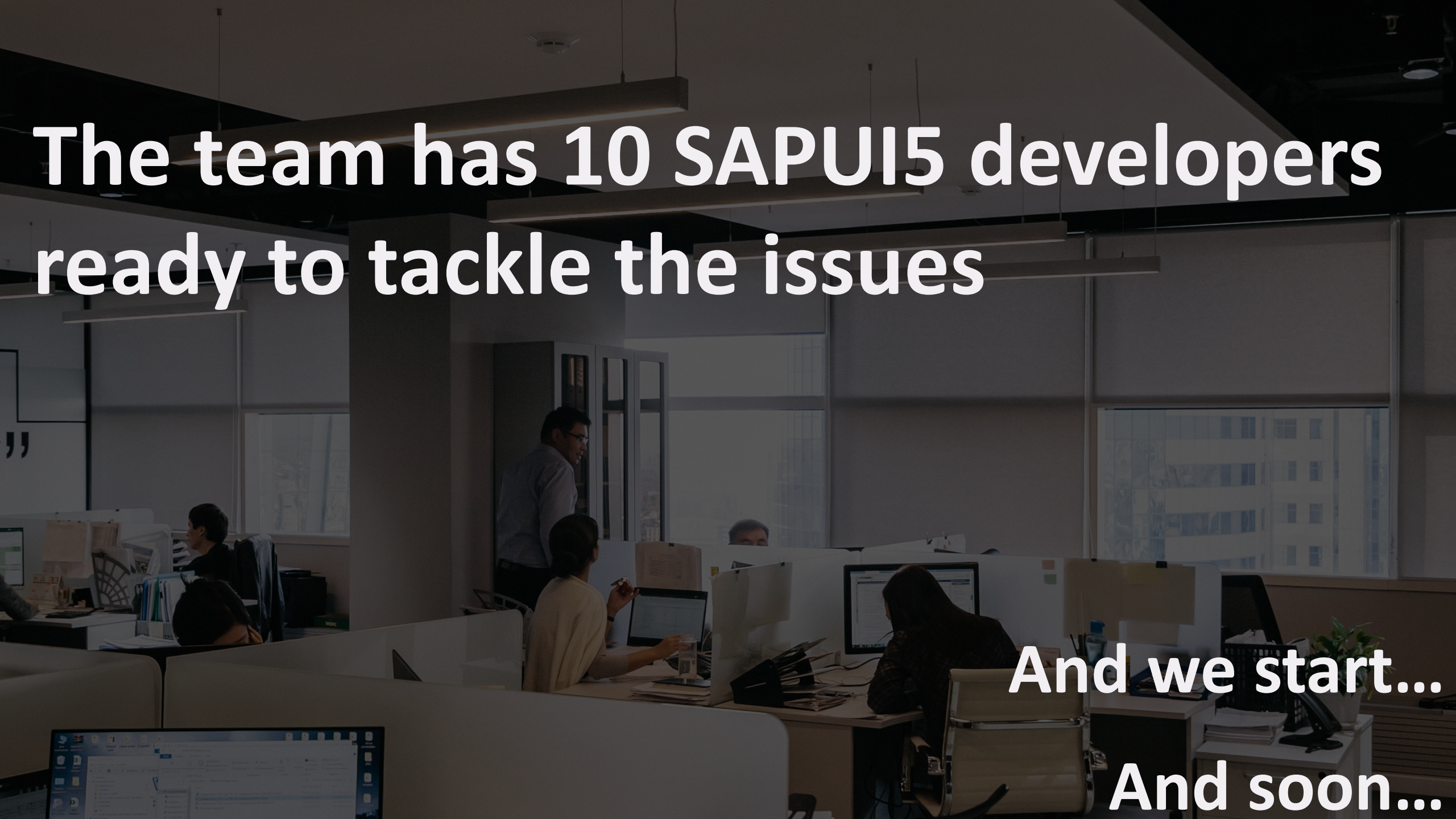
Using TypeScript and PubSub to scale freestyle UI5 apps

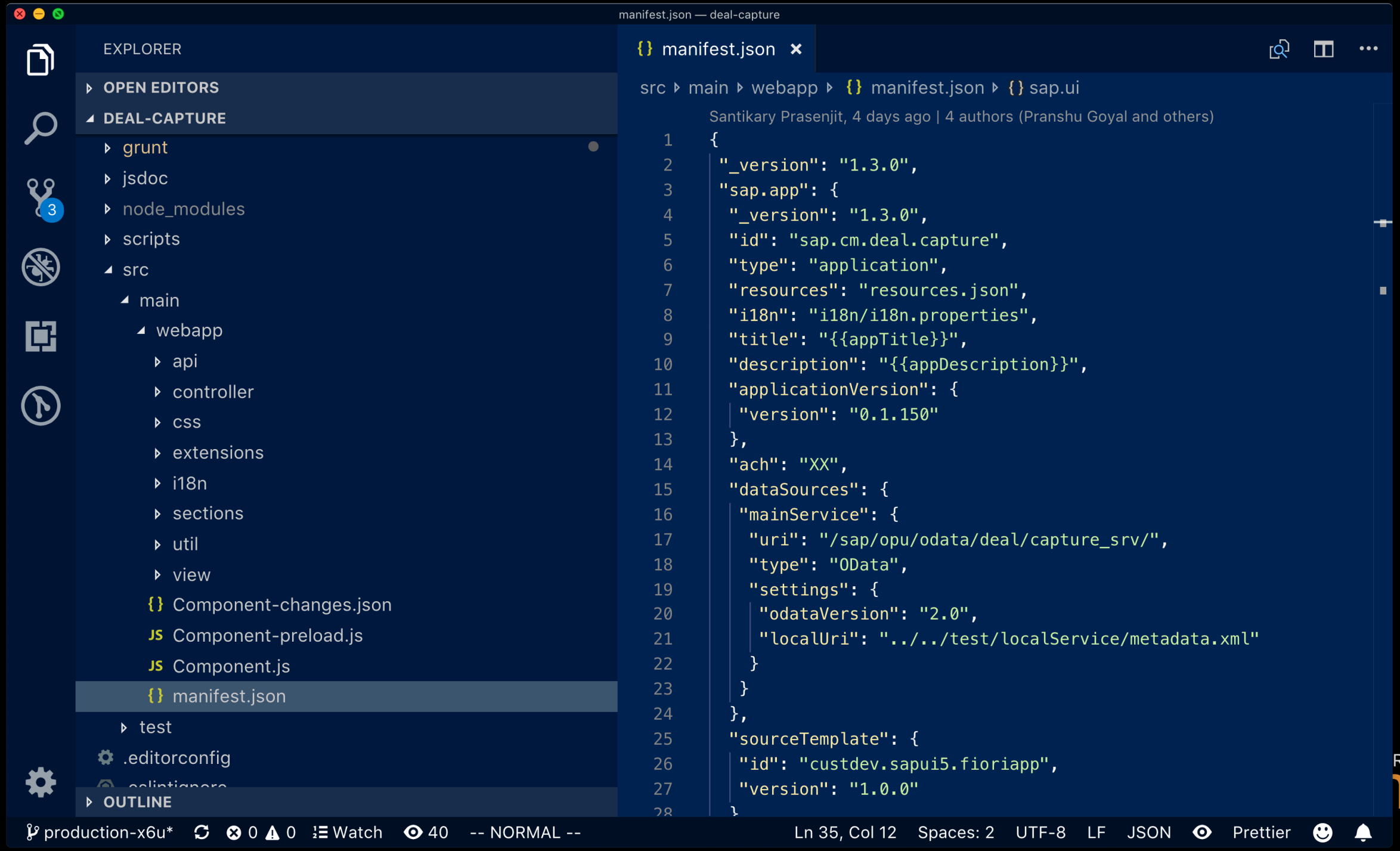
Gabriel Borges, SAP

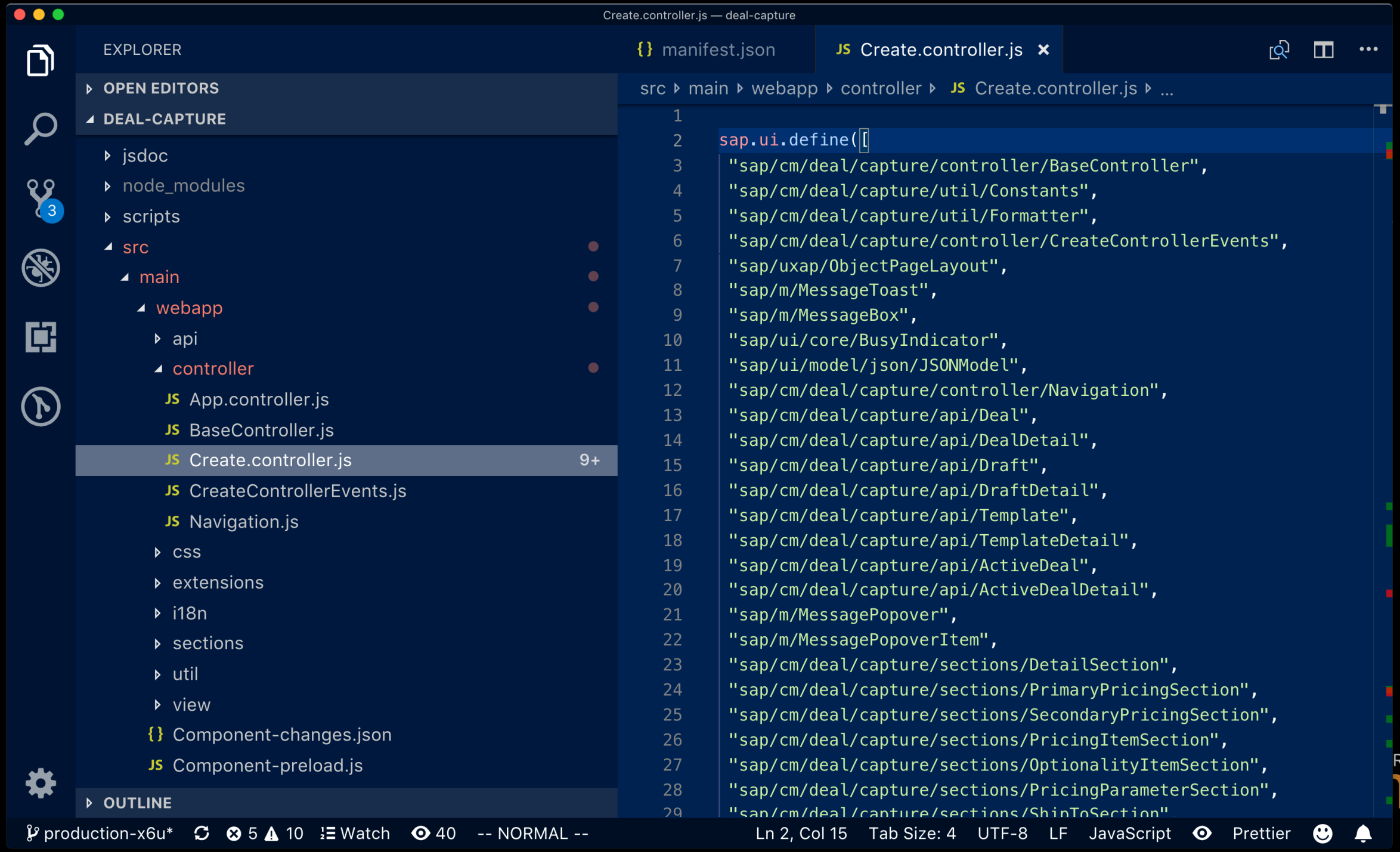
We are working in an oil trading software

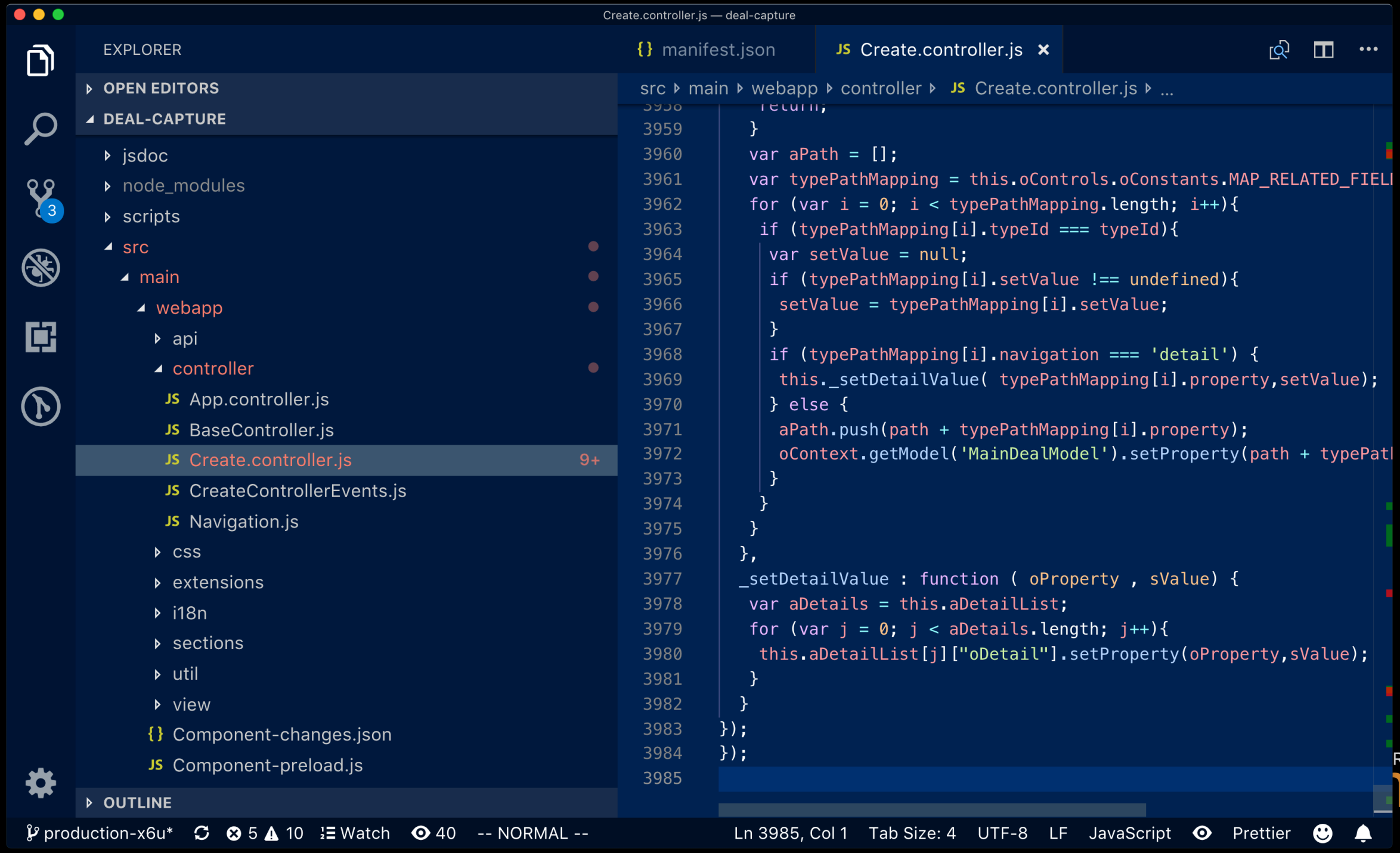
It is a big project

Lots of features built into it


The team has 10 SAPUI5 developers ready to tackle the issues
And we start…
And soon…


manifest.json — deal-capture
EXPLORER
OPEN EDITORS
DEAL-CAPTURE
grunt
jsdoc
node_modules
scripts
src
main
webapp
api
controller
css
extensions
i18n
sections
util
view
Component-changes.json
Component-preload.js
Component.js
manifest.json
test
.editorconfig
OUTLINE
manifest.json
src ▸ main ▸ webapp ▸ manifest.json ▸ sap.ui
Santikary Prasenjit, 4 days ago | 4 authors (Pranshu Goyal and others)
{
"_version": "1.3.0",
"sap.app": {
"_version": "1.3.0",
"id": "sap.cm.deal.capture",
"type": "application",
"resources": "resources.json",
"i18n": "i18n/i18n.properties",
"title": "{{appTitle}}",
"description": "{{appDescription}}",
"applicationVersion": {
"version": "0.1.150"
},
"ach": "XX",
"dataSources": {
"mainService": {
"uri": "/sap/opu/odata/deal/capture_srv/",
"type": "OData",
"settings": {
"odataVersion": "2.0",
"localUri": "../../test/localService/metadata.xml"
}
}
},
"sourceTemplate": {
"id": "custdev.sapui5.fioriapp",
"version": "1.0.0"
}
production-x6u*
0 0
Watch
40
-- NORMAL --
Ln 35, Col 12
Spaces: 2
UTF-8
LF
JSON
Prettier


Create.controller.js — deal-capture
EXPLORER
manifest.json
Create.controller.js
OPEN EDITORS
DEAL-CAPTURE
jsdoc
node_modules
scripts
src
main
webapp
api
controller
App.controller.js
BaseController.js
Create.controller.js 9+
CreateControllerEvents.js
Navigation.js
css
extensions
i18n
sections
util
view
Component-changes.json
Component-preload.js
OUTLINE
src ▸ main ▸ webapp ▸ controller ▸ Create.controller.js ▸ ...
sap.ui.define([
"sap/cm/deal/capture/controller/BaseController",
"sap/cm/deal/capture/util/Constants",
"sap/cm/deal/capture/util/Formatter",
"sap/cm/deal/capture/controller/CreateControllerEvents",
"sap/uxap/ObjectPageLayout",
"sap/m/MessageToast",
"sap/m/MessageBox",
"sap/ui/core/BusyIndicator",
"sap/ui/model/json/JSONModel",
"sap/cm/deal/capture/controller/Navigation",
"sap/cm/deal/capture/api/Deal",
"sap/cm/deal/capture/api/DealDetail",
"sap/cm/deal/capture/api/Draft",
"sap/cm/deal/capture/api/DraftDetail",
"sap/cm/deal/capture/api/Template",
"sap/cm/deal/capture/api/TemplateDetail",
"sap/cm/deal/capture/api/ActiveDeal",
"sap/cm/deal/capture/api/ActiveDealDetail",
"sap/m/MessagePopover",
"sap/m/MessagePopoverItem",
"sap/cm/deal/capture/sections/DetailSection",
"sap/cm/deal/capture/sections/PrimaryPricingSection",
"sap/cm/deal/capture/sections/SecondaryPricingSection",
"sap/cm/deal/capture/sections/PricingItemSection",
"sap/cm/deal/capture/sections/OptionalityItemSection",
"sap/cm/deal/capture/sections/PricingParameterSection",
production-x6u*
5 10
Watch
40
-- NORMAL --
Ln 2, Col 15
Tab Size: 4
UTF-8
LF
JavaScript
Prettier


Create.controller.js — deal-capture
EXPLORER
manifest.json
Create.controller.js
OPEN EDITORS
DEAL-CAPTURE
jsdoc
node_modules
scripts
src
main
webapp
api
controller
App.controller.js
BaseController.js
Create.controller.js 9+
CreateControllerEvents.js
Navigation.js
css
extensions
i18n
sections
util
view
Component-changes.json
Component-preload.js
OUTLINE
src ▸ main ▸ webapp ▸ controller ▸ Create.controller.js ▸ ...
3959 }
3960 var aPath = [];
3961 var typePathMapping = this.oControls.oConstants.MAP_RELATED_FIELI
3962 for (var i = 0; i < typePathMapping.length; i++){
3963 if (typePathMapping[i].typeId === typeId){
3964 var setValue = null;
3965 if (typePathMapping[i].setValue !== undefined){
3966 setValue = typePathMapping[i].setValue;
3967 }
3968 if (typePathMapping[i].navigation === 'detail') {
3969 this._setDetailValue(typePathMapping[i].property,setValue);
3970 } else {
3971 aPath.push(path + typePathMapping[i].property);
3972 oContext.getModel('MainDealModel').setProperty(path + typePat
3973 }
3974 }
3975 }
3976 },
3977 _setDetailValue : function (oProperty , sValue) {
3978 var aDetails = this.aDetailList;
3979 for (var j = 0; j < aDetails.length; j++){
3980 this.aDetailList[j]["oDetail"].setProperty(oProperty,sValue);
3981 }
3982 }
3983 });
3984 });
3985
production-x6u*
5 10
Watch
40
-- NORMAL --
Ln 3985, Col 1
Tab Size: 4
UTF-8
LF
JavaScript
Prettier

```
3979                for (var j = 0
3980                    this.aDetailL
3981                }
3982            }
3983        });
3984    });
3985
```

Thank you

Gabriel Borges
Fiori Architect, SAP Innovative Business Solutions
SAP Labs Latin America – São Leopoldo, RS, Brazil

gabriel.borges@sap.com

Twitter: @psidium_
Github: Psidium

UI5con ON AIR

Agenda:

Why does this happen?

How to fix it?

Some rules of thumb

Code

Final regards

Way too many codebases look like that in the wild.

- Controllers getting out of hand;
- Not a single line of testing;
- Missing deadlines ;
- Sprints failing;
- Projects canceled.

Why does this happen?


It boils down to state management.

Welcome to the MVC village

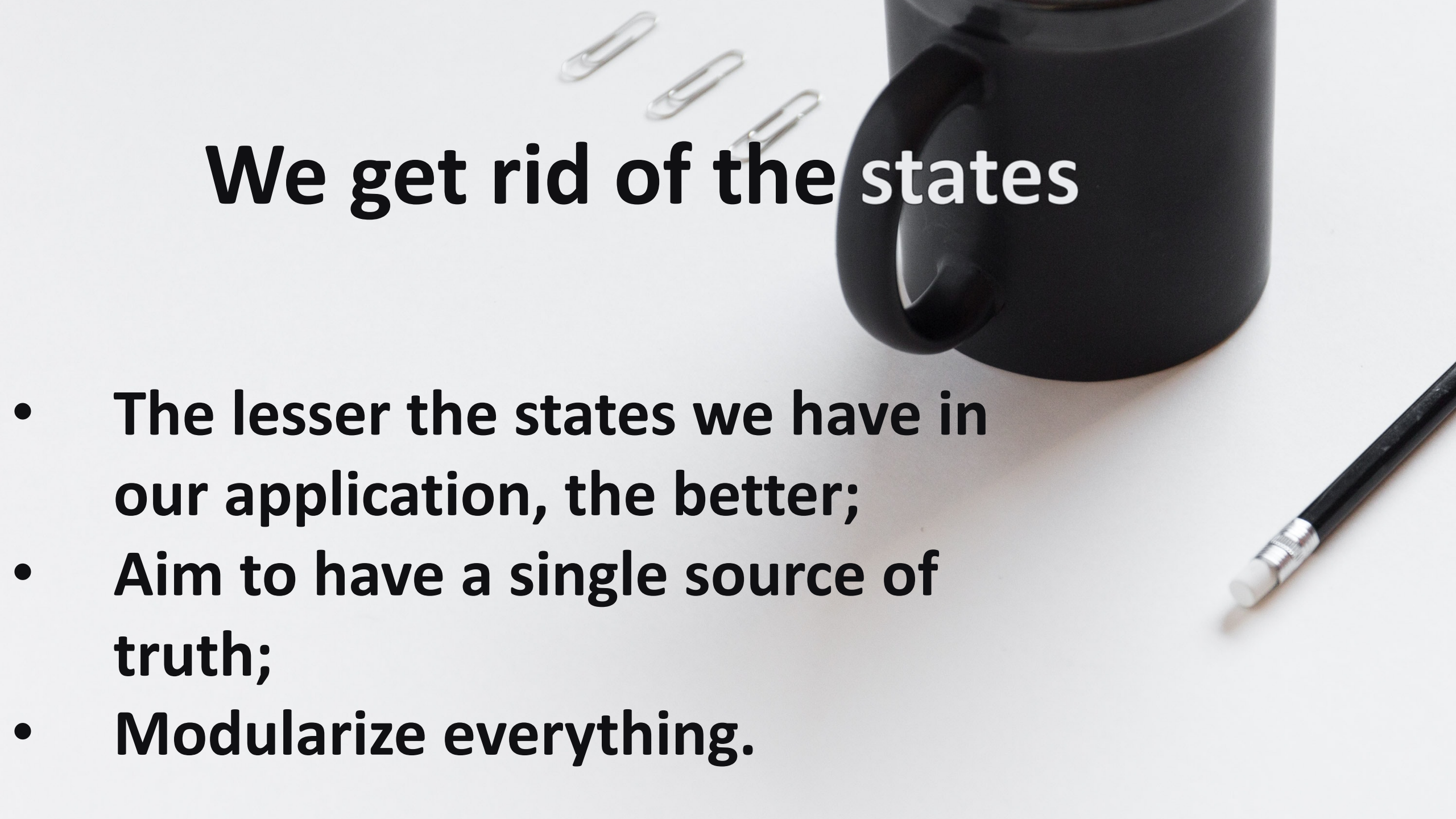

M
The Model is the state of the application

The View is responsible for displaying the state of the application, but has a state of its own as well

The Controller is responsible for changing the state of the application, but has a state of its own, and can change the state of the View as well


M
V
C

How to fix it?



We get rid of the states

- The lesser the states we have in our application, the better;
- Aim to have a single source of truth;
- Modularize everything.

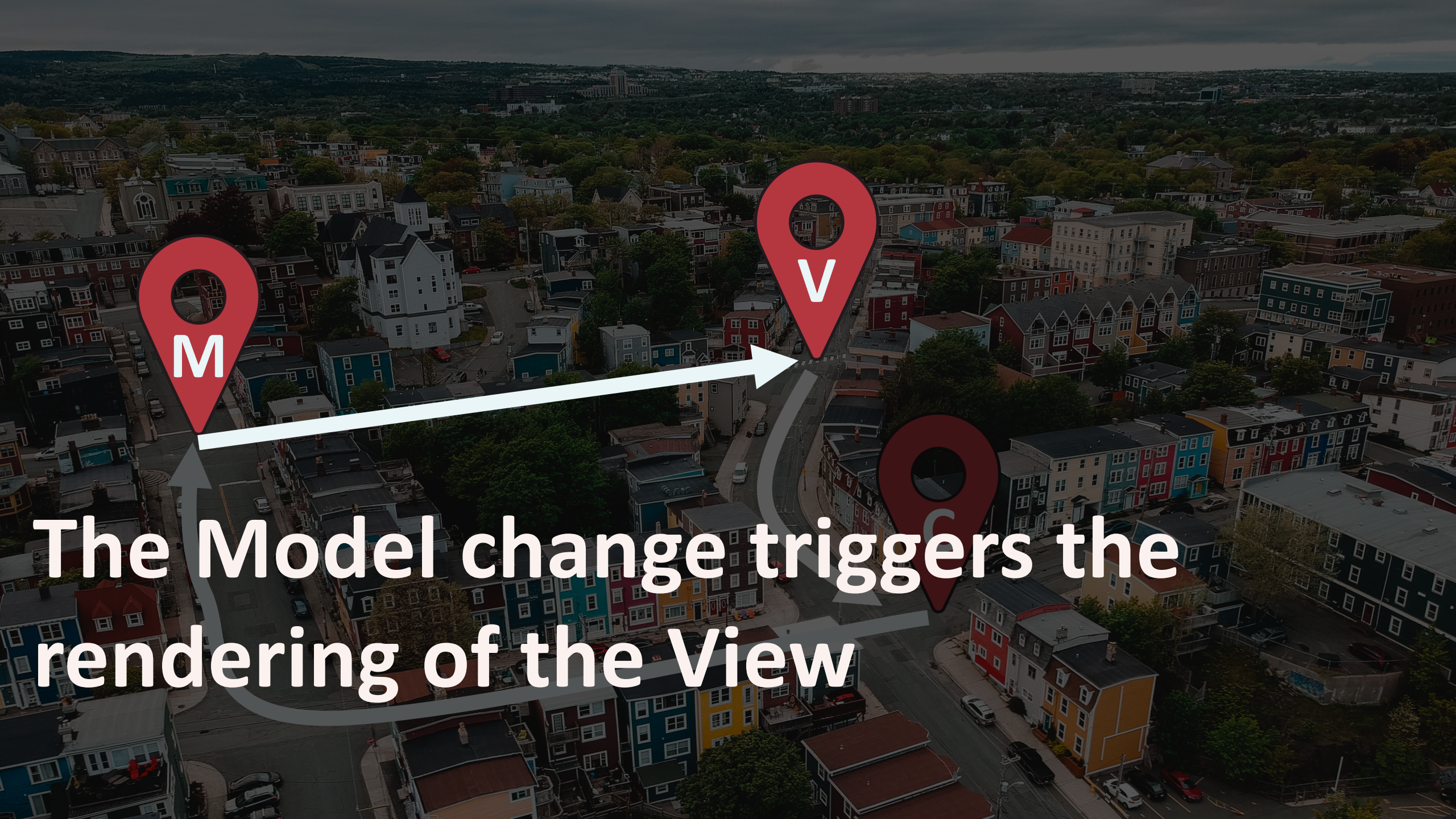
Redeveloping our village


M
V
C

The View can send events to the Controller, and we will avoid all internal states of the view

The controller is forbidden to access the internal state of the View

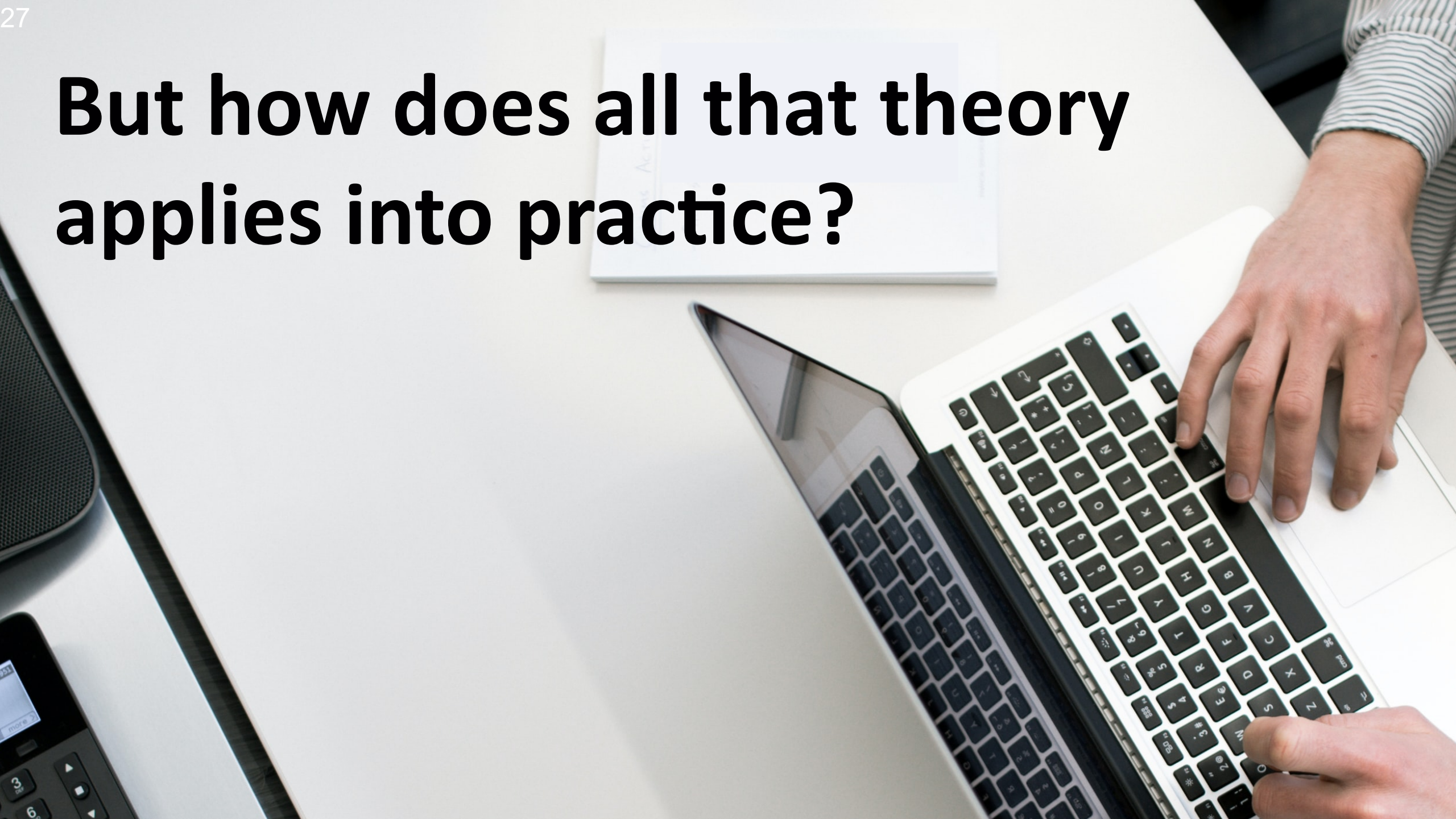

The Controller will read and write on the Model, and that is the only way to interact with the state of the application
M
C



The Model change triggers the rendering of the View

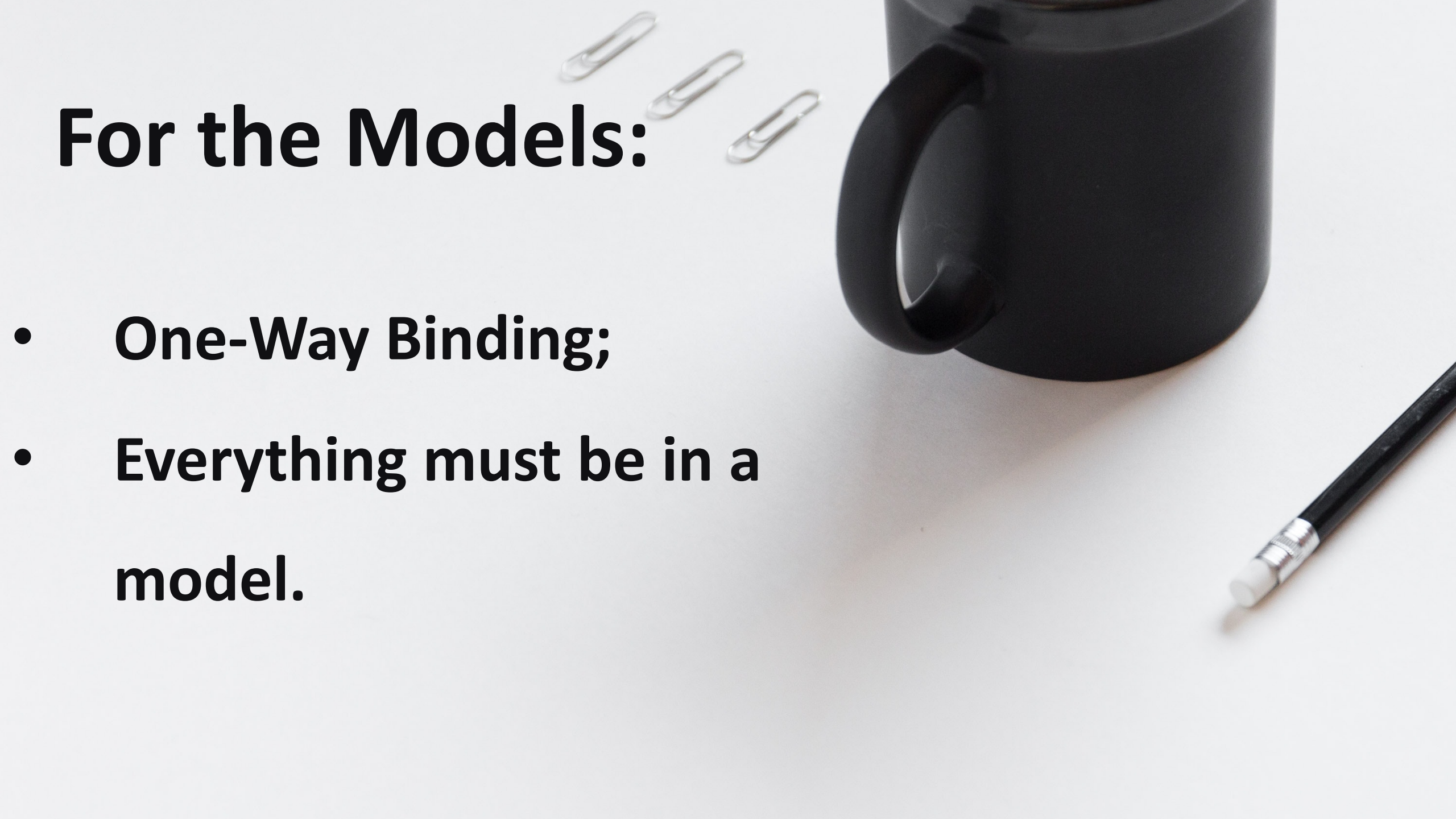

M
V
C

But how does all that theory applies into practice?



For the Models:

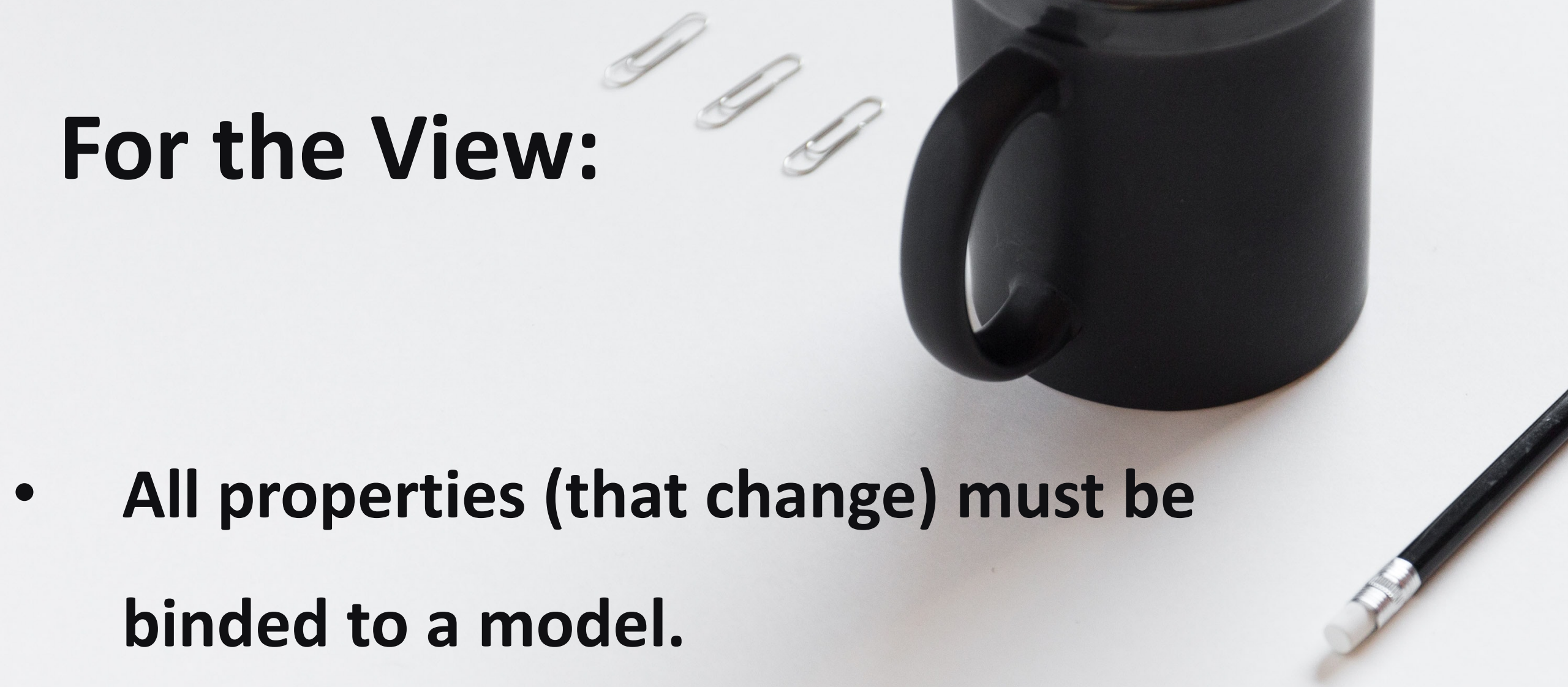
- One-Way Binding;
- Everything must be in a model.



For the Controller:

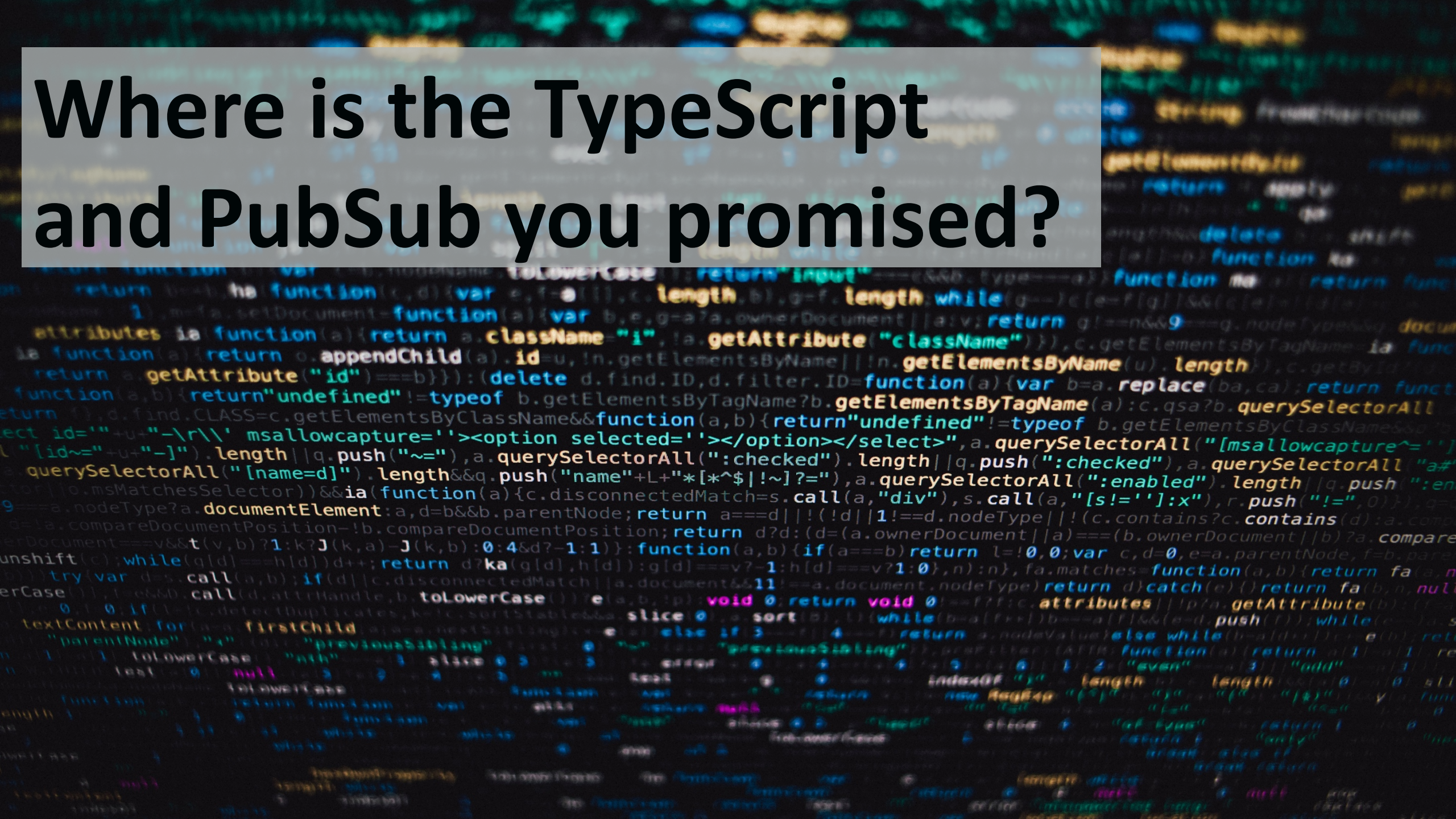
- `byId` is forbidden, to avoid touching the View's state;
- No reference to the View, or Controls, ever;
- Only for removing data from UI5 and sending to your classes.

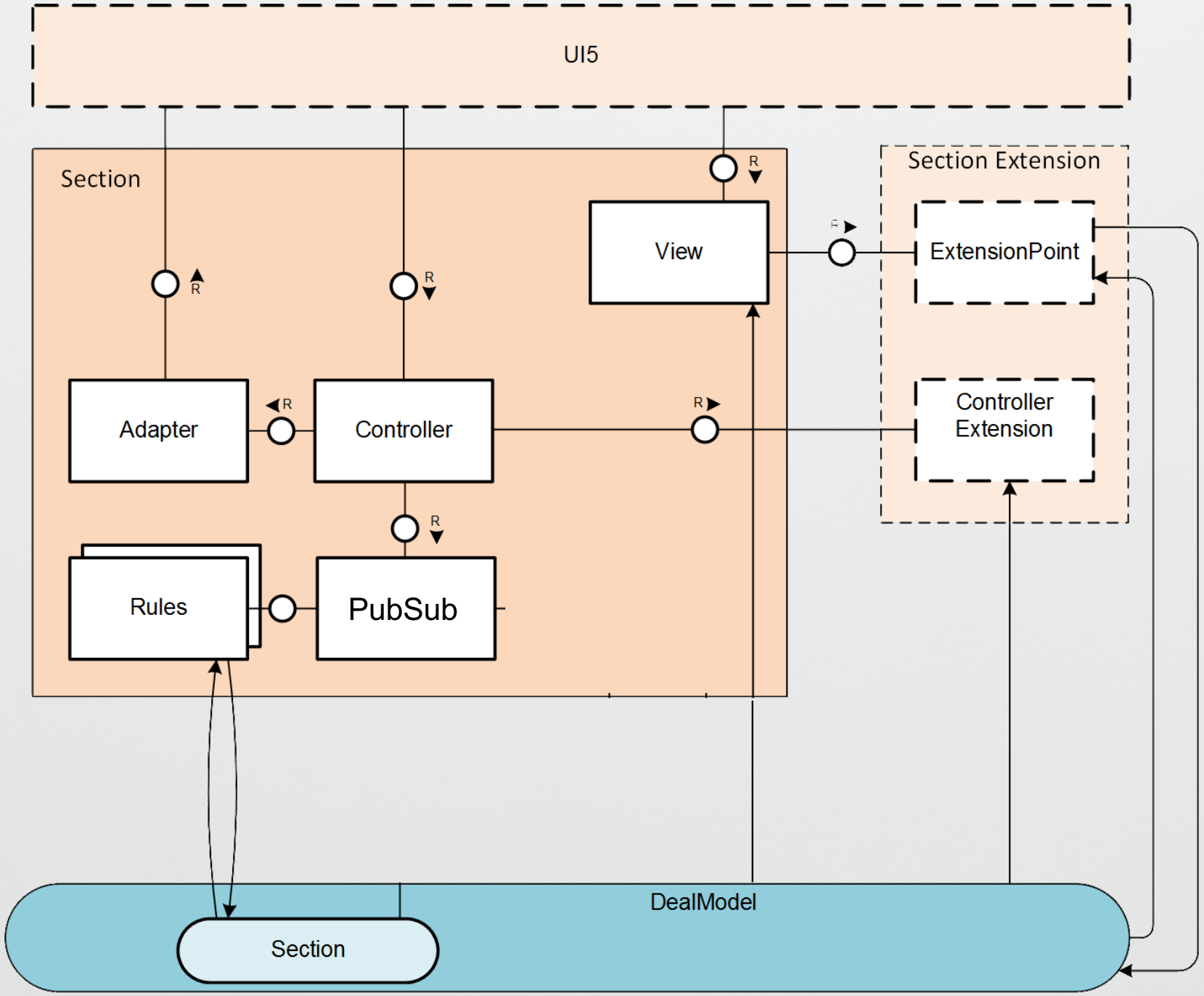
For the View:

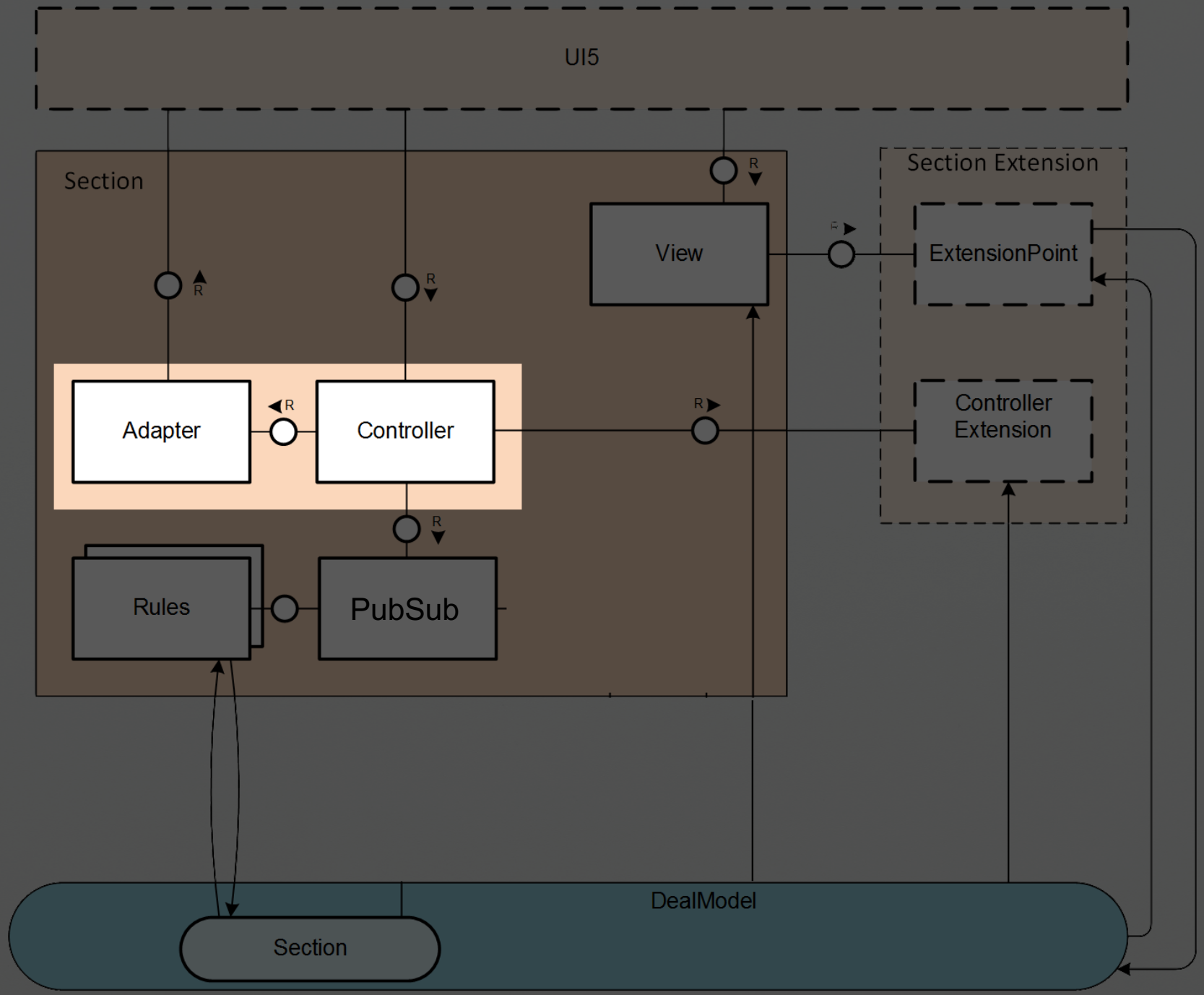


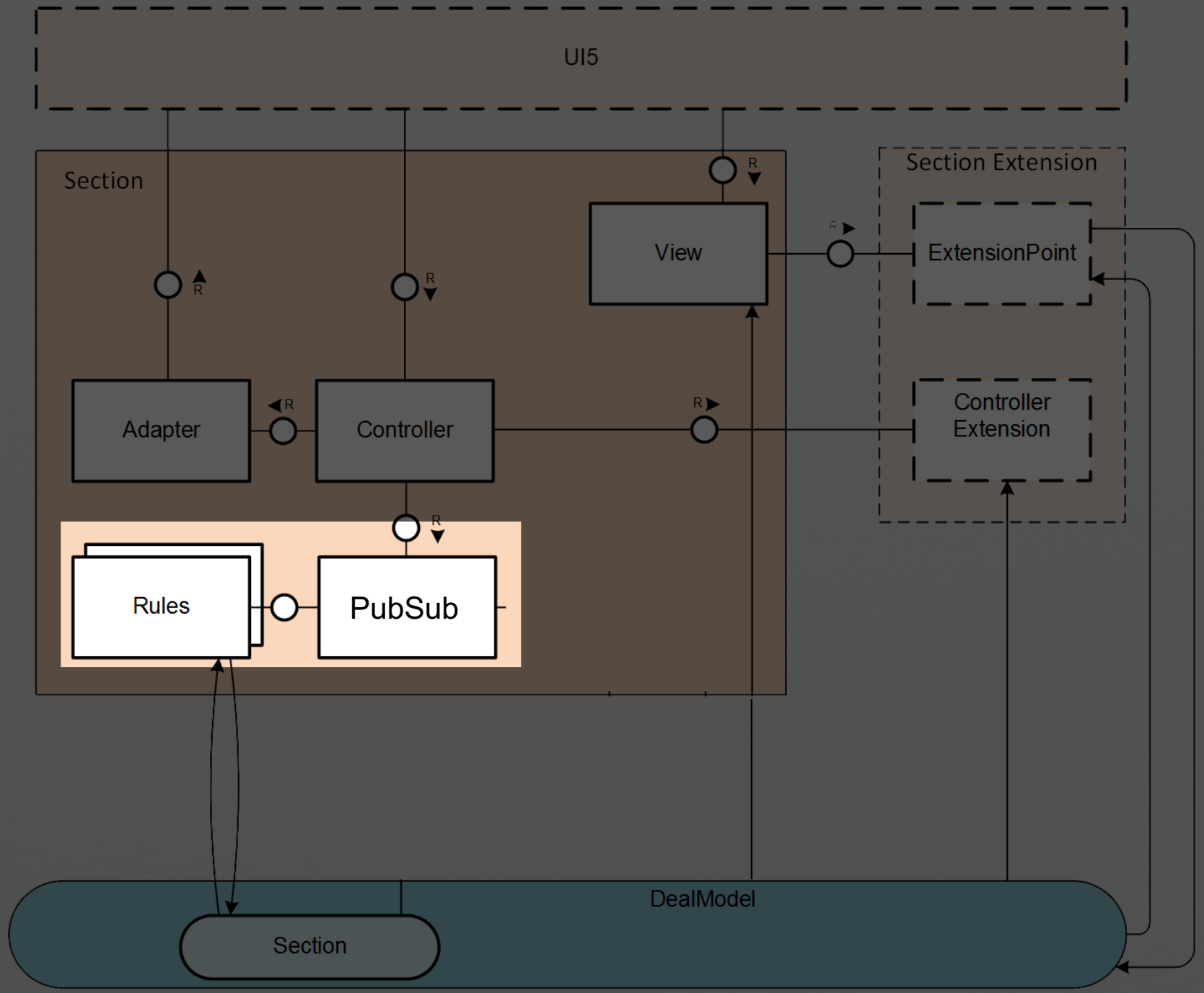
- All properties (that change) must be binded to a model.

Where is the TypeScript and PubSub you promised?




UI5
Section
R
View
Section Extension
ExtensionPoint
R
R
Adapter
R
Controller
R
Controller
Extension
R
Rules
PubSub
DealModel
Section


UI5
Section
R
R
R
View
R
Section Extension
ExtensionPoint
Adapter
R
Controller
R
Controller
Extension
R
Rules
PubSub
DealModel
Section


UI5
Section
Section Extension
View
ExtensionPoint
Adapter
Controller
Controller Extension
Rules
PubSub
DealModel
Section
R

TypeScript

- A Superset of JavaScript;
- Made by Microsoft;
- Imagine the power of typed languages allied with the versatility of JavaScript.

PubSub:

- Publish & Subscribe;
- Wrapper around SAPUI5’s EventBus;
- Optimized for TypeScript.

```
    .fromEvent("Detail.CreateDetailPressed")
    .publish({
      createDetailPressedParameters,
      detailComposite
    });
}

public onDealTransactionTypeChanged(): void {
  const headerComposite = this.adapter.getHeaderComposite();
  const transactionType = this.adapter.getDealTransactionType();
  ServiceLocator.getInstance()
    .locate(LocatableServices.EventBus)
    .fromEvent("Header.DealTransactionTypeChanged")
    .publish({
      transactionType,
      headerComposite
    });
}

public onErrorMessagePopoverItemSelected(target: Target): void {
  ServiceLocator.getInstance()
    .locate(LocatableServices.EventBus)
    .fromEvent("Capture.ErrorMessageClicked")
```

```
  ServiceLocator.getInstance()
    .locate(LocatableServices.EventBus)
    .fromEvent("Detail.CreateDetailPres
    .publish({
      createDetailPressedParameters,
      detailComposite
    });
}

public onDealTransactionTypeChanged():
  const headerComposite = this.adapter.
  const transactionType = this.adapter.getDealTransactionType();
  ServiceLocator.getInstance()
    .locate(LocatableServices.EventBus)
    .fromEvent("Header.DealTransactionTypeChanged")
    .publish({
      transactionType,
      headerComposite
    });
}

public onErrorMessagePopoverItemSelected(target: Target): void {
  ServiceLocator.getInstance()
    .locate(LocatableServices.EventBus)
    .fromEvent("Capture.ErrorMessageClicked")
```

```
public getDealTransactionType(): TransactionType {
  const headerAPI = this.getHeaderAPI();
  const dealTransactionType = String(
    headerAPI.getProperty("transactionType")
  ) as TransactionType;
  return dealTransactionType;
}
```

```
(method) MainAdapter.getDealTransactionType(): TransactionType
```

```
  );
  const createDetailPressedParameters = this.extractCreateDetailPressedParametersFromUI5();
  ServiceLocator.getInstance()
    .locate(LocatableServices.EventBus)
    .fromEvent("Detail.CreateDetailPressed")
    .publish({
      createDetailPressedParameters,
      detailComposite
    });
}

public
  cons
  cons
  if (transactionType === "RANDOM_STRING") {

  }
  ServiceLocator.getInstance()
    .locate(LocatableServices.EventBus)
    .fromEvent("Header.DealTransactionTypeChanged")
    .publish({
      transactionType,
      headerComposite
    });
}
```

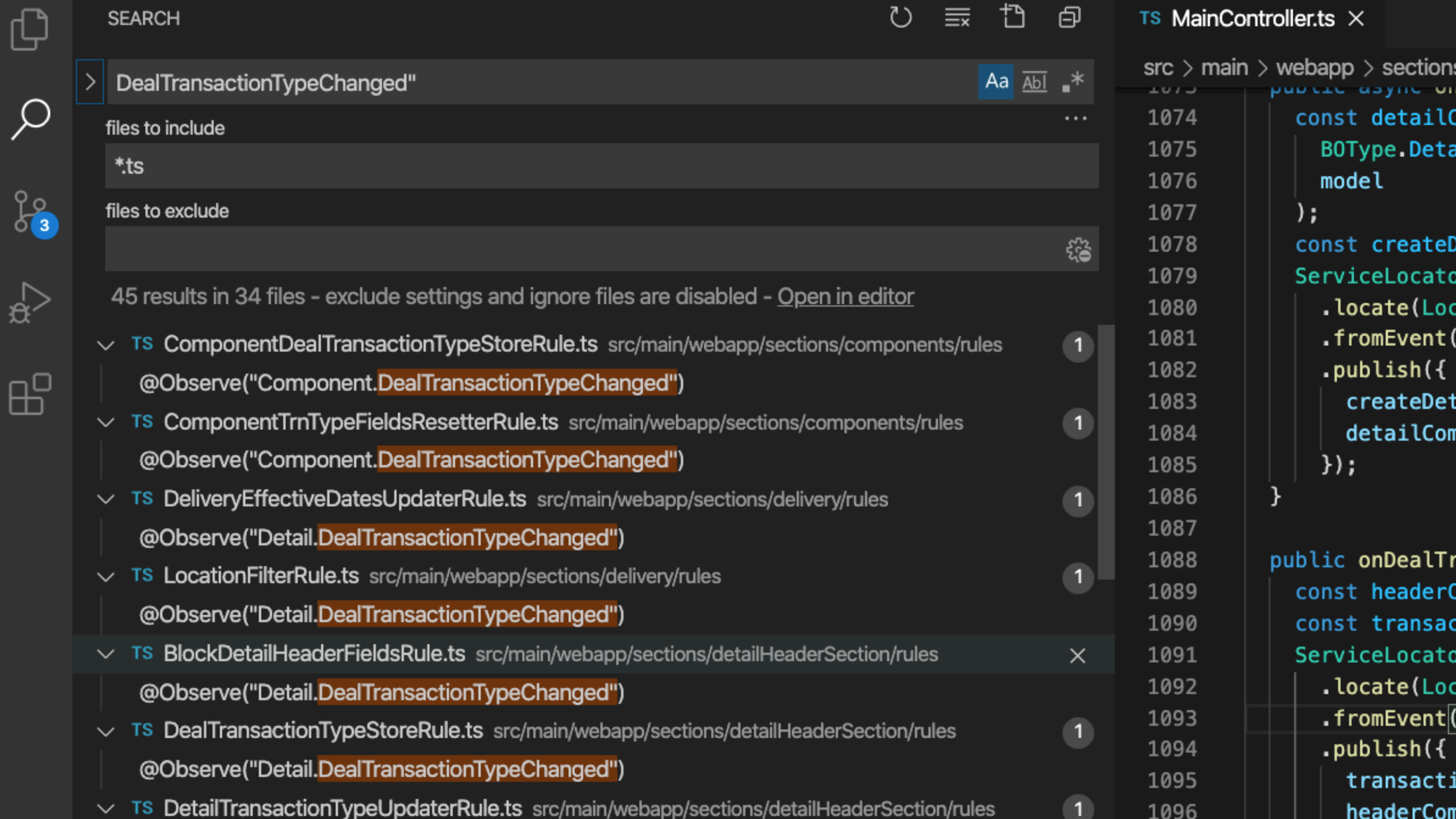
This condition will always return 'false' since the types 'TransactionType' and '"RANDOM_STRING"' have no overlap. ts(2367)

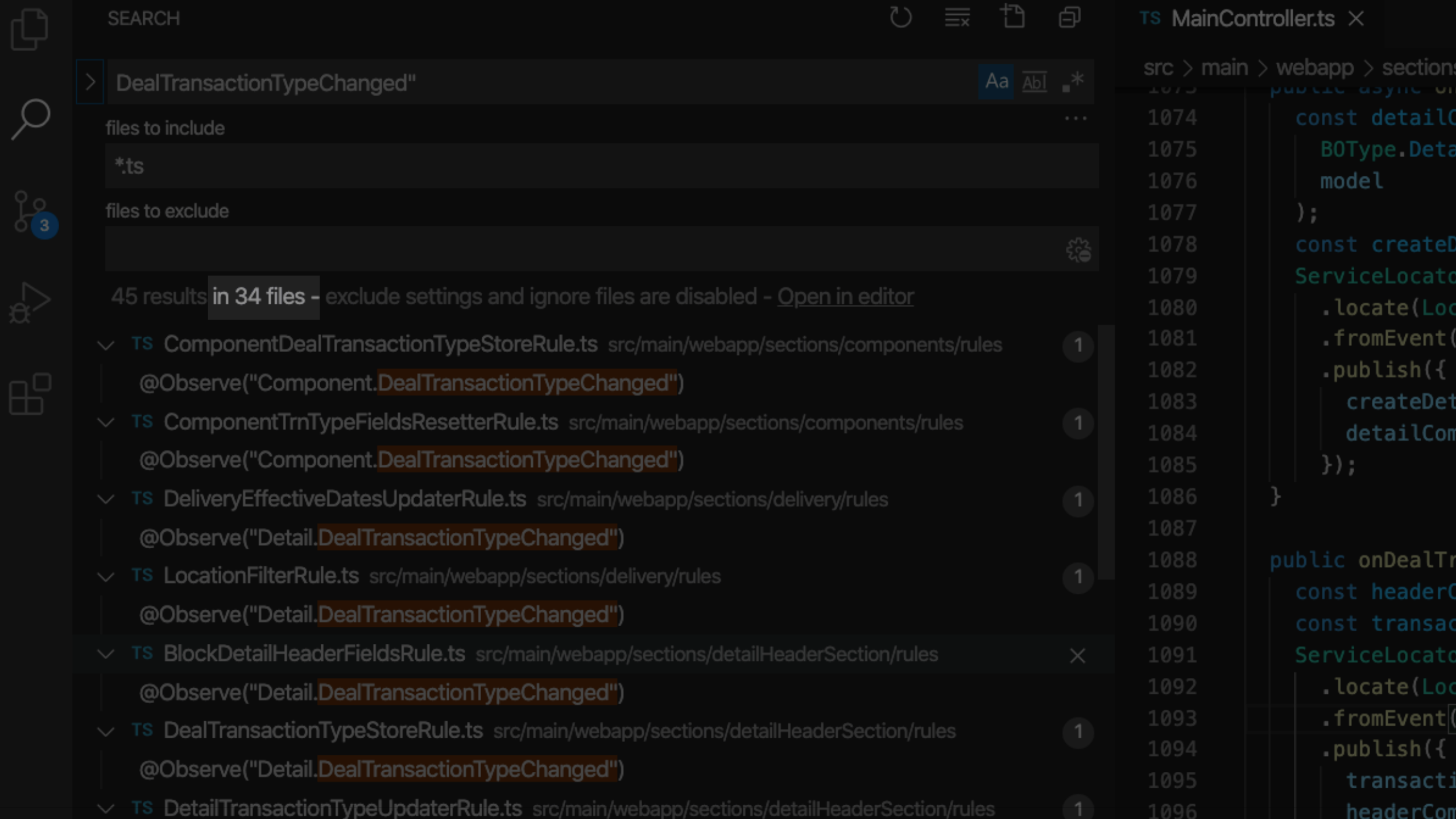
Peek Problem (⌥F8) No quick fixes available

```
}

public onDealTransactionTypeChanged(): void {
  const headerComposite = this.adapter.getHeaderComposite();
  const transactionType = this.adapter.getDealTransactionType();
  ServiceLocator.getInstance()
    .locate(LocatableServices.EventBus)
    .fromEvent("Header.DealTransactionTypeChanged")
    .publish({
      transactionType,
      headerComposite
    });
}

public onErrorMessagePopoverItemSelected(target: Target): void {
  ServiceLocator.getInstance()
    .locate(LocatableServices.EventBus)
    .fromEvent("Capture.ErrorMessageClicked")
```


SEARCH
DealTransactionTypeChanged"
Aa
files to include
*.ts
files to exclude
45 results in 34 files - exclude settings and ignore files are disabled - Open in editor
ComponentDealTransactionTypeStoreRule.ts src/main/webapp/sections/components/rules 1
@Observe("Component.DealTransactionTypeChanged")
ComponentTrnTypeFieldsResetterRule.ts src/main/webapp/sections/components/rules 1
@Observe("Component.DealTransactionTypeChanged")
DeliveryEffectiveDatesUpdaterRule.ts src/main/webapp/sections/delivery/rules 1
@Observe("Detail.DealTransactionTypeChanged")
LocationFilterRule.ts src/main/webapp/sections/delivery/rules 1
@Observe("Detail.DealTransactionTypeChanged")
BlockDetailHeaderFieldsRule.ts src/main/webapp/sections/detailHeaderSection/rules
@Observe("Detail.DealTransactionTypeChanged")
DealTransactionTypeStoreRule.ts src/main/webapp/sections/detailHeaderSection/rules 1
@Observe("Detail.DealTransactionTypeChanged")
DetailTransactionTypeUpdaterRule.ts src/main/webapp/sections/detailHeaderSection/rules 1
MainController.ts
src > main > webapp > sections
1074 const detailC
1075 BOType.Deta
1076 model
1077);
1078 const createD
1079 ServiceLocato
1080 .locate(Loc
1081 .fromEvent(
1082 .publish({
1083 createDet
1084 detailCom
1085 });
1086 }
1087
1088 public onDealTr
1089 const headerC
1090 const transac
1091 ServiceLocato
1092 .locate(Loc
1093 .fromEvent(
1094 .publish({
1095 transacti
1096 headerCom


SEARCH
DealTransactionTypeChanged"
files to include
*.ts
files to exclude
45 results in 34 files - exclude settings and ignore files are disabled - Open in editor
ComponentDealTransactionTypeStoreRule.ts src/main/webapp/sections/components/rules
@Observe("Component.DealTransactionTypeChanged")
ComponentTrnTypeFieldsResetterRule.ts src/main/webapp/sections/components/rules
@Observe("Component.DealTransactionTypeChanged")
DeliveryEffectiveDatesUpdaterRule.ts src/main/webapp/sections/delivery/rules
@Observe("Detail.DealTransactionTypeChanged")
LocationFilterRule.ts src/main/webapp/sections/delivery/rules
@Observe("Detail.DealTransactionTypeChanged")
BlockDetailHeaderFieldsRule.ts src/main/webapp/sections/detailHeaderSection/rules
@Observe("Detail.DealTransactionTypeChanged")
DealTransactionTypeStoreRule.ts src/main/webapp/sections/detailHeaderSection/rules
@Observe("Detail.DealTransactionTypeChanged")
DetailTransactionTypeUpdaterRule.ts src/main/webapp/sections/detailHeaderSection/rules
MainController.ts
src > main > webapp > sections
const detailC
BOType.Deta
model
);
const createD
ServiceLocato
.locate(Loc
.fromEvent(
.publish({
createDet
detailCom
});
}
public onDealTr
const headerC
const transac
ServiceLocato
.locate(Loc
.fromEvent(
.publish({
transacti

files - exclude settings and ignore
d - Open in editor

Detail.DealTransactionTypeChanged")
ler.DealTransactionTypeChanged"
PricingHeader.DealTransactionTypeChan...
DealTransactionTypeChanged"
VisibilityRule.ts src/main/webapp/sec... 1
Header.DealTransactionTypeChanged")
erEffectiveDatesReseter.ts src/main... 1
Header.DealTransactionTypeChan... ×
ypeFilterRule.ts src/main/webapp/se... 1
Header.DealTransactionTypeChanged")
PartyEditableRule.ts src/main/weba... 1
Header.DealTransactionTypeChanged")
PartyRule.ts src/main/webapp/sectio... 1
Header.DealTransactionTypeChanged")
nsactionTypeDescriptionRule.ts sr... 1
Header.DealTransactionTypeChanged")
TypeFieldsResetterRule.ts src/main... 1
Header.DealTransactionTypeChanged")
DefaultRule.ts src/main/webapp/sect... 1
Detail.DealTransactionTypeChanged")
nitialParametersRule.ts src/main/we... 1
Detail.DealTransactionTypeChanged")
SectionResetterRule.ts src/main/w... 1
Detail.DealTransactionTypeChanged")
atesDefaultRule.ts src/main/webapp... 1
PricingItem.DealTransactionTypeChange...
TypeFieldsResetterRule.ts src/main/... 1
PricingHeader.DealTransactionTypeChan...

```
@Listener<DealHeaderEffectiveDatesReseter>()
export class DealHeaderEffectiveDatesReseter
  implements DealTypeChangedObserver {
  private header: Context;

  constructor(private dealTypeLoader: DealTypeLoader) {}

  @Observe("Header.DealTypeChanged")
  public async onDealTypeChanged({
    headerComposite
  }: DealTypeArgument & ApibHeaderEventArgument): Promise<void> {
    this.header = headerComposite.getContext();
    await this.resetDealHeaderEffectiveDates();
  }

  @Observe("Header.DealTransactionTypeChanged")
  public onDealTransactionTypeChanged({
    headerComposite
  }: DealTransactionTypeArgument & ApibHeaderEventArgument): Promise<void> {
    this.header = headerComposite.getContext();
    return this.resetDealHeaderEffectiveDates();
  }

  private async resetDealHeaderEffectiveDates(): Promise<void> {
    const isEvergreen = await this.isEvergreen();
    if (this.header && !isEvergreen && !this.isIntraStrategy()) {
      this.header.setProperty("effectiveTo", null);
      this.header.setProperty("effectiveFrom", null);
    }
  }

  private async isEvergreen(): Promise<boolean> {
    const dealTypeSelected = this.header.getProperty("dealType");
    const dealType = await this.dealTypeLoader.getDealType(dealTypeSelected);
    return Boolean(dealType && dealType.isEvergreen);
  }

  private isIntraStrategy(): boolean {
    const transactionType = this.header.getProperty("transactionType");
    return transactionType === TransactionType.INTRASTRATEGY;
  }
}
```

```
@Listener<DealHeaderEffectiveDatesReseter>()
export class DealHeaderEffectiveDatesReseter
  implements DealTypeChangedObserver {
  private header: Context;

  constructor(private dealTypeLoader: DealTypeLoader) {}

  @Observe("Header.DealTransactionTypeChanged")
  public onDealTransactionTypeChanged({
    headerComposite
  }: DealTransactionTypeArgument & ApibHeaderEventArgument): Promise<void> {
    this.header = headerComposite.getContext();
    return this.resetDealHeaderEffectiveDates();
  }

  @Observe("Header.DealTypeChanged")
  public async onDealTypeChanged({
    headerComposite
```

```
@Listener<DealHeaderEffectiveDatesReseter>()
export class DealHeaderEffectiveDatesReseter
  implements DealTypeChangedObserver {
  private header: Context;

  constructor(private dealTypeLoader: DealTypeLoader) {}

  @Observe("Header.DealTransactionTypeChanged")
  public onDealTransactionTypeChanged({
    headerComposite
  }: DealTransactionTypeArgument & ApibHeaderEventArgument): Promise<void> {
    this.header = headerComposite.getContext();
    return this.resetDealHeaderEffectiveDates();
  }

  @Observe("Header.DealTypeChanged")
  public async onDealTypeChanged({
    headerComposite
```

```
private async resetDealHeaderEffectiveDates(): Promise<void> {
  const isEvergreen = await this.isEvergreen();
  if (this.header && !isEvergreen && !this.isIntraStrategy()) {
    this.header.setProperty("effectiveTo", null);
    this.header.setProperty("effectiveFrom", null);
  }
}

private async isEvergreen(): Promise<boolean> {
  const dealTypeSelected = this.header.getProperty("dealType");
  const dealType = await this.dealTypeLoader.getDealType(dealTypeSelected);
  return Boolean(dealType && dealType.isEvergreen);
}

private isIntraStrategy(): boolean {
  const transactionType = this.header.getProperty("transactionType");
  return transactionType === TransactionType.INTRASTRATEGY;
}
```

Sadly, it’s time to go.
More information:

Boilerplate to start a new project following this style:

https://github.wdf.sap.corp/ASE-IBSO-Brazil/UI5-ts-pubsub-boilerplate

Soon to be open sourced!

Main takeaways

- Limit the MVC flow inside your application to have greater control;
- Prefer One-Way binding;
- Avoid calling methods in the View/Controls (byId/setVisible/setBusy);
- All View properties must binded to a Model;
- Use TypeScript if you can;
- Split your Controller into several classes;

Thank you (for real this time)

Gabriel Borges
Fiori Architect, SAP Innovative Business Solutions
SAP Labs Latin America – São Leopoldo, RS, Brazil

gabriel.borges@sap.com

Twitter: @psidium_
Github: Psidium

UI5con ON AIR