

Sitecore MVC Advanced

Kevin Brechbühl



Unic AG



@aquasonic


SUGCON EUROPE 2015

JUNE 12TH THE EXPERIENCE STARTS HERE



sitecore | user group nederland

About

- Senior Sitecore developer & architect  @unic
- Sitecore Technology MVP 2015
- Co-founder of  @sugch



Agenda

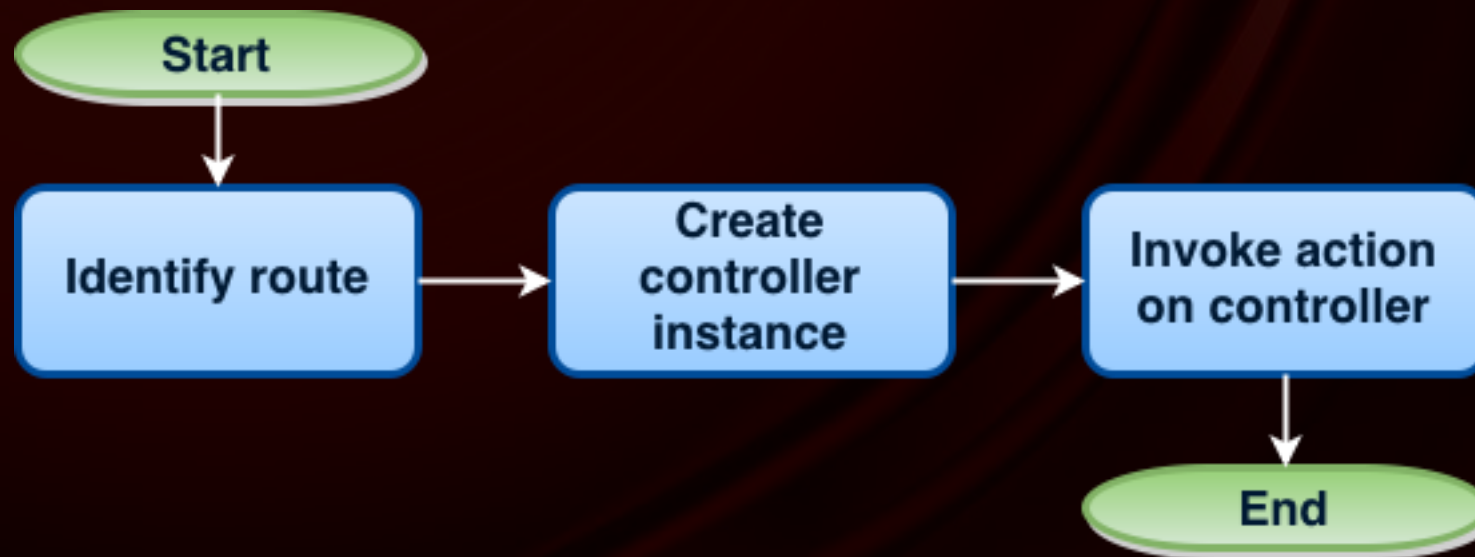
- ASP.net MVC vs. Sitecore MVC
- Multiple Forms
- MVC Forms Localization
- MVC Areas
- There is always more...

ASP.net MVC vs. Sitecore MVC

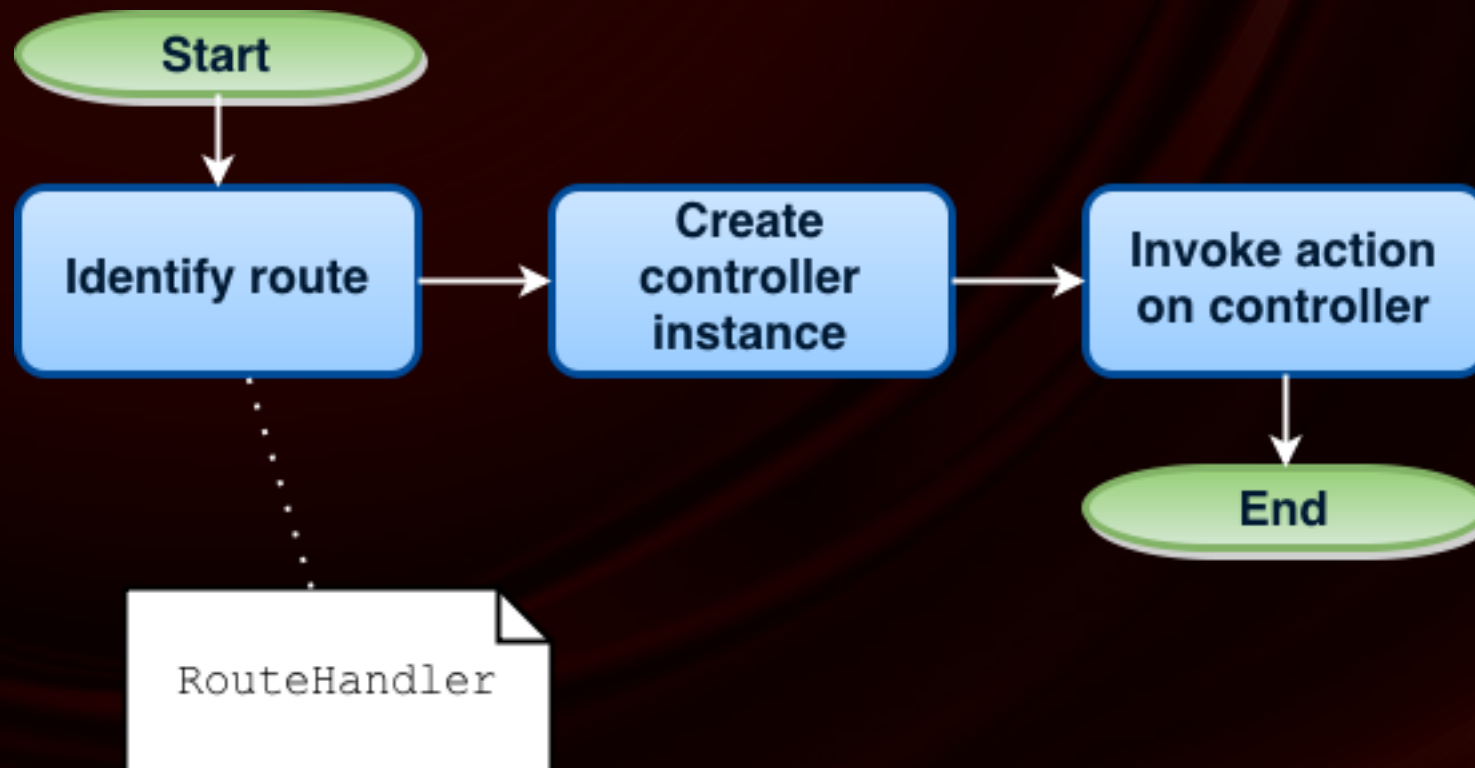


STUDERT

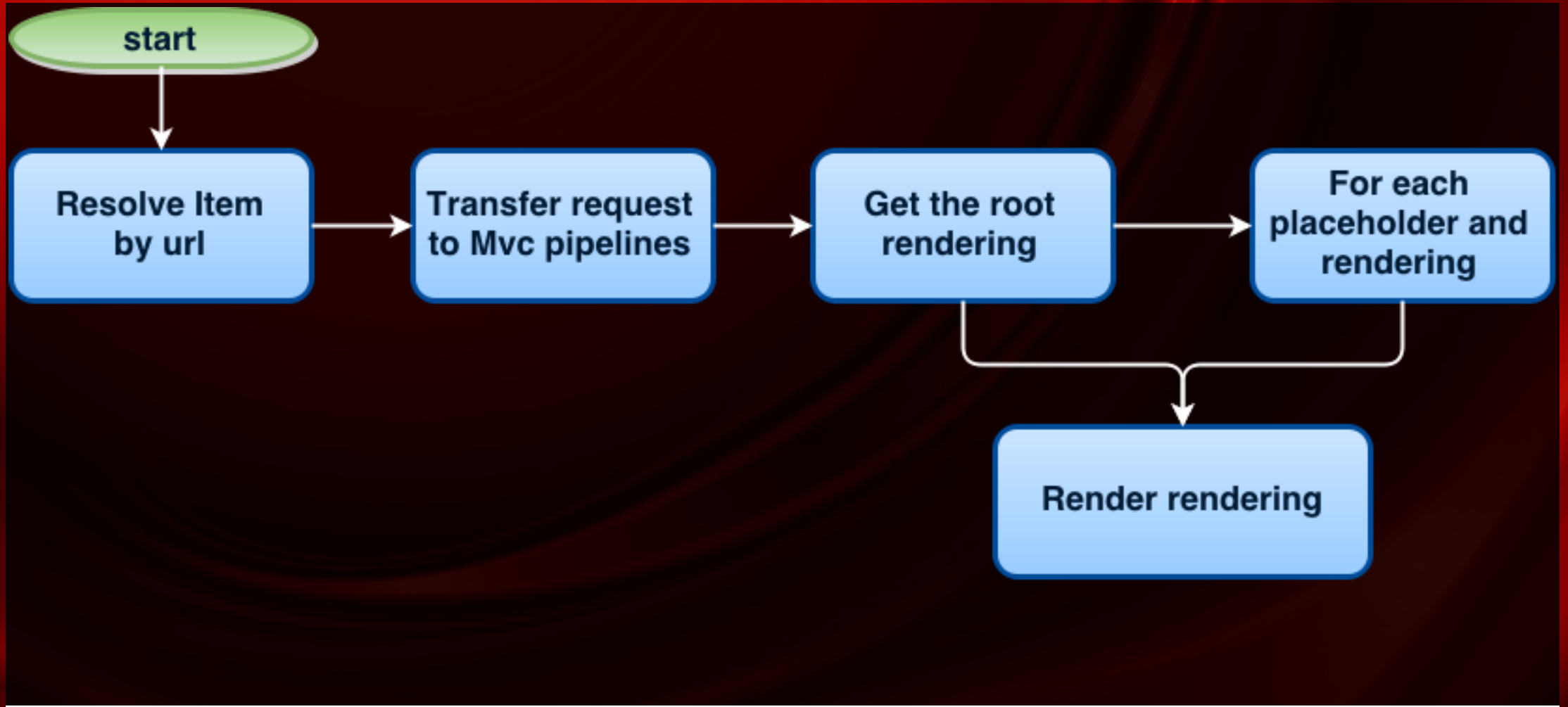
Traditional ASP.net MVC



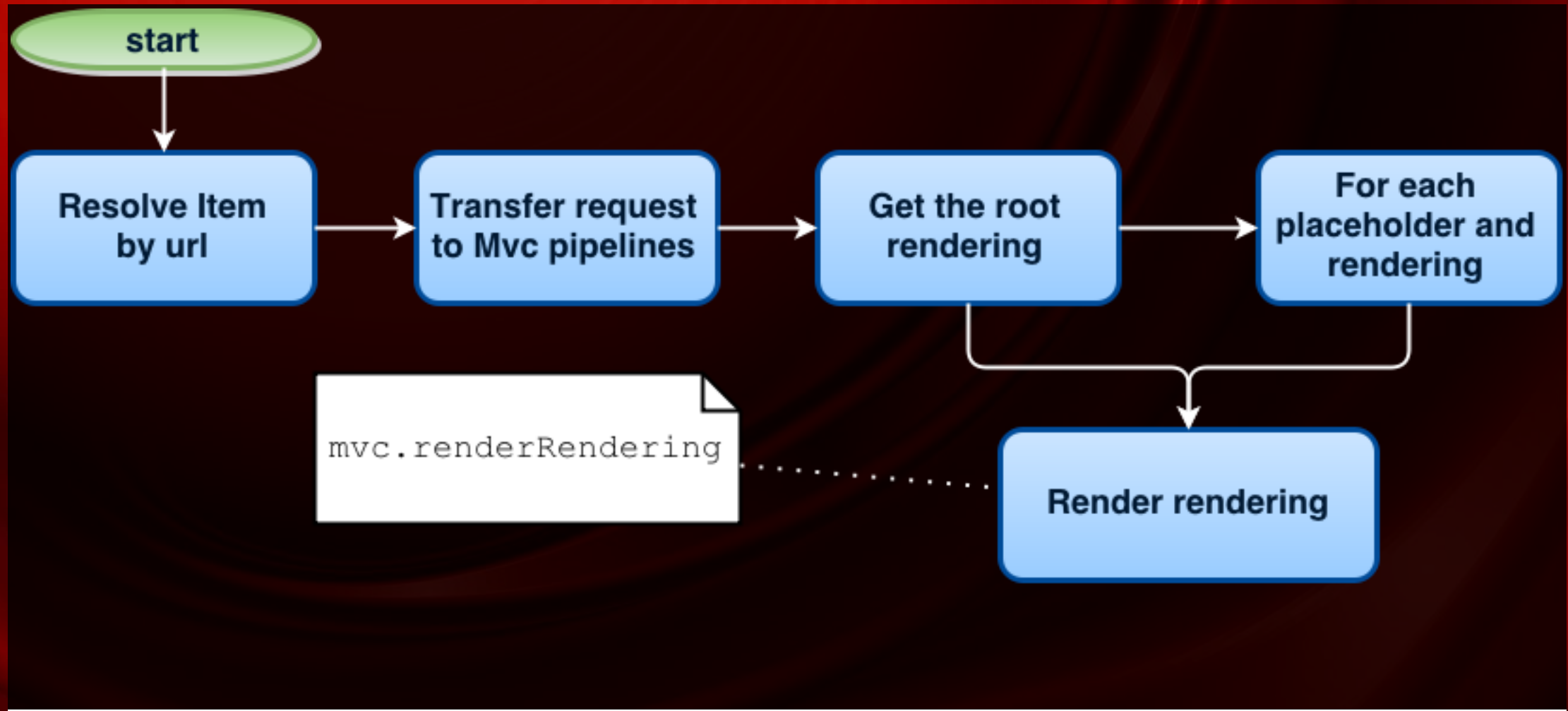
Traditional ASP.net MVC



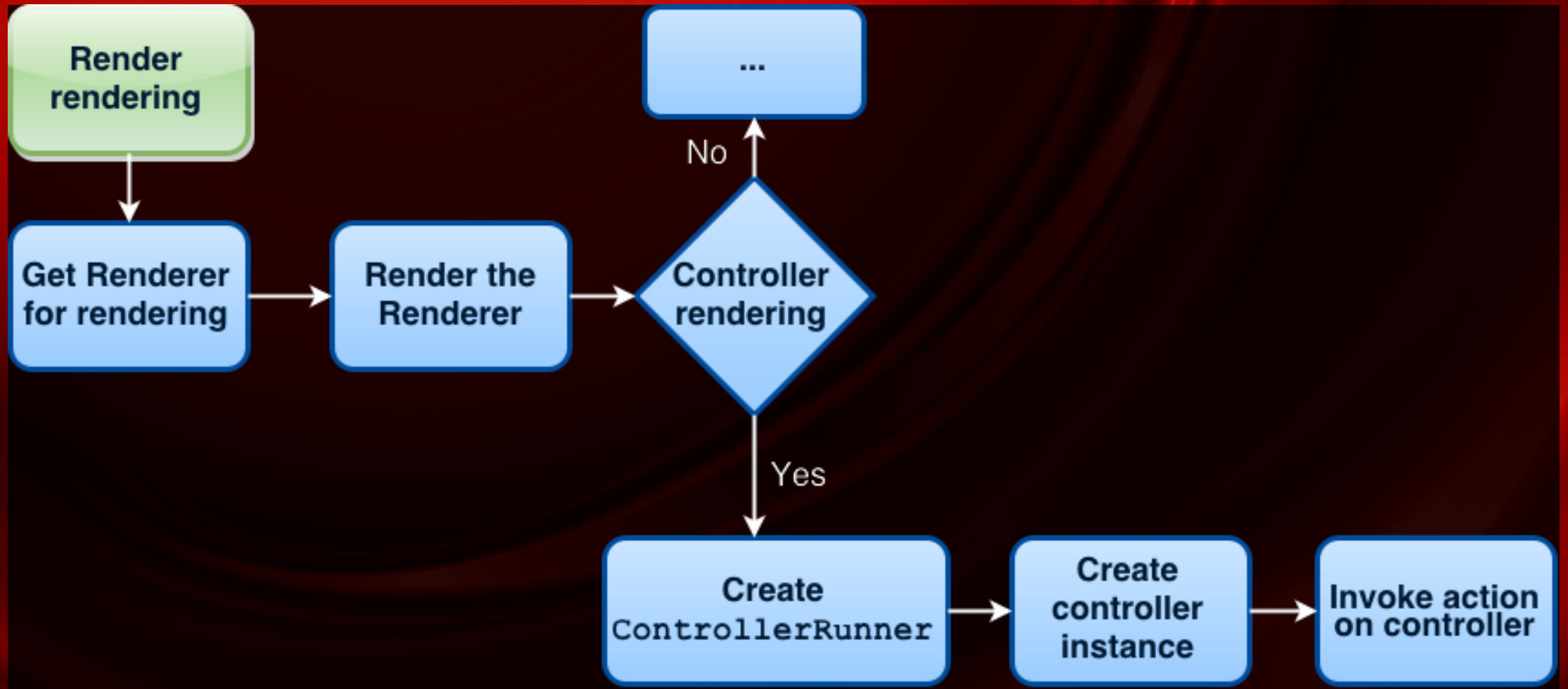
Sitecore MVC



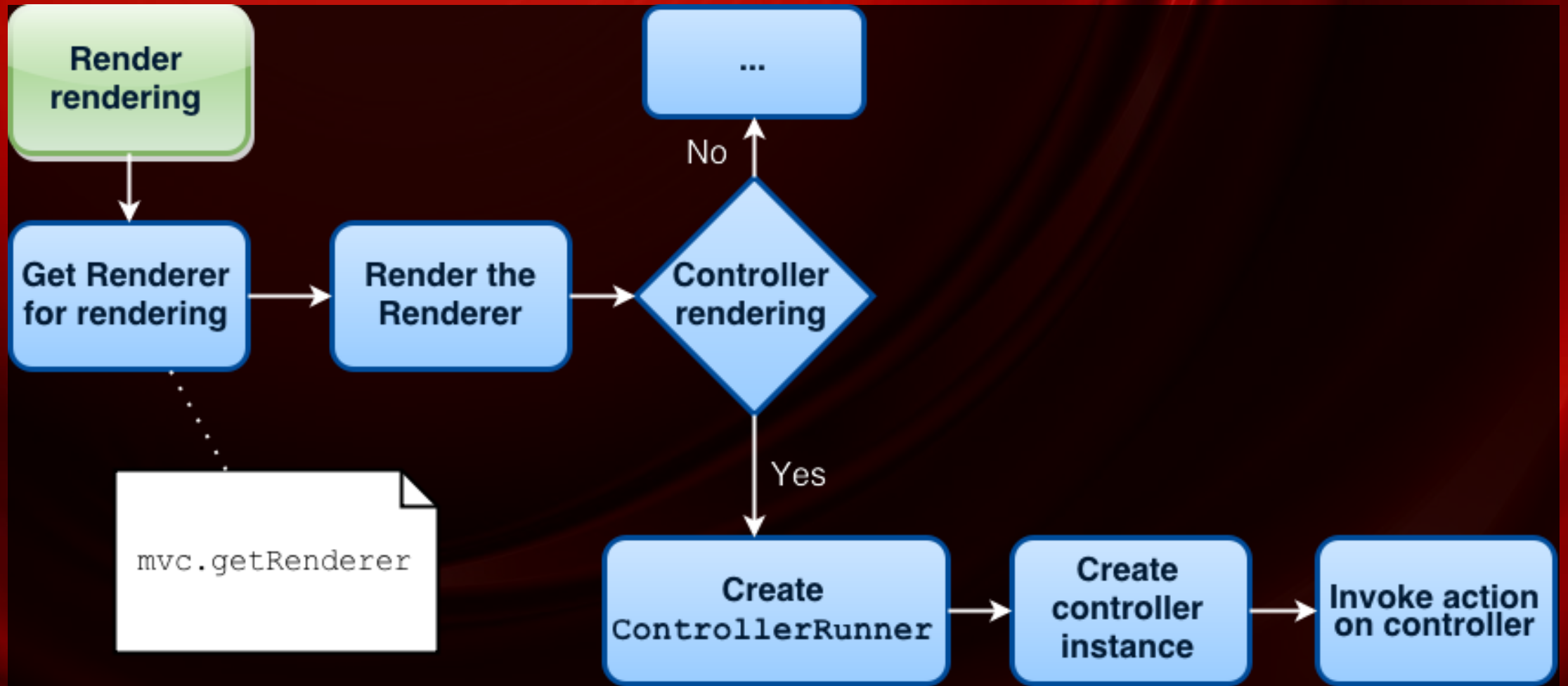
Sitecore MVC



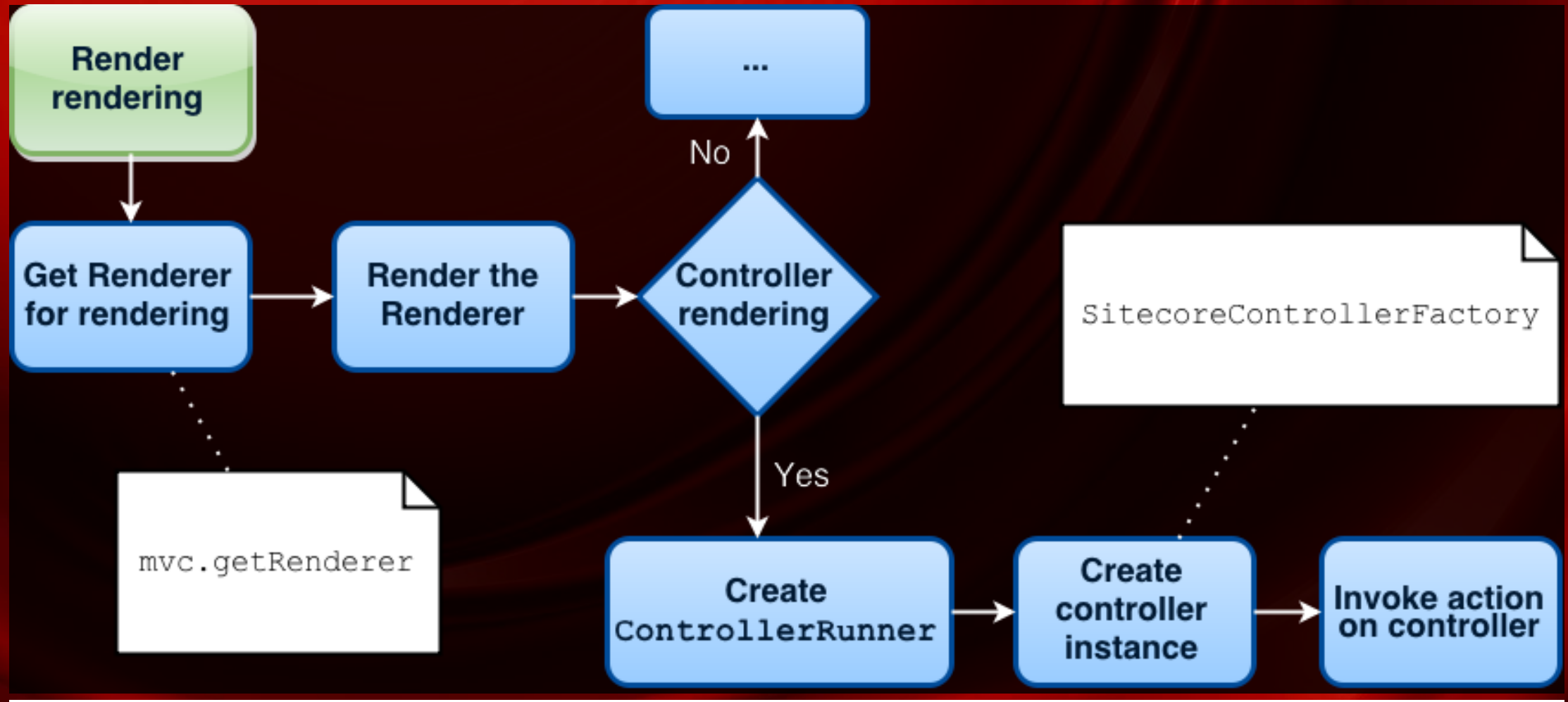
Sitecore MVC



Sitecore MVC



Sitecore MVC



ASP.net MVC vs. Sitecore MVC

ASP.net MVC

- Request correspond to controller action
- One view with several partial views
- **1 controller & 1 action**

Sitecore MVC

- Request correspond to Sitecore item
- Views defined in presentation details
- **n controllers & n actions**

Resources

- Comparison from John West

<http://bit.ly/SitecoreMvcComparison>

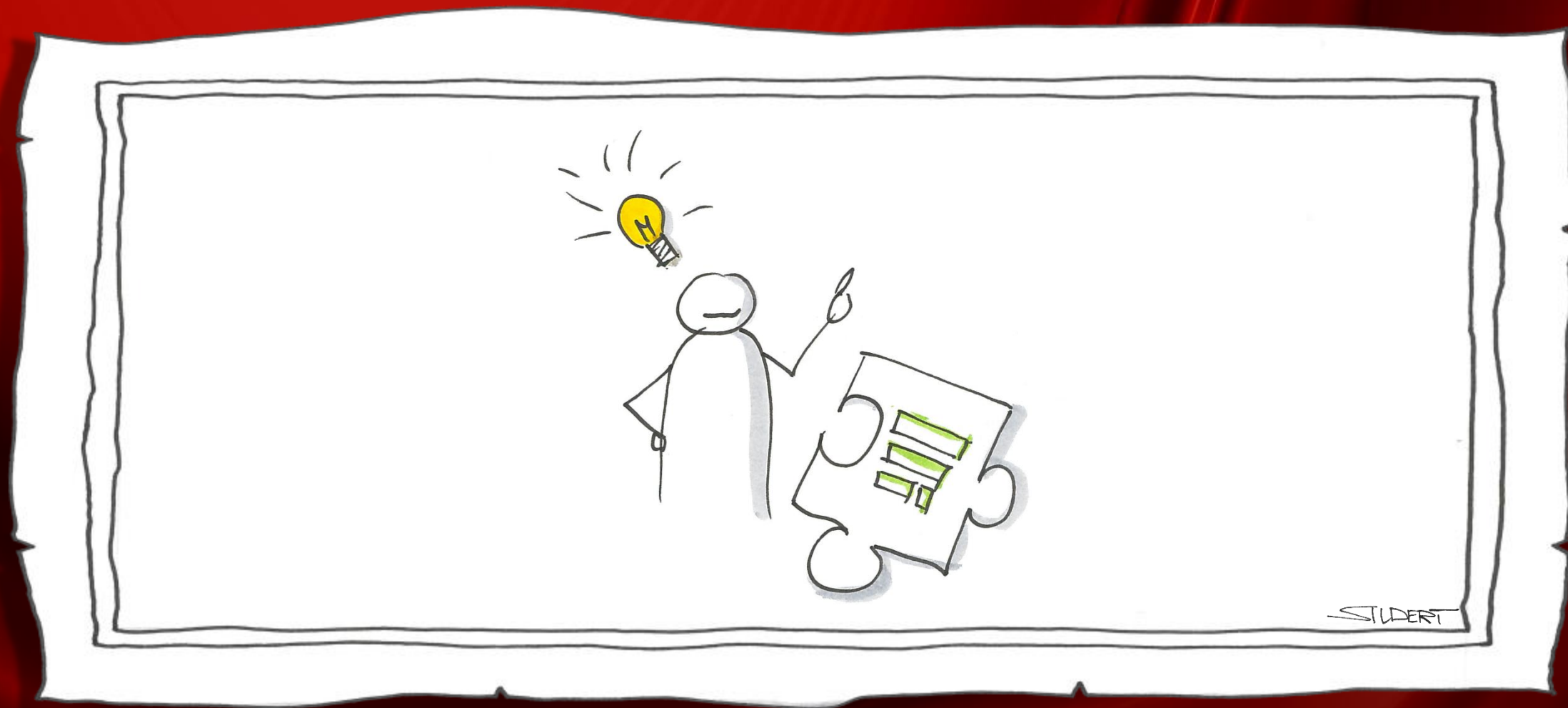
- ASP.net MVC Lifecycle

<http://bit.ly/MvcLifecycle>

- Sitecore MVC Pipelines

<http://bit.ly/SitecoreMvcPipelines>

Multiple Forms



ASP.net MVC form posts

- Each form has 2 actions
 - Handling GET requests
 - Handling POST requests
- **Action** has `ActionMethodSelector`
 - `[HttpGet]` or empty
 - `[HttpPost]`

ASP.net MVC form posts

{ CODE }

Issues with Sitecore?

- Multiple controllers & multiple actions
- Post to same page / url
- Each [HttpPost] action is called
- Each form is validated
- AntiForgeryToken does not work properly

Solutions

- Post via Ajax
- Sitecore MVC Form Handler
- Custom `ActionMethodSelector`
 - As close to ASP.net MVC as possible
 - None of the described issues

Custom ActionMethodSelector

- Adding a hidden field to the form
 - Unique Id of the rendering
- Post to same page / url
- Adding the `ActionMethodSelector` to the action
 - Verify the posted rendering id

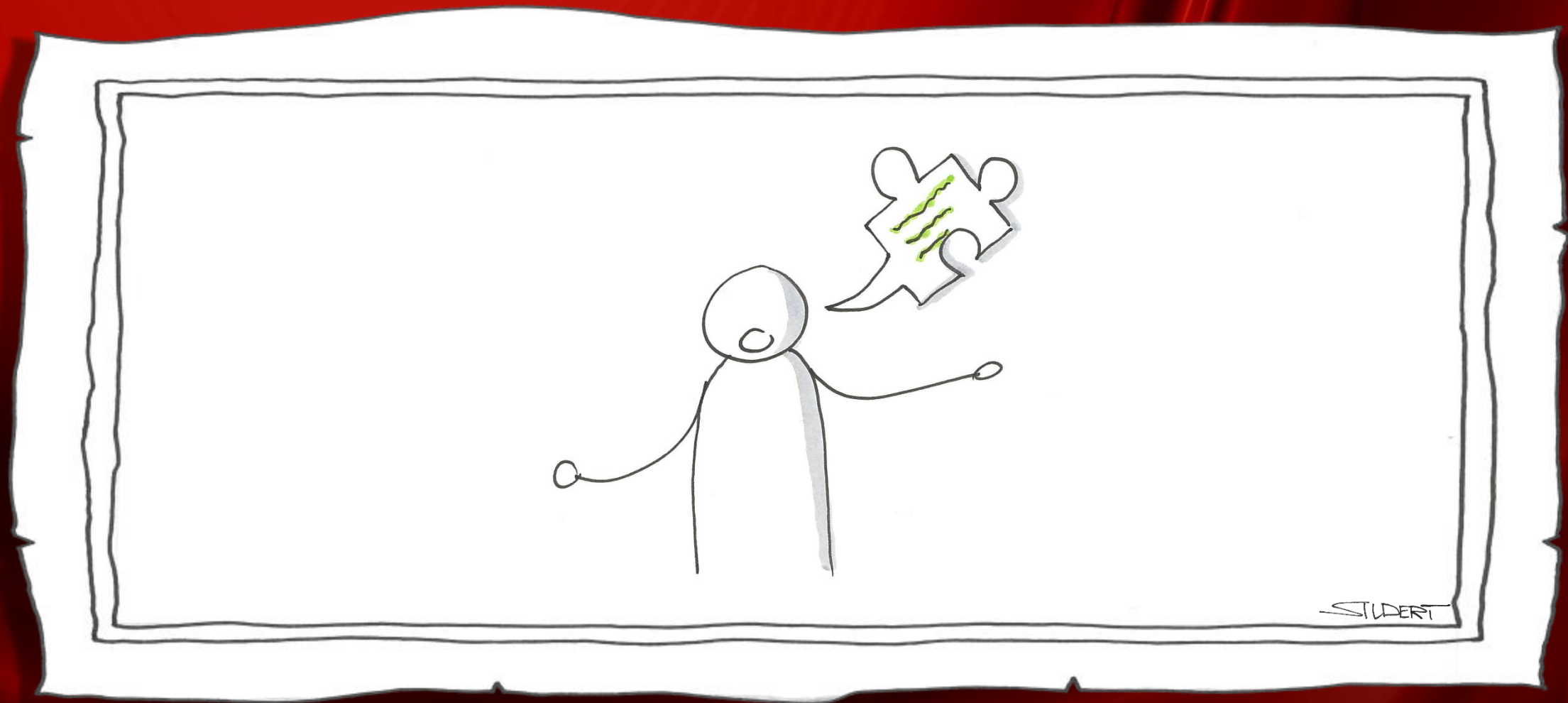
Custom ActionMethodSelector

```
{ CODE }
```


Resources

- Mine: <http://bit.ly/MultipleForms1>
- Renoud van Dalen: <http://bit.ly/MultipleForms2>

MVC Forms localization



MVC Forms localization

- Ways to translate an ASP.net MVC form
 - Use .NET language pack
 - ~~Add static textes~~
 - Use .NET resource files
- Localization within Sitecore
 - Sitecore dictionaries

MVC Forms localization

{ CODE }

Translate labels

- Use custom `MetaDataProvider`
- First create the default metadata for a field
 - Type, Format etc.
- Override the `DisplayName`

Translate labels

{ CODE }

Translate labels

- Conflicts with WFFM MVC
 - Override the WFFM provider
 - Register custom provider after WFFM provider

Translate validation messages

- Custom `DataAnnotationsModelValidator`
- Works as an adapter
- Each validator type needs a custom adapter

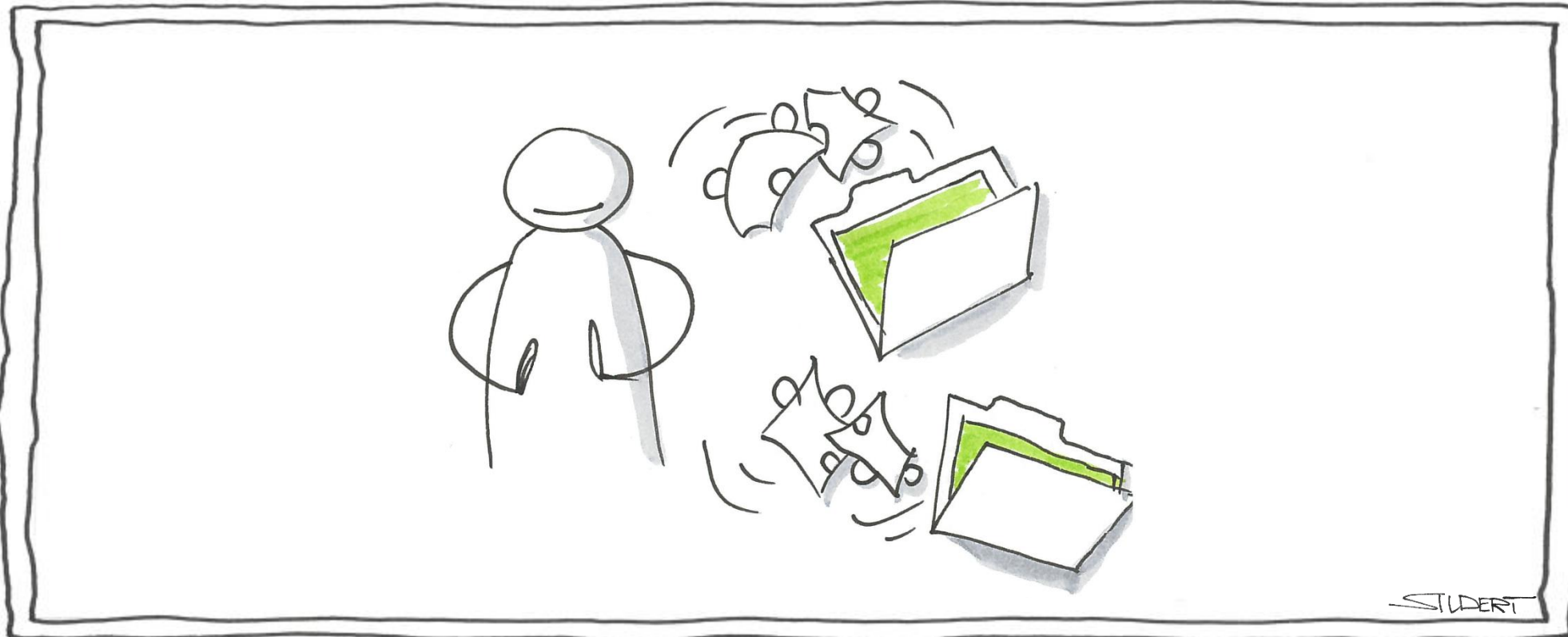
Translate validation messages

```
{ CODE }
```

Resources

- Sitecore Usergroup Switzerland
- From  @studert and me
- <http://bit.ly/FormsLocalization>

MVC Areas



Structure of Areas

Default

```

└─ AreasStructure
   ├─ Properties
   ├─ References
   └─ Areas
      ├─ Area1
      │  ├─ Controllers
      │  ├─ Models
      │  └─ Views
      │     └─ Area1AreaRegistration.cs
      └─ Area2
         ├─ Controllers
         ├─ Models
         └─ Views
            └─ Area2AreaRegistration.cs
└─ Controllers
   └─ Models
      └─ Views
         └─ packages.config
         └─ Web.config

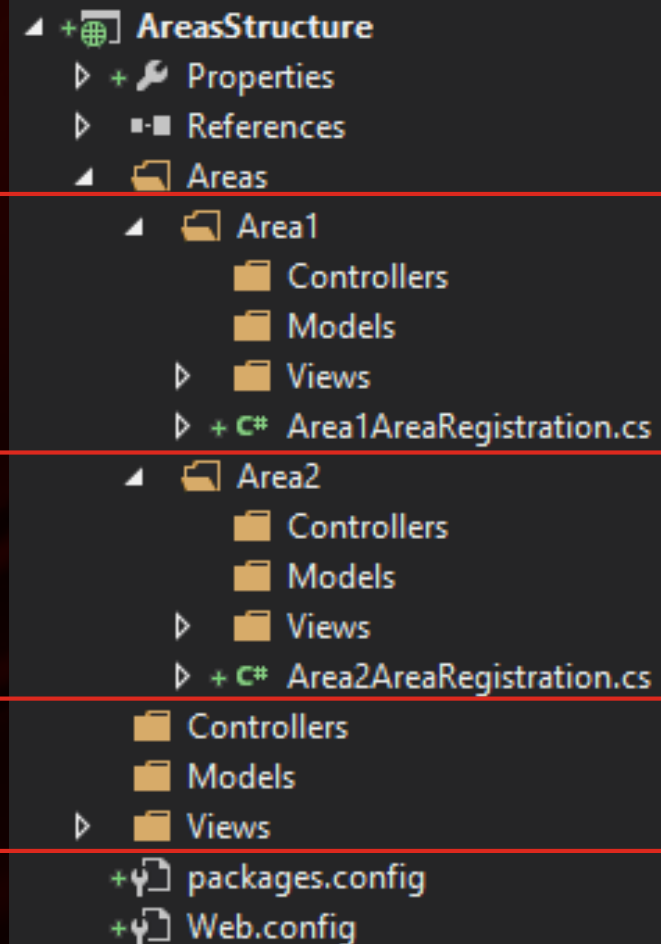
```

The screenshot displays a Visual Studio solution explorer for a project named 'AreasStructure'. The project is organized into a hierarchical structure. At the top level, there are 'Properties', 'References', and 'Areas'. Under 'Areas', there are two sub-areas: 'Area1' and 'Area2'. Each area contains its own 'Controllers', 'Models', and 'Views' folders. Additionally, each area has a corresponding registration file: 'Area1AreaRegistration.cs' for Area1 and 'Area2AreaRegistration.cs' for Area2. Below the 'Areas' folder, there are global project folders for 'Controllers', 'Models', and 'Views', along with configuration files 'packages.config' and 'Web.config'.

Structure of Areas

Area 1

Default

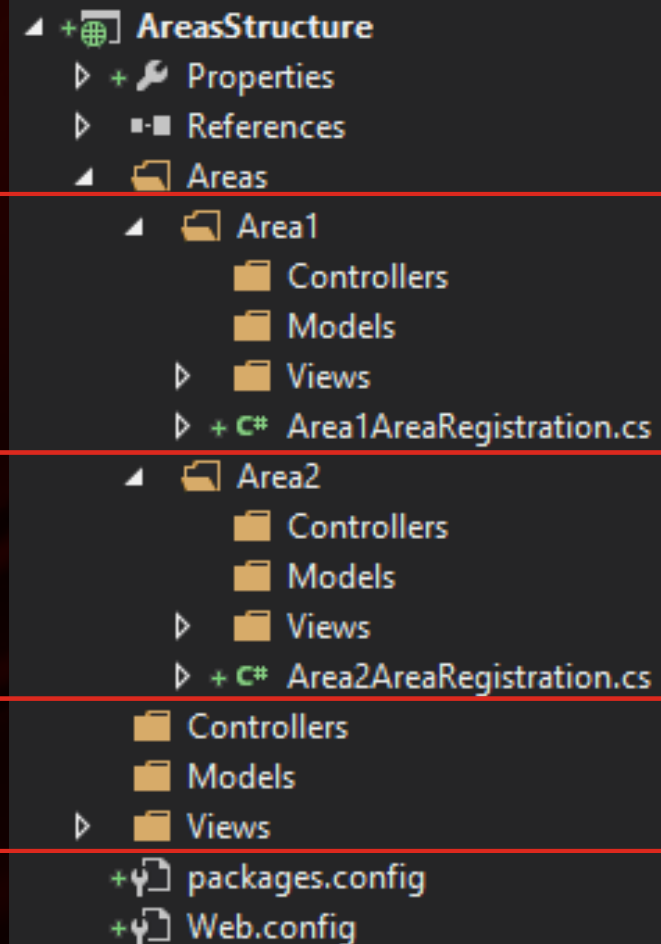


Structure of Areas

Area 1

Area 2

Default



Areas what?

- Split models/views/controllers in different areas
- Used with multiple sites/layouts in one Sitecore instance
- No support by Sitecore, because
 - No area token in Sitecore default route
 - Layout termination over pipelines and presentation details

Solutions

- Resolve in `mvc.renderRendering` pipeline
- Use custom `ControllerRunner` and `Renderer`
- Resolve by `<site>`

Resolve in mvc.renderRendering pipeline

- Set `RouteData.DataTokens["area"]`
- Controller is then resolved in area
- Resolve the area in a custom pipeline
 - New fields on folders / renderings
- Resource
 - BrainJocks SCORE: <http://bit.ly/MvcAreas1>

Resolve in mvc.renderRendering

```
{ CODE }
```


Use custom ControllerRunner

- Custom ControllerRunner in `mvc.getRenderer` pipeline
- Also set `RouteData-token`
- Resolve area by new «Area Controller Rendering»
 - With new `AreaControllerRenderer`
- Resource
 - Christian van de Luitgaarden: <http://bit.ly/MvcAreas2>

Use custom ControllerRunner

```
{ CODE }
```

Resolve by <site>

- Adding new route per <site>
 - Configure with new parameter
- Before the Sitecore default route
- Add RouteData-token and constraint to each route
- Resource
 - Chris van de Steeg: <http://bit.ly/MvcAreas3>

Resolve by <site>

{ CODE }

But, wait...

46

votes

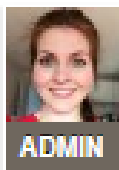
Vote

Area support

Sitecore MVC should support MVC areas



[Kern Herskind Nightingale](#) shared this idea · March 25, 2015



STARTED

· [Martina Welander](#) (Technical Consulting Engineer, SC Community Docs) responded · April 24, 2015

Great news – we have started work on Area support!

Regards,
Kern

Sitecore Area support

- Pattern from BrainJocks
- Multiple strategies to resolve the area
- Extensible by developers

There is always more...



Other resources

- Sample Items in MVC

<http://bit.ly/Sample7Mvc> & <http://bit.ly/Sample8Mvc>

- Async Sitecore controllers

<http://bit.ly/AsyncControllers>

- Using Dependency Injection to determine MVC Models

<http://bit.ly/ModelsDI>


- BrainJocks SCORE

<http://bit.ly/BJScore>

UserVoice

- Sitecore needs your feedback
- Share and vote your ideas
- Not only for MVC ;-)
- <http://bit.ly/MvcUserVoice>

/me

- Twitter:  @aquasonic
- Private Blog: <http://ctor.io>
- Sitecore Vibes: <http://sitecore.unic.com>
- Sitecore Usergroup Switzerland: <http://sugch.github.io>

Thank you



SILBERT

Thank you



Colours

CONCLUSION

FORWARD
search



iquality

partTECH
WE MASTER SITECORE



Redhotminute | digital agency



Tahzoo

unic