CQRS

moje własne podejście

Jakub Gutkowski
@gutek

Jakub Gutkowski


kuba@gutek.pl


blog.gutek.pl


@gutek


gutek

A dzisiaj w menu

- Czym jest CQRS
- Czym nie jest CQRS
- Gdzie można CQRS zastosować
- Jak ja do CQRS podchodzę

CQS

Command and Query Seperation

Bertrand Meyer, 1988

```
public interface IRepository<T>
{
    // queries
    T GetById(Guid id);
    IEnumerable<T> GetAll();

    // commands
    void Create(T item);
    void Update(T item);
}
```

Czym jest CQRS

Czym NIE jest CQRS

Co to jest CQRS?

CQRS !<=>? CQS

```
public interface IRepository<T>
{
    // queries
    T GetById(Guid id);
    IEnumerable<T> GetAll();

    // commands
    void Create(T item);
    void Update(T item);
}
```

Segregation

```
class WriteSrvc<T>
{
    // commands
    void Create(T item);
    void Update(T item);
}
```

```
class ReadSrvc<T>
{
    // queries
    T GetById(Guid id);
    IEnumerable<T> GetAll();
}
```

blame

him

him

&

flame war!

Po lewej, *Udi Dahan*, po prawej *Greg Young*

https://unsplash.com/toddquackenbush

side note

with astonishing view

https://unsplash.com/dusty_blanco

side note

https://unsplash.com/pauleharrer

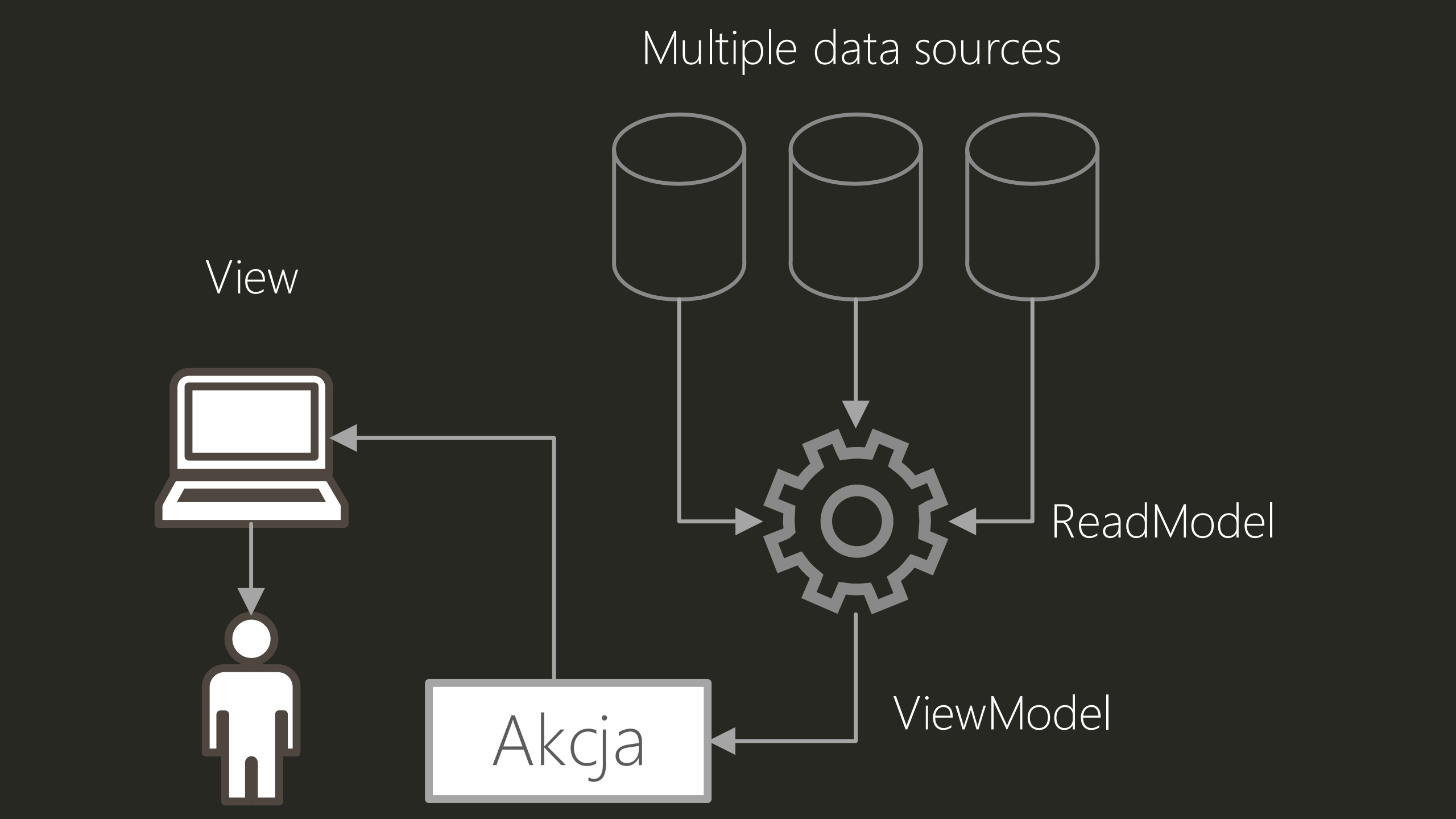
with astonishing view

side note

https://unsplash.com/wolfgangmoritzer

with astonishing view

Query


Multiple data sources
View
ReadModel
ViewModel
Akcja

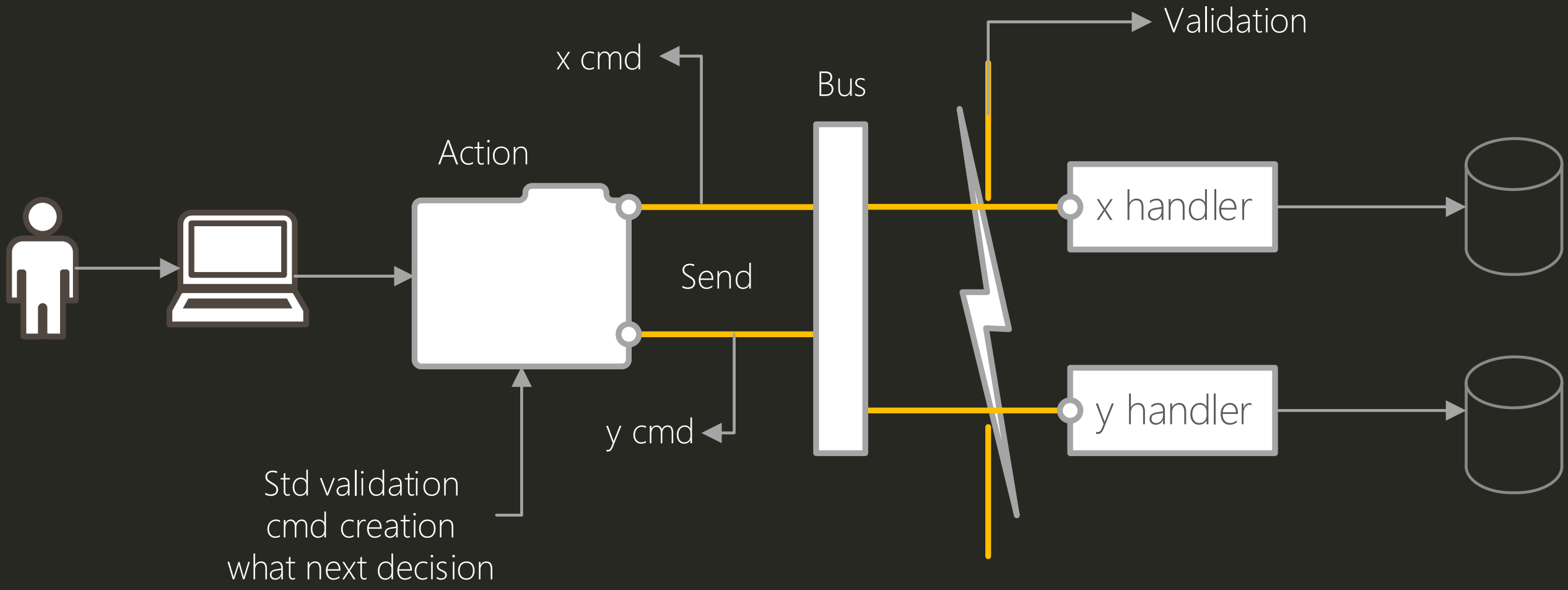
```
// marker interface
public interface IReadModel {}

public interface ISepticTankRegistrationReadModel
    : IReadModel
{
    NotActiveSepticTanksVm GetNotActive();
}

public class SepticTankRegistrationReadModel
    : ISepticTankRegistrationReadModel
{
    public SepticTankRegistrationReadModel(IXrmService crm
        , IDbConext ctx)
    { /* ... */ }

    public NotActiveSepticTanksVm GetNotActive()
    { /* ... */ }
}
```

```
[ClaimsAuthz("SepticTank", "ViewNotActive")]
public ActionResult NotActive()
{
    var vm = _readModel.GetNotActive();
    return View("not-active", vm);
}
```

Query

- Zwraca „ViewModel”
- Wiele źródeł danych
- Jest odpowiedzialne tylko za JEDNĄ rzecz
- NIE jest re-używalne
- Inne miejsca NIE korzystają z *Query*

Command


Validation
x cmd
Bus
Action
x handler
Send
y cmd
y handler
Std validation
cmd creation
what next decision

```
// marker interface
public interface ICommand {}

public ActivateSepticTankCommand : ICommand
{
    /* ... */
    public ActivateSepticTankCommand(/* params */)
    {/* ... */}
}

public class ActivateSepticTankCommandHandler
    : ICommandHandler<ActivateSepticTankCommand>
{
    public ActivateSepticTankCommandHandler(IDbConext ctx)
    { /* ... */ }

    public [void|object] Handle(ActivateSepticTankCommand cmd)
    { /* ... */ }
}
```

```
[HttpPost]
[ValidateAntiForgeryToken]
[ClaimsAuthz("SepticTank", "Activate")]
public ActionResult Activate(NotActiveSepticTanksInput model)
{
    if(!ModelState.IsValid)
    {
        return View("activate", model);
    }

    var activateCmd = new ActivateSepticTankCommand(/* ... */);
    _commandBus.Send(activateCmd);

    var assignCmd = new AssignPropertyToSepticTankCommand(/* ... */);
    _commandBus.Send(assignCmd);

    return RedirectToAction("NotActive", "SepticTankActivation");
}
```

https://unsplash.com/dietmarbecker

Command

- INTENCJA użytkownika
- Command Pattern i Command Bus ułatwiają życie
- Komenda MUSI mieć JEDNEGO odbiorcę
- Może być walidowana
- Może być odrzucona przez odbiorcę
- Może działać na danych z bazy, NIE na *Query*

Podsumowanie

Pytania?

Dziękuję


VISA
VISA

Jakub Gutkowski


kuba@gutek.pl


blog.gutek.pl


@gutek


gutek

https://twitter.com/mihcall/status/388265569104842752

```
public ModuleAdminListViewModel GetModuleAdminList(Guid userId, Guid organisationId)
{
    var result = new ModuleAdminListViewModel();
    result.EditModuleAllowed = false;

    var modules = _edenContext.ModuleAuthorisations.Where(m => m.Active
        && m.UserId == userId
        && m.OrganisationId==organisationId
        && m.Roles.Select(r=>r.StaticName).Contains(Constants.AdminRole)
        && m.Module.ModuleAdminApprovalAllowed);

    var admins = modules.ToList().Select(m => new ModuleAdminListViewModel.ModuleAdmin()
    {
        Module = new ModuleViewModel()
        {
            ModuleId = m.ModuleId,
            Name = m.Module.Name,
            StaticName = m.Module.StaticName,
            ContactPersionEmail = m.Module.Application.ResponsibleContact.Email,
            ContactPersionName = string.Format("{0} {1}",m.Module.Application.ResponsibleContact.FirstName, m.Module.Application.ResponsibleContact.LastName),
            Url = m.Module.URL,
            ApplicationLogo = m.Module.Application.Logo,
            ApplicationLogoMimeType = m.Module.Application.LogoName,
            Description = m.Module.Description
        },
        NumberOfPendingRequests = m.Module.ModuleAuthorisationRequests.Count(r => r.OrganisationId == organisationId
            && r.UserProfile.OrganisationMemberships.Any(o => o.Active && o.OrganisationId == organisationId))
    }).ToList();

    var user = _edenContext.UserProfiles.Find(userId);
    var globalAdmins = user.GlobalModuleAuthorisations.Where(m => m.Active && m.Module.GlobalAdminApprovalAllowed
        && m.Module.ModuleOrganisationTypes.Select(t=>t.OrganisationTypeId).Contains(user.Organisation.OrganisationTypeId)).Select(m => new ModuleAdminListViewModel.ModuleAdmin()
    {
        Module = new ModuleViewModel()
        {
            ModuleId = m.ModuleId,
            Name = m.Module.Name,
            StaticName = m.Module.StaticName,
            ContactPersionEmail = m.Module.Application.ResponsibleContact.Email,
            ContactPersionName = string.Format("{0} {1}", m.Module.Application.ResponsibleContact.FirstName, m.Module.Application.ResponsibleContact.LastName),
            Url = m.Module.URL,
            Description = m.Module.Description
        },
        NumberOfPendingRequests = m.Module.ModuleAuthorisationRequests.Count(r => r.OrganisationId == organisationId
            && r.UserProfile.OrganisationMemberships.Any(o => o.Active && o.OrganisationId == organisationId))
    }).ToList();

    result.Modules = admins.Concat(globalAdmins).DistinctBy(m => m.Module.ModuleId).ToList();

    return result;
```

```
24  ····public·class·OrganisationMembershipCreateCommand·:·ICommand
25  ····{
26  ········/*·...·*/
27  ········public·OrganisationMembershipCreateCommand(Guid·orgMembershipRequestId
28  ············,·bool·administrator
29  ············,·Guid·requesterUserId)...
35  ····}
36
37  ····public·class·OrganisationMembershipCreateCommandHandler·
38  ········:·ICommandHandler<OrganisationMembershipCreateCommand>
39  ····{
40  ········private·static·readonly·Logger·_log·=·LogManager.GetCurrentClassLogger();
41
42  ········public·OrganisationMembershipCreateCommandHandler(IXrmZeroService·xrmZero
43  ············,·IEdenContext·edenContext
44  ············,·IEventBus·eventBus
45  ············,·IXrmOneService·xrmOne)
46  ········{·}
47
48  ········public·object·Handle(OrganisationMembershipCreateCommand·command)...
96
97  ········private·Guid?·GrantUserAccessToPortalModule(OrganisationMembershipCreateCommand·command
98  ············,·Guid·userId
99  ············,·Guid·organisationId)...
225 ········
226 ········private·CommandResponse·CreateNew(OrganisationMembershipCreateCommand·command
227 ············,·OrganisationMembershipRequest·request)...
304
305 ········private·CommandResponse·ReActivate(Data.Models.OrganisationMembership·existingMembership
306 ············,·bool·administrator
307 ············,·Guid·administratorId
308 ············,·UserProfile·userProfile
309 ············,·Guid·organisationId)...
390 ····}
```