

# Recapitulare – autentificare si autorizare

Bogdan Macovei @Essensys

# Despre ce discutăm acum...

- Ne reamintim cum ne autentificăm
- Ne reamintim cum are loc autorizarea
- Înțelegem cum se populează userul curent

# Despre ce discutam acum...

- Ne reamintim cum ne autentificam
- Ne reamintim cum are loc autorizarea
- Intelegem cum se populeaza userul curent



# Autentificare

- demonstrez cine sunt

# Autentificare

- demonstrez cine sunt



# Autentificare

- demonstrez cine sunt

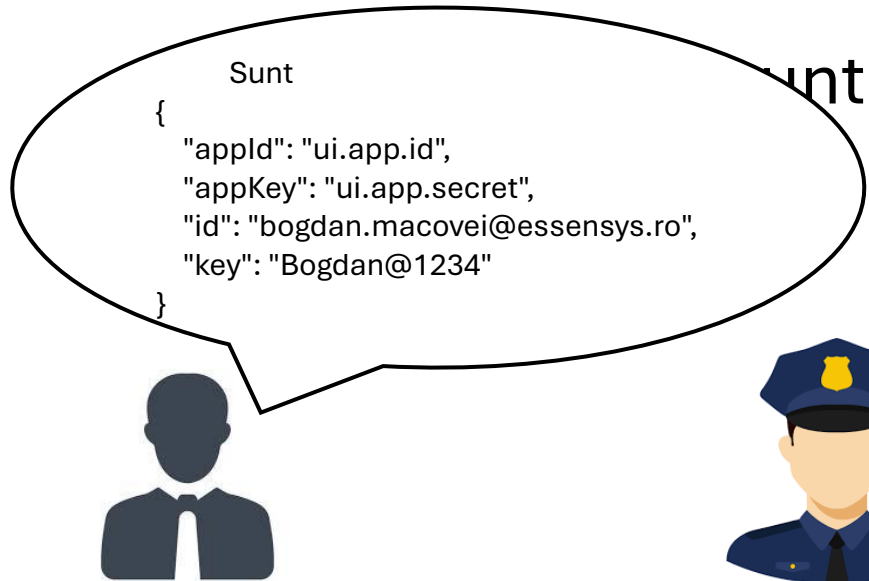


# Autentificare

- demonstrez cine sunt



# Autentificare





# Autentificare

- demonstrez cine sunt



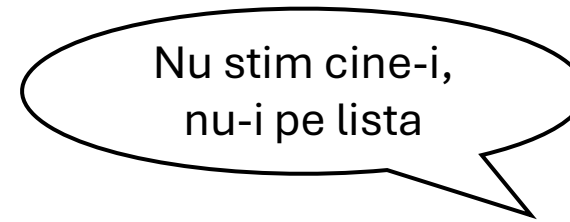
# Autentificare

- demonstrez cine sunt



# Autentificare

- demonstrez cine sunt



Results	Messages							
Id	Email	CreatedBy	CreatedDate	LastModifiedBy	LastModifiedDate	IsActive	Password	RoleId

# Autentificare

- demonstrez cine sunt



# Autentificare

- demonstrez cine sunt

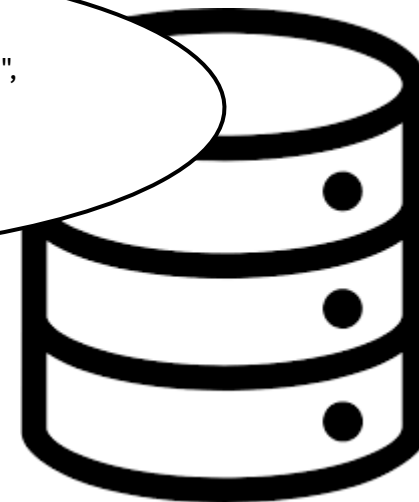


# Autentificare

- demonstrez cine sunt



Trece-ma si pe mine {  
"email": "bogdan.macovei@essensys.ro",  
"password": "Bogdan@1234"  
} sunt din Onesti comuna Gura Vaii satul  
Temelia, va arat buletin...



# Autentificare

- demonstrez cine sunt



Nu, nu, nu, nu suntem  
de la Politie, e asa,  
discutie prieteneasca



# Autentificare

- demonstrez cine sunt



Results		Messages							
	Id	Email	CreatedBy	CreatedDate	LastModifiedBy	LastModifiedDate	IsActive	Password	RoleId
1	F20BAFC3-64E2-47...	bogdan.macovei@essensys.ro	F20BAFC3-64E2-474...	2024-07-26 ...	F20BAFC3-64E2-474A...	2024-07-26 01:22:50.050	1	t6XxRaxEI8uhk4FPU...	2



# Autentificare

Gata, deci eu sunt

```
{  
  "appId": "ui.app.id",  
  "appKey": "ui.app.secret",  
  "id": "bogdan.macovei@essensys.ro",  
  "key": "Bogdan@1234"  
}
```



Results		Messages							
	Id	Email	CreatedBy	CreatedDate	LastModifiedBy	LastModifiedDate	IsActive	Password	RoleId
1	F20BAFC3-64E2-47...	bogdan.macovei@essensys.ro	F20BAFC3-64E2-474...	2024-07-26 ...	F20BAFC3-64E2-474A...	2024-07-26 01:22:50.050	1	t6XxRaxEI8uhk4FPU...	2

# Autentificare

sunt

Ok, esti pe lista, pune-ti bratara  
asta de claim-uri ca sa nu te mai  
intrebam o vreme, e valabila in  
aceasta seara; esti Visitor, de  
asta ai bratara verde, ca doar  
atat ai platit



Results

Messages

	Id	Email	CreatedBy	CreatedDate	LastModifiedBy	LastModifiedDate	IsActive	Password	RoleId
1	F20BAFC3-64E2-47...	bogdan.macovei@essensys.ro	F20BAFC3-64E2-474...	2024-07-26 ...	F20BAFC3-64E2-474A...	2024-07-26 01:22:50.050	1	t6XxRaxEI8uhk4FPU...	2

# Autentificare

- demonstrez cine sunt

An illustration showing a user silhouette on the left and a police officer silhouette on the right. The user has a speech bubble saying "Ok, ms". Below the user is a green and white token labeled "jwt" with the number "630658" on it. A bracket connects the token to a list of claims.

Ok, ms

Jti: ... (JSON Web Token Identity)  
Sub: ... (subject claim)  
UniqueName: [bogdan.macovei@essensys.ro](mailto:bogdan.macovei@essensys.ro)  
Id: F20BAFC3-64E2-474A-A4EA-A6CE6C83831E  
Email: [bogdan.macovei@essensys.ro](mailto:bogdan.macovei@essensys.ro)  
RoleId: 2

Results		Messages							
	Id	Email	CreatedBy	CreatedDate	LastModifiedBy	LastModifiedDate	IsActive	Password	RoleId
1	F20BAFC3-64E2-47...	bogdan.macovei@essensys.ro	F20BAFC3-64E2-474...	2024-07-26 ...	F20BAFC3-64E2-474A...	2024-07-26 01:22:50.050	1	t6XxRaxEI8uhk4FPU...	2

(identificare unica)  
(este user-ul)

(cel din baza)


# Autentificare

- demonstrez cine sunt



# Autorizare

- ce imi este permis sa fac?



Results			Messages
	Id	RoleName	
1	1	Admin	
2	2	Visitor	

# Autorizare

- ce imi este permis sa fac?



Jti: ... (JSON Web Token Identity)  
Sub: ... (subject claim)  
UniqueName: [bogdan.macovei@essensys.ro](mailto:bogdan.macovei@essensys.ro)  
Id: F20BAFC3-64E2-474A-A4EA-A6CE6C83831E  
Email: [bogdan.macovei@essensys.ro](mailto:bogdan.macovei@essensys.ro)  
RoleId: 2


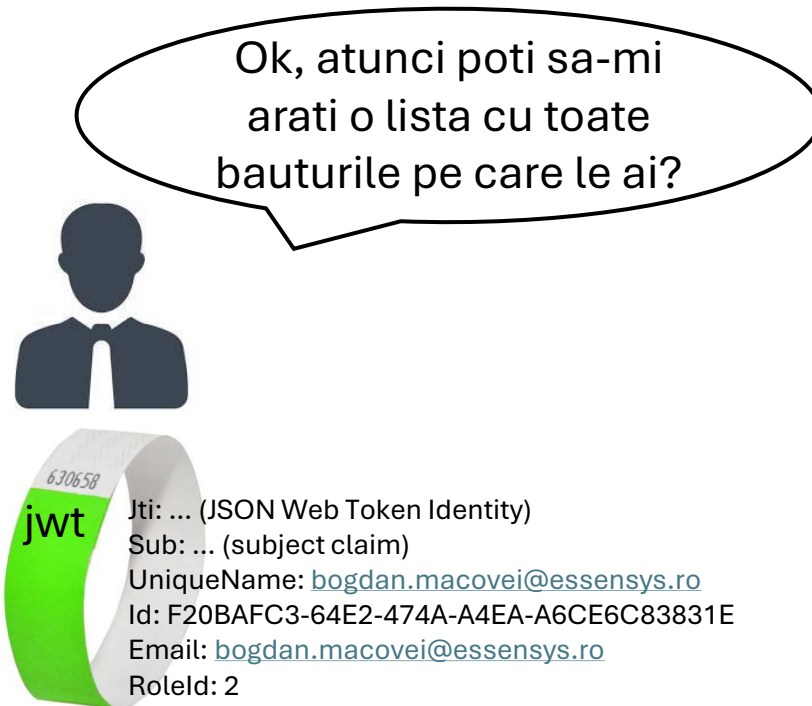
Trebuia sa ai bratara  
de Admin pentru  
asta, dar ai bratara  
de Visitor



Results			Messages
	Id	RoleName	
1	1	Admin	
2	2	Visitor	

# Autorizare

- ce imi este permis sa fac?



Results			Messages
	Id	RoleName	
1	1	Admin	
2	2	Visitor	

# Autorizare

- ce imi este permis sa fac?



Jti: ... (JSON Web Token Identity)  
Sub: ... (subject claim)  
UniqueName: [bogdan.macovei@essensys.ro](mailto:bogdan.macovei@essensys.ro)  
Id: F20BAFC3-64E2-474A-A4EA-A6CE6C83831E  
Email: [bogdan.macovei@essensys.ro](mailto:bogdan.macovei@essensys.ro)  
RoleId: 2



Results			Messages	
	Id	RoleName		
1	1	Admin		
2	2	Visitor		



# Autorizare

- ce imi este permis sa fac?



Jti: ... (JSON Web Token Identity)  
Sub: ... (subject claim)  
UniqueName: [bogdan.macovei@essensys.ro](mailto:bogdan.macovei@essensys.ro)  
Id: F20BAFC3-64E2-474A-A4EA-A6CE6C83831E  
Email: [bogdan.macovei@essensys.ro](mailto:bogdan.macovei@essensys.ro)  
RoleId: 2


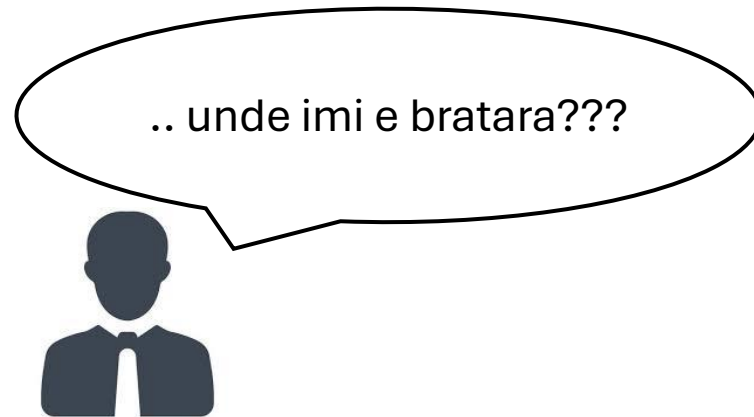
Asta se accepta 😊



Results			Messages	
	Id	RoleName		
1	1	Admin		
2	2	Visitor		

# Autorizare

- ce imi este permis sa fac?



	Id	RoleName
1	1	Admin
2	2	Visitor

# Autorizare

- ce imi este permis sa fac?

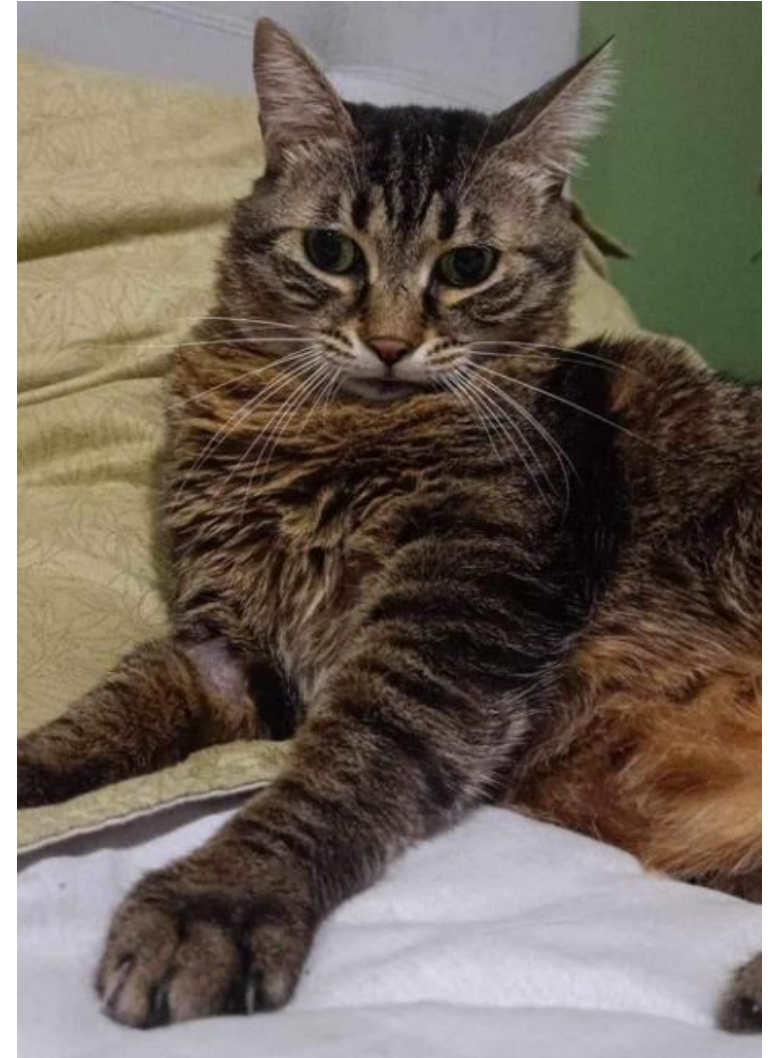
S-a terminat seara, ori  
ne demonstrezi iar cine  
esti, ori pleci



Results			Messages	
	Id	RoleName		
1	1	Admin		
2	2	Visitor		

# Cum se intampla acest lucru?

- DEMO: register si autentificare

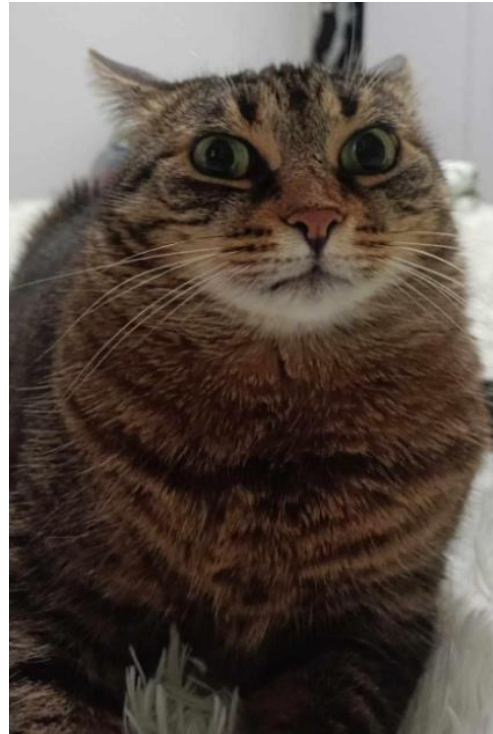


# Dar cum are loc autorizarea?

- Raspunsul scurt: Request lifecycle

# Dar cum are loc autorizarea?

- Raspunsul scurt: Request lifecycle
- Raspunsul lung: tot Request lifecycle



Process pipeline



Controller creation

Process pipeline



```
graph TD; A[Controller creation] --> B[Invoke authentication filters<br/>IAuthenticationFilter]; B --> C[ ]; style C fill:none,stroke:none;
```

Controller creation

Invoke authentication filters  
IAuthenticationFilter



Process pipeline

Controller creation

Invoke authentication filters  
IAuthenticationFilter

Invoke authorization filters  
IAuthorizationFilter



Process pipeline

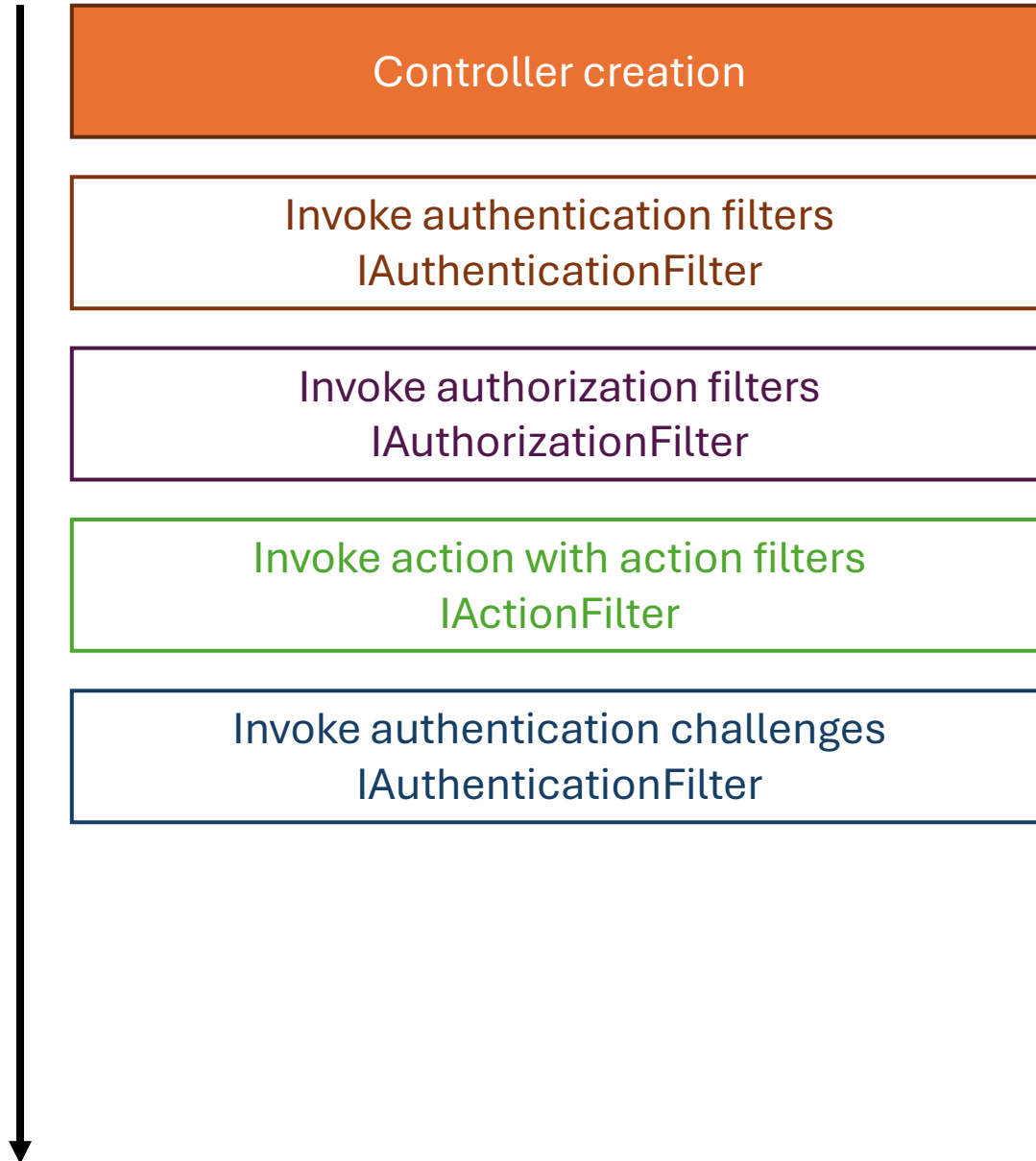
Controller creation

Invoke authentication filters  
IAuthenticationFilter

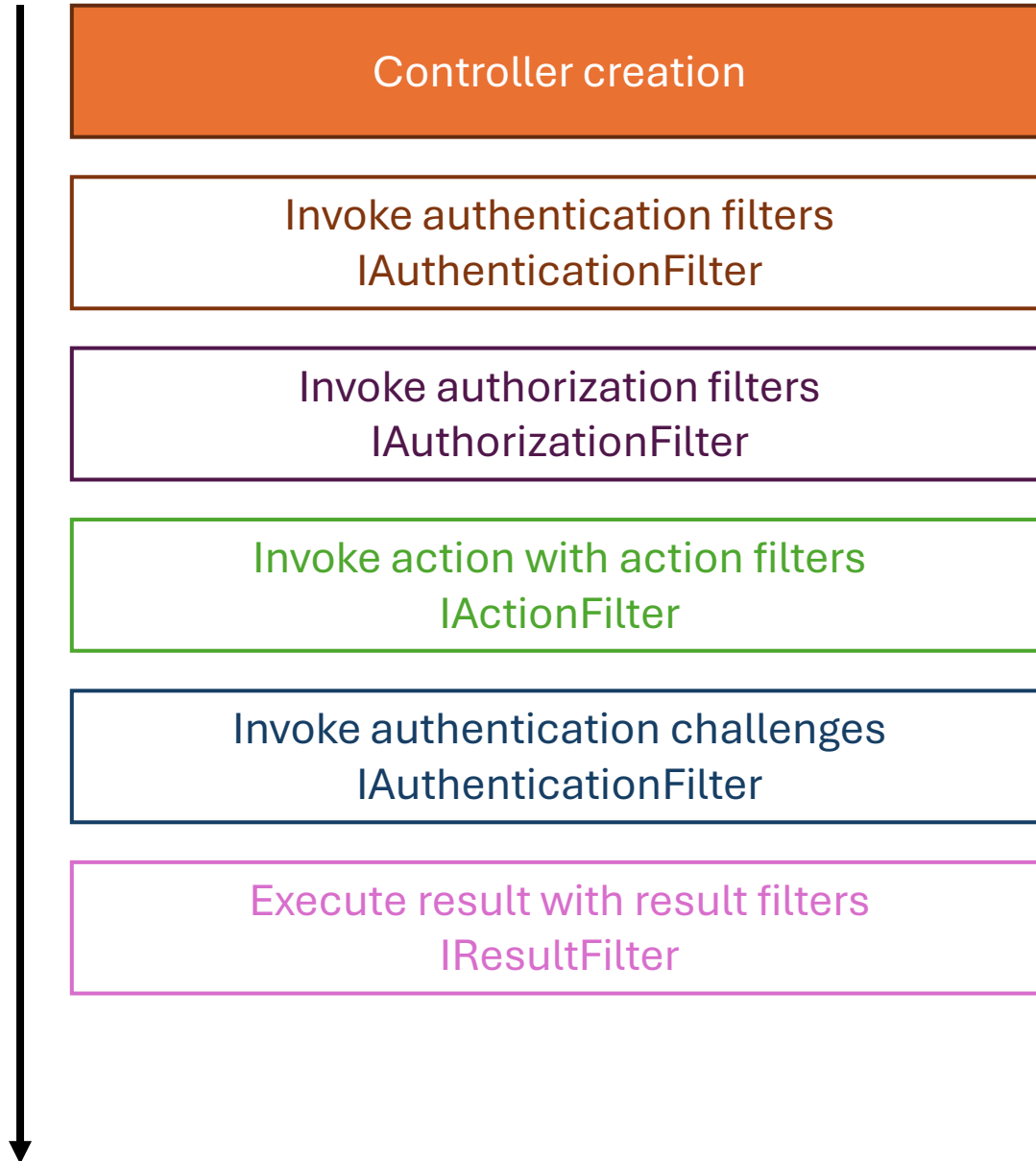
Invoke authorization filters  
IAuthorizationFilter

Invoke action with action filters  
IActionFilter

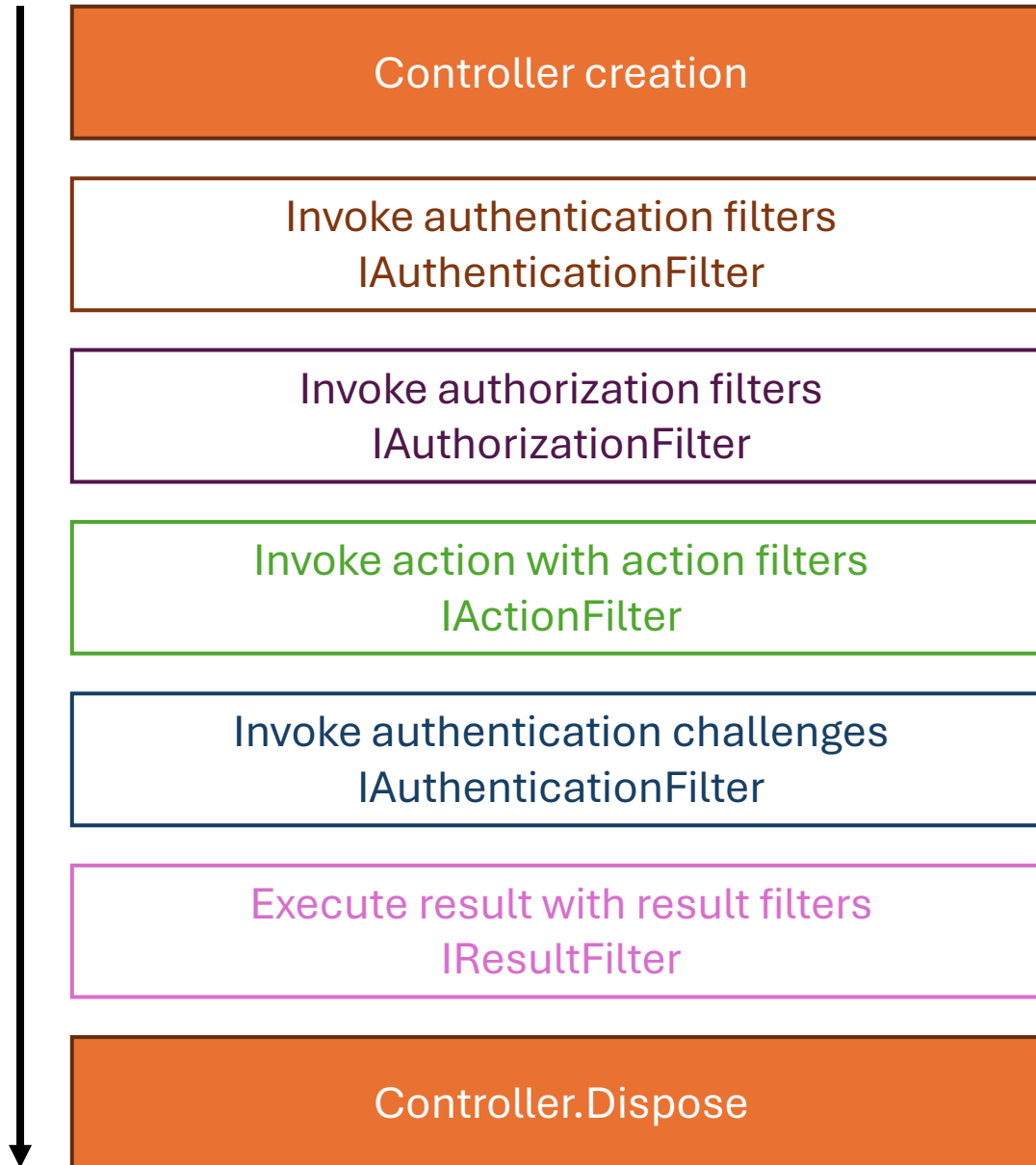
## Process pipeline



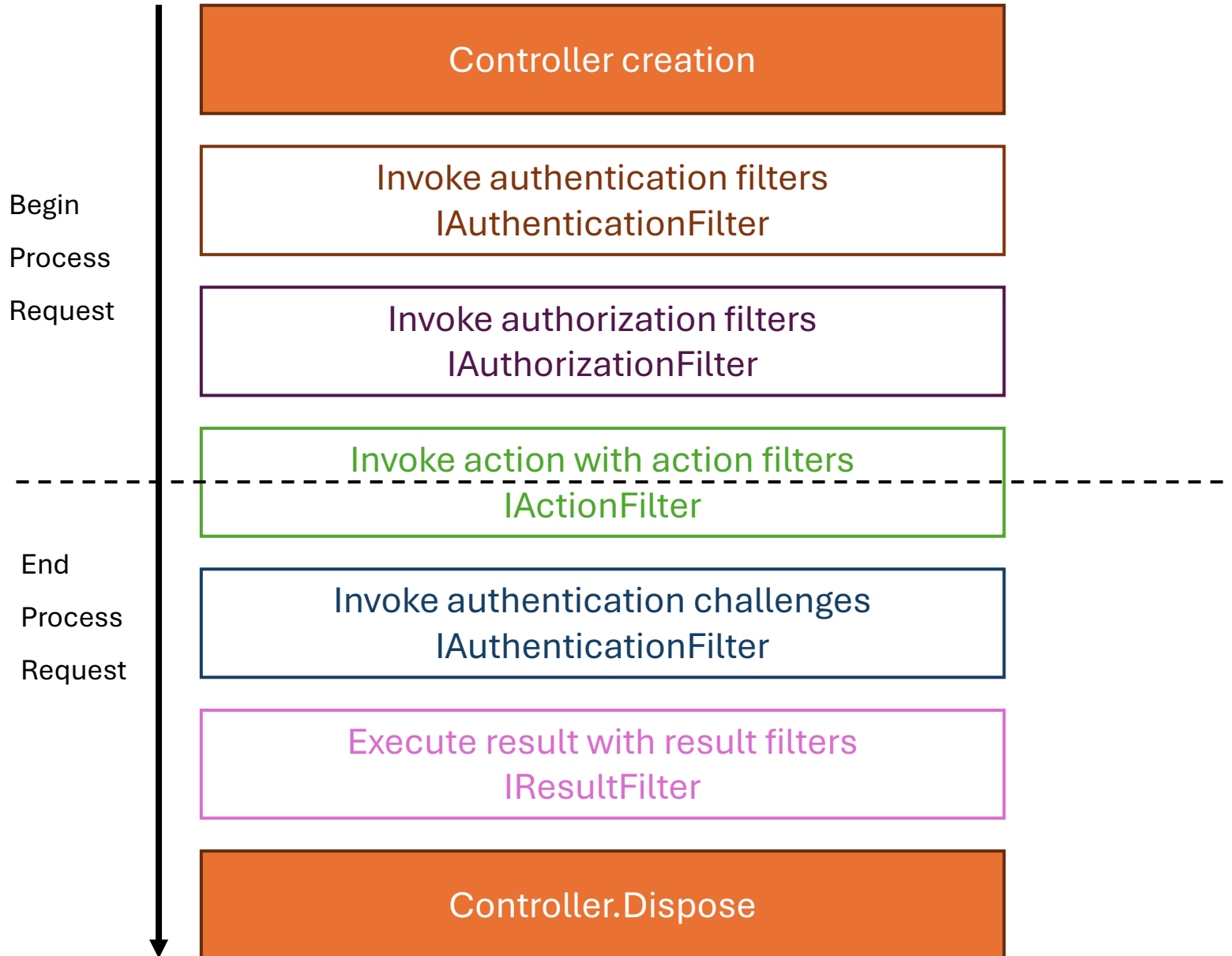
## Process pipeline



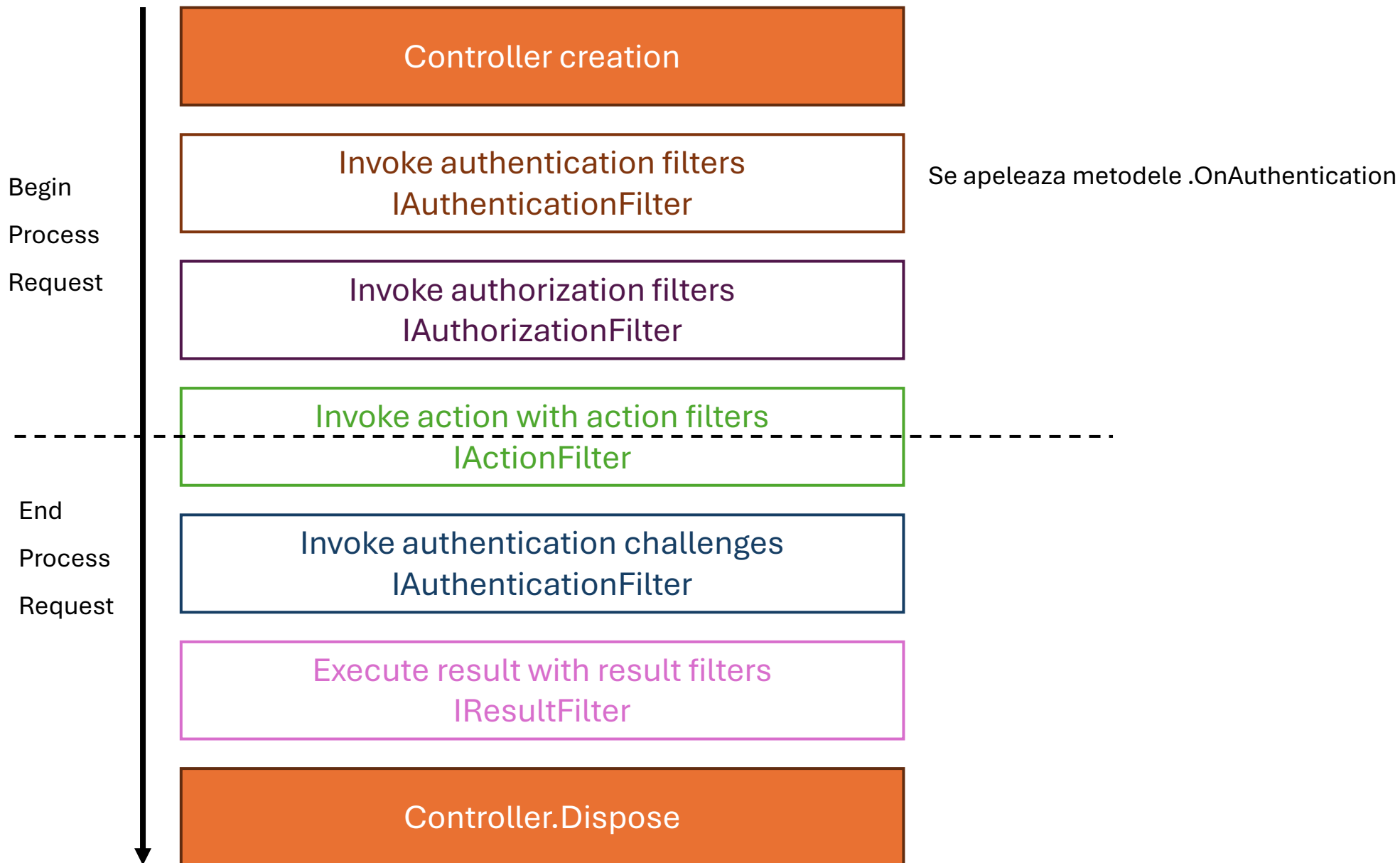
## Process pipeline



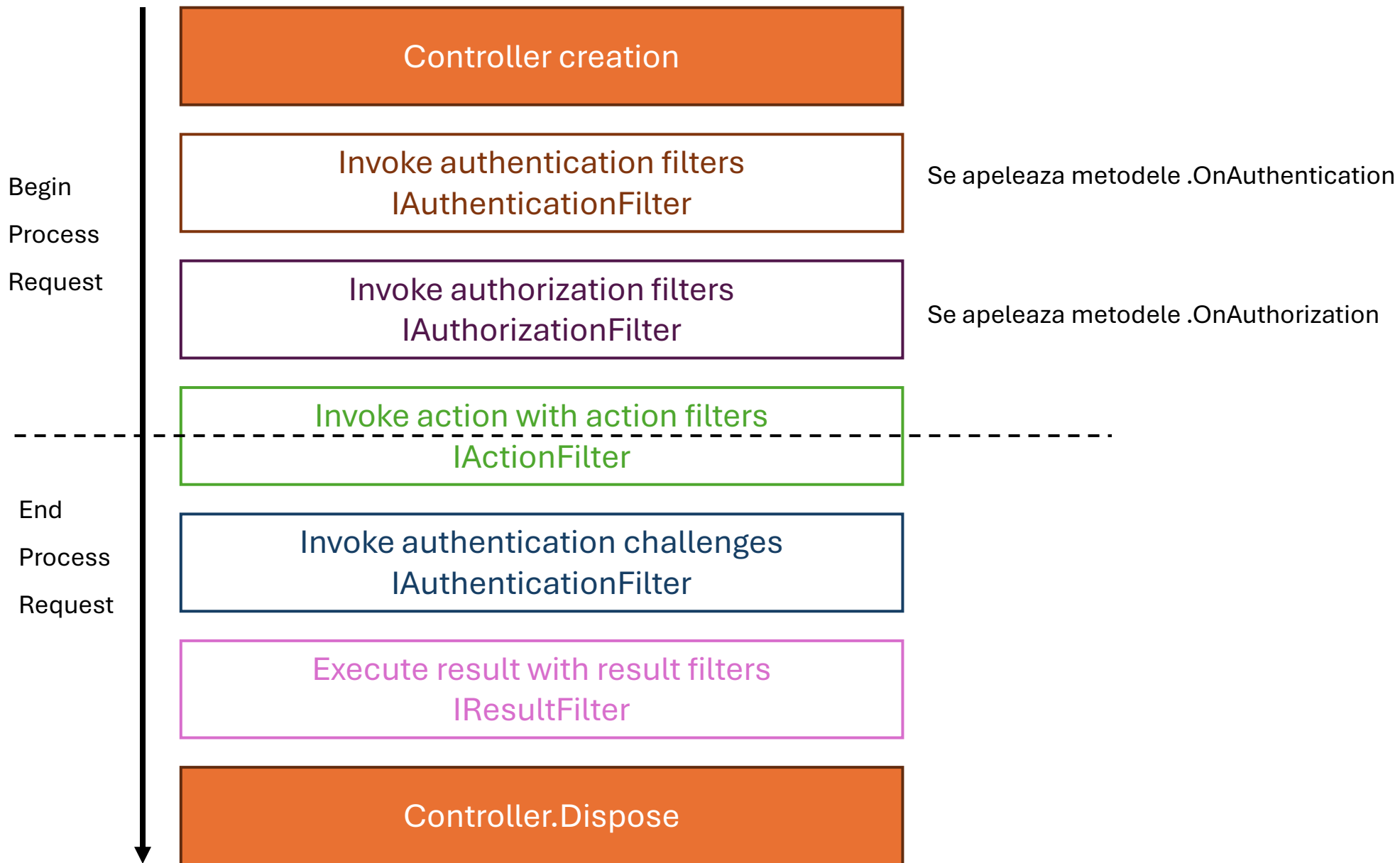
Process pipeline



## Process pipeline

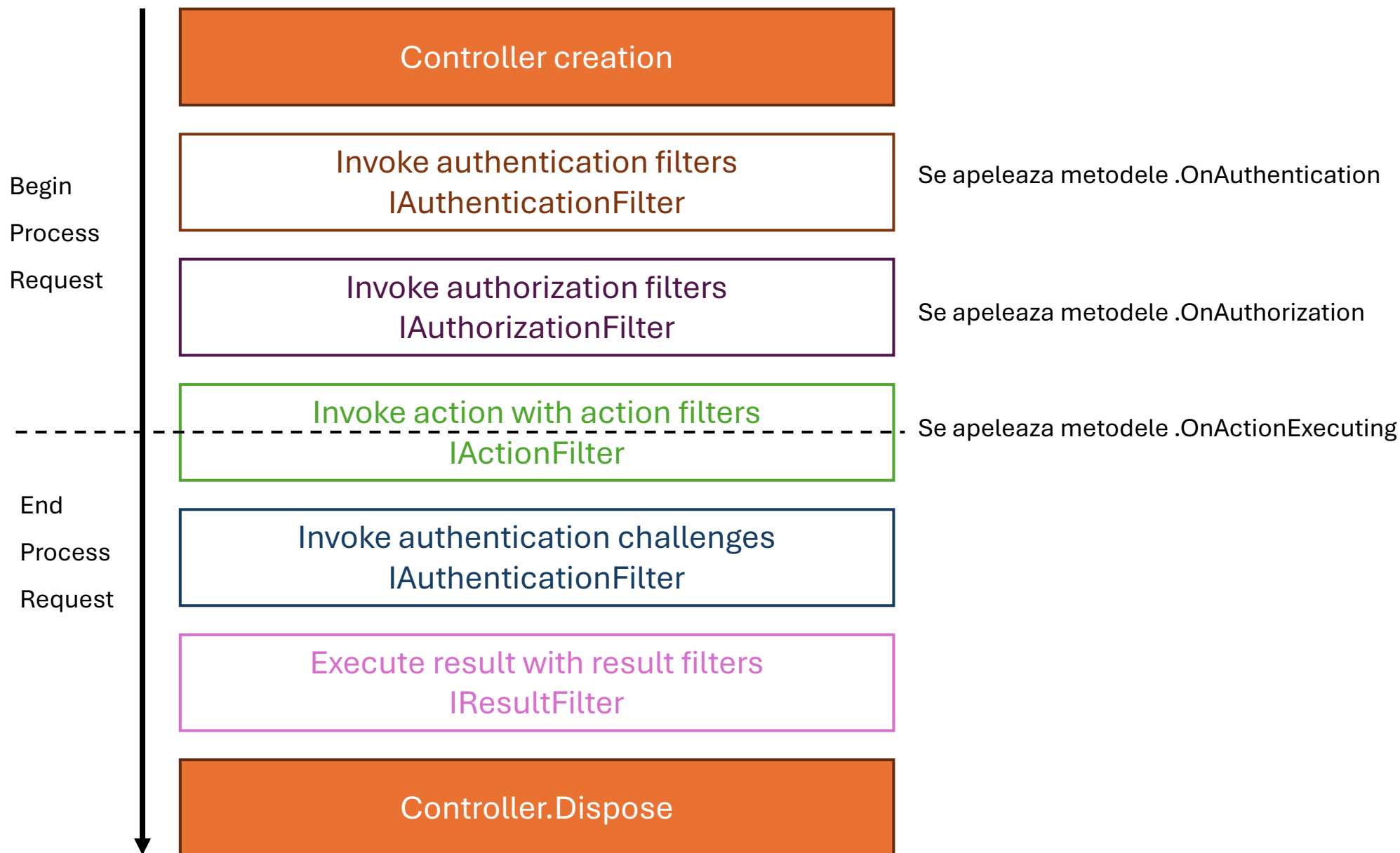


## Process pipeline

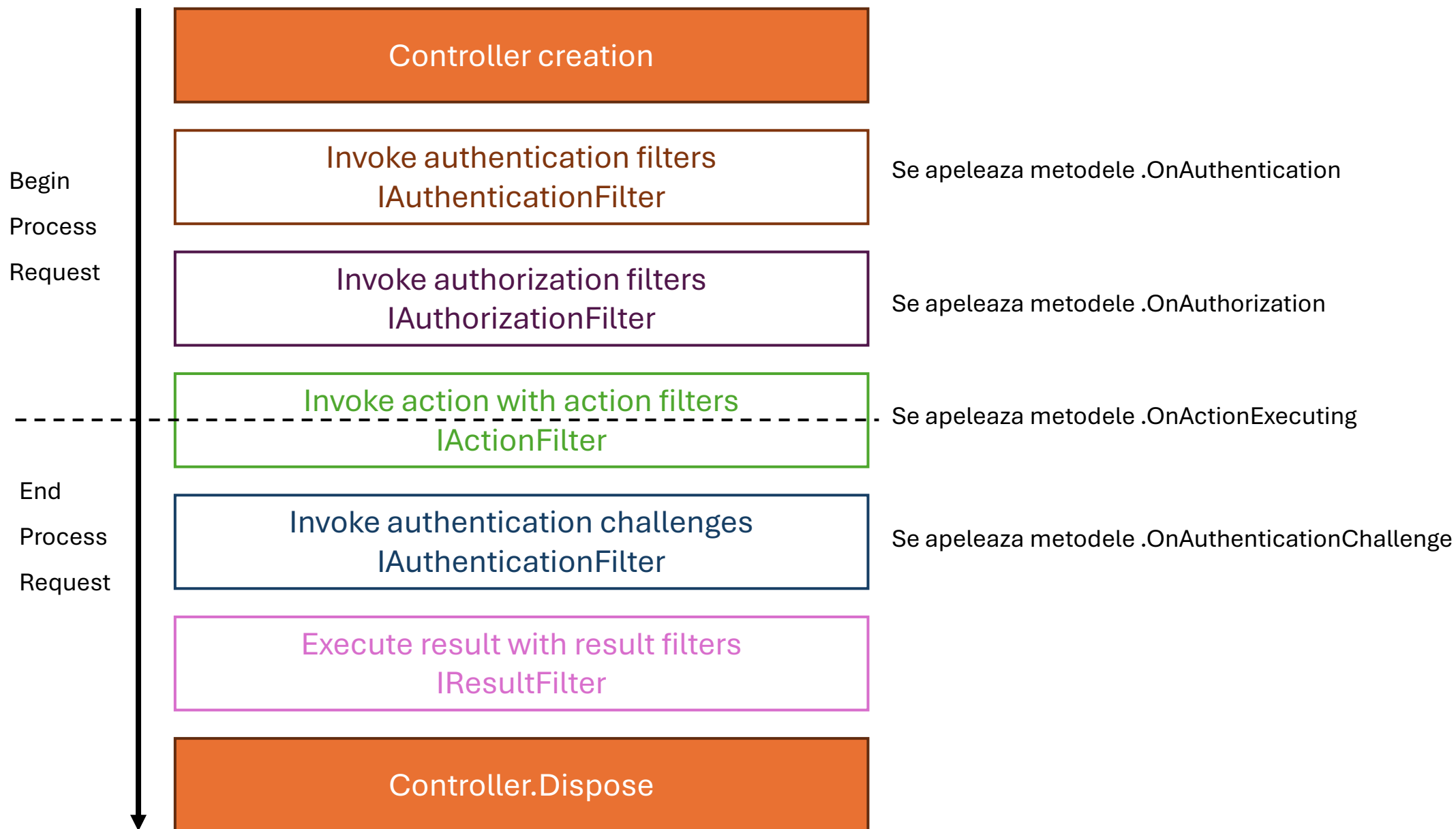




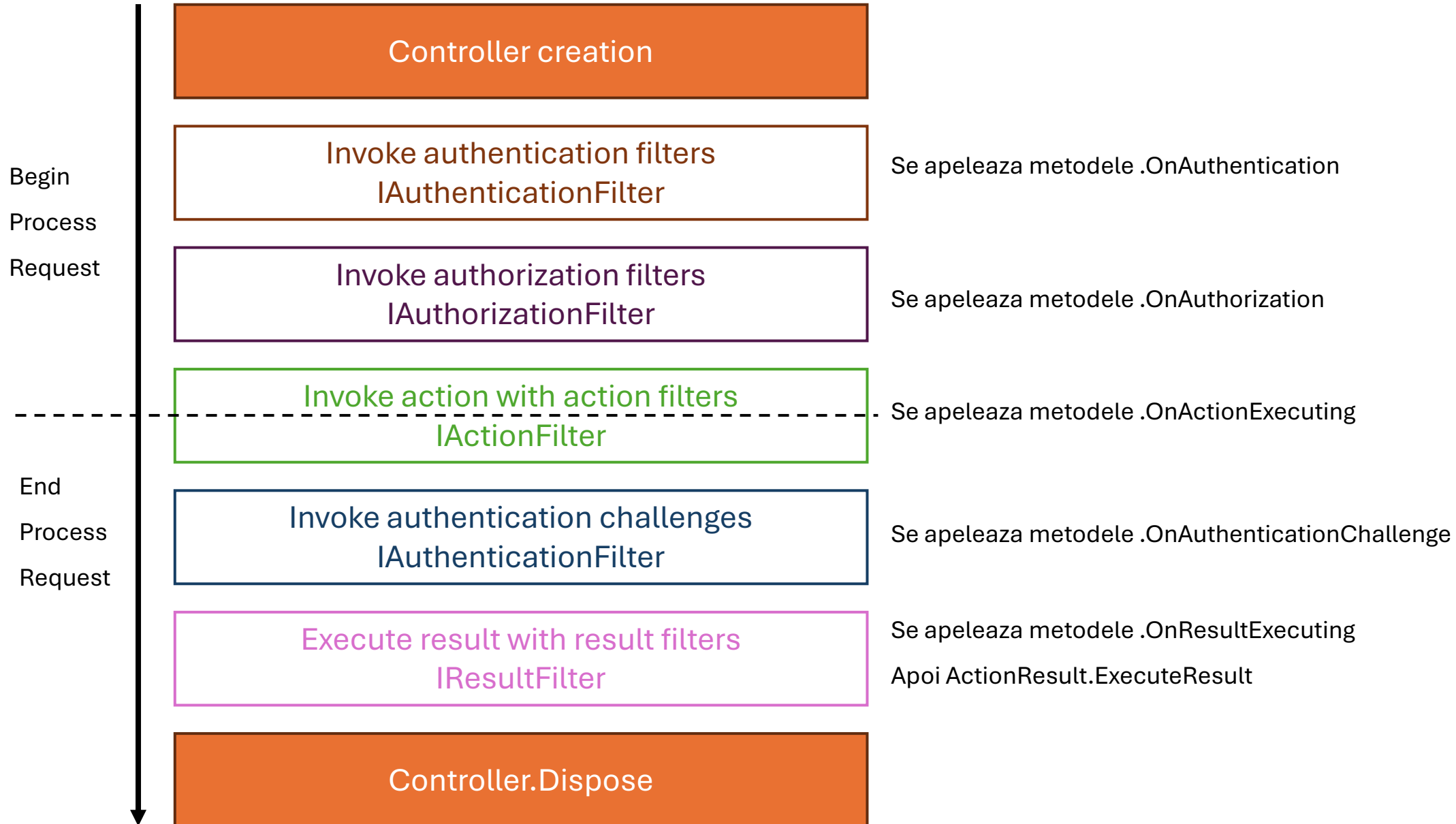
## Process pipeline



## Process pipeline



## Process pipeline



```
[ApiController]
[APIEndpoint]
[Route("author")]
```

1 reference

```
public class AuthorController : ControllerBase
```

```
{
```

```
    private readonly AuthorService AuthorService;
```

0 references

```
    public AuthorController(AuthorService authorService)
```

```
    {
```

```
        AuthorService = authorService;
```

```
    }
```

```
    [HttpGet]
```

```
    [CheckRole(Roles.Visitor)]
```

0 references

```
    public async Task<List<AuthorDTO>> GetAuthors()
```

```
    {
```

```
        return await AuthorService.GetAllAuthors();
```

```
    }
```

```
}
```

```

[ApiController]
[APIEndpoint]
[Route("author")]
1 reference
public class AuthorController : ControllerBase
{
    private readonly AuthorService AuthorService;

    0 references
    public AuthorController(AuthorService authorService)
    {
        AuthorService = authorService;
    }

    [HttpGet]
    [CheckRole(Roles.Visitor)]
    0 references
    public async Task<List<AuthorDTO>> GetAuthors()
    {
        return await AuthorService.GetAllAuthors();
    }
}

```

```

5 references
public class CheckRoleAttribute : TypeFilterAttribute
{
    4 references
    public CheckRoleAttribute(Roles roleToCheck) : base(typeof(CheckRoleFilter))
    {
        Arguments = new object[] { roleToCheck };
    }
}

2 references
public class CheckRoleFilter : IAuthorizationFilter
{
    private readonly Roles RoleToCheck;
    private readonly string RoleClaimName = "RoleId";
    0 references
    public CheckRoleFilter(Roles roleToCheck) ...
}

0 references
public void OnAuthorization(AuthorizationFilterContext context)
{
    var claim = context.HttpContext.User.Claims.FirstOrDefault(c => c.Type == RoleClaimName);
    if (claim == null)
    {
        context.Result = new UnauthorizedResult();
        return;
    }
    try
    {
        if((Roles)int.Parse(claim.Value) != RoleToCheck)
        {
            context.Result = new ForbidResult();
        }
    }
    catch (Exception e)
    {
        context.Result = new ForbidResult();
    }
}

```

```

[ApiController]
[APIEndpoint]
[Route("author")]
1 reference
public class AuthorController : ControllerBase
{
    private readonly AuthorService AuthorService;

    0 references
    public AuthorController(AuthorService authorService)
    {
        AuthorService = authorService;
    }

    [HttpGet]
    [CheckRole(Roles.Visitor)]
    0 references
    public async Task<List<AuthorDTO>> GetAuthors()
    {
        return await AuthorService.GetAllAuthors();
    }
}

```

```

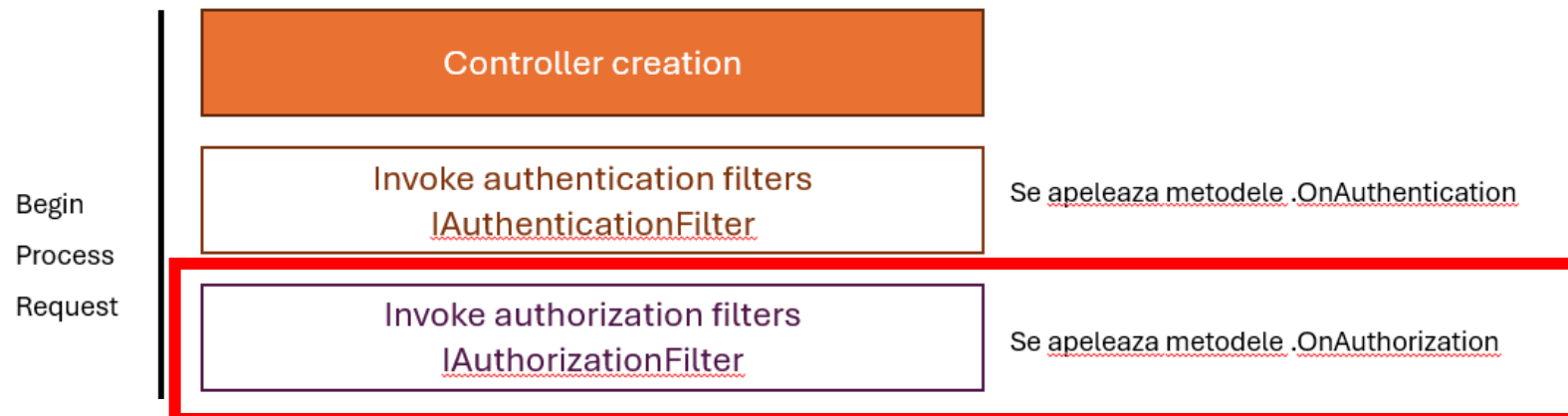
5 references
public class CheckRoleAttribute : TypeFilterAttribute
{
    4 references
    public CheckRoleAttribute(Roles roleToCheck) : base(typeof(CheckRoleFilter))
    {
        Arguments = new object[] { roleToCheck };
    }
}

2 references
public class CheckRoleFilter : IAuthorizationFilter
{
    private readonly Roles RoleToCheck;
    private readonly string RoleClaimName = "RoleId";
    0 references
    public CheckRoleFilter(Roles roleToCheck) ...

    0 references
    public void OnAuthorization(AuthorizationFilterContext context)
    {
        var claim = context.HttpContext.User.Claims.FirstOrDefault(c => c.Type == RoleClaimName);
        if (claim == null)
        {
            context.Result = new UnauthorizedResult();
            return;
        }
        try
        {
            if((Roles)int.Parse(claim.Value) != RoleToCheck)
            {
                context.Result = new ForbidResult();
            }
        }
        catch (Exception e)
        {
            context.Result = new ForbidResult();
        }
    }
}

```

Process pipeline



```

[ApiController]
[APIEndpoint]
[Route("author")]
1 reference
public class AuthorController : ControllerBase
{
    private readonly AuthorService AuthorService;

    0 references
    public AuthorController(AuthorService authorService)
    {
        AuthorService = authorService;
    }

    [HttpGet]
    [CheckRole(Roles.Visitor)]
    0 references
    public async Task<List<AuthorDTO>> GetAuthors()
    {
        return await AuthorService.GetAllAuthors();
    }
}

```

```

[AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, AllowMultiple = false, Inherited = false)]
10 references
public class APIEndpointAttribute : ActionFilterAttribute, IOrderedFilter
{
    private readonly HttpMethodTypes? ActualHttpMethod = null;
    private readonly bool SendRawResult = false;

    4 references
    public APIEndpointAttribute() {...}
    0 references
    public APIEndpointAttribute(bool sendRawResult) {...}
    2 references
    public APIEndpointAttribute(HttpMethodTypes actualHttpMethod) {...}
    0 references
    public APIEndpointAttribute(HttpMethodTypes actualHttpMethod, bool sendRawResult) {...}
    0 references
    public override void OnResultExecuting(ResultExecutingContext context)
    {
        if (context.ModelState.IsValid
            && context.Result is not UnprocessableEntityObjectResult
            && context.Result is not BadRequestObjectResult
            && (context.Filters.Count(f => f is ActionFilterAttribute) == 1 || Order == 0))
        {
            var resultObj = (context.Result as ObjectResult)?.Value;
            var method = ActualHttpMethod ?? context.HttpContext.Request.GetRequestMethodType();
            switch (method)
            {
                case HttpMethodTypes.Post:
                    if (resultObj == null) context.Result = new NotFoundResult();
                    context.Result = new CreatedResult(string.Empty, null);
                    break;
                case HttpMethodTypes.Get:
                case HttpMethodTypes.Put:
                case HttpMethodTypes.Delete:
            }
        }
    }
}

```

```

[ApiController]
[APIEndpoint]
[Route("author")]
1 reference
public class AuthorController : ControllerBase
{
    private readonly AuthorService AuthorService;

    0 references
    public AuthorController(AuthorService authorService)
    {
        AuthorService = authorService;
    }

    [HttpGet]
    [CheckRole(Roles.Visitor)]
    0 references
    public async Task<List<AuthorDTO>> GetAuthors()
    {
        return await AuthorService.GetAllAuthors();
    }
}

```

```

[AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, AllowMultiple = false, Inherited = false)]
10 references
public class APIEndpointAttribute : ActionFilterAttribute, IOrderedFilter
{
    private readonly HttpMethodTypes? ActualHttpMethod = null;
    private readonly bool SendRawResult = false;

    4 references
    public APIEndpointAttribute() {...}
    0 references
    public APIEndpointAttribute(bool sendRawResult) {...}
    2 references
    public APIEndpointAttribute(HttpMethodTypes actualHttpMethod) {...}
    0 references
    public APIEndpointAttribute(HttpMethodTypes actualHttpMethod, bool sendRawResult) {...}
    0 references
    public override void OnResultExecuting(ResultExecutingContext context)
    {
        if (context.ModelState.IsValid
            && context.Result is not UnprocessableEntityObjectResult
            && context.Result is not BadRequestObjectResult
            && (context.Filters.Count(f => f is ActionFilterAttribute) == 1 || Order == 0))
        {
            var resultObj = (context.Result as ObjectResult)?.Value;
            var method = ActualHttpMethod ?? context.HttpContext.Request.GetRequestMethodType();
            switch (method)
            {
                case HttpMethodTypes.Post:
                    if (resultObj == null) context.Result = new NotFoundResult();
                    context.Result = new CreatedResult(string.Empty, null);
                    break;
                case HttpMethodTypes.Get:
                case HttpMethodTypes.Put:
                case HttpMethodTypes.Delete:

```

```

namespace Microsoft.AspNetCore.Mvc.Filters
{
    ...public abstract class ActionFilterAttribute : Attribute, IActionFilter, IFilterMetadata, IAsyncActionFilter, IResultFilter
    {
        protected ActionFilterAttribute();

        public int Order { get; set; }

        public virtual void OnActionExecuted(ActionExecutedContext context);
        public virtual void OnActionExecuting(ActionExecutingContext context);
        ...public virtual Task OnActionExecutionAsync(ActionExecutingContext context, ActionExecutionDelegate next);
        public virtual void OnResultExecuted(ResultExecutedContext context);
        public virtual void OnResultExecuting(ResultExecutingContext context);
        ...public virtual Task OnResultExecutionAsync(ResultExecutingContext context, ResultExecutionDelegate next);
    }
}

```



Si cum se populeaza userul curent?

# Si cum se populeaza userul curent?

```
builder.Services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();
```

# Si cum se populeaza userul curent?

```
builder.Services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();
```

```
builder.Services.AddTransient(s =>
{
    var contextAccesor = s.GetService<IHttpContextAccessor>() ?? throw new ArgumentNullException();
    var context = contextAccesor.HttpContext;
    return context?.User ?? new ClaimsPrincipal();
});
```

# Si cum se populeaza userul curent?

```
builder.Services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();
```

```
builder.Services.AddTransient(s =>
{
    var contextAccesor = s.GetService<IHttpContextAccessor>() ?? throw new ArgumentNullException();
    var context = contextAccesor.HttpContext;
    return context?.User ?? new ClaimsPrincipal();
});
```

+ tot ce am vazut in request lifecycle pana acum

+ pe DEMO daca nu l-am facut deja

Intrebari, curiositati, propuneri?