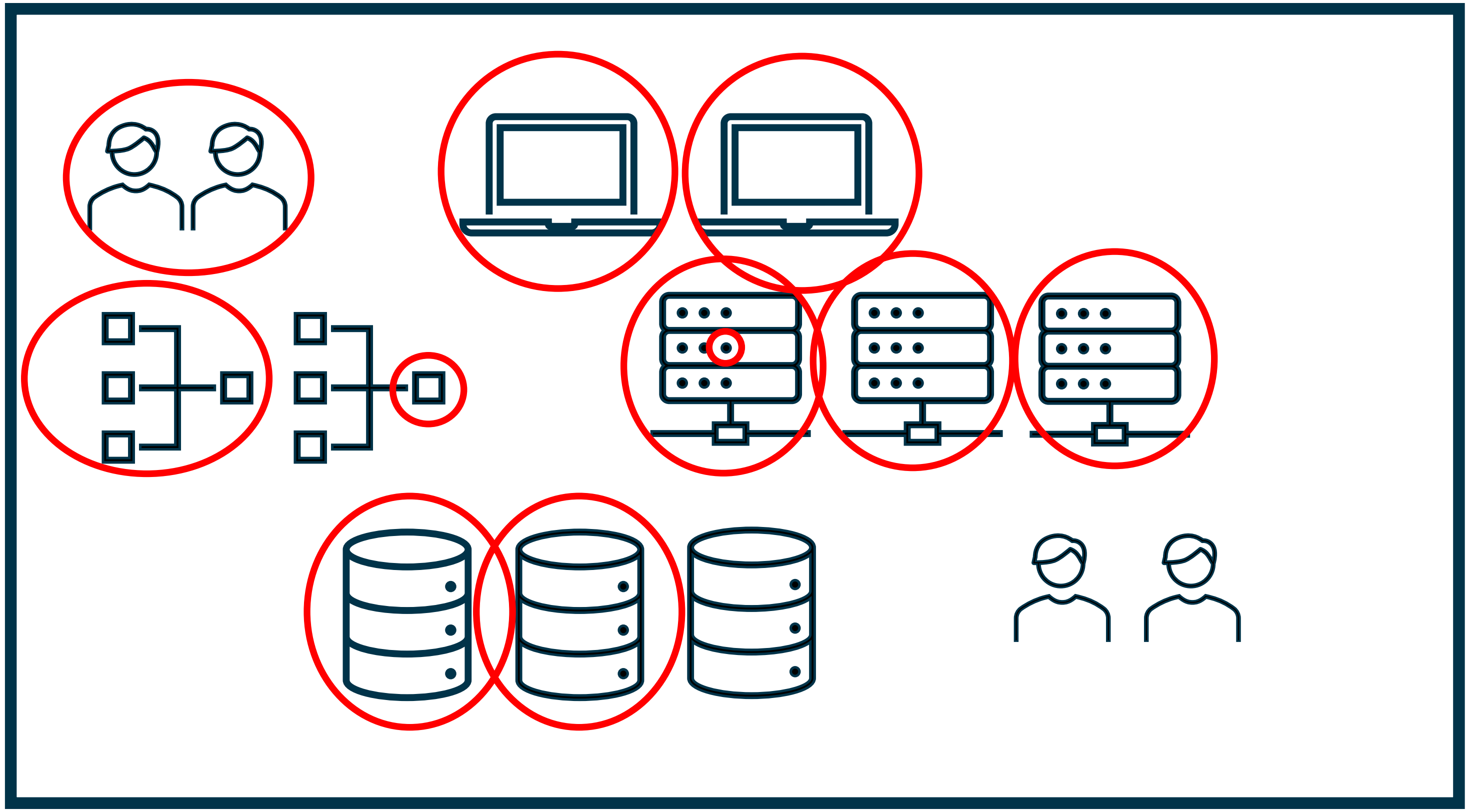
omega point.



«Secure By Design»

Domain driven security

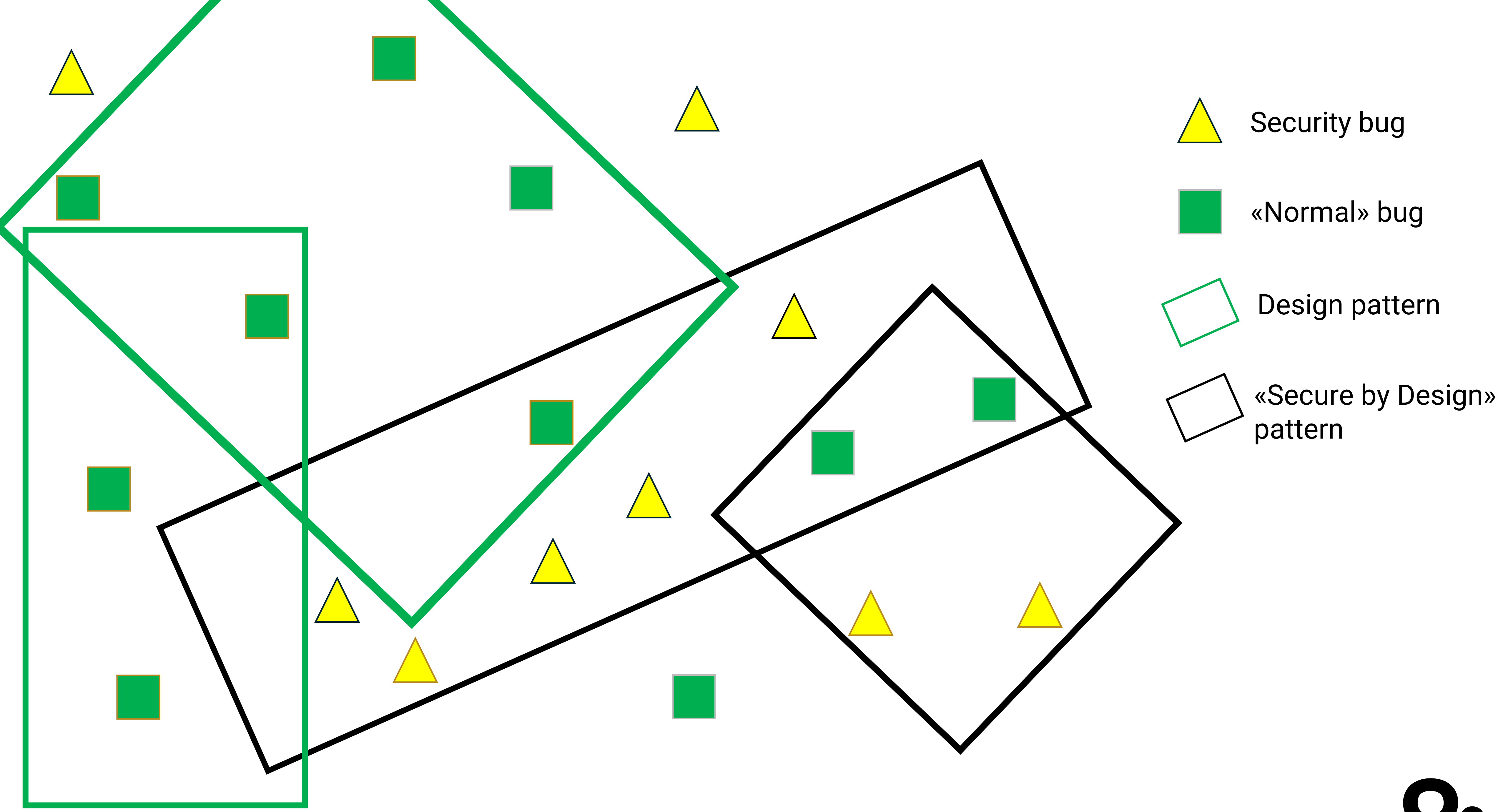
Cybersecure digitalization

Håkon Anders Strømsodd

- 25 år
- Sandefjord
- Datateknologi + AI (NTNU 2023)
- Sommerstudent (2022) og
  Konsulent (2023+) i Omegapoint Norge

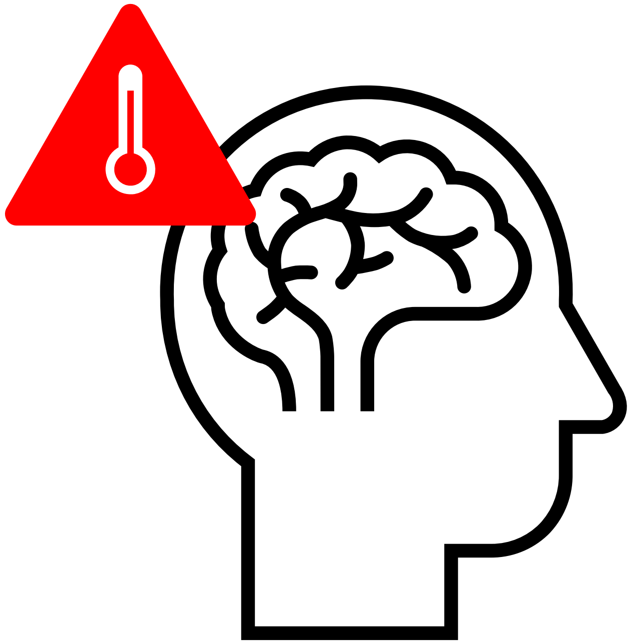

Secure by Design
Dan Bergh Johnsson
Daniel Deogun
Daniel Sawano
Foreword by Daniel Terhorst-North
MANNING

omega point.


Security bug
«Normal» bug
Design pattern
«Secure by Design» pattern

“Traditional” approach

- ✓ OWASP
- ✓ XSS
- ✓ SQL Injection
- ✓ Attack vectors
- ✓ 0-day vulnerabilities
- ✓ …



“Traditional” approach

- Possible XSS and SQL injection
- `<script>alert("Hacked");</script>`

```
public class User
{
    public int Id { get; set; }
    public string Username { get; set; }

    public User(int id, string username)
    {
        Id = id;
        Username = username;
    }
}
```

“Traditional” approach

```
public class User
{
    public int Id { get; set; }
    public string Username { get; set; }

    public User(int id, string username)
    {
        Id = notNull(id);
        Username = validateForXSSAndSQLI(username);
    }
}
```

Secure by Design

```
public class User
{
    private static readonly int USERNAME_MAX_LENGTH = 20;
    private static readonly int USERNAME_MIN_LENGTH = 5;
    private static readonly string USERNAME_ALLOWED_CHARS = "[a-zA-Z0-9_-]+";
    private int Id { get; }
    private string Username { get; }
    public User(int id, string username)
    {
        var trimmedUsername = username.Trim();
        lengthBetween(trimmedUsername, USERNAME_MAX_LENGTH, USERNAME_MIN_LENGTH);
        matches(trimmedUsername, USERNAME_ALLOWED_CHARS);
        Id = notNull(id);
        Username = trimmedUsername;
    }
}
```

Secure by Design

```
public class Username
{
    private static readonly int USERNAME_MAX_LENGTH = 20;
    private static readonly int USERNAME_MIN_LENGTH = 5;
    private static readonly int USERNAME_ALLOWED_CHARS = "[a-zA-Z0-9_-]+";

    public string Value { get; }

    public Username(string username)
    {
        var trimmedUsername = username.Trim();
        lengthBetween(trimmedUsername, USERNAME_MAX_LENGTH, USERNAME_MIN_LENGTH)
        matches(trimmedUsername, USERNAME_ALLOWED_CHARS);
        Value = trimmedUsername;
    }
}
```

Secure by Design

```
public class User
{
    public int Id { get; }
    public Username Username { get; }
    public User(int id, Username username)
    {
        Id = notNull(id);
        Username = username;
    }
}
```

Postal Code


posten
Frist for innlevering:
Kl. 16.00 mandag – fredag
posten
Frist for innlevering:
Kl. 16.00 mandag – fredag
posten
Frist for innlevering:
Kl. 16.00 mandag – fredag

```
Person CreatePerson(string name, string address, int postalCode)
{
    var person = new Person(name, address + postalCode);
    db.Save(person);
    return person;
}

CreatePerson(“Binneveien 13B“, “Håkon Anders Strømsodd", 0774);
```

```
User CreatePerson(Name name, Address address, PostalCode postalCode)
{
    return new Person(name, address + postalCode);
}

CreatePerson(
        new Address("Binneveien 13B"),
        new Name("Håkon Anders Strømsodd"),
        new PostalCode(0484));
```

Money!

```
public class Money
{
    private static readonly string[] _allowedCurrencies =
        new string[] {"NOK", "EUR", "USD", "SEK"};

    public decimal Amount { get; }
    public string Currency { get; }

    public Money(decimal amount, string currency)
    {
        Amount = amount;
        Currency = isOneOf(currency, _allowedCurrencies);
    }
}
```

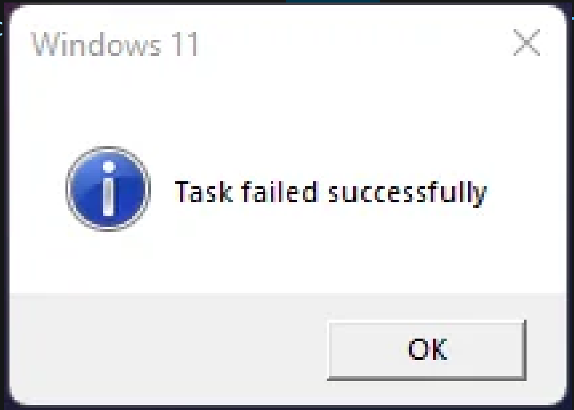
```
public class Money
{
    //......
    public static Money operator +(Money a, Money b)
    {
        if(a.Currency != b.Currency)
            throw new ArgumentException("Different currencies cannot be added");

        return new Money(a.Amount + b.Amount, a.Currency);
    }

    public static Money operator -(Money a, Money b)
    {
        if(a.Currency != b.Currency)
            throw new ArgumentException("Different currencies cannot be subed");

        return new Money(a.Amount - b.Amount, a.Currency);
    }
}
```

Failures are not Exceptions

```
public class Account
{
    //......
    public void Transfer(Money money, Account toAccount)
    {
        if (Balance() < money) {
            throw new InsufficientFundsException();
        }
        //...

        return;
    }

    Money Balance()
    {
        //...
    }
}
```

Failures are not Exceptions

```
public class Account
{
    //......
    public Result Transfer(Money money, Account toAc
    {
        if (Balance() < money) {
            return AccountResult.InsufficientFunds;
        }
        //...

        return AccountResult.Success;
    }

    Money Balance()
    {
        //...
    }
}
```




```
public enum AccountResult {
    Success,
    fficientFunds
```

Handling permissions

- Don't do custom stuff everywhere
- Permissions are domain logic

- Three levels of authorization:
  - Functional level
  - Object level
  - Property level

```
interface IProductPermissionService
{
    bool CanRead(Product product);
    bool CanWrite(Product product);

    //...
}
```

Secure by Design

Focus on good design

Less bugs

More secure systems

Workshop Introduction

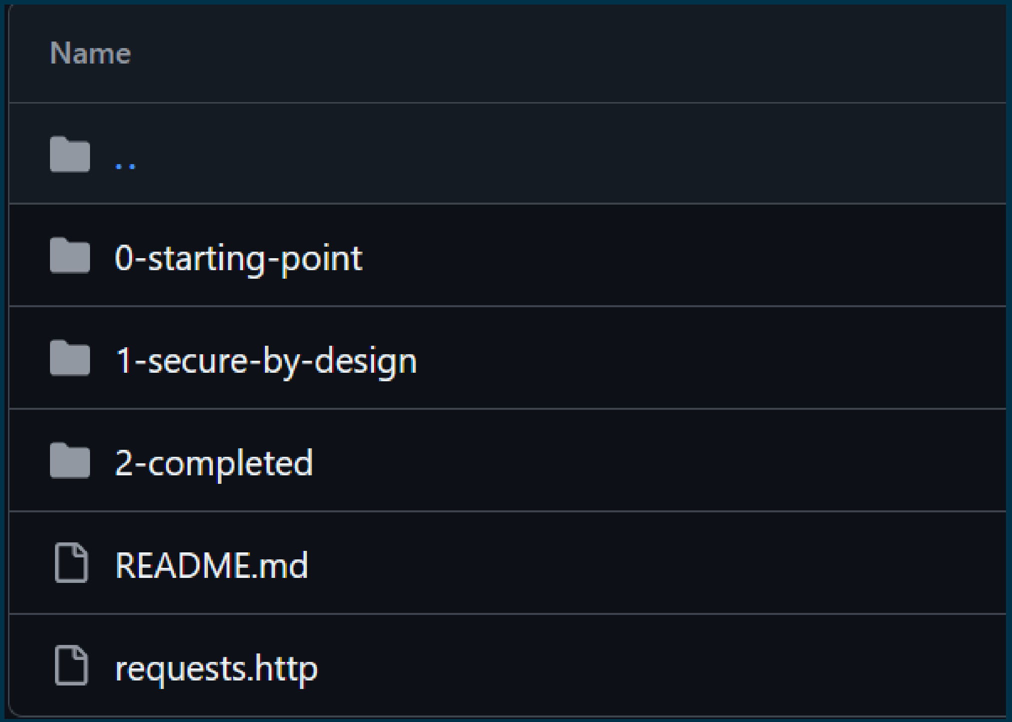
Secure by design

Repository Structure

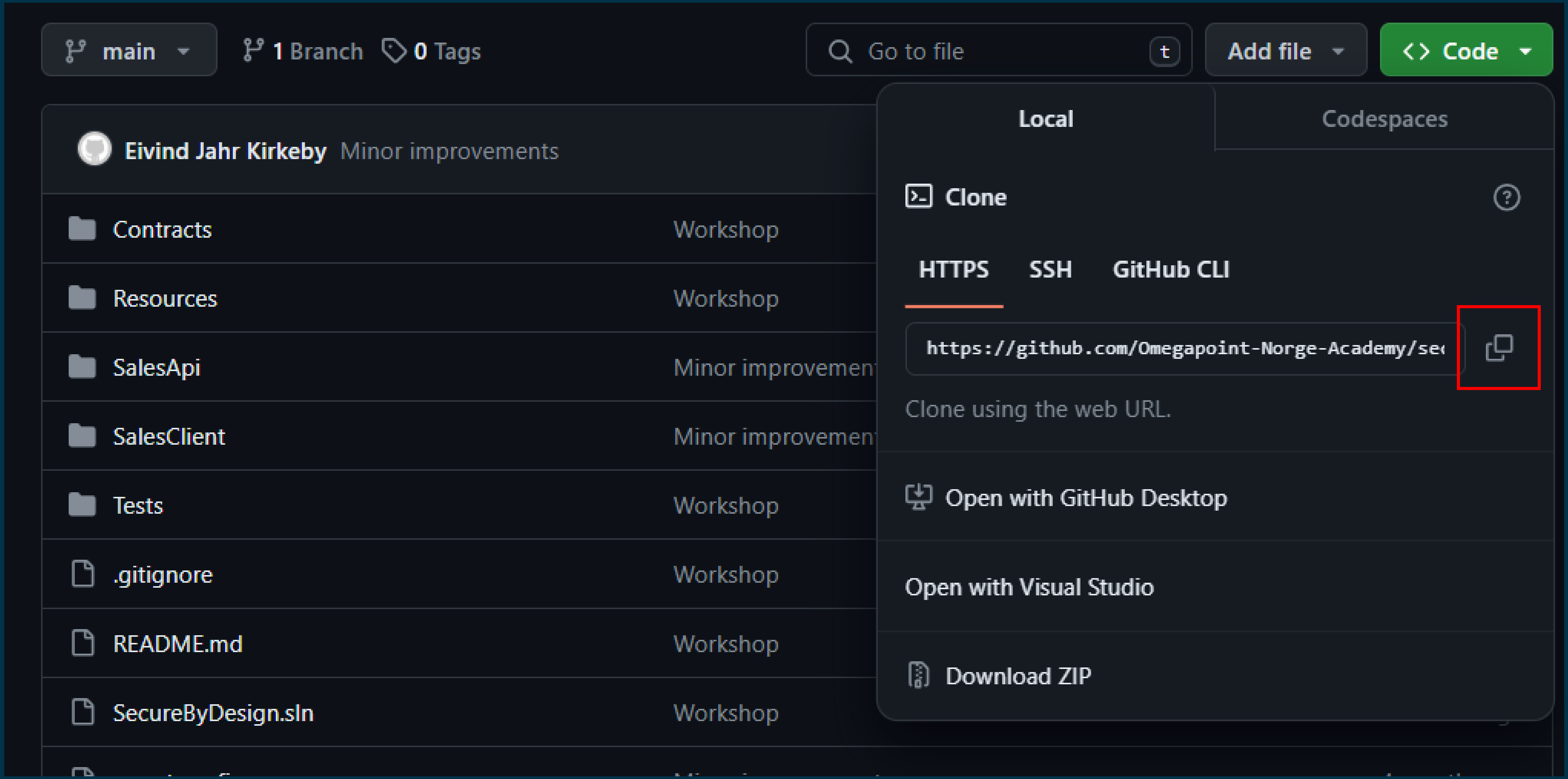
Contracts	Workshop
Resources	Workshop
SalesApi	Remove files that aren't neec
.gitignore	Workshop
OAuht2 and OpenID Connect best practices.pdf	Presentations
README.md	Remove files that aren't neec
SecureByDesign.sln	Remove files that aren't neec
Workshop introduction.pdf	Presentations
nuget.config	Minor improvements
secure-by-design-keynote.pdf	Updated slides

Sales API

- Start with step 0 in the readme
- Start coding in folder 0
- No shame in having a peek at the spoilers



bit.ly/op-secure-by-design



Git Clone

```
$ git clone https://github.com/Omegapoint-Norge-Academy/secure-by-design.git
$ cd secure-by-design/
$ code .
```

OPTIONAL

requests.http

- Install the VSCode extension: REST Client
- Send HTTP requests from VSCode
- Or use other tool of your choice to test manually

```
###
Send request
GET https://localhost:7094/api/product

###
Send request
# @name getReadToken
POST https://omegapoint-norge-workshop.eu.auth0.com/oauth/token
content-type: application/json

{
    "client_id":"<client-id-here>",
    "client_secret":"<client-secret-here>",
    "audience":"sales-api",
    "grant_type":"client_credentials",
    "scope": "products.read"
}

###
@readToken = {{getReadToken.response.body.access_token}}

###
Send request
GET http://localhost:5017/api/product
Authorization: Bearer {{readToken}}
```

OPTIONAL

Run the code

- Run the code with .NET 8

```
> cd SalesAPI
> cd 5-data-access-validation
> dotnet run
```

TL;DR

- Start coding in folder 0 Sales API
  - SalesAPI/0-starting-point/

Workshop: bit.ly/op-secure-by-design

Client IDs and Secrets: bit.ly/op-secure-by-design-secrets