# “Classic” Authentication Scenarios

Dominick Baier
http://leastprivilege.com
@leastprivilege

pluralsight
hardcore dev and IT training

# Agenda

- Windows authentication
- Basic authentication
- SSL client certificates

- Learn a bit more about Katana and authentication middleware

# Windows Authentication

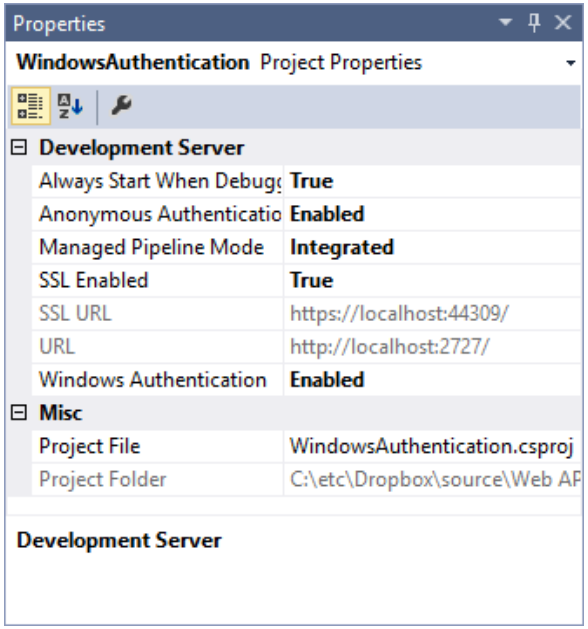
- **Classic intranet scenarios**
  - all parties belong to Active Directory
  - no code, just configuration

- **Not really suited for anything else**
  - CSRF issues

# System.Web Hosting

1)

```
<system.webServer>
  <security>
    <authentication>
      <windowsAuthentication enabled="true" />
    </authentication>
  </security>
</system.webServer>
```




2)

```
<system.web>
  <authentication mode="Windows" />
</system.web>
```

# Katana hosting

```
public static IAppBuilder UseWindowsAuthentication(this IAppBuilder app)
{
    object value;
    if (app.Properties.TryGetValue("System.Net.HttpListener", out value))
    {
        var listener = value as HttpListener;
        if (listener != null)
        {
            listener.AuthenticationSchemes =
                AuthenticationSchemes.IntegratedWindowsAuthentication;
        }
    }

    return app;
}
```

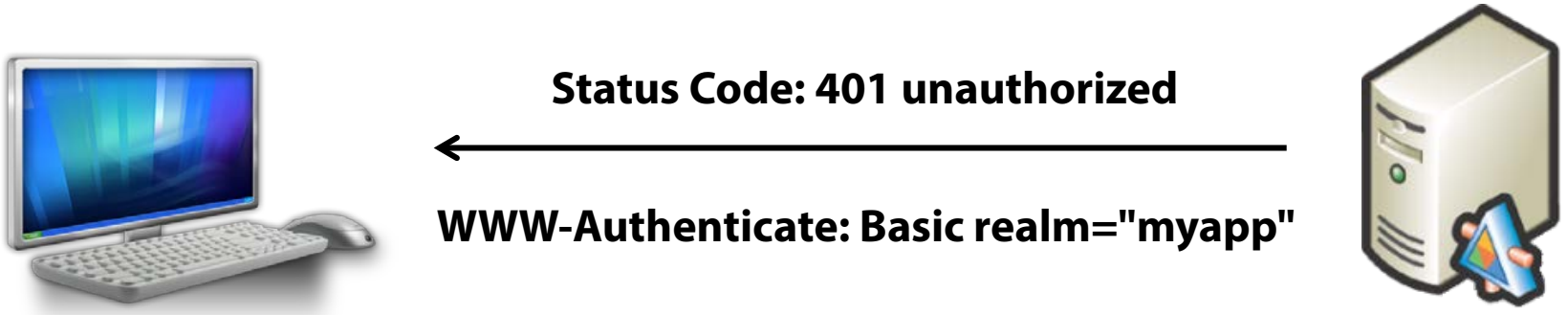
# Basic Authentication

- **Anti pattern**
  - The client must store the secret or obtain it from the user (on every request)
    - storage must be done in clear text (or reversible encryption)
  - Server has to validate the secret on every request
    - high computational cost due to brute force protection

- **The probability of accidental exposure of the secret is increased**

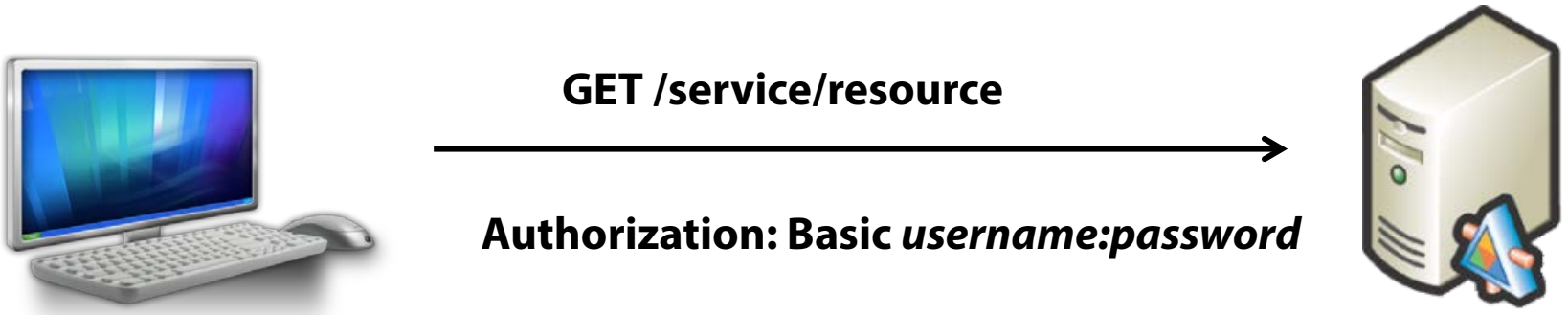
# Basic Authentication on the Wire I

- Uses *Basic* scheme
- Realm can give additional context

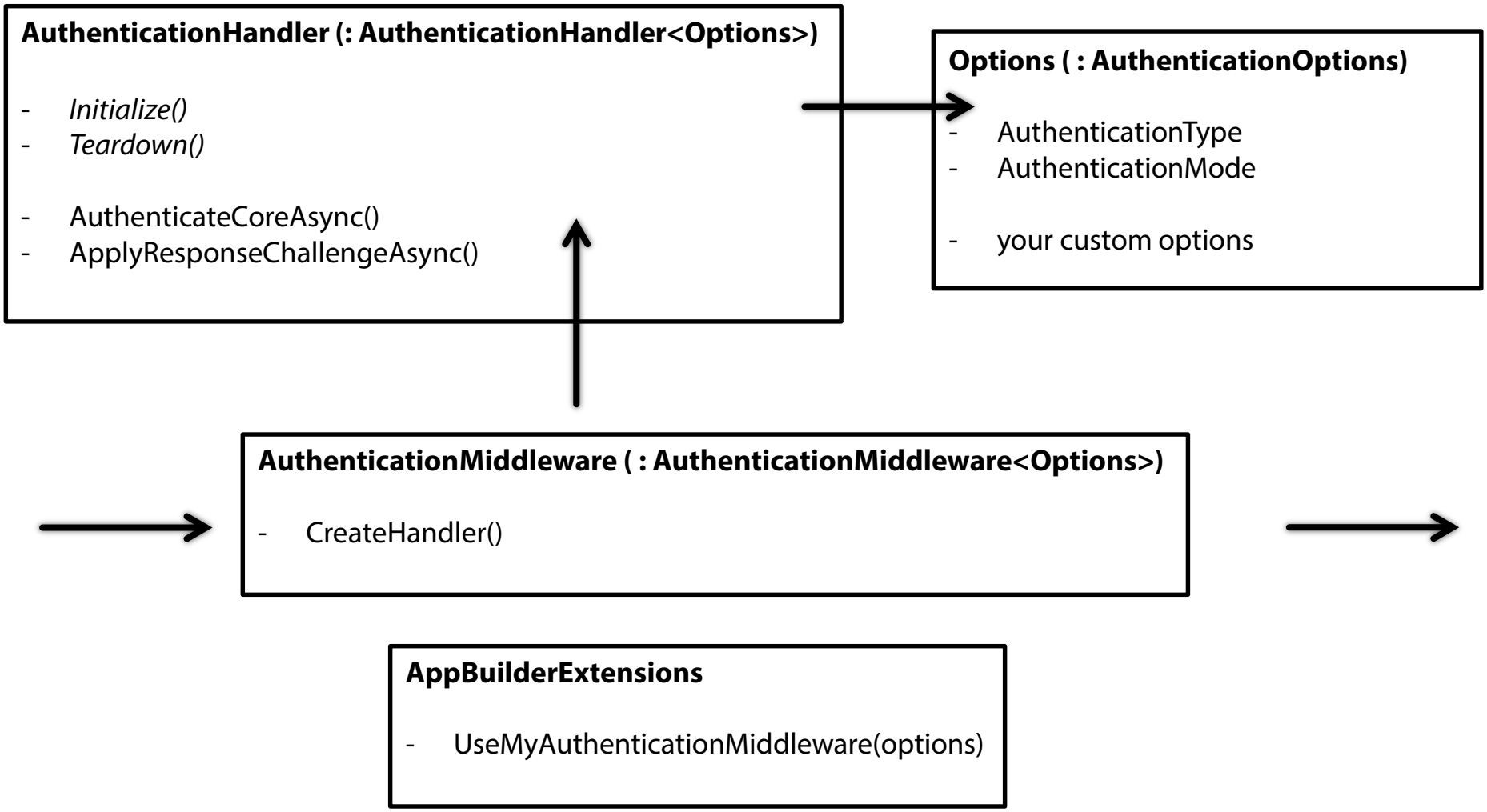


# Basic Authentication on the Wire II

- Base64 encoded credentials on authorization header



# Katana Authentication Framework



# X.509 Client Certificates

- **Popular option for "high security" scenarios**
  - two factor authentication
  - can be bound to additional hardware
    - smart cards, USB tokens …
  - can be combined with other authentication methods
    - e.g. Basic Authentication

# Enabling Client Certificates

IIS

```
<system.webServer>
  <security>
    <access sslFlags="Ssl, SslNegotiateCert, SslRequireCert" />
  </security>
</system.webServer>
```

...or use http.sys command line tools

# Accessing the Client Certificate

- **Can retrieve it from the RequestContext**
  - or via Katana middleware
- **Pre-validation is host specific**
  - application needs to validate as well

```
public IHttpActionResult Get(HttpRequestMessage request)
{
    var clientCert = request.GetRequestContext().ClientCertificate;
    if (clientCert != null)
    {
        // inspect client cert
        var subject = clientCert.Subject;
    }

    // further logic
}
```

# Summary

- **Web API supports classic authentication methods**
  - Windows Authentication
  - Basic Authentication
  - Client certificates
- **Katana authentication middleware gives you a unified authentication infrastructure**
  - also allows combining multiple authentication methods