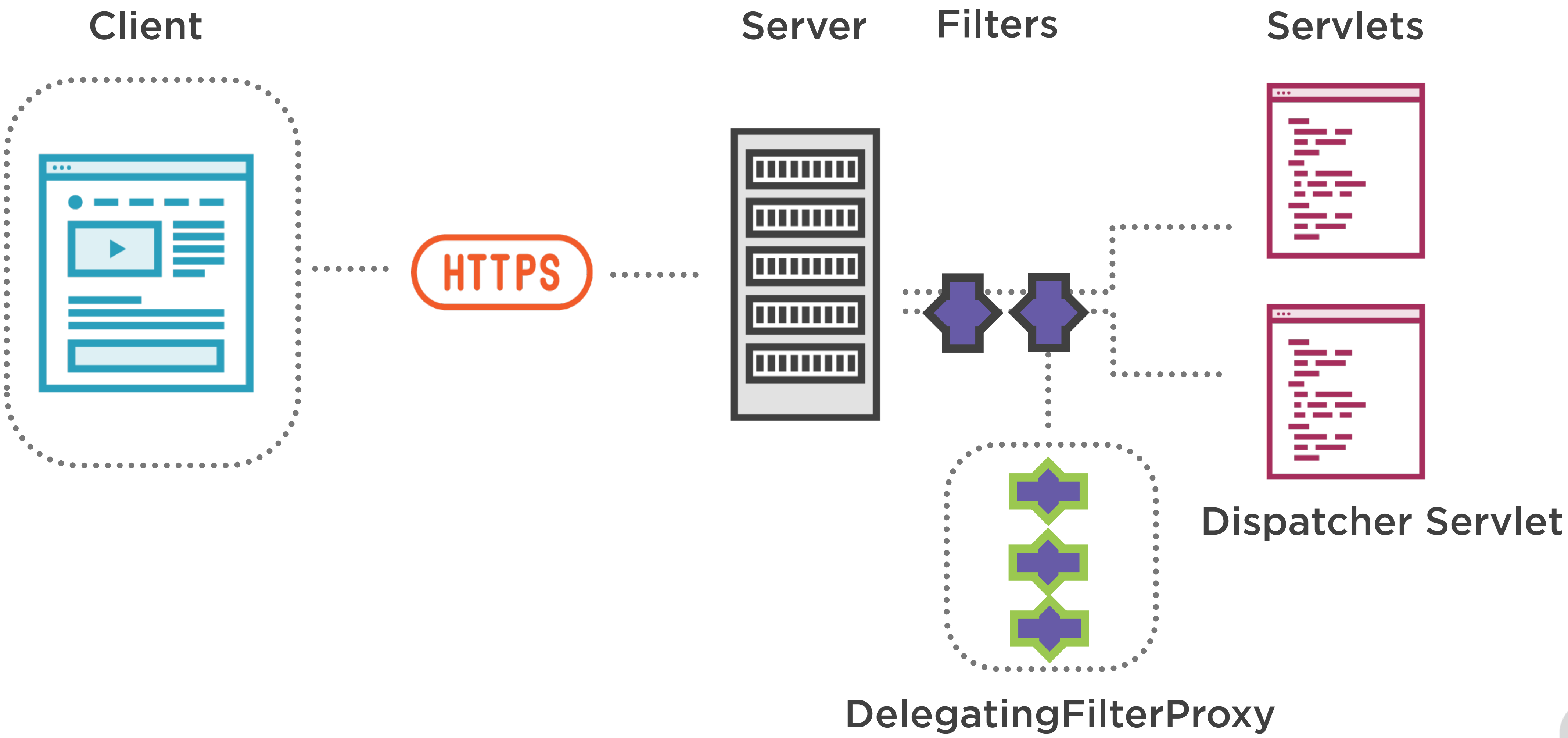
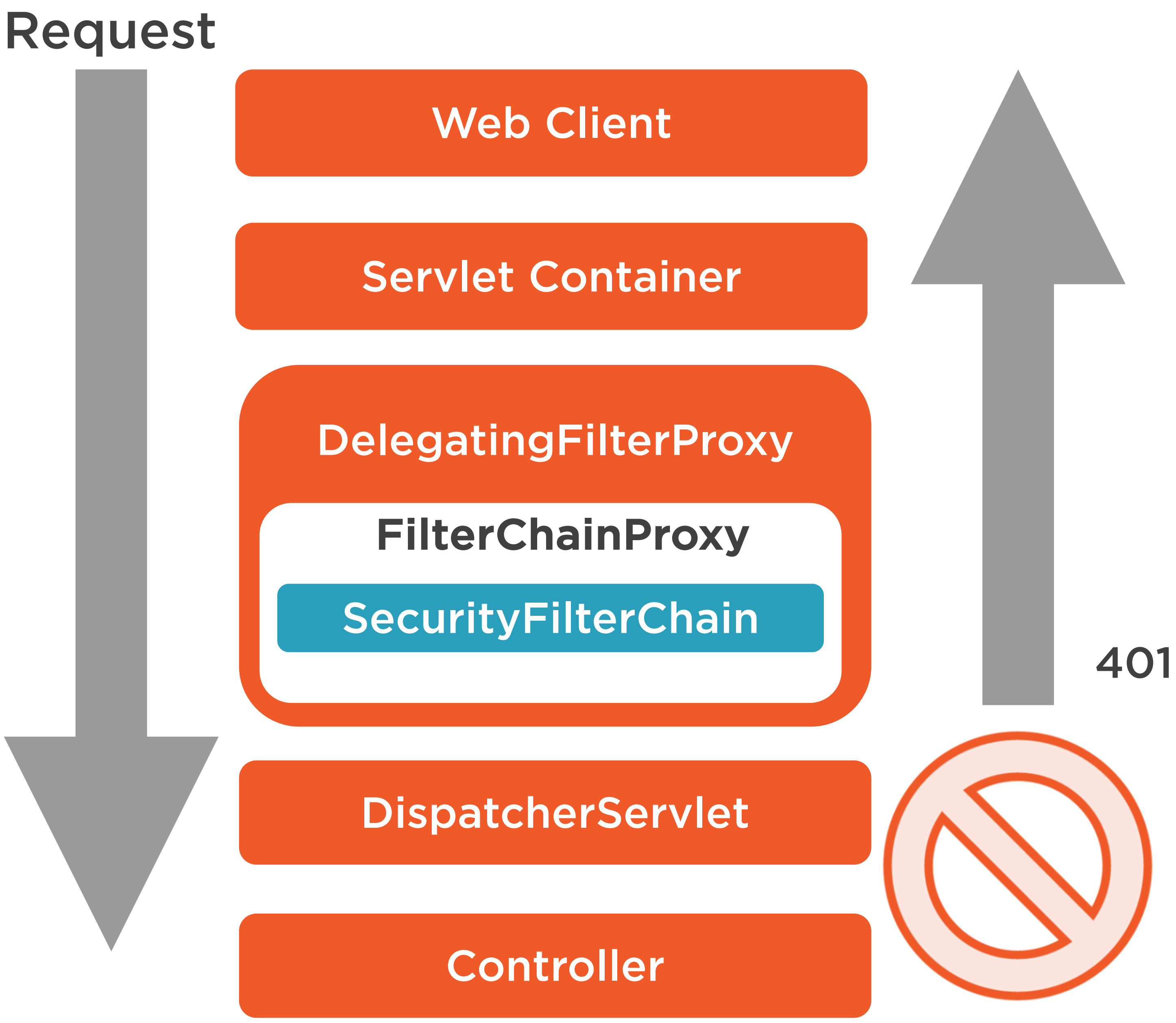
Servlets and Filters





Request

Web Client

Servlet Container

DelegatingFilterProxy

FilterChainProxy

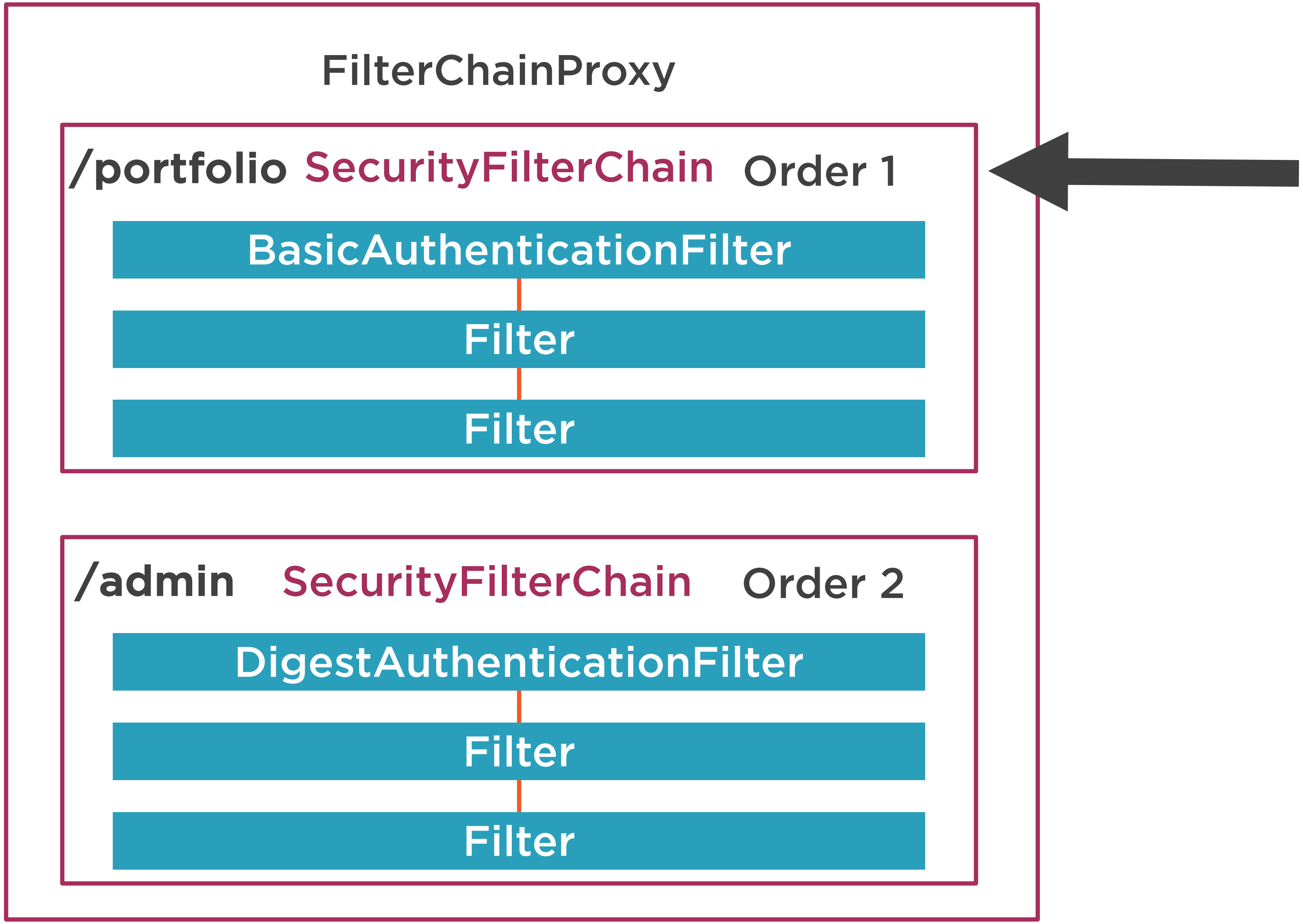
SecurityFilterChain

DispatcherServlet

Controller

401

https://acme/admin



FilterChainProxy

/portfolio SecurityFilterChain Order 1

BasicAuthenticationFilter

Filter

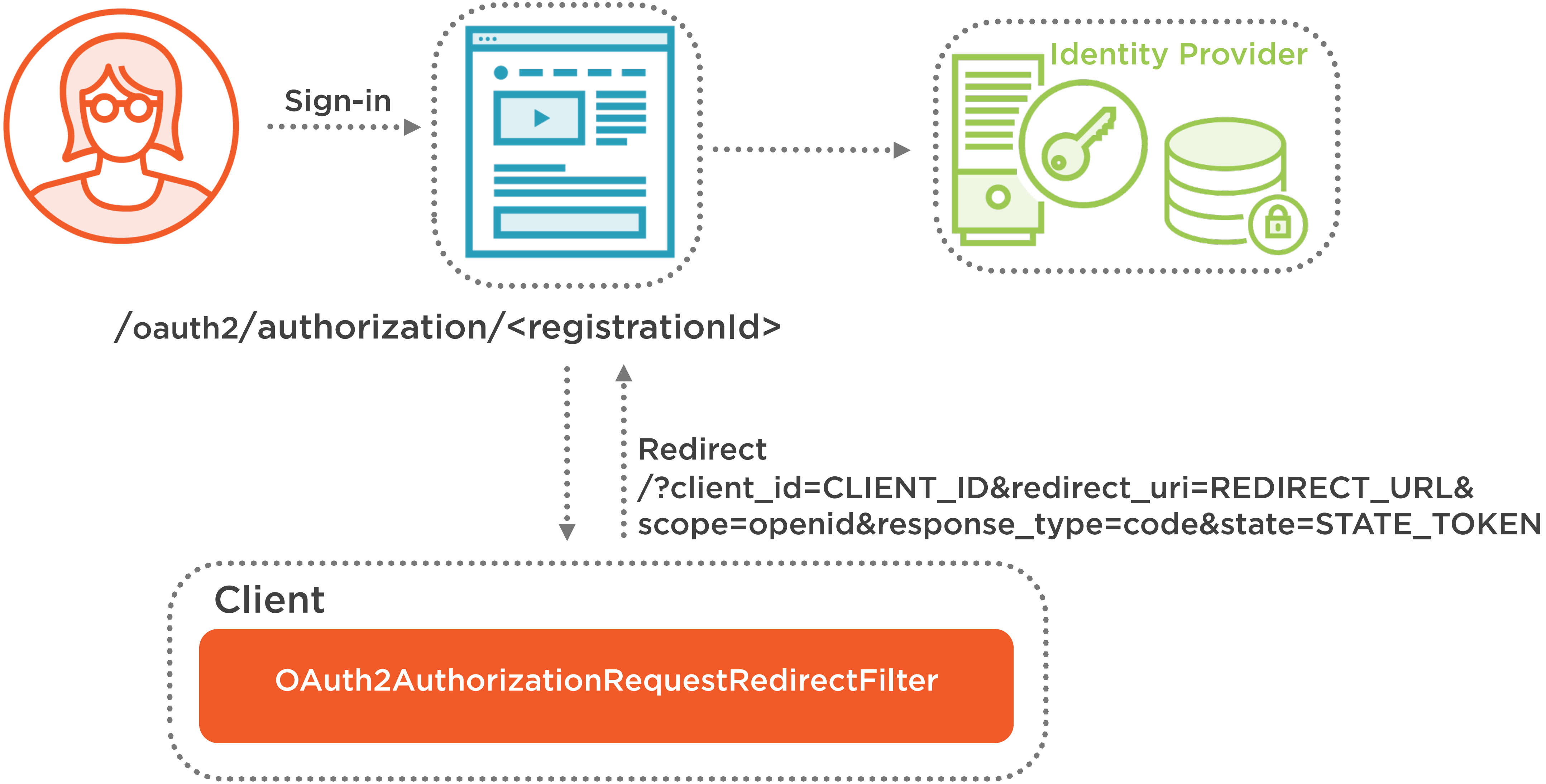
Filter

/admin SecurityFilterChain Order 2

DigestAuthenticationFilter

Filter

Filter



Sign-in

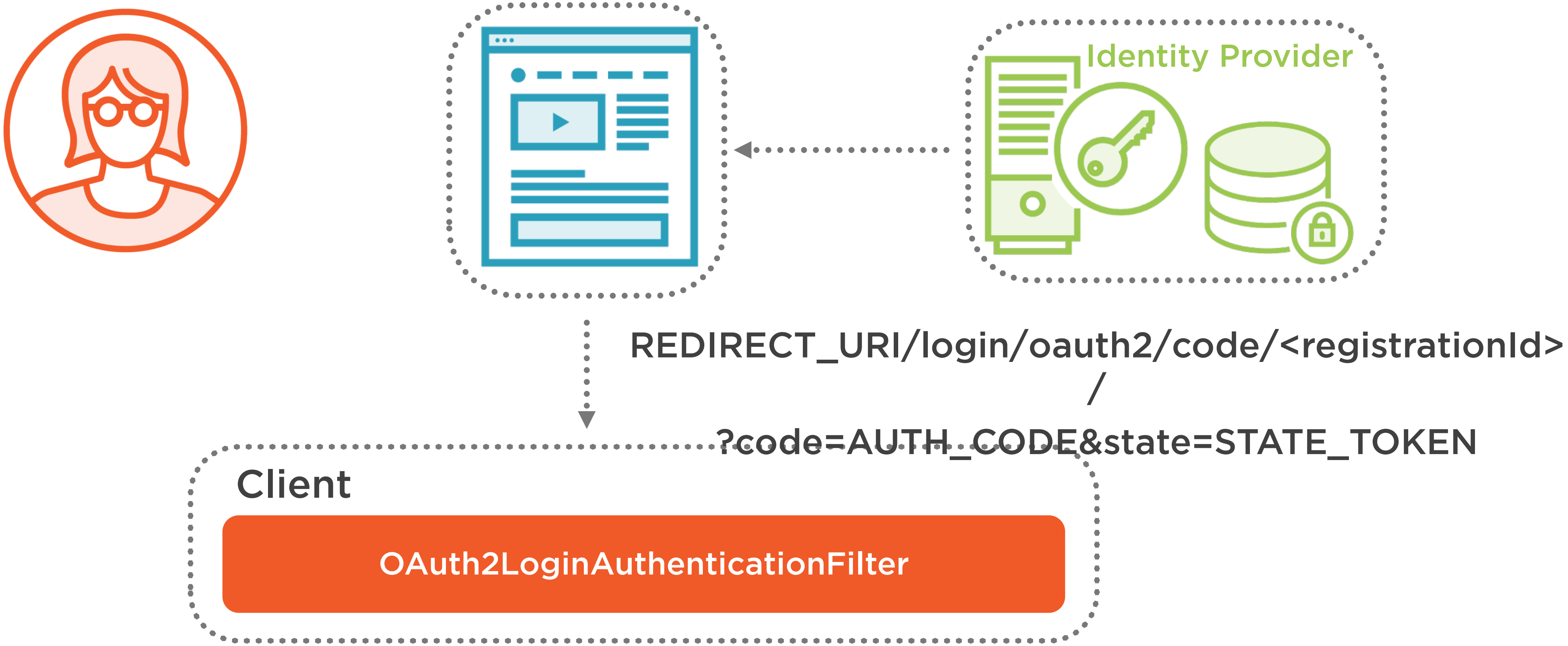
Identity Provider

/oauth2/authorization/<registrationId>

Redirect
/?client_id=CLIENT_ID&redirect_uri=REDIRECT_URL&
scope=openid&response_type=code&state=STATE_TOKEN

Client

OAuth2AuthorizationRequestRedirectFilter

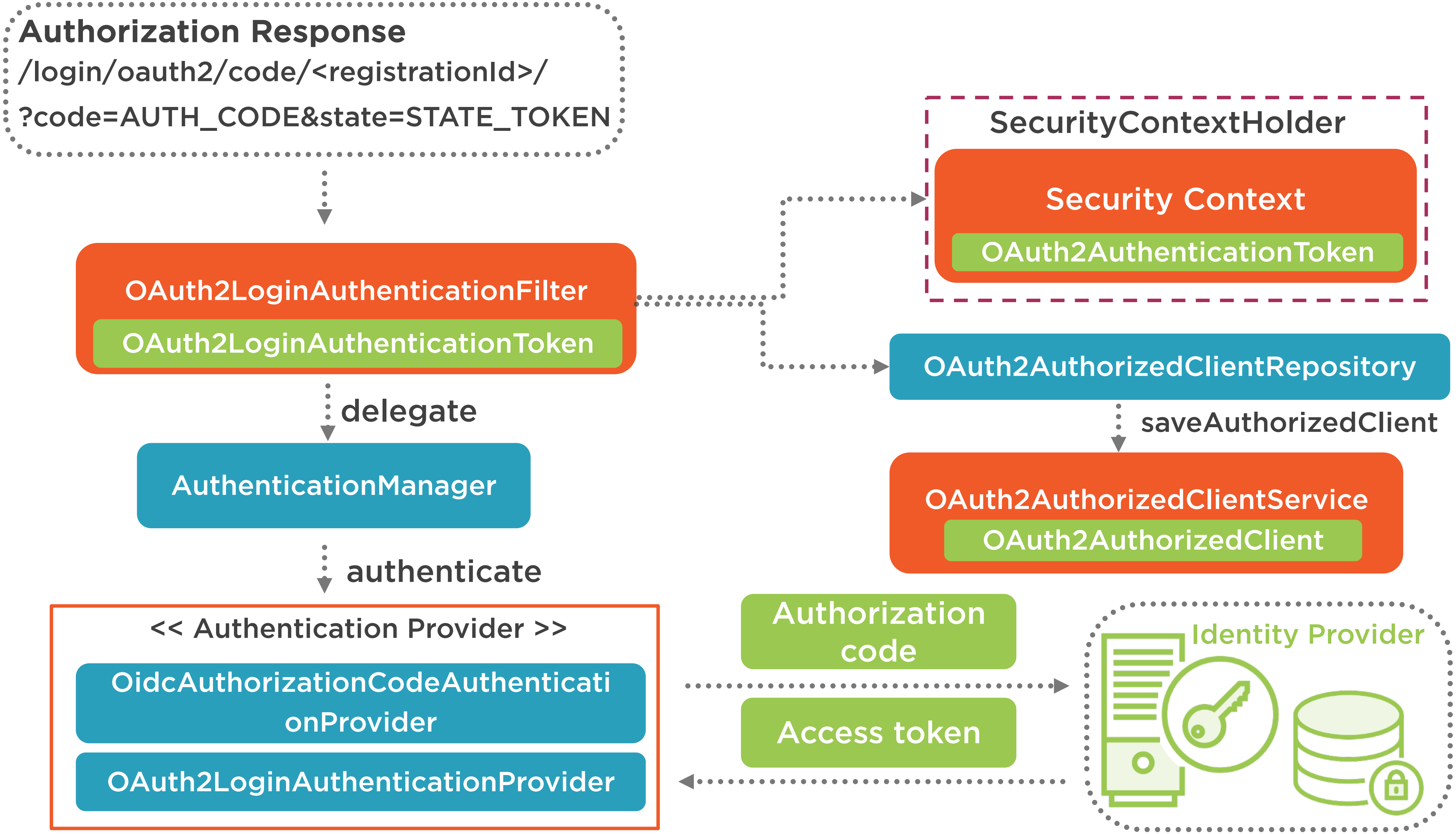


Identity Provider

REDIRECT_URI/login/oauth2/code/<registrationId>
/
?code=AUTH_CODE&state=STATE_TOKEN

Client

OAuth2LoginAuthenticationFilter



Authorization Response

/login/oauth2/code/<registrationId>/

?code=AUTH_CODE&state=STATE_TOKEN

OAuth2LoginAuthenticationFilter

OAuth2LoginAuthenticationToken

delegate

AuthenticationManager

authenticate

<< Authentication Provider >>

OidcAuthorizationCodeAuthenticationProvider

OAuth2LoginAuthenticationProvider

SecurityContextHolder

Security Context

OAuth2AuthenticationToken

OAuth2AuthorizedClientRepository

saveAuthorizedClient

OAuth2AuthorizedClientService

OAuth2AuthorizedClient

Authorization code

Access token

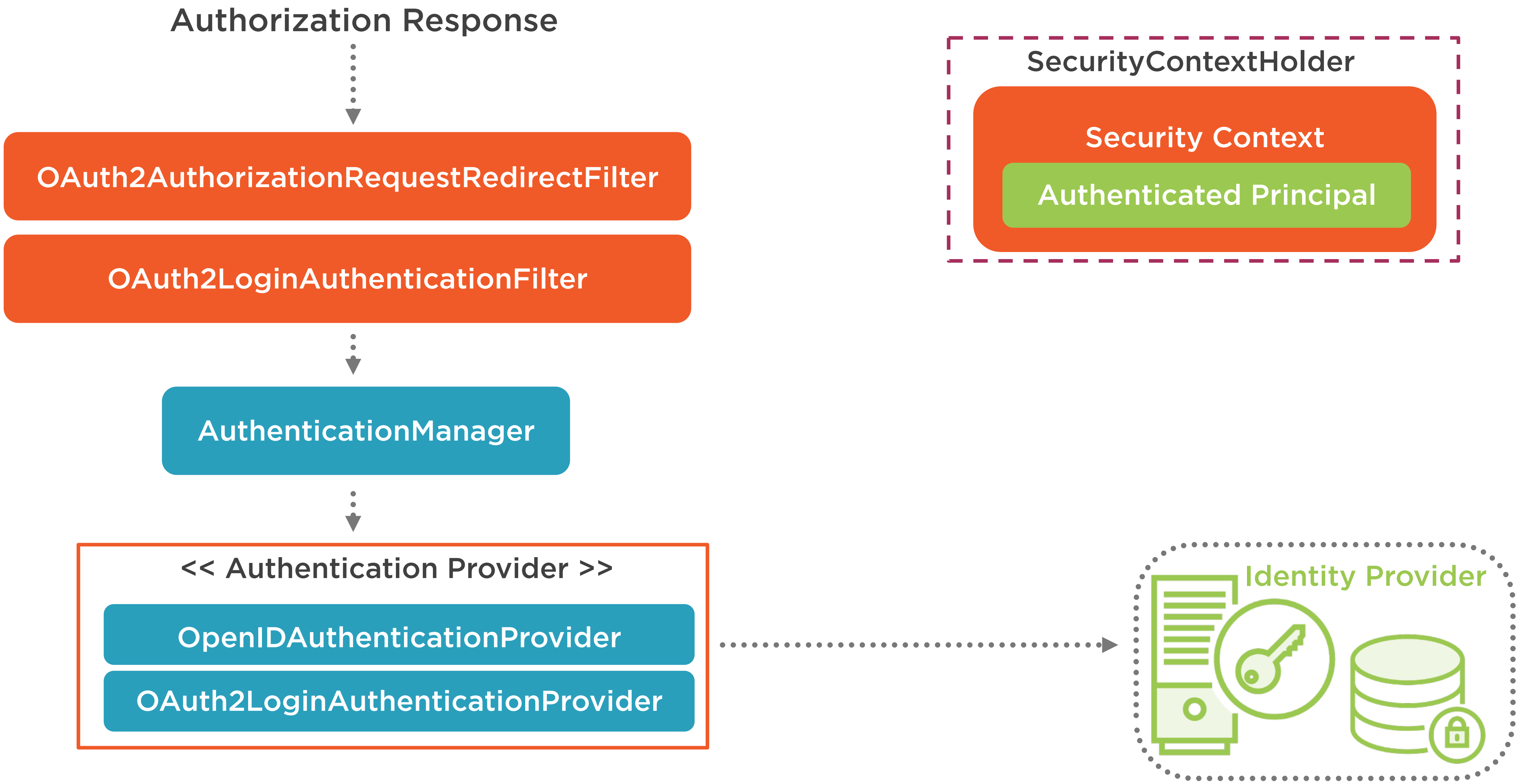
Identity Provider

Interface OAuth2AuthorizedClientService

```
<T extends OAuth2AuthorizedClient> T loadAuthorizedClient(String
clientRegistrationId, String principalName);

void saveAuthorizedClient(OAuth2AuthorizedClient authorizedClient,
Authentication principal);

void removeAuthorizedClient(String clientRegistrationId, String
principalName);
```



Authorization Response

OAuth2AuthorizationRequestRedirectFilter

OAuth2LoginAuthenticationFilter

AuthenticationManager

<< Authentication Provider >>

OpenIDAuthenticationProvider

OAuth2LoginAuthenticationProvider

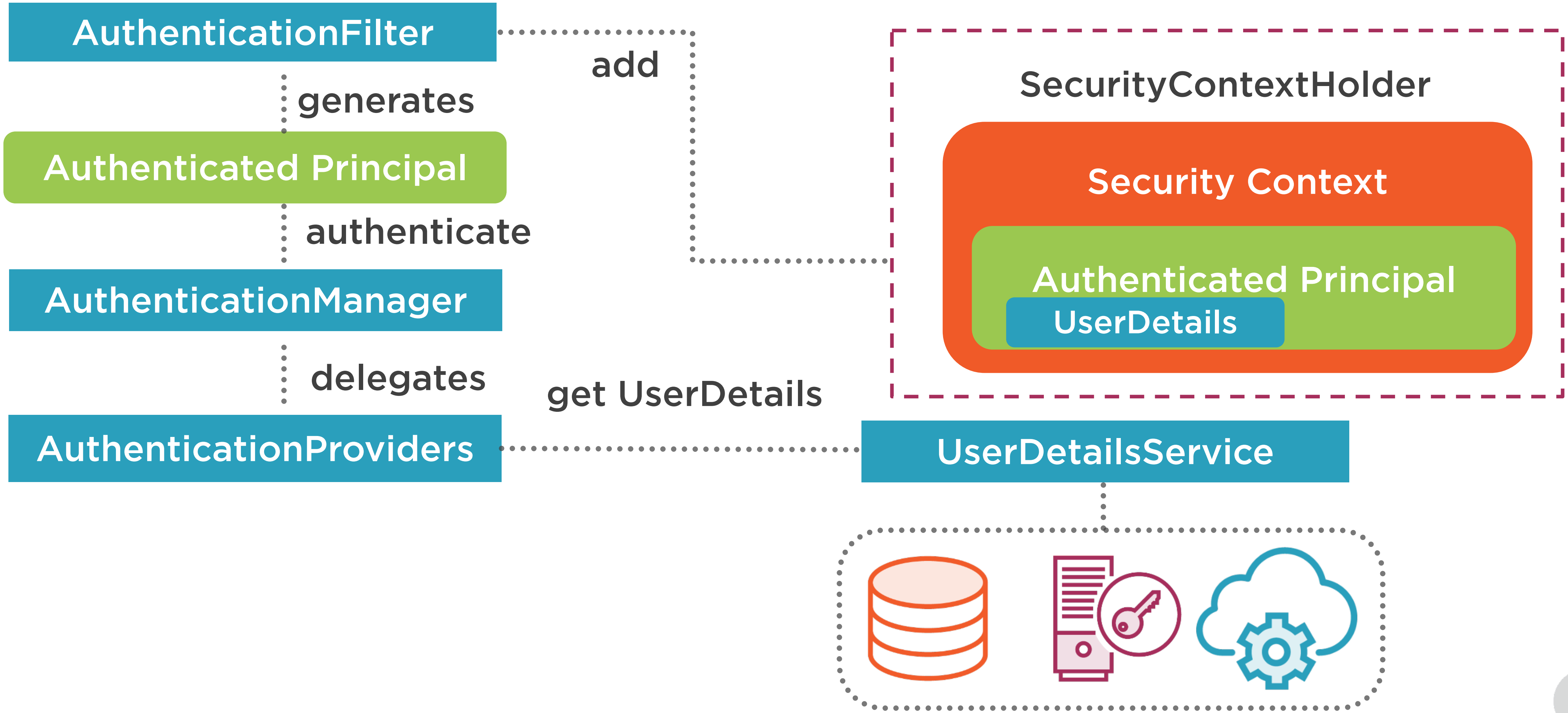
Identity Provider

SecurityContextHolder

Security Context

Authenticated Principal

Form Authentication



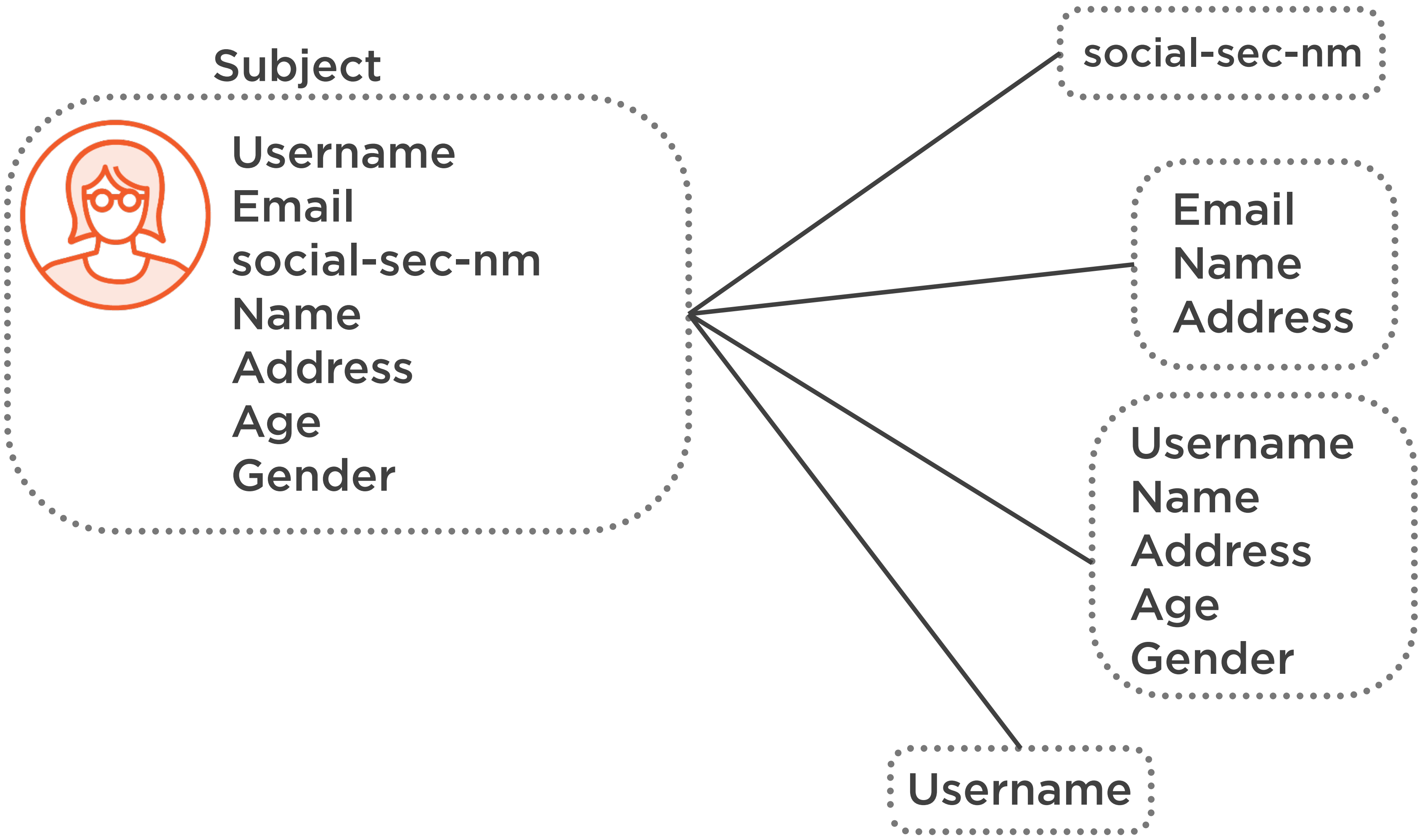
Subject

Human

Server

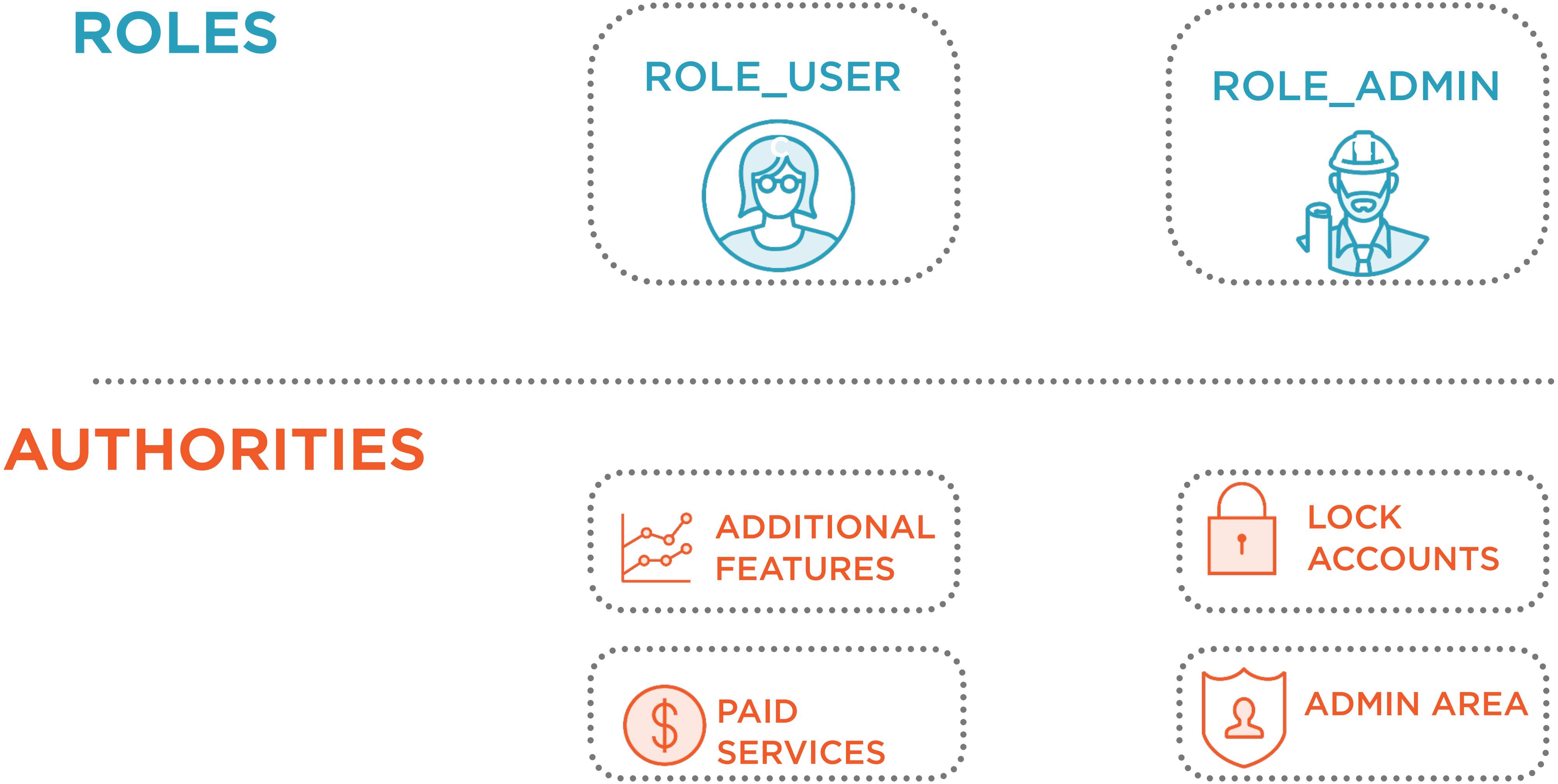
Service

Principal


Subject
Username
Email
social-sec-nm
Name
Address
Age
Gender
social-sec-nm
Email
Name
Address
Username
Name
Address
Age
Gender
Username

- Roles are prefixed with: ROLE_
- Scopes are prefixed with: SCOPE_
- Authorities are not prefixed

Roles and Authorities



ROLES

ROLE_USER

ROLE_ADMIN

AUTHORITIES

ADDITIONAL FEATURES

LOCK ACCOUNTS

PAID SERVICES

ADMIN AREA

getPrincipal()

Form Login

UserDetails

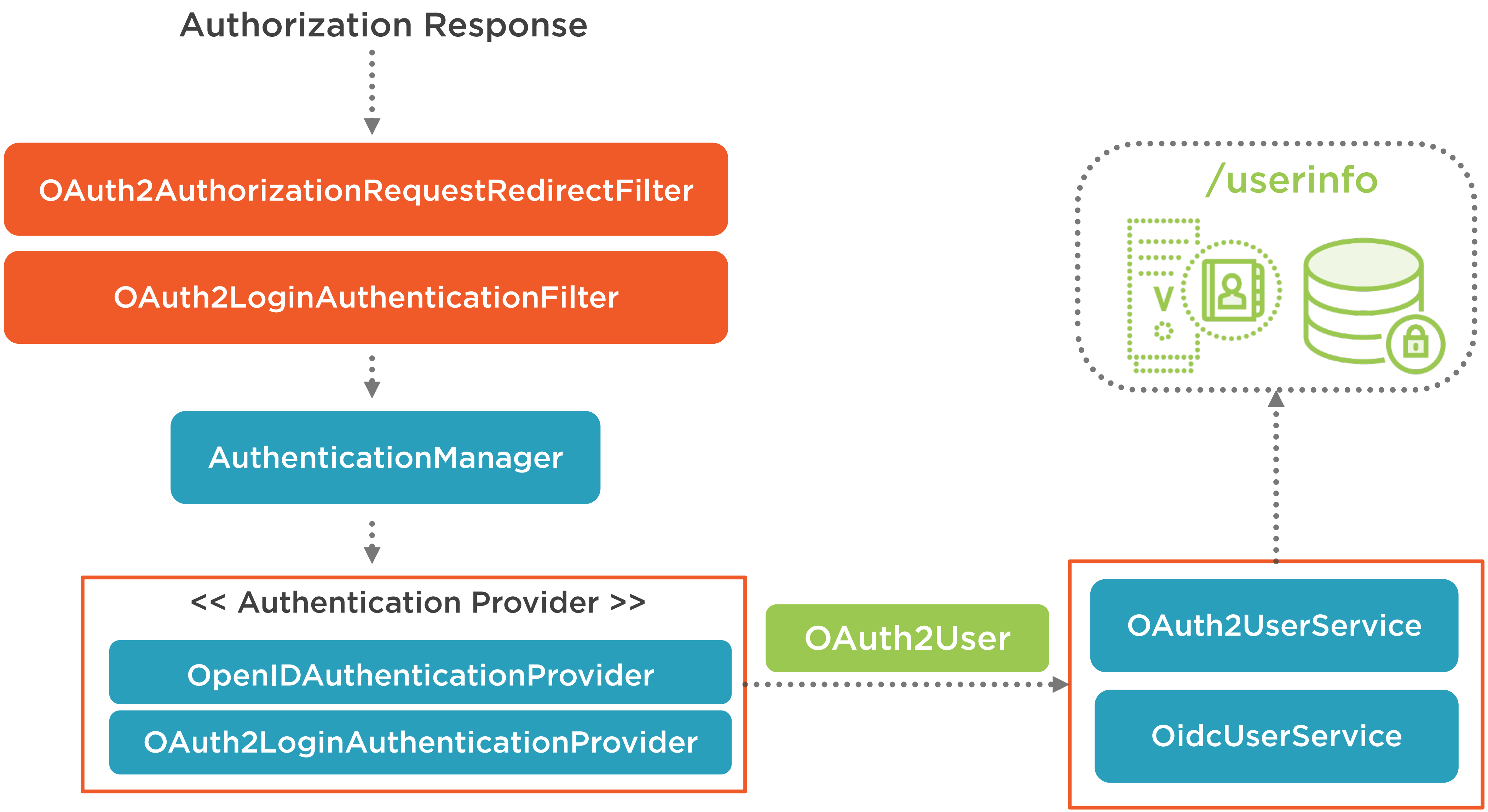
Oauth2

DefaultOAuth2User

OIDC

DefaultOidcUser

CustomUserTypesOAuth2UserService



Authorization Response

OAuth2AuthorizationRequestRedirectFilter

OAuth2LoginAuthenticationFilter

AuthenticationManager

<< Authentication Provider >>

OpenIDAuthenticationProvider

OAuth2LoginAuthenticationProvider

OAuth2User

OAuth2UserService

OidcUserService

/userinfo

Summary

Current state:

- Google and Facebook sign-in
- Simplified user registration

Current vulnerabilities:

- Handles user passwords
- Maintains secrets
- Stores user’s sensitive data

Future improvements:

- Build an Authorization Server