Authentication Outline

- AuthenticationManager
- ProviderManager – I
- ProviderManager – II
- AuthenticationManagerBuilder
- Application that configures the **global** (parent) AuthenticationManager
- Application that configures the **local** AuthenticationManager

AuthenticationManager

<<interface>>

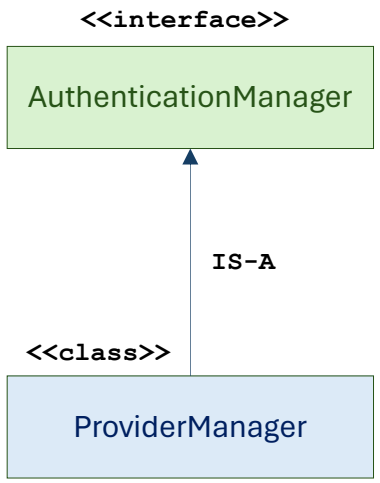
AuthenticationManager

```
public interface AuthenticationManager {
    Authentication authenticate(Authentication authentication)
    throws AuthenticationException;
}
```

- Returns an Authentication
  - (normally with authenticated=true) if it can verify that the input represents a valid principal.
- Throw an AuthenticationException if it believes that the input represents an invalid principal.
- Return null if it cannot decide.

- Spring Boot provides a **default global** AuthenticationManager (with only one user) unless you pre-empt it by providing your own bean of type AuthenticationManager.
- If you do any configuration that builds an AuthenticationManager, you can often ***do it locally to the resources that you are protecting*** and not worry about the **global default**.

ProviderManager - I

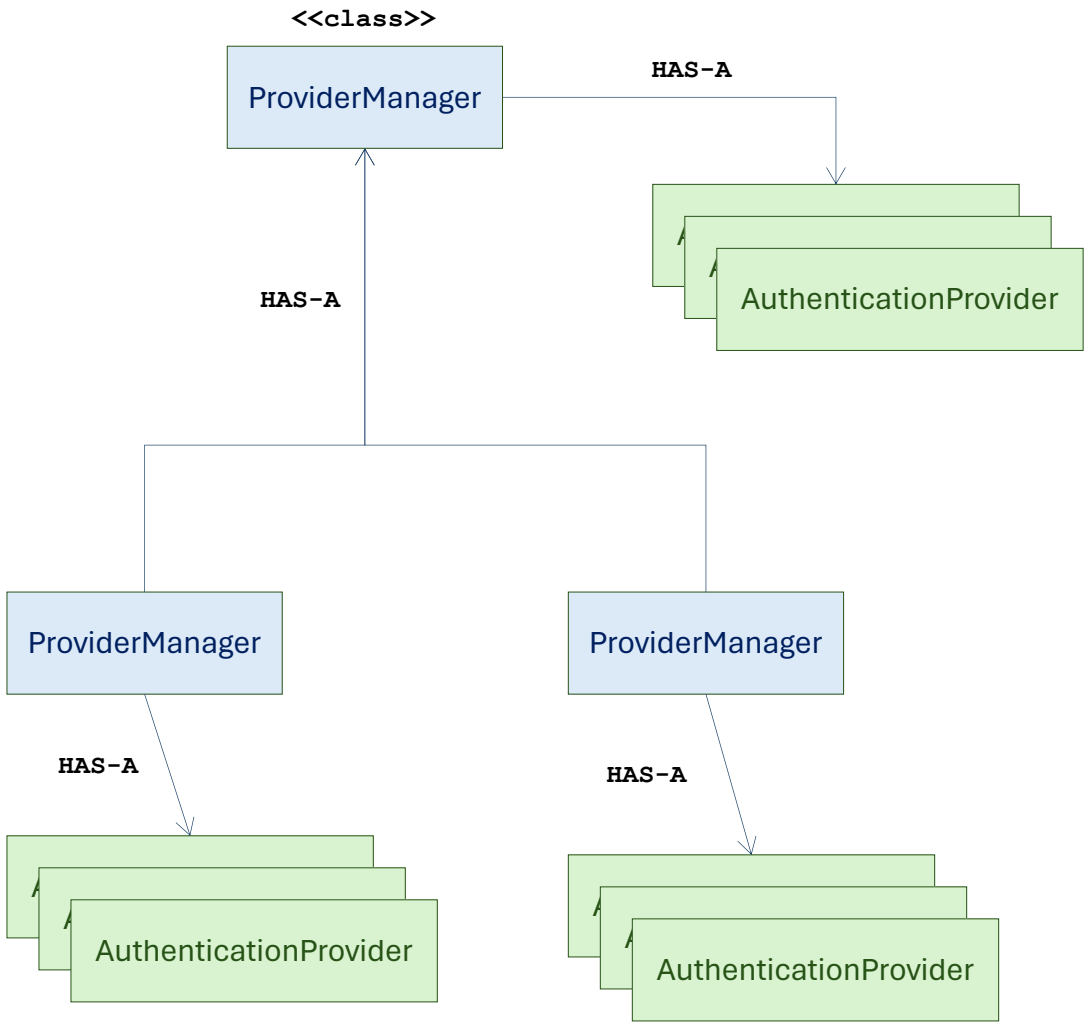


- delegates to a chain of AuthenticationProvider instances.
  - can support multiple different authentication mechanisms in the same application by delegating.
  - If it does not recognize a particular Authentication instance type, it is skipped.

- has an extra method to allow the caller to query whether it supports a given Authentication type.
- The Class<?> argument in the supports() method is really Class<? extends Authentication>.

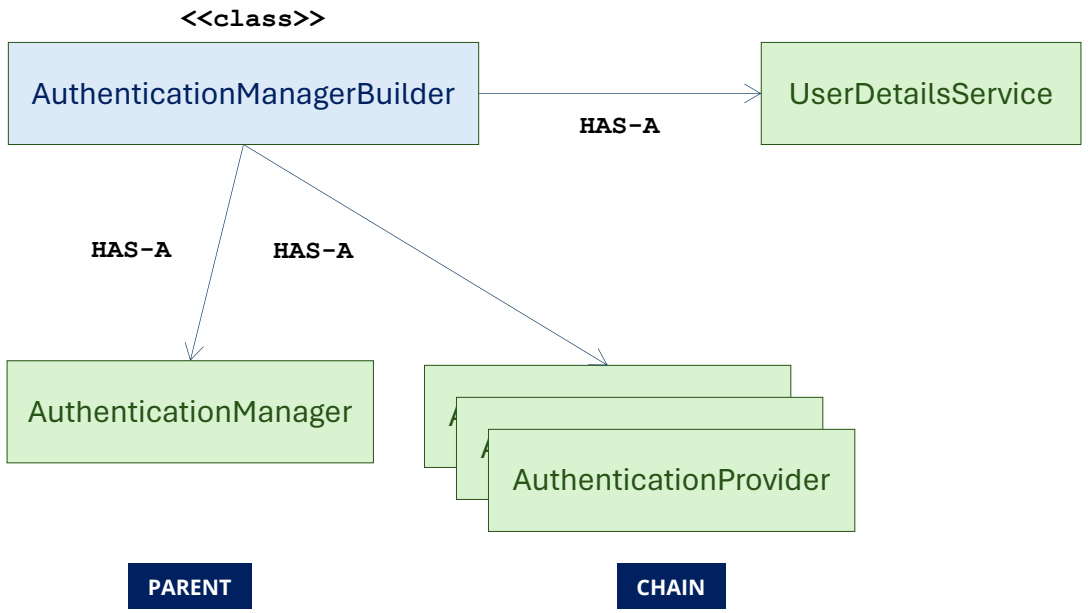
```
public interface AuthenticationProvider {
    Authentication authenticate(Authentication authentication) throws AuthenticationException;
    boolean supports(Class<?> authentication);
}
```

ProviderManager – II



- delegates to a chain of AuthenticationProvider instances.
  - can support multiple different authentication mechanisms in the same application by delegating.
  - If it does not recognize a particular Authentication instance type, it is skipped.
- has an optional parent, which it can consult if all providers return null.
  - If the parent is not available, a null Authentication results in an AuthenticationException.
- Sometimes, an application has logical groups of protected resources
  - i.e., all web resources that match a path pattern, such as /api/**.
- Each group can have its own dedicated AuthenticationManager.
- Often, each of those is a ProviderManager, and they share a parent.
- The parent is then a kind of **global resource** acting as a fallback for all providers.

AuthenticationManagerBuilder



- Spring Security provides some configuration helpers to quickly get common authentication manager features set up in your application.
- AuthenticationManagerBuilder
  - most used helper
  - great for setting up in-memory, JDBC, or LDAP user details or for adding a custom UserDetailsService.

Application that configures the **global** (parent) AuthenticationManager

```
@Configuration
public class ApplicationSecurity extends WebSecurityConfigurerAdapter {
    ... // web stuff here

    @Autowired
    public void initialize(AuthenticationManagerBuilder builder, DataSource dataSource) {
        builder.jdbcAuthentication()
               .dataSource(dataSource)
               .withUser("dave")
               .password("secret")
               .roles("USER");
     }
}
```

- AuthenticationManagerBuilder is injected (or auto-wired) into a method in a @Bean.
- This method body builds the **global** (parent) AuthenticationManager.

Application that configures the **local** AuthenticationManager

```
@Configuration
public class ApplicationSecurity extends WebSecurityConfigurerAdapter {
    @Autowired
    DataSource dataSource;

    ... // web stuff here

    @Override
    public void configure(AuthenticationManagerBuilder builder) {
        builder.jdbcAuthentication()
               .dataSource(dataSource)
               .withUser("dave")
               .password("secret")
               .roles("USER");
     }
}
```

- We overrode the configure() method in the configurer.
- AuthenticationManagerBuilder builds a **local** AuthenticationManager which would be a **child** of the **global** one.