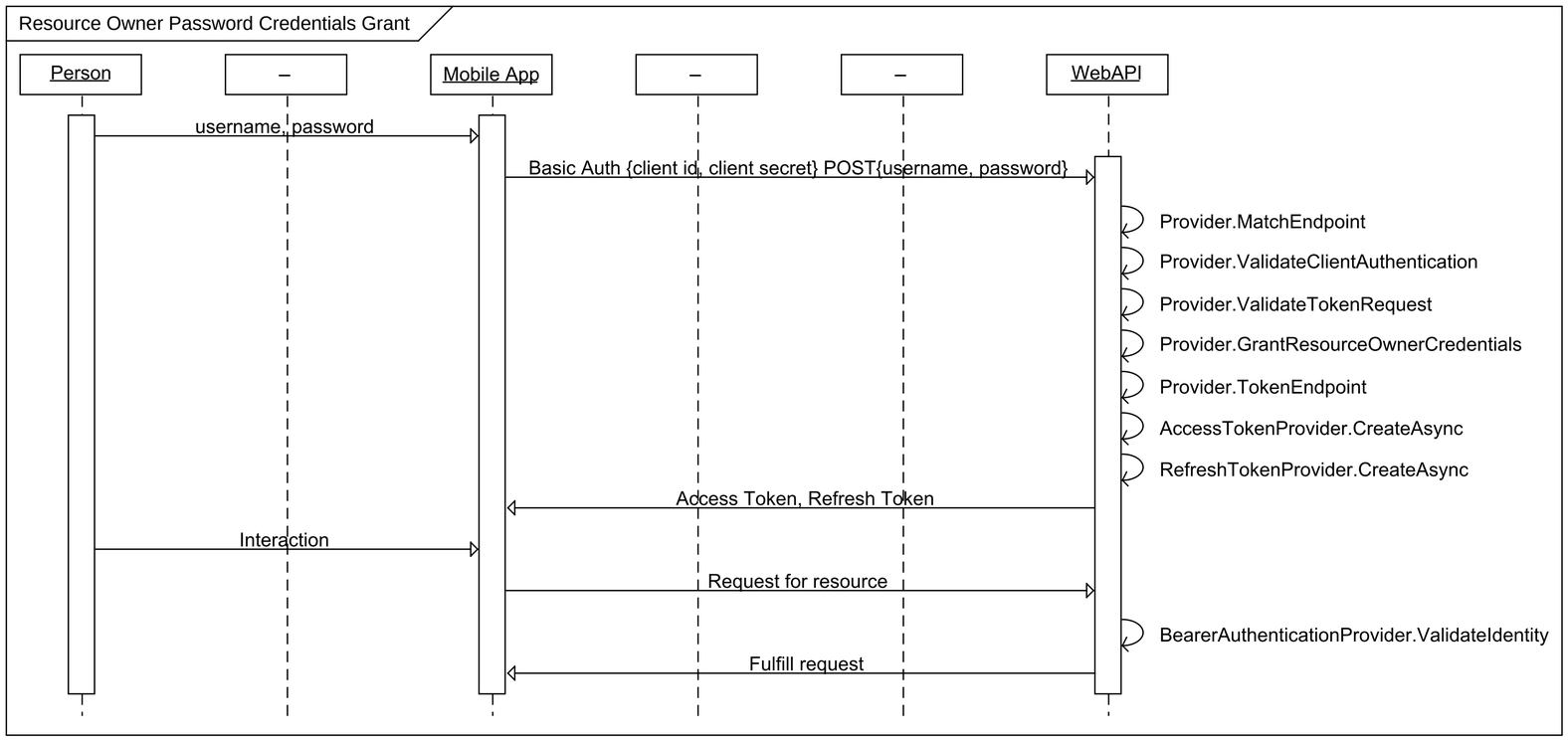
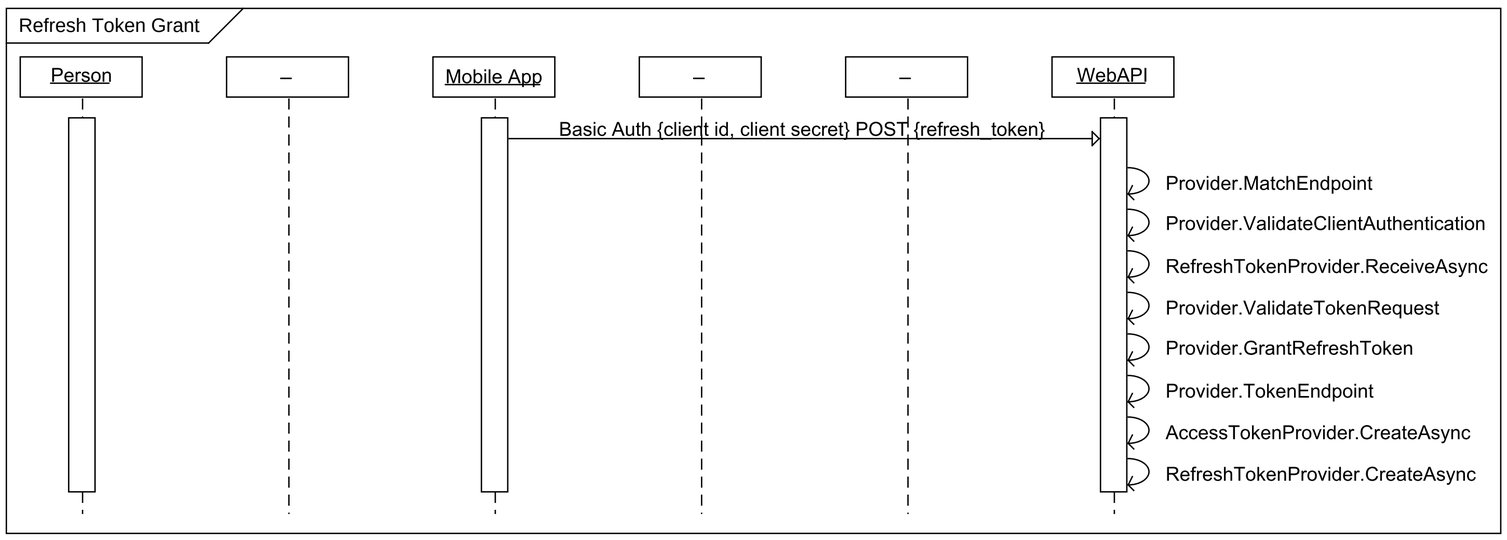
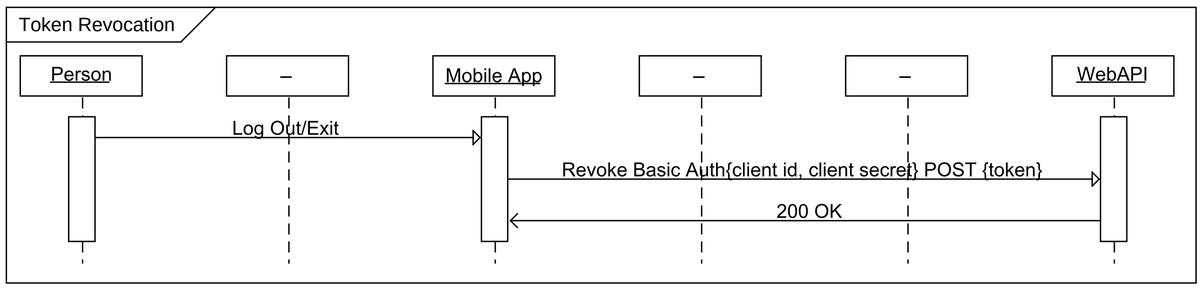


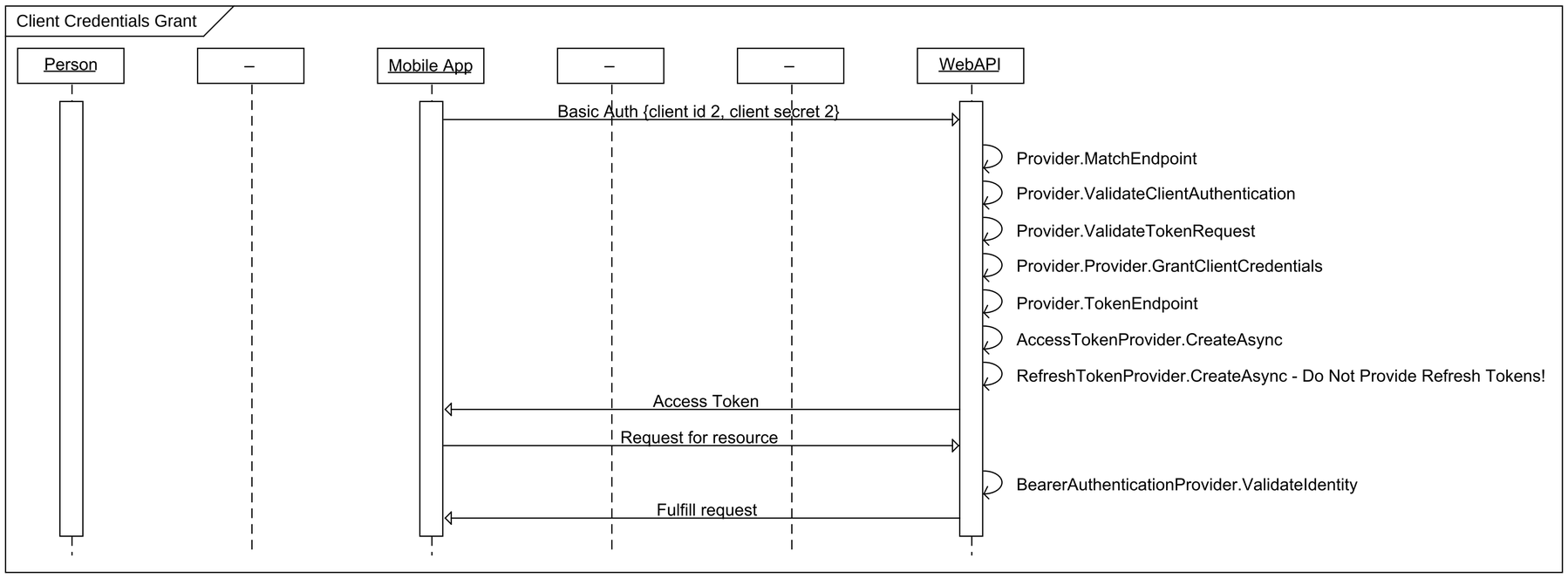
Resource Owner Password Credentials Grant
Provider.MatchEndpoint
Provider.ValidateClientAuthentication 1. Retrieve client id and client secret from basic authentication header 2. Authenticate client 3. context.Validated(clientId);
Provider.ValidateTokenRequest
Provider.GrantResourceOwnerCredentials 1. Ensure client is permitted to use Resource Owner Password Credentials Grant 2. Authenticate the user using username and password 3. Verify that the user's org and client's org match 4. Create a session in the database 5. Add claims/scope 6. context.Validated(...)
Provider.TokenEndpoint 1. Add any additional properties to be communicated to the client
AccessTokenProvider.CreateAsync
RefreshTokenProvider.CreateAsync 1. Generate Guid 2. Add Guid as refresh token to session table of database 3. context.SetToken(refreshToken.ToString())
BearerAuthenticationProvider.ValidateIdentity 1. Validate framework has populated user identity in context.Ticket 2. Check sessions database table to ensure user hasn't logged out or relogged in



Refresh Token Grant
Provider.MatchEndpoint
Provider.ValidateClientAuthentication Same code as before
RefreshTokenProvider.ReceiveAsync 1. Look up Guid in Database, retrieve session 2. Validate that organization in client and session match 3. Verify refresh token has not expired 4. Create Authentication Ticket 5. Set issued and expiration date 6. context.SetTicket(ticket);
Provider.ValidateTokenRequest
Provider.GrantRefreshToken 1. Verify ticket is present and set 2. Verify user org matches client org 3. Create a session in the database 4. context.Validated(); Provider.TokenEndpoint
AccessTokenProvider.CreateAsync
RefreshTokenProvider.CreateAsync Already covered



Token Revocation
OAuthBearerAuthenticationProvider.ValidateIdentity 1. Authenticate the Basic Auth client via client id and client secret 2. Verify client is enabled and active 3. Resolve the token into a user 4. Refresh the user from the database 5. Verify user's org and client's org matches 6. Verify the user's session was created by that client 7. Verify sessions exist at all 8. Delete all active sessions (access token and refresh token)



Client Credentials Grant
Provider.MatchEndpoint
Provider.ValidateClientAuthentication Same code as Resource Owner Password Credentials Grant
Provider.ValidateTokenRequest
Provider.GrantClientCredentials 1. Ensure client is permitted to use Client Credentials Grant 2. Create identity and apply claims/scope 3. Shorten access token expiration 4. context.Validated(...)
Provider.TokenEndpoint Same as Resource Owner Password Credentials Grant
AccessTokenProvider.CreateAsync
RefreshTokenProvider.CreateAsync Do not generate a refresh token for Client Credentials Grant
BearerAuthenticationProvider.ValidateIdentity 1. Validate framework has populated user identity in context.Ticket 2. Check sessions database table to ensure user hasn't logged out or relogged in