Resilient Access provides an intuitive interface for creating policies. Authorities are displayed in a panel on the left. Resilient provides a set of out of the Box authorities that are well suited for additional factors of authentication and authorization. This list currently consists of the following authorities

- **Email Authentication:** Ensures that the user can access the account of the email address that they provide. The Email Authentication Authority sends an email to the specified email address. Inside the email is a code. Upon receiving the email, the user must type the code into the web form.

- **SMS Authentication:** Provides out-of-band authentication. Ensures that the end user is in possession of a mobile phone and can receive an SMS message. After receiving the SMS from the SMS Authentication Authority, the end user is prompted to type the code in the Trust Broker Display screen.

- **Phone Authentication:** Provides out-of-band authentication. Ensures that the end user is in possession of a phone and can answer a specific phone number. After receiving a call from the Phone Authentication Authority, the end user types the code displayed on screen into the phone.**Voice Authentication:** Provides out-of-band authentication. Ensures that the end user is in possession of a phone and their voice matches the voice imprint stored in the Voice Authentication service . The first time, the user receives a phone call prompting to record their voice imprint by speaking out a series of authorization code values. The user then receives a second call to verify their voice imprint by speaking out the authorization code(s) displayed on the screen. With the voice imprint in file, Voice Authentication involves a single phone call and speaking out the authorization code(s) displayed on the screen. The system will prompt for up to 4 authorization codes to determine a voice imprint match.

- **Google Authenticator:** Provides an option to add a popular TOTP (Time-based One Time Password) authentication factor to a policy. This authentication displays a 6 digit number in a smart phone app that changes every 30 seconds. The user is prompted to enter the 6 digit number corresponding to the account they are trying to access in the *Authority Credentials* popup. A pre-requisite is to install the Google Authenticator smart phone app. There are 2 authorities, a provisioning authority *Google Authenticator Provisioning,* that should be part of a registration/provisioning policy. A QR code will be displayed in the *Authority Credentials* popup, the user should put their Google Authenticator app in 'Scan Barcode' mode and scan the QR code to create an account in the app. The 2nd authority is the *Google Authenticator* authentication authority that should be part of an authentication policy. This authority prompts the user to enter the 6 digit code corresponding to the provisioned account the user is attempting to authenticate into.

- **Acceptto Mobile Authenticator:** Provides an option to use a smart phone as an authentication device. There are 2 authorities, a provisioning authority *Acceptto Provisioning*, that should be part of a registration/provisioning policy. This authority provisions a user account in the Acceptto service. The user info required for provisioning is a user's name, verified email address, verified phone number, password and 2 security question and answers. Assuming the provisioning policy executes on the submission of a user registration form, the provisioning policy should include *Email Authentication* and *Phone Authentication* or *SMS Authentication* as preceding factors for email and phone verification. The provisioning authority will ask the security question

and answers if not provided in the user registration data and will prompt the user to download the *Acceptto It'sMe* iOS or Android apps. After installing the Acceptto app, the user should login with their provisioned email address. The 2nd authority is the *Acceptto Authentication* authority, that should be part of an authentication policy. When the authority executes, a push notification is sent to the Acceptto smart phone app requesting to accept the authentication request.

- **CAPTCHA Verification:** Protects against robots. Requires the end user to read a series of distorted characters that current computer programs cannot interpret.

- **LexisNexis Knowledge Based Authentication:** Given an individual's name, address and date of birth, the Lexis Nexis service will prompt for 3 identity questions with 4 options to select from. The sources for these questions are credit reporting agencies, mortgage records, DMV records, utility records and other public record systems. e.g. "Have you lived on any of the following streets" etc. If there are any incorrect answers, there is one additional follow up question. Rate limiting prevents bad actors to perform brute force attacks to break the system.

- **Infutor Identity Verification:** Resilient Access has integrations with the Infutor ID Max API that returns identity verification attributes and demographic data based on some basic input attributes such as name, email, phone number or address. The input attributes are full name and one or more of phone number, address and/or email. The service provides a score/rating of the match between the name and other identity attributes. For identities the verification succeeds, the authority returns various additional attributes that can be used as context for other policy decisions or can be returned as claims or assertions to relying parties. Similar to Neustar verification authorities there are three types of Infutor Identity Verification authorities
    - **Name - Phone Verification** - returns a score of the name to phone number match and other attribute
    - **Name - Email Verification** - returns a score of the name to email match and other attributes
    - **Name - Address Verification** - returns a score of the name to address match and other attributes.

- **FIDO U2F Authenticator:** The Fido Alliance has specified a second factor authentication standard that uses a USB hardware device that can be used for strong cryptographic authentication. An implementation of this standard by Google and Yubico is being used for enhanced security for Google products. The USB device acts as a certificate authority and issues an RSA key pair, storing the private key in the device and the public key is held by a U2F service implemented by an Identity provider. A cryptographic value is generated using the user ID, application ID and tenant ID attributes during the registration process. In the authentication phase the cryptographic value generated at authentication time is compared to the value generated during registration. Google Chrome and Firefox browsers incorporate the plugins to interact with the U2F device.

  Resilient Access has a component that implements the U2F Server API and a client that incorporates U2F javascript libraries and client code to invoke the U2F Server API. This functionality is providesd by two authorities - U2F Registration and U2F Authentication. The U2F

Registration authority is usually part of an account registration policy and the U2F Authentication authority can be used in an MFA authentication policy.

- **FIDO2/WebAuthn Authenticator:** The FIDO Alliance and the W3C have produced the FIDO2/WebAuthn standard. It specifies the interaction between three parties:

    - *FIDO2 Authenticators* - protocols such as USB, BLE and NFC are used to connect authenticators to user agents; U2F authenticators are supported
    - *WebAuthn User Agents* - builtin browser and OS agents that bridge RP clients with FIDO2 authenticators
    - *RP clients* - JS browser clients and client side programs that act on behalf of RPs to register and authenticate users

  The objective of this standard is to support user registration and authentication with strong public key credentials using a broad range of authenticators. This is important for several reasons:

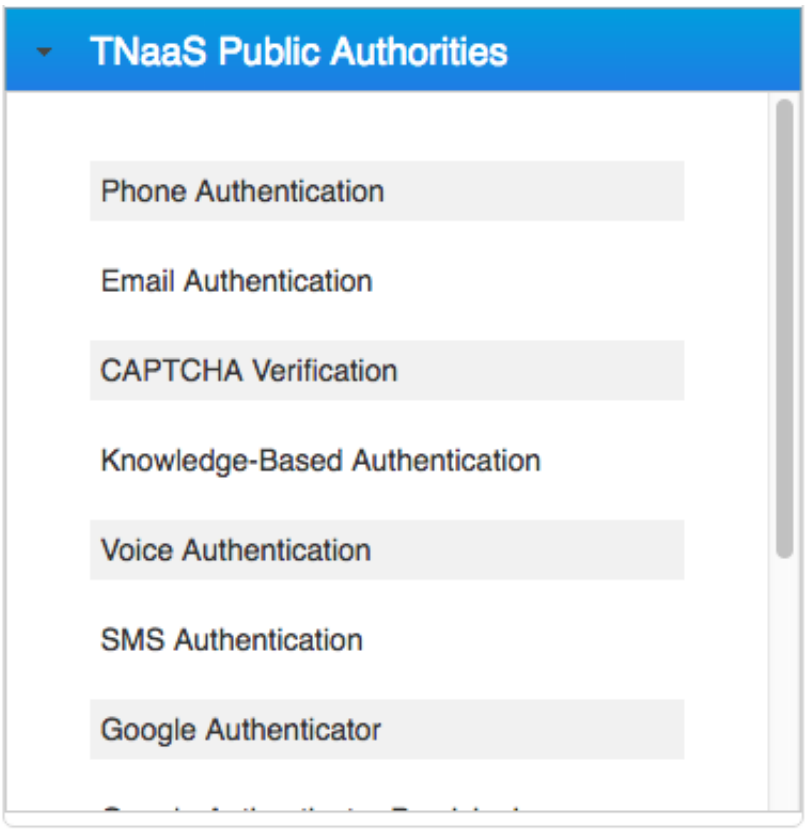
    - Each user registration with an RP is done with a unique public key credential. The credential's private key never leaves the authenticator's secure enclave - it cannot be fished from the user and it cannot be stolen from an RP.
    - Google and Apple(rumored) are planning to add FIDO2 authenticators to Android and iOS. This will provide a ubiquitous, secure, easy-to-use authenticators on consumer phones.
    - Mozilla and Google have added WebAuthn user agents to Firefox and Chrome. Microsoft just announced the availability of one for Edge and Apple is rumored to be adding one to Safari.

  The result has the potential to practically eliminate the password as a consumer credential. In addition, mobile authenticators will also eliminate requiring the user to enter their user ID - the authenticators will automatically provide this to RPs.

  Resilient Access has a component that implements the WebAuthn RP function and is in the process of adding a WebAuthn RP client. Together these will greatly simplify the work required to add RP support for FIDO2/WebAuthn. This functionality is provided by two authorities - WebAuthn Registration and WebAuthn Authentication. An RP can include the WebAuthn Registration authority as an element of their account registration policy; and, can include the WebAuthn Authentication authority as an element of their authentication policy.

Administrator can also create their own authorities for identity and access verification functions that perform the verification function against their back-end resources as described in Create Authorities section. These authorities will appear in a section under the organization domain in the authorities selection UI.

The screenshot below shows the authority selection UI:

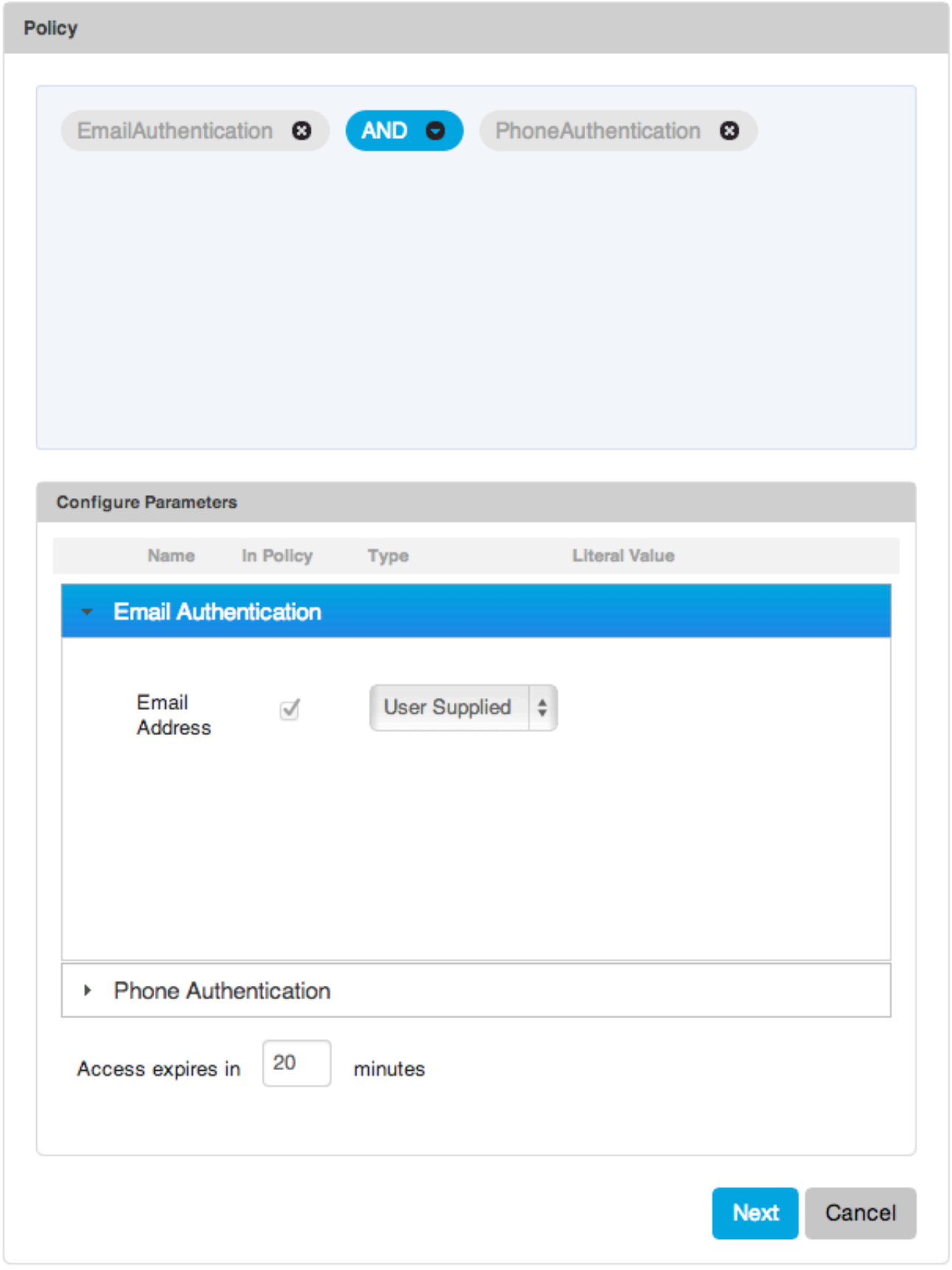


The right side consists of the policy expression panel and **Configure Parameters** section. The policy expression is created by selecting an authority from the **Authorities** control and dragging it to pale blue policy expression panel, the panel is highlighted to indicate it is a drop target. When there are more than one authorities in the policy expression panel, by default it uses AND as the join operator. This can be changed to one of the other three operators (OR, Ordered OR, Ordered AND). When there is a mix of different operators the precedence rules are:

1. Ordered AND
2. AND
3. Ordered OR
4. OR

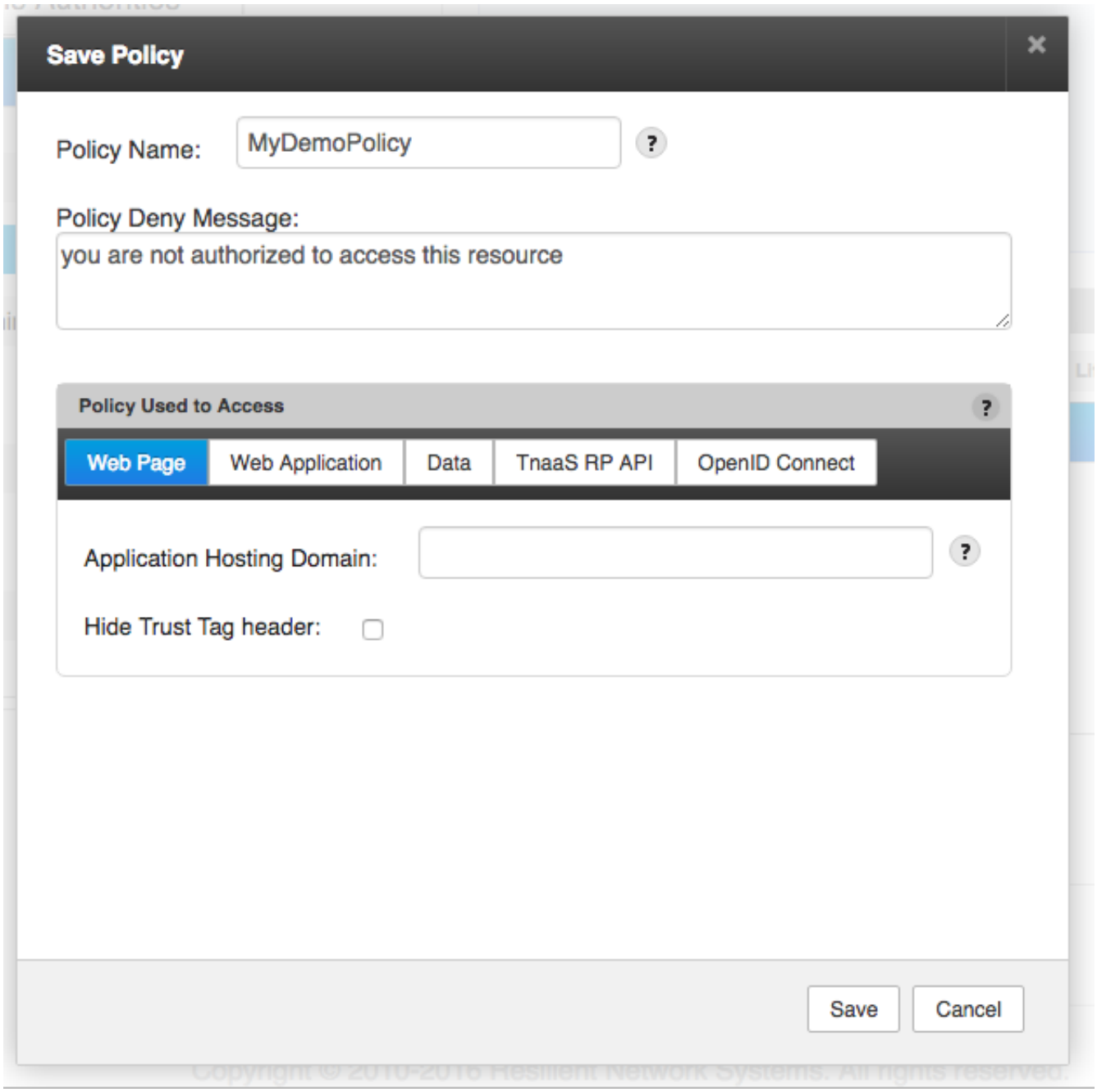
Since policy evaluation is a network operation, by default the AND and OR operators will evaluate the child operand with least complexity first (smallest sub-tree). Ordered OR and Ordered AND provides the semantics to override this and evaluate the child operands of the operator in strict left to right order.

The right hand side of the policy expression builder is show below.


Policy
EmailAuthentication
AND
PhoneAuthentication
Configure Parameters
Name
In Policy
Type
Literal Value
Email Authentication
Email Address
User Supplied
Phone Authentication
Access expires in
20
minutes
Next
Cancel

Authorities have parameters, the configuration of the authority can specify, certain parameters are required to be passed into the policy expression before the policy can be evaluated. For these authority parameter one should either select **User Supplied** or **Literal**. For **User Supplied** before policy evaluation begins the Trust Tag or Data Proxy policy evaluation UI will display prompts for these authority parameters. In the policy configuration one may optionally choose to pass any of the other authority parameters in the policy before evaluation. For the authority parameters that are not supplied into the policy, the policy evaluation run-time will dynamically request for these parameters through a Trust Broker Display screen. It is best practices for privacy enhancing policy evaluation to not pass parameters that contain sensitive information, like passwords before policy evaluation begins to minimize the transmission of sensitive data through the network.

Once the policy expression is built and on clicking the **Next** button, a popup is displayed for providing policy configuration parameters such as the Name and an optional custom deny message etc. The popup for policy configuration is shown below. Please refer to Web Page, Web Application, Data Proxy, TNaaS RP API and OpenID Connect for more details on the configurations of each of the *Policy Used to Access* options.


Save Policy
Policy Name:
MyDemoPolicy
Policy Deny Message:
you are not authorized to access this resource
Policy Used to Access
Web Page
Web Application
Data
TnaaS RP API
OpenID Connect
Application Hosting Domain:
Hide Trust Tag header:
Save
Cancel