Qpid Dispatch Router (QDR) Access Policy

Table of Contents

DISCLAIMER

This work is prelimiary and subject to change. In particular the schema will change after the 0.6 release.

Introduction

Qpid Dispatch Router (QDR) access policy (Policy) is an optional authorization mechanism enforcing connection limits and resource access control.

Definitions

Application

Policy settings are scoped to a QDR ***application***. An application is typically the name of the host to which the client AMQP connection is directed. For example, if a client application opens an AMQP connection URL:

```
amqp://bigbroker.example.com:5672/favorite_subject
```

then a typical AMQP client should embed the host name ***bigbroker.example.com*** in the AMQP Open frame's *hostname* field. QDR calls *bigbroker.example.com* the name of the application. QDR will apply a Policy named *bigbroker.example.com* to the connection and QDR will forward traffic on this connection to a service with that name.

Hostname in this instance is very similar to the HTTP/1.1 Host request-header field. From the AMQP 1.0 Open specification:

> hostname
>
> The dns name of the host (either fully qualified or relative) to which the sending peer is connecting. It is not mandatory to provide the hostname. If no hostname is provided the receiving peer should select a default based on its own configuration. This field can be used by AMQP proxies to determine the correct back-end service to connect the client to.

The AMQP Open hostname is mandatory for traffic to be forwarded properly through a QDR that has Policy in effect. In some cases a client may not populate the Open hostname or may force the Open hostname to an arbitrary value. QDR Policy will accept a blank Open hostname in normal course. QDR Policy will optionally assign undefined Open hostname connections to a fallback policy.

Policy Features

Global Policy

Total Connection Limits

Policy enforces a total connection limit.

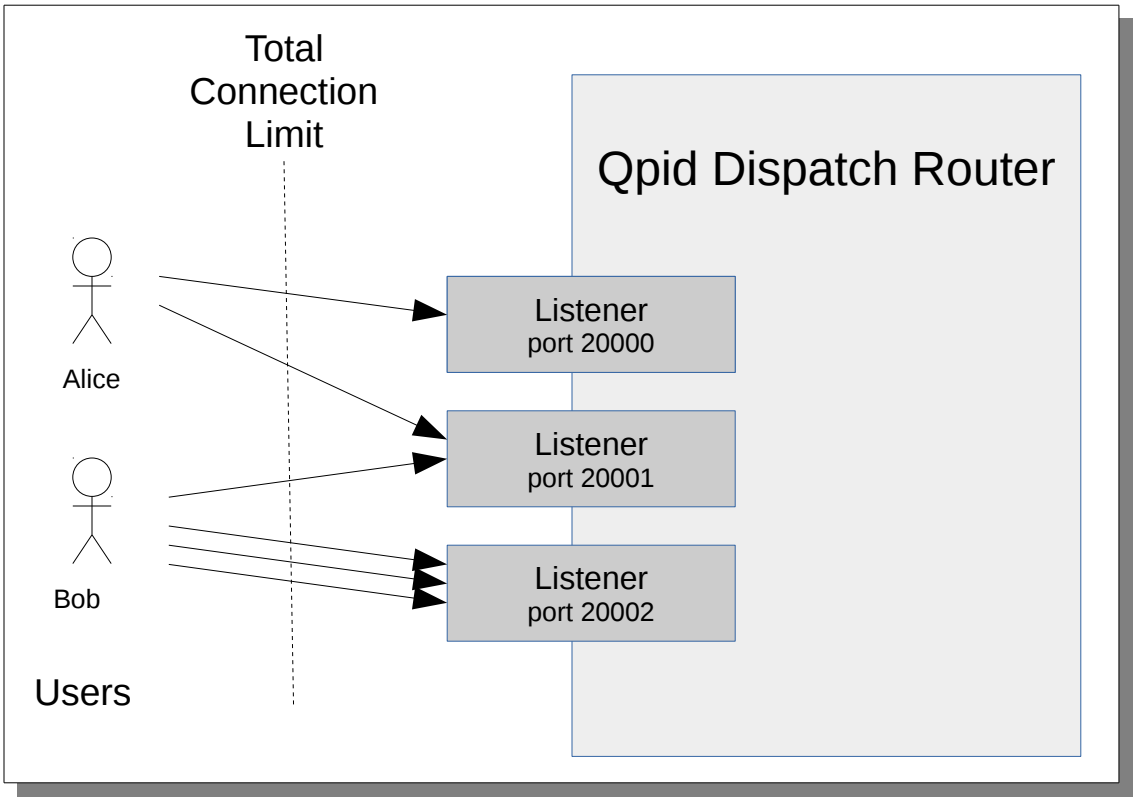

Figure – Policy Total Connection Limit

The Total Connection Limit enforces a maximum number of user connections to the QDR service listeners.

The total connection limit is enforced at a low level in QDR when a TCP socket is being opened. The same limit value applies to all user connections from any remote host. If the total connection limit is reached then the socket is simply closed. This limit protects against file descriptor resource exhaustion. This limit is specified and enforced independently of any other Policy settings.

Note that the total connection limit applies only to incoming user network connections. The limit does not apply to outbound connections created by QDR nor to inter-router connections.

Enable Access Rules

This boolean enables or disables Application rule processing. When this setting is false then none of the Policy denial decisions are effected.

Policy Folder

The *policyFolder* setting is the path to a folder that holds PolicyRuleset (ruleset) definition .json files. All rulesets in all .json files in this folder are added to the router configuration.

All rulesets may be defined in the main router configuration file and this may be adequate for a small deployment. However, this approach does not scale well. In a larger deployment it is better to have the Policy files in their own folder. The *policyFolder* setting defines the location of that folder.

Application Policy

Policy for an Application is self-contained. Definitions and lists defining one Application may be updated or deleted without affecting any other application.

Policy for an Application is defined in four sections:

- Connection Limits
- User Groups
- Connection Ingress Rules
- Group Settings

Application Connection Limits.

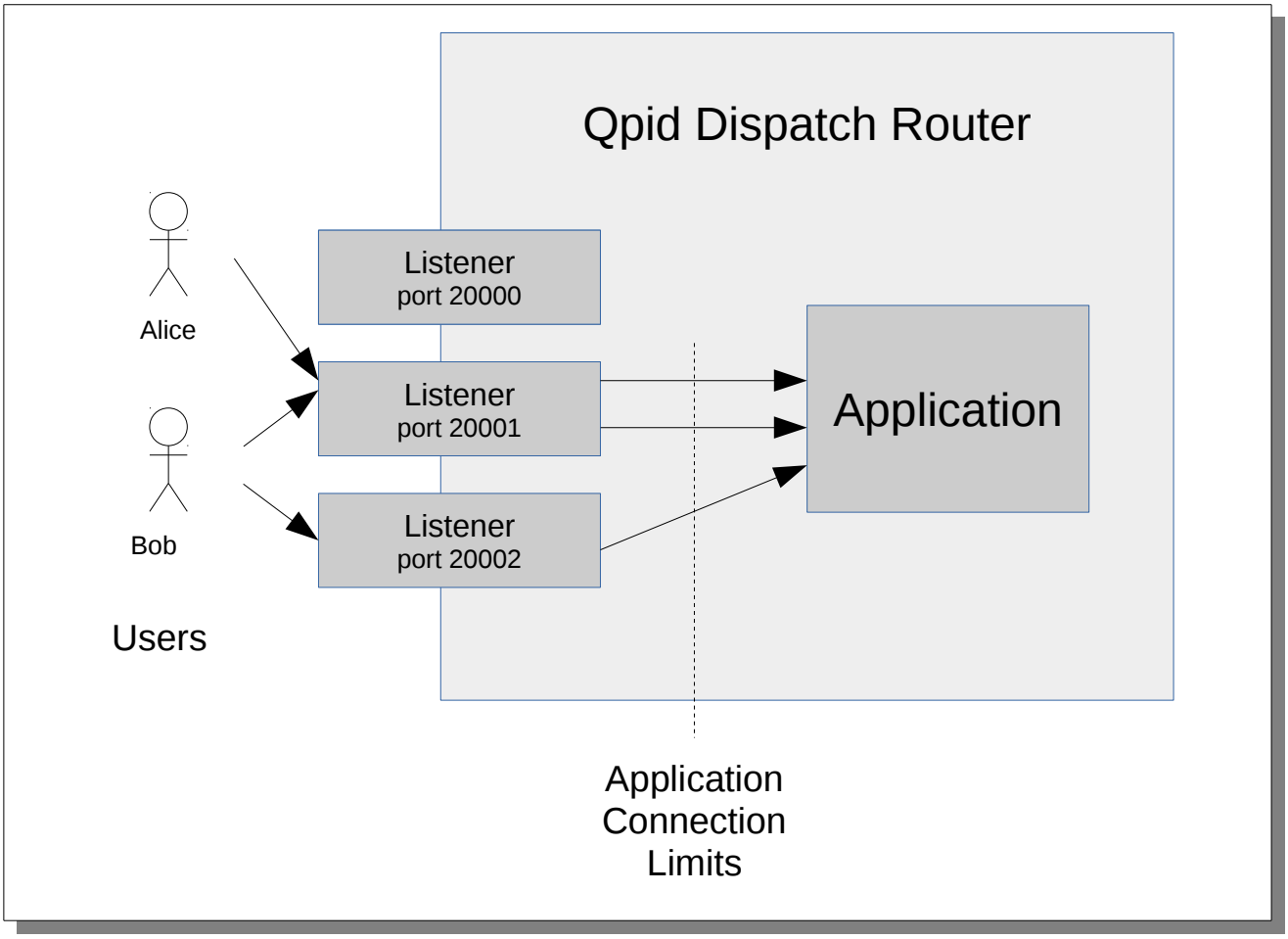

Figure – Policy Application Connection Limits

The Application Connection Limits enforce a maximum number of user connections that the QDR will allow to a given application. Several limits may be specified.

Application Connection Limit	Enforces
maximumConnections	The total number of connections allowed regardless of source.
maximumConnectionsPerUser	The total number of connections allowed for a given authenticated user regardless of which remote host the user is connecting from.
maximumConnectionsPerHost	The total number of connections allowed from a remote host regardless of which user is originating the connection.

Table – Policy Application Connection Limits

User Groups

QDR configuration for user group assignment consists of a ***userGroups*** dictionary where the key is the user group name and the value is the list of authenticated user names in that group. For instance:

```
userGroups[“admins”] = “Alice, bob, harry”
userGroups[“users”]  = “user1, user2, user3”
```

When user ***Alice*** connects she is assigned to the ***admins*** user group.

- A user may be in only one group.
- Users in a user group are subject to Connection Ingress policy rules.
- Users who are not in any group are assigned to the group ***default***. A configuration boolean ***connectionAllowDefault*** controls whether users in the default group are allowed to connect.

Connection Ingress Rules

QDR configuration defines groups of network hosts and which user group users are allowed to connect from those hosts. These are effectively white lists of hosts that limit user connections. The configuration defines two tables to effect ingress checks.

Table ***ingressHostGroups*** is a dictionary where the key is an ingress host group name and the value is a CSV list of IP hosts in that group.

```
ingressHostGroups[“localhost”] = “127.0.0.1, ::1”
ingressHostGroups[“Ten18”]     = “10.18.0.0-10.18.255.255”
ingressHostGroups[“social”]    = “example.com”
ingressHostGroups[“anywhere”]  = “*”
```

A user connecting from 10.18.33.145 would be in the ***Ten18*** ingress host group.

- Numeric host IP addresses may be IPv4 or IPv6.
- Host ranges are two numeric addresses of the same family separated by a hyphen. Do not mix IPv4 and IPv6 addresses in a range.

- Host group ranges may overlap. IP addresses may be in any number of host groups.
- The host wildcard is '*****'.

Table ***ingressPolicies*** is a dictionary where the key is a user group name and the value is a list of ingressHostGroups for those users.

```
ingressPolicies[“admins”] = “localhost, Ten18”
ingressPolicies[“users”]  = “social”
```

User ***Alice*** in the user group ***admins*** must connect from a host in the ***localhost*** or the ***Ten18*** ingressHostGroups entry. Similarly user ***user1*** would be assigned to the user group ***users*** and constrained to connecting from a host in the ***social*** ingressHostGroups entry.

- If a user group has no ingress policy then all connections from that group are allowed.

Application Resource Limits

If an authenticated user is allowed to connect to an application then the connection is subject to further restrictions. Limits are applied to various AMQP settings and protocol operations.

Application Resource Limit	**Enforces**
AMQP Connection Open	
maxFrameSize	Largest frame that may be sent on this connection. (AMQP Open, max-frame-size)
maxSessions	Maximum number of sessions that may be created on this connection. (AMQP Open, channel-max)
AMQP Session Begin	
maxSessionWindow	Incoming capacity for new sessions. (AMQP Begin, incoming-window)
AMQP Link Attach	
maxMessageSize	Largest message size supported by links created on this connection. (AMQP Attach, max-message-size)
maxSenders	Maximum number of sending links that may be created on this connection.
maxReceivers	Maximum number of receiving links that may be created on this connection.
allowDynamicSrc	This connection is allowed to create receiving links using the Dynamic Link Source feature.
allowAnonymousSender	This connection is allowed to create sending links using the Anonymous Sender feature.
sources	List of Source addresses allowed when creating receiving links. This list may be expressed as a CSV string or as a list of strings.
targets	List of Target addresses allowed when creating sending links.This list may be expressed as a CSV string or as a list of strings.

Table – Policy Application Resource Limits

Policy Schema

This section describes the QDR schema associated with Policy.

Global Policy

policy

Configuration Attribute	
maximumConnections	Global maximum number of concurrent client connections allowed. Zero implies no limit. This limit is always enforced even if no other policy settings have been defined.
path	Path to a folder that holds policyRuleset definition .json files. All rulesets in all .json files in this folder are processed.
enableAccessRules	Enable user rule set processing and application connection denial.
defaultApplication	Application policyRuleset to use for connections with no open.hostname or a hostname that does not match any existing policy. For users that don't wish to use open.hostname or any multi-tennancy feature, this default policy can be the only policy in effect for the network.
defaultApplicationEnabled	Enable *defaultApplication* policy fallback logic.
Status Attribute	
connectionsProcessed	Number of client connections processed.
connectionsDenied	Number of client connections denied.
connectionsCurrent	Number of client connections currently active.

Table – Policy Configuration: policy

Application Policy

policyRuleset

Configuration Attribute	
applicationName	Application name.
maxConnections	Maximum number of concurrent client connections allowed. Zero implies no limit.
maxConnPerUser	Maximum number of concurrent client connections allowed for any single user. Zero

	implies no limit.
maxConnPerHost	Maximum number of concurrent client connections allowed for any remote host. Zero implies no limit.
userGroups	A map where each key is a user group name and the corresponding value is a CSV string naming the users in that group. A user may be assigned to only one group.
ingressHostGroups	A map where each key is an ingress host group name and the corresponding value is a CSV string naming the IP addresses or address ranges in that group. IP addresses may be FQDN strings or numeric IPv4 or IPv6 host addresses. A host range is two host addresses of the same address family separated with a hyphen. The wildcard host address '*' represents any host address.
ingressPolicies	A map where each key is a user group name and the corresponding value is a CSV string naming the ingress host group names that restrict the ingress host for the user group. Users who are members of the user group are allowed to connect only from a host in one of the named ingress host groups.
connectionAllowDefault	Unrestricted users, those who are not members of a defined user group, are allowed to connect to this application. Unrestricted users are assigned to the 'default' user group and receive 'default' settings.
settings	A map where each key is a user group name and the value is a map of the corresponding settings for that group.

Table – Policy Configuration: policyRuleset

policyStats

Status Attribute	
applicationName	The application name.
connectionsApproved	
connectionsDenied	
connectionsCurrent	
perUserState	A map where the key is the authenticated user name and the value is a list of the user's connections.

perHostState	A map where the key is the host name and the value is a list of the host's connections.
sessionDenied	
senderDenied	
receiverDenied	

Table – Policy Configuration: policyStats

Policy Operation

Connection Approval

With no Policy in place QDR returns an AMQP Open to the client with default field values.

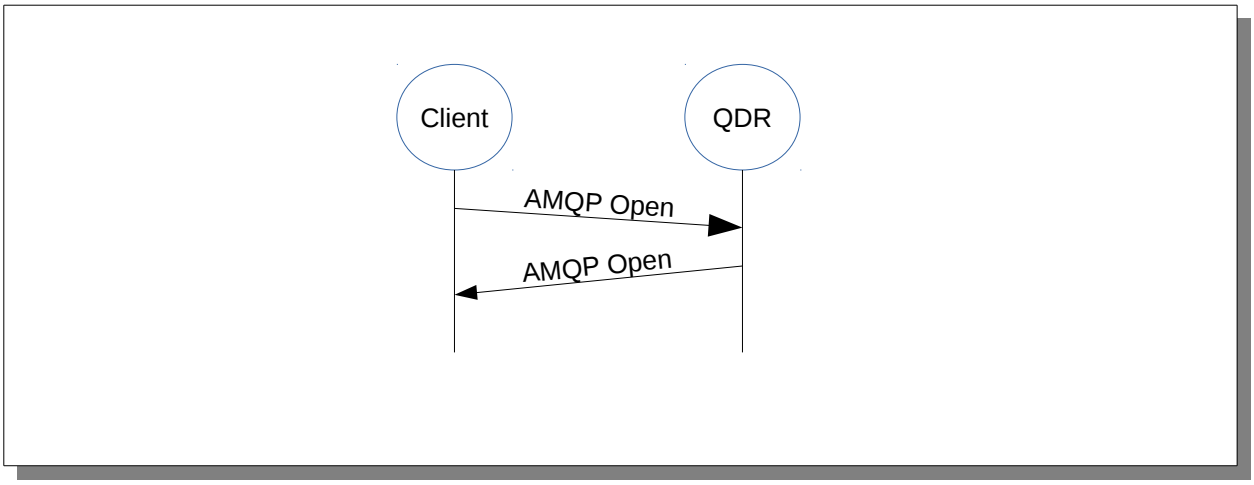

Figure – AMQP Open Forwarding with No Policy Approval

With Policy enforcement QDR intercepts the AMQP Open and queries Policy for approval.

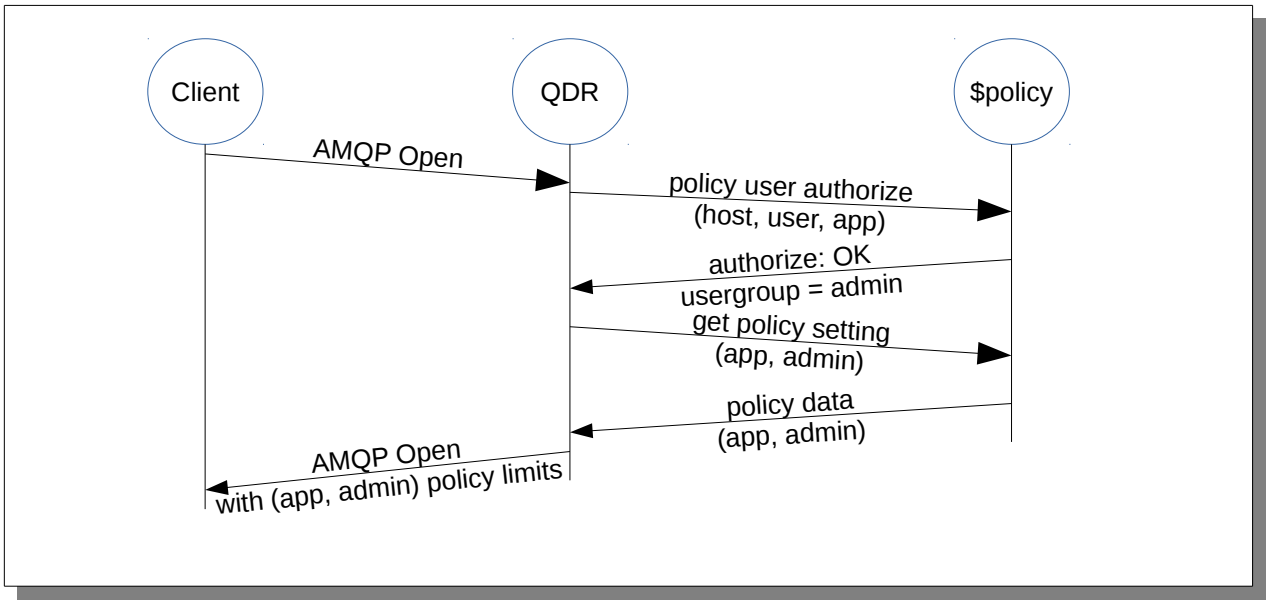

Figure – AMQP Open Forwarding with Policy Approval

The first step is the ***authorize(host, user, app)*** query to approve the user, verify the ingress host, and check application connection counts. The result of this query on approval is the name of the user group. The next step is the ***get policy settings (app, usergroup)*** where the application and user group names are presented and the policy engine returns the Resource Limit settings.

Connection Denial

When QDR denies a connection the following exchange takes place over the wire.

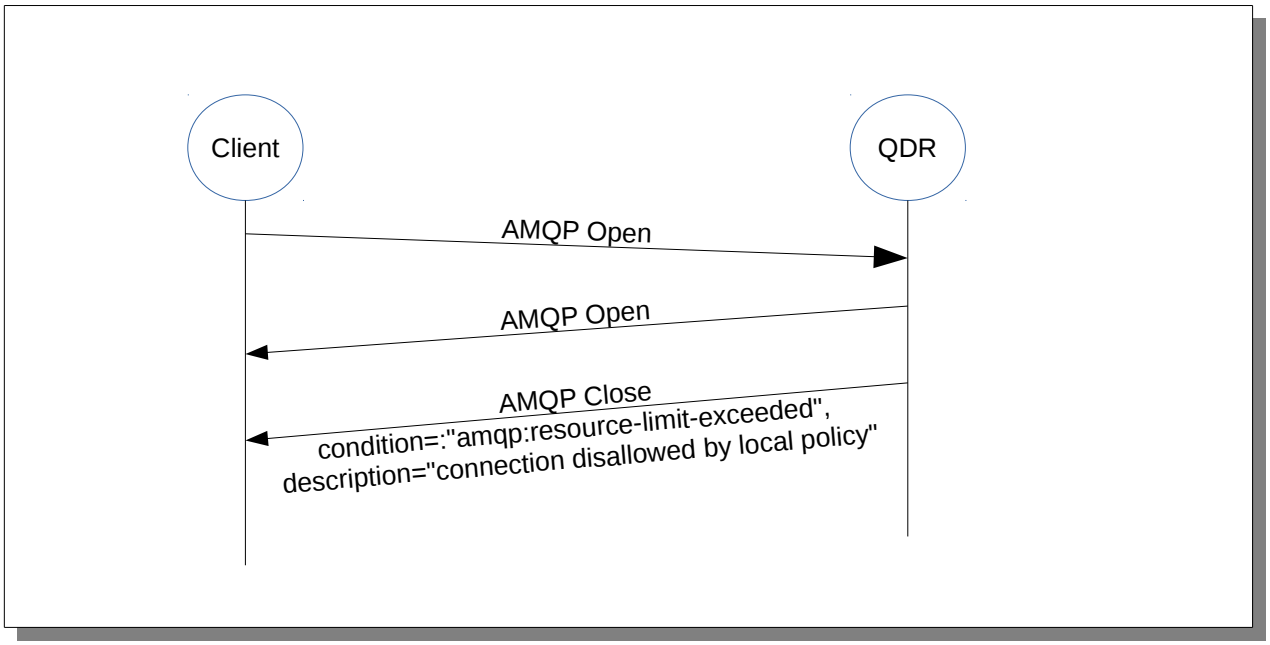

Figure – Connection Denial

1. The client sends an Open to QDR.
2. QDR goes through the user approval logic and determines that the connection should be denied. The first step of denial is to return an Open as if the connection is opening normally.
3. QDR then immediately sends a Close with an error condition.

Session Denial

Session denials are effected by limiting the ***channel-max*** attribute returned to the client in the AMQP

Open during an approved connection. The client should be unable to create a session beyond the channel-max value and thus the session limit is enforced by the AMQP protocol itself.

In the event that the client goes awry and opens a session using a forbidden channel number then QDR will respond the event the same as a connection denial except that the commands over the wire are *Begin* instead of *Open* and *End* instead of *Close*.

Link Denial

Link denials are handled the same as connection denials except the commands over the wire are *Attach* instead of *Open* and *Detach* instead of *Close*.

Links are counted across all sessions in the connections. If a Policy specifies maxSenders = 7 for example then only seven senders are allowed even if the connection has twenty sessions.

User Approval

For a given application user approval consists of

- assigning the user to a group name or to the default group
- verifying that the user is connecting from an approved network host

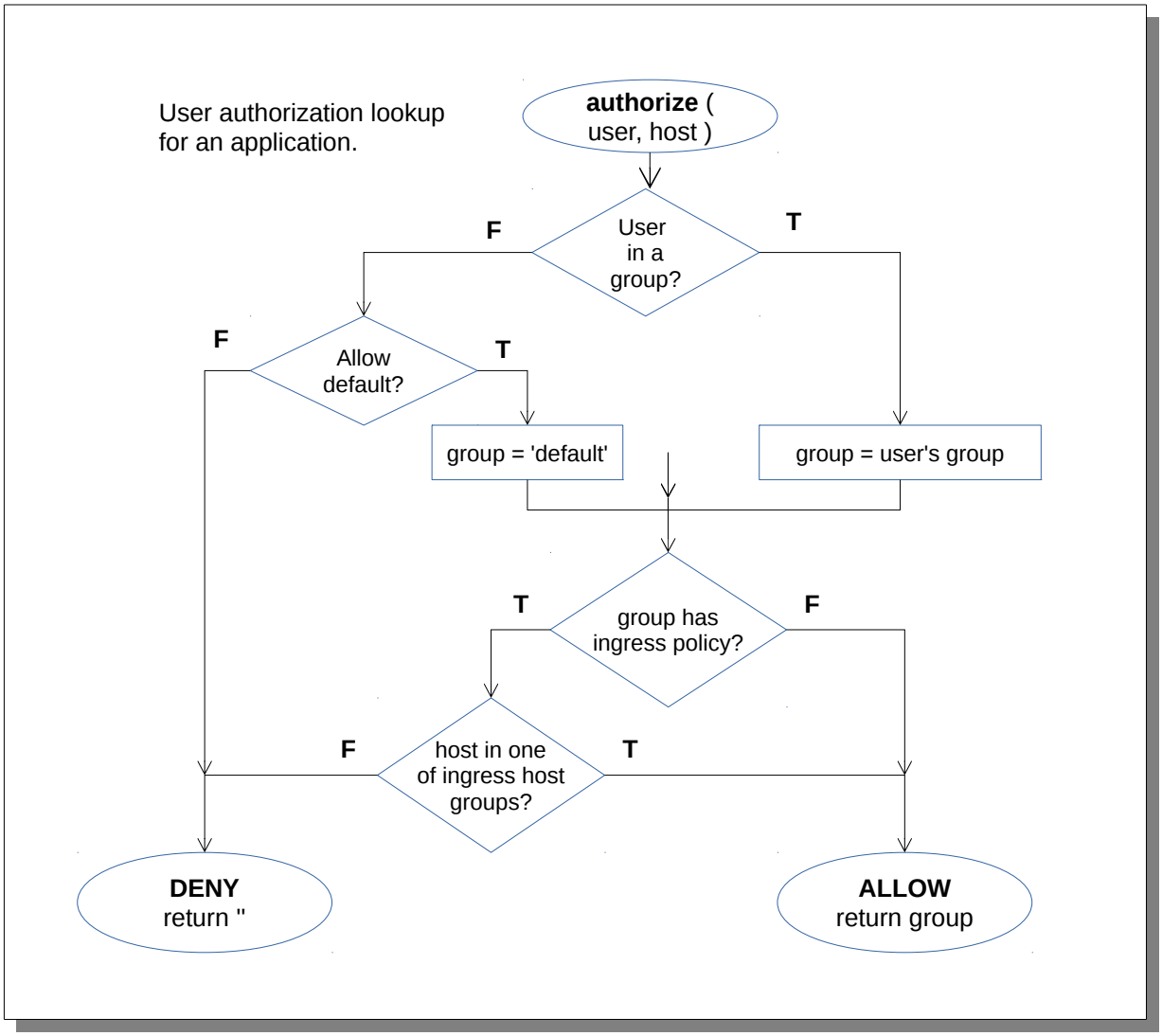

Figure – User authorization policy logic

The result of the authorization lookup for allowed access is the user's group name and for denied access is a blank string.

The final settings used for this user's connection are stored in the application's settings indexed by the user's group name.

Composing a Policy

User Name Wildcard

A username in a Policy ruleset userGroup specification may be terminated with an **'*'** to allow any user name beginning with the specified prefix to be accepted.

```
"Customers”: “customer*, subscriber_*”
```

Source and Target Lists User Name Substitution and Wildcard

In the ruleset definition for settings:sources and settings:targets may include the token “***${user}***” or a trailing **'*'**.

The *${user}* token is replaced with authenticated user name for the connection. Using this token an administrator may allow access to some resources for all users without naming each of them individually. This token may be used with the following restrictions:

- The token is substituted only once for the leftmost user name found in the incoming source our target link name.

The '*' is only recognized if it is the last character in the link name specification.

The *${user}* and '*' features may be combined.

Example Policy

Here is a simple demonstration Policy. **TO DO:** get some more realistic values.

```
["policyRuleset",
    {
        "applicationName": "boardwalk",
        "maxConnections": 10,
        "maxConnPerUser": 2,
        "maxConnPerHost": 5,
        "userGroups": {
            "anonymous": "anonymous",
            "users": "u1 u2 u3",
            "superuser": "ellen"
        },
        "ingressHostGroups": {
            "Ten18": "10.18.0.0-10.18.255.255",
            "EllensWS": "72.135.2.9",
            "TheLabs": "10.48.0.0-10.48.255.255, 192.168.0.0-192.168.255.255",
            "Localhost": "127.0.0.1, ::1"
        },
        "ingressPolicies": {
            "anonymous": "Ten18, TheLabs",
            "superuser": "Localhost, EllensWS"
        },
        "connectionAllowDefault": true,
        "settings": {
            "anonymous": {
                "maxFrameSize": 111111,
                "maxMessageSize": 111111,
                "maxSessionWindow": 111111,
                "maxSessions": 1,
                "maxSenders": 11,
                "maxReceivers": 11,
                "allowDynamicSrc": false,
                "allowAnonymousSender": false,
                "sources": "public*",
                "targets": ""
            },
            "users": {
                "maxFrameSize": 222222,
                "maxMessageSize": 222222,
                "maxSessionWindow": 222222,
                "maxSessions": 2,
                "maxSenders": 22,
                "maxReceivers": 22,
                "allowDynamicSrc": false,
                "allowAnonymousSender": false,
                "sources": "public, private_${user}*",
                "targets": "public, private_${user}*"
            },
            "superuser": {
                "maxFrameSize": 666666,
                "maxMessageSize": 666666,
                "maxSessionWindow": 666666,
                "maxSessions": 6,
                "maxSenders": 66,
                "maxReceivers": 66,
```

```
                "allowDynamicSrc": false,
                "allowAnonymousSender": false,
                "sources": "public, private, management, root*",
                "targets": "public, private, management, root*"
            },
            "default": {
                "maxFrameSize": 222222,
                "maxMessageSize": 222222,
                "maxSessionWindow": 222222,
                "maxSessions": 2,
                "maxSenders": 22,
                "maxReceivers": 22,
                "allowDynamicSrc": false,
                "allowAnonymousSender": false,
                "sources": "public, private",
                "targets": "public"
            }
        }
    }
]
```