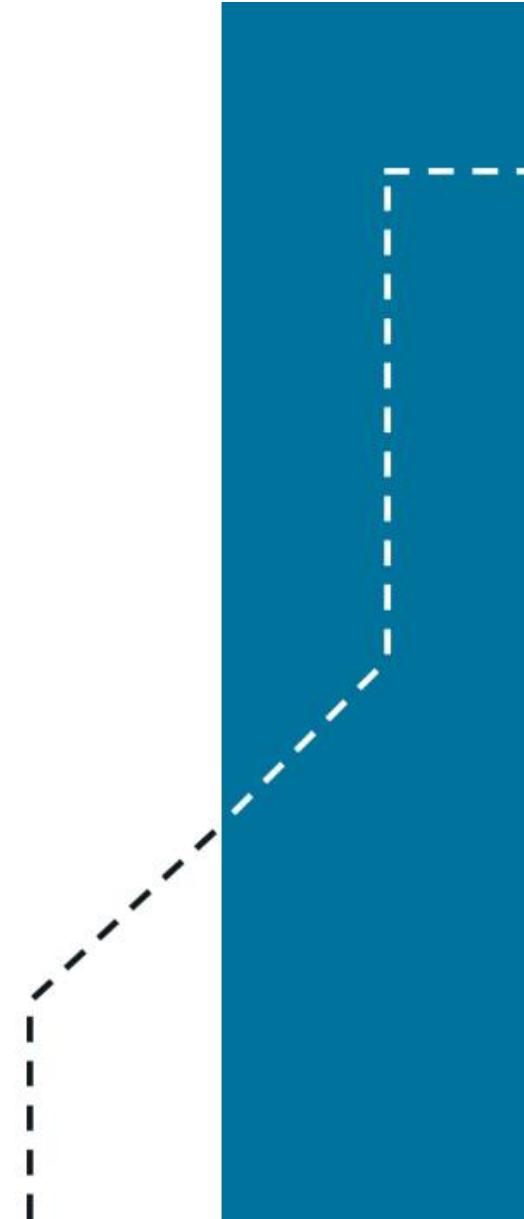




# ASSIGNING ACTIVITIES

## Assigning Activities



# ----- Lesson objectives

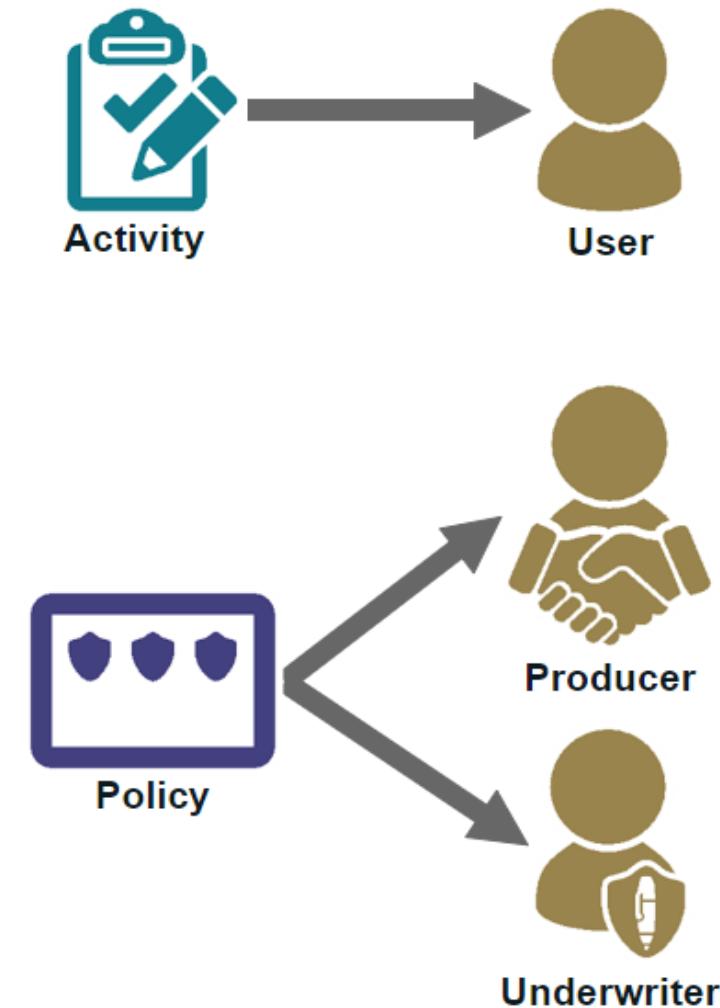
**By the end of this lesson, you will be able to:**

- Describe primary assignment
- Assign an activity

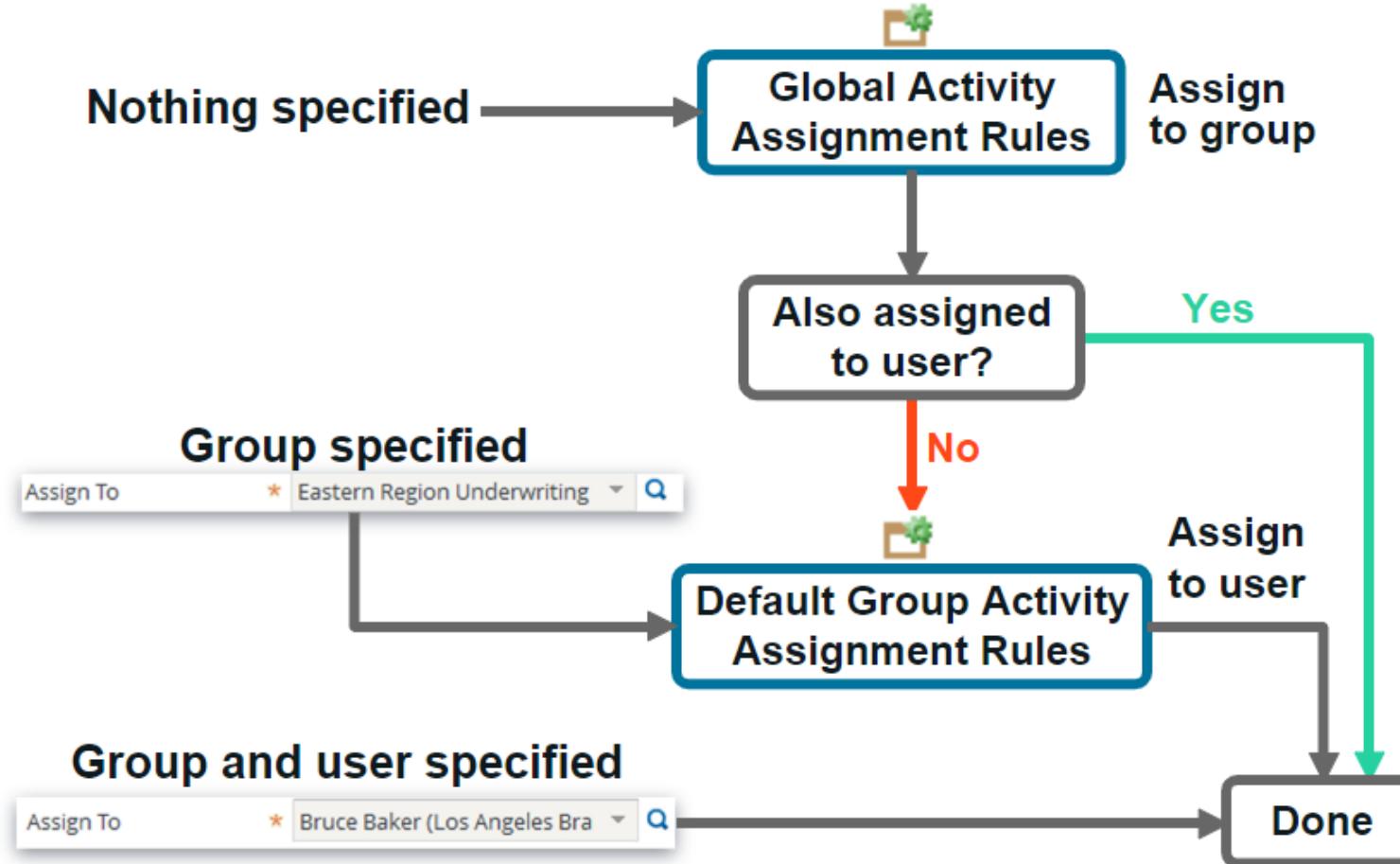
## Assignment basics

## Primary versus role-based assignment

- **Primary assignment** (discussed in this lesson)
  - The entity can have a single owner only
  - In the base configuration, this applies to **Activity** entities
- **Role-based assignment**
  - The entity has a set of users – with different roles – assigned to it
  - In the base configuration, this applies to **Account**, **Job** (and its subtypes), and **Policy** entities



# Typical flow of assignment execution



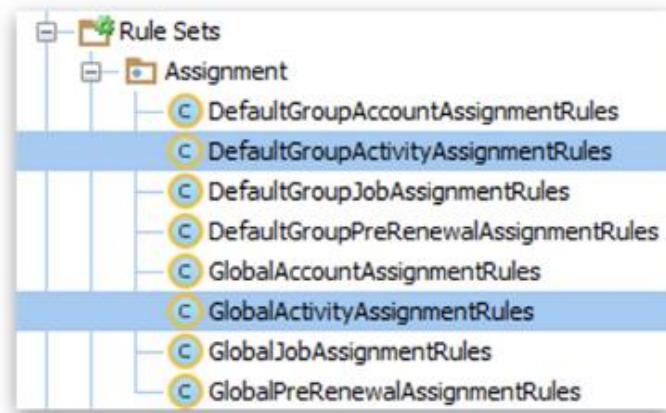
## Activity ownership

- An activity is owned by (assigned to) both a user and a group
  - The assigned user must be a member of the assigned group
  - Assigning “just” a user also assigns that user’s default group
- When an activity is assigned to a user:
  - It appears in that user’s My Activities desktop list
  - It can be edited only by that user (or other users with appropriate permissions, such as the owner’s supervisor)
  - It can be viewed by other users in the group

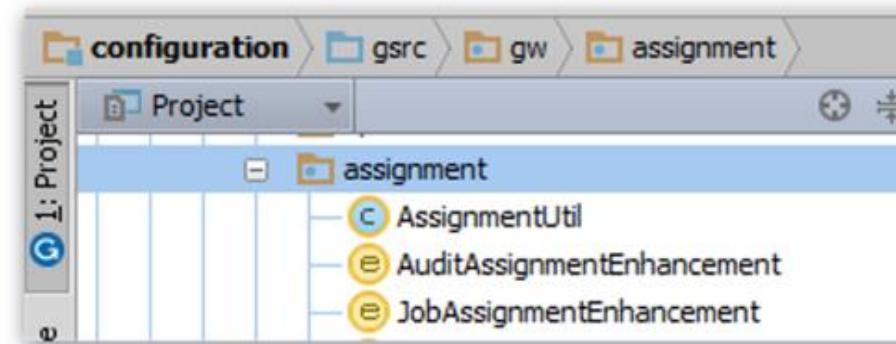
# Activity assignment

In PolicyCenter, an activity is assigned by:

- Assignment rules, or

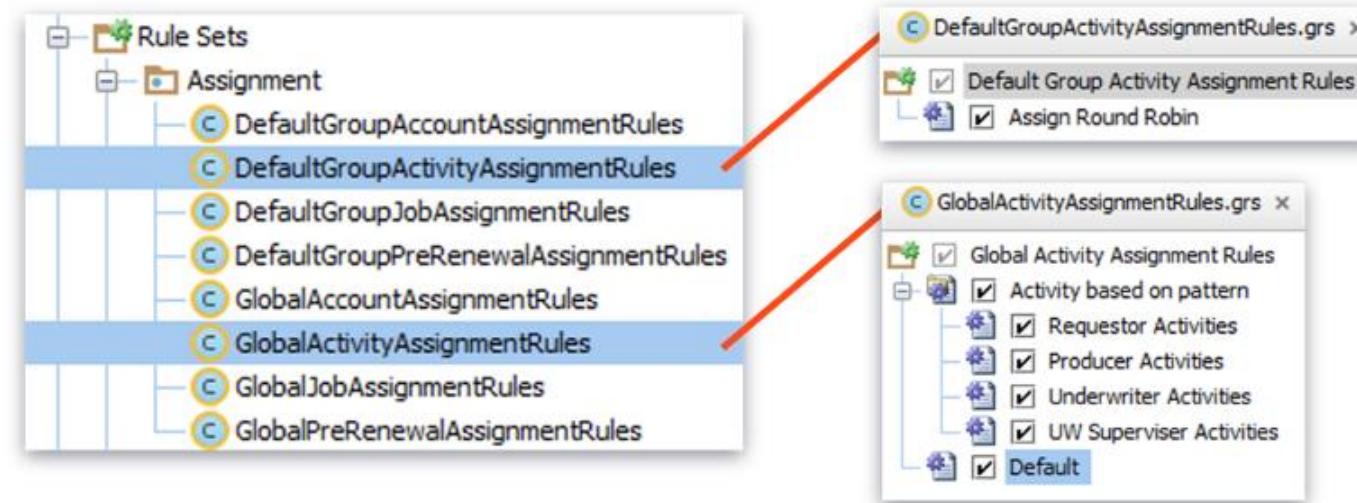


- Assignment classes, or



- Combination of both

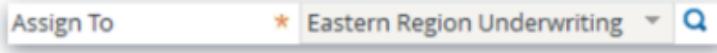
# Activity assignment rule sets



- **GlobalActivityAssignmentRules** assign activity to a group and may also assign to a user
  - Has rules that assign an activity by job role
  - OOTB Default rule assigns to the underwriter
- **DefaultGroupActivityAssignmentRules** assigns activity to user in assigned group
  - Has a rule that assigns activity to user in a group using round-robin

# Initiating activity assignment from Gosu

- By calling *autoAssign()* methods
  - Invokes activity assignment engine
  - Can cause assignment rules to be run
    - Do not call autoAssign() from within an assignment rule! 
- By calling other activity assignment methods directly in assignment classes

Signature	Assignment rules to be run
<i>Activity.autoAssign()</i>	Global and Default Group
<i>Activity.autoAssign( group, null )</i>	Default Group 
<i>Activity.autoAssign( group, user )</i>	None
<i>Activity.assign( group, user )</i>	None 

## autoAssign() example

- Business requirement: *Create an activity informing a user in the account creator group that a participant was updated on the account*
- Solution
  - Add code to PreUpdateHandlerImpl class, which executes logic before saving data to DB
  - Only default group rule set will run since group is specified

```
26  for (bean in context.UpdatedBeans) {
27      addBeansToBePreUpdated(entitySet, bean)
28
29      if(bean typeis AccountUserRoleAssignment) {
30          var act : Activity
31          var account = bean.Account
32          var actGroup : Group
33          var pattern = ActivityPattern.finder.getActivityPatternByCode("general_reminder")
34          act = account.newActivity(pattern)
35          act.Subject = "Account participant updated"
36
37          for(role in account.RoleAssignments) {
38              if(role.Role == typekey.UserRole.TC_CREATOR)
39                  actGroup = role.AssignedGroup
40          }
41
42          act.autoAssign(actGroup, null)
43      }
44  }
```

## Common assignment strategies

- Assign to a specific group and then round-robin among users
  - Round robin ignores users that lack permission to own the object
  - Round-robin functionality can be configured to take into account user workload/load factors
- Assign to a specific group and user
- Assign to a queue

# Common assignment methods

These methods on the Activity entity

- Can be used in assignment rules
- Can be called in assignment classes directly to initiate activity assignment

Context	Method	Assigns to...
Group	<i>assignGroup()</i>	Named group
Group	<i>assignGroupByLocation()</i>	Group matching specified location and group type
User	<i>assign()</i>	Named user (and group)
User	<i>assignUserAndDefaultGroup()</i>	Named user (default group)
Queue	<i>assignActivityToQueue()</i>	Named queue

## Classes you can configure

Methods you may need to modify activity assignment logic are under gsrc/gw/assignment:

- AuditAssignmentEnhancement
- JobAssignmentEnhancement
  - Example: **createRoleActivity()** method on Job entity
    - Creates activity from a specified ActivityPattern
    - Assigns to a user with a specified role on the job

## Always exit rule when configuring assignment rules

- When assignment succeeds
- This avoids reassigning an entity that has just been assigned
- Assignment methods return boolean values
  - **True** → assignment was successful
  - **False** → unsuccessful, appropriate group (or user) could not be found
- Place methods in if/exit wrapper to exit assignment rule set as needed

```
var theGroup = activity.Job.getUserRoleAssignmentByRole( "Underwriter" ).AssignedGroup
if (activity.assign( theGroup, theGroup.Supervisor )) {
    actions.exit()
}
```

If assignment is successful,  
ensures that no other  
assignment is done

## Group assignment

# Assigning to a specific group

- Assign object to a specified group
- Syntax:

*object.assignGroup ( group )*

```
// Assign activity to Los Angeles Branch UW group
var group = activity.findGroup("Los Angeles Branch UW")
if(activity.assignGroup(group)) {
    actions.exit()
}
```

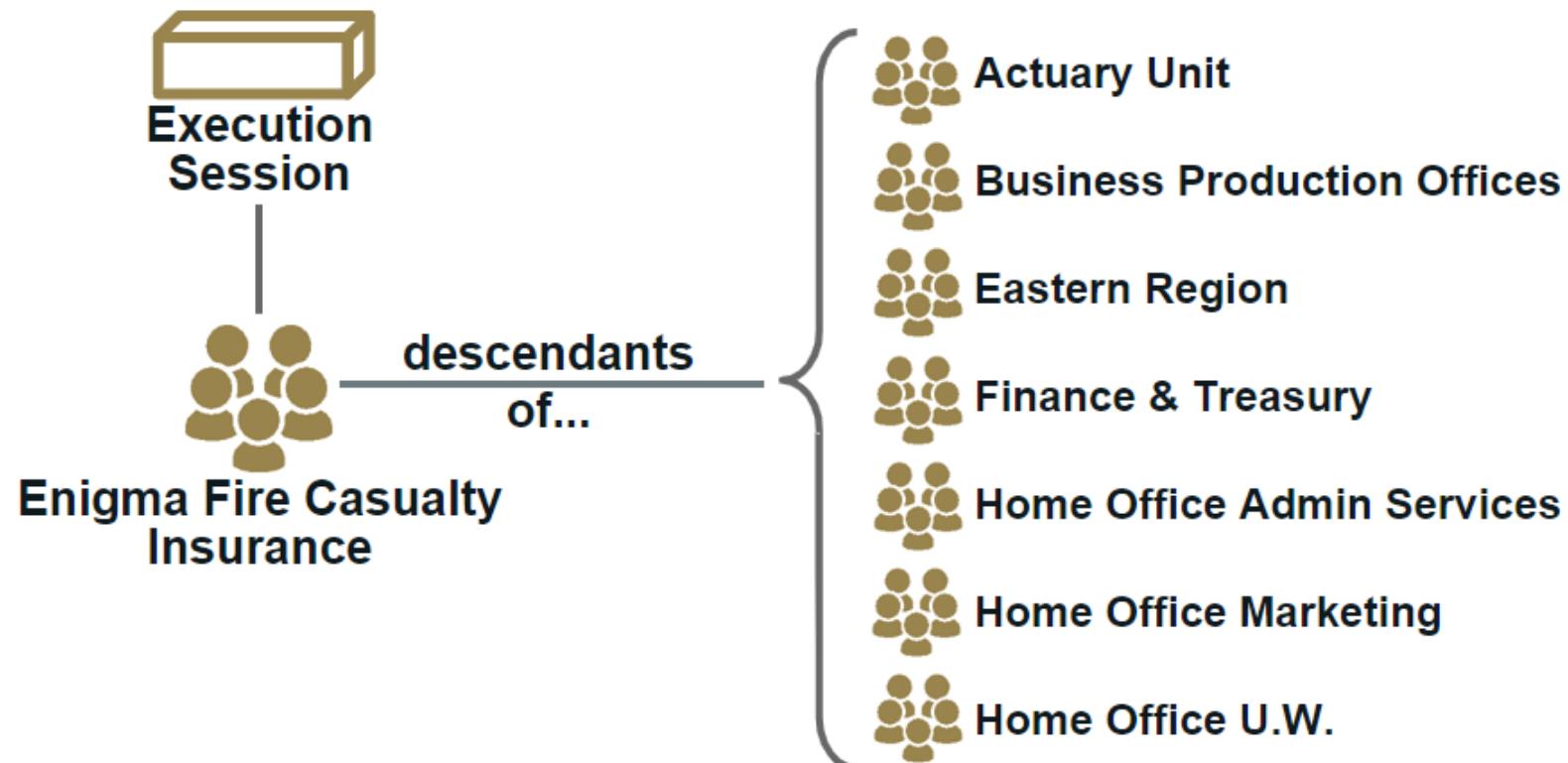
## Assigning to group by type and location

- Assigns to group:
  - Of specified group type...
  - ...whose region covers specified address...
  - ...including only direct child groups or all descendent groups...
  - ...beginning with children of specified group
    - Specified group typically set to *null* or *<entity>.AssignedGroup*
  - Using round-robin
- Syntax: *object.assignGroupByLocation (groupType, address, directChildrenOnly, group)*

```
if(activity.assignGroupByLocation(
    typekey.GroupType.TC_HOMEOFFICEADMIN, policyPeriod.PolicyAddress.Address, false, null)) {
    actions.exit()
}
```

## Step 1: Identify candidate groups

```
if(activity.assignGroupByLocation(  
    typekey.GroupType.TC_HOMEOFFICEADMIN, policyPeriod.PolicyAddress.Address, false, null)) {  
    actions.exit()  
}
```



## Step 2: Filter by group type

```
if(activity.assignGroupByLocation(  
    typekey.GroupType.TC_HOMEOFFICEADMIN, policyPeriod.PolicyAddress.Address, false, null)) {  
    actions.exit()  
}
```



## Step 3: Select by location

- Assume policy address is in California
- Based on zone-config.xml (see next slide), activity is assigned to based on zip, then county, then state

```
if(activity.assignGroupByLocation(  
    typekey.GroupType.TC_HOMEOFFICEADMIN, policyPeriod.PolicyAddress.Address, false, null)) {  
    actions.exit()  
}
```



**Home Office Admin Services**



**Eastern Mid-West Clerical**



**Eastern Region Clerical**



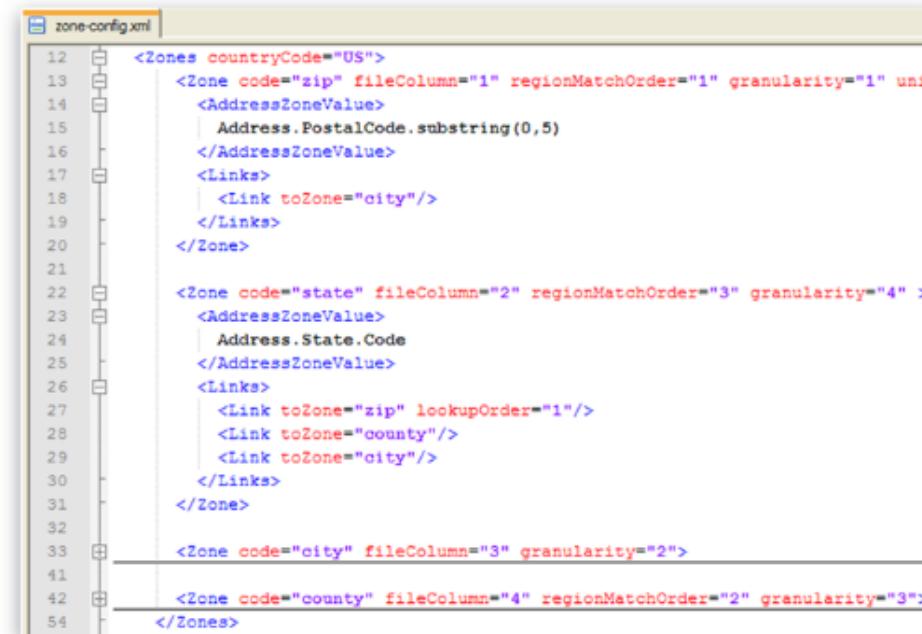
**Western Mid-West Clerical**



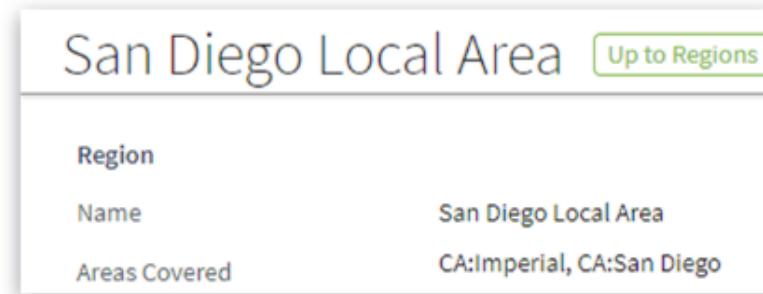
**Western Region Clerical**

# Configuring location selection

- Selection by location uses zones and regions
- A zone is a type of geographic area
  - Examples: State/province, county, city, postal code
  - Configured in zone-config.xml
- A region is a collection of specific zone values, configured through Administration tab
  - Western States
  - San Diego Local Area
  - State of Michigan plus Chicago



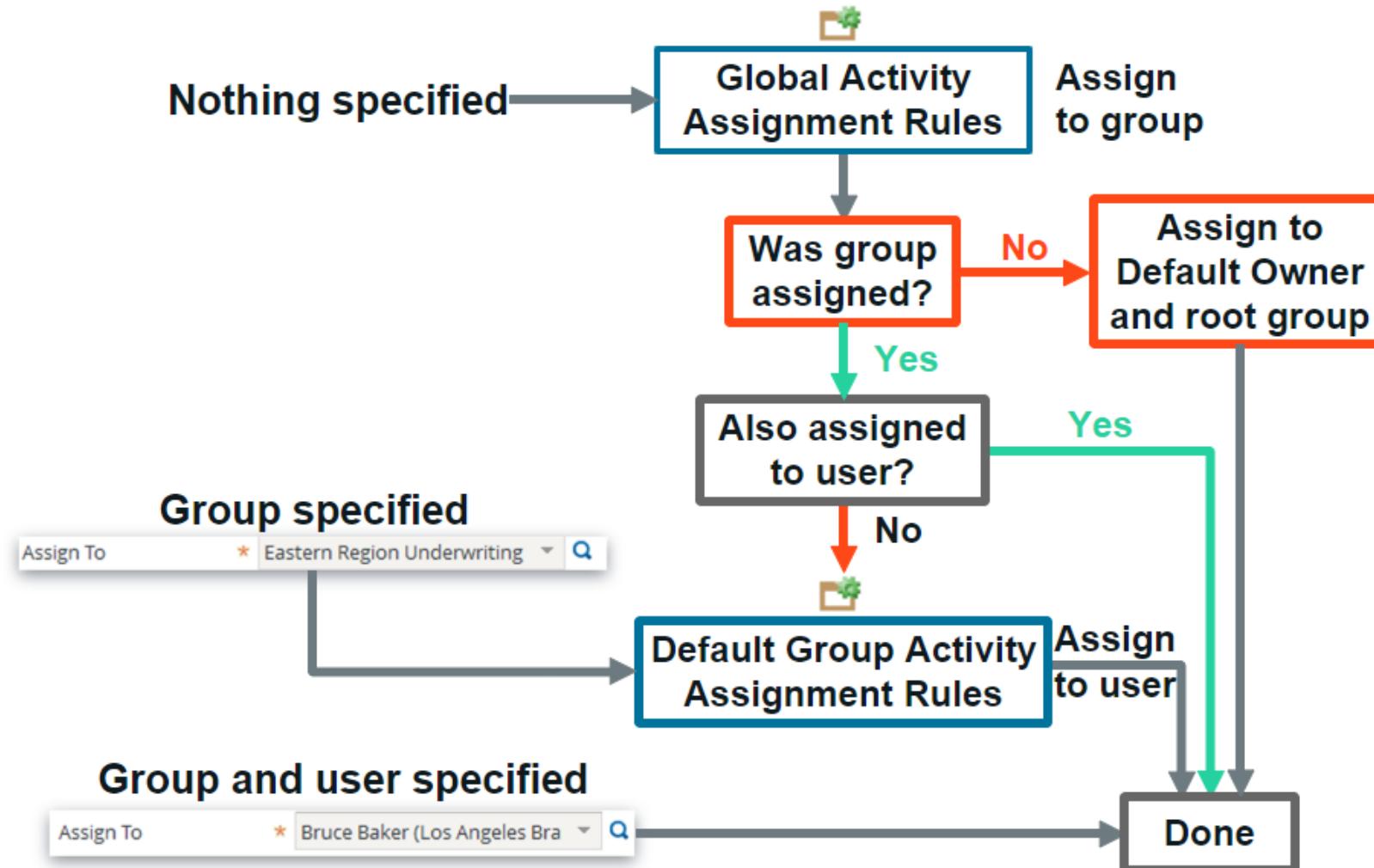
```
zone-config.xml
12 <Zones countryCode="US">
13   <Zone code="zip" fileColumn="1" regionMatchOrder="1" granularity="1" unique="true">
14     <AddressZoneValue>
15       Address.PostalCode.substring(0,5)
16     </AddressZoneValue>
17     <Links>
18       <Link toZone="city"/>
19     </Links>
20   </Zone>
21
22   <Zone code="state" fileColumn="2" regionMatchOrder="3" granularity="4" >
23     <AddressZoneValue>
24       Address.State.Code
25     </AddressZoneValue>
26     <Links>
27       <Link toZone="zip" lookupOrder="1"/>
28       <Link toZone="county"/>
29       <Link toZone="city"/>
30     </Links>
31   </Zone>
32
33   <Zone code="city" fileColumn="3" granularity="2">
34
35     <Zone code="county" fileColumn="4" regionMatchOrder="2" granularity="3">
36
37   </Zone>
38
39 </Zones>
```



San Diego Local Area [Up to Regions](#)

Region	
Name	San Diego Local Area
Areas Covered	CA:Imperial, CA:San Diego

# What if group assignment fails?



## User assignment

## Assigning to a specific user

- Assigns object to both a group and user
- Syntax:

*object.assign( group, user )*

```
// Assign activity to Los Angeles Branch UW group and Steve Smith
var group = activity.findGroup("Los Angeles Branch UW")
var user = activity.findUser("ssmith")
if(activity.assign(group, user)) {
    actions.exit()
}
```

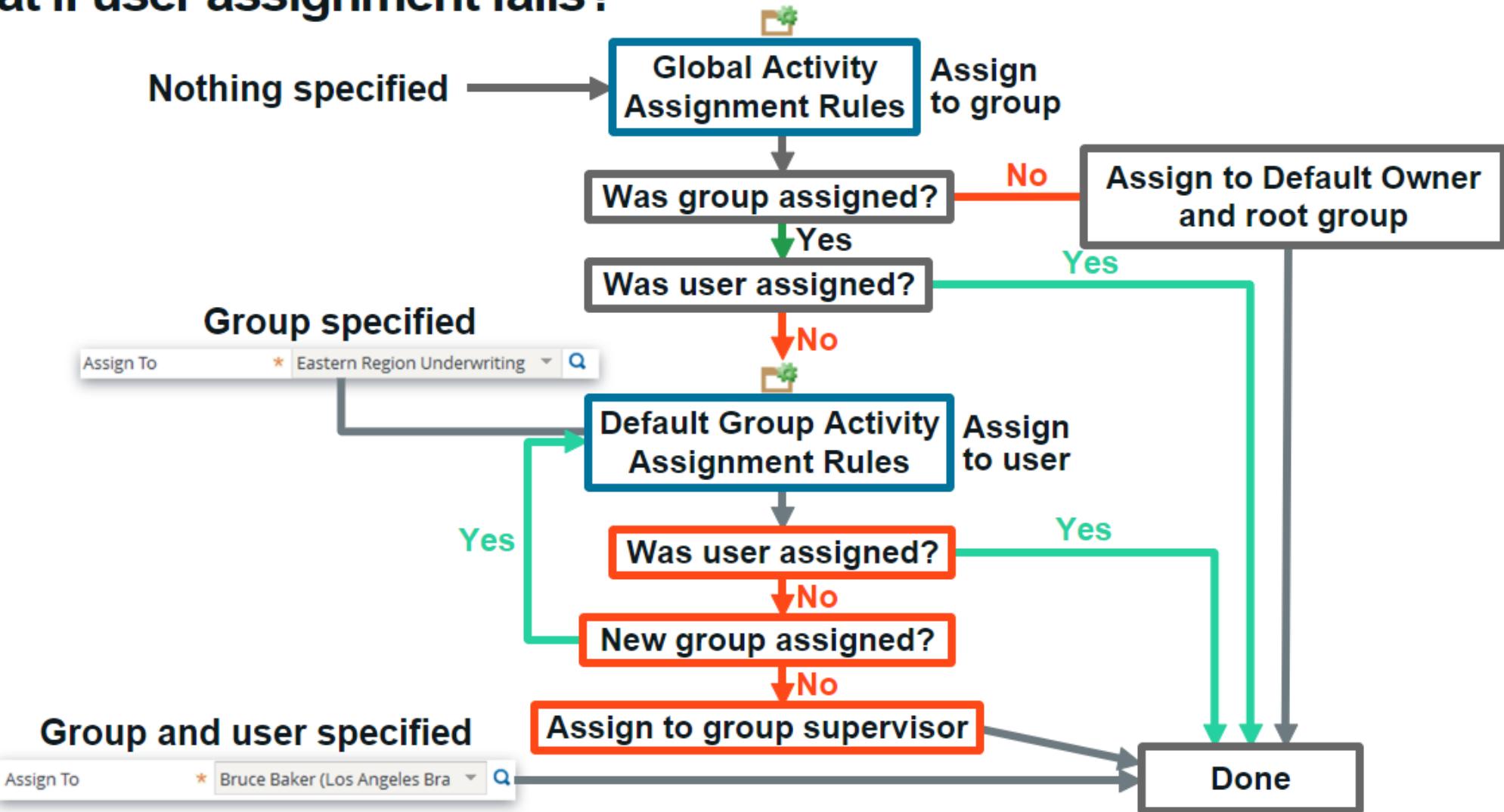
## Assigning to a user and any group

- Assigns object to a user and any group that the user belongs to
- Syntax:

*object.assignUserAndDefaultGroup (User)*

```
// Assign activity to Steve Smith
var user = activity.findUser("ssmith")
if(activity.assignUserAndDefaultGroup(user)) {
    actions.exit()
}
```

## What if user assignment fails?



# Review

# ----- Lesson objectives review

**You are now able to:**

- Describe primary assignment
- Assign an activity



This presentation contains information that may be privileged or confidential  
and is the property of the Capgemini Group.

Copyright © 2022 Capgemini. All rights reserved.

