CREATING ACTIVITIES

Creating Activities

Lesson objectives

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

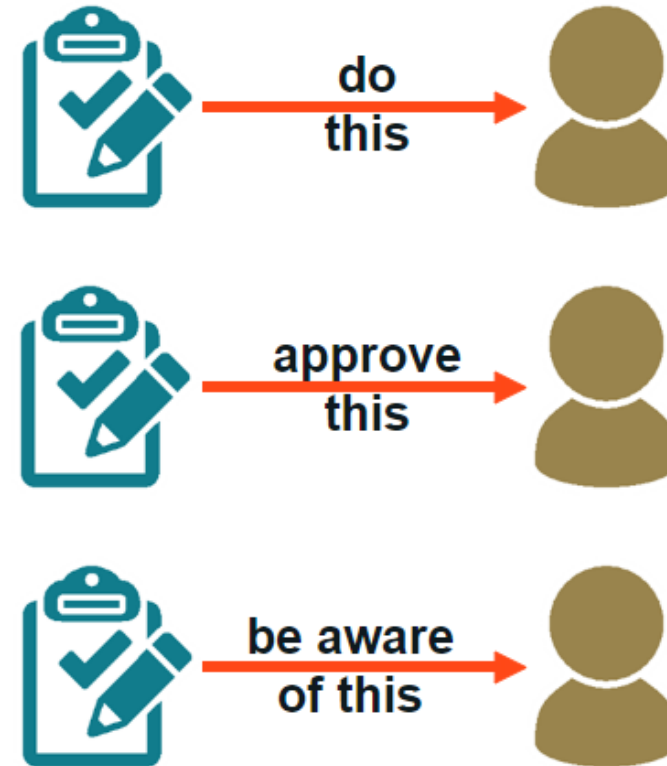- Create and configure activity patterns
- Create and configure activities
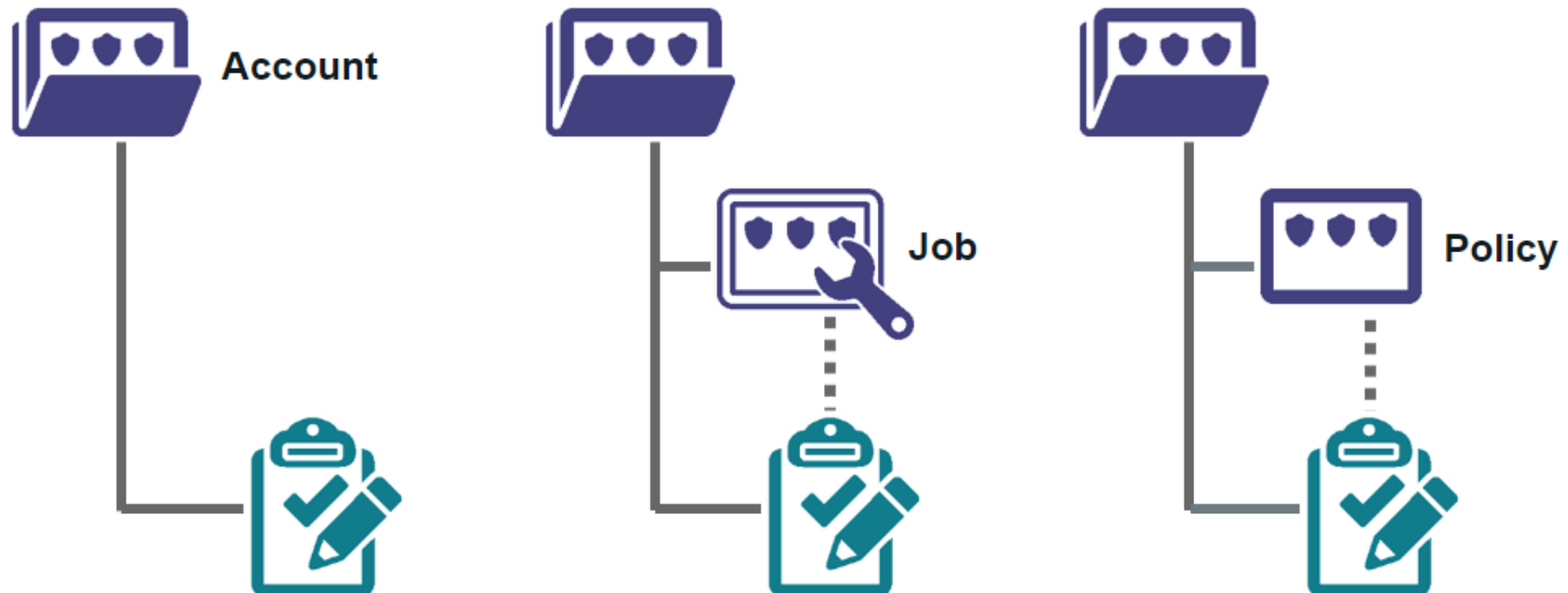
Activity overview

Activity

- An **activity** represents a task that someone must attend to or an issue that someone should be aware of

- Used to:

  - Let automatic processes assign work to user

  - Let a user assign work to another user

  - Alert user to certain event, such as an underwriting issue

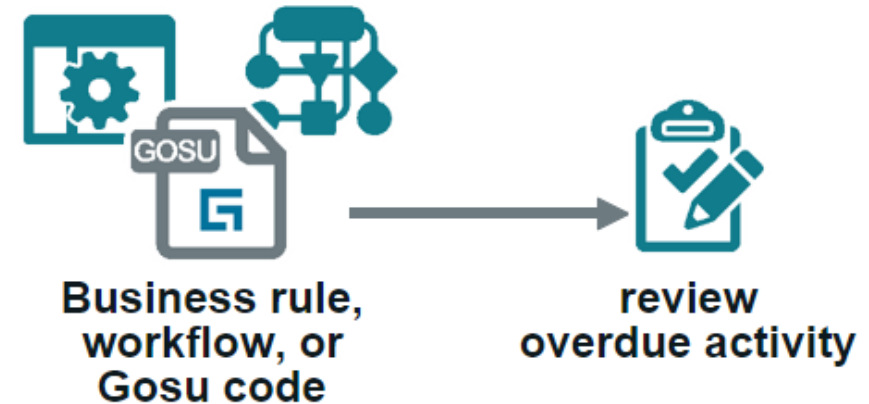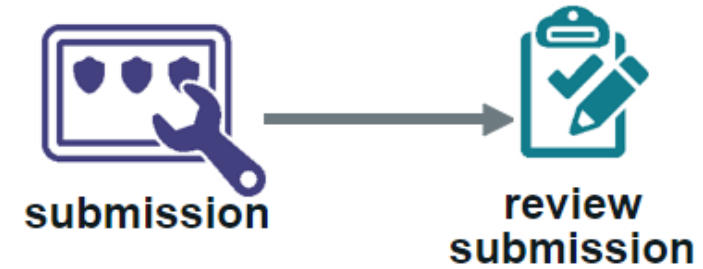do this

approve this

be aware of this

Activity parent objects

- Every activity is associated with an account

- Can also be associated with a job (and its subtypes) or a policy, which implicitly links it to their account, which implicitly links it to their account

Account

Job

Policy

When are activities created?

- Automatically during policy job
  - Typically created when work must be done by a user to move the job forward

- Automatically by
  - Business rule
  - Gosu code
  - Workflow

- Manually by end user
  - Typically done to request work to be done by another user

submission → review submission

Business rule, workflow, or Gosu code → review overdue activity

User → review producer

Activity pattern

Activity patterns

- Activity patterns are templates that standardize how activities are generated

- Both the user interface and Gosu code create activities based on these patterns

Activity pattern

Subject = Rate Policy
Priority = medium
Due Days = 5
Escalation Days = 10

Activity

Subject: Rate Policy
Priority: medium
Due Date: May 12
Escalation Date: May 17

Managing activity patterns

- Activity patterns are typically created and modified by administrators

- Administration tab > Business Settings > Activity Patterns

- To create a new activity pattern, click the New Activity Pattern button

  - Code field is a unique internal name for the pattern and is used to reference an activity pattern in Gosu

- To edit an existing activity pattern, click the name of the activity pattern in the list

  - Changes to an activity pattern affect only activities from that point forward

Creating activities in Gosu

Using activity patterns in Gosu

- There are two operations that you can perform in Gosu involving activity patterns

    - One is to test which activity pattern an existing activity uses

        - *Entity.ActivityPattern.Code == "pattern_code"*

    - The other is to retrieve an activity pattern for use in creating a new activity

        - *ActivityPattern.finder.getActivityPatternByCode("pattern_code")*

        - Returns an ActivityPattern object

- Use the activity pattern code to refer to an activity pattern in Gosu code

    - Do not use a pattern ID or PublicID value

Create activity related to an account

- Gosu code

activityPattern.createAccountActivity(bundle, activityPattern, Account, subject, description, approvalIssue, priority, mandatory, targetDate, escalationDate)

or

account.newActivity(activityPattern)

- Example

```
var anAccount : Account
var anActivityPattern : ActivityPattern
anActivityPattern = ActivityPattern.finder.getActivityPatternByCode("notification")
anAccount.newActivity(anActivityPattern)
```

Create activity related to a policy

- Gosu code

*activityPattern.**createPolicyActivity(**bundle, policy, subject, description, approvalIssue,*
*priority, mandatory, targetDate, escalationDate**)*

- Example

```
if(policy != null) {
  var a : Activity
  var pattern = ActivityPattern.finder.getActivityPatternByCode("coverage")
  a = pattern.createPolicyActivity(
          policy.Bundle,
          policy,
          "Verify coverage",
          "Call customer to verify coverage",
          null,
          typekey.Priority.TC_NORMAL,
          false,
          null,
          null)
}
```

- By default, an activity is always related to an account

Create activity related to a job

- Gosu code

  activityPattern.createJobActivity(bundle, job, subject, description, approvalIssue, priority, mandatory, targetDate, escalationDate)

- Example

```
if(policyPeriod != null) {
  var a : Activity
  var pattern = ActivityPattern.finder.getActivityPatternByCode("coverage")
  a = pattern.createJobActivity(policyPeriod.Job.Bundle, policyPeriod.Job, null, null, null, null, null, null, null)
}
```

- By default, an activity is always related to an account

Review

Lesson objectives review

You are now able to:

- Create and configure activity patterns
- Create and configure activities

Capgemini