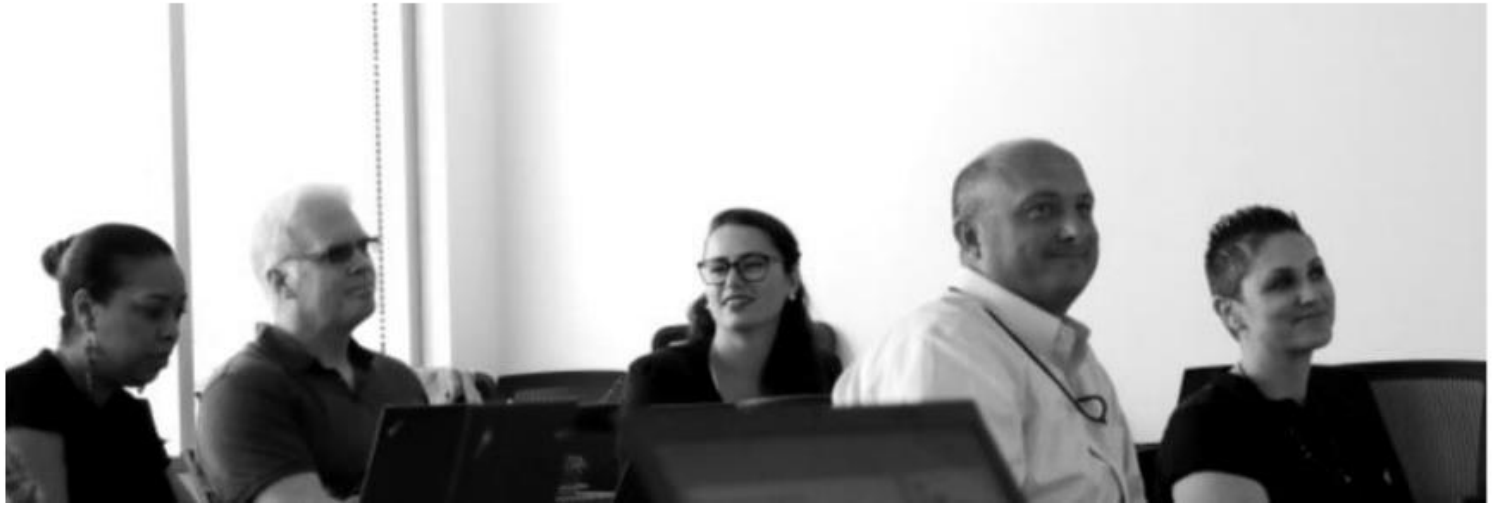
INTRODUCTION TO PERMISSION CONFIGURATION

Source : Guidewire Educational Platform

Introduction to Permission Configuration



Source : Guidewire Educational Platform

Lesson objectives

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

- Configure system permissions
- Write Gosu expressions that evaluate user permissions
- Describe static and object-based permissions

Source : Guidewire Educational Platform

PolicyCenter security and concepts

Source : Guidewire Educational Platform

Review: Role based Security in PolicyCenter

- Security is provided using:

Role

Permissions

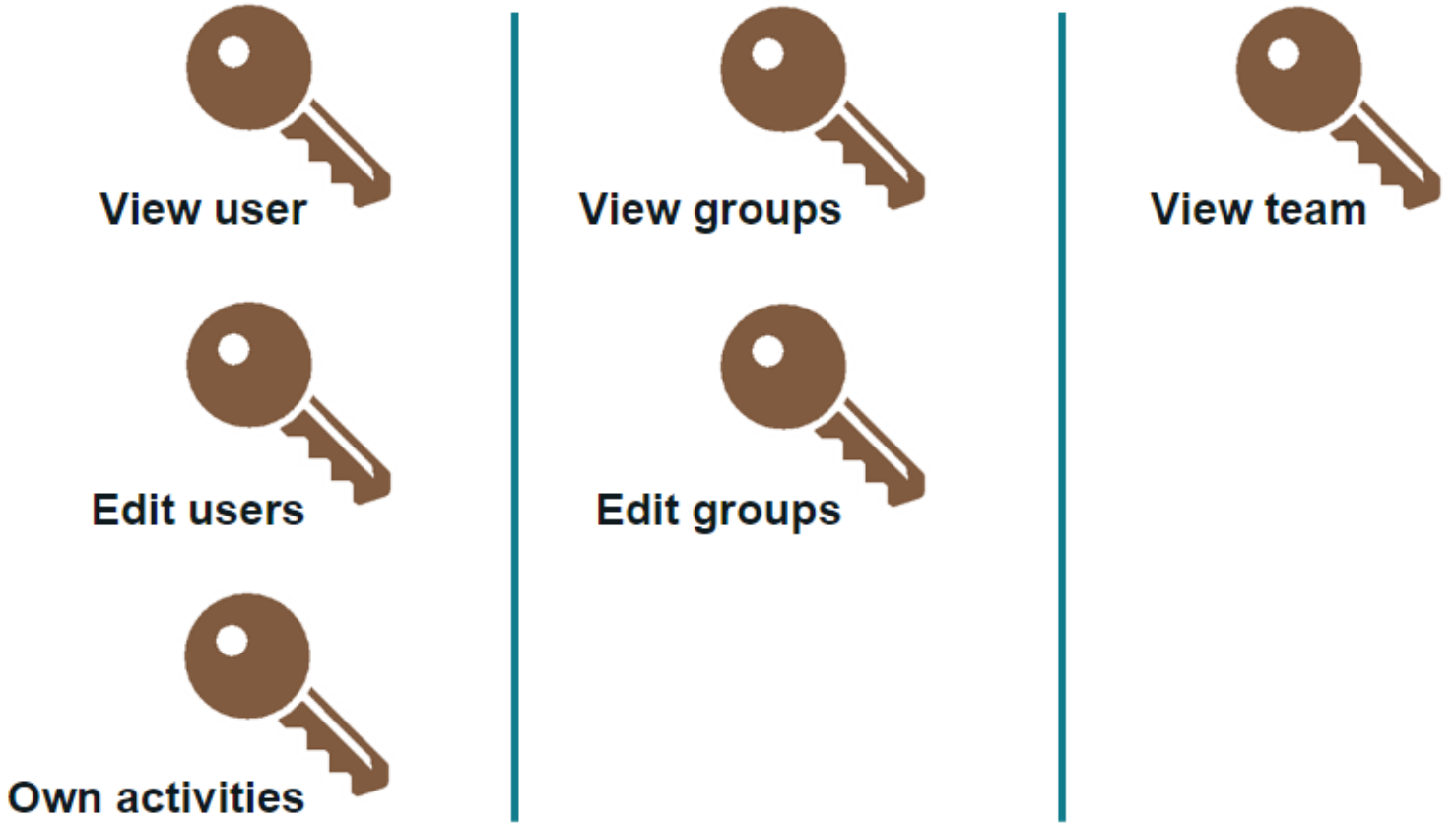
Producer codes

- A user ***must*** be assigned a role with the appropriate permissions
- In the base configuration, the **Superuser** role is:
  - Granted all permissions and
  - Responsible for granting permissions to other roles

Source : Guidewire Educational Platform

System permissions

A **system permission** is a granular ability to see or do something within PolicyCenter



Source : Guidewire Educational Platform

Roles

A **role** is a named collection of permissions used to simplify the assignment of permissions to users

- Typically, a role maps to a job title or a job function

Underwriter Assistant

View account

Create account
Create activities

Edit submission
Edit rewrite

...

Underwriter

View account
View sensitive notes
View sensitive docs

Create account
Create activities
Create audit

Edit submission
Edit rewrite
Edit rates and premium overrides

...

Source : Guidewire Educational Platform

Review: Security Dictionary

Security Dictionary is a series of HTML pages that document the permissions and roles in your application

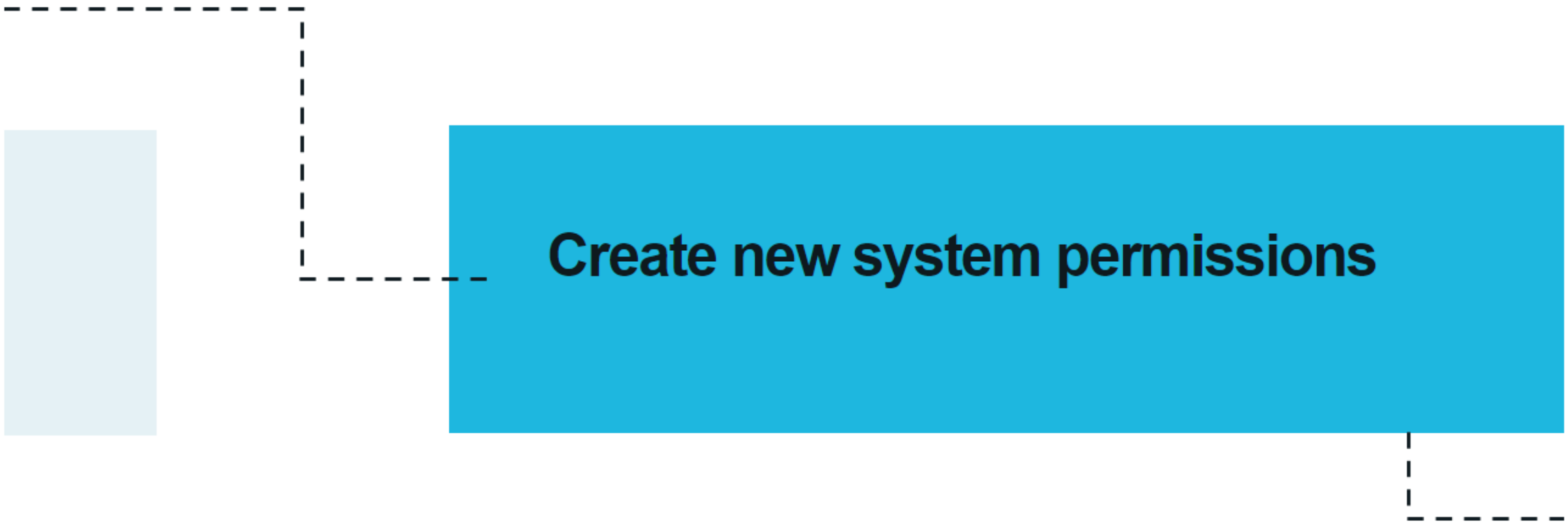
- Located at: *<server directory>\build\dictionary\security\index.html*
- When you add or edit permissions or roles, you can regenerate the information in these sections to reflect the changes
- The dictionary has four main sections

Source : Guidewire Educational Platform

Review: Security Dictionary main sections

- Each **Application Permission Key** displays the set of system permissions and the pages and elements referencing it in the UI
- Each **Page** lists conditions that check either system permission or application permission keys to view or edit the page
- Each **System Permission** lists the roles containing it and UI components referencing it
- Each **Role** lists the permissions it contains

Source : Guidewire Educational Platform



Source : Guidewire Educational Platform

Creating System permissions

- System permissions are defined in the SystemPermissionType typelist
  - Permissions that appear grayed are internal and cannot be modified
- Add typecodes in SystemPermissionType to create new system permissions
  - Permission codes should be all lowercase without white spaces
  - Specify one permission per action
  - Name permission codes as a verb for the entity name and action
  - Permission code: [entity][action] – can be interchanged
- Examples:
  - accountcreate: Permission to create an account
  - bindsubmission: Permission to bind a submission
  - viewsensnote: Permission to view sensitive notes

Source : Guidewire Educational Platform

Check permissions using Gosu

Source : Guidewire Educational Platform

The perm namespace

- **perm** is a Gosu namespace used to create expressions that determine if current user has a given permission
  - Expression returns true or false
- perm Syntax
  - perm.**System**.*permission* is used when user calls a system permission defined in the SystemPermissionType typelist
    - perm.System.viewteam
  - perm.**Entity**.*permission* is used when an application permission key is called
    - perm.Account.create
    - perm.Organization.create

Source : Guidewire Educational Platform

Typical uses of perm expressions

- Atomic widget attributes
  - Control visibility, editability, or availability of a field, button, or menu item
- Container widget attributes
  - Control visibility or editability of a detail view or list view
- Location attributes
  - Control ability to visit or edit a location
- Business rules
  - Modify rule behavior (such as generating additional activity if current user lacks a given permission)

Source : Guidewire Educational Platform

Static and object-based permissions

Source : Guidewire Educational Platform

Application permission keys (APKs)

- An **APK** is a set of one or more system permissions defined in security-config.xml
- APKs use custom handlers to define the mapping of different objects in PolicyCenter to system permissions
  - For example, StaticHandlers, WrapHandler, AccountProducerCodeHandler
- PolicyCenter defines APKs for following objects:
  - Account
  - PolicyPeriod
  - Jobs such as submission, rewrite, or policy change
  - Notes
  - Documents
  - Activities

Source : Guidewire Educational Platform

Static and object-based permissions

- **Static permissions**
  - Do not require an object as an argument
  - May or may not use APKs
  - For example, the permission to create a new activity pattern

Create activity pattern

- **Object-based permissions**
  - Always require an object as an argument
  - Always use APKs
  - For example, permission to reassign an owned activity (must pass in the activity to be reassigned)

Reassign owned activities

Source : Guidewire Educational Platform

Static or object-based in security dictionary

Whether a permission is static or object-based is indicated in the security dictionary

Source : Guidewire Educational Platform

Static system permissions

- **perm.System.permission**
  - permission: typecode of a permission defined in SystemPermissionType typelist
  - Returns true if current user has the permission
- Example:
  - perm.System.viewteam

Source : Guidewire Educational Platform

Static permissions on entities

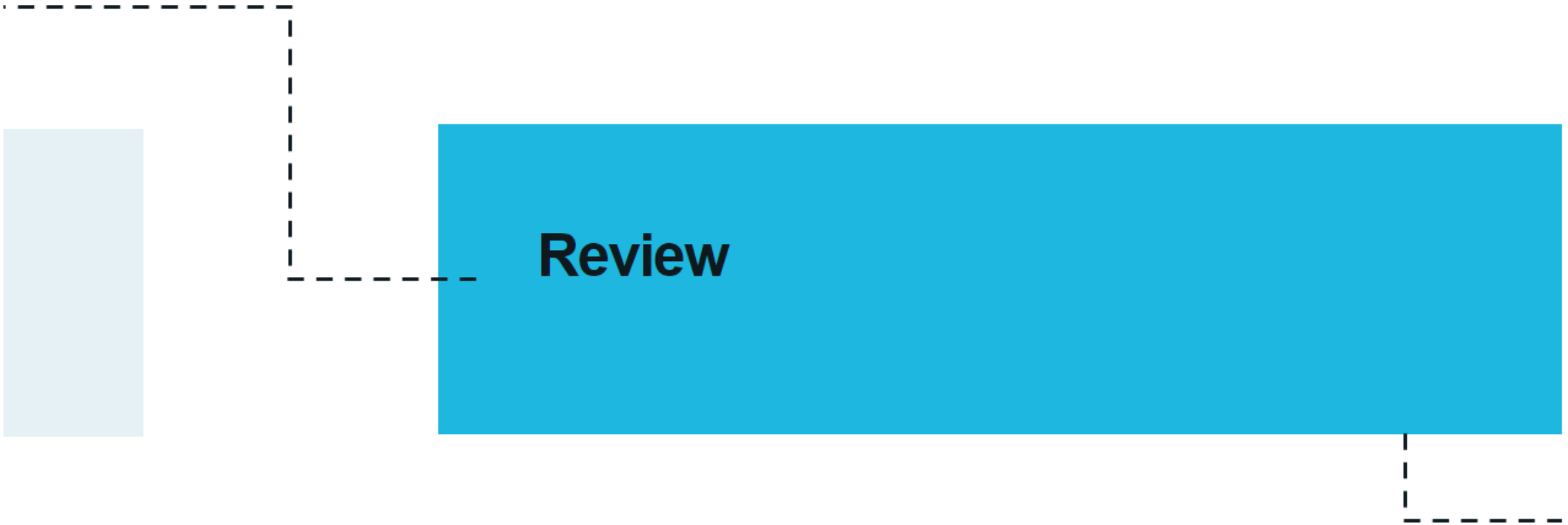
- APKs defined using StaticHandlers in security-config.xml
- **perm.Entity.permission**
  - Entity: entity on which the permission is defined
  - permission: permKey attribute defined in security-config.xml
  - Returns true if current user has the permission
- Example:
  - perm.Account.create

Source : Guidewire Educational Platform

Object-based permissions

- APKs defined using custom Handlers in security-config.xml
- Object has to be passed in as an argument in the perm expression for checking permissions
- **perm.Entity.permission (object)**
  - Entity: entity on which the permission is defined
  - permission: permKey attribute defined in security-config.xml
  - object: the current object in memory
  - Returns true if current user has the permission
- Examples:
  - perm.Account.edit(account)
  - perm.Submission.view(submission)

Source : Guidewire Educational Platform



Source : Guidewire Educational Platform

Lesson objectives review

You are now able to:

- Configure system permissions
- Write Gosu expressions that evaluate user permissions
- Describe static and object-based permissions

Source : Guidewire Educational Platform

Capgemini