ORACLE

Product Management Security Implementation

Two State Security Model

Steve Cascio
Senior Principal Product Manager
Supply Chain Management, Product Management
April 29, 2021

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

Security Overview

Two State Data Security Model Overview

Initial Startup of Product Management

Two State Data Model Behavior

Public Item Classes With Private Items

Why the Two State Data Security Model

Use Cases

Common Issues With Security

Two State Data Security Project

Security Overview

Product Management provides a framework to secure the customer's applications and data.

The components of the security infrastructure work together to delivery a secured Oracle SCM Cloud

The security framework secures the user interface as well data returned to the it

Functional Security: The control of access to a page or a specific use of a page. Function security controls what a user can do. Functional security is managed by adding pre-defined job roles, duty roles and privileges to user.

Data Security: The control of access and action a user can take against which data. Data security is managed by creating data policies that define data access for user and groups. Two types of data security policies are supported:

- **Instance Set Data Grants:** A single data policy applies to a set, defined by context of instance of items
- **Instance Data Grant:** A single data policy applies to a single instance of item

Two State Data Security: Data in Product Management can have two states; public where data is accessible to all users and privilege where data is restricted to set of user or individual object instances.

Two State Data Security has been implemented with Item, Catalogs, Change Orders and Trading Partner Items

Why Do We Have Two State Data Security Model?

Most Customers wanted to promote reuse of their content, but also have cases where access to certain content must be controlled for either for business requirement or be controlled for a portion of the lifecycle of the product

Controlled access is often optional and not required for all content

Some content needs to be secured for portion of its lifecycle, for example; new product release where a limited set of user have access until the release date

Business needs may require data to completely secured, this is what the Private state at an item class level does today

Security is easy to manage when you dealing with exceptions to public access

Control access for external users such as contractors, or suppliers

Two State Data Security

Private Data State

Prior to Product Management Release 19A all data had to be secured with data policies

Initial system start required the customer to define data policies for all user that will view or edit data in the system

Job roles and duty roles were defined to control groups of user with similar access requirements

To manage common access to data, large number of users would have to be assigned and managed for these roles

For example;

- A duty was used to define the group of internal users with read only access to all data for a class of items to promote reuse and sharing. The user could search of this data and view it.
- A duty was used to define the group of external user that had read only access to select data. The external user could search and view the data

Manage of these groups was difficult for large organization

The item class structure was also used to control access for these groups

All data policies are inherited by all items within item class, instance level data policies just extend the inherited data policies, but do not override the inherited data policies

Two State Data Security

Public Data State

In Release 19A we added the public state to data security, initially all data created was public, that is requiring no data policies for users or group to access the data

The **Public** flag was added to the user interface and data tables to indicate the state of data security

The Public flag set to True at the item class level indicated all items created for this item class are accessible, two exceptions:

- Specific set of items within the item class could be private
- User definable attributes access can be controlled through the creation of data policies

Child item classes would inherit the valued of the Public flag

The Root Item Class was set to the Public state and can't be changed. The customer class hierarchy is always created as child of the Root Item Class

Rules are enforced to control inheritance of the Public flag:

- An item class with a public flag set to True can have child item classes that are either public or private
- An item class with public flag set to False can only have child item classes that are private

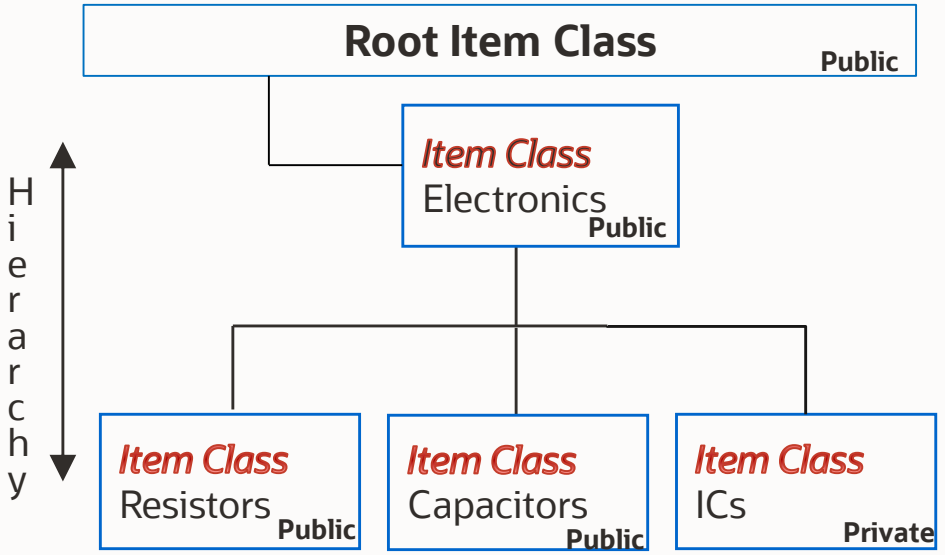
Initial Start Up

New Customers with Release 19A and forward will see the Root Item Class is public

Each item class created under the Root Item Class is public, until made private

Behavior Rules:

- All items created in a public item class are public by default
- All item created in a private item class are private by default
- A public item class can have both private and public child item classes
- A private item class can only have private child item classes
- A public item class can have both public and private items



Public Item Class With Private Item

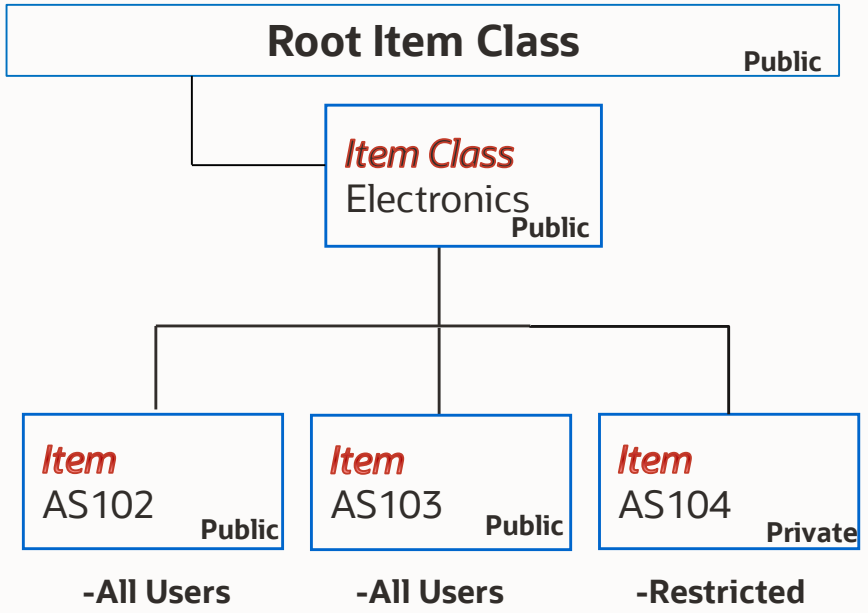
Public Item Class can have both private and public items

An item in public item class can be changed from public to private

Behavior:

- Person changing the item to private becomes the owner with full actions, an instance type data policy is created
- The Owner can delegate authority to add additional users or groups by creating additional data policies
- The Owner can manage data policies for the item
- Other users will not be able to search for the private item without a data policy granting them access

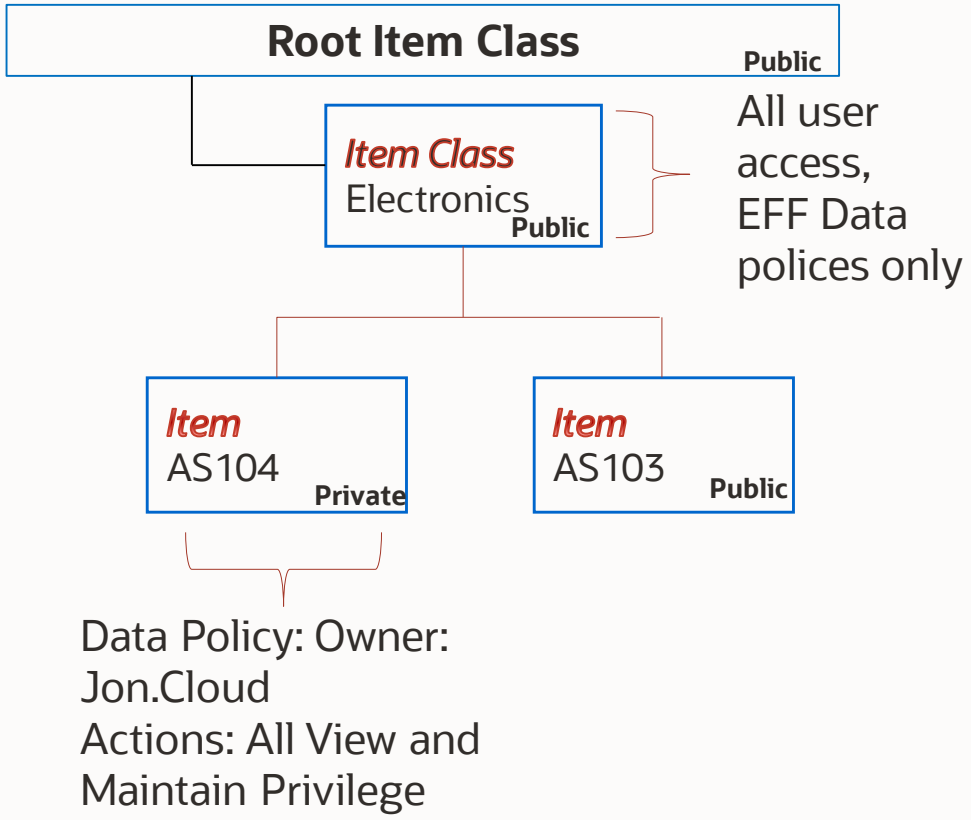
Automation can be used with this process



What Happens When a Public Item Is Made Private?

The user edit the item and launches the Manage Data Security dialog and unchecks the Public flag;

- A data policy is created for the user as the owner of the item
- The owner is given a full set of actions both view and maintain for the item, at this time the owner is the only one with access to the item
- The inherited EFF actions are still available on the private item
- The owner can grant additional data policies to allow other access to the item
- Optionally, an automated program can be run to determine the access policies that are need and the data policies can be created using the Product Management Data Security REST API



Controlling View and Edit of Item Attributes in Public Item Classes

All public items in a public item class are accessible by all users without a data policy

Functional privileges can be granted to users, job roles or duty roles to control if the user can view and/or edit

The **View** functional privilege:

Will allow searching for the item

Will display data in the user interface in a read only mode

The **Manage** functional privilege:

Will display data in the user interface in an editable mode

User Definable Attributes (Extensible Flexfields)

User definable Attributes in public items as well as private items can be controlled by data policies using custom data actions created for the attribute group

When a public item, is made private the data polices for user definable attribute will be inherited by private item

The owner can edit the data policy at the instance level

DEMO

ITEM SECURITY AT THE ITEM CLASS LEVEL

ITEM SECURITY AT THE ITEM LEVEL

Use Case: New Product Introduction

Use Case: A new product is to introduced and only a limited number of user will have access to the product record as it is being prepared for product release. The product is new model of an existing line of product.

Solution: The product record is created in an existing public item class and made private. The owner creates a set of data policies for the item allowing additional members access to the product. After the product is released, the product accessibility is opened to additional internal users.

Use Case: Contract Manufacturer Access To Product

Use Case: A Contractor is working on a specific product that has a structure. The Contractor needs access to the product content as well as an internal owner of the product. The Customer does not open up all similar products to the Contractor.

Solution: The product is created and made private by the product owner, the owner adds additional data policies to give the Contractor access to the product content and structure. Other Contractors working for the Customer do not have access to the product content. After completion of the project, the data polices for the Contractor are removed by the internal owner.

Product Data Security Automation

Use Case: the data access to product data is determined by set of data related to the product, for example; the product maybe specific to a product line, an organization, product type, or region where the product is being used.

When the product is created, an automated program is run to determine what data access different groups within the company will get.

Solution: the product is created by internal users which triggers an event that runs a program to analyze the new product and determine the data access that are needed for the product. The program uses the Product Management Data Securities REST API to make the product private and to create data policies for the product across different groups and organizations.

Product Management Data Securities REST API

The item object has multiple tables that must be secured for data security

Multiple data polices are created in the central tables, but the user interface simplifies this process by displaying a single data policy that hides the complexity of data in central tables for items

The REST API provides actions to manage data security for items, change orders and trading partner items

The REST API is being used for security automation

HTTP Action	Action Type	Action Description
GET	Collection	Query for data grant
POST	Collection	Create data grant
PATCH	Collection	Update data grant
DELETE	Collection	Delete data grant
SECURE AN OBJECT	Collection	Transition State to Private

Create a New Item Security Grant Using Rest API

Post Operation

curl -u username:password -X POST -H "Content-Type:application/vnd.oracle.adf.resourceitem+json" -d 'request payload'
"https://servername/fscmRestApi/resources/version/productManagementDataSecurities" -H 'cache-control: no-cache'

Example Request

```
{
        "ObjectName":"Item",
        "Principal":"Person",
        "OrganizationCode":"V1",
        "ItemNumber":" AS1234",
        "Name":" Jane.Powel ",
        "Actions": "View Item Attribute | View Item Basic | View Item Pack | View Item Structure",
        "ItemEFFTranslationActions": "",
        "ItemRevisionEFFActions": "",
        "ItemRevisionEFFTranslationActions": "",
        "ItemSupplierEFFActions": "",
        "ItemSupplierEFFTranslationActions": ""

}
```

Common Security Issues In Our Customer Base

Most Customers wanted reuse of their content, but also have cases where access to certain content must be controlled for either a business case or for a portion of the lifecycle of the product

Customer found it difficult to model security for both internal and external user, this often led to multiple item classes under a common parent item class to allow for different data security policies for both set of users

Customers found that items were available to most or all internal users and were restricted to external users or groups within the company

Items were restricted for period of their lifecycle and then accessible to everyone, security changes over time

Maintaining duty roles or job roles for large groups of user was difficult

Controlling access to items where users belong to multiple groups was difficult

Modeling security by exception was emerging as preferred method, for example; the public state does not require data policies to be entered, only add data security policies to restrict access and control EFF access

Two State Data Security Model Project

Phase 1 : Benefit New Customer Data Security *-Complete*

- Flip the start up model from all Private to all Public
- Implement consistent rules for Private-Public behavior
- Create Private Item in Public Item Class model with automatic ownership data policies
- Root Item Class is always Public
- Control Creation of items at Root Item Class
- Transition all items in a public item class to private when the item class is made private
- Rest API support for public to private transition of items

Phase 2: Benefit Existing Customers Data Security – *Analysis Phase*

- Private Item to Public Item Transition, within Public Item Class
- Private Parent Item Class and child Item Classes transition to Public

Thank you


ORACLE