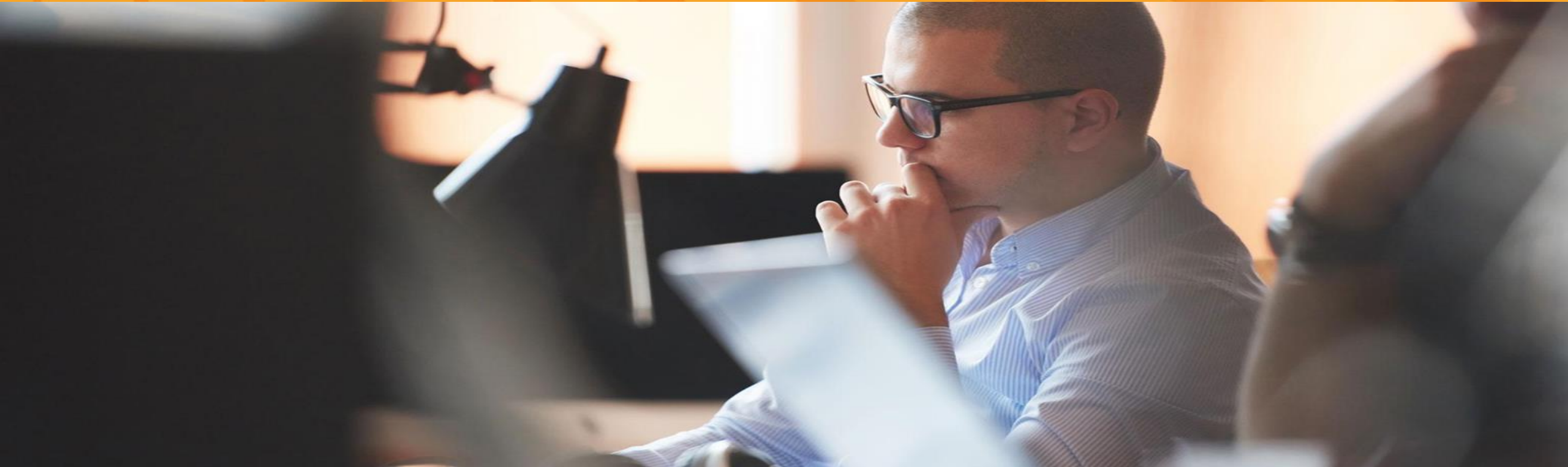


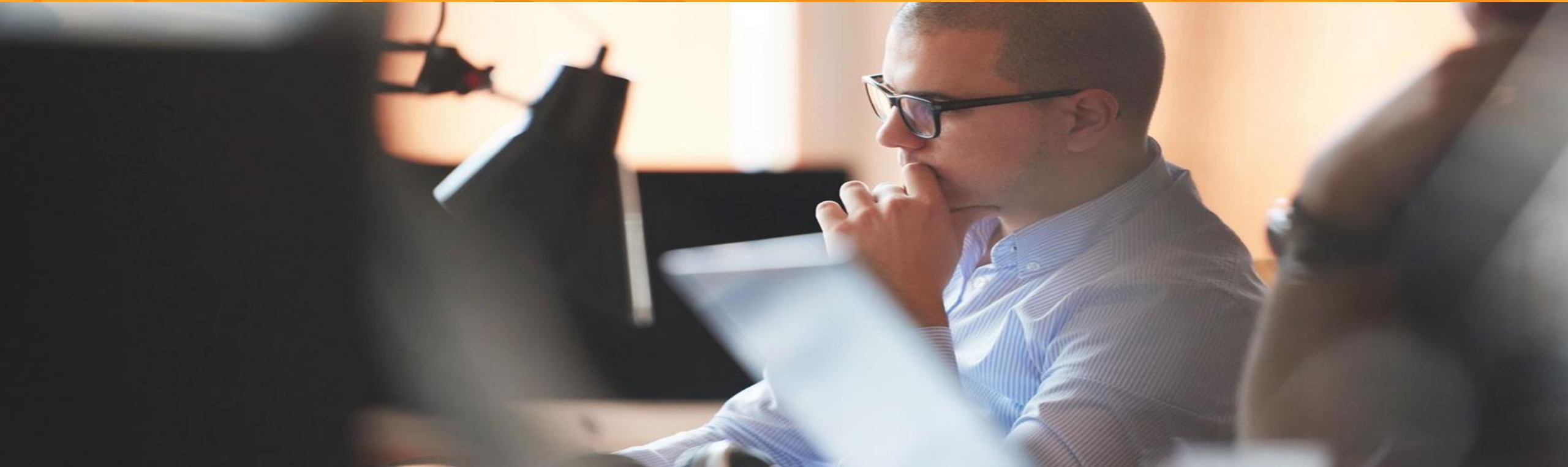


SERVICENOW OVERVIEW



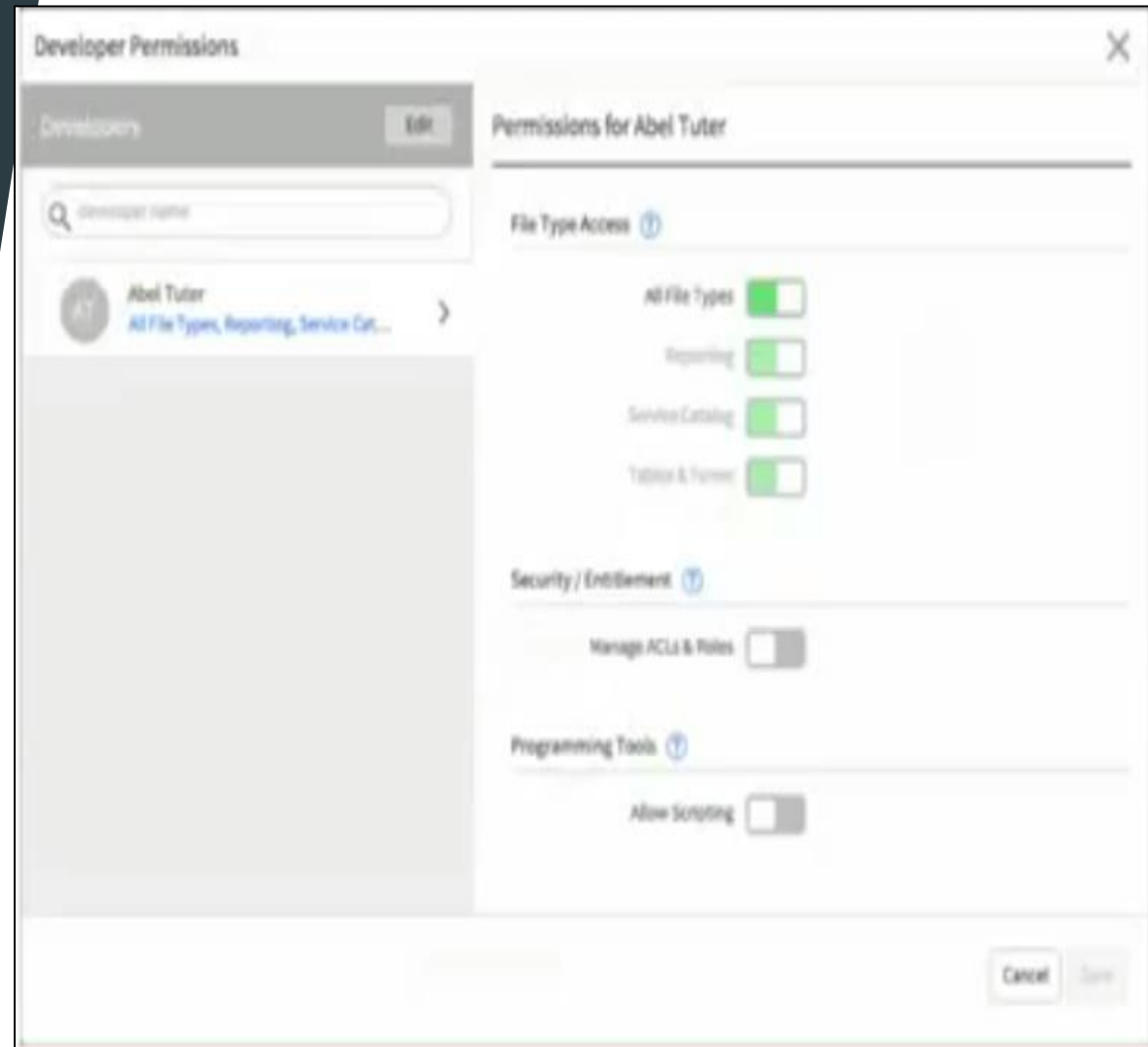


ServiceNow – Webservices



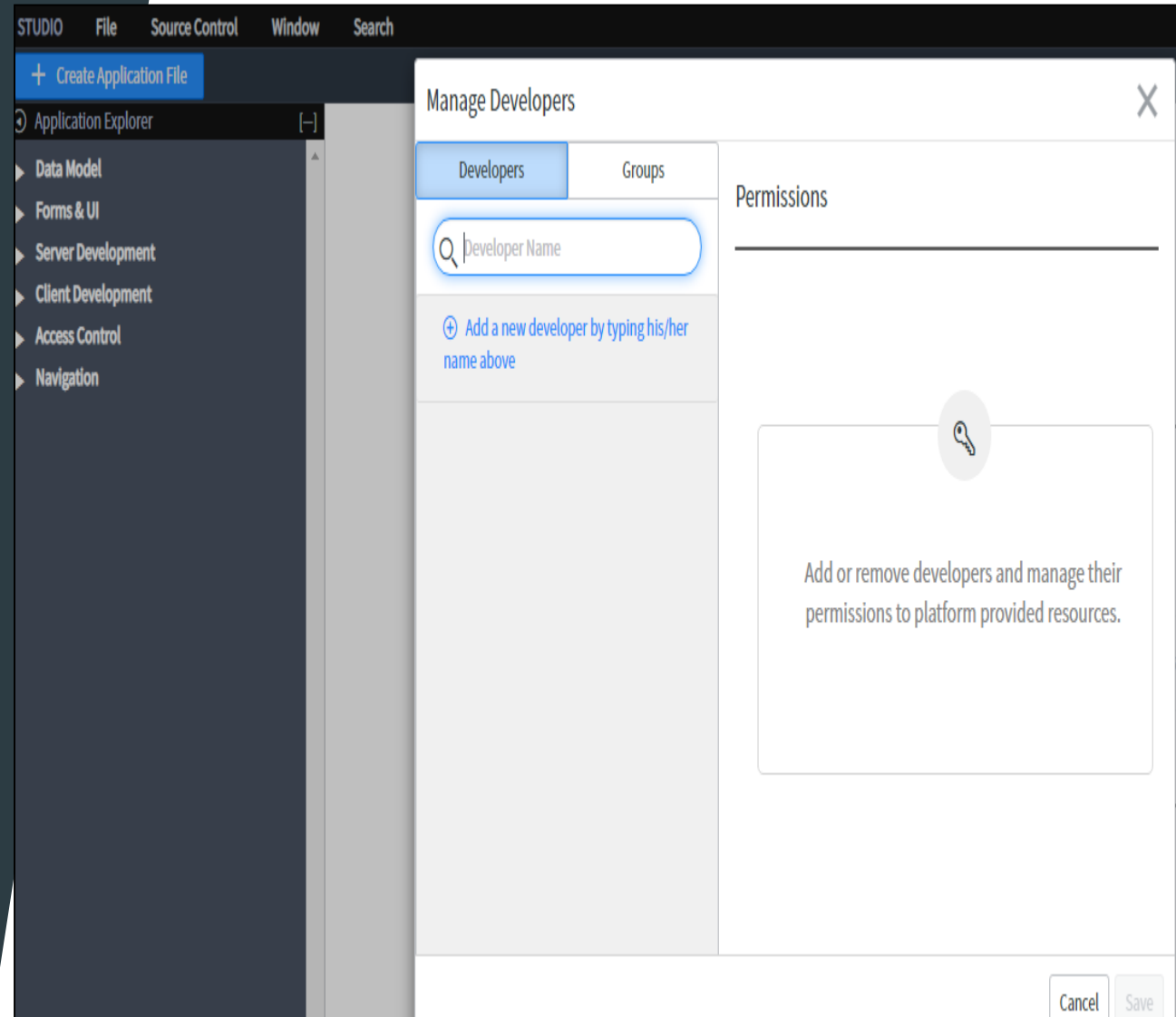
Delegated Development

- ▶ Delegated development entitles non development administrators to develop applications on ServiceNow platform
- ▶ Administrators can use Studio to manage development users and application content they can access.
- ▶ Delegated role basically is providing access to application to users for development activities
- ▶ Role : delegated_developer,roles starting with sn_dd
- ▶ Only admin can give delegated role and delegated developer cannot add admin



Delegated Development

- ▶ Navigate to Studio and Load Needit application
- ▶ Go to File→Manage Developers



Delegated Development

- ▶ Search for Ted Bozelle and assign him File Type Access → Tables & Forms.
- ▶ He is IT manager but has not written a piece of code and would not want him to mess around with business rules, Script includes or client scripts. That's the permission he would deserve.
- ▶ Save it

The screenshot shows the 'Manage Developers' window with the 'Developers' tab selected. A search bar contains 'ted'. Below the search bar, a list shows 'Ted Bozelle' with a circular profile icon containing 'TB' and the role 'Tables & Forms' in blue text. The right pane is titled 'Permissions for Ted Bozelle' and contains three sections of toggle switches:

- File Type Access** (with a help icon):
 - All File Types: ☐
 - Reporting: ☐
 - Service Catalog: ☐
 - Flow Designer: ☐
 - Integrations: ☐
 - Workflow: ☐
 - Service Portal: ☐
 - Tables & Forms: ☒
- Security / Entitlement** (with a help icon):
 - Manage ACLs & Roles: ☐
- Programming Tools** (with a help icon):
 - Allow Scripting: ☐

At the bottom right are 'Cancel' and 'Save' buttons.

Delegated Development

- ▶ Navigating to users ,putting Ted and making him favourites.
- ▶ We can observe roles being assigned to the particular user
- ▶ If we try to impersonate we will not be able get scoped application access data i.e he cannot test his changes done to the record as he would not have read access.Hence we need to provide him another role.

The screenshot displays the ServiceNow 'Service Management' interface. The top navigation bar includes the 'service now' logo, a 'Need it' dropdown, and a user profile for 'System Administrator'. The left sidebar shows a 'Filter navigator' and a list of items: 'Home', 'System Security - Users', and 'ted.bozelle'. The main content area shows the user 'User Ted Bozelle' with 'Update' and 'Delete' buttons. Below this, there are 'Related Links' for 'View Subscriptions' and 'Reset a password'. A tabbed interface shows 'Roles (2)' selected, with sub-tabs for 'Groups', 'Delegates', 'Subscriptions', and 'Manage Subscriptions'. A table lists the roles assigned to the user:

Role	State	Inherited	Inheritance Count
delegated_developer	Active	false	
sn_dd_needit_tableforms	Active	false	

At the bottom, there is an 'Actions on selected rows...' dropdown and pagination controls showing '1 to 2 of 2'.

Delegated Development

- ▶ Hence grant him role to app user itself to give application and data access

The screenshot displays the 'Edit Members' dialog box. At the top, there are 'Cancel' and 'Save' buttons. Below them are 'Add Filter' and 'Run filter' buttons, with a help icon. A filter configuration row contains three dropdowns: '-- choose field --', '-- oper --', and '-- value --'. The main area is divided into two panels. The left panel, titled 'Collection', has a search bar with 'x' and a list of items: 'snc_read_only' (highlighted), 'x_333305_innovate.innovate_user', and 'x_58872_needit.needit_user'. The right panel, titled 'Roles List', shows 'Ted Bozelle' and a list of roles: 'delegated_developer' and 'sn_dd_needit_tablesforms'. Between the panels are right and left arrow buttons.

Delegated Development

- ▶ Thus the user will have 2 delegated developer roles and application & data access role.

Roles (3)	Groups	Delegates	Subscriptions	Manage Subscriptions
Roles	Edit...	Go to	Role	Search
User = Ted Bozelle				
		Role	State	Inherited
		Inheritance Count		
<input type="checkbox"/>		x 58872 needit.needit user	Active	false
<input type="checkbox"/>		delegated developer	Active	false
<input type="checkbox"/>		sn dd needit tablesforms	Active	false
<input type="checkbox"/>	Actions on selected rows...			

Delegated Development

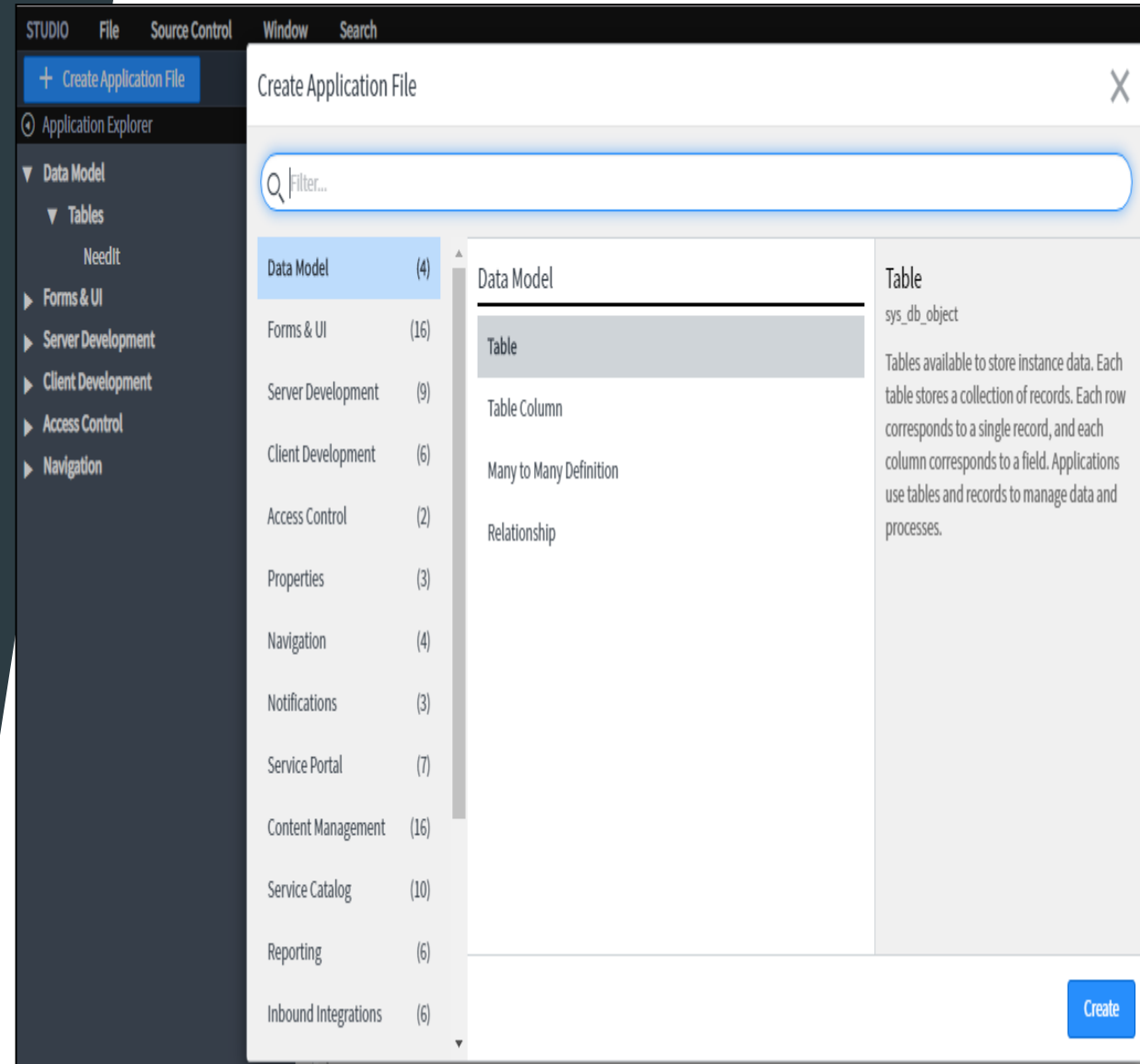
- ▶ Now impersonate user to Ted.
- ▶ Ted now has access to Needit application where he can read records.
- ▶ He would also get access to Studio where he has access to only that application.

The screenshot shows the ServiceNow 'Needit' application form. The left sidebar contains a 'Filter navigator' and a list of applications: Demo table, Flow Designer, Guided Setup, Innovate, Needit (selected), Create New, All, and Open. The main form area displays fields for Number (NI002001), Priority (4 - Low), State (Requested), Requested for (System Administrator), Requested for email (admin@example.com), Request type (Legal), What needed (Legal 1), When needed, Short description, and Description. A 'Submit' button is visible in the top right corner.

The screenshot shows the ServiceNow 'Studio' application form. The left sidebar contains a 'Filter navigator' and a list of applications: Studio (selected), System Applications, and Studio. The main form area displays fields for Number (NI002001), Priority (4 - Low), State (Requested), Requested for (System Administrator), Requested for email (admin@example.com), Request type (Legal), What needed (Legal 1), When needed, Short description, and Description. A 'Submit' button is visible in the top right corner.

Delegated Development

- ▶ Ted can now create a new Table and let's name it



Delegated Development

- ▶ Label: Inspection code
- ▶ Application scope is created .Copy it

The screenshot shows the 'New record' form for the 'Inspection code' table in a Microsoft Dynamics 365 application. The form is titled 'NeedIt Table' and 'Inspection code Table'. It includes a 'Table' tab and a 'New record' button. A blue information banner at the top states: 'A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. [More Info](#)'. The form contains several fields and checkboxes:




- * Label:** Inspection code
- * Name:** x_58872_needit_inspection_code (highlighted with a blue border)
- Extends table:** (empty field with a search icon)
- Application:** NeedIt
- Create module:** ☒
- Create mobile module:** ☒
- Add module to menu:** -- Create new --
- New menu name:** Inspection code

At the bottom, there are tabs for 'Columns', 'Controls', and 'Application Access'. The 'Columns' tab is selected, showing a 'Table Columns' section with a search bar and a 'Search' button.

Delegated Development

- ▶ Create 3 simple fields on table Code,Description and Active.
- ▶ Code : String ,Display :True
Description
Active : True/False,Default value:True
- ▶ Submit it

Table
New record



Submit

Cancel

Columns

Controls

Application Access








Table Columns

Search

for text

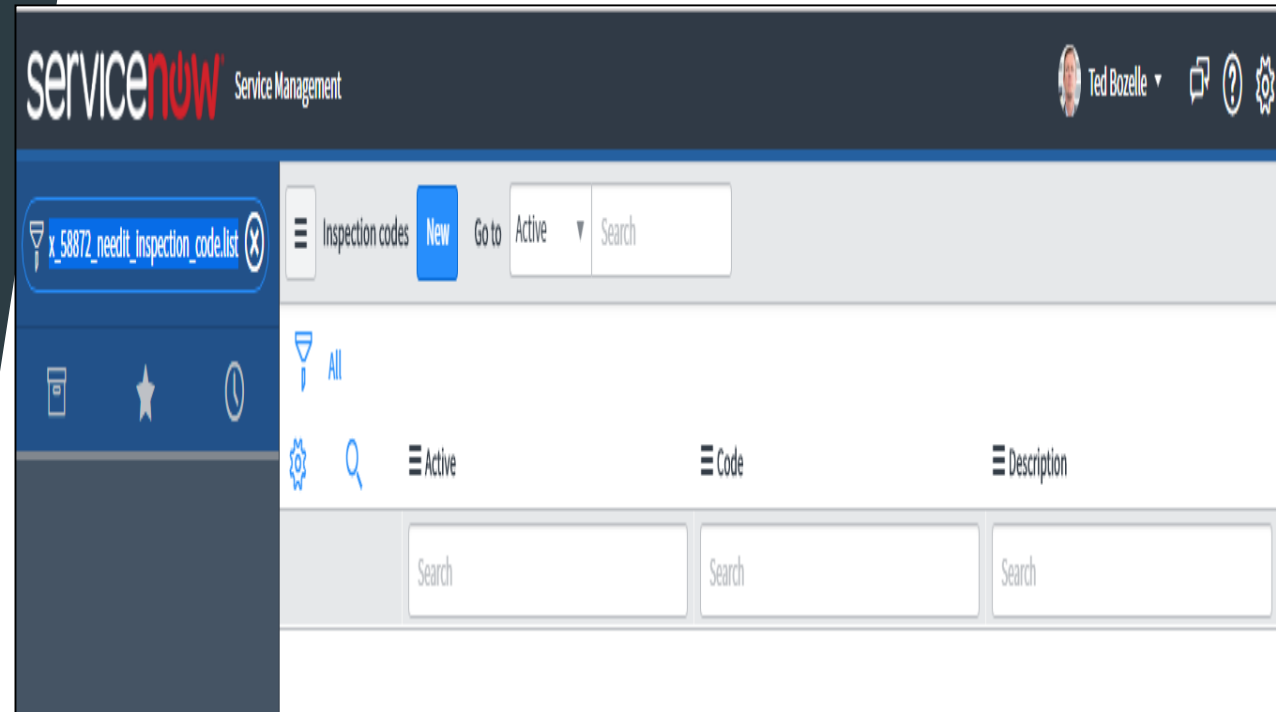
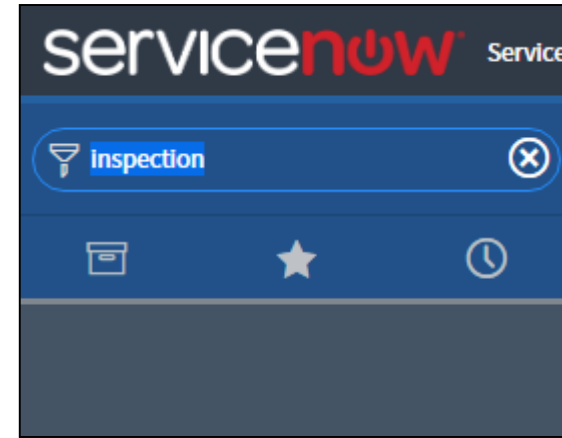
Search

Dictionary Entries

	Column label	Type	Reference	Max length	Default value	Display
 	Code	String				true
 	Description					false
 	Active	True/False			True	false
	Insert a new row...					

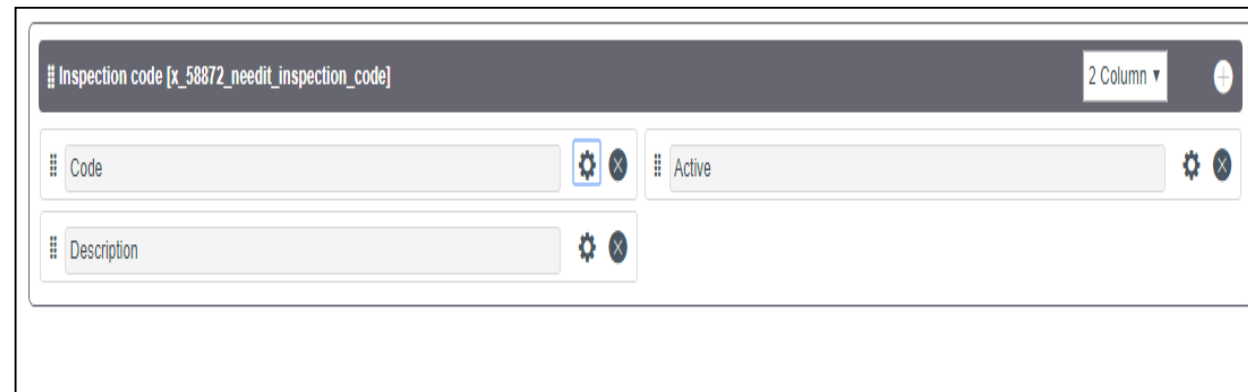
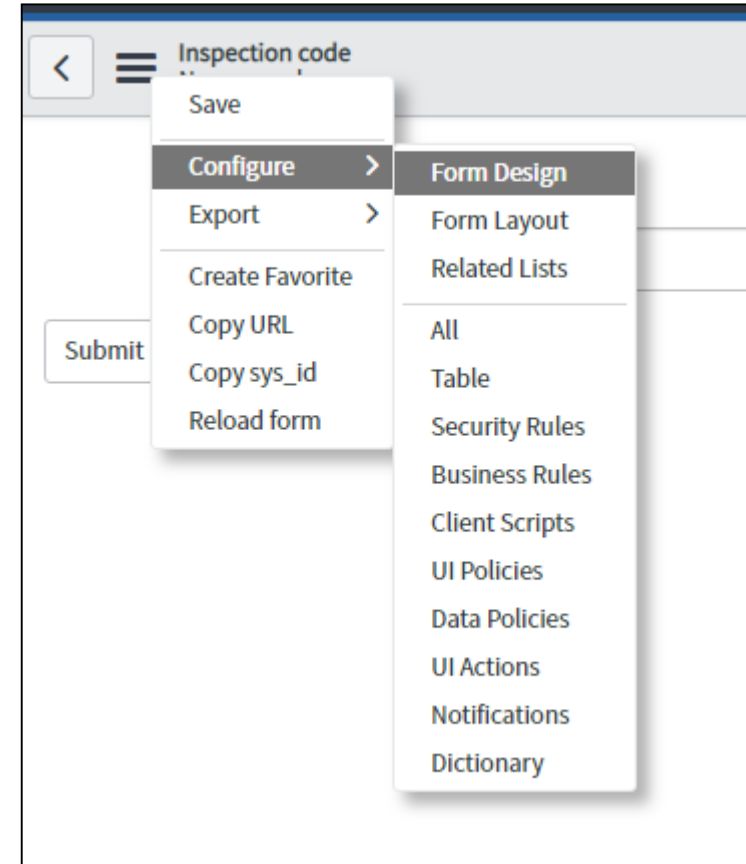
Delegated Development

- ▶ If we observe Ted would not have access to view the table from Navigator.
- ▶ Now go to `x_58872_needit_inspection_code.list`
- ▶ Create new Inspection code



Delegated Development




- ▶ Configure the form design
- ▶ Rearrange the fields and make Code mandatory field



Delegated Development

- ▶ In the configured form add records

< ≡ Inspection code
New record



Submit

* Code

OK




Active ☒

Description

Everything is awesome

Submit

< ≡ Inspection code
New record



Submit

* Code

Fail




Active ☒

Description

Please try again

Submit

< ≡ Inspection code
New record



Submit

* Code

Unknown

Active ☐

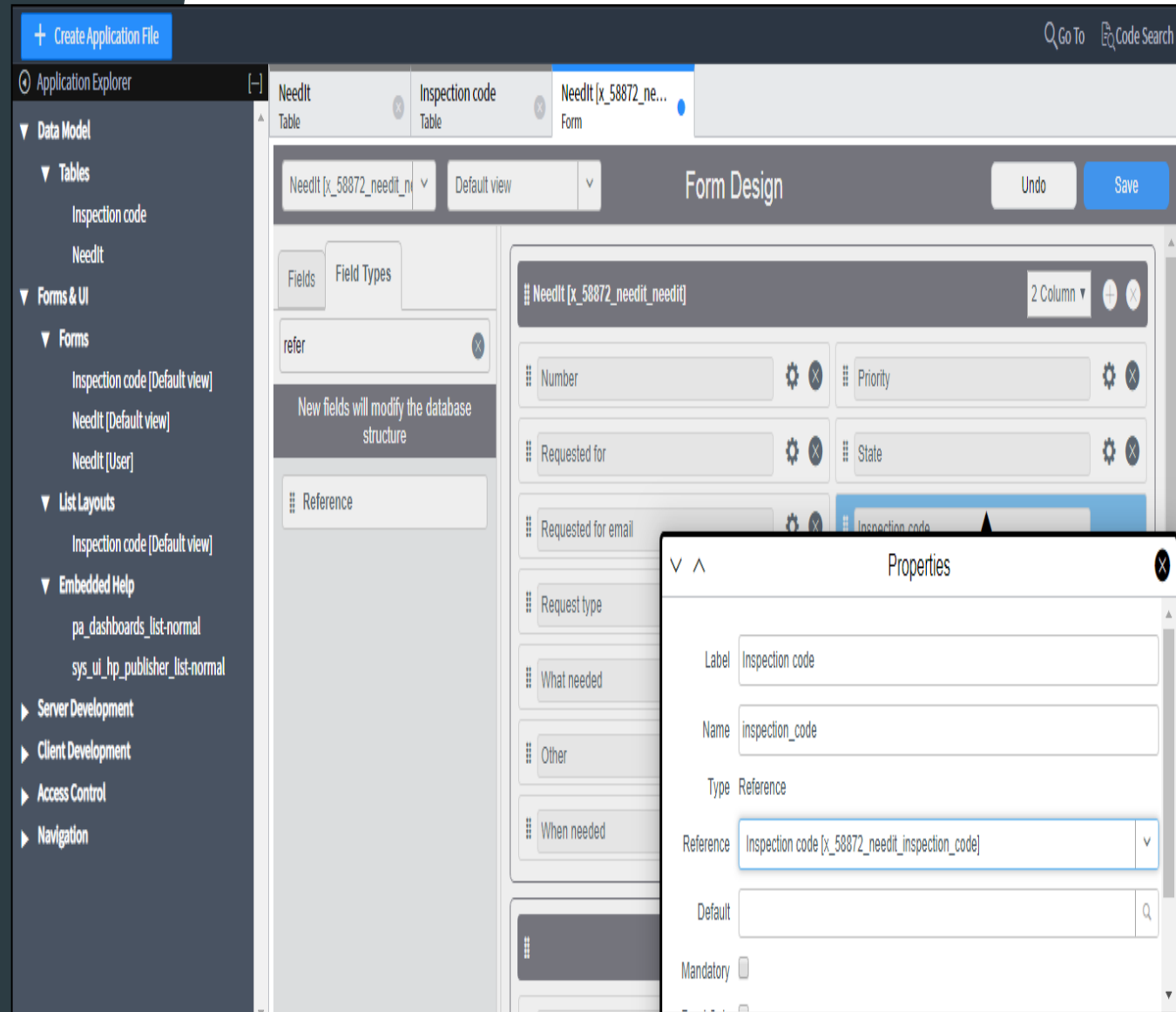
Description

Do not use

Submit

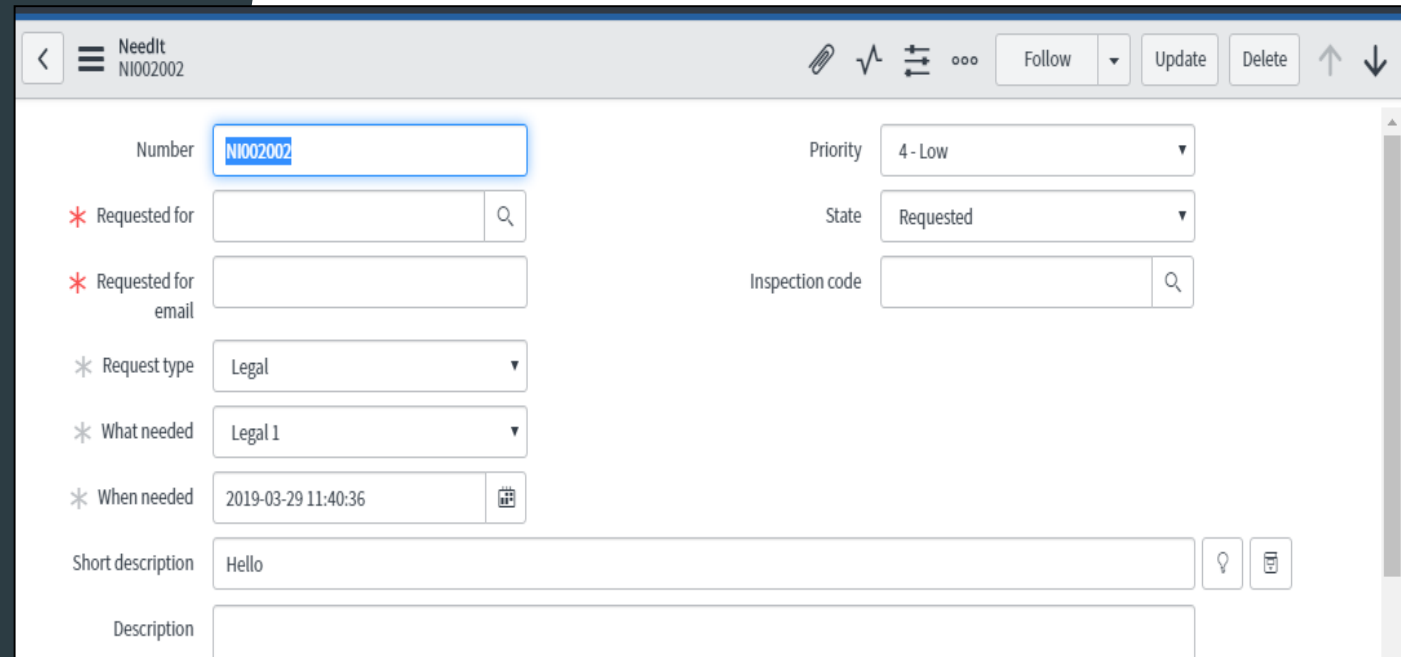
Delegated Development

- ▶ Now lets reference the field from the Needit form
- ▶ In studio navigate to Forms → Needit [Default view]
- ▶ Create a reference field with
Label:inspection code
Name:inspection_code
Reference : Inspection code
- ▶ Save the Form

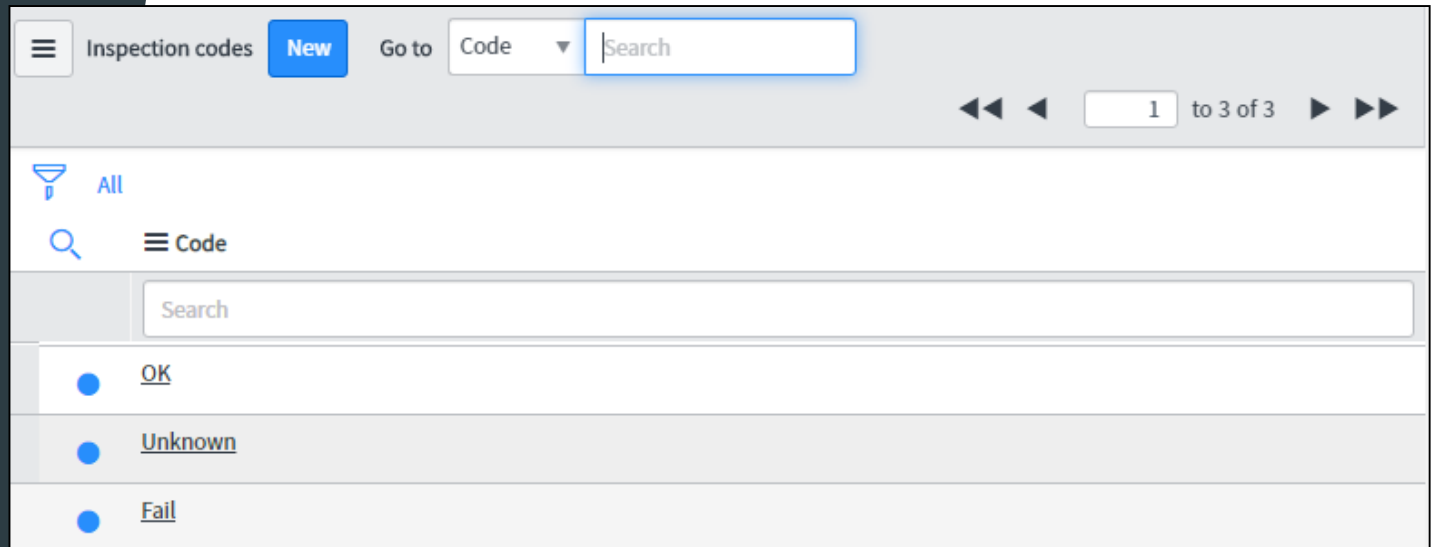


Delegated Development

- ▶ Reload the needit form and we can observe Inspection code field referenced.
- ▶ We see all fields referenced
- ▶ We will now include reference qualifier



The screenshot shows a web form titled "Needit NI002002". The form contains several fields: "Number" (text input with "NI002002"), "Priority" (dropdown menu with "4 - Low"), "State" (dropdown menu with "Requested"), "Inspection code" (text input with a search icon), "Requested for" (text input with a search icon), "Requested for email" (text input), "Request type" (dropdown menu with "Legal"), "What needed" (dropdown menu with "Legal 1"), "When needed" (text input with "2019-03-29 11:40:36" and a calendar icon), "Short description" (text input with "Hello" and a lightbulb icon), and "Description" (text input). The form has a header bar with navigation icons and buttons for "Follow", "Update", and "Delete".



The screenshot shows a web interface for "Inspection codes". It features a header bar with a "New" button, a "Go to" dropdown menu with "Code" selected, and a search input field. Below the header, there is a filter icon and the text "All". A search icon and the text "Code" are also present. A search input field is located below these elements. The main content area displays a list of items, each with a blue dot and a label: "OK", "Unknown", and "Fail". The list is paginated, showing "1 to 3 of 3" items.

Delegated Development

- ▶ Go back to Studio→Tables→Needit and add reference specification to field inspection code
- ▶ Add a filter in the dictionary entry of Inspection code with Active → True
- ▶ Update the dictionary entry.

Table Needit

Columns Controls Application Access

Table Columns New Go to Column label Search 1 to 36 of 36

Dictionary Entries > GOTOcolumn_label=>inspection

Column label	Type	Reference	Max length	Default value	Display
Inspection code	Reference	Inspection code	32	false	false

Inspection code Table Needit Table Inspection code Table Column

Dictionary Entry Inspection code

Update Delete Column

Reference Specification Choice List Specification Default Value

The Reference field specifies what table this field displays values from.

* Reference Inspection code

Reference qual condition

Add Filter Condition Add "OR" Clause

Active is true AND OR X

Delegated Development

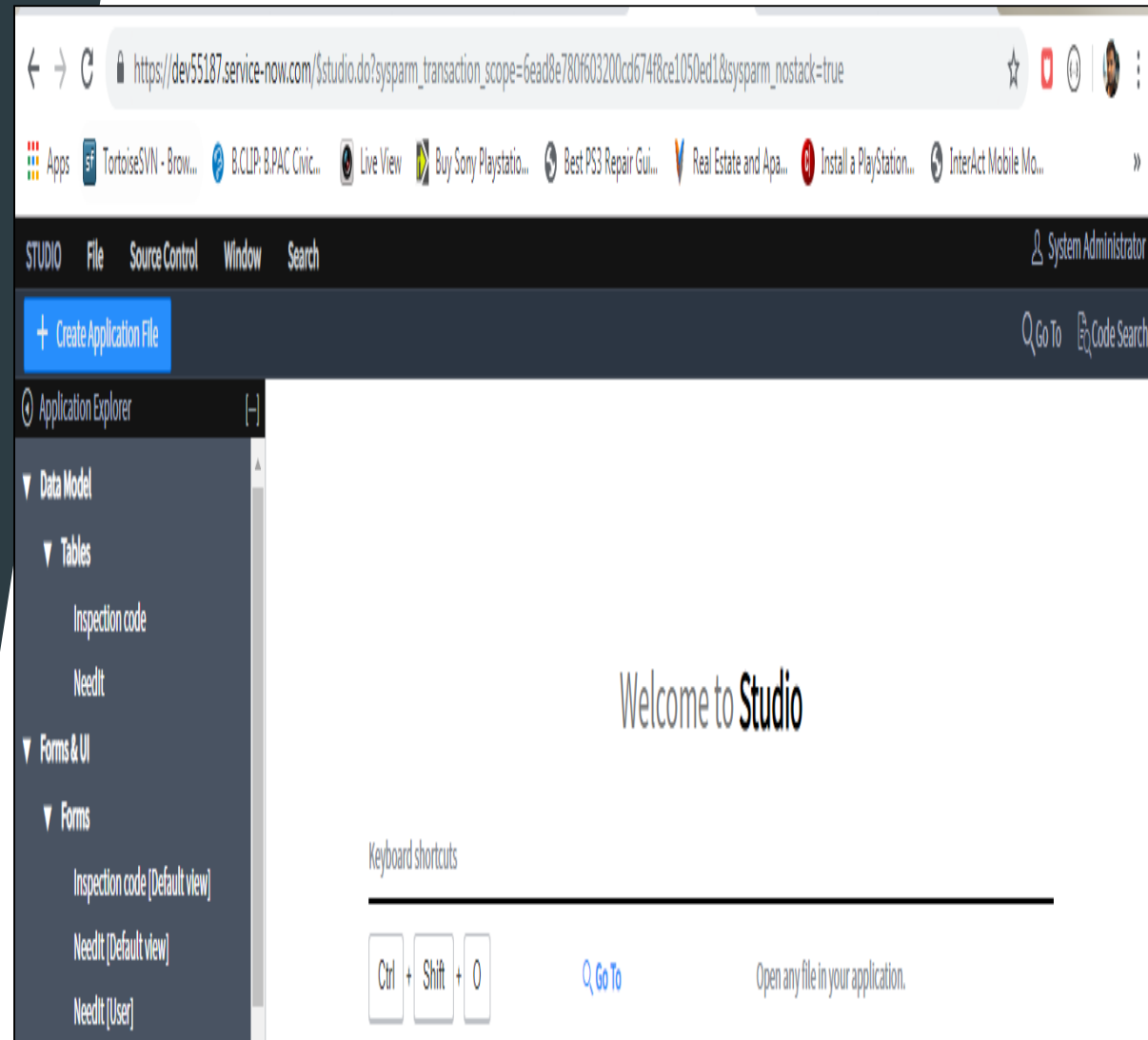
- ▶ Now reload the form we will be able to see Inspection code for active records only.
- ▶ Hence Delegated developer would be given role of field access and development in particular application

The image shows a ServiceNow interface for a 'Needit' record (NI002002). The form includes fields for Number, Priority (4 - Low), State (Requested), Requested for, Requested for email, Request type (Legal), What needed (Legal 1), When needed (2019-03-29 11:40:36), Short description (Hello), and Description. At the bottom, there are tabs for Log, Statistics, and Work notes.

An 'Inspection codes' pop-up window is overlaid on the form. It shows a list of inspection codes with a search bar and a 'Go to' dropdown. The list displays 'OK' and 'Fail' status indicators. The URL in the browser is https://dev55187.service-now.com/x_58872_needit_inspec...

Delegated Development

- ▶ We can go back and login as admin and check fields created as part of delegated developer access we will be able to see all the tables and forms having inspection code reference.
- ▶ Hence we can give prescriptive access to delegated developers like just modifying forms and fields without giving them access to scripting



ITOM

- ▶ **ITOM -IT operations management provides visibility to Business services and resources allowing to manage Operations estate and proactively eliminate service outages.**
- ▶ **Modules/Applications include CMDB, Discovery,MID SERVER,CLOUD management,Orchestration, LDAP,Event Management,Service Mapping and Operational Intelligence**



Discovery



Operational Intelligence



Event Management



Service Mapping



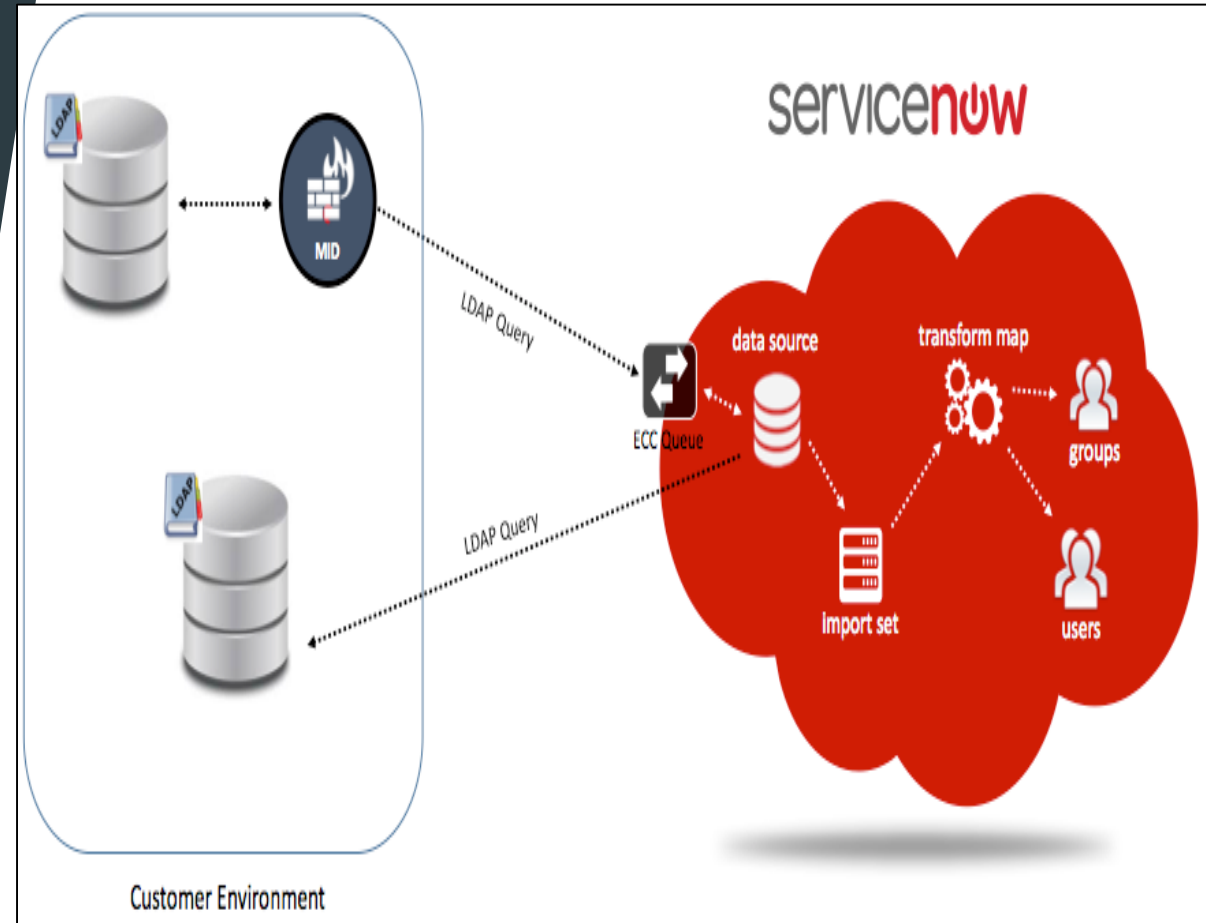
Orchestration



Cloud Management

ITOM

- ▶ **Midservers(MID - Management,Instrumentation and Discovery are setup for communication to move data between external systems and ServiceNow ITOM applications.**
- ▶ **Midservers are java applications that run on Enterprise network behind a firewall or on the cloud(AWS,Azure).**
- ▶ **Midservers help setup and run ITOM applications**



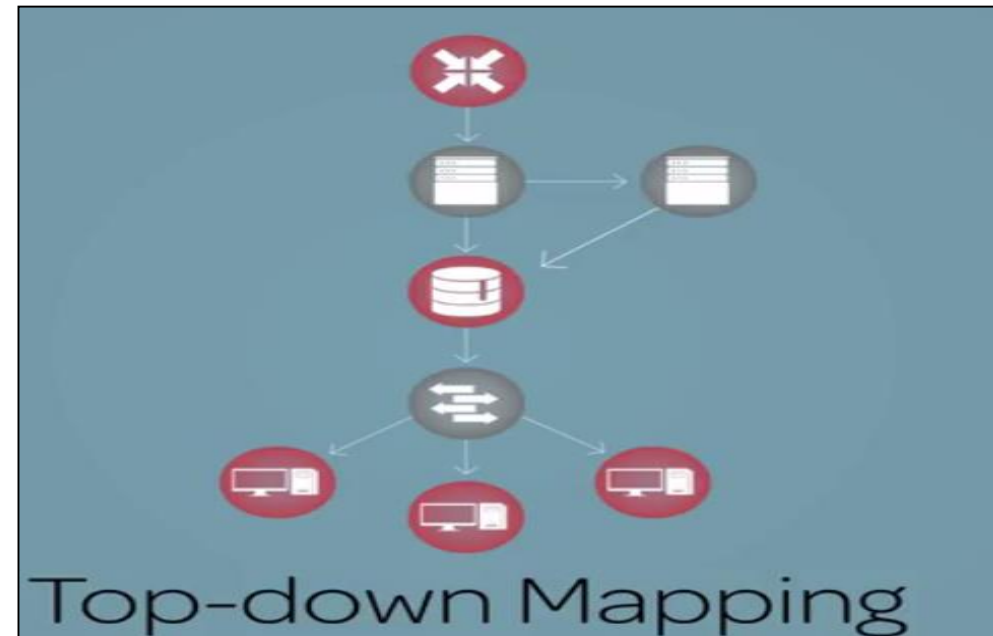
ITOM

- ▶ **Discovery** - It finds applications and devices on the network and important attributes of those devices such as OS, Software, Memory and others and updates.
- ▶ It then updates **Configuration Management Database** with the information it finds.



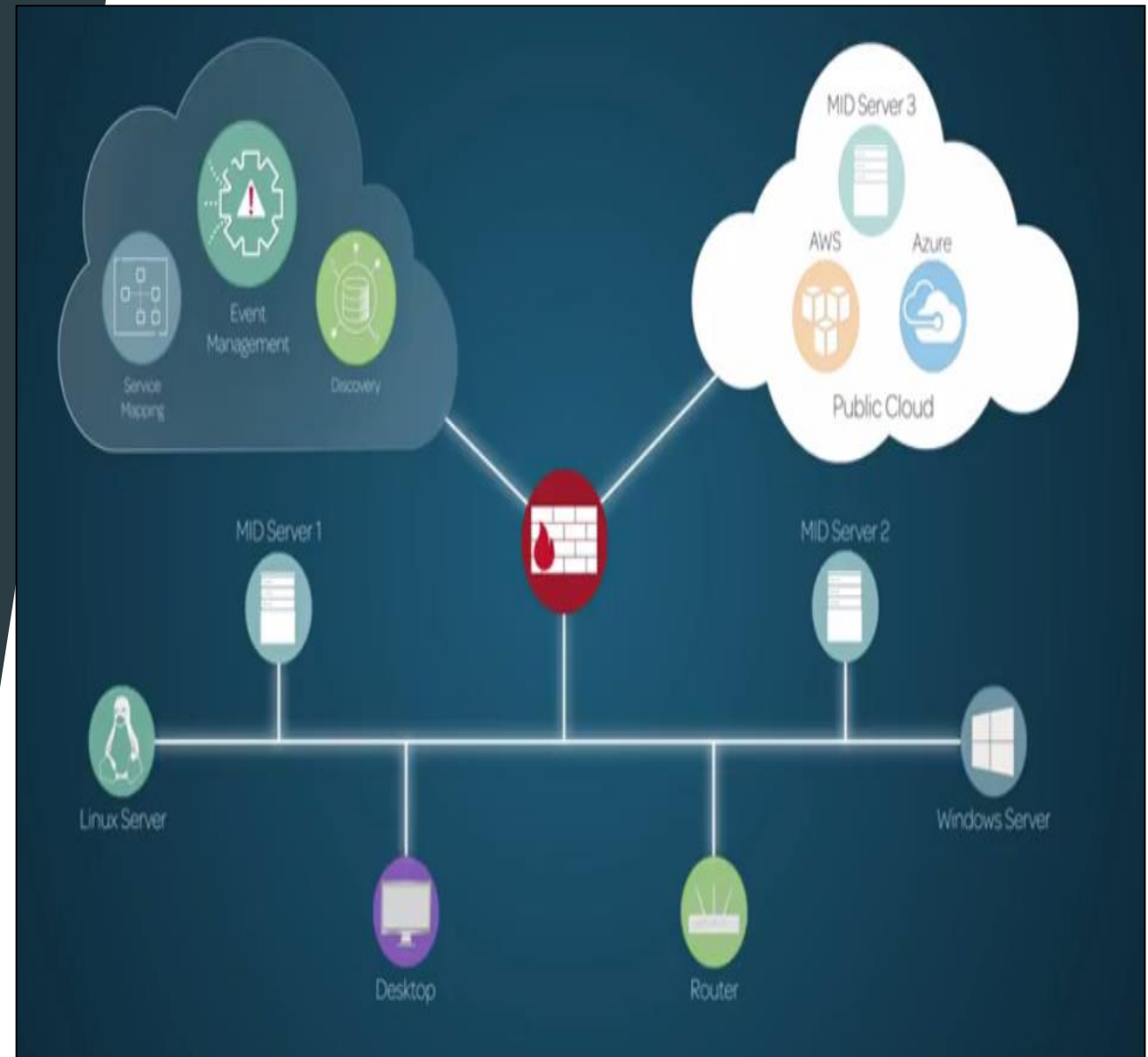
ITOM

- ▶ Discovery provides Horizontal discovery as it finds devices on the network but does not draw relationship between configuration item (Cis) that are part of specific business service.
- ▶ Service Mapping provides complementary capability where it uses Top Down mapping to map CI that are part of business services such as Email service. It helps visualize all Cis used to deliver a service.



ITOM

- ▶ **Event Management** - If anything goes wrong with Cis associated it is captured as Events which are basically notification araised in the IT system like failure or warning.It collects and analyzes the events to monitor the health of IT system and helps anticipate and prevent outages and resolve issues quickly with minimal impact on the operations



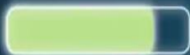
ITOM

- ▶ Cloud Management - Using cloud mgt we can create and import blueprints for provisioning cloud infrastructure and resources from any Cloud service provider which users can request from catalog items via the Cloud user portal.
- ▶ Users can also manage lifecycle of the resources on Cloud service provider



ITOM

- ▶ From the ITOM portal we can also track utilization and track costs to enforce usage limitation on available resources as well as setup capacity limits on the data centers and usage quota for users and groups



Thankyou