* List View vs Form View
* Related Records
* UI Actions & Buttons
* Up or down arrow to move to the next record from the list query
  + Wild card search’s in the List
  + Wildcard Syntax Search Criteria Example
  + \*[term] contains \*Mark
  + !\*[term] does not contain !\*Mark
  + =[term] equals =Miller
  + !=[term] does not equal !=Miller
  + [term]% starts with Hello%
  + %[term] ends with %goodbye
* Copy URL/Query feature to use in the Client Scripts
* Form/List/Column & Cell context menu
* Personalize (individual) vs Configure list (for all users) & form, personalization saved in user settings.
* Impersonating the user
* Aggregate properties (min,Max avg etcc)
* We can add list controls to list view
* Configure =>Form Layout => section
* Configure =>Form Design =>
* Configure => Related Lists (we can select the view that is applicable)

# Workflows

## UI Policies

Form control

• Run on client-side

• Easy to use (no scripting required)

• Used to set form fields to:

• Mandatory

• Read-only

• Show/Hide

Set an incident’s Short description field to read-only if the incident state is Closed

Hide an incident’s Resolution notes field if the state is Open

## UI Actions

Add buttons, links, and items to context menus

• Server-side and client-side

• Leverage JavaScript

1. Trigger Salesforce integration, creating

an associated Salesforce ticket

2. Reject an approval record

## Business Rules

Run off specific table & triggered by database operations

• JavaScript that runs on server-side

Configure when to run

• Before

• After

• Display

• Async

During what operation

• Insert

• Update

• Delete

• Query

1. User sends request to server for specific incident (query)

2. Application server requests record from database server

3. Database server responds to application server with record

4. Application server checks for display business rules, then sends response back to client

5. User modifies incident record via form and sends update request

6. Application server receives update, checks for before business rules, then sends to database server

7. Database server updates record

8. Application server checks for after business rules

1. Create an associated CI when a new

asset is created

2. When an incident is reopened,

increment the reopen count

## Client Scripts

• JavaScript on client-side; shipped to browser

• Form view

• Access to helper methods

• Triggers:

• On load

• On change

• On submit

• On cell edit

1. Highlight Caller field if user is a VIP

2. Run Conflict checker for Change Management

## Data Policies

• UI policies for the backend

• Restrict data through imports (import sets)

• Web services



1. Require the Type field on the Change form, for web services

2. Require the Close notes on an Incident before changing the status to Closed/ Resolved

## Script Includes

• Store JavaScript functions and classes

• Reusable code

• Server-side

• Only ran when called

1. Create commonly used helper functions

2. Call a custom function via GlideAjax

**Client Side**

• Client Scripts

• UI Policies

• UI Actions

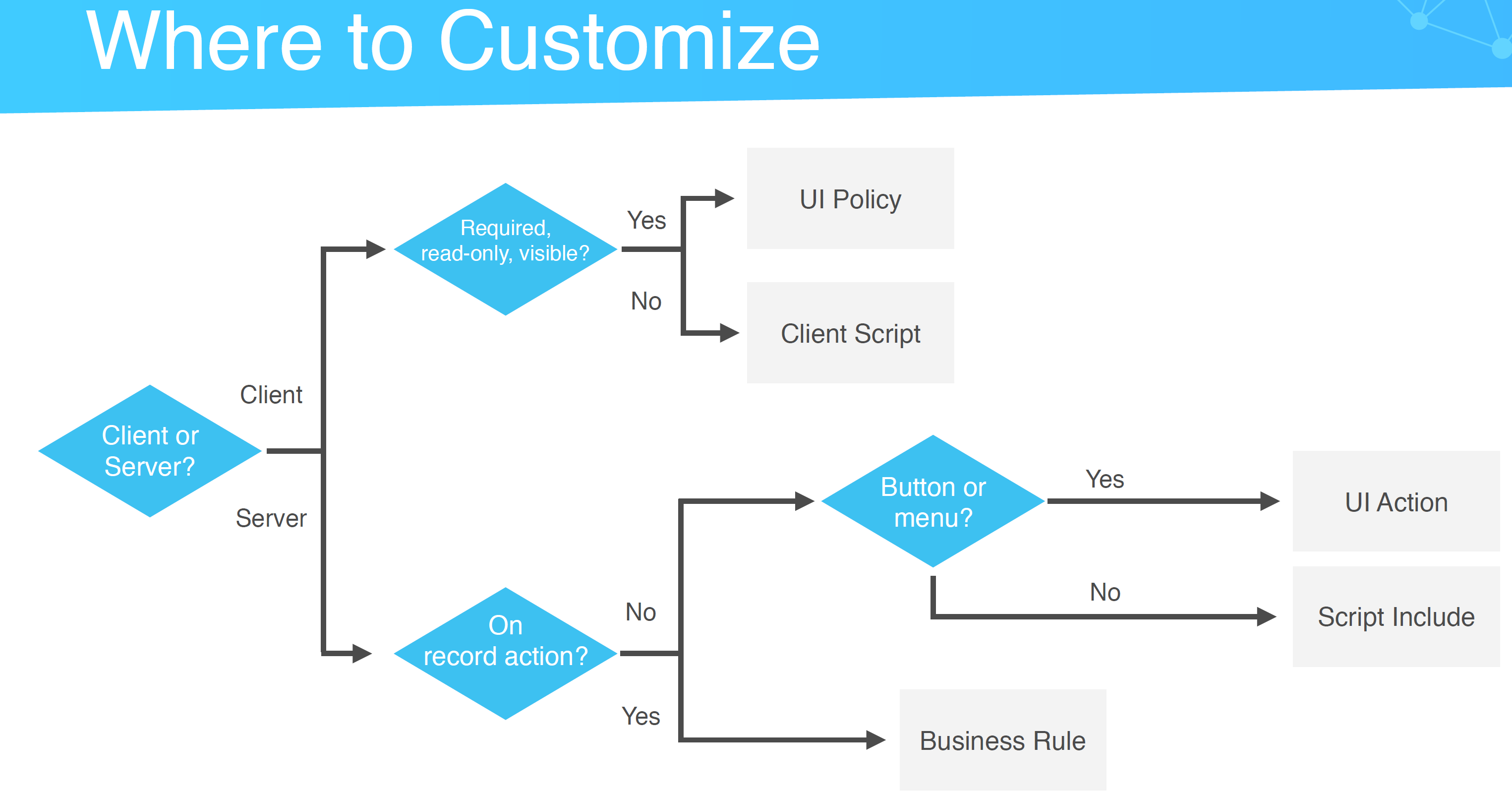
**Server Side:**

• Business Rules

• Script Includes

• UI Actions

• Data Policies



## Update Sets

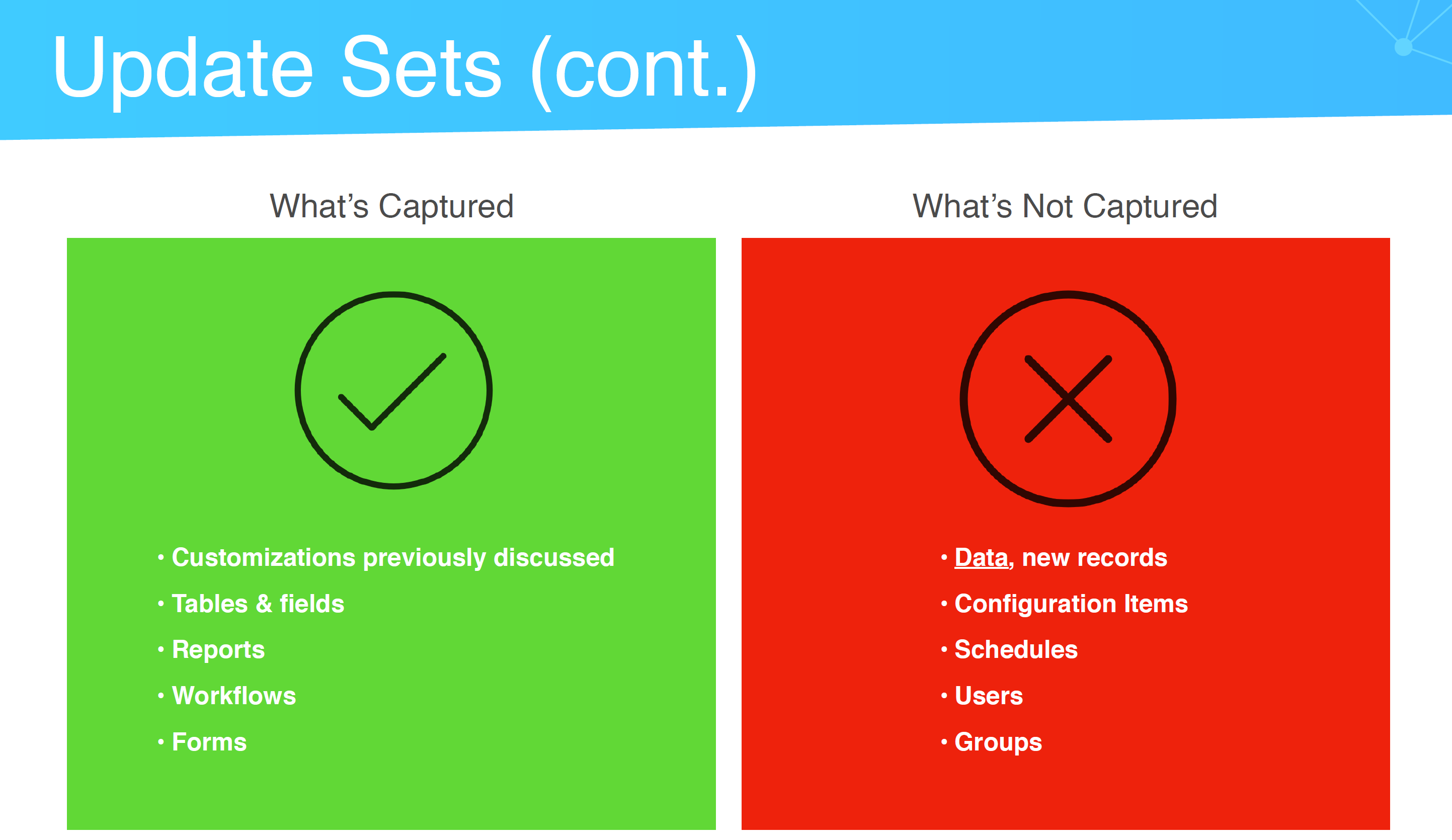
• Record most customizations & configurations

• Used for moving changes from instance to instance

• XML snapshot of record

• Versions & merging

• Previewing & committing



## Plugins

• Activate plugins at any time

• May require subscriptions

• Hundreds of plugins

• Demo data

## Major Tables

• Task [task]

• Incident [incident]

• Problem [problem]

• Change [change\_request]

• User [sys\_user]

• Group [sys\_user\_group]

• Role [sys\_user\_role]

• Location [cmn\_location]

• Company [core\_company]

• Knowledge [kb\_knowledge]

• Knowledge Category [kb\_category]

• Knowledge Base [kb\_knowledge\_base]

• Service Catalog [sc\_catalog]

• Catalog Items [sc\_cat\_item]

• Configuration Item [cmdb\_ci]

• Server CI [cmdb\_ci\_server

## Data Dictionary Tables

• Contain metadata about tables

• sys\_db\_object record represents a table

• sys\_dictionary record represents a field on a table

• sys\_documentation record represents a field label, etc.

**Fields**

• Each table contains many fields

• Different field types

• Calculated values

• Attributes

• Default values

• Dictionary overrides

Field Types

• String

• Date

• Time

• Choice

• True/False

• List

• HTML

• Script

• Reference

**Globally Unique Identifier (GUID)**

• Referred to by sys\_id

• A unique 32-character hexadecimal string

• Every record has a sys\_id

• sys\_id’s are automatically generated for all records

• Example: ef4225a40a0a0b5700d0b8a790747812

**Reference Fields**

• Power of RDBMS

• References are everywhere

• Store sys\_id in reference field

• Must match exact record (1 to 1)

• Reference qualifiers

* If we type “incident.list” in the application search then it opens the incident form in list mode search mode ( with out any qualification, no filters)

Table Relationships

• Example:

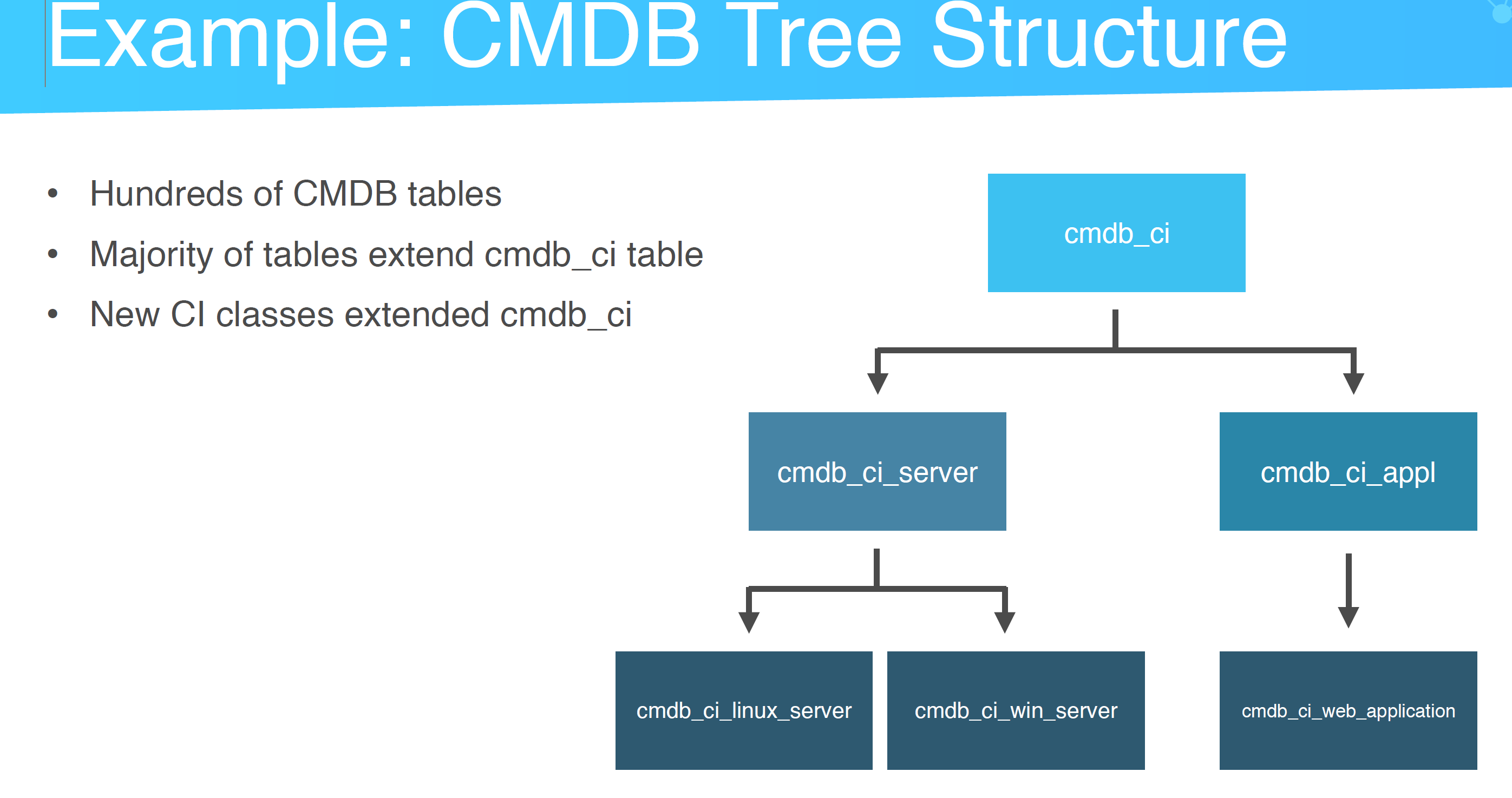
• Person table has first name and last name fields

• Student & Faculty tables extend Person table, thus Student and Faculty tables

inherit first name and last name fields

• Dictionary overrides

• Very common in CMDB



Tables & Columns Module

• Great for exploring tables and fields

• Shows field attributes

• Link to schema map

**Schema Map**

• Visual schema map

• Shows extended tables

• Focus on different tables

• First delete all records, then delete table

• Cannot delete out-of-box tables

## Creating an Application

• 3 templates

• From scratch

• Create custom application

• Start from template

• Application scope

* Configure dictionary entry for the fields DB property (readonly, mandatory, display, max length,name & label), Choice list, dependent field, default value, acess controls, dictionary overrides,
* Attributes section is used for reference values during the drop-down selection
* Number Maintance record is used to change the prefix and number of digits for auto generated id etc..?
* Table => Controls =>
* Table relationships
* Schema Maps (we can also use it for debugging)
* Application is a table
* System => Application => new
  + Create a table
  + Scope => unique id
* Service Now Studio
* Makesure to select the right scope & update set while modifying the workflows

## Users

• A ServiceNow account

• A record in the sys\_user table

• Users may be apart of 0 or more groups

• Users may have 0 or more roles

• Users may also be assigned delegates

## Grous

• A record in the sys\_user\_group table

• Buckets which hold users who share a

common purpose/role

• Roles are assigned to groups

• Groups contain 0 or more roles

• May inherit other groups

**Roles**

• Grants permissions to parts of the system

• A record in the sys\_user\_role table

• Assign roles to a group

• Contain access control rules

• Many out-of-box roles

**Out-of-Box Roles**

• admin

• security\_admin

• itil

• itil\_admin

• impersonator

• knowledge\_admin

• report\_admin

• catalog\_admin

• asset

• ecmdb\_admin

**Access Controls**

• A record in the sys\_security\_acl table

• Used to grant access

• Specifies

• Object and operations

• Permissions required

• \* wildcard

• Thousands of out-of-box access controls

**Tying It All Together**

• Access controls are assigned to roles

• Roles are assigned to groups

• Users are assigned to groups

**Access Operations**

**Operation Action**

execute Run app or script

create Insert records

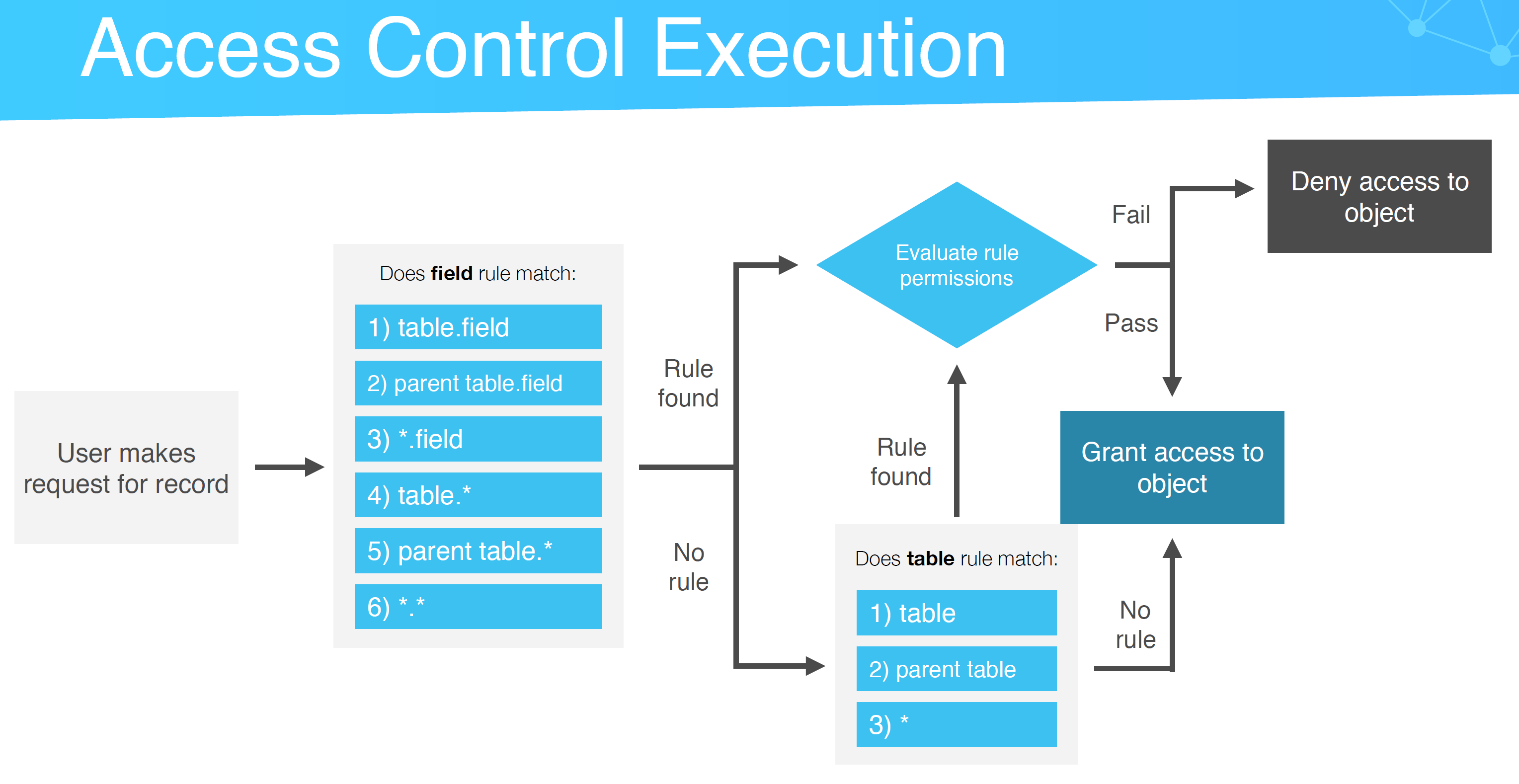
read Display records

write Update records

delete Remove records

list\_edit Update records from list view

report\_on Create reports



**LDAP**

• Lightweight Directory Access Protocol

• Industry standard for directory services

• Active Directory

• Defines structure for users & groups

• 2 parts

• Authentication

• Data population

Single Sign-On (SSO)

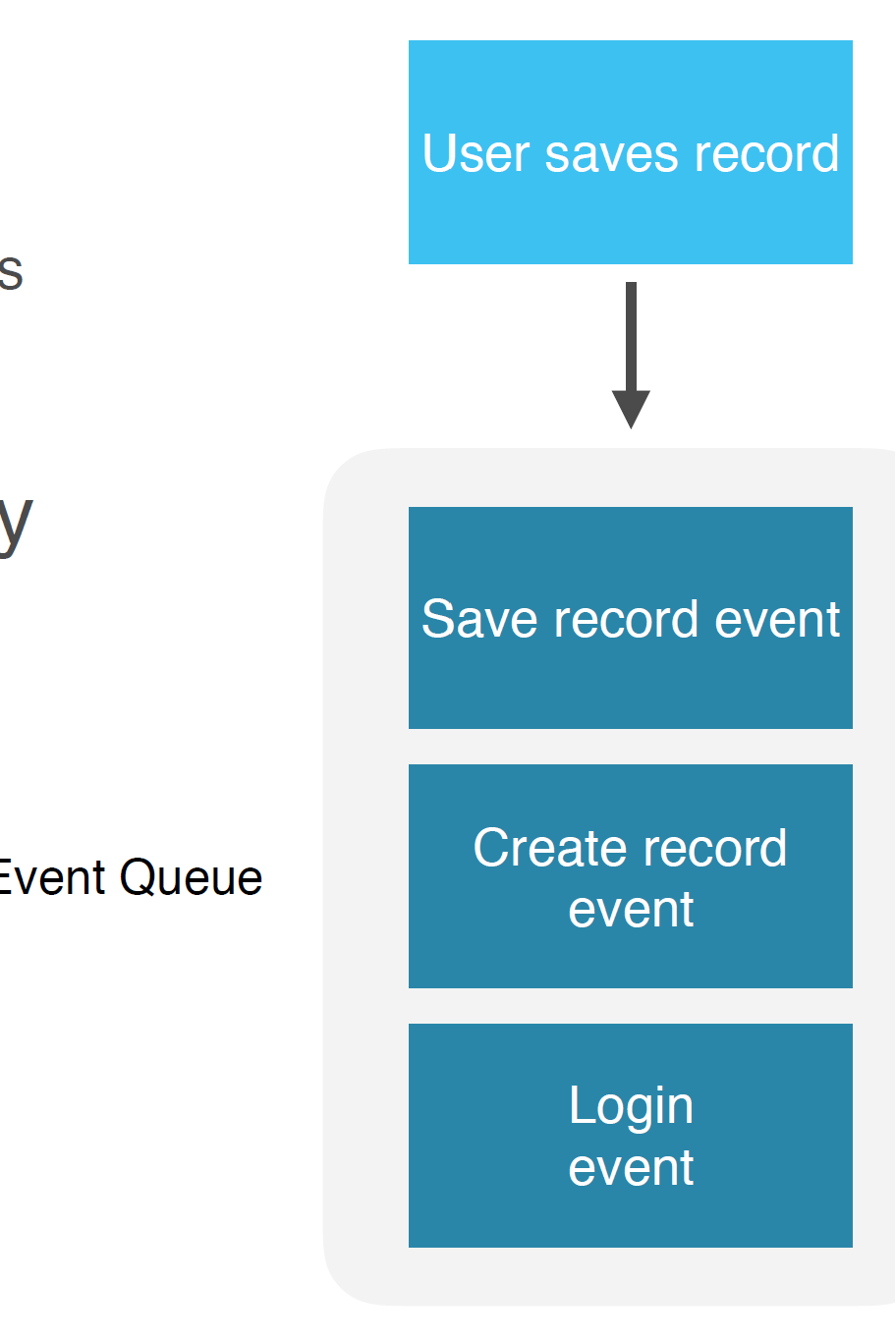
• Used for authentication

• 3rd party (provider)

• Single account for all system access

• SSO provider communicates with ServiceNow

• Example: Okta, Facebook

* Elevated Privilege
* Roles => Contains roles/Applications with Role/modules with Roles
* Contains roles => contains other roles
* System Security => ACL
* 7800+ OOB ACL’s
* Operation => Read/write/delete/execute/ save\_as\_template
* Type => UI Page/record etc..
* Admin overrides
* Show ACL Execution Plan => Show the additional ACL’s for the field
* For the record when the filed name is none then it is applicable for the entire table
* We can also use Script to determine the scope of the ACL
* LDAP: u0392 active
* Authentication & data population
* Impersonator rule is needed, but with this we can’t impersonate the admin role
* LDAP Server confgurations
  + URL
  + Userid/password/ Search directory (DC)
  + LDAP OU Definations
  + Query field
  + Transform Map
  + Connection timeout etc..
* SSO (multiple provider Single Sign-on installer plugin to the activated for this)
  + Identity providers
    - URL
    - Authencation request/SSO request
    - ServiceNow Homepage
    - Audience URI
    - Nameid Policy
    - External redirect
    - Signing encryption key
    - Sign Algoritham
* Incident
  + “The goal of Incident Management is to restore normal service operation as quickly as possible, while minimizing impact to business operations and ensuring quality is maintained.”
  + Most popular
  + • Used for logging incidents
  + • Any user may create an incident
  + • Record in incident table
  + • Classify incidents by impact & urgency
  + May be created from many
  + triggers
  + • Out-of-box configurations
  + • Contextual search
  + • Data lookups
  + • Related records
  + • Email/SMS notifications
  + • Service Level Agreements
  + Incident Creation
    - Create New
    - Service Portal
    - Email
    - Integration
  + “A service level agreement (SLA) is a record that specifies the time within which service must be provided.”
  + Timezones, business rules pay key role in SLA
  + • Used to track if a certain level of service
  + has been provided
  + • Workflows
  + • Start, Pause, Stop, Reset conditions
  + • Retroactive starts
  + Number Maintaince
    - Use Dynamic default
    - Get Next Padded number
  + edge\_encryption\_enabled=true,ignore\_filter\_on\_new=true,ts\_weight=50
  + Multiple dependent fields? Or a qualification?
  + Data lookups (impact & urgency)
  + Data lookups (impact & urgency)
    - Data Lookup definitions
    - Priority lookup (impact & urgency)
    - Source table (incident)
    - Matcher Table (Priority Data lookup (dl\_u\_priority))
  + Contextual Search
  + Watchlist view
  + Incident Related list
    - Task SLA’s
    - Affected CI’s
    - Impacted Services/CI’s
    - Child Incidents
  + “The primary objectives of problem management are to prevent problems and resulting incidents from happening, to eliminate recurring incidents, and to minimize the impact of incidents that cannot be prevented.”
  + • Record in problem table
  + • Related incidents
  + • Problem tasks
  + • Workarounds
  + • Root cause
  + Related records for Problems
    - Incidents
    - Problem tasks
  + “The objective of change management in this context is to ensure that standardized methods and procedures are used for efficient and prompt handling of all changes to control IT infrastructure, in order to minimize the number and impact of any related incidents upon service.
    - • 3 change types
    - • Normal
    - • Standard
    - • Emergency
    - • Record in change\_request table
    - • Quite complex
    - Change Management Features
      * • Risk assessment
      * • Out-of-box workflows for all 3 change types
      * • Out-of-box service catalog for change proposals
      * • Schedules
      * • Calculated risks
  + “Configuration management is a process that tracks all of the individual Configuration Items (CI) in an IT system which may be as simple as a single server, or as complex as the entire IT department.”
    - • “Any component that needs to be managed in order to deliver an IT service”
    - • CI examples: servers, desktops, software, routers, switches
    - • CMDB contains many Configuration Items (CI’s)
    - • Class = Category
    - • Records in cmdb\_ci table with hundreds
    - of extending tables
    - • Hundreds of out-of-box CMDB tables
    - • Each CI class has its own table
    - • Relationships
    - • Dependency Views map
    - • Granularity varies by maturity
    - • Discovery vs manual population
    - CI Class Manager
      * Single pane of class to show different classes and attributes
    - CMDB Groups
      * Groups to assign to the CI’s
    - CMDB Query builder => created CMDB queries using GUI
    - CMDB Remediations
    - CMDB Reports
    - Business Services (used for dependency mapping & dependency view)
    - CI Relationship Type (Parent relationship type & Child relationship type)
    - Relationship Editor ( define the relationships, parent & Child)
    - Mostly these relationships are populated using Discovery
  + “With the ServiceNow Service Catalog application, create service catalogs that provide your customers with self-service opportunities.”
    - • Create product offerings for your users
    - • Multiple catalogs & categories
    - • Track progress of request
    - • Many tables
    - • sc\_cat\_item
    - • sc\_request
    - • sc\_req\_item
    - • sc\_task
    - • sc\_cart
  + Service Catalog Features
    - • Requests, Request Items, SC Tasks
    - • Record producers
    - • Order guides
    - • Variable sets
    - • Workflows
    - • User criteria
    - Catalog => Open Records => Requests/Items/Tasks
    - My Catalogs => Catalog Definations => Catalogs/categories
    - Catalog Definitions => Catalog Items
    - Catalog Item
      * Variables
      * Variable Sets
      * Image
      * Cost
    - Content items => re direct to KA
    - Order Guides (like wizards and dynamically show the catalog items depending on the questions
      * Rule in order guide to show the required catalog items
    - Record producers
      * It is powerful to create a record in any table on the platform
      * Uses a script to create the record
      * Can submit the requests or tickets with few questions
    - Roles & user criteria is used to limit the access of the catalog items
  + “The ServiceNow Knowledge
  + Management (KM) application enables the sharing of information in knowledge bases. These knowledge bases contain articles that provide users with information such as selfhelp, troubleshooting, and task resolution.”
    - • Sharing of information across organization
    - • Troubleshooting, self-help, task resolution,
    - procedures, etc.
    - • Knowledge bases, categories, articles
    - Knowledge Management Features
      * • Multiple knowledge bases
      * • Approval process w/ workflows
      * • i18n support
      * • Public articles
      * • Article feedback
      * • Article lifetime
      * • Article population
    - KA ritire process
    - KA categories
    - Different Knowledge Bases like IT, Business etc..
    - KA publishing, approval and review
    - Articles & Questions
    - Social questions
    - User criteria is used to restrict the access of catalog items and Knowledge Articles
    - Can contribute
    - Feature Content
  + “Service Portal is an alternative to the Content Management System (CMS) and a simple way to create portals for end users. It is a portal framework that allows administrators to build a mobile-friendly self service experience for users. It interacts with parts of the ServiceNow platform, so users can access specific platform features using Service Portal.”
    - • Create portals for end users as alternative
    - to platform view
    - • Provides better UI & UX
    - • Responsive
    - • Replaces archaic CMS
    - • Leverages Bootstrap & AngularJS
    - Responsive UI
    - Supports mobile interface
    - Right click + inspect element
    - Home => System Status (can show outages for CI’s)
    - User profile => preferences etc.
    - Service Portal Configuration
    - => Braning Editor =>Theam colors, Change color, logo, background images
    - Changing the order of the widgets
    - Page Editor
    - Widget editor
    - Portals => to configure SP & other OOB dashboards etc..
    - Announcements => appear at the top of the selected service portals
    - Themes => contain CSS, java scripts, which can be used for customizing the pages
    - Pages =>
  + “ServiceNow Connect is a real-time messaging platform that connects you to your coworkers, bypassing email and static documents.”
    - • Messaging platform in ServiceNow
    - • Integrates to other parts of the platform
    - • Connect Chat & Connect Support
    - • Multi-queue support
    - Similar to teams
    - Connect Chat
      * Right top chat icon
      * Chat Queue
      * Avg wait time
  + “Visual Task Boards (VTB) transform the navigation of lists and forms into an interactive graphical experience. With Visual Task Boards, you can view and update multiple task records, which appear as cards that can be moved between lanes.”
    - • Interactive graphical user interface for
    - working with task records
    - • Tasks are viewed as cards
    - • Lanes represent the state of the card
    - • Freeform or data-driven
    - This is like a Kanban board
    - We can create multiple boards
    - We can also create data driven boards based on the table input
  + Basic Configuration
    - • Define common properties
    - • Configure logo, colors, and banner text
    - • Configure system defaults
  + System Properties
    - • sys\_properties table
    - • Hundreds of system properties
    - • Categories
    - • Examples
    - • List v3 properties
    - • UI properties
    - • Email properties
    - • System properties
    - • Global text search properties
    - Block out properties
    - System Properties
      * Basic Configuration & Basic Configuration UI 16
      * My Company
      * UI Properties
      * Mobile UI Properties
      * Email Properties
      * Sys\_properties.list
        + Type
        + Value
        + Ignore Cache
        + Private
        + Read roles
        + Write roles
        + Categories ( this is linked to application specific configurations like Incident properties etc..)
  + Dashboards
    - • Dashboards are the new homepages
    - • May contain reports, and dashboard widgets
    - • Any user may create a new dashboard
    - Self-Service
      * • Accessible by every user in ServiceNow
      * • Does not require a role
      * • Personalized information
      * Business Applications
      * Dashboards
      * Service Catalog
      * Knowledge
      * Help the Helpdes k
      * My Requests
      * My Incidents
      * Connect Chat
      * My Profile
      * My Tagged Documents
      * My Tags
      * System Defination => Application Menu’s => Self Service, here we can define the apps to be tagged under self service (tagging the roles)
      * Add widgets to the Dashboard
  + Mobile
    - • ServiceNow mobile app (native app)
    - • iOS & Android
    - • ServiceNow mobile web app (browser)
    - • Leverage phone hardware
    - Alerts
    - Can take pictures for attachments
    - We can crate a custom portal for mobile apps
  + Upgrades
    - • Upgrade history
    - • stats.do for current version
    - • Request upgrades through ServiceNow HI
    - Stats.do to show the app versions etc..
  + ServiceNow HI
    - • Customer administration
    - • ServiceNow for ServiceNow
    - • hi.service-now.com
    - • Manage instances
    - • Request upgrades
    - • Request clones
    - • Create incidents
    - • Access ServiceNow Knowledge Base
  + System Diagnostics
    - • stats.do
    - • Build information
    - • Memory allocation
    - • Semaphore information
    - • Response times
    - • Active sessions
    - • System logs
    - • Debugger
    - • ServiceNow Docs
    - • Community
    - Build name
    - Build date
    - Patach version
    - Instance name
    - IP address
    - OS configuration
    - Server response times
    - Network time
    - System Diagnostics
      * Script debugger
      * Diagnostics Page
      * Componets status
      * Progress Workers
      * Active transactions
      * Upgrade monitor
      * Stats
        + Stats
        + Slow events
        + Slow Mutex Locks
        + Slow Queries
        + Slow Scripts
        + Slow Transcations
        + Table IOStats
      * Performance Windows
        + Performance Events
        + Latest Upgrade
        + Latest Periodic event
        + Active Query Index hints
      * Index Suggestion
        + Index Suggestions => to review
        + Index Suggestions => in progress
        + Index Suggestions => Done
      * Session Debug
        + Enable All (Enables All the debugging and these logs can be seen under the respective screen while performing the user action)
        + Disable All
        + Debug Business Rule/SQL etc..
      * Database
        + Information
        + Cache
        + Term Optimization
      * Browser
        + Browser Page Timer
        + Browser JavaScript Timer
  + Gear Icon => Developer => Java Script Logging/Field Watcher
  + Response Time
    - • Server (orange)
      * • Process data
    - • Client (green)
      * • Render data
    - • Network (red)
      * • Transfer data
  + Security
    - • High-Security plugin
    - • Security properties
    - • IP address access controls
    - 
    - High Security Settings
    - System Properties
      * Security
      * Attachment list
      * IP Address Access Control
        + Whitelisting & block listing the IP’s
        + Range start & END (IP address range for accessing the users)
  + Events
    - • Triggered by
    - • User actions
    - • Scripts
    - • Time/Schedules
    - • Event queue
    - • Event registry
    - • Parameters
    - 
    - Pass the parameters to the events (inc id, receipent id etc..}
    - System Logs => Events
      * Incident assigned/updated etc..
      * Provides URI/other details
    - System Logs => Emails
      * All the emails generated in the system
      * Preview HTML body
      * Show Log Entries
      * Receipents
      * We can copy the event name from the email logs and search it in the Events
      * Open the Notification record
      * When to send? Record inserted or updated & conditions
      * Who will receive it=> Users/groups, Subscripe
      * What it will contain => Static HTML content/email template/
    - Email
      * Digest Intervals
      * Notifications
      * Notification Email Scripts
      * Notification Categories
      * Notification Filters
  + Notifications
    - • Email, SMS
    - • Templates
    - • Triggered by events
  + Workflows
    - Workflow Editor
    - Published/Checked in
    - Core Activities
      * Approval Group
      * Approval User
        + Users => request.requested\_for.department.dept\_head
        + Condition for approval =>

Wait for =>everyone to approve/anyone to approve

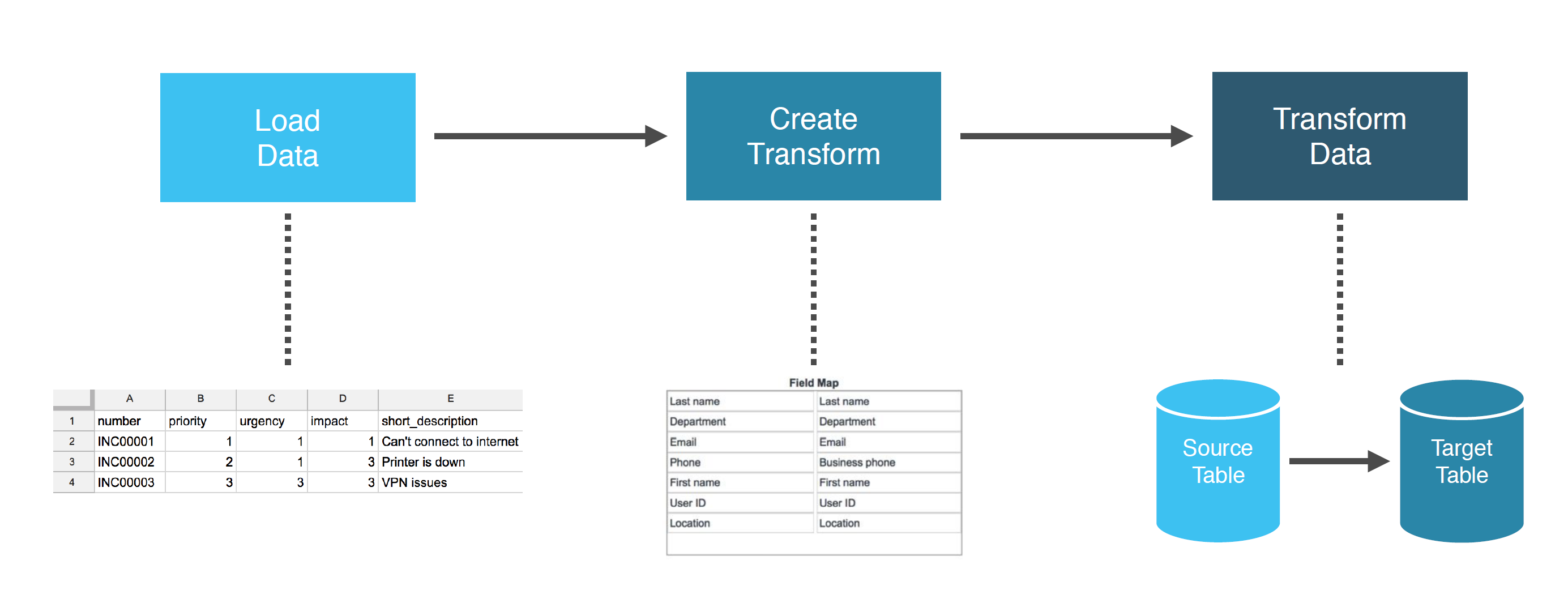
When anyone rejects => reject the approval

Approval Column => Approval

Approval Journal Column => Approval\_history

Schedule

* + - * Approval Action
      * Approval Coordinator
      * Generate
      * Manual Approval
      * Rollback to
    - Conditions
      * If
      * Switch
      * Wait for condition
      * Wait for WF event
    - Notifications
      * Create Event
      * Notification
    - Service Catalog
      * Scriptable order guides
    - Subflows
      * Parallel Flow Launcher
    - Tasks
      * Attachment Note
      * Catalog Task
      * Create Task
    - Timers
      * Timer
    - Utilities
      * Branch
      * Join
      * Lock
      * Log Messages
      * Log Trace Messages
      * Return Value
      * Run Script
      * Set Values
      * Turnstile
      * Unlock
    - Workflow Properties
      * General
      * Activities
      * Application
      * Catalog Variables
      * Schedule
      * Stages
      * Estimated Runtime
  + Import Sets
    - • Import data
    - • Powerful & flexible
    - • Data sources
    - • Database connectors, web services, CSV, XML, Excel,
    - FTP
    - • Transform mapping
    - Import Process



* + - System Import Sets
      * Load Data

Import set table (create table/existing table)(default table is created with string data types)

* + - * + Source of the import (file/data source)
        + File
        + Sheet number
        + Header row
      * Create Transform Map
        + Source Table
        + Target table
        + Run business rules
        + Enforce Mandatory fields
        + Copy empty fields
        + Order
        + Run Script
        + After the submission, open the transform map and select these options

Auto Map Matching Fields

Mapping Assist

Transform

Index Coalesce

* + - * + Transform Scripts

On After

On Before

onChoiceCreate

onComplete

onForeigninsert

onReject

onStart

* + - * Transform
        + Find the import set
        + Select the transform map
        + Transform
        + Check the transform history
        + Check the import log
    - Data Sources
      * JDBC Connection
        + Import set table name
        + Type (JDBC/LDAP/File)
        + Use MID Server
        + Format
        + Orable SID
        + Oracle port
        + User name
        + Password
        + Server
        + Query (all rows from table)
        + Query timeout
        + Connection timeout
        + Table name
        + Use last run datetime
      * Cleanup/Scheduled clenup
  + Reports
    - • Many different types
      * Pie Charts
      * Bar Charts
      * Time Series charts
      * Donugt charts
      * Pivot tables
    - • Roles
    - • Create & share reports
    - • Schedule reports
    - View/Run (we can run the reports available in the system)
    - My reports/Group/Global/All
    - Source Type => Table/Data Source
    - Header & Footer templates
    - Administration
      * Report Statistics
      * Color Schemes
  + Creating a Custom App
    - Creating the APP
      * System Applications => Studio
      * Create Application
      * Name/Description/Scope(unique scope name for app)/Logo (image)
      * General Info(Interface)-Workspace/Mobile/Classic
      * Data (Table)-(Create new table)-upload spread sheet/extent table/create from scratch
      * Add fields
      * Table Properties – Auto Number (Prefix, starting number, number of digits)+ Make Extendable
      * Manage Access
        + Role -Create/Read/write/delete
        + + Add another role
      * Design
    - Creating the Application Menu’s
      * System Definition => Application Menus
      * List of records
      * Module Name
    - Craete a new Table with the Application as existing application
      * Create Module
      * Create Mobile Module
      * Add Module to Menu (Select the application menu where you want to add this)
    - List Layout => to configure the list view of the form
      * View Name
    - Form Layout
      * Begin Split/end split
      * View Name
    - Configure Dictionary for the filed and set the default value
    - Table => select the column => set the display flag to true to sent the field value in the reference field.
    - Creating the relation ship between the Tables
      * System Definition=> Relationships
      * Applies to table => Target table
      * Queries from table => Source table
      * Query with => Java Script
        + Function refineQuery(current,parent)

{

Current.addQuery(‘field1’,parent.sys\_id);

Current.addQuery(‘status’,’Approved’);

}

* + - This is useful in creating the related lists
  + Application Explorer =>Create Application File => Business Rule etc..
  + Business Rule

|  |  |
| --- | --- |
| Name |  |
| Table |  |
| Application |  |
| Active |  |
| Advanced |  |
| When to Run | When=> Before/After  Order=>  Insert/Update/Delete/Query  Filter Conditions  Role Conditions |
| Actions |  |
| Advanced |  |