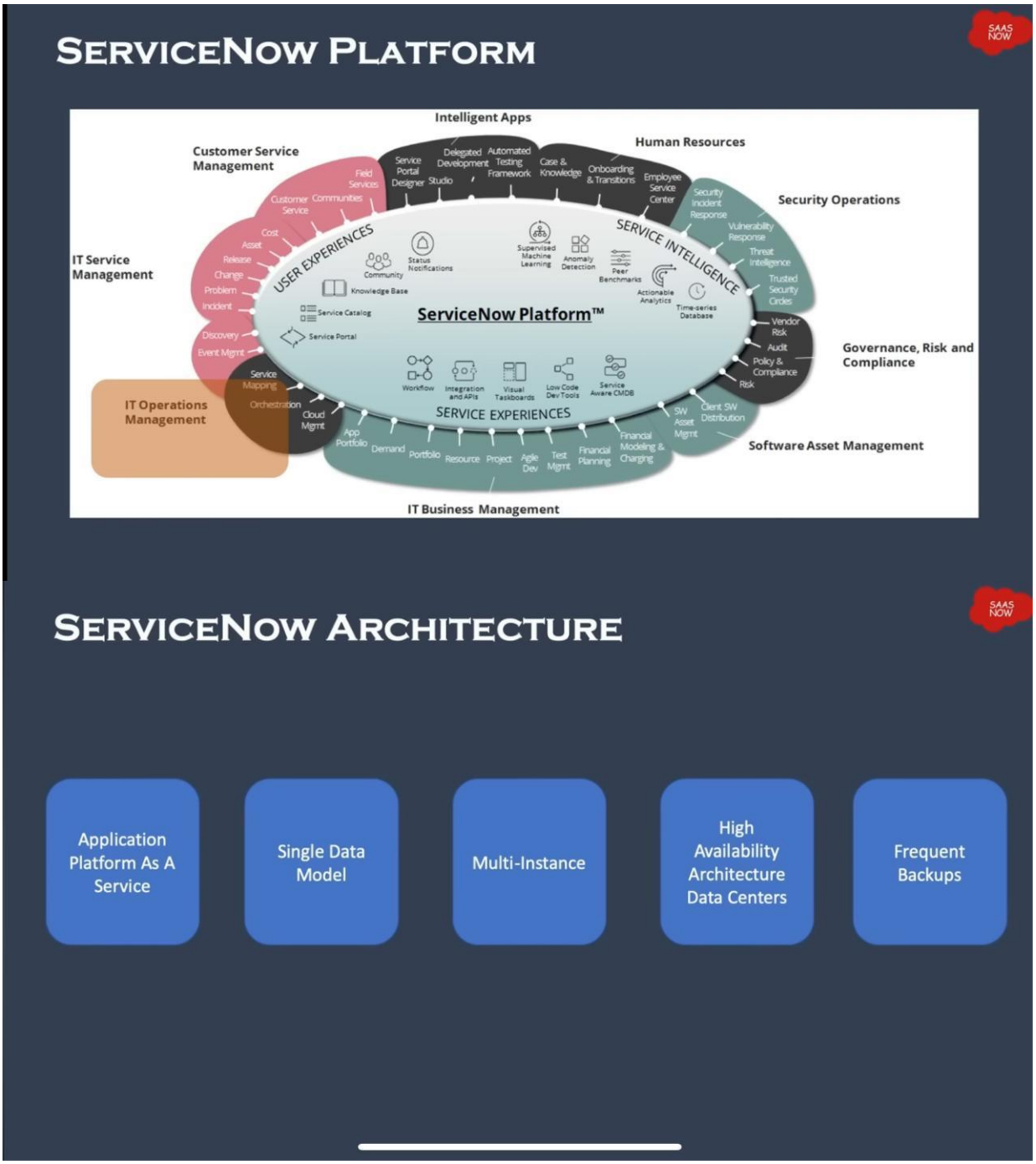
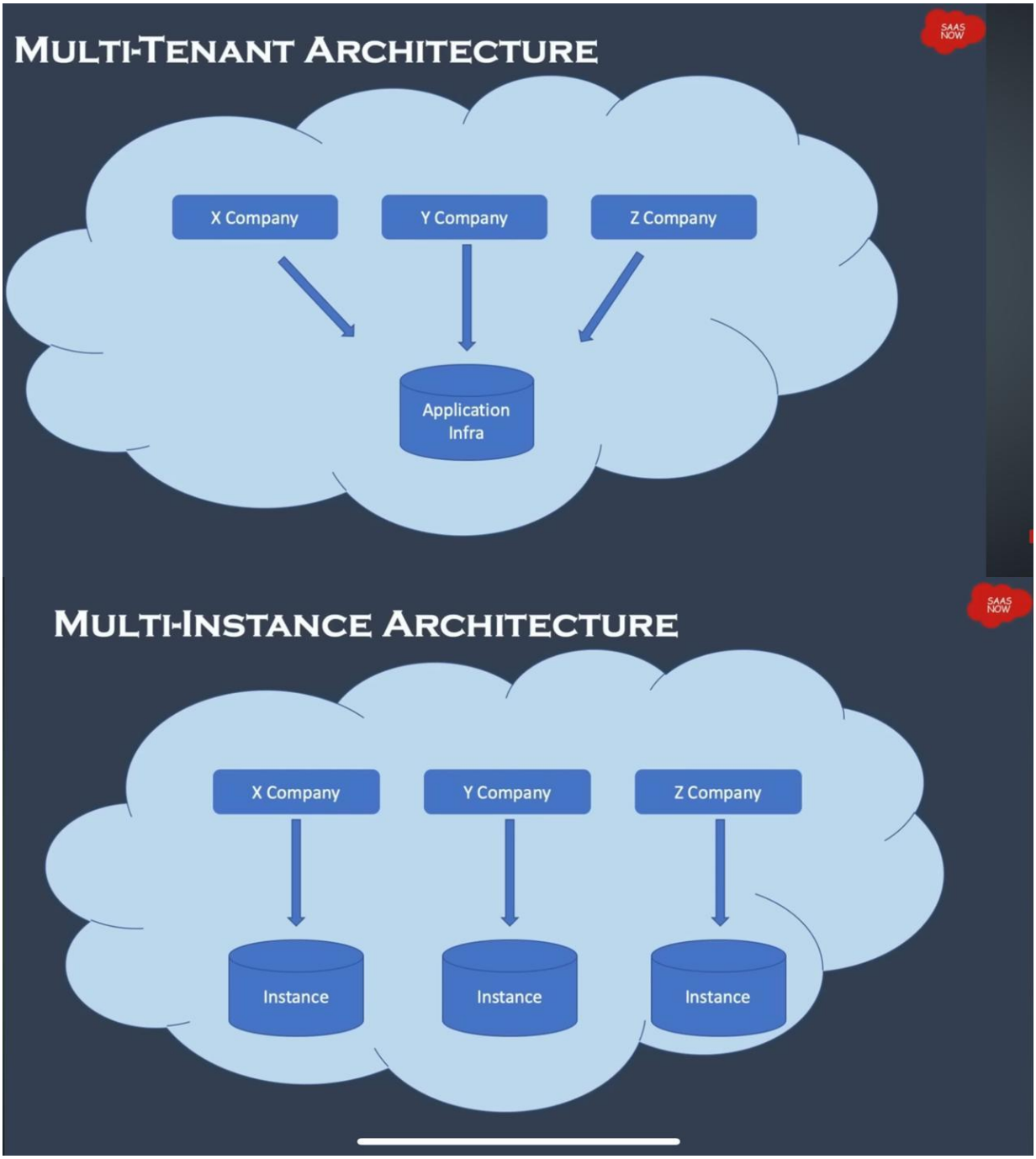
SERVICENOW ADMINSTRATION

PLATFORM OVERVIEW AND ARCHITECTURE





COMPONENTS

1)USER

- A user in ServiceNow is any individual who needs access to the platform. Each user is uniquely identified by a user ID.

2)GROUP

- A group is a collection of users who share similar responsibilities or perform similar tasks. Groups make it easier to manage permissions and assign tasks.

3)ROLE

- A role defines a set of permissions that control what a user can access and what actions they can perform within ServiceNow. Roles can be assigned to both users and groups.

BASE SYSTEM ROLES

- Admin
- Approver_user
- Itil
- Catalog_admin

USER INTERFACE AND BRANDING

- **Navigation Bar:**
  - Located on the left side, the navigation bar provides access to applications, modules, and other system features.
  - Users can search for applications or features using the global search functionality.
- **Homepage and Dashboards:**
  - Dashboards provide a visual representation of key metrics, reports, and performance indicators.
  - Users can customize their homepage by adding widgets, charts, reports, and lists to display relevant information.
- **Forms:**
  - Forms are used for data entry and task management. They allow users to create, view, or modify records (e.g., incidents, change requests).
  - Forms are fully customizable, allowing organizations to add or remove fields as needed.
- **Lists:**
  - Lists display records in a tabular format, providing users with the ability to sort, filter, and edit multiple records simultaneously.
  - Lists can be personalized by individual users to show relevant columns and information.
- **Service Catalog:**

- The service catalog offers a user-friendly interface where users can request services or products from different categories.
- Items in the service catalog can be customized with descriptions, icons, and user-specific options.

- **Self-Service Portal:**

  - Provides a simple, branded interface for end users to access knowledge articles, submit requests, or report incidents.
  - The portal is designed for users who may not be familiar with the backend of the platform, offering a more consumer-like experience.

SERVICE NOW BRANDING

- **Logo Customization:**

  - Organizations can upload their own logos to replace the default ServiceNow logo.
  - The logo typically appears in the top navigation bar, self-service portal, and email notifications.

- **Theme and Colors:**

  - Admins can customize the color schemes (primary and secondary colors) of the platform to match the organization's branding guidelines.
  - Custom themes can be applied to backgrounds, headers, buttons, and other UI elements.

- **Banner and Icon Customization:**

  - Banners across pages, especially on the self-service portal and service catalog, can be customized with colors, images, or icons.
  - Icons representing applications and services in the service catalog can also be modified.

- **Self-Service Portal Branding:**

  - The self-service portal can be heavily branded, with custom themes, layouts, fonts, and images that reflect the organization's style.
  - Custom CSS (Cascading Style Sheets) can be applied to adjust various visual elements, offering full control over how the portal appears.

- **Email Branding:**

  - ServiceNow allows customization of email notifications, enabling organizations to modify the format, branding, and content.
  - Organizations can include their logos, colors, and specific information in email templates.

- **Custom Fonts and Styles:**
    - The platform allows for adjustments in font type, size, and style to create a unique look aligned with brand standards.
    - Custom CSS or themes can be used for more advanced branding needs.

LISTS

- **Rows and Columns:**
    - Each row represents a record.
    - Each column corresponds to a field from that record's table.
- **Personalization:**
    - Users can personalize their view of the list by adding/removing columns and reordering them to display relevant data.
- **Inline Editing:**
    - Users can edit records directly from the list without opening individual forms.
- **Sorting:**
    - Lists can be sorted by any column by clicking on the column header.
- **Context Menus:**
    - Right-clicking on a list column or record allows users to access various actions (e.g., view, edit, assign, delete).

FILTERS

- **Filter Conditions:**
    - Filters consist of one or more conditions that records must meet to be displayed in the list.
    - Conditions are based on field values (e.g., “State is Open,” “Priority is High”).
- **Filter Operators:**
    - Operators such as "is," "is not," "contains," "greater than," etc., are used to define how a record should match the condition.

- **Save Filters:**
    - Users can save frequently used filters for quick access.
    - Saved filters can be shared with other users or groups.
- **Breadcrumbs:**
    - Filters are visually represented as "breadcrumbs" at the top of the list, making it easy to see and modify the criteria applied.

FORMS

- **Field Types:**
    - Forms contain various field types such as text boxes, date pickers, checkboxes, drop-downs, and reference fields.
- **Form Layout:**
    - Administrators can customize the form layout by adding/removing sections, tabs, and fields.
    - The form is flexible and can display additional related information like activities, attachments, and related lists.
- **Form Views:**
    - Different views can be created for a form, showing different sets of fields depending on the user's role or task.
    - Users can switch between views depending on their needs (e.g., a simplified view for end users and a detailed view for IT staff).
- **UI Policies and Scripts:**
    - Forms can be dynamically modified based on user input through UI policies (e.g., hiding/showing fields) or client-side scripts.
- **Form Buttons and Actions:**
    - Common actions such as "Submit," "Save," and "Delete" are available at the bottom of forms.
    - Contextual buttons allow users to perform record-specific actions like "Resolve," "Close," or "Approve."

ServiceNow Task Management

- **Task Management** in ServiceNow is a key functionality that allows users to create, track, and manage tasks across various workflows and processes. Tasks are central to the ServiceNow platform and are used in multiple modules such as **Incident Management**, **Change Management**, **Problem Management**, and **Request Management**.

Attributes:

- **Task Number:** A unique identifier for each task (e.g., INC0001234 for incidents).
- **Assigned To:** The user or group responsible for completing the task.
- **State:** The current status of the task (e.g., New, In Progress, On Hold, Closed).
- **Priority:** Determines the urgency and importance of the task (e.g., Critical, High, Medium, Low).
- **Due Date:** When the task is expected to be completed.
- **Description:** Details about the task's purpose or the work require

Task Types:

- **Incident:** Manage issues that disrupt normal service operations.
- **Problem:** Identify and manage root causes of incidents.
- **Change:** Handle requests to modify IT systems or infrastructure.
- **Service Request:** Manage requests for new services or products.
- **HR Case:** Handle employee-related issues or requests in HR Service Delivery.

Task Lifecycle:

- Tasks typically follow a defined lifecycle with various states (e.g., New, Assigned, In Progress, Completed, Closed).
- Workflow automations and business rules help ensure tasks progress through their lifecycle efficiently.

ServiceNow Notifications

In ServiceNow, **notifications** are essential for keeping users informed about key events, task updates, approvals, and changes within the platform. Notifications ensure that the right users receive timely information, helping to enhance productivity, collaboration, and process efficiency.

1. **Notification Types:**
    - **Email Notifications:** The most common type, used to send important updates and alerts via email.
    - **SMS Notifications:** For sending alerts and updates via text messages.

    - **Push Notifications:** Alerts sent to users on the ServiceNow mobile app.
    - **In-App Notifications:** Notifications that appear within the ServiceNow interface for logged-in users.
2. **Triggering Events:** Notifications are triggered by specific events in the system, such as:
    - **Record Changes:** When a task, incident, or request is created, updated, or closed.
    - **SLA Breaches:** When a service level agreement (SLA) is about to be violated.
    - **Approvals:** When an approval request is required or completed.
    - **User Actions:** When specific actions are performed by a user (e.g., a record is assigned to them).
3. **Conditions for Notifications:**
    - Notifications can be configured to send alerts only under certain conditions (e.g., "Priority is high" or "State is closed").
    - They can also be targeted to specific roles, groups, or individual users based on defined criteria.
4. **Notification Templates:**
    - **Email Templates:** Predefined email formats that determine how the notification looks and what content it includes. This could include dynamic fields such as task number, priority, and assigned user.
    - **SMS and Push Templates:** Shorter, more concise templates tailored for mobile delivery.
5. **Customizing Notifications:**
    - **Message Content:** Administrators can customize the content of notifications, including subject lines, body text, and variables that pull data from the records.
    - **Recipients:** Notifications can be sent to specific users, groups, or roles. They can also be targeted at specific conditions (e.g., assigned to, requested by, task owner).
    - **Notification Frequency:** You can configure notifications to be sent immediately, scheduled, or sent as part of a daily digest.
6. **Subscription-Based Notifications:**
    - ServiceNow also allows users to **subscribe** to certain notifications so they can receive updates on records they are interested in, without being the direct owner or assignee.

ServiceNow Knowledge Management

Knowledge Management in ServiceNow enables organizations to capture, store, and share knowledge across teams and departments. It centralizes information such as how-to guides, troubleshooting tips, and best practices, making it easier for employees and customers to find answers to common questions, resolve issues, and improve productivity.

1. **Knowledge Base:**
    - **Definition:** A repository where knowledge articles are stored and categorized. Organizations can create multiple knowledge bases for different departments or topics (e.g., IT, HR, customer support).

    - **Categories:** Articles are organized into categories, making it easy to browse or search for relevant content.
2. **Knowledge Articles:**
    - **Definition:** Individual documents containing information or instructions that address a specific topic.
    - **Types of Articles:** Articles can include FAQs, tutorials, troubleshooting guides, policies, and more.
    - **Article States:** Articles go through states such as Draft, Review, Published, and Retired to ensure accuracy and relevance.
3. **Search and Discovery:**
    - **Search Functionality:** Users can search knowledge articles using keywords or phrases.
    - **Contextual Knowledge:** The platform can suggest relevant knowledge articles based on user activity, such as while submitting an incident or service request.
    - **Filtering:** Users can filter search results by categories, date, popularity, or rating to quickly find relevant information.
4. **Knowledge Approval and Publishing Workflow:**
    - **Review and Approval Process:** Knowledge articles can go through an approval workflow before being published to ensure content quality and accuracy.
    - **Version Control:** Multiple versions of an article can be maintained, allowing users to view previous or current versions.
    - **Ownership:** Articles can be assigned to authors and owners, who are responsible for keeping the content up-to-date.
5. **Feedback and Ratings:**
    - **User Feedback:** End users can rate articles and leave feedback, allowing knowledge managers to improve or update content based on user experience.
    - **Analytics:** Knowledge managers can track article usage, popularity, and effectiveness to identify areas for improvement.
6. **Permissions and Access Control:**
    - **Role-based Access:** Permissions can be configured to control who can view, create, edit, or publish knowledge articles. Different teams can have access to their specific knowledge bases.
    - **Public vs. Private Knowledge Bases:** Some knowledge bases can be made public for customer or external access, while others are restricted to internal users only.
7. **Knowledge-Centered Support (KCS):**
    - ServiceNow supports the **KCS** methodology, which encourages creating and sharing knowledge as a part of resolving incidents and requests.
    - Agents can quickly create new knowledge articles or update existing ones based on the resolutions of specific issues.

SERVICE NOW Service Catalog

The **ServiceNow Service Catalog** is a self-service portal that enables users to request services, products, or information from different departments within an organization (e.g., IT, HR,

Facilities). It provides a user-friendly interface for submitting and tracking requests, streamlining the fulfillment process while ensuring a consistent experience for users.

1. **Service Categories:**
    - Services and products are organized into categories (e.g., IT Services, Employee Services, Software Requests), making it easy for users to browse and find what they need.
    - Categories can be customized based on department or function.
2. **Service Items:**
    - Each service or product available in the catalog is a **catalog item**. Items can range from hardware requests (e.g., laptops) to services (e.g., password reset, software installation).
    - **Item Details:** Each item includes a description, estimated fulfillment time, and any associated costs, ensuring users know exactly what they are requesting.
3. **Forms and Fields:**
    - When a user selects a service item, they are presented with a form to provide the necessary information (e.g., justification for a laptop, software requirements).
    - Forms can be customized with different field types (text boxes, dropdowns, attachments) to gather detailed information specific to the service.
4. **Workflows and Automation:**
    - Each catalog item can have an associated **workflow** that defines how the request is handled after submission.
    - Workflows can include steps for approval, fulfillment, and notification. For example, a request for new software might require approval from a manager before being fulfilled by the IT department.
    - **Automation** ensures requests are routed to the right teams or departments without manual intervention.
5. **Approvals:**
    - Many service requests require approval from managers or specific roles before being fulfilled. The ServiceNow Service Catalog automates this process by sending approval notifications and tracking responses.
    - Approvers can approve or reject requests directly from email, the ServiceNow interface, or the mobile app.
6. **Request Status and Tracking:**
    - Users can track the status of their requests from the moment they submit them. They receive real-time updates on the progress (e.g., "Pending Approval," "In Progress," "Completed").
    - Users can also view a history of their requests for reference.
7. **Service Catalog Items for Different Departments:**
    - **IT Services:** Hardware and software requests, password resets, access to systems.
    - **HR Services:** Employee onboarding, benefits inquiries, training requests.
    - **Facilities Services:** Maintenance requests, office space booking, equipment installations.
8. **Integration with Knowledge Base:**

- The Service Catalog can be integrated with the **Knowledge Base** to suggest articles or FAQs related to the selected service. This enables users to find solutions on their own without needing to submit a request.

Customization of the Service Catalog

1. **Tailored User Experience:**
   - Organizations can customize the Service Catalog to match their branding, including logos, color schemes, and themes.
   - Different catalog views can be set up for different user groups, ensuring that each department or user role sees the services relevant to them.
2. **Dynamic Forms:**
   - ServiceNow allows for dynamic forms that adjust based on the user's input. For example, if a user selects "Laptop Request," additional fields may appear to capture specific details like operating system preferences or accessories.
3. **Item-Specific Approvals and Workflows:**
   - Each catalog item can have its own unique workflow and approval process. For instance, a hardware request might require approvals from IT and the user's manager, while an office supply request may only need approval from one party.

ServiceNow Tables and Fields

Tables and **fields** are the core components of the data structure in ServiceNow. They store and manage data across the platform, allowing users to create records, build applications, and automate processes. Tables organize data into records (rows), while fields define the properties or attributes of each record (columns).

Tables in ServiceNow

- A **table** is a collection of records that store data for a specific function or process (e.g., Incident, Change, User).
- Each record represents a specific instance of the data stored in the table (e.g., an individual incident).

Types of Tables:

- **Base Tables:** These are the core tables provided by ServiceNow (e.g., `Task`, `Incident`, `Problem`, `Change`).
- **Extended Tables:** ServiceNow uses a hierarchical table structure. Many tables are extended from base tables to inherit their properties and fields. For example, the `Incident` table is extended from the `Task` table, inheriting fields like Assigned To, State, and Priority.

- **Custom Tables:** Administrators can create custom tables to store data specific to their organization's needs.

Table Hierarchy:

- The **Task Table** is one of the most important base tables in ServiceNow. Many tables, such as `Incident`, `Change`, `Problem`, and `Request`, extend from the `Task` table to inherit its fields and functionalities.
- **Parent-Child Relationships:** Tables can have parent-child relationships, where a parent table holds general information and child tables hold more specific data. For example, the `Incident` table inherits fields from the `Task` table but adds fields specific to incident management.

Important System Tables:

- **`sys_user`:** Stores user information.
- **`cmdb_ci`:** Stores Configuration Item (CI) information.
- **`incident`:** Stores incident management records.
- **`change_request`:** Stores change request records.
- **`problem`:** Stores problem management records.

Fields in ServiceNow

1. **Definition:**
   - A **field** is a single piece of data within a record. It defines an attribute or property of the data stored in a table.
   - Each record (row) in a table has multiple fields (columns) that store specific information (e.g., Short Description, Priority, Status).
2. **Field Types:**
   - ServiceNow supports a variety of field types to handle different kinds of data:
     - **String:** For text data.
     - **Integer:** For numeric values.
     - **Date/Time:** To capture dates and timestamps.
     - **Reference:** A field that links to a record in another table (e.g., Assigned To field refers to a user in the `sys_user` table).
     - **Choice:** A predefined list of options (e.g., Priority with values like High, Medium, Low).
     - **Boolean:** Stores true/false values.
3. **Field Properties:**
   - **Mandatory:** A field can be marked as mandatory, meaning it must be filled out before a record can be saved.
   - **Read-Only:** Fields can be set to read-only to prevent users from modifying them.
   - **Default Value:** Fields can have default values that automatically populate when a record is created.

4. **Field Inheritance:**
    - Fields in extended tables inherit properties from the base table. For example, the `Incident` table inherits fields such as `Priority` and `Assigned To` from the `Task` table.

Table and Field Relationships

1. **Reference Fields:**
    - A **reference field** creates a relationship between tables by allowing one table to store a reference to a record in another table. For example, the `Assigned To` field in the `Incident` table is a reference to the `sys_user` table, linking the incident to the user handling it.
2. **One-to-Many Relationships:**
    - In this type of relationship, one record in a table can be related to many records in another table. For example, a user in the `sys_user` table can be related to many incidents in the `incident` table.
3. **Many-to-Many Relationships:**
    - This type of relationship allows multiple records in one table to be related to multiple records in another table. ServiceNow supports this through special relationship tables.
4. **Dependent Fields:**
    - Some fields can be dependent on the value of other fields. For instance, a **Subcategory** field might only become available after selecting a specific **Category** in a request form.

Table and Field Management

1. **Table Customization:**
    - Administrators can create new custom tables and fields to store specific data relevant to their organization.
    - Existing tables and fields can also be customized by adding new fields, changing field properties, or creating relationships with other tables.
2. **Data Dictionary:**
    - The **Data Dictionary** is a key part of ServiceNow's architecture. It defines the schema (structure) of tables and fields, including data types, default values, and relationships.
3. **Business Rules and Client Scripts:**
    - Tables and fields can be enhanced with **business rules** (server-side scripts) and **client scripts** (browser-side scripts) to enforce data policies, validate inputs, and automate actions based on specific conditions.

ServiceNow Access Control List (ACL)

The **Access Control List (ACL)** in ServiceNow is a set of rules that define what data users can access and what actions they can perform on that data within the platform. ACLs ensure that users only see or modify information they are authorized to access, maintaining data security and integrity across the system.

1. **Access Control Rules:**
    - An **Access Control Rule** specifies the permissions a user or group has to interact with a specific table, field, or record. The rule can control the following actions:
        - **Create:** Ability to add new records to a table.
        - **Read:** Ability to view records in a table.
        - **Write:** Ability to modify existing records in a table.
        - **Delete:** Ability to remove records from a table.
2. **Types of Access Control:**
    - **Table-level Access:** Defines permissions for an entire table.
    - **Field-level Access:** Controls access to specific fields within a table. For example, a user might be able to view a record but not see sensitive fields like salary or social security number.
    - **Record-level Access:** Rules can be applied to specific records based on conditions, such as the record owner or a particular field value.
3. **Conditions and Scripts:**
    - ACLs can be defined using **conditions** (e.g., "if Priority is high") or **scripts** (JavaScript) to further refine the rules. These scripts allow administrators to create complex access control logic.
4. **Evaluating Access Controls:**
    - **Roles:** Access is often determined based on the **roles** assigned to users. For example, users with the "admin" role might have full access to all tables, while users with a "catalog_user" role can only access the service catalog.
    - **Permissions Hierarchy:** When a user tries to access a record, the system evaluates ACL rules from most specific (field-level) to most general (table-level). A user must pass all relevant rules to gain access.
    - **User Authentication:** When a user logs in, ServiceNow evaluates their roles and access permissions, determining what data they can interact with.
5. **Access Control Inheritance:**
    - **Extended Tables:** When a table extends another table, the child table inherits the ACL rules of the parent table. For example, the `Incident` table inherits access controls from the `Task` table.
6. **Order of Execution:**
    - ServiceNow evaluates access control rules in a specific order: first, it checks **table-level rules**, then **field-level rules**. If the user fails any of the rules in the evaluation, access is denied.

Components of an Access Control Rule

1. **Name:** Specifies the table and field the ACL applies to (e.g., `incident.number` or `incident.*`).
2. **Operation:** Defines what action is controlled (Create, Read, Write, Delete).
3. **Roles:** Lists the user roles required to pass the rule.
4. **Condition:** Optional conditions that further refine access (e.g., only allow access if the state is "Closed").
5. **Script:** Custom server-side logic written in JavaScript to enforce more complex conditions.

Data Import

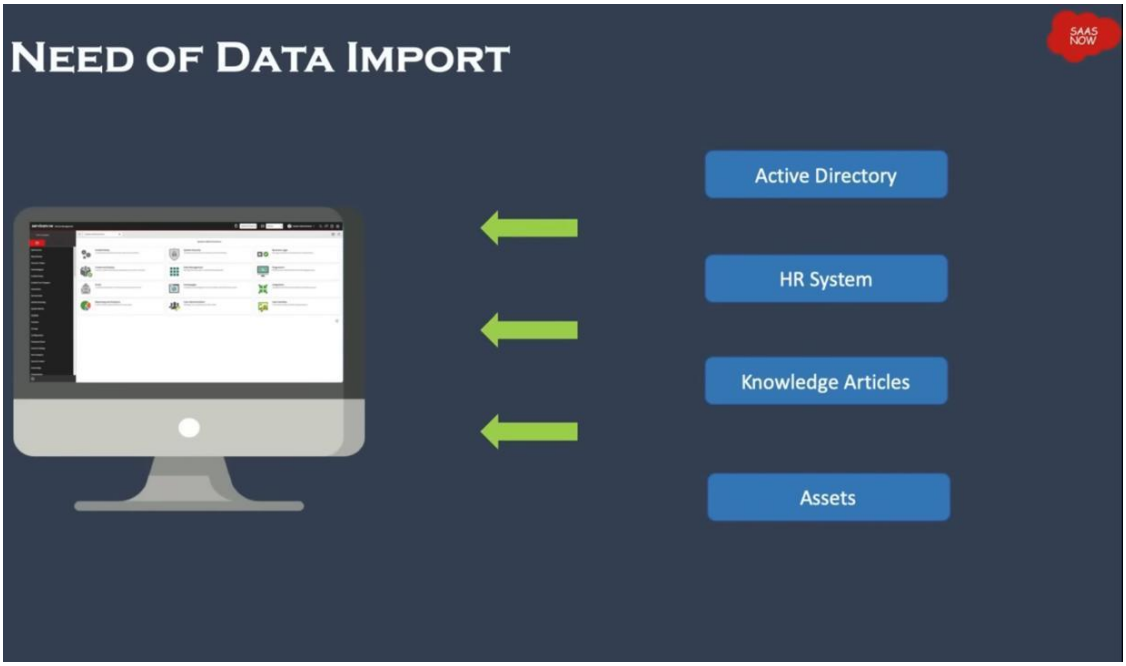
Data Import in ServiceNow is the process of bringing external data into the platform to populate tables. This feature allows administrators to import data from various sources such as spreadsheets, databases, or external applications and map it into ServiceNow's tables. It's commonly used for migrating data, updating existing records, or integrating with other systems.

1. **Data Source:**
    - A **data source** defines the location of the data you want to import into ServiceNow. Common data sources include:
        - **CSV files:** Import data from spreadsheets.
        - **Excel files:** Import from `.xls` or `.xlsx` files.
        - **XML files:** Import data in XML format.
        - **JDBC:** Connect to external databases and retrieve data.
        - **Web Services:** Import data through SOAP or REST APIs.
    - Administrators configure the data source by specifying the file or connection details.
2. **Import Set:**
    - **Import Sets** are temporary tables where incoming data is staged before it is transformed and moved to the target table.
    - Data is first imported into the import set, where it can be reviewed, mapped, and cleaned before being moved to the actual destination table.
3. **Transform Maps:**
    - **Transform Maps** define the rules for how data from the import set is transferred to the target table.
    - Administrators map fields from the import set to fields in the target table to ensure the data is correctly inserted into the appropriate columns.
    - **Field Mapping:** Data in an import set can be automatically mapped or manually mapped to the correct fields in the target table.
4. **Coalesce Fields:**
    - Coalescing fields are used to identify whether an incoming record is new or should update an existing record.

    - When a field is marked as a **coalesce field**, ServiceNow checks whether a matching record exists in the target table. If a match is found, the record is updated; otherwise, a new record is created.
    - Multiple fields can be coalesced to ensure unique record identification.
5. **Data Transformation:**
    - **Transformation** is the process of moving data from the import set to the target table.
    - During transformation, scripts can be used to manipulate the data, perform validation, or ensure that the imported data conforms to business rules.
6. **Error Handling and Reconciliation:**
    - Any errors or issues encountered during the import process (e.g., missing required fields or data type mismatches) are logged and can be reviewed and corrected.
    - The platform provides tools to track and resolve import errors to ensure data quality.

Data Import Process Flow

1. **Create Data Source:** Define the location of the external data, such as uploading a CSV file or connecting to an external database.
2. **Load Data into Import Set:** The data is staged into a temporary import set table for review and mapping.
3. **Create Transform Map:** Define how the fields from the import set map to the target table's fields.
4. **Transform Data:** Execute the transform process to move the data from the import set to the destination table.
5. **Review and Correct Errors:** Identify and fix any errors that occurred during the import and transformation.
6. **Finalize the Import:** Once the data is imported and validated, it is fully available in the ServiceNow system.



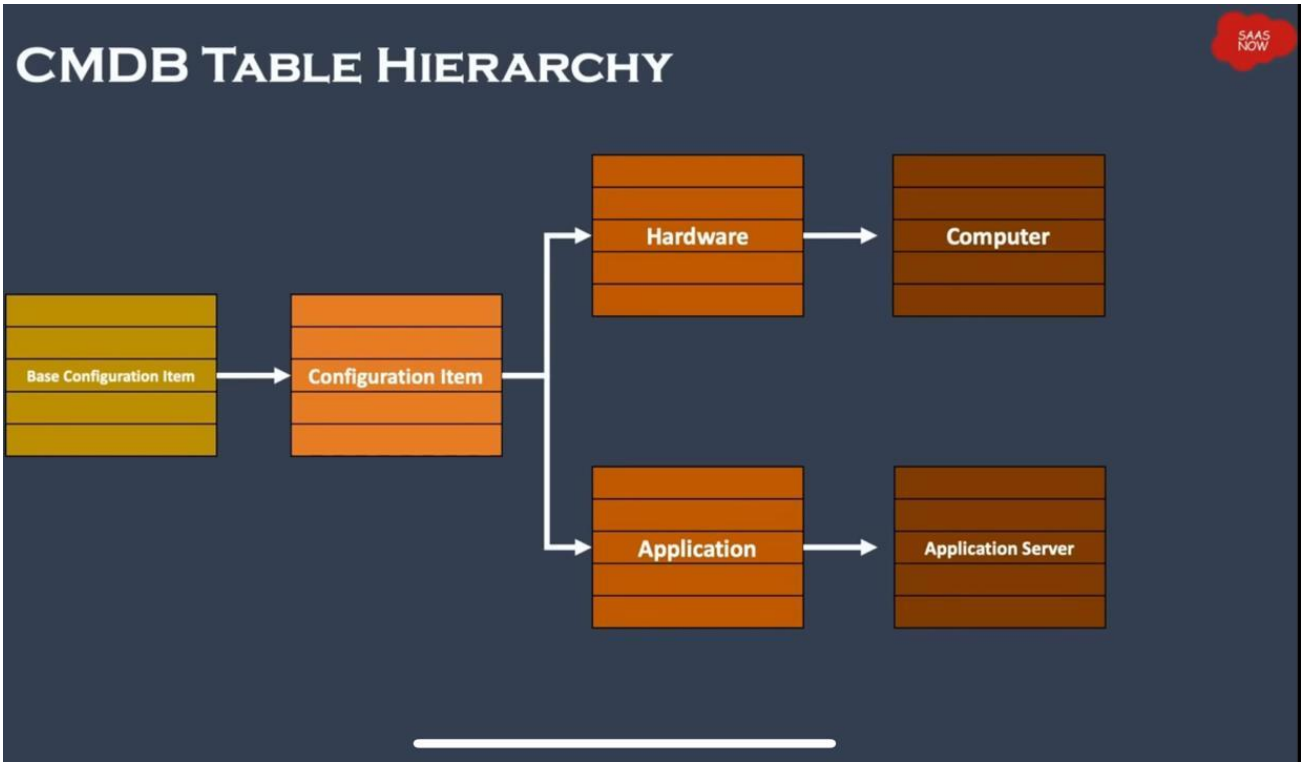
ServiceNow Configuration Management Database (CMDB)

The **Configuration Management Database (CMDB)** in ServiceNow is a centralized repository that stores information about all the IT assets, services, and their relationships within an organization. These assets, also known as **Configuration Items (CIs)**, can include anything from servers, software, network devices, databases, and applications to business services. The CMDB is crucial for effective IT service management (ITSM), as it provides a clear and organized view of the organization's infrastructure and services.

1. **Configuration Items (CIs):**
    - **CIs** represent the components within the IT environment, such as hardware, software, network devices, and even users or business services.
    - Each CI is a record in the CMDB with its own set of attributes (e.g., name, type, status, location).
    - CIs can be categorized by type (e.g., application, server, database) and grouped in different classes.
2. **CI Classes:**
    - CIs are organized into **classes** based on their type and purpose. For example, servers, routers, applications, and databases all belong to different classes.
    - Each class inherits attributes from its parent class, allowing for efficient classification and management of CIs.
3. **Relationships Between CIs:**

- The CMDB not only stores individual CIs but also tracks the **relationships** between them. For instance, a server might host several applications, or a network device might connect to multiple servers.
- These relationships help in understanding how changes or issues in one CI might impact other CIs, aiding in impact analysis and root cause analysis.

4. **CMDB Health:**
   - **CMDB Health** monitors the overall health of the CMDB data, ensuring its accuracy, completeness, and compliance with business rules.
   - It measures three main aspects: **Completeness**, **Correctness**, and **Compliance** of the data to ensure that the CMDB remains a reliable source of information.



ServiceNow Integration

ServiceNow provides robust integration capabilities that allow it to connect with other systems, applications, and services within an organization. This enables data exchange, process automation, and improved collaboration across various IT and business functions. ServiceNow supports both inbound and outbound integrations through various protocols and technologies, making it highly adaptable for different use cases.

Types of ServiceNow Integrations

1. **Inbound Integrations:**

- Data or requests come **into** ServiceNow from external systems.
- Examples: Importing data from HR systems, receiving incident tickets from monitoring tools, or pulling configuration items (CIs) into the CMDB.

2. **Outbound Integrations:**
   - ServiceNow sends data **out** to external systems.
   - Examples: Sending change requests to third-party change management systems, pushing incident data to communication platforms, or updating external asset management systems.

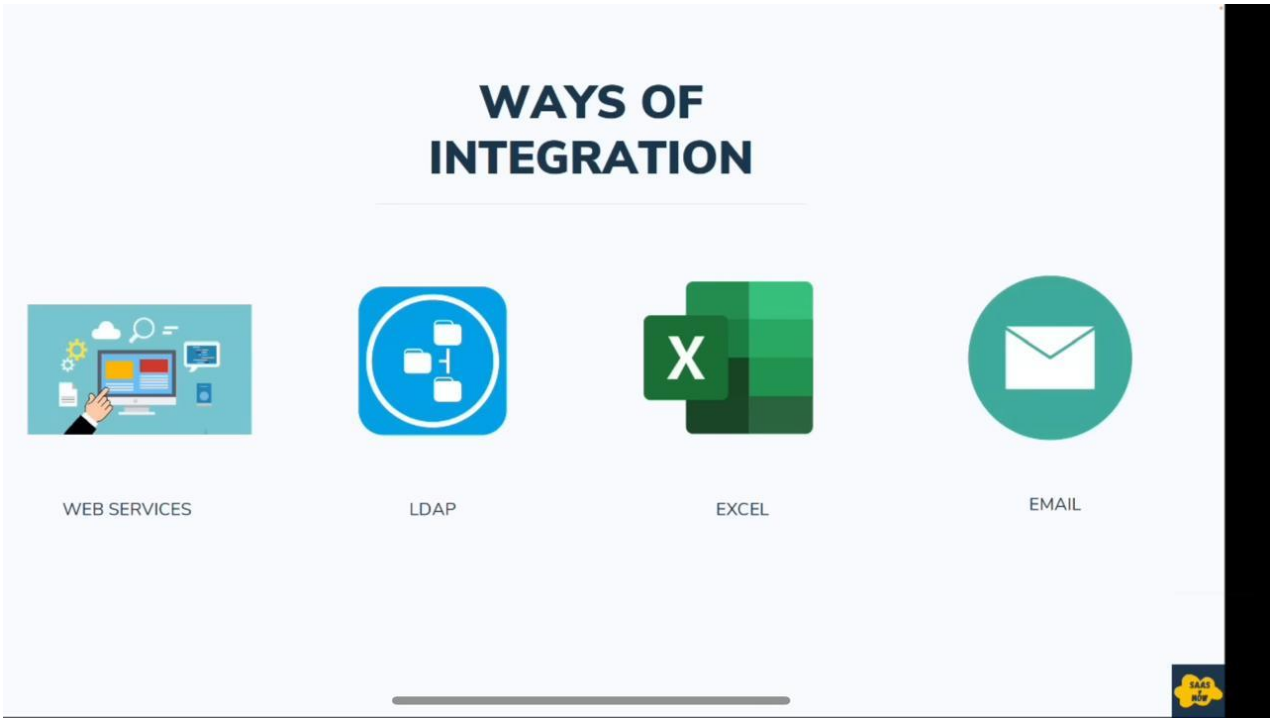
Integration Methods in ServiceNow

1. **APIs (Application Programming Interfaces):**
   - **REST API:** The REST (Representational State Transfer) API is one of the most common methods for integrating ServiceNow with other platforms. It allows systems to communicate over HTTP/HTTPS, sending and receiving data in JSON or XML format.
     - Example: Using REST API to send incident information from ServiceNow to a Slack channel or another ITSM tool.
   - **SOAP API:** ServiceNow also supports SOAP (Simple Object Access Protocol) for exchanging data. SOAP is more structured and often used in legacy systems where REST is not supported.
     - Example: Integrating with older financial systems or ERPs using SOAP to send and receive data.
2. **IntegrationHub:**
   - **IntegrationHub** is a low-code platform for building integrations without extensive coding knowledge. It provides out-of-the-box connectors (spokes) for popular systems like Salesforce, Microsoft Teams, AWS, and more.
   - It is part of the ServiceNow Flow Designer, enabling integration workflows through a graphical interface.
   - Example: Integrating ServiceNow with AWS for managing cloud infrastructure events or automating tasks in Microsoft Teams.
3. **MID Server:**
   - The **Management, Instrumentation, and Discovery (MID) Server** is a lightweight Java application installed in an organization's network. It facilitates secure communication between ServiceNow and systems behind firewalls, such as databases, ERP systems, and network devices.
   - Example: Using a MID server to integrate with on-premise databases, allowing ServiceNow to perform queries and pull data securely.
4. **Data Import and Export:**
   - **Import Set:** ServiceNow allows importing data from CSV files, Excel, XML, or other structured data sources through import sets.
   - **Export Options:** ServiceNow can export data into various formats (CSV, Excel, XML, JSON) or push data to external systems through APIs or file transfers.
5. **Webhooks:**

- Webhooks are HTTP callbacks that allow real-time communication between ServiceNow and other systems.
- Example: A monitoring tool can trigger a webhook to create an incident in ServiceNow when a critical threshold is reached, enabling automated incident creation.

6. **Email Integration:**
   - ServiceNow can integrate with email systems to send and receive emails. Emails can trigger actions such as creating or updating incidents, or notifications.
   - Example: Receiving an email from an external monitoring system, which creates a new incident record in ServiceNow.
7. **Third-Party Connectors:**
   - ServiceNow provides out-of-the-box connectors for popular third-party applications like Jira, Microsoft Azure, AWS, Salesforce, and more.

Example: Integrating with Jira to synchronize project management tasks or incident tickets between ServiceNow and Jira



ServiceNow Update Sets

An **Update Set** in ServiceNow is a tool used to capture, package, and move customizations from one instance of ServiceNow to another (e.g., from a development instance to a production instance). Update Sets track changes made to applications, configurations, or other custom

elements in ServiceNow, allowing for efficient deployment and version control of system modifications.

Key Concepts of Update Sets

1. **What an Update Set Captures:**
    - Update Sets track and capture changes such as:
        - **UI changes:** Forms, fields, views, lists, reports.
        - **Workflows and business rules.**
        - **Client scripts and server-side scripts.**
        - **Changes to table schema and field configurations.**
        - **Changes to roles, groups, and access controls (ACLs).**
        - **Custom applications and modules.**

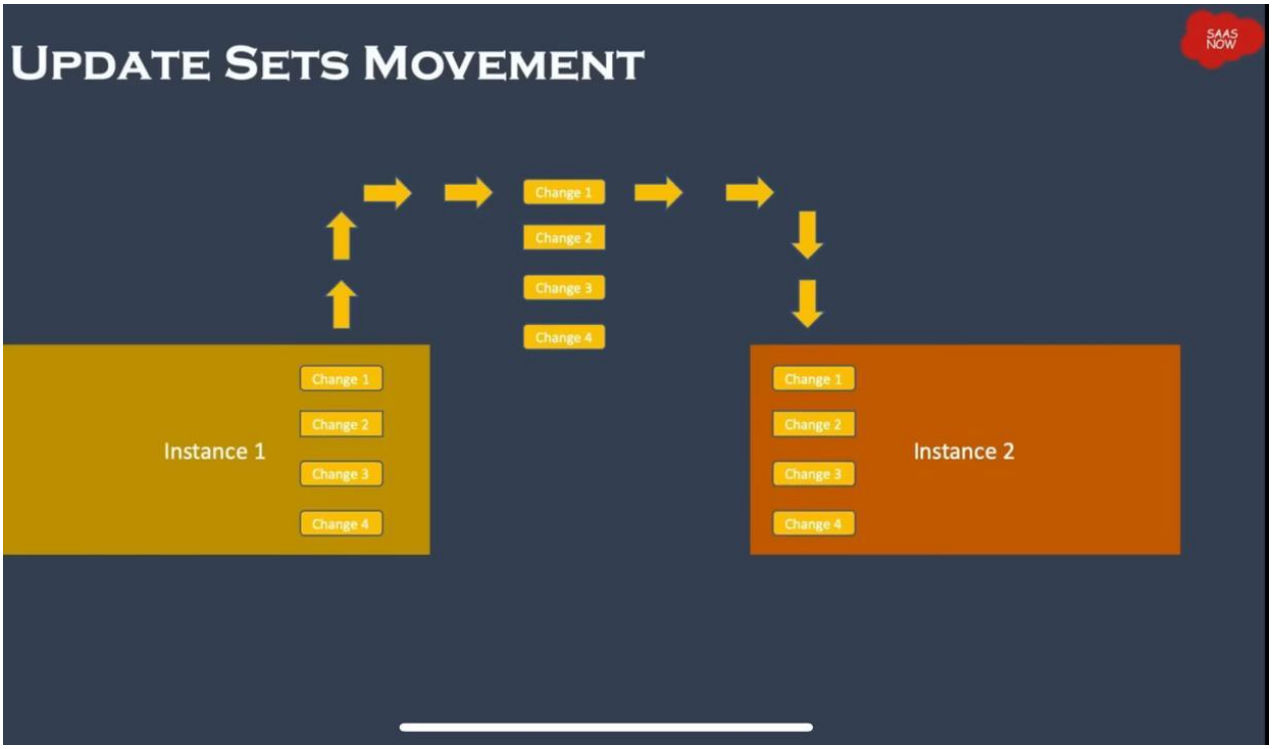
    Note: Not all data is captured by an Update Set, such as records and data itself (e.g., incidents, change requests, task data), nor changes to items like scheduled jobs and users.

2. **What an Update Set Does Not Capture:**
    - Changes related to:
        - **Data Records:** Incidents, change requests, task records.
        - **System properties:** Instance-specific configurations, such as system mail configurations.
        - **Schedules, users, and groups:** These typically require manual handling or separate migration.

Update Set Lifecycle

1. **Create an Update Set:**
    - In the development instance, an Update Set is created to begin capturing customizations. Only the active Update Set captures the changes.
2. **Make Changes:**
    - As changes are made to the instance (custom forms, workflows, scripts, etc.), those changes are automatically tracked and added to the active Update Set.
3. **Close and Complete Update Set:**
    - Once all changes are complete, the Update Set is **marked complete** to finalize it and stop capturing further updates. This also makes it ready for export or transfer.
4. **Move Update Set to Another Instance:**
    - The completed Update Set is exported from the development instance and moved to another instance (e.g., testing, production) via XML or a direct instance-to-instance transfer.
5. **Preview and Commit Update Set:**
    - In the target instance, the Update Set is **previewed** to check for any conflicts or errors (such as dependencies or collisions with existing elements).

- Once verified, the Update Set is **committed**, which applies all the captured changes to the target instance.



ServiceNow Events

In ServiceNow, **Events** are system activities or occurrences that are triggered when specific conditions are met. These events are typically used to respond to changes in the system by initiating automated actions such as notifications, script executions, or logging. Events allow for real-time monitoring and reaction to incidents, changes, or other activities in the ServiceNow environment.

1. **Event Generation:**
   - Events can be triggered automatically by system processes (e.g., a record update, incident creation, or workflow execution) or manually by scripts.
   - Common sources of event generation include:
     - Business rules.
     - Workflows and Flow Designer.
     - Scheduled jobs.
     - API calls.
2. **Event Registry:**
   - The **Event Registry** is where events are defined and configured in the system. Each event has:

        - **Name:** A unique identifier for the event (e.g., `incident.closed`, `change.requested`).
        - **Description:** Provides context about what the event represents.
        - **Queue:** Determines how the event is processed (e.g., default queue or a specific queue for high-priority events).
        - **Parameters:** Events can pass additional data (e.g., record ID, user details) to further customize actions.
3. **Event Queue:**
    - When an event is generated, it is placed in the **Event Queue**, where it waits to be processed.
    - The queue ensures that events are handled in the order they were generated, and it helps manage load by processing events asynchronously.
4. **Event Processing:**
    - Once an event is in the queue, it can trigger specific actions, such as:
        - **Notifications:** Sending emails, SMS, or push notifications based on the event.
        - **Scripts:** Running server-side scripts to perform actions like updating records or integrating with other systems.
        - **Logging:** Events can log information for auditing or monitoring purposes.

Common Event Triggers in ServiceNow

1. **Record-Based Events:**
    - Events are often triggered by changes to records. For example:
        - When an **incident is closed**, an event like `incident.closed` is generated.
        - When a **change request is created**, the `change.requested` event may trigger.
2. **Workflow and Flow Designer:**
    - Events can be triggered within **Workflows** or **Flow Designer** when specific conditions are met during the execution of a process (e.g., approval steps, task completions).
3. **Business Rules:**
    - Events are commonly triggered by **Business Rules**, which run when records are created, updated, or deleted. A business rule can be configured to trigger an event when certain conditions are met.
4. **Scheduled Events:**
    - **Scheduled jobs** can trigger events based on time-based conditions, such as daily reports, maintenance windows, or task reminders.

ServiceNow Platform Sets

In ServiceNow, **Platform Sets** are a way to group and manage a collection of related configuration elements, features, or settings within the platform. While not as commonly

referenced as other ServiceNow features like Update Sets or Application Scope, the concept of Platform Sets can be useful in organizing and deploying various platform configurations and customizations.

1. **Purpose of Platform Sets:**
    - **Organize Configurations:** Platform Sets help in grouping related configurations and customizations to streamline deployment and management.
    - **Manage Dependencies:** They allow administrators to manage dependencies and ensure that related configurations are deployed together.
    - **Simplify Deployment:** By bundling configurations into a single set, Platform Sets simplify the deployment process across different instances or environments.
2. **Components of Platform Sets:**
    - **Configuration Items:** Includes various elements like forms, fields, business rules, scripts, workflows, and other customizations.
    - **Application Scope:** Often tied to a specific application scope, ensuring that the configurations are managed and deployed within the context of that application.
    - **Version Control:** Provides versioning and tracking of changes within the set, enabling rollback or updates as needed.
3. **Deployment and Management:**
    - **Export and Import:** Platform Sets can be exported from one instance and imported into another. This is useful for moving configurations from development to testing or production environments.
    - **Preview and Test:** Administrators can preview Platform Sets before deployment to check for conflicts or dependencies, ensuring that the deployment will be successful.
    - **Rollbacks:** Allows for easy rollback of configurations if something goes wrong during deployment, helping maintain system stability.