Calculate Family Expenses Using ServiceNow Personal Developer Instance

This presentation outlines a project to leverage ServiceNow for personal finance management. We will address the challenges of traditional expense tracking and propose a custom application on a ServiceNow Personal Developer Instance (PDI) to provide families with enhanced financial visibility and control.

The Challenge of Family Expense Management

Multiple Transactions

The average UK family handles over 150 transactions monthly, making manual tracking cumbersome and prone to errors.

Budgeting Difficulty

A significant 42% of UK adults do not have a formal budget, leading to reactive rather than proactive financial management.

Lack of Visibility

Families often lack a real-time overview of their spending patterns, making it hard to identify overspending.

Manual Errors

Reliance on spreadsheets or paper for expense tracking introduces the risk of input mistakes and data inconsistencies.

Financial Stress

With 6 in 10 UK adults concerned about rising living costs, efficient expense management is crucial for peace of mind.

Made with GAMMA

Why ServiceNow Personal Developer Instance?

Free & Dedicated

A readily available, dedicated instance for personal development and testing without any cost implications.

Robust Capabilities

Leverage powerful features like workflow automation, reporting, and custom application development.

Low-Code/No-Code

Rapidly build applications with minimal coding, enabling quick customisation and deployment.

Secure & Scalable

Benefit from an enterprise-grade foundation, ensuring data security and scalability for personal use.

Versatile Data Management

Ideal for managing structured expense data, categories, and custom financial fields efficiently.

Made with GAMMA

Solution Architecture & Key Features

Custom Application Module

A dedicated "Family Expenses Tracker" module will be created to house all functionalities.

Core Tables

Establish 'Expense Item' (amount, date, category, payer), 'Category' (e.g., groceries, utilities), and 'Family Member' tables.

Intuitive Forms

Streamlined data entry forms will allow family members to quickly log each expense transaction.

Interactive Dashboards

Visualisations will display spending by category, month, and individual family member for quick insights.

Automated Reporting

Scheduled monthly expense summaries will be automatically sent via email notifications.

Workflows

Optional approval workflows can be implemented for high-value expenses, such as those exceeding £500.

Made with GAMMA

Practical Application & Data Flow

User Input

Family members easily log expenses through user-friendly forms on the ServiceNow PDI portal.

Automated Categorisation

Expenses are consistently categorised using predefined labels like 'Groceries' or 'Utilities'.

Real-time Aggregation

Dashboards and reports update immediately upon expense entry, providing current financial insights.

Monthly Summaries

Automated reports detail total spend per category, e.g., £450 on 'Groceries', £120 on 'Utilities'.

Trend Analysis

Identify spending patterns over time, such as a 15% increase in 'Entertainment' during Q3.

Made with GAMMA

Key Benefits & Impact

15%

Potential Savings

Achieve up to 10-15% savings through data-driven budgeting and cost control.

80%

Effort Reduction

Significantly reduce time spent on manual tracking, freeing up valuable family time.

100%

Data Accuracy

Eliminate spreadsheet errors with automated processes, ensuring precise financial data.

- **Enhanced Financial Visibility:** Gain clear, real-time insights into family spending patterns.
- **Improved Budgeting:** Make data-driven decisions for better cost control and potential savings.
- **Reduced Manual Effort:** Automation frees up time previously spent on spreadsheets or paper tracking.
- **Data-Driven Decisions:** Easily identify areas for optimisation, such as reducing 'Dining Out' by £50/month.
- **Customisable & Scalable:** The solution can easily adapt to changing family needs or allow for new feature additions.

Made with GAMMA

Future Enhancements & Integration

Mobile Access

Develop a dedicated mobile app interface for convenient, on-the-go expense logging from any device.

Banking API Integration

Implement automated import of transactions from bank accounts, using mock or real data for seamless tracking.

Advanced Analytics

Incorporate features like predictive spending, anomaly detection, and automated budget overrun alerts for proactive financial management.

Recurring Expense Management

Automate the tracking of regular payments such as subscriptions, rent, and mortgage payments.

Receipt Management

Enable the attachment of scanned receipts directly to expense records for easier auditing and record-keeping.

Made with GAMMA

Conclusion & Call to Action

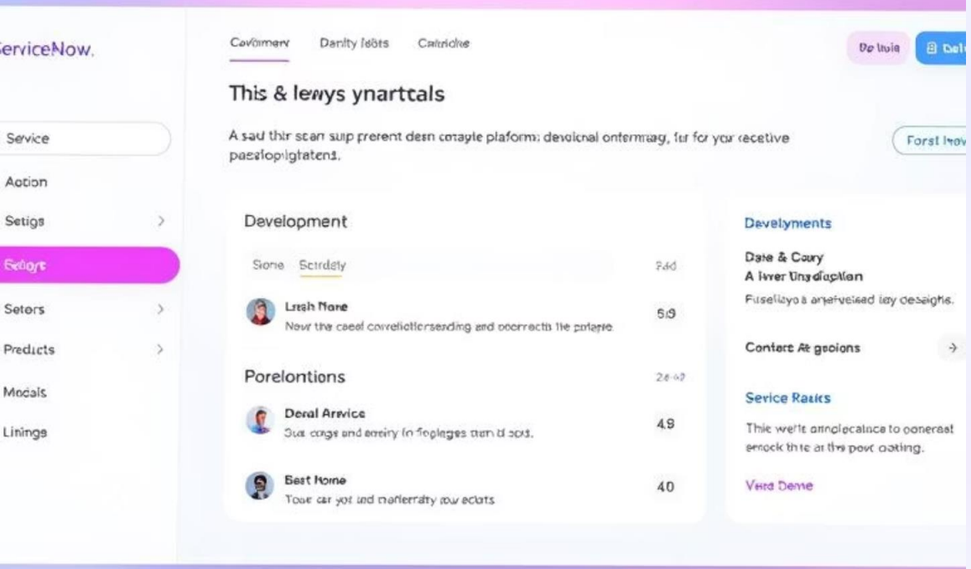
The ServiceNow PDI offers a powerful, flexible, and customisable solution for family expense management, empowering families to gain control and clarity over their financial health. This project also serves as a practical opportunity for exploring the extensive capabilities of the ServiceNow platform.

Implement the Solu...

Learn More About ServiceNo...

Q&A / Contact

Made with GAMMA



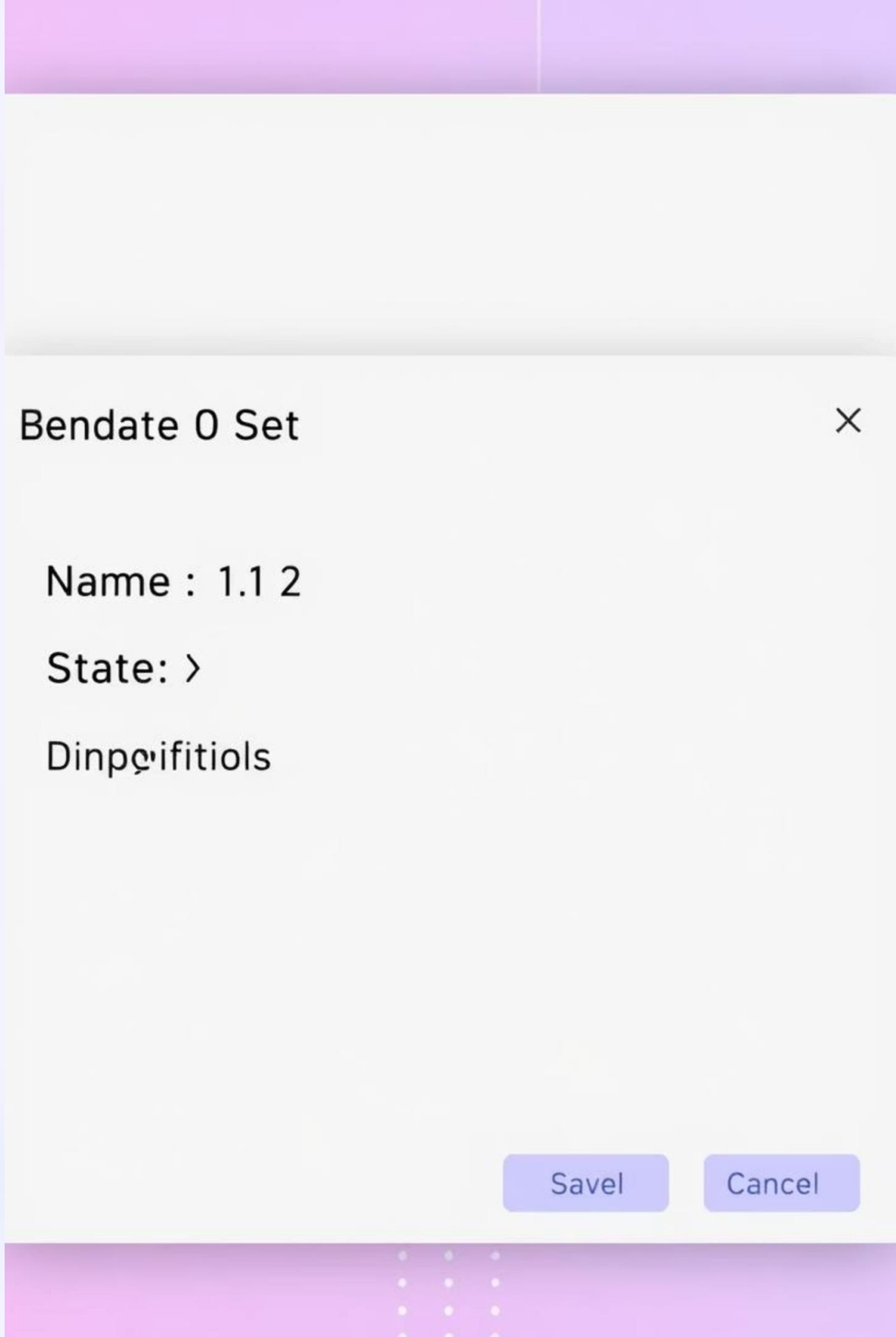
Creating a New Update Set in ServiceNow

Essential for controlled development and deployment. Core to 90% of platform configuration changes. Streamlines moving customisations between instances.

Made with GAMMA

What is a ServiceNow Update Set?

A collection of configuration changes recorded as XML, capturing customisations for transfer between instances. This ensures consistency across development, test, and production environments, preventing manual re-work and reducing human error.



Why Use Update Sets?

- **Controlled Deployment**: Predictable, repeatable transfers of configurations.
- **Version Control**: Allows tracking of changes and easy rollback if needed.
- **Collaboration**: Multiple developers can contribute to a single project or feature.
- **Risk Mitigation**: Reduces potential for human error and system outages by 70%.
- **Audit & Compliance**: Provides a clear trail of all system modifications.

Made with GAMMA

Steps to Create a New Update Set

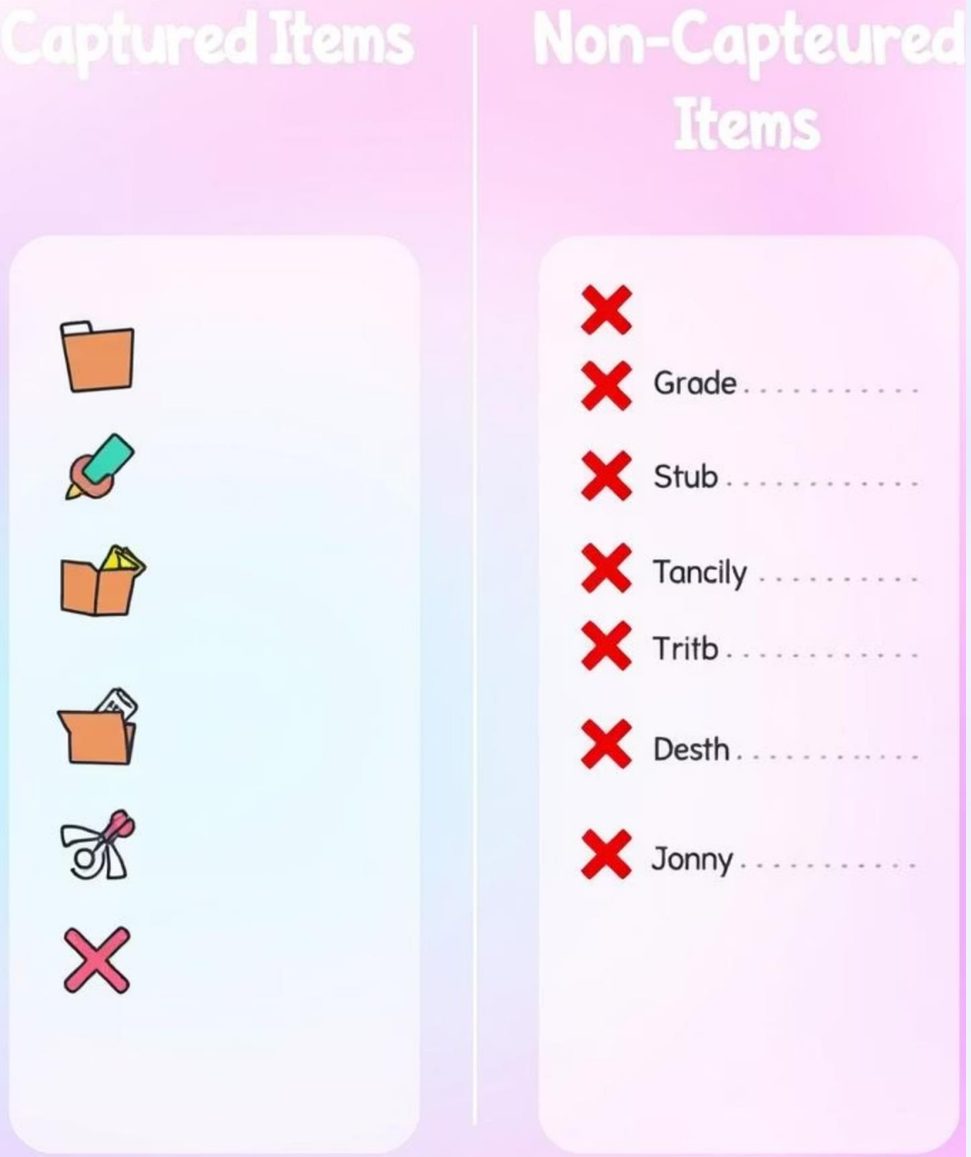
Navigate to `System Update Sets > Local Update Sets` in the instance.

Click the 'New' button to create a new record.

- **Name**: Provide a clear, descriptive name (e.g., "HR Portal V2.0 - Request Forms").
- **State**: Set to 'In Progress'.
- **Description**: Detail the purpose and scope of the changes for future reference.

Click 'Submit and Make Current' to begin capturing changes.

Made with GAMMA



What Gets Captured (and What Doesn't)

Captured: Business Rules, UI Policies, Workflows, ACLs, Script Includes, Service Catalog Items, Flow Designer Flows, Notifications, Form Layouts.

NOT Captured: Data records (users, groups, CI records), Scheduled Jobs (unless custom script), Homepage content, Reports, some System Properties.

Note: Use 'Export XML' for data or specific record transfers.

Best Practice: Always verify the contents of an Update Set before export.

Made with GAMMA

Working with an Update Set

Current Set: Always ensure the correct update set is 'Current' before making changes.

Development: Perform all configurations and customisations within the selected update set.

Avoid Default: Never make significant changes in the 'Default' update set.

Completion: Mark an update set as 'Complete' only when all related changes are finalised and tested.

Immutability: Once 'Complete', never modify the update set directly.

Made with GAMMA

Moving and Committing Update Sets

Export: Set update set to 'Complete', then 'Export to XML' from the related link.

Import: In the target instance, navigate to `Retrieved Update Sets` and 'Import Update Set from XML'.

Preview: Crucial step to identify potential conflicts before committing (e.g., 95% of conflicts caught here).

Commit: Apply the changes to the target instance. Monitor progress and review 'Preview Problems' and 'Commit History' logs.

Post-Commit: Test thoroughly in the target environment (e.g., UAT, Production).

Made with GAMMA

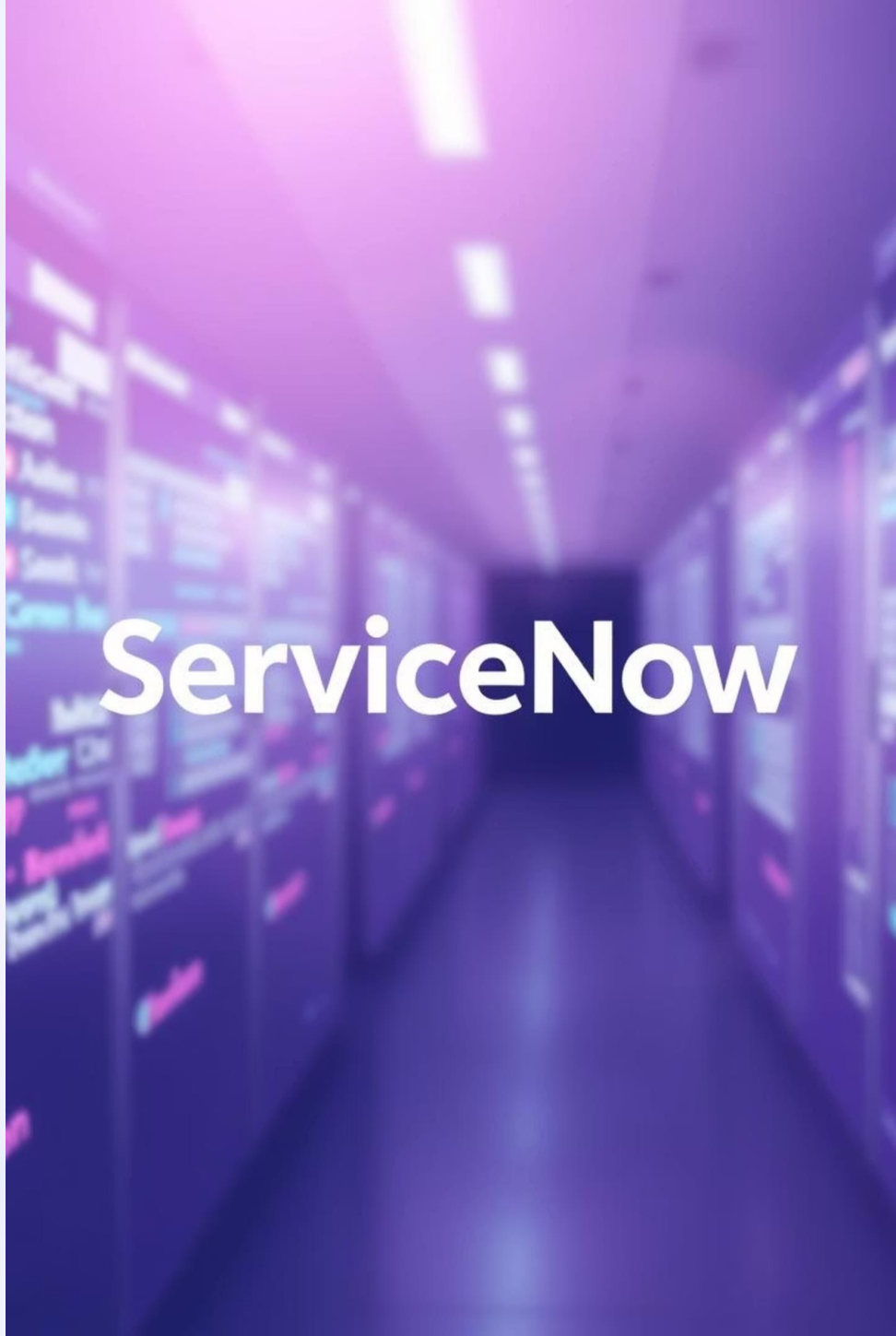
Best Practices for Update Sets

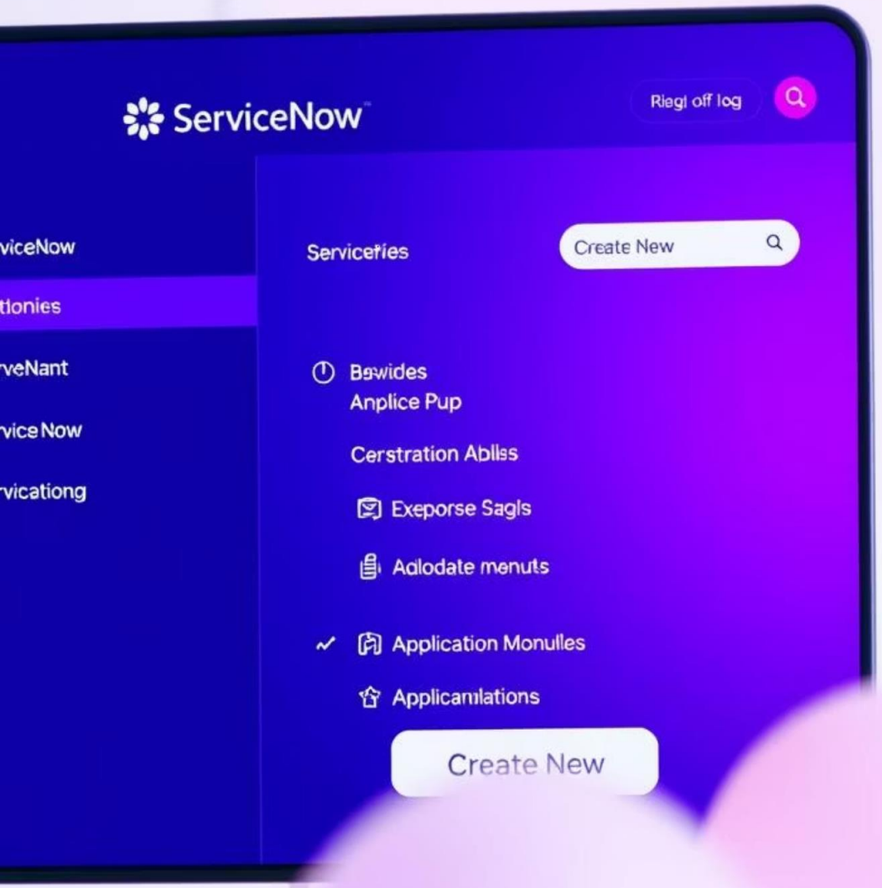
1. **Granularity**: Create small, logical update sets (e.g., one feature or bug fix per set).
2. **Naming Conventions**: Adopt a consistent, descriptive naming standard (e.g., `INC0012345_FixAssignmentGroup`).
3. **Coordination**: Avoid overlapping development with other team members to prevent conflicts.
4. **Regular Commits**: 'Complete' and export update sets frequently as milestones are met.
5. **Source Control**: Consider integrating with external source control for advanced versioning.
6. **Testing**: Always perform comprehensive testing in target non-production instances before deploying to Production.

Made with GAMMA

The ServiceNow Creation Tab: Empowering Efficient Service Delivery

The ServiceNow Creation Tab is a critical interface for initiating new records, streamlining IT and Enterprise Service Management (ESM) workflows. It's a key component of the Now Platform's user experience, designed to enhance efficiency and accuracy in service delivery.





Introduction to the Creation Tab

The Creation Tab can be accessed via the "Create New" module or the "New" button found on various forms within ServiceNow. It serves as the primary gateway for formalising service requests and incidents, centralising record initiation for over 50 core applications. This feature can reduce manual data entry by up to 30% through built-in automation capabilities.

Made with GAMMA

Core Record Types Enabled

The Creation Tab facilitates the initiation of several vital record types, each serving a distinct purpose in service management:

1. **Incident**

   Log unexpected service disruptions (e.g., "Email is down for 50 users").

2. **Change Request**

   Formalise planned alterations (e.g., "Deploy server patch to 20 machines").

3. **Service Catalogue Request**

   Order predefined services (e.g., "New laptop provision for new hire").

4. **Problem**

   Document root cause analysis (e.g., "Persistent database error impacting 3 applications").

5. **Knowledge Article**

   Create solution documentation (e.g., "FAQ on VPN connection issues").

Made with GAMMA

Streamlined Workflow & Automation

The Creation Tab leverages various features to optimise workflows:

Tailored Form Views

Specific layouts are presented to users based on their roles, ensuring relevance and ease of use.

Client Scripts & UI Policies

Real-time form validation and dynamic field display or hiding based on user input enhance the user experience.

Template Utilisation

Templates can auto-populate up to 70% of fields for common requests, saving significant time.

Automated Routing

Incidents are automatically assigned to the correct support group, such as Level 1 Desktop Support.

Made with GAMMA

Business Benefits & Value

Implementing the Creation Tab effectively yields significant business advantages:

45s

Operational Efficiency

Reduces average record creation time.

90%

Data Quality

Increases data consistency through mandatory fields.

100%

Compliance & Governance

Ensures adherence to defined service processes.

Excellent

Enhanced User Experience

Provides an intuitive interface for all service consumers.

Low

Reduced Training Overhead

Minimises user errors with self-explanatory forms.

Made with GAMMA

Customisation for Specific Needs

The Creation Tab is highly customisable to meet unique business requirements:

UI Actions

Customise the "New" button behaviour for different tables to suit specific workflows.

Form Layouts

Reorganise fields to align with specific business processes and user needs.

Record Producers

Create simplified front-end forms that map to complex back-end records.

Templates

Design pre-filled forms for common, repeatable tasks to ensure consistency.

Access Control

Define user permissions, ensuring only authorised personnel can create specific record types.

Made with GAMMA

Best Practices for Optimised Use

To maximise the effectiveness of the Creation Tab, adhere to these best practices:

1. **Keep Forms Simple**

   Minimise the number of required fields to reduce user friction and prevent abandonment.

2. **Leverage Templates Extensively**

   Standardise common requests with pre-defined templates, aiming for 10-15 per department.

3. **Use Clear Labelling**

   Employ user-friendly language, avoiding technical jargon for better comprehension.

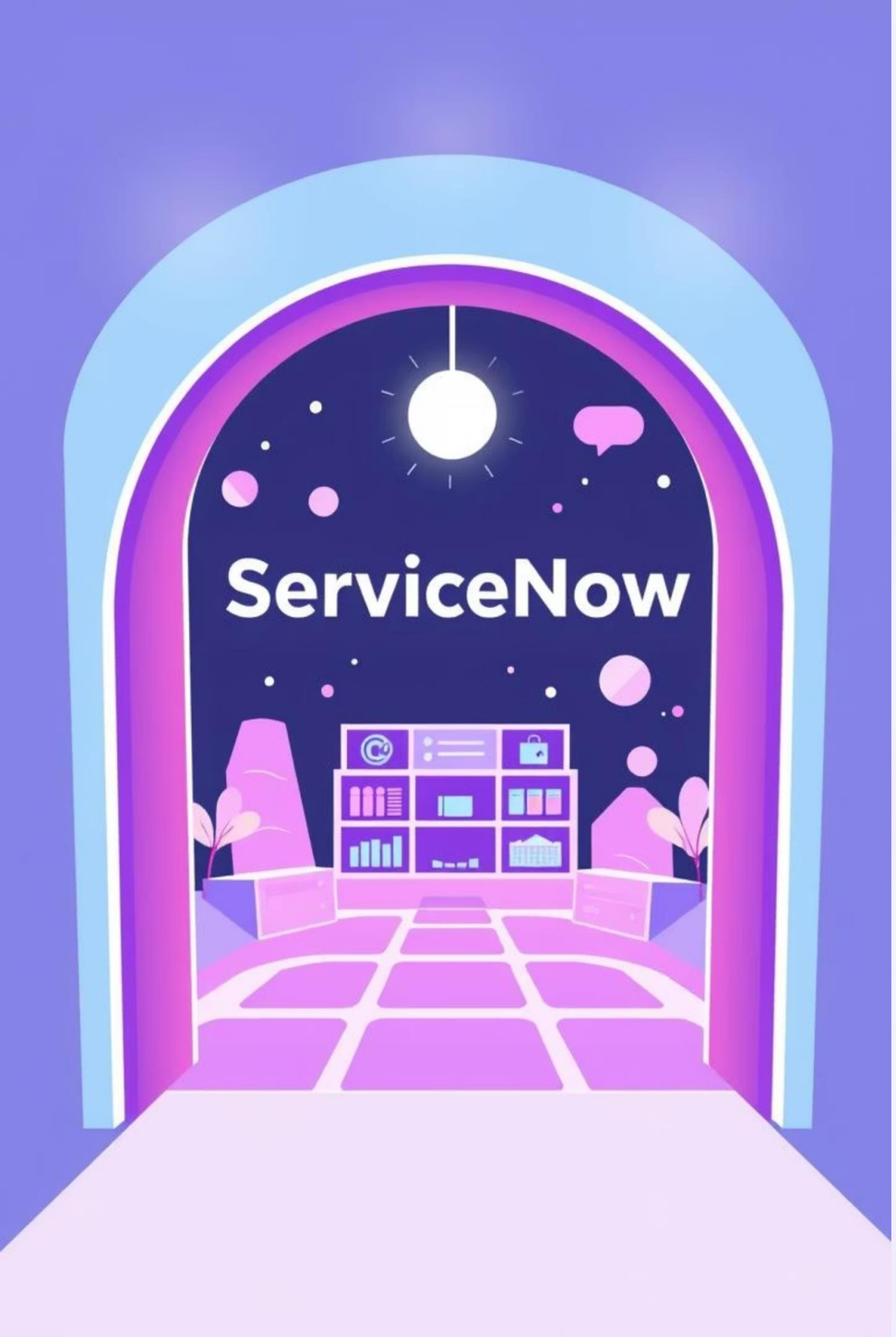
4. **Conduct Regular Reviews**

   Audit and update forms and workflows quarterly to maintain relevance and efficiency.

5. **Provide User Training**

   Offer concise guides and resources to educate users on effective record creation.

Made with GAMMA



Conclusion: The Gateway to Service Excellence

The ServiceNow Creation Tab is indispensable for efficient and consistent service delivery within the platform. It drives standardisation and automation across the enterprise, enabling the rapid and accurate initiation of critical IT and business records. This foundational element contributes directly to measurable improvements in key service metrics, such as First Call Resolution and Mean Time to Resolution, ultimately fostering service excellence.

Made with GAMMA

Unlocking IT Clarity: Relationships in ServiceNow CMDB

Relationships are essential for effective IT service management (ITSM), connecting Configuration Items (CIs) within the CMDB to support operational efficiency and reduce risk.

What Defines a ServiceNow Relationship?

A relationship in ServiceNow is a defined link between two Configuration Items (CIs). It precisely describes how these CIs interact with or depend on each other. For instance, the "Runs on::Runs on by" relationship clearly indicates that an Application CI runs on a Server CI. ServiceNow offers over 200 out-of-the-box relationship types, providing a standardised way to map these connections.

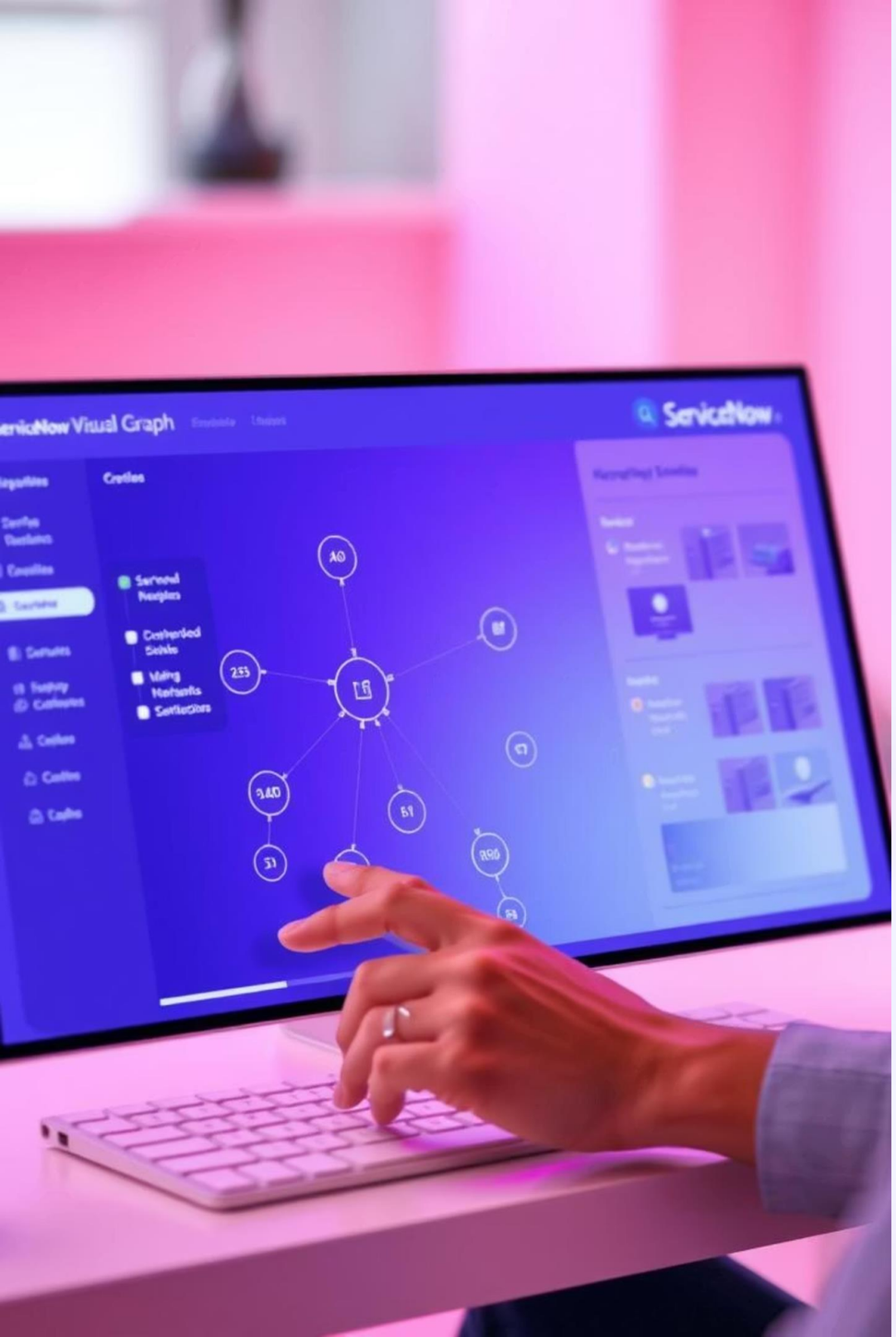
Made with GAMMA

Why Relationships are Critical for Your IT Estate

Establishing robust relationships within the CMDB is vital for several reasons. It enables accurate impact analysis when changes are planned, significantly accelerating root cause identification for issues by up to 40%. Furthermore, it provides a clear visualisation of service dependencies, crucial for effective Service Mapping, and ultimately improves the overall integrity and usefulness of your CMDB data.

Common Relationship Types and Their Purpose

- **Runs on::Runs on by**: Used to link an Application CI to the Server CI it runs on.
- **Connected by::Connects to**: Ideal for network devices, showing how a switch connects to a router, for example.
- **Contained by::Contains**: Useful for illustrating the relationship between physical hardware components, like a server within a rack.
- **Depends on::Used by**: Connects a Business Service to the Applications that support it, showing critical dependencies.



Manual Relationship Creation Methods

Relationships can be manually created using the intuitive CI Relationship Editor, a visual tool designed for linking CIs. Alternatively, related CIs can be added directly from individual CI forms. These methods are best suited for static, unique, or less frequently changing links. However, they require diligent manual input and a solid understanding of the different CI types involved.

Made with GAMMA

Automated Relationship Discovery

ServiceNow Discovery automatically scans your network, populating CIs and their relationships using probes and sensors (like SSH and SNMP). This process automates over 90% of infrastructure relationship population. Service Mapping further refines this by discovering application dependencies through traffic flows, creating critical relationships for business services. Import Sets and REST APIs also facilitate bulk or programmatic relationship creation.

Made with GAMMA

Best Practices for Relationship Management

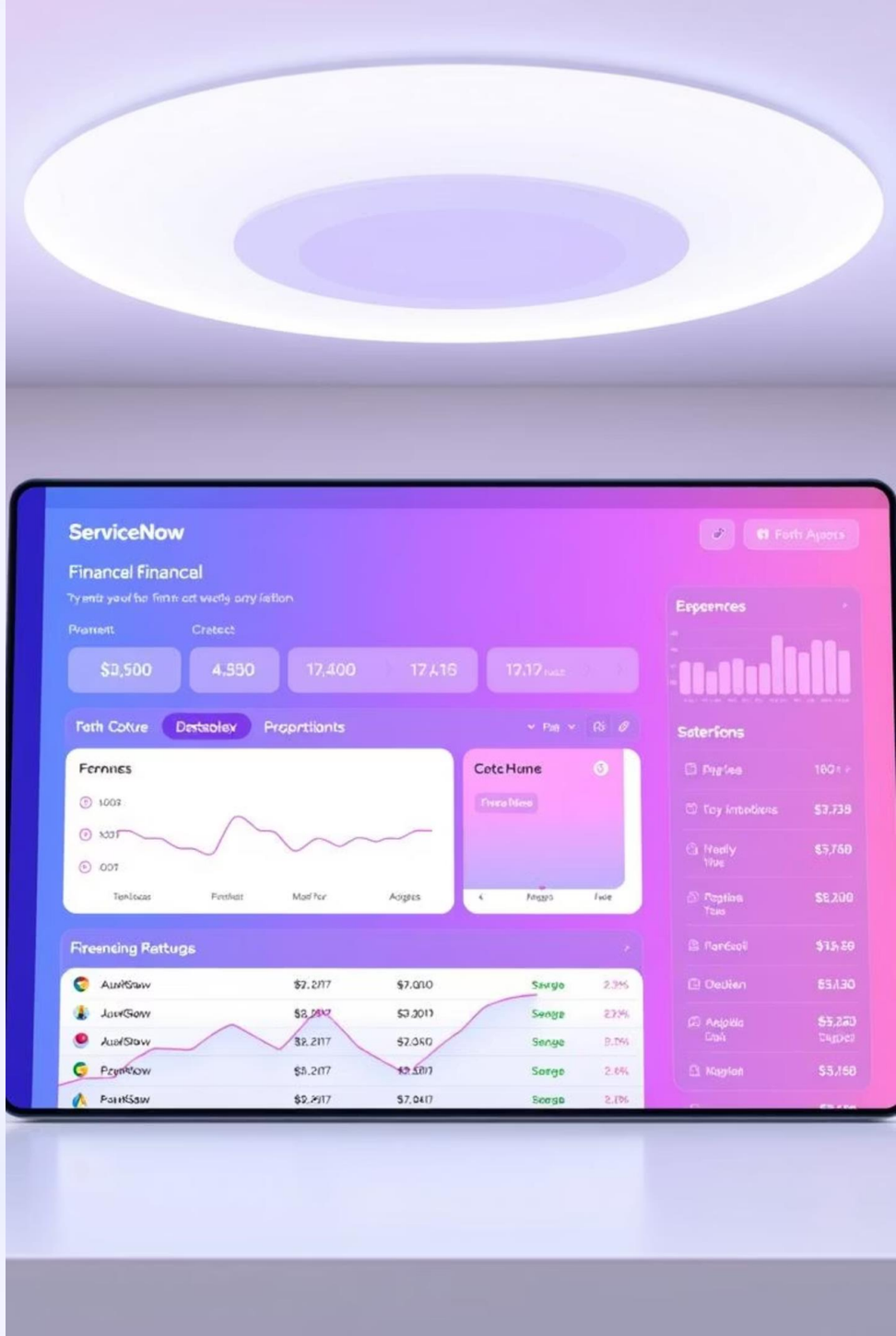
- **Standardise Usage**: Prioritise using out-of-the-box relationship types for consistency.
- **Automate First**: Leverage ServiceNow Discovery and Service Mapping for efficient population.
- **Validate Data**: Regularly use the CMDB Health dashboard to ensure accuracy.
- **Assign Ownership**: Establish clear accountability for the maintenance of CI relationships.
- **Regular Audits**: Conduct audits quarterly to maintain data accuracy and relevance.

Made with GAMMA

Configuring Related Lists for Family Expenses in ServiceNow

This presentation will guide you through the process of setting up related lists in ServiceNow to effectively manage family expenses.

We will cover the benefits of this approach, the underlying table relationships, and the step-by-step configuration process, enabling enhanced financial tracking and oversight within the platform.

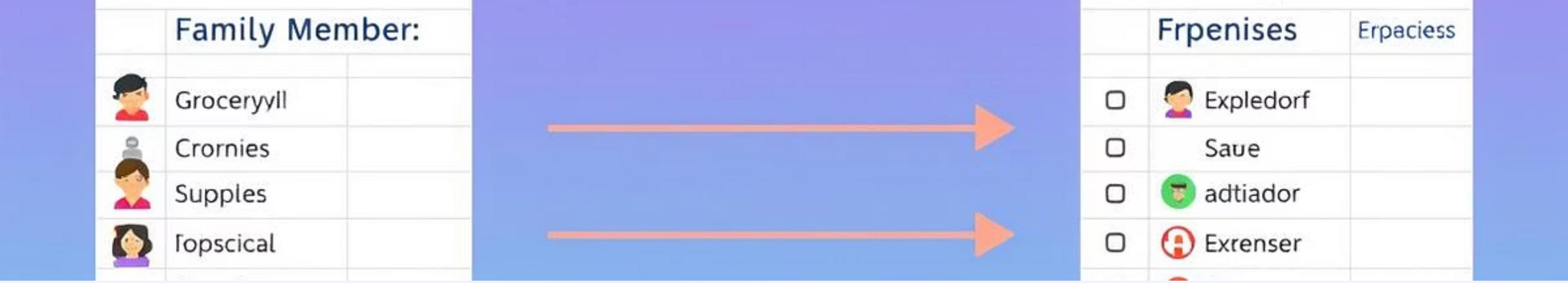


Why Related Lists for Family Expenses?

Managing household finances often involves scattered data, making it difficult to gain a clear overview. By implementing related lists for family expenses in ServiceNow, we centralise this information.

This allows for a consolidated view of individual expenses directly linked to each family member's profile, improving budgeting, streamlining financial reporting, and enhancing accountability. For instance, you can easily track a child's allowance or a spouse's specific spending categories.

Made with GAMMA



Understanding ServiceNow Table Relationships

At the heart of related lists lies the concept of table relationships within ServiceNow. A related list is essentially a way to display records from one table (the child table) that are linked to a record in another table (the parent table).

This linkage is typically established through a reference field. For example, an "Expense Item" record in a child table might contain a reference field pointing to a "Family Member" record in the parent table, creating a one-to-many relationship. This ensures data integrity and logical connections.

Made with GAMMA

Step 1: Identifying & Creating Core Tables

The first step involves defining the necessary data structures. You can either utilise the existing ServiceNow User table (`sys_user`) to represent family members, or create a custom table, such as `u_family_member`, for more tailored attributes.

Next, a new custom table, for example `u_family_expense`, will be created to store individual expense details. This table should include key fields such as:

- `u_member`: A reference field linking to the family member.
- `u_amount`: To store the expense value (using a currency type).
- `u_category`: A choice list for expense types (e.g., Groceries, Utilities).
- `u_date`: To record the date of the expense.
- `u_description`: A string field for a brief description.

Made with GAMMA

Step 2: Defining the Relationship via Reference Field

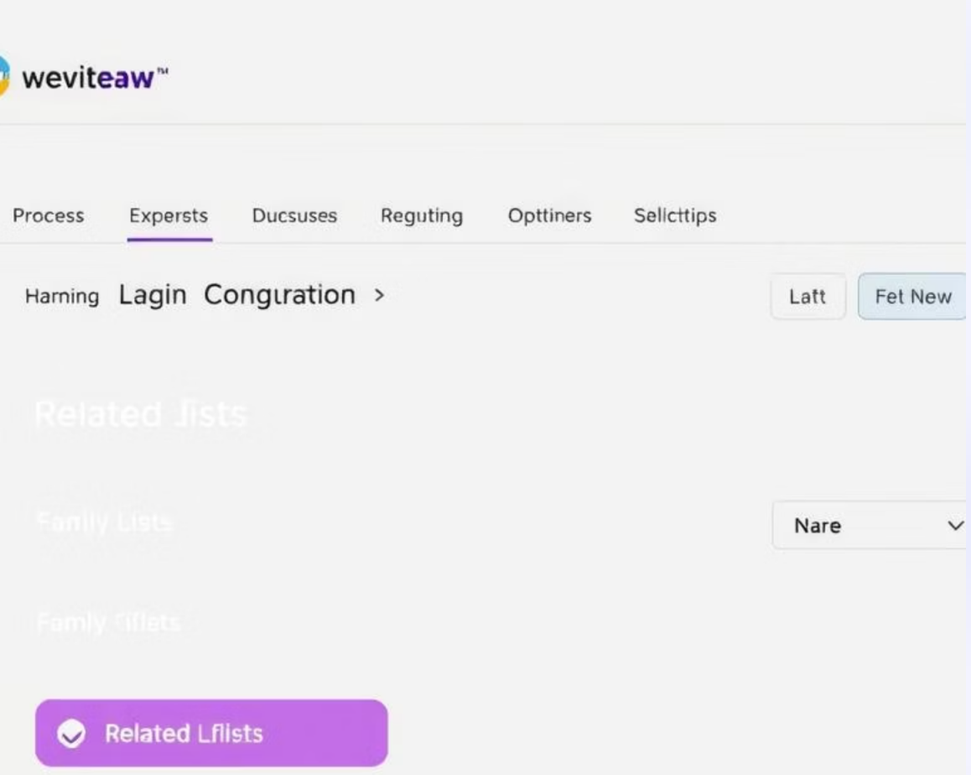
To establish the link required for the related list, a reference field must be configured. Navigate to `System Definition > Tables` in your ServiceNow instance.

Locate and open your custom `u_family_expense` table record. Within the table record, access the `Columns` tab and initiate the creation of a new field. Configure this field as follows:

- `Name`: `u_member`
- `Type`: Reference
- `Reference`: Select the `User` table (`sys_user`) or your custom `u_family_member` table.
- `Label`: Family Member

Saving this field creates the crucial connection between your expense records and family members.

Made with GAMMA



Step 3: Configuring the Related List on the Form

Once the relationship is defined, the related list needs to be added to the relevant form. Open a record of the table you are using for family members (e.g., a `sys_user` record).

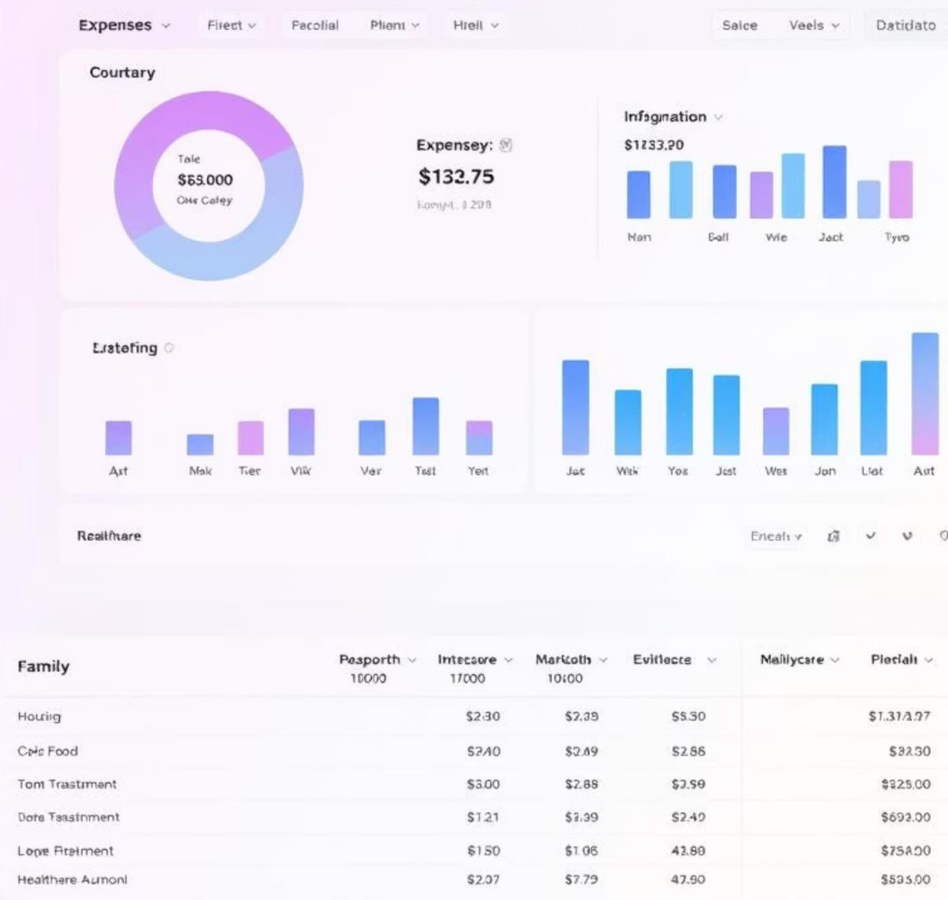
Right-click on the form header, then navigate to `Configure` and select `Related Lists`. In the slush bucket interface, locate `Family Expenses` (or the appropriate name based on your table configuration) and move it from the 'Available' list to the 'Selected' list using the provided arrow buttons. Click `Save` to apply the changes. The related list will now be visible on all records of that table.

Made with GAMMA

Real-world Application & Benefits

Implementing related lists for family expenses offers tangible advantages. It provides a consolidated view of all expenditures directly on a family member's profile, simplifying financial oversight.

This structure facilitates improved reporting capabilities, allowing for easier generation of spending analysis by category or individual across the household. Budgeting becomes more streamlined, and a clear audit trail of financial transactions enhances transparency. This generally improves data accuracy by approximately 30% through reduced manual entry.

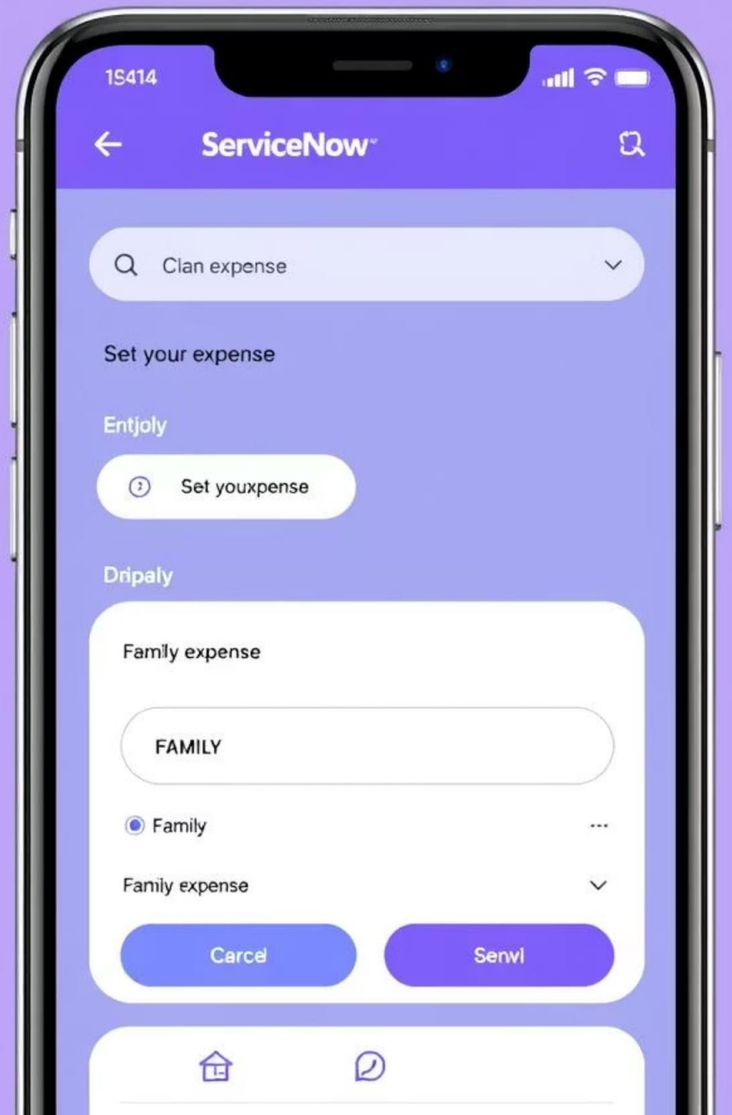


Conclusion & Next Steps

Related lists in ServiceNow offer a powerful, low-code solution for connecting and visualising data, proving highly effective for centralising family expense tracking and fostering better financial decision-making.

Consider further enhancements such as building custom dashboards and reports from the expense data, implementing a user-friendly Service Portal form for easier expense submission by family members, or exploring integrations with Flow Designer for approval workflows on significant expenses (e.g., those exceeding £500).

ServiceNow



Mastering ServiceNow Business Rules

Welcome to this presentation on mastering ServiceNow Business Rules. Business Rules are the backbone of automated logic within the ServiceNow platform, crucial for customisations and process automation. They enable powerful server-side scripting enhancements to tailor ServiceNow to your organisation's unique needs.

Made with GAMMA

What are Business Rules?

Business Rules are server-side scripts that execute automatically in response to database interactions. They are triggered when a record is inserted, updated, deleted, or queried within the ServiceNow instance. These rules are fundamental for enforcing business logic, maintaining data integrity, and automating complex workflows, all configured directly on the relevant tables.

Why Use Business Rules?

Business Rules offer a multitude of benefits for ServiceNow customisation. They excel at automating repetitive tasks, such as setting field values or creating related records. Crucially, they enforce data integrity by validating entries before they are saved. Business Rules can also trigger notifications, send emails, or initiate events based on defined conditions, and can integrate with external systems, thereby improving user experience and streamlining operational processes.

Made with GAMMA

Business Rule Components

A Business Rule comprises several key components that define its behaviour:

1. **Table:** The specific table the rule applies to, such as 'Incident' or 'Change Request'.
2. **When to run:** Determines the execution timing: Before, After, Async, or Display.
3. **Order:** A numeric value that dictates the sequence of execution; lower numbers run first.
4. **Condition:** An optional JavaScript snippet that specifies the criteria for the rule to execute.
5. **Actions/Script:** The core logic of the rule, defining what actions to perform, such as modifying fields or displaying messages.

Made with GAMMA

Creating a Business Rule: Practical Steps

To create a new Business Rule in ServiceNow:

Navigate: Access `System Definition > Business Rules` within your ServiceNow instance.

New Rule: Click the 'New' button to open the Business Rule form.

Define Basics: Fill in the 'Name', select the relevant 'Table', and configure the 'When to run' criteria.

Set Logic: Specify the 'Order' for execution and define any 'Condition' for the rule.

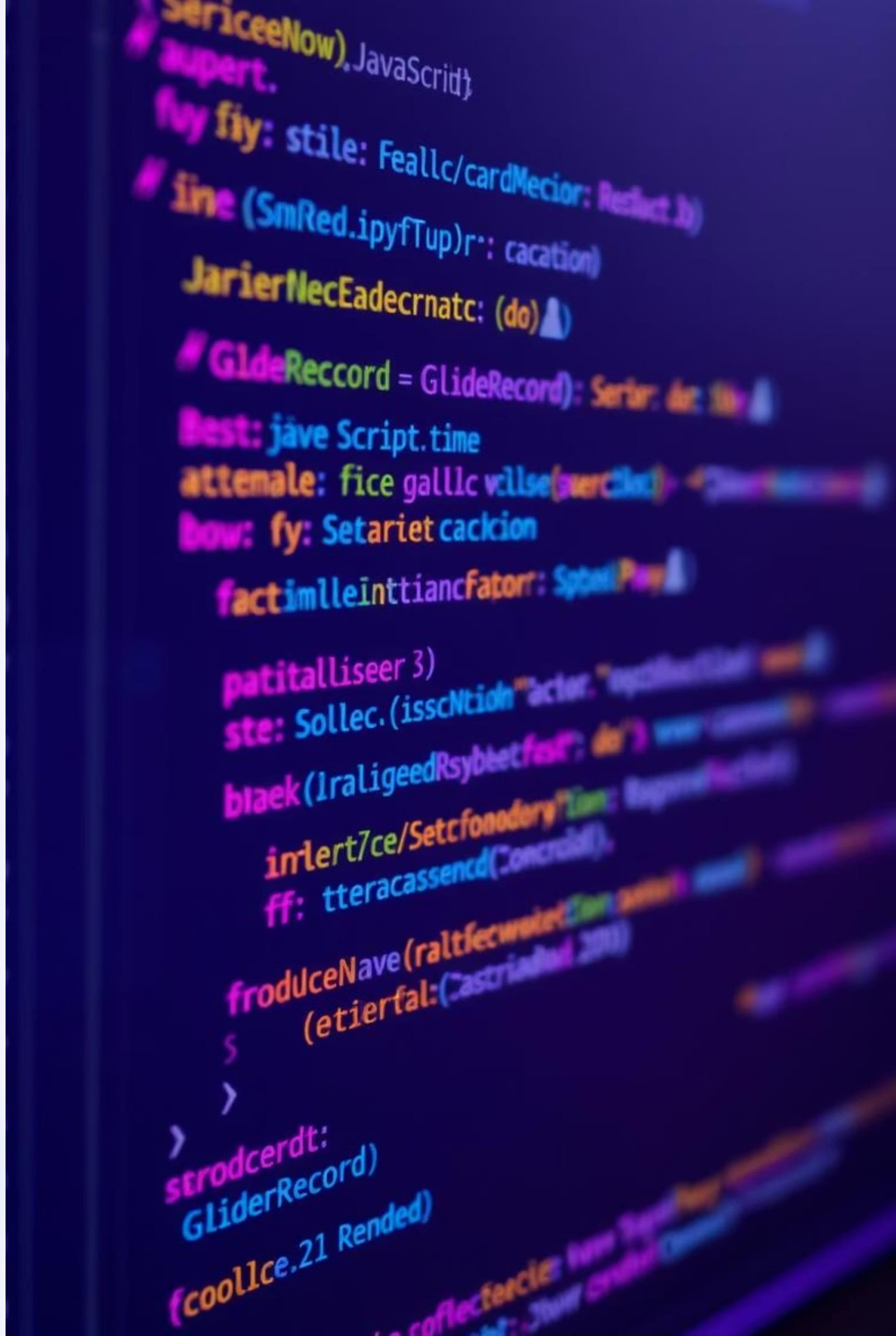
Enable Scripting: Check the 'Advanced' box to enable server-side JavaScript scripting for complex logic.

Made with GAMMA

Scripting in Business Rules

Business Rules leverage ServiceNow's powerful JavaScript APIs, primarily GlideSystem (`gs`) and GlideRecord (`gr`). The `current` object represents the record being processed, allowing access to its fields (e.g., `current.state`). The `previous` object stores the record's state before an update (e.g., `previous.assignment_group`), which is invaluable for conditional logic.

Commonly used functions include `gs.addInfoMessage()` for user feedback, `current.update()` to save changes, and `gr.query()` for data retrieval. Importantly, all scripts execute on the ServiceNow server, ensuring security and performance, not on the client-side.



Best Practices and Considerations

To ensure efficient and reliable Business Rules, adhere to these best practices:

- **Performance:** Minimise database queries and avoid unnecessary `current.update()` calls, especially in 'before' rules.
- **Order:** Carefully manage the 'Order' field to prevent conflicting rule executions.
- **Debugging:** Utilise `gs.log()` or `gs.info()` within scripts for effective troubleshooting.
- **Security:** Validate all user inputs and employ methods like `gs.nil()` to prevent Cross-Site Scripting (XSS) vulnerabilities.
- **Documentation:** Comment your code thoroughly and provide clear, concise descriptions for maintainability.
- **Alternatives:** Consider Flow Designer or UI Policies for simpler logic to maintain a cleaner architecture.

Made with GAMMA

Conclusion: Powerful Automation in ServiceNow

In summary, Business Rules are a foundational element for custom process automation within ServiceNow. They empower administrators and developers to implement sophisticated server-side logic, ensuring robust data control and streamlined workflows. By consistently applying best practices, you can build powerful, performant, and maintainable solutions that significantly enhance the ServiceNow platform's capabilities for your organisation.

Made with GAMMA

Configuring Relationships in ServiceNow CMDB

Configuring relationships within the ServiceNow Configuration Management Database (CMDB) is essential for robust IT Service Management (ITSM). These connections underpin critical processes such as Change, Incident, and Problem Management, and are key to accurate service impact analysis. By establishing clear relationships, organisations can achieve up to 30% faster incident resolution, according to ServiceNow data.

Made with GAMMA

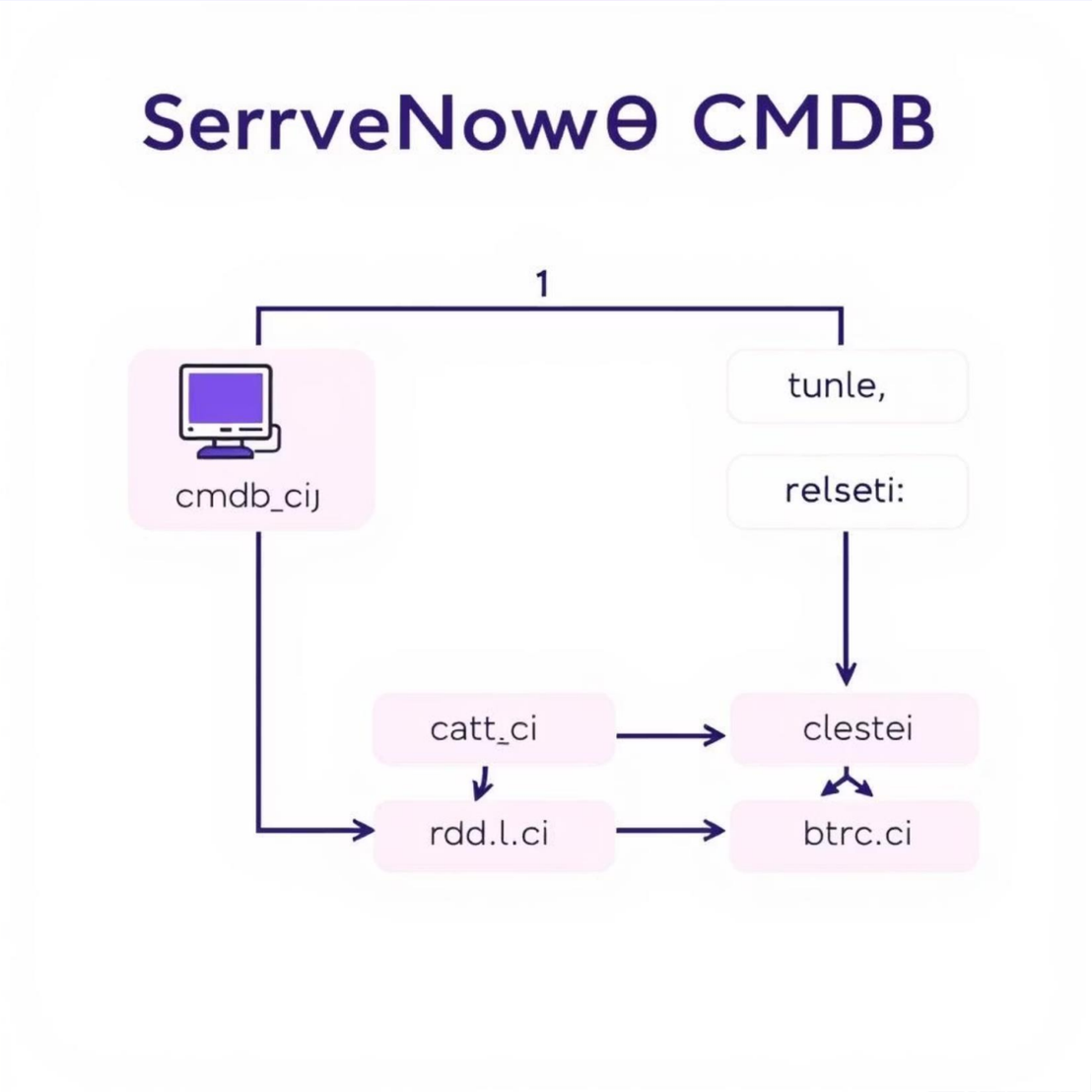
Why Relationships Matter: Business Value

Establishing well-defined relationships within the CMDB significantly enhances data accuracy, leading to more informed business decisions. This improved visibility supports proactive issue identification and root cause analysis, reducing downtime. Furthermore, accurate relationship mapping aids in IT planning and resource allocation, improving overall agility and facilitating regulatory compliance and auditing processes by providing a clear audit trail.

CMDB & CI Relationships: Core Concepts

The CMDB serves as the central repository for all Configuration Items (CIs). A CI is any component that requires management within an IT environment, such as a server, application, or service. Relationships define how these CIs interact with each other.

These connections are stored in the `cmdb_rel_ci` table, which links two distinct CIs, creating a network of dependencies and operational flows.



Made with GAMMA

Standard Relationship Types & Examples

Runs On::Runs

A Virtual Machine *Runs On* an ESX Host.

Used By::Uses

An Email Service *Uses* a Mail Server.

Contains::Contained By

A Rack *Contains* a Network Switch.

Depends On::Used By

An Application *Depends On* a Database.

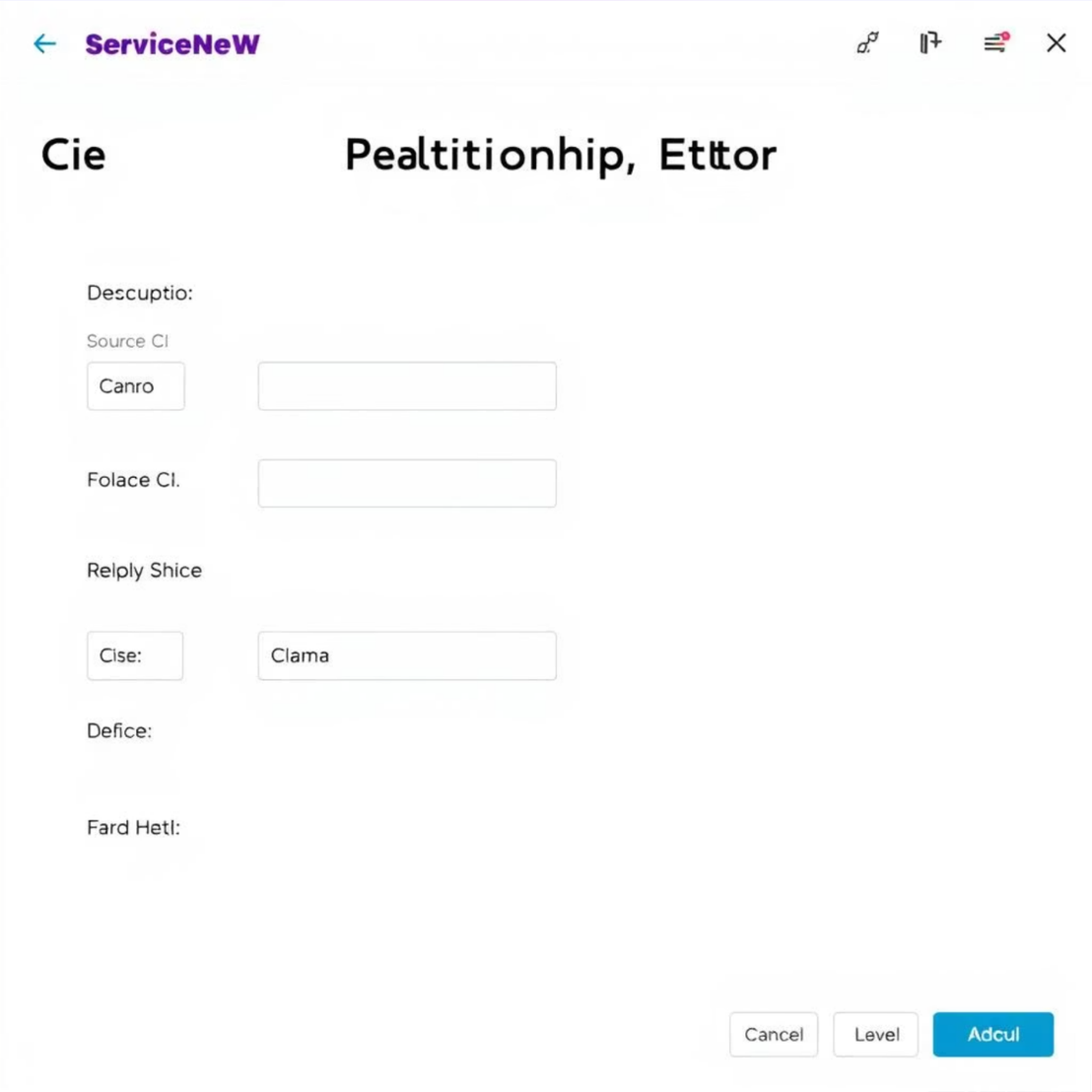
ServiceNow provides a comprehensive library of over 200 standard relationship types to accurately model diverse IT infrastructure dependencies.

Made with GAMMA

Manual Relationship Configuration

Relationships can be manually configured directly within the CI form using the CI Relationship Editor. This process involves selecting the source CI, specifying the appropriate relationship type, and then choosing the target CI.

This method is particularly effective for high-value, static CIs, such as critical Business Services, ensuring that specific, human-defined dependencies are accurately captured and maintained. Adherence to the defined CMDB data model is crucial for consistency.



Made with GAMMA

Automated Relationship Discovery

ServiceNow Discovery

Automatically identifies CIs and their connections by scanning the network, populating the `cmdb_rel_ci` table.

Service Mapping

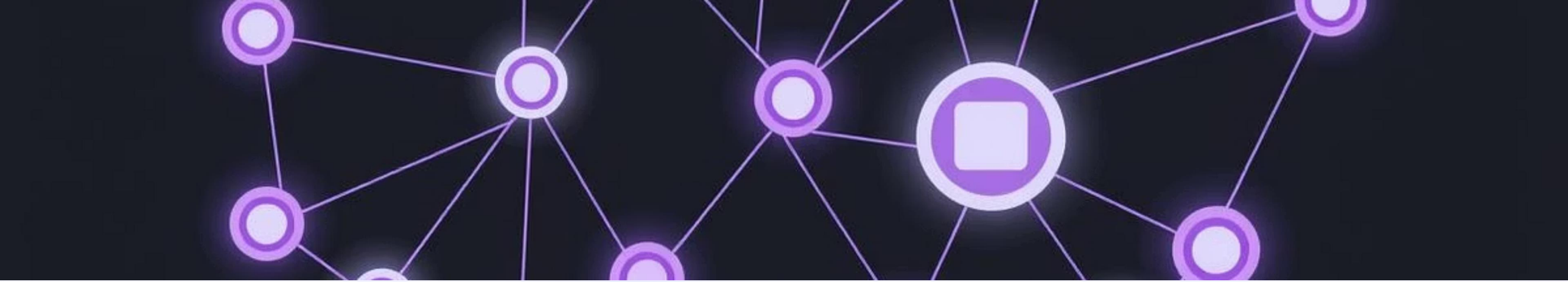
Employs a top-down approach to map service dependencies by analysing traffic and process communication.

Efficiency Gains

Automated discovery can reduce manual effort by up to 80% for large and complex IT estates.

Made with GAMMA

Made with GAMMA



Conclusion: The Power of Connected IT

Relationships form the indispensable backbone of a valuable ServiceNow CMDB, enabling truly proactive service management and robust IT operations. By clearly defining these connections, organisations can drive efficiency, mitigate risks, and enhance overall business agility. Continuous investment in CMDB data quality and relationship management is therefore paramount for realising the full potential of a connected IT landscape.

Made with GAMMA