

A comparison of four complex ServiceNow architectural models

The information in this Success Insight describes different ways to implement ServiceNow®. We don't intend for you to use this document as a decision-making tool—instead, it's critical that you consult with your ServiceNow implementation partner before you make a decision.

What's in this Success Insight

Choosing an architectural model is critical to creating a foundation to build your long-term Now Platform® success. This Success Insight will provide information that will inform your architectural model decision if your organization has complex requirements.

This Success Insight answers these key questions:

1. What is an architectural model?
2. Why is it critical to choose an architectural model before starting development?
3. What are some key factors in my architecture choice?
4. What are some architectural models you can consider?

1. What is a ServiceNow architectural model?

ServiceNow is a flexible platform that you can design to meet specific business requirements using configuration and/or customization. You create an architectural model based on which combination of ServiceNow capabilities and features meet your organizational requirements.

Some organizations meet their requirements using one instance with basic configurations. Other very large or global organizations may consider more complex architectural models because they have intricate needs due to compliance regulations, business models, data sovereignty, data segregation, and/or user segregation.

The most commonly used architectural models used to meet complex requirements are:

- **Deliberate customization** – This is a very flexible model that uses baseline capabilities to extend out-of-the-box (OOTB) applications for complex business requirements.
- **Application extension** – This model extends baseline capabilities to provide a new, scoped, application based on an OOTB application. This lets you maintain the app's original purpose with development and configuration for each independent user group.
- **Domain separation** – This model takes advantage of OOTB functionality on a single ServiceNow instance to enable data and process segregation appropriate for separate business lines and/or customers. A shared database is leveraged across these segments.
- **Multiple instances** – This model involves multiple production instances of ServiceNow that are each purpose-built for a specific set of business needs.

2. Why is it critical to choose an architectural model before starting development?

Your architectural model will be the foundation for your success with ServiceNow. If you start developing with a model that isn't the right fit for your requirements, the time and rework to align to a more appropriate model can be costly. To make matters worse, you can't easily reversed some architectural models after you've started development. For example, if you activate domain separation but later decide it wasn't the right choice, the best way to undo it is to start over in a new ServiceNow instance. On the other hand, if if you don't choose domain separation initially, it's very difficult to adopt later.

3. What are some key factors in my architecture choice?

Your architecture decision should be informed by your application and platform requirements. Consider how these five factors relate to your organization:

1. **Data** – Think about the capabilities you need to segregate users and/or data to meet your business needs and regulations, such as data sovereignty. Also consider the integrations and data sharing you need to meet your requirements. These could include transactional data and master data management sharing with other systems, interactional workflows to enable business process automation, and the volume and frequency of integration processing, which could impact performance.
2. **Process** – Determine your business needs and the processes that must be in place for your organization to manage workflows and to adhere to regulatory compliance.

3. **Manageability** – Evaluate whether you have internal resources with the right skills and availability to take on the roles of administrator and developer. Are the internal governance processes you have in place strong? Do you have well-defined processes for scoping and managing development on the platform?
4. **Performance and scalability** – Assess how quickly you need to be able to adapt to changes in the organization. Also consider the expectations for system performance.
5. **Technical sustainability** – Weigh the priority of keeping the platform running smoothly and to easy to upgrade against your need to have specific, customized features that may make ServiceNow more difficult to manage. This would include the complexity of your implementation as well as the level of technical debt and risk the organization wants to assume.

4. How do different complex models compare?

We have identified four common complex implementation models: deliberate customization, domain separation, multiple instances, and application extension. Let's look at each of these models in more detail.

Deliberate customization

Description	This is a very flexible model that uses baseline capabilities to extend out-of-the-box (OOTB) applications for complex business requirements.
Common customer profile	All organization types can use this model but those with a high degree of process flexibility are most likely to select it.
Example use cases	<ul style="list-style-type: none"> • A customer had a unique process requirement that could not be met with OOTB capabilities. • A large technology company switched from scoped apps to deliberate customization because it needed the ability to share data across teams. • A large bank needed to have six countries use the same process, be able to share records, and segregate some data. They used access control lists to create a customized data model.

Application extension

Description	This is a very flexible model that uses baseline capabilities to extend OOTB applications for complex business requirements. This model can become overly complicated and problematic as complexity increases over time, so only consider it if you have compelling or essential business value that you can't achieve with OOTB applications.
Common customer profile	All organization types can use this model but those with a high degree of process flexibility are most likely to select it.
Example use cases	<ul style="list-style-type: none"> • An organization wanted to provide data and process flexibility to sub-units within the organization and/or delegate development to those sub-units. • A restaurant group required process and data separation between brands but wanted to retain the ability aggregate reporting and provide shared services across brands. • A banking customer wanted to separate data, integrations, and configuration between service groups to fulfill their unique requirements.

Domain separation

Description	This model uses OOTB functionality on a single instance of ServiceNow so the data and process segregation is appropriate for separate business lines and/or customers. Organizations use a shared database across these segments.
Common customer profile	<ul style="list-style-type: none"> • Managed service providers who must keep customers separated • Large organizations with discrete business entities • Highly regulated companies with strict data segregation needs
Example use cases	<ul style="list-style-type: none"> • A customer had a requirement to segregate data or processes between separate legal or operational entities that are under a common management or governance team. • A global business services organization needed to provide broad sets of capabilities across an entire organization, often in a lift-and-shift mode for process. • A holding company that performs a lot of mergers, acquisitions, and divestitures needed to quickly provision an environment to new companies without changing processes. • A managed service provider had a business model that required them to support multiple legal entities.

Multiple instances

Description	This model involves multiple production instances of ServiceNow that are each purpose-built for a specific set of business needs. One of the most common use cases is the need to separate customer data from internal data. Depending on the use case, significant integration between the instances may be necessary to accomplish cross-departmental workflows and visibility.
Common customer profile	<ul style="list-style-type: none"> • Highly siloed or highly advanced customers with the ability to manage multiple independent implementations of ServiceNow • Customers with specific, dictated business requirements, such as separate business units
Example use cases	<ul style="list-style-type: none"> • A large wholesaler had an international team with its own instance because the way they operated was too different from the US and they didn't want to share the cost of maintaining the primary instance in the US. • An enterprise web-based retailer has each business unit operate as a separate company. Each company wanted to have complete control of how its instance was configured. • A large clothing retailer had two instances—one group had its own and the rest of the company ran on a main instance. The company wanted separate cost management and maintenance since each group was its own business unit.

Key takeaway

Your architectural model will be key to your long-term success with ServiceNow. While some organizations will meet their requirements with a basic architectural model, others with more elaborate requirements may need a more complex model. Understanding the requirements for the products you'll implement—and as your platform requirements—will help inform your selection of an architectural model that will scale with your organization.

Note: Consult your ServiceNow implementation team for additional guidance before you make an architectural model decision.

Additional resources

- [Success Checklist – State your transformation vision and outcomes](#)

- [The essentials for implementation success with ServiceNow](#)
- [Organizational change management \(OCM\) planning](#)
- [Success Quick Answer – How can I enable multitenancy using domain separation?](#)
- [Success Checklist – Implement integrations with ServiceNow](#)
- [Video – Implementing integrations with ServiceNow](#)
- [FAQs – Implementing integrations with ServiceNow](#)

Additional resources are also available on our [Customer Success Center](#).

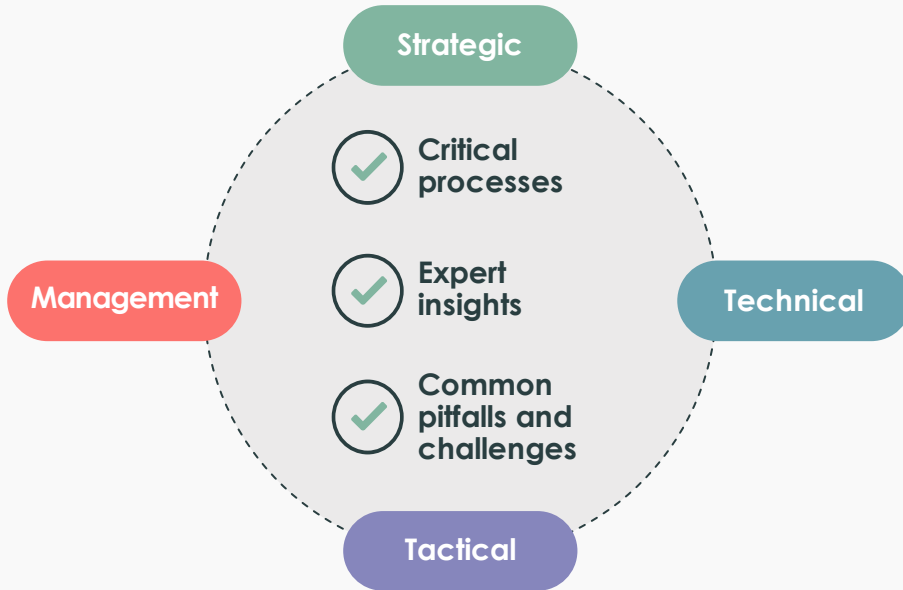
If you have any questions on this topic or you would like to be a contributor to future ServiceNow best practice content, please contact us at best.practices@servicenow.com.

Customer Success Best Practices




ServiceNow's Best Practice Center of Excellence provides prescriptive, actionable advice to help you maximize the value of your ServiceNow investment.



Definitive guidance on a breadth of topics



Designed for:

-  Executive sponsors
-  Platform owners and teams
-  Service and process owners

Created and vetted by experts



Best practice insights from customers, partners, and ServiceNow teams



Based on thousands of successful implementations across the globe

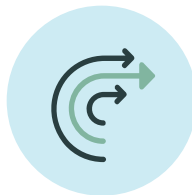


Distilled through a rigorous process to enhance your success

Proven to help you transform with confidence



Practical



Actionable



Value-added



Expert-validated

Get started today.

Visit [Customer Success Center](#).

Contact your ServiceNow team for personalized assistance.