

NAVCI!TE

**How to master the
integration challenges
for modern hybrid IT
environments**

1. MODERN BUSINESSES NEED FLEXIBILITY AND AGILITY TO DRIVE OPERATIONAL EFFICIENCY	3
2. THE CURRENT SITUATION FOR MANY MODERN ORGANIZATIONS	6
3. UNDERSTANDING API-LED CONNECTIVITY	11
4. NAIS AN ENABLER OF API-LED CONNECTIVY	17
5. THE BENEFITS AND HOW BUSINESS NEEDS WILL BE FULFILLED	22
6. THE NAIS POC IMPLEMENTATION STEPS	23

1. Modern businesses need flexibility and agility to drive operational efficiency

Today's businesses face unprecedented disruption and change. The digitization of every aspect of our lives, economies, and societies continues to rise. To thrive in this dynamic ecosystem, an organization needs true business flexibility, agility, and innovation at scale. This calls for a new operating paradigm to drive digital evolution.

Another trend is that today's customers and employees expect real-time, personalized, and connected user experiences on any platform. As the number and complexity of enterprise applications grow and evolve to address these needs, application integration becomes increasingly important. Building point-to-point integrations manually is **time-consuming, inefficient, and costly**.

A growing number of companies recognize the value of using data to make better decisions, optimize processes, and drive innovation. Organizations need a better way to consume and share data and a more flexible and agile way to add new features and solutions.

Many of the challenges that companies face that are not making it possible to achieve what they want are:

- It is not possible to create the different processes required because it is connected in a complex system between different applications inside and outside the company in its value chain
- Once the processes are established, it is not possible to maintain and change them efficiently
- No one has an overview of how problems should be solved. Different parts of the organization see different needs and solve problems in isolated ways.
- Acquiring companies and integrating them with your business processes and applications takes time and effort when you want to leverage the synergies of your acquisition.

This is where an efficient and modern integration strategy based on API-led connectivity comes in.

API-led connectivity, also known as API-led integration or API connectivity, addresses these requirements by helping organizations break down data silos, improve collaboration, respond quickly to changes, and increase innovation. This modern methodology leverages reusable APIs, enabling more efficient and scalable integration.

“IT integration may be the most important success factor for modern businesses that have ended up in a best-of-breed applications and hybrid solutions environment. If implemented correctly, it allows them to drive business development, mergers and acquisitions, and become truly flexible and agile businesses that can innovate at scale.”

– Olof Wistrand, Senior Integration Architect at Navcite

2. The current situation for many modern organizations

For many businesses, the responsibility for IT decisions has been distributed across different business units. Instead of the IT department centrally deciding which systems or applications should be used, various departments have been given the authority to choose and implement their own solutions. This often happens as companies recognize the need to quickly adapt to changing market conditions, customer expectations, and internal goals. The ability to invest in different types of cloud-based application solutions and mergers and acquisitions has created a situation where companies have ended up with many best-of-breed applications in a hybrid environment that are not optimally integrated.

Organizations also have best-of-breed applications and hybrid solutions because they are often better for modern businesses than focusing more on implementing an all-in-one solution. Based on our experience, a number of factors drive this development.

Be able to keep previous investments

Many organizations have invested in existing IT solutions that they do not need to change, that are well-working, and have a long lifetime. They need to function with modern, often cloud-based solutions. They want to replace these old investments step by step when needed.

Specialization and Customization

Best-of-breed applications and hybrid solutions combine the best-specialized tools for specific tasks, optimizing each part of the system for its purpose. This results in higher performance and better solutions for specific needs.

Flexibility

With best-of-breed applications and hybrid solutions, businesses can select the components that best suit their unique requirements. This means they can customize and upgrade individual parts without replacing the entire system.

Cost-effectiveness

By using specialized solutions for critical business parts and a basic all-in-one solution for simpler functions, companies can optimize their investments and get more value for their money.

Seamless user experience

Best-of-breed applications and hybrid solutions must communicate and share information, creating a seamless user experience. Many specialized vendors now offer APIs or bespoke integration solutions that allow different tools to be easily connected with a proper integration strategy.

Faster Innovation

With access to the latest specialized solutions, businesses can more quickly take advantage of new technologies and innovations in different areas, keeping them competitive and adaptable in a fast-changing market.

Risk diversification

By not depending on a single supplier, companies reduce the risk of the entire system becoming obsolete or being locked into a specific ecosystem with limited development opportunities.

In summary, best-of-breed applications and hybrid solutions offer a combination of specialization, flexibility, cost-effectiveness, interoperability, faster innovation, and risk diversification, often making them a better choice than traditional all-in-one solutions.



What is critical for a modern integration platform

There is a need for efficient integration to enable the best-of-breed hybrid solution to function efficiently. The integration platform needs to be a layered solution with the following characteristics.

Fast implementation

A layered solution should allow for fast implementation of some business-critical integrations without much specification work. The integrations can then be enhanced and iterated over time to align with new functionality and requirements.

Modularity and scalability

A layered integration platform should be built modularly, making adding, changing, or upgrading individual components easier without affecting the whole system. This makes the system scalable and flexible for future needs.

Maintenance and updates

A layered integration platform facilitates system maintenance and updates. When one part of the platform needs to be updated or fixed, this can be done without disrupting other parts.

Interoperability

Layering the integration platform makes it easier for different systems and applications to communicate with each other. This promotes a seamless data exchange and improves the system's efficiency and usability.

Security

By separating different functions and services into layers, security measures can be implemented specifically for each layer. This reduces the risk of security breaches and makes it easier to address security issues if they occur.

Performance optimization

A layered integration platform should allow the performance of each layer to be optimized. Each component can be optimized for its specific function, improving overall system performance.

Cost-effectiveness

A layered integration platform should reduce development and maintenance costs by enabling the reuse of existing components and tools and reducing the need for extensive customization.

Simplified fault identification

With clearly delineated layers, tracking and identifying fault sources becomes easier. This simplifies the troubleshooting process and minimizes system downtime in case of problems.

Agile development

A layered integration layer supports agile development by allowing parallel work on different layers. This means teams can work more efficiently and deliver features and improvements faster.

In summary, by implementing a layered integration platform, organizations can ensure that their systems are flexible, scalable, secure, and cost-effective, which is critical to meeting today's rapidly changing business and technical requirements.

3. Understanding API-led Connectivity

API-led connectivity has been developed as a new layered integration methodology to address the integration needs of modern businesses with an IT landscape characterized by best-of-breed and hybrid technologies.

What is an API?

Before we move into API-led connectivity, let's start by defining what an API is. An Application Programming Interface (API) is a set of defined rules that enables different applications to communicate with each other. APIs function as intermediaries that allow different software components to interact, facilitating data exchange and functional interoperability. This approach to connectivity helps organizations open their application's data and functionality to external developers, business partners, and internal departments, promoting enhanced collaboration and innovation.

With APIs, businesses can:

- Enhance collaboration: Enable data and service sharing across departments and external partners.
- Promote innovation: Facilitate the rapid development and deployment of new features and solutions.
- Achieve scalability: Efficiently scale operations by reusing existing APIs across multiple applications.

The traditional solution for setting up an integration is based on point-to-point integration P2P. Some of the challenges with P2P solutions for integration are:

- Increased complexity and maintenance
- Lack of centralized governance
- Limited scalability and adaptability
- Higher dependency on technical expertise
- Potential security risks

What is API-led Connectivity?

API-led connectivity is a strategic methodology that integrates applications and data through reusable APIs in real time. Unlike traditional P2P integration methods described above, API-led connectivity organizes integration into three distinct layers as per described below. You will find an illustration for visualization on the next page.

EXPERIENCE LAYER

The top layer exposes integrated and aggregated data in a format consumable by end-users, whether internal developers, partners, or external developers.

- Represents the external-facing interface that applications or users interact with.
- Focuses on delivering a consistent and user-friendly experience.
- Abstracts the complexity of underlying systems and processes, presenting a unified and simplified interface.

PROCESS LAYER

- Builds on top of System APIs and defines the logic for how different systems interact to accomplish a specific business process or workflow.
- Orchestrates and coordinates data and services from various sources, often spanning multiple System APIs.
- Implements business logic and workflow orchestration.
- Provides a higher-level abstraction by defining the sequence of steps needed to fulfill a particular business task.

SYSTEM LAYER

- The foundational layer that provides connectivity to systems of record, such as databases, ERPs, or legacy systems.
- Exposes data and capabilities from underlying systems.
- Represents a standardized and consumable interface to access and manipulate data in systems of record.
- Offers a level of abstraction, shielding the complexities of the underlying systems from higher layers.

Each layer has common characteristics like common security configurations, ownership and overall frequency of changes.

API-led connectivity visualized



What are the benefits for businesses of implementing API-led Connectivity?

Implementing API-led connectivity offers numerous advantages for businesses seeking to enhance their integration capabilities. Below are some of the key benefits that organizations can realize through this approach.

REDUCED INTEGRATION COMPLEXITY AND COST

API-led connectivity allows organizations to create an API once and reuse it in various contexts to serve multiple business needs across the enterprise. This reusability significantly reduces the complexity involved in integration efforts, leading to the following:

- **Cost Savings:** Lower development and maintenance costs thanks to reusable APIs.
- **Increased Productivity:** Faster development cycles as developers can leverage existing APIs rather than building new integrations from scratch.

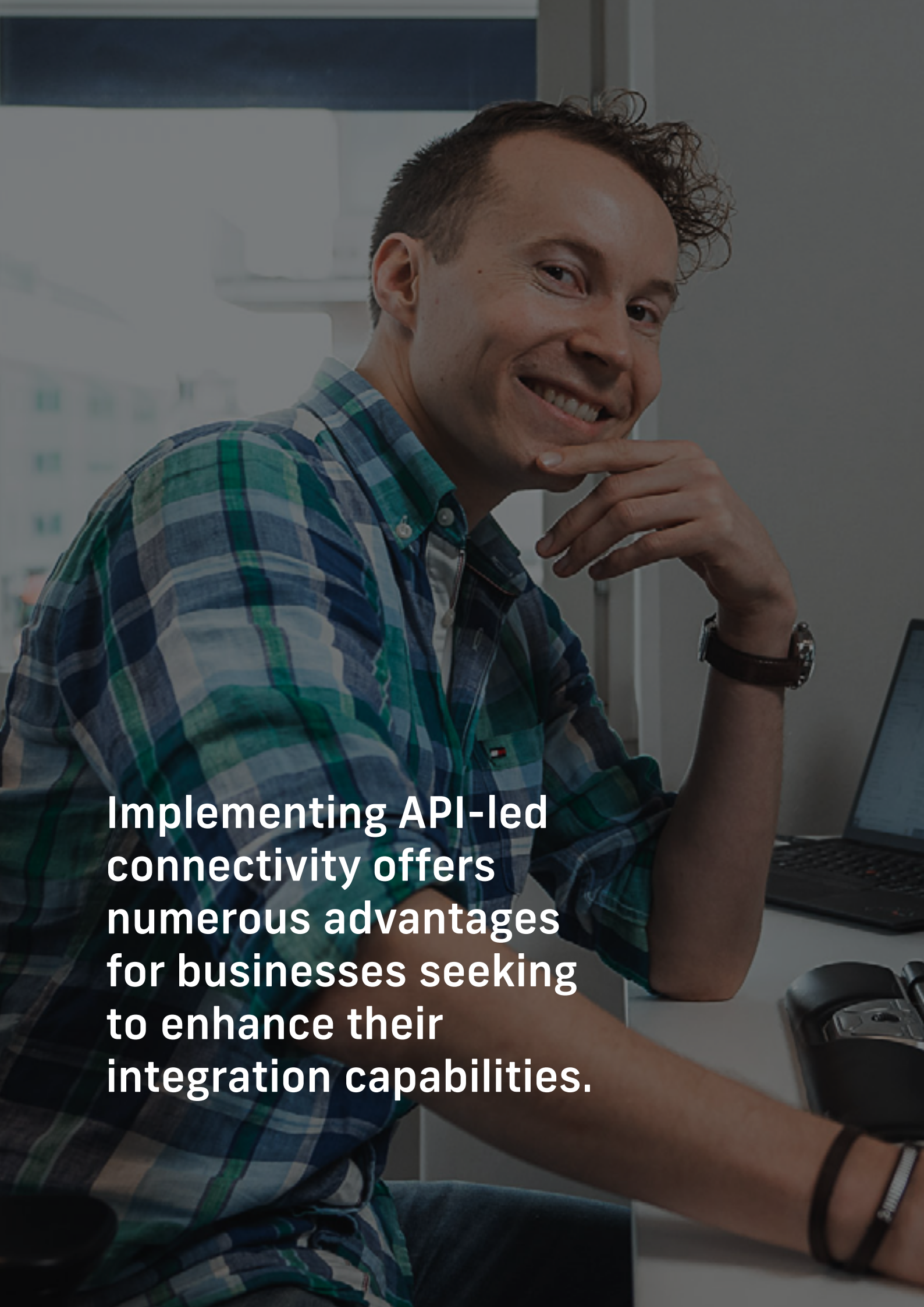
IMPROVED ACCURACY

By standardizing integration through API-led Connectivity, organizations can achieve greater accuracy in their systems:

- **Fewer Errors:** Reducing the number of different APIs minimizes the risk of errors.
- **Consistency:** Ensures consistent data and functionality across various applications and platforms.

PROVIDE A CONSISTENT CUSTOMER EXPERIENCE

Using the same APIs to power back-end source systems and front-end customer interfaces helps provide a unified and consistent customer experience. This consistency can enhance customer satisfaction and loyalty by ensuring that users receive the same level of service regardless of the platform they are using.

A man with short, curly hair, wearing a blue and green plaid button-down shirt, is sitting at a desk. He is smiling and looking towards the camera, with his right hand resting on his chin. He is wearing a black watch on his left wrist and a black wristband on his right wrist. A laptop and a keyboard are visible on the desk in front of him. The background is a blurred office setting.

Implementing API-led connectivity offers numerous advantages for businesses seeking to enhance their integration capabilities.

BETTER VISIBILITY AND CONTROL

Integrating applications via API-led Connectivity allows for better management and security:

- **Enhanced monitoring:** APIs can be monitored and managed centrally, providing better oversight of their usage and performance.
- **Improved security:** Standardizing security protocols across APIs ensures the protection of sensitive data.

FASTER ADAPTATION TO CHANGE

One of the significant advantages of API-led Connectivity is its ability to help organizations quickly adapt to changing business requirements and technological advancements:

- **Scalable solutions:** APIs can be easily adjusted or replaced without affecting the overall system.
- **Rapid innovation:** Businesses can introduce new features and services faster as changes can be implemented at scale with reusable APIs.

ACCELERATED APPLICATION DEVELOPMENT

API-led Connectivity encourages a composable architecture, where different components can be assembled and reassembled as needed:

- **Efficiency:** Developers can build on existing APIs to create new integrations and applications, boosting development speed.
- **Scalability:** One API built by a single developer can power hundreds of integrations and applications within the organization.

4. NAIS an enabler of API-led Connectivity

NAIS (Navcite Azure Integration Services) is a cutting-edge integration platform hosted on Azure that is designed to streamline and accelerate the creation of new integrations. Leveraging Azure functions, configurations, and building blocks, NAIS offers an intuitive and fast solution for your integration needs in today's best-of-breed and hybrid environments.

NAIS operates on the core principles of API-led Connectivity described in the chapter above. It is a low-code solution in which integrations are built by configuration.

NAIS includes a library of pre-configured building blocks configured and linked to form an integration flow. The integration configuration is the recipe the runtime NAIS executor interprets and executes.

NAIS is also designed to handle any anomalies that may arise within integration. This means that the NAIS executor can be overlaid on a specific flow so that you can easily implement your own .net code if the integration requires a custom solution.

The transformations are handled by NAIS transformers based on Scriban. Scriban is an open-source project based on `{{}}` syntax.

Scriban offers advantages such as you get WYSIWYG (What You See Is What You Get), which makes it extremely fast to develop/change transformations.

Additionally, Scriban ensures high-speed execution.

The NAIS components

NAIS is based on the following four components.

1. NAIS EXECUTER

The NAIS Executer is the heart of the NAIS platform and is responsible for interpreting and running integration configurations. Acting as the run-time engine, the Executer processes the "recipe" created in the NAIS configuration, connecting systems, orchestrating workflows, and handling data transformations in real time. The NAIS Executer ensures reliability and flexibility, supporting both standard integration flows and custom .NET logic to address specific business requirements. With its robust anomaly-handling capabilities, the Executer ensures seamless operation even in complex or unpredictable integration scenarios.



2. NAIS TRANSFORM

The NAIS Transform module handles all data transformations required within an integration flow. Powered by Scriban, an open-source transformation language, NAIS Transform offers a user-friendly, WYSIWYG (What You See Is What You Get) interface for designing and modifying transformations with speed and precision. This component is optimized for high performance, ensuring that transformations are executed efficiently, even for large or complex data sets. With Scriban's familiar `{{}}` syntax, NAIS Transform enables developers to standardize and simplify data mapping across systems while minimizing development time.



3. NAIS CONFIGURATIONS

The NAIS Configurations component represents the integration "recipe," where all steps, building blocks, and integration flows are defined. Configurations are created through a low-code approach, making it easy to design and implement integrations without extensive coding expertise. Each configuration specifies the systems, data flows, and transformations required for a given integration, which are then interpreted and executed by the NAIS Executer. This modular, reusable approach to integration design ensures scalability, consistency, and faster deployment of new solutions.



4. NAIS BUCKET OF STEPS

The NAIS Bucket of Steps is a library of pre-built integration steps (building blocks) that can be configured and linked together to create custom integration flows. Each step represents a discrete function or action, such as sending or receiving data, invoking APIs, applying transformations, or orchestrating workflows. By leveraging this bucket of steps, users can quickly assemble integrations without writing extensive custom code. The modular nature of these steps ensures flexibility and reusability across different integration projects, accelerating development and reducing complexity.



Full implementation of NAIS



The benefits of the NAIS structure for the IT organization

FASTER DEVELOPMENT WITH LOW-CODE CONFIGURATIONS

- Reduces reliance on traditional development by enabling integrations to be built through configuration rather than custom coding.
- Speeds up delivery timelines for new integrations, empowering IT teams to respond more quickly to business demands.

REUSABLE COMPONENTS AND STANDARDIZATION

- Pre-built building blocks (NAIS Bucket of Steps) and reusable APIs ensure consistency across integrations while minimizing redundant work.
- Promotes a standardized approach to integration, simplifying maintenance and governance.

FLEXIBILITY TO HANDLE COMPLEX SCENARIOS

- Supports custom .NET code overlays for unique use cases, giving IT teams the flexibility to address anomalies or advanced requirements without disrupting the overall system.

IMPROVED MONITORING AND TROUBLESHOOTING

- Centralized monitoring and logging through Azure tools enable IT teams to proactively manage integrations, troubleshoot issues quickly, and ensure high availability.
- Reduces downtime and operational disruptions.

SCALABILITY AND FUTURE-PROOFING

- Modular, API-led architecture allows the IT organization to scale integrations easily as business needs grow or evolve.
- Seamlessly supports hybrid and multi-cloud environments, making it adaptable to future technology changes.

COST EFFICIENCY

- Reduces the total cost of ownership (TCO) by enabling the reuse of APIs and steps, cutting development time, and minimizing technical debt from legacy point-to-point solutions.

5. The benefits and how business needs will be fulfilled

Many of the challenges that companies face that were presented at the beginning of this document that is the result of the business drivers like:

- Master unprecedented disruption and change
- Drive digital transformation to increase competitiveness and operational efficiency.
- Enhanced customer experience
- Be able to innovate at scale
- Drive the business based on data to deliver more value to customers and make faster and better decisions.

We can master more effectively with a well-implemented integration strategy based on API-led Connectivity and NAIS.

You can start with a small step and investment, learning and developing the strategy's implementation over time rather than focusing on making significant investments in implementing standard integration platforms or replacing the current ERP solution.

This gives you a fast ROI and a new approach to driving digital transformation, focusing on short-term and long-term business needs.

6. The NAIS POC implementation steps

A full implementation of NAIS always starts with a POC implementation. A full implementation can be rolled out when the POC has been executed.

Navcite's project model and established ways of working are applied when a complete implementation project starts to ensure quality, efficiency, and structure throughout the integration process. The focus is on delivering quickly and with high quality through simplicity, iterative development, and clearly defined expectations. Close collaboration among all involved parties is essential to achieving the desired results and delivery quality.

How to Create an Integration with NAIS as a Proof of Concept (POC) Project

STEP 1 - DEFINE THE BUSINESS USECASE

Start by identifying the specific integration scenario you want to address. For example, synchronizing customer data between a CRM system (e.g., Salesforce) and an ERP (e.g., Microsoft Dynamics 365 or Infor M3). Clearly define the data flows, systems involved, and the desired business outcome.

STEP 2 - DEFINE THE TIME IT WILL TAKE TO IMPLEMENT THE POC AND THE COST

STEP 3 - SET UP THE AZURE ENVIRONMENT AND NAIS - CUSTOMERS OWN AZURE OR USE NAVCITES AZURE ENVIRONMENT

STEP 4 - ANALYZE INVOLVED APPLICATIONS

STEP 5 - SET UP THE INTEGRATION CONFIGURATION

- Open the NAIS Configurations tool and create a new configuration.
- Specify the source system (e.g., the CRM) and the target system (e.g., the ERP).
- Outline the sequence of steps required to complete the integration, leveraging the NAIS Bucket of Steps for pre-built actions like API calls, data extractions, and message routing.

STEP 6 - DESIGN DATA TRANSFORMATIONS

- Use the NAIS Transform module to map and transform data between the source and target systems.
- With Scriban's intuitive `{{}}` syntax and WYSIWYG interface, define transformation rules quickly and accurately, ensuring data formats and structures align with the requirements of each system.

STEP 7 - HANDLE ANOMALIES (IF REQUIRED)

- If your integration flow requires custom logic or has unique requirements, overlay your specific .NET code using the NAIS Executer's anomaly-handling capabilities.
- This allows you to address edge cases or perform advanced processing within the integration.

STEP 8 - TEST THE INTEGRATION

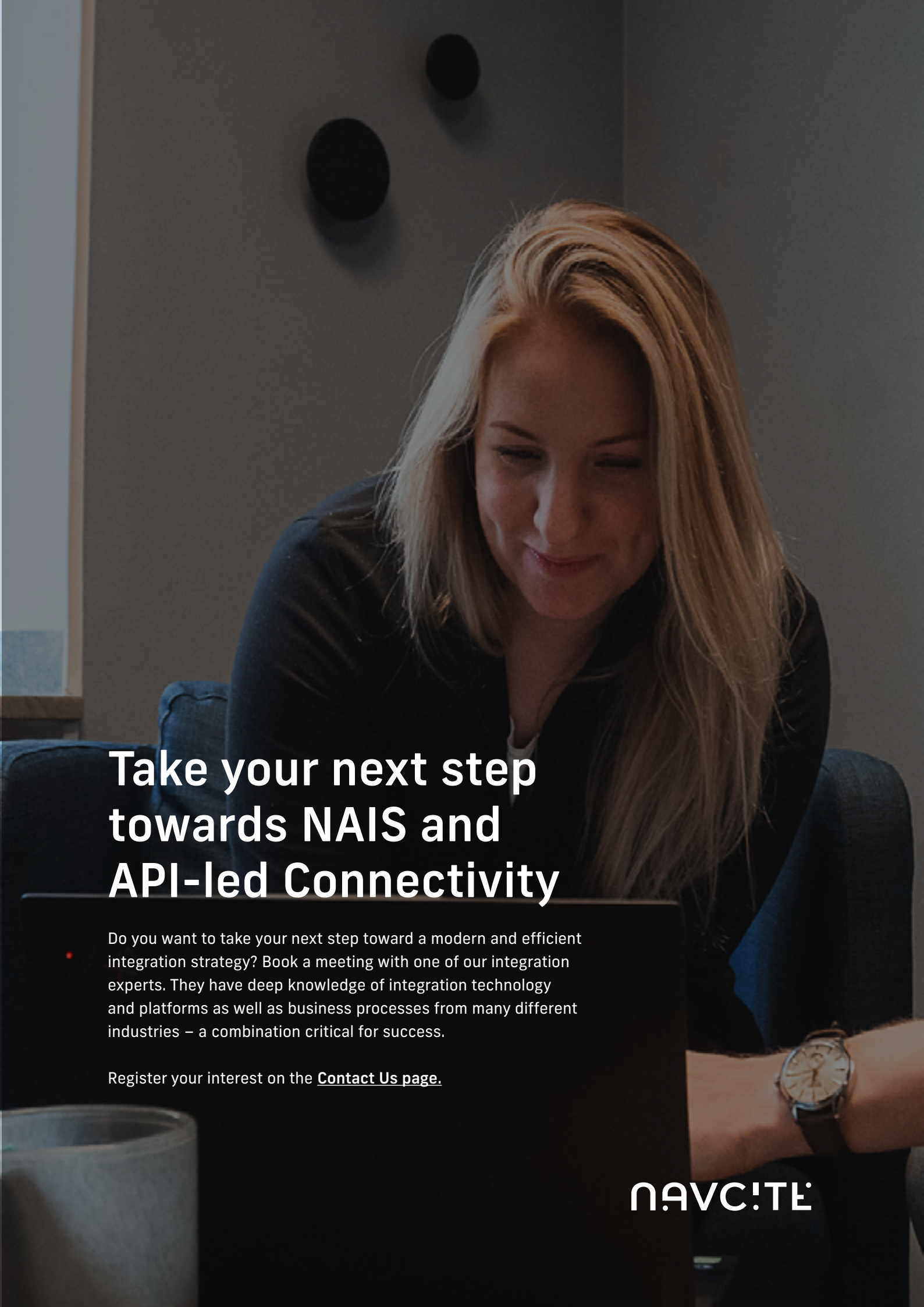
- Run the configuration in a controlled test environment using the NAIS Executer.
- Validate the data flows, transformations, and overall performance to ensure everything works as expected.
- Use Azure's monitoring tools and logs to identify and resolve any issues.

STEP 9 - DEPLOY THE INTEGRATION

- Once validated, deploy the integration to production.
- Monitor the execution through Azure's centralized monitoring tools to ensure smooth operation and capture any unexpected errors.

STEP 10 - ITERATE AND OPTIMIZE

- Gather feedback from stakeholders and measure the integration's performance against the business goals.
- Make adjustments to the configuration or transformations as needed to refine the solution and prepare for scaling to additional use cases.



Take your next step towards NAIS and API-led Connectivity

Do you want to take your next step toward a modern and efficient integration strategy? Book a meeting with one of our integration experts. They have deep knowledge of integration technology and platforms as well as business processes from many different industries – a combination critical for success.

Register your interest on the [Contact Us page](#).

NAVCI:TE