



White Paper

BUILDING A DIGITAL ENTERPRISE ECOSYSTEM



Table of contents

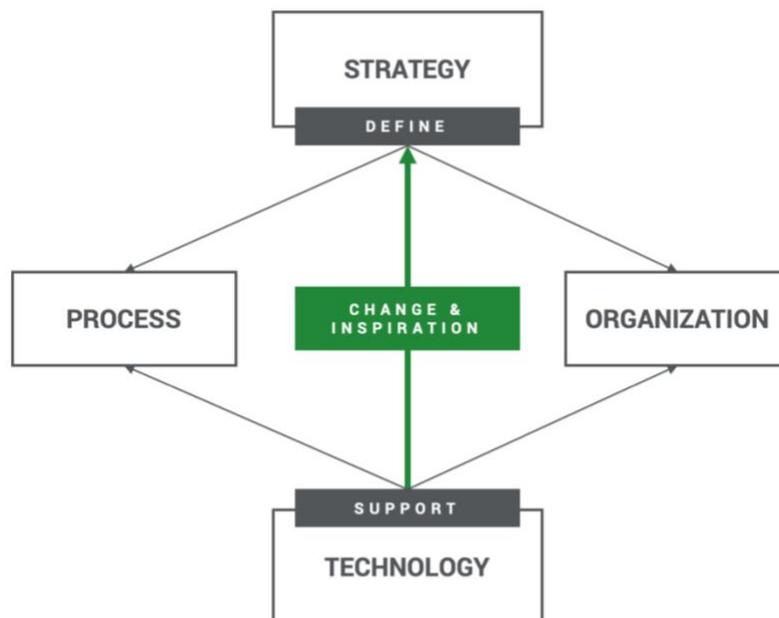
1. Introduction	3
2. Digital ecosystem approach	5
2.1. Product-friendly architecture	6
2.2. Vendor ecosystem	7
2.3. Key business benefits	8
2.4. Transition to digital ecosystem	9
3. Solution architecture	10
3.1. Domain-driven design and API-based architecture	10
3.1.1. Layered API architecture	11
3.1.2. Domain-driven design-based microservice architecture	12
3.1.3. API documentation	12
3.1.4. Benefits of domain-driven design and API-driven architecture	12
3.2. Service Layer Architecture	13
3.2.1. Service layer development principles and core features	14
3.3. Cloud architecture	15
3.3.1. Infrastructure-as-a-Service (IaaS)	15
3.3.2. Platform-as-a-Service (PaaS)	15
3.3.3. Functions-as-a-Service (FaaS)	16
3.3.4. Software-as-a-Service (SaaS)	16
3.4. System integrations	16
3.5. Logging and monitoring	17
3.6. Updates and version upgrades	17
3.7. Reporting and business intelligence	18
4. Case study: Digital ecosystem in healthcare	19
4.1. Background	19
4.2. Solution overview	19
4.3. High-level enterprise architecture	20
4.4. Functional architecture	21
4.5. Solution details	21
5. Nortal is building a seamless society globally	22
5.1. About Nortal	22
5.2. How we work	22



1. Introduction

Digitization is one of the leading trends in society and the economy. It is characterized, above all, by the acceleration of the development and dissemination of information and communication technologies, which are leading to faster and more far-reaching changes in a large number of industries and areas of life.

In order to remain competitive, industries have to adapt their strategies, methods and organization and bring their technological process into shorter cycles. Agility is no longer just a competitive advantage, but rather a basic prerequisite for survival in competitive sectors.



Many companies have difficulties to expand and renew their processes and technologies with the necessary speed. In this context, the existing IT systems are increasingly seen as a burden and obstacles that slow down the transformation of the company and hinder the introduction of new technologies. Companies often fear that the introduction of technological innovations could cause the existing landscape of older systems to collapse. While it is also a frequent case, that neither a big-bang approach to introduce



a completely new system landscape nor a gradual transformation of individual systems over several years is a viable option.

This is often linked to several interrelated challenges:

- Many systems are near the end of their life cycle.
- Systems overlap in functionality or in some cases even have no clearly defined purpose.
- Products have been heavily customized to meet company-specific needs and cannot be upgraded without significant costs and risks.
- System integrations are implemented using a point-to-point approach or are poorly orchestrated.
- Various workarounds are used to bypass the limitations of current systems, leading to even more complexity in processes.

Although there is no miracle cure for this situation, a well-considered use of technology coupled with the right business and IT strategy and appropriate partnerships can become a key factor that rapidly increase the adaptability and competitiveness of companies.

Nortal's approach to developing technological landscapes for modern industrial enterprises is based on a modular, digital ecosystem. The ecosystem enables gradual and continuous improvement while bringing tangible benefits for businesses. This approach is based on the conviction that an open, efficient software architecture is a critical component for success in the modern business world.

Key elements of the solution:

- Open ecosystem, free from manufacturer lock-ins
- Modular architecture to adapt to ever-changing business requirements
- Flexible and close cooperation between the customer and all ecosystem providers

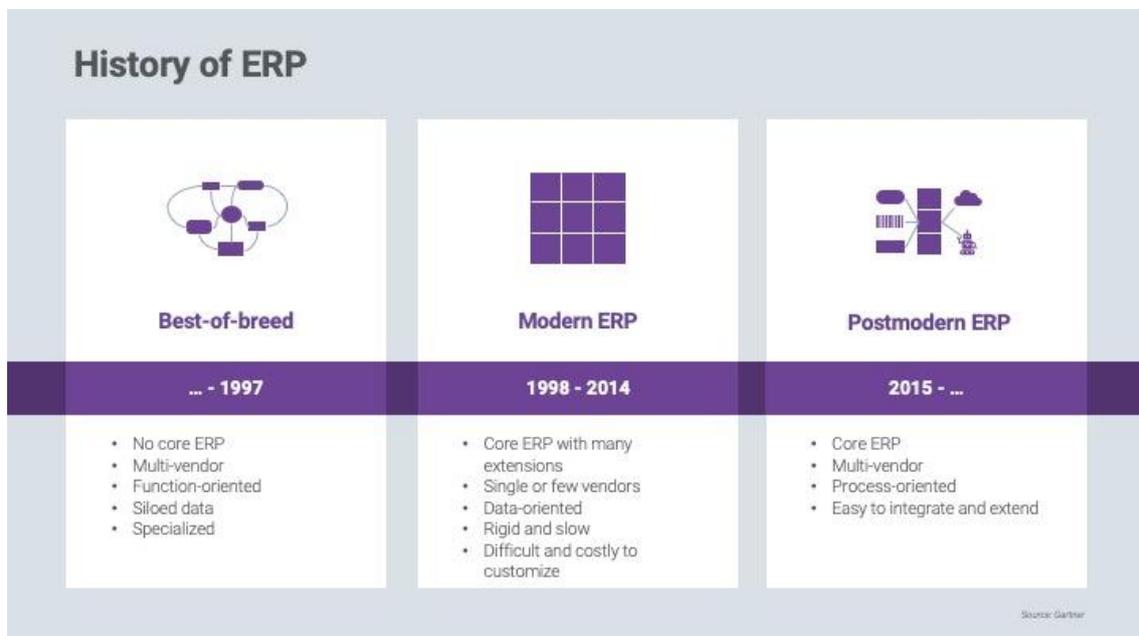


2. Digital ecosystem approach

The shift towards viewing IT as an ecosystem, rather than a simple business function, is based on the growing interdependence between core business and IT. As technology can change the business models of entire industries and create new business opportunities for innovators, IT must become a strategic partner to management and an enabler of business transformation.

At the same time, technological progress has led to an ever-increasing networking of different IT systems and devices (IoT) and to continuously growing volumes of usable data. As a result of these trends, a holistic view of the IT systems in use is necessary in order to be able to exploit the full potential of digitization. In particular, business processes need to be completely digitized across the business functions and data needs to be made available in a way that is beneficial to all business segments.

These changing requirements for cooperation between IT and business units are pushing large monolithic applications, which cover a large number of business functions, to their limits. In the area of ERP systems, according to the market research company Gartner, these developments have already led to a new era of post-modern ERP systems. This approach aims to restrict ERP systems, that have been extended far beyond their core functions over the years, back to their valuable role. Also, to integrate them via specialized standard or individual software for specific business processes.



Therefore, a simple and a secure networking of the most diverse systems and devices and the intelligent use of their data form the core of the ecosystem approach. In addition, since the importance to delimit the functionalities of individual systems is growing, it becomes critical to make appropriate choices regarding the products and their areas of application as well as external IT service providers.



Our ecosystem approach covers all these issues and forms a basis for a sustainable and more easily adaptable corporate IT. In the following, individual aspects of the ecosystem approach are examined in more detail.

2.1. Product-friendly architecture

While the deployment of globally sold standard IT products and multi-tenant cloud services (SaaS) poses its own challenges, also building a complex digital ecosystem without existing market-ready solutions is too slow, expensive and risky. Therefore, our ecosystem approach is based on a service-layer architecture that enables a smooth cooperation between different products and individual solutions.

This open ecosystem approach is relevant when complex IT systems support a significant part of business activities, regardless of the industry or systems in use. For example, it can be applied both to companies in the manufacturing industry, where Manufacturing Execution Systems (MES) are the core production solutions, and in the healthcare sector, where Electronic Medical Record (EMR) systems or Hospital Information Systems (HIS) are at the center of the technological landscape.

In addition to production systems, standard products for supporting business processes such as sales and finance are an important cornerstone of the ecosystem approach.

There are very few industries where sales and marketing are not an integral part of the business. Therefore, a robust CRM is one of the key elements of a thriving ecosystem. However, in many companies, CRM systems are barely integrated with the rest of the landscape. Often, CRM has been purchased retroactively and only used as a stand-alone tool by salespeople. To harness the power of modern CRM, such as Microsoft Dynamics or Salesforce, it must be integrated into the ecosystem as a key element.

ERP systems, despite their bad reputation, carry also fundamental importance for the core business processes. They should be used as a minimum for proper accounting and efficient financial controlling, since it is not recommended to use proprietary solutions for these sensitive and highly regulated processes. A solid ERP product (or a small collection of individual products) is still the best option for financial management, both in the short and long term.

The products and cloud services provide a comprehensive and robust ecosystem foundation, but a digital ecosystem can only function if these products and modules work together seamlessly. Nortal's service layer architecture provides layers between the various products and business units and acts as a dynamic link that allows the ecosystem to thrive.

Service layers provide the necessary tools to avoid complex customization of products and allow adding, updating and removing products and modules without making changes



to other components. The architecture also provides a consistent and well-defined integration for all applications, making it easier to bring new applications into the ecosystem. When translated into business benefits, this approach allows the IT architecture to better adapt to existing and future business needs, it provides significantly improved lifecycle management, and it reduces considerably the risk of lock-in to specific system vendors.

2.2. Vendor ecosystem

Once the core components are in place and the foundation is robust, you can implement a vendor ecosystem to develop the solution further. By providing a common user experience for all application developers (API architecture, authorization logic, error handling, etc.) and novel ways to integrate with the core products at all levels, the innovative power of start-ups and smaller vendors can be leveraged in unpredictable ways.



2.3. Key business benefits

The key to successfully realize the benefits of the new solution lies not only in the selection of specific products or modules, it requires also an understanding of holistic approach when implementing business-critical systems based on valid problem definitions. The ecosystem approach offers various business and technological advantages:

- Combining business continuity with agility
 - › The modular structure of the ecosystem enables greater agility in responding to changes in the business environment through easier integration of new technologies.
 - › It also lowers the threshold for trying out new business ideas.
- Better adaption to existing and future business needs
 - › A more adaptable IT landscape allows also more proactive business management, making it easier to add, update and replace parts of the ecosystem as needed.
 - › Dependence on individual vendors is reduced and it becomes easier to develop certain components independently.
- Improved IT lifecycle management
 - › Update and replace individual IT systems without interrupting other ecosystem applications.
 - › Full control over the IT architecture via standardized and open integration and data extraction interfaces.
 - › Avoid the use of legacy systems after they have reached the end of their life cycle.
- Reduced risk of vendor lock-in
 - › Vendors cannot establish a monopoly position towards the customer.
 - › Cost savings and better service through increased competition between vendors.
- Increased level of automation
 - › Automation of the information transfer reduces manual activities and allows a stronger focus on value-adding work.
 - › Introduction of advanced analysis and AI solutions for automation and decision support (e.g. predictive maintenance).



2.4. Transition to digital ecosystem

As with all large IT projects, the key to a successful transition into an ecosystem approach lies in dividing the project into sufficiently small, manageable components. As the size of the components increases, the amount of work that does not add value to the final solution tends to grow exponentially. At the same time, the threshold for successful project completion also rises.

Successful implementation requires agile development practices that allow the solution to be developed either in short sprints or in continuous development and release cycles. Agile methods make it possible to ensure that the direction, quality, and speed of the transformation throughout the entire project are in line with the specified goals. These practices, combined with DevOps-based continuous integration and deployment (CI/CD), allow small changes to be quickly put into production. This approach reduces risks, training needs, friction and resistance to change within the organization.

The development of an IT ecosystem can be divided into work packages of different sizes. Instead of forcing the same development process for each work package and team, we recommend allowing some variations in the methods. The same processes may not be suitable for product configuration-oriented and customer-specific software development projects.

3. Solution architecture

3.1. Domain-driven design and API-based architecture

The following principles serve as general guidelines for ecosystem development. The guiding principle is that all modern applications must be controlled via APIs. Data must not be encapsulated within a single system.

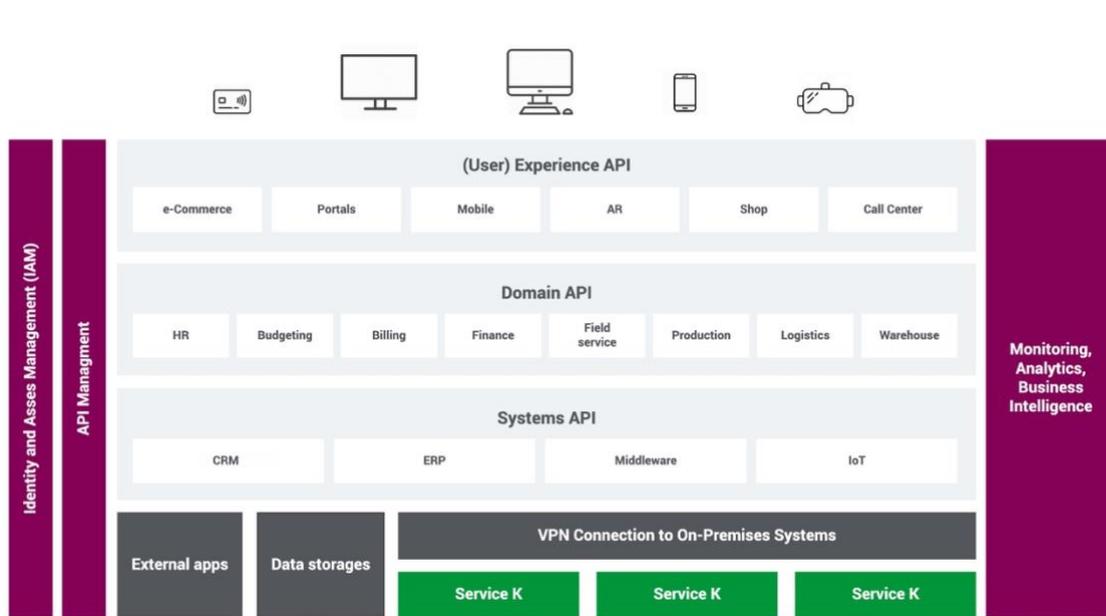
One of the keys to a successful and maintainable architecture is Domain-Driven Design (DDD) approach. Domain modeling is often overlooked, and individual systems are built in silos to meet immediate requirements. However, this leads to problems later in the process and often creates a large refactoring requirement in the ecosystem lifecycle.

Core principles of the ecosystem approach

1. Product-friendly architecture <ul style="list-style-type: none">• Product-friendly and vendor-agnostic architecture• Use of proven products where appropriate• Preference for products with high-quality APIs• Avoidance of complex adaptations of COTS products	2. Microservice architecture <ul style="list-style-type: none">• "Smart endpoints and dumb pipes"• Lightweight API management over heavyweight ESBs• Manageable number of services• Expand products beyond their original capabilities
3. Domain-based architecture <ul style="list-style-type: none">• Definition of architecture domains (Domain-Driven Design)• Provides domain APIs• Ongoing communication with API consumers for a common understanding• Hides the complexity of the underlying products	4. Cloud-based architecture <ul style="list-style-type: none">• Focus on deployment of cloud-based services• Use of reliable, certified data centers/cloud providers• Improved flexibility, productivity and stability• Improved security against cyber attacks



3.1.1. Layered API architecture



System APIs are the native APIs that provide access to central back-end systems, such as local legacy systems, external services and various data stores. The idea is to provide the API user with a clean and reusable API interface while protecting the user from the complexity or changes to the underlying systems. System APIs are for internal use only and point-to-point integrations with system APIs should be avoided.

Domain APIs are responsible for modeling and encapsulating the business units and processes of the enterprise. Domain APIs shape the data by orchestrating (split, aggregate, route, etc.) and calling one or more system APIs. Domain APIs are generic and can be used by both internal and external systems. Domain APIs are generated by the service layers.

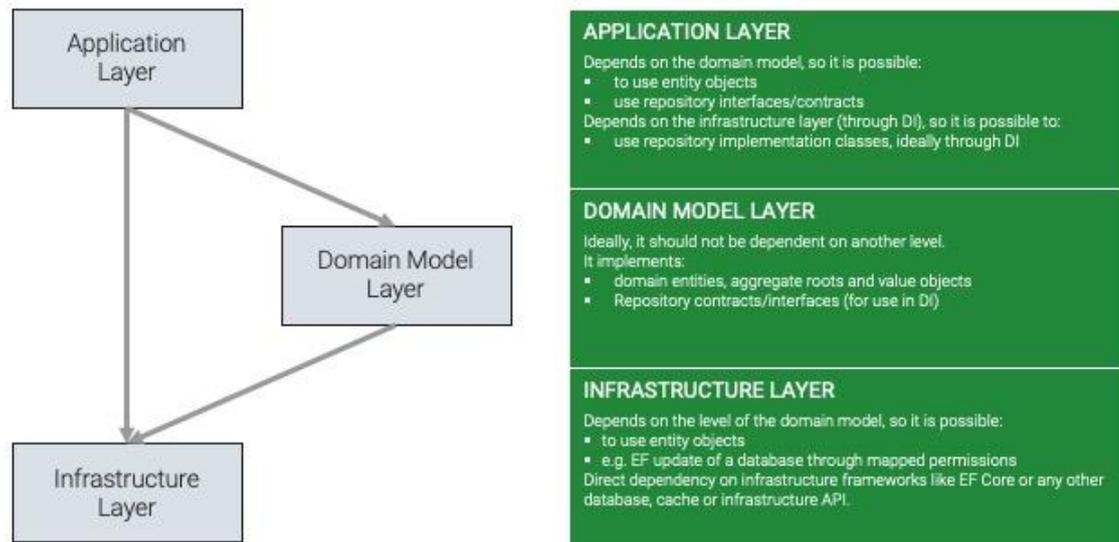
Experience APIs are designed for a specific user experience or audience, such as a customer portal, mobile application, or IoT device. Experience APIs are the means by which data can be reconfigured to be most easily consumed by the target audience.



3.1.2. Domain-driven design-based microservice architecture

For a layered architecture based on DDD principles, the dependencies between the layers should follow the principles defined in the following diagram:

N



3.1.3. API documentation

The **OpenAPI** standard is used to describe and document individually developed APIs. The API documentation is online and based on a source code (and always up to date).

Swagger and API management tools provide a web-based user interface for quick testing of the APIs.

3.1.4. Benefits of domain-driven design and API-driven architecture

The combination of domain-driven design with API-driven architecture enables to:

- Identify different business domains and create well-defined interfaces (domain APIs) for these domains.
- Identify dependencies between different domain concepts and entities and define bounded contexts that serve as a blueprint for a microservice-based architecture.



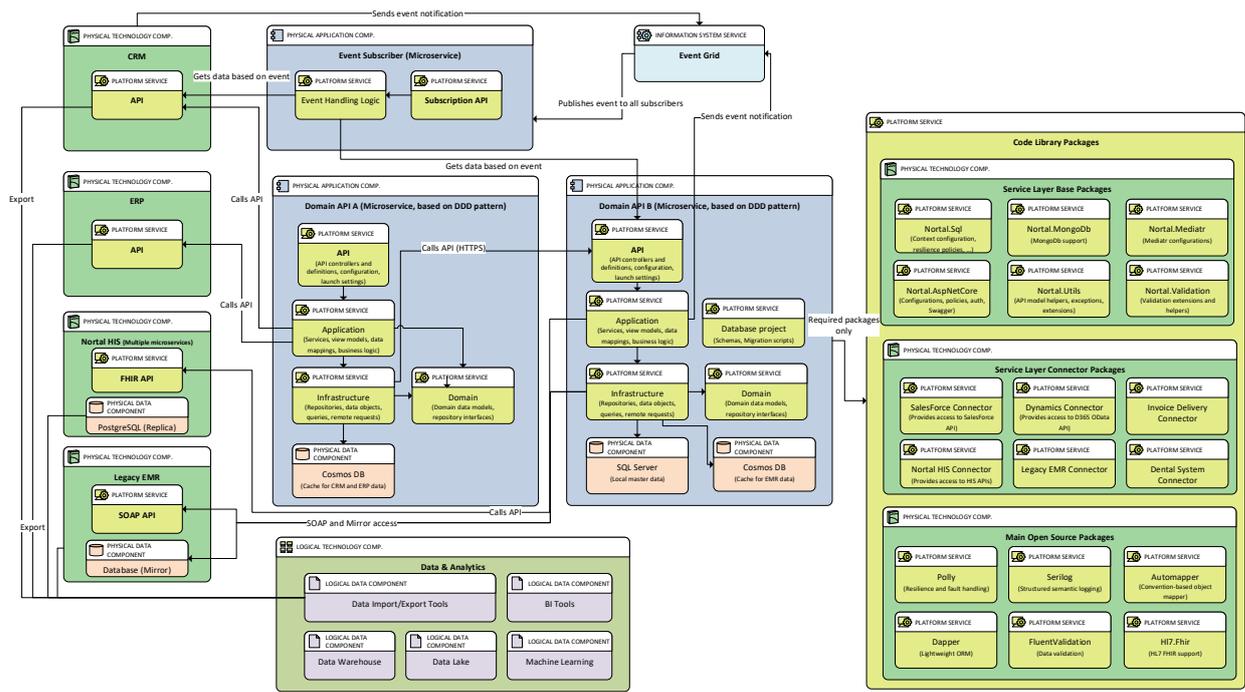
- Loosely couple different products and other ecosystem elements. As long as the domain APIs remain intact, processes will continue to function relatively independently of the future product choices.
- Model business units based on company specifics rather than following the logic of specific products. This simplifies the work of the IT department and IT partners, especially by making it easier to add new components to the architecture.
- Create a foundation from COTS products. A key advantage of Commercial of The Shelf (COTS) products is that they are reliable as long as they do not contain modifications. Internal customization of the products should be avoided. Instead of adjusting a product by hook or crook into the business logic of the company, it should be inserted into the service layer so that the product remains intact and ready for upgrades.

3.2. Service Layer Architecture

Proven products are used where applicable, but tailored *service layers* are an important part of our ecosystem approach. Service layers provide a way to capture and define different domains within the organization and make them work together efficiently. In addition to integrating different ecosystem domains and components, service layers provide a foundation for common data models and custom business logic. While product customization has always been problematic, it has become increasingly difficult and cumbersome with modern cloud-based solutions.

Service layers are created using robust and proven back-end technologies. Data models and business logic are combined into coherent units and implemented as microservices. Common utility libraries are distributed via package managers so that they can be used in all microservices. Continuous integration and deployment pipelines are established at an early stage to ensure smooth development and deployment. API Management Services are used to centrally manage API documentation, access control, throttling and service endpoints. Infrastructure-as-code (IaC) solutions are used to automate all aspects of setting up cloud environments.

The diagram below shows both an enterprise architecture for a given scenario (healthcare ecosystem) and the internal architecture of microservices provided by the service layer. The scenario contains two different microservices with inter-service dependencies, core products as the main building blocks of the ecosystem, and a data analysis layer.



3.2.1. Service layer development principles and core features

The following principles are adopted for the development of service layers:

- › Use of DevOps (CI and CD)
- › Microservices architecture with a manageable number of services
- › Reusable base libraries published via a package manager
- › Asynchronous internal pipelines
- › DI (Dependency Injection) -friendly architecture
- › Internal architecture according to the principles of Domain-Driven Design
- › Clear responsibilities between the different levels
- › Convention-based model mapper for data transformations
- › Fast and robust in-memory and out-of-memory caching
- › Policy-based resilience and error handling
- › Support for multiple authentication and authorization providers
- › Support for data versioning and auditing
- › Support for semantic logging and various log providers
- › Online automatically generated API documentation (Swagger/OpenAPI)
- › Comprehensive test automation
- › Infrastructure-as-code (IaC) tools that are used when needed (namely Terraform in cloud solutions)



3.3. Cloud architecture

Benefits of a cloud-based solution compared to an on-premise solution:

- › More robust and secure operating environment (highly certified data centers and employees)
- › Faster setup time
- › Automatic version upgrades behind the scenes
- › Practically unlimited scalability
- › Lower TCO (Total Cost of Ownership)
- › Predictable service fees, no unexpected maintenance costs
- › Advanced PaaS services that cannot be used in on-premise solutions

The connectivity between the cloud and local systems can be established to create a hybrid solution that can access data both from the cloud and on-premise.

Different types of cloud services are introduced in the next chapters.

3.3.1. Infrastructure-as-a-Service (IaaS)

With IaaS, the required virtual machines (VMs) as well as the associated network and storage components are provided. The required software and applications are then deployed on these VMs. This model is closest to a traditional local environment, except that the cloud provider manages the infrastructure. While other types of services are on the rise, IaaS is still a useful scenario for many areas, especially in combination with container technologies.

3.3.2. Platform-as-a-Service (PaaS)

PaaS provides a managed hosting environment where organizations can deploy their application without having to manage VMs or network resources. Instead of creating and managing individual VMs, an initial configuration is defined, and the service delivers, configures, and manages the requested resources. API management and database-as-a-service are examples of PaaS services.

Our approach is to leverage PaaS services as much as possible. VM-based IaaS is only used when PaaS services are not available or when their use is not appropriate for technical or business reasons.



3.3.3. Functions-as-a-Service (FaaS)

FaaS goes even further by eliminating the need to take care of the hosting environment. Instead of creating compute instances and deploy code for these instances, code is simply provided, and the service automatically runs it. There is no need to manage compute resources. These services leverage the serverless architecture and scale seamlessly up or down to the level needed to handle traffic. Azure Functions and AWS Lambda are examples of FaaS service.

FaaS services are relatively new and evolving rapidly. More experience in versioning, configuration management, and other core areas needs to be gathered before utilizing them in business-critical components. FaaS is on the technology radar but will not be used extensively until they reach a more mature state.

3.3.4. Software-as-a-Service (SaaS)

SaaS offers a complete software solution that is purchased on a pay-as-you-go basis from a cloud service provider. Apps can be rented by companies and used over the Internet, usually with a web browser. All underlying infrastructure, middleware, app software and app data are located in the data center of the service provider. The service provider manages the hardware and software and ensures the availability and security of the app and data with the corresponding service contract. Power BI is an example of a SaaS service.

3.4. System integrations

The number of system integrations in large companies tends to be very high. The traditional solution for managing integrations has been to acquire a centralized ESB (Enterprise Service Bus) and concentrate all integration logic there. However, integrations are more complex in the real world than in sales pitches, and this can lead to a situation, where ESB becomes the default location for business logic, which was not originally planned this way. Slowly, the ESB is evolving into an uncontrollable monolith on which many business processes depend.

Modern cloud-based solutions avoid the monolithic ESB solution by providing the functionalities in smaller packages. Our approach is to use these lightweight ESB PaaS components, such as API management, for what they do well, but to place the business logic in the service layers.

API management is used to manage all APIs in the ecosystem. APIs can be published to external parties, partners and internal developers. API Management provides a portal for API publishers and a self-service portal for developers. In addition to managing and documenting the APIs, API Management acts as a central gateway that controls access



to the APIs. It supports modern authentication and authorization standards for securing APIs.

Cloud-based message queues, such as the Azure Service Bus, provide reliable messaging between ecosystem applications. They support asynchronous operations, message brokering and publish/subscribe functions.

Azure Logic Apps offers a wide range of integration tools, from job schedulers and visual workflow editors to on-premise connectors. It is closest to the traditional ESB, but with one key difference: you can choose when you want to use it and pay only when you need it.

3.5. Logging and monitoring

Performance and other problems have a significant negative impact on business. With multiple components and frequent releases, problems will inevitably occur at some point. The goal is to predict problems before they occur, or to find and fix problems before they are even noticed.

This requires advanced diagnostic functions and a central logging solution. Azure Application Insights and Log Analytics can be used as the basis for the monitoring platform. On-site logs can also be forwarded there for further analysis.

It is important to establish basic rules for ecosystem applications. All applications should be able to send logs to central logging as early as possible.

3.6. Updates and version upgrades

One of the benefits of the new ecosystem and associated DevOps practices is that they allow frequent updates and upgrades. Update processes and responsibilities for Nortal's software can be fully defined during the project. For multi-tenant cloud services such as Microsoft Dynamics 365, updates and version upgrades are managed by the cloud service provider.

Updates are scheduled to minimize production system downtime. Any update or upgrade that causes downtime is carefully planned and clearly communicated to all parties involved. DevOps practices are used in development, but all production updates pass through a strict approval process.



3.7. Reporting and business intelligence

Existing reports created on legacy systems can be still used, but it makes sense to develop new business intelligence capabilities on **Microsoft's Power BI platform**. Users can create advanced reports and publish them for use on the Web and mobile devices. Anyone can create personalized dashboards with a 360-degree view of their business.



4. Case study: Digital ecosystem in healthcare

4.1. Background

Due to the major changes in healthcare in Finland, there are new market opportunities for private healthcare companies, but due to the current IT landscape, which is mainly based on monolithic legacy products, the enterprise architecture no longer meets the current and future business needs of healthcare providers.

Many healthcare systems were originally built for billing purposes only and then expanded to include structured electronic medical records, e-booking systems, customer relationship management systems and integrations with national and regional healthcare systems. The solutions are either completely monolithic or only internally modular. Very few systems support a truly modular approach, where the system leaves room for third-party solutions and does not attempt to control all data and business processes.

In order to solve customers' current problems, achieve important goals, and prepare for the future, Nortal uses the digital ecosystem approach for a major player in the Finnish healthcare industry. It is based on proven products, limited product customization and the service layer approach to complement products and create robust integrations.

4.2. Solution overview

We are able to work with any product that provides APIs (or database access), but we recommend using modern products that provide comprehensive, well-documented REST APIs. The core products selected for our largest healthcare customer cases are Nortal's HIS, Microsoft Dynamics 365 Sales and Customer Engagement and Microsoft Dynamics 365 Finance and Operations.

Nortal's HIS is the leading hospital information system in the Baltic countries. It is a comprehensive solution for the operations of a clinic or a hospital and has been in use since 2007. The system offers fully web-based user interfaces and FHIR compliant REST APIs for data management and orchestrating the processes. Nortal has succeeded in combining several generations of technology in a single solution. The first fourth generation modules are in production and we are currently developing the solution further to meet the needs of the Finnish and global market.

Microsoft Dynamics 365 for Sales and Customer Engagement is a cloud-based CRM/xRM system that supports the management of accounts, contacts, products, sales agreements and their various relationships. It provides tools for managing sales



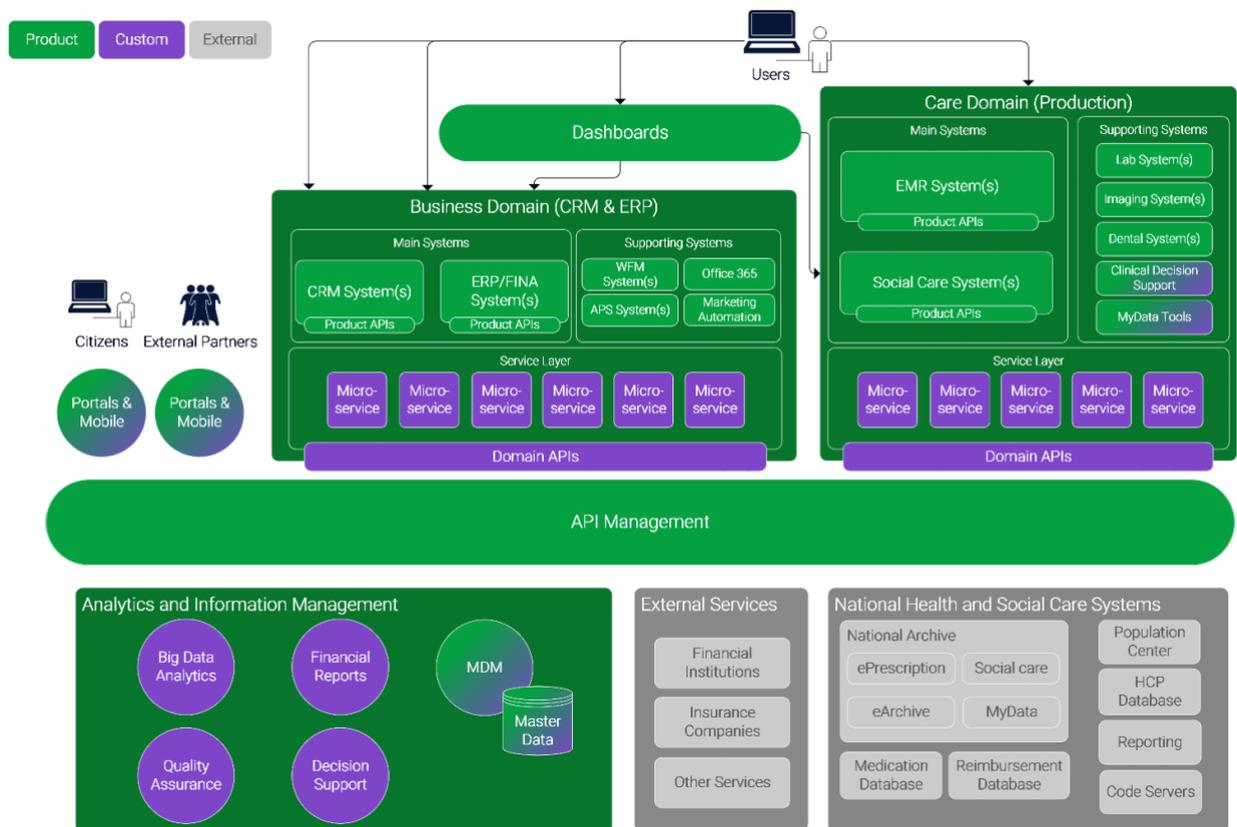
pipelines, including prospect for cash pipelines, and advanced reporting via Power BI. In addition to out-of-the-box functionality, it provides a powerful dynamic data and forms engine that allows you to create custom solutions without modifying or overwriting the original product code.

Microsoft Dynamics 365 for Finance and Operations is a cloud-based, modular Enterprise Resource Planning (ERP) system for financial management, project accounting, supply chain and warehouse management, manufacturing, retail, and talent management. It is an integrated system in which everyday transactions automatically generate accounting transactions, significantly reducing manual workload.

Both Microsoft Dynamics 365 for Sales and Microsoft Dynamics 365 for Finance and Operations are part of the Microsoft 365 product line.

4.3. High-level enterprise architecture

The high-level enterprise architecture for the healthcare provider is shown in the following figure.

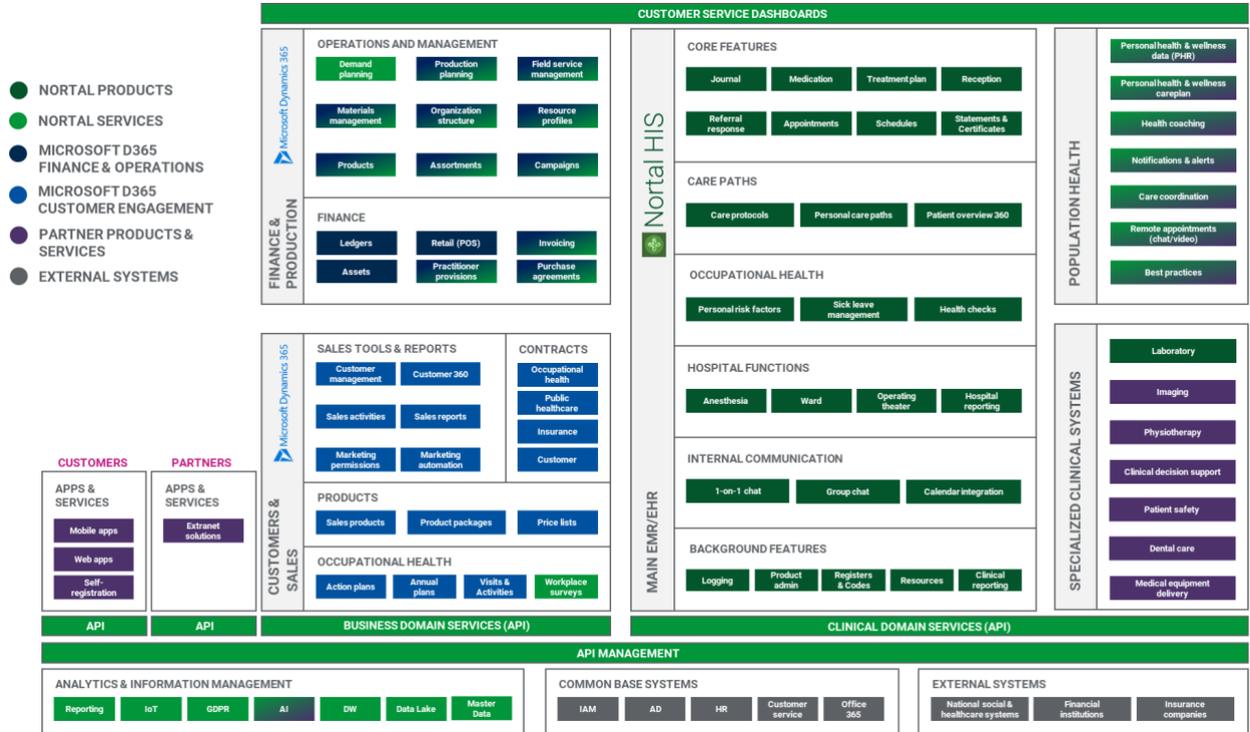


Green boxes show the core offering of Nortal. Modules marked in purple can be delivered together with our partners.



4.4. Functional architecture

The following diagram illustrates our approach to sharing responsibilities between the different systems.



4.5. Solution details

For more information about Nortal's novel healthcare ecosystem or other solutions, please contact your local Nortal representative.



5. Nortal is building a seamless society globally

5.1. About Nortal

Nortal has 20 years of experience in digitizing governments and healthcare institutions in Germany, in the Nordics and the GCC region, as well as creating a competitive edge and agility for large businesses, industry and manufacturing companies in Germany and the U.S. Combining that with the unique experience of transforming Estonia into a digital leader, our vision is to build a seamless society, where all communications and transactions are effortless. Nortal doesn't merely deliver software — we are known for supporting successful structural reforms, focusing on the underlying impact. We operate in 10 countries and employ more than 1000 specialists who carry out high-impact digital transformation projects across the globe.

5.2. How we work

We combine strategy, long-term business sector know-how with data-driven solutions and innovative technologies, that help our customers to simplify and optimize complex processes for efficiency, sustainability and agility. We take full ownership of digital transformation from strategy to technology, thereby helping our customers tackle challenges and prepare for the future. Be it public or private sector, government, large businesses, healthcare, manufacturing or industry.

Our methodology concept of building innovation concentrates on 3 enter points:

- AS-IS mapping and ROI or business case analysis;
- Hackathon or Proof of Concept (PoC), including solution prototyping and testing. TO-BE mapping with feasibility analysis and solution development;
- Agile project development with the solution production.

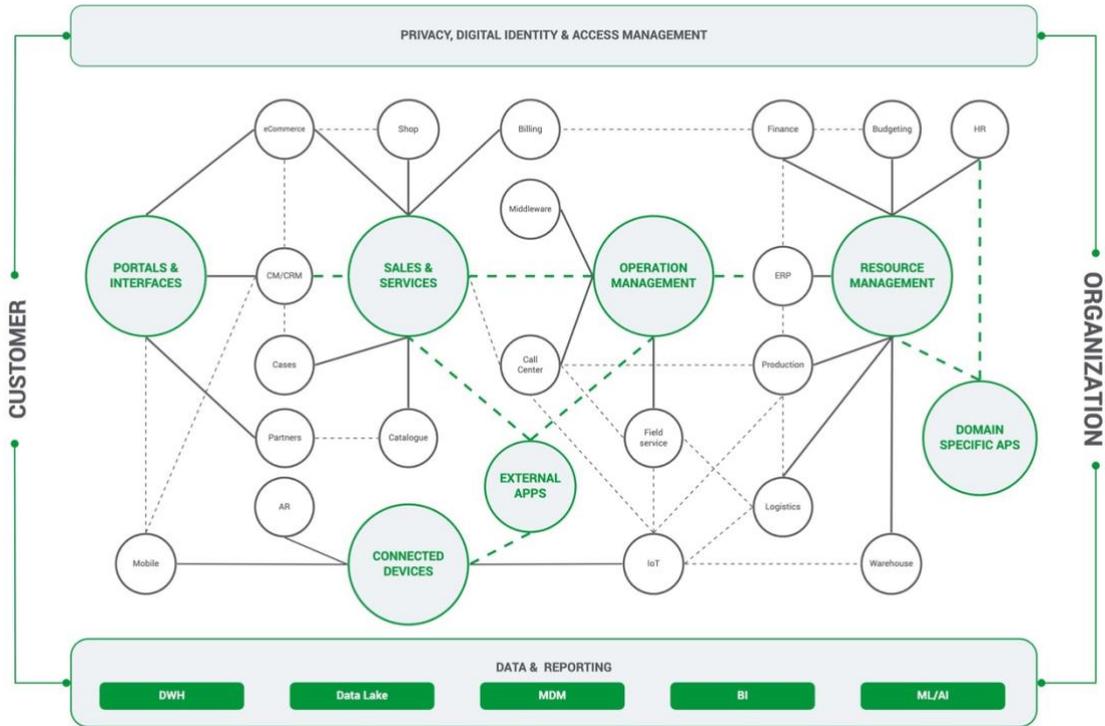
All stages of the process are carried out via the interactive approach of test and adjust, that enables to bring immediate value and visible results.

Furthermore, exploiting the opportunities of Modern Enterprise Ecosystem requires a careful analysis and assessment of the organization's maturity and technological environment, including:

- Clarity of strategy and goals;



- Change management;
- Business processes;
- IT system landscape and data governance.



A decorative graphic consisting of multiple thin, light gray wavy lines that flow from the bottom left towards the top right. Scattered along these lines are several small circular dots, some in a dark green color and others in a light gray color. A large, solid dark green circle is positioned in the upper right quadrant, containing the text 'Read more: nortal.com'.

Read more:
nortal.com