# Platform for Banks: Key Benefits, Architecture, and Modules

Platform: NextGen

Release Version: Release 11.7

### Table of Contents

- Key Benefits of the Platform for Banks
- Platform Design Elements (UI/UX)
- Platform Architecture and Modules: Brief Overview for Banks
  - o Admin
  - o <u>AgentAdditionalService</u>
  - o Agents
  - o **ApplicationsManagement**
  - o ApplicationTimeService
  - o **AppStateTools**
  - o Audit
  - o **Auditor**
  - o AuthenticationServer
  - o CleanDataTools
  - o **ConfiguratorWeb**
  - o <u>CountersCalculator</u>
  - o <u>FormsDesigner</u>
  - o <u>IdentityProvider</u>
  - o <u>Integration.IBM</u>
  - o <u>IntegrationService</u>
  - o <u>JtcDmsFakeServer</u>
  - o <u>LogServer</u>
  - o <u>Maven</u>
  - o <u>NotificationService</u>
  - o **ScoringScripts**
  - o <u>Lib</u>
  - o <u>Tools</u>
  - Data Layer and DBMS Support

## Key Benefits of the Platform for Banks

The platform is an ecosystem for the digital transformation of a bank.

	Description	
Process Automation	KYC, scoring, compliance, anti-fraud, applications, integrations	
Flexible Integration	REST, SOAP, queues, file exchange, scalability	
Security	Centralized audit, roles, digital signatures, encryption	
Operational Analytics	Monitoring, reports, BI, SLA control	
Fast Implementation	Modularity, plugins, visual editors, API	
Cost Reduction	Component reuse, automated testing	
Focus on Innovation	Support for event-driven, ML/AI, microservices, containerization	

## ✓ Advantages for the Bank

- Comprehensive automation: rapid deployment of new products, reduced manual labor
- Flexible integration: connection to any bank systems and external services
- **Security:** compliance with KYC, AML, GDPR, centralized control
- Analytics: process transparency, efficiency control, risk reduction
- Scalability: adaptation to any volumes and requirements
- **Innovation:** support for modern technologies and approaches

## Platform Design Elements (UI/UX)

### Main Interface Elements

- Forms and Wizards dynamic forms, auto-completion, step-by-step scenarios
- Tables and Grids filtering, sorting, export, drag&drop
- Dashboards and Widgets charts, KPIs, metric visualization
- Modal Windows confirmations, notifications, tooltips
- Menus and Navigation multi-level menus, breadcrumbs, quick search
- Cards and Lists tasks, events, notifications
- Filters and Search advanced filters, view saving
- Talendar and Scheduler drag&drop tasks, event visualization
- Charts and Diagrams line, bar, Sankey, workflow

- Security authentication forms, MFA, status indicators
- **Multilingual** language switching, localization
- **Responsiveness** support for desktop, tablet, mobile

### Principles and Patterns

- **Unified style:** corporate colors, icons, typography
- Minimalism: concise forms, focus on key actions
- Accessibility: keyboard navigation, alt-texts
- / Interactivity: animations, feedback
- Modularity: reusable components
- **§ Security:** confirmations, indicators, error protection

### **Visual Element Examples**

- Button: [Save changes](#)
- Attention:
  - Changes will take effect immediately after saving.
- Table:

Field Name	Туре	Description
client_id	string	Client identifier
status	enum	Application status
created_at	datetime	Creation date

### - Checklist:

- o [x] Check access rights
- o [] Fill in required fields
- o [] Submit for approval
- Code/script:

```
import requests
response = requests.get('https://api.bank.com/v1/clients')
print(response.json())
```

Diagram (Mermaid):

```
graph TD
A[User] -->|Data entry| B(Form)
B --> C{Validation}
C -- ok --> D[Save]
C -- error --> E[Error]
```

### Platform Architecture and Modules: Brief Overview for Banks

Below is a structured overview of all key platform modules. For each module, the purpose, main functions, benefits for the bank, and architectural features are indicated.



### **Purpose:**

Centralized platform administration: user, role, license, access, settings, and monitoring management.

### Main functions:

- Luser and role management
- Licensing and access control
- System parameter configuration
- Server services (web-service ASMX)
- Plugin support for functionality extension
- Data migration and updates
- Interface localization
- Logging and audit

### Benefits for the bank:

- Centralized access and security management
- Flexible role and permission configuration
- Integration with external systems via plugins and API
- Audit of user and administrator actions
- Scalability and extensibility

### **Architectural features:**

- Modular structure (core + plugins)
- Server part as a web-service (ASMX)
- Migration and update support
- Localization and multilingual support



## AgentAdditionalService

### **Purpose:**

A service to support and extend the functionality of the main agent system, providing additional service tasks and event processing.

### Main functions:

Management of signal groups and client information

- © Collection and processing of completed applications
- ✓ Temporary file cleanup
- Recovery of asynchronous results
- Service and application information provision
- Zogging and extended log properties
- Middleware and request processing filters

- Increased reliability of business processes
- Reduced manual labor in application processing
- Flexible integration with other modules via API
- Improved monitoring and system state management

### **Architectural features:**

- Modular structure with services and controllers
- Middleware and filters
- Unit testing
- Configurability via appsettings.json
- Logging via log4net



#### Purpose:

Automation, orchestration, and monitoring of business processes based on events, signals, and workflow. The core of the process management system, supporting integration with queues, external services, custom handlers, and scalable scenarios.

### Main functions and subsystems:

- Engine: Core for event, signal, and workflow processing, node registration and management, extensibility support (custom functions, dynamic nodes).
- EngineLib: Business logic library and auxiliary components for the core.
- Schedulers: Task and event schedulers, support for cron and custom schedules.
- ServiceStarter: Service for launching and managing the lifecycle of agents and services.
- **WatchDog:** Monitoring agent and process states, automatic recovery and notification.
- Appointments: Management of appointments, events, triggers within processes.
- Flow.Signal.ExternalNodeExecutionReset: Support for external signals and events to manage process states.
- **Autonodes.WaitingNode:** Automatic nodes for waiting for events, signals, conditions.

- AgentWindowsService: Service for running and managing agents in Windows environment.
- Testing: Tests, Translator.Test infrastructure for testing business logic, scenarios, integrations.

- 🙀 Automation of complex business processes, service scenarios, compliance, KYC, anti-fraud, etc.
- Flexible configuration of routes, event processing logic, integration with internal and external services
- Scalability and modularity for new product and process implementation
- Monitoring, audit, resilience, and self-healing of processes
- Reduced manual labor costs, increased speed and quality of service

### **Architectural features:**

- Modular architecture: core, handlers, schedulers, services, monitoring, tests
- Use of workflow, event-driven, state machine patterns
- Extensibility via plugins, custom functions, dynamic node registration
- Integration with queues, external services, scalability support
- Documented architecture (PlantUML diagrams, tests)
- Localization, multilingual support

## ApplicationsManagement

**Purpose:** Module for managing the lifecycle of applications, processes, and their states, providing an API for integration with other systems and client applications.

### Main functions:

- Management of applications and their states
- In Monitoring the execution state of processes
- Service health check
- Client library for integration
- OpenAPI/Swagger support
- Zogging and configurability

### Benefits for the bank:

- Centralized application management
- Transparency and control over process execution
- Easy integration via standardized API
- Automation and monitoring of the application lifecycle

### **Architectural features:**

Modular structure: API, business logic, client

- OpenAPI/Swagger
- Proposition (Dockerfile)
- Zogging and monitoring

## ApplicationTimeService

**Purpose:** Module for collecting, storing, and providing information about the execution time of applications and processes, as well as analyzing business process performance.

#### Main functions:

- II Collection and provision of data on application execution time
- API for obtaining processing time analytics
- Use of DTOs for standardized data structure
- Zogging and configurability

### Benefits for the bank:

- Process performance analytics
- SLA monitoring and processing time control
- Process optimization based on data
- > Integration with other modules

#### **Architectural features:**

- Modular structure: service, DTO, middleware
- API versioning
- II REST API
- Zogging and configuration support

## AppStateTools

**Purpose:** Module for managing application states, archiving, cleaning, monitoring, and working with operations within business processes.

- Retrieval, archiving, and cleaning of application state data
- Management of work operations and tasks
- In Memory usage control by applications
- Is Service operations with archives
- II Support for various storage mechanisms
- Zogging and configurability

- Centralized management of application data lifecycle
- Reduced storage costs through automatic archiving and cleaning
- Monitoring and resource usage control
- Increased reliability of business processes
- Flexible integration with other modules

### **Architectural features:**

- Modular structure: API, business logic, storages, tests
- Support for various storage mechanisms (DB, file system)
- 🚺 REST API
- Zogging and configuration support

## Audit

**Purpose:** Module for auditing, monitoring, storing, and analyzing events, states, and metrics of business processes and applications.

### Main functions:

- Keeping and providing event audit
- Management and export of attachments
- Monitoring and analytics of state metrics
- Integration with external services for data acquisition
- Working with data sources and their queries
- Use of DTOs for standardized data structure
- Zogging and configurability

### Benefits for the bank:

- Transparency and control of all key events and changes
- Quick response to incidents and anomalies
- Compliance with regulatory audit requirements
- Integration with BI systems and analytics platforms
- Increased trust in the platform through full traceability

### **Architectural features:**

- Modular structure: API, business logic, DTO
- API versioning
- II REST API
- Zogging and configuration support

## **Auditor**

**Purpose:** Module for advanced audit, analysis, OLAP reporting, and management of events and operations in business processes.

### Main functions:

- Server part for processing and storing audit events
- OLAP reporting and analytics
- III Plugin support for functionality extension
- Susiness logic and integration library
- P Security and access management
- Mapping and processing of business events
- Testing and performance tuning

### Benefits for the bank:

- Deep audit and analysis of operations, compliance with regulatory requirements
- CLAP reports for business analytics
- Flexible configuration and extension via plugins
- increased transparency and manageability of processes
- Integration with other platform modules

#### **Architectural features:**

- Modular architecture: server, library, plugins, OLAP
- Extensibility via plugins
- Web services (ASMX)
- Support for testing and performance tuning

## AuthenticationServer

**Purpose:** Module for centralized user authentication on the platform, supporting various login scenarios (web, desktop), session management, and working with digital certificates.

- Puser authentication for web and desktop clients
- Windows authentication support
- P User session management
- Password change
- Retrieving user and certificate information
- Ontegration with security policies and digital signature
- Zogging and configurability

- Single authentication point for all services
- Support for corporate security standards
- Flexible integration with internal and external systems
- Increased security and access manageability
- Compliance with regulatory requirements

### **Architectural features:**

- Web service (ASMX) with SOAP header support
- S Integration with UserManager, SessionManager, DigitalSigningProviders
- III Flexible configuration via configs
- III DI (StructureMap)

### ✓ CleanDataTools

**Purpose:** Module for automating data cleaning, processing, and management within business processes and platform applications.

### Main functions:

- API for launching and managing data cleaning tasks
- Implementation of data cleaning and processing strategies
- Management of work tasks and operations
- Zogging and configurability
- Support for testing and extensibility

### Benefits for the bank:

- Reduced data storage and maintenance costs
- 🙀 Increased process performance and reliability
- Compliance with data storage and deletion requirements
- Flexible integration with other modules

### **Architectural features:**

- Modular structure: API, business logic, worker services
- Use of strategies and interfaces
- 🚺 REST API
- Zogging and configuration support

## ConfiguratorWeb

**Purpose:** Module for centralized management of directories, strategies, schedules, and locales, providing an API for business process configuration and integration.

#### Main functions:

- II Directory and data management
- Istrategy and schedule management
- Working with schedules based on cron expressions
- III Account and locale management
- Repositories for storing and processing configuration data
- Zogging and configurability
- Swagger/OpenAPI support

### Benefits for the bank:

- Centralized and flexible management of directories and strategies
- Automation and scheduling of operations
- Fast integration via standardized API
- Multilingual support

### **Architectural features:**

- Modular structure: API, repositories, business logic
- API versioning
- REST API and OpenAPI
- Zogging and configuration support

## CountersCalculator

**Purpose:** Module for calculating, updating, and managing counters and metrics within business processes and platform applications.

### Main functions:

- Lalculation and updating of various counters
- Initialization and management of event listeners
- Business logic testing support
- Zogging and configurability
- Service installation and configuration

### Benefits for the bank:

- Automation of key metric calculations
- Process state monitoring
- Flexible integration with other modules
- Increased process transparency

### **Architectural features:**

- Modular structure: service, business logic, tests
- Event-driven model for counter updates
- Service interfaces
- III Unit testing

## FormsDesigner

**Purpose:** Module for visual design, configuration, and integration of user forms, documents, and interfaces.

### Main functions:

- Server part for form processing and storage
- Visual editor for forms and interfaces
- Business logic and integration library
- Digital signature and security support
- Working with data sources and strategies
- Interface localization and multilingual support
- Test data and examples

### Benefits for the bank:

- Rapid creation and deployment of new forms and documents
- Flexible interface configuration
- integration with internal and external systems
- Digital signature support
- Fast rollout of new products

### **Architectural features:**

- Modular architecture: server, editor, business logic
- Web services (ASMX)
- Localization and multilingual support
- Interest Extensibility via strategies and data sources

## IdentityProvider

**Purpose:** Module for managing user identification, authorization, and authentication, providing APIs and interfaces for working with accounts.

- Puser authorization and authentication
- Account and session management

- Support for various login scenarios
- Zogging and configurability
- S Integration with other modules

- Centralized access management
- > Increased security
- Flexible integration with bank systems
- Support for SSO, MFA, etc.

### **Architectural features:**

- Modular structure: controllers, view models, services
- II REST API
- Logging and configuration support

## 

**Purpose:** Module for integration with external IBM services, particularly Watson (AI, NLP), providing an API for message exchange and data processing.

### Main functions:

- API for interaction with Watson and other IBM services
- Message processing and routing
- Integration parameter configuration
- Logging and configuration support

### Benefits for the bank:

- integration with AI services for automated request processing
- implementation of intelligent assistants and chatbots
- Platform functionality extension
- > Improved service quality

### **Architectural features:**

- Modular structure: controllers, business logic, configuration
- REST API
- II Flexible integration parameter configuration

## ■ IntegrationService

**Purpose:** Module for integration with external and internal services, routing and processing integration events, providing an API for data exchange between systems.

- API for integration and request routing
- Asynchronous processing of integration events
- Support for various integration scenarios
- Authentication and session management
- Zogging, tracing, and error handling
- Extensibility via event handlers

- Centralized integration with bank services
- Automated data exchange
- > Increased reliability of integration processes
- Flexible event routing and processing
- Compliance with security requirements

### **Architectural features:**

- Modular structure: services, controllers, event handlers
- Synchronous and asynchronous scenarios
- Web services (ASMX, WCF)
- Zogging, tracing, configuration support

## JtcDmsFakeServer

**Purpose:** Emulator module for integration with an external document management system (DMS), implementing typical document exchange scenarios via SOAP.

### **Main functions:**

- Zreation, upload, and search of records/documents
- In Emulation of working with files and metadata
- SOAP interface for integration
- Zogging and configurability

### Benefits for the bank:

- Safe testing of DMS integration
- Automated document exchange
- Reduced risks during integration implementation
- Expandability for bank needs

### **Architectural features:**

- Web service (ASMX) with SOAP support
- Modular structure: service, business logic, configuration
- Example 2 Data storage repository

- S Integration with other modules

## LogServer

**Purpose:** Module for centralized collection, storage, processing, and analysis of platform logs and events, supporting various data sources and formats.

### Main functions:

- Service for receiving and processing logs
- Receiving logs via UDP
- Log storage and processing
- Lonsole application for management
- Zogging, support for various formats
- Unit testing

### Benefits for the bank:

- Centralized monitoring and event audit
- Fast incident and error analysis
- Compliance with audit requirements
- integration with external monitoring systems

#### **Architectural features:**

- Modular structure: service, UDP receiver, console, storage
- Support for various protocols and formats
- Integration with other services
- Zogging and configuration support

## Maven

**Purpose:** Module for designing, modeling, automating, and managing business processes, strategies, and scenarios.

- Server part for business logic processing
- Visual editor for strategies and processes
- Basic components and extensible modules
- Architectural diagrams and documentation
- II Plugin and custom function support
- Logging, help, and documentation

- Rapid design and deployment of new processes
- Flexible scenario configuration and automation
- 🔀 Integration with bank systems
- Scalability and extensibility

### **Architectural features:**

- Modular architecture: server, editor, components, plugins
- Web services (ASMX)
- Architectural diagrams and documentation
- Localization, help, custom functions

### NotificationService

**Purpose:** Module for centralized notification management, message distribution, and real-time alerts.

#### Main functions:

- API for sending and managing notifications
- Real-time notifications via SignalR
- III Priority, template, and export management
- Notification distribution business logic
- Logging and configurability

### Benefits for the bank:

- rompt notification of employees and clients
- Increased response speed to events
- Centralized notification channel management
- Flexible integration with bank systems

### **Architectural features:**

- Modular structure: web, business logic, services, hubs
- Real-time notifications (SignalR)
- REST API and web interface.
- Zogging and configuration support

## ■ ScoringScripts

### **Purpose:**

Module for automating scoring calculations, predictive analytics, and data verification using Python scripts and machine learning models.

#### Main functions:

- Universal root script for running scoring models based on h5 files
   (h5\_model\_engine\_root\_script.py)
- Automation of calculations by business rules and formulas (BAS, Engine\_BAS\_Automation/BAS\_formulas.py)
- Image and biometric verification (face verification, Engine\_image\_verification/Engine\_script\_for\_face\_verification.py)
- Predictive scoring using neural networks (DNN,
   Engine\_nn\_scoring/Engine\_script\_for\_prediction.py)
- Error and result logging, CLI argument support
- In Flexible routing to the required engine depending on the model type (BAS, DNN, Face Recognition)

### Benefits for the bank:

- Automation of scoring processes (credit scoring, KYC, anti-fraud, etc.)
- 🙀 Fast implementation of new models and algorithms without changes to the platform core
- Support for complex business rules and formulas (BAS)
- 🙀 Integration with external ML/AI services and internal processes
- Scalability and extensibility of scoring scenarios

#### **Architectural features:**

- Modular structure: separate engines for BAS, DNN, image verification
- Universal root script for routing and launching the required engine
- In Support for h5 (Keras/TF), Excel, CSV formats
- LI interface for integration with other services and pipelines
- Zogging and error handling



### **Purpose:**

A large library of general, infrastructure, and specialized components used by all platform modules. Serves as the foundation for building business logic, integration, automation, UI, data handling, security, testing, and functionality extension.

### Main functions and subsystems:

- Workflow and processes: WorkflowLibrary, WorkflowLoader, WorkflowIntf tools for building, executing, and integrating workflows.
- UI and clients: WindowsClientLib, Windows.UI, WebClientLib, WinFormsPlugins, Web.DynamicPresentation.GridRenderingControls — components for building user interfaces (WinForms, Web).

- Integration and services: WebService, RabbitMq, S3Storage, RequestsManager.Lib,
   Telemetry.ApplicationInsights integration with external services, queues, cloud storage, telemetry.
- Data handling: NHibernateObjects, SpreadSheet, StatLibrary, SqlParser,
   ImportExportData, PacketDataFramework database access, data processing, statistics, parsing, import/export.
- Security and authorization: Security, Jose\_Jwt, SystemClientCache security mechanisms, tokens, caching.
- Localization and multilingual support: Localization, TranslationDictionary, Transliteration
   multilingual support, translations, transliteration.
- Logging and audit: Logging, LoggingEx, Signals.Tracing centralized logging, event tracing.
- Modules and plugins: Plugins, WinFormsPlugins, Modules platform extensibility.
- **III Formula and expression handling:** ExpressionLib, ExpressionEditor, Expressions calculations, formulas, expression editors.
- Testing and automation: UnitTests, TestsAutomation, TestsAutomation.Test,
   UnitTestUtils testing and automation infrastructure.
- Caching and utilities: GenericCache, UsefulUtilities, ObjectPooling cache, utilities, resource optimization.
- **III** File and format handling: Formatters, JsonToXsdConverter, HDF5 conversion and handling of various data formats.
- Analytics and reports: Reporting, Reports. Iterop, Patron Analytics tools for building reports and analytics.

- Unified base for rapid development and integration of new modules and services
- increased reliability and security through proven components
- Platform scalability and extensibility
- Reduced development costs through code reuse
- Support for modern integration, security, UI, and analytics standards

### **Architectural features:**

- III Modular and hierarchical structure: each component is a separate library/module
- II Support for various storage, integration, UI, security technologies
- Extensibility via plugins, modules, utilities
- Infrastructure for testing, monitoring, localization
- Li Use of modern design patterns (DI, caching, event-driven, ORM, etc.)



### **Purpose:**

A set of auxiliary tools and utilities for automation, administration, testing, integration, data migration, localization, monitoring, and increasing the reliability of the NextGen platform.

### Main functions and subsystems:

### State and process management:

AppStateTools.ClientApp, AppStatesViewer, AppStateSync — viewing, synchronizing, and analyzing application and process states.

### Logging and audit:

LogViewer, Web.Log — log viewing and analysis, event monitoring, quick error search.

### Testing and load:

ReliabilityTester, IntegrationServiceLoadTest, ExecutionTest, ExecutionTests.UI — reliability, performance, and integration testing.

### Substitution and multilingual support:

Localizer, Localizator, Langexport, NodesUsedLocalesActualizer — export/import of language files, multilingual support.

### Security:

PasswordHashCalculator, Encrypt — password hash generation and verification, data encryption.

### Data migration and processing:

SqlLoader, CsvMigrationToolDodrioCore, Generator\_Restr\_And\_Expr\_From\_Excel\_To\_States — migration, loading, generation of constraints and expressions from Excel/SQL.

### - © Configuration and automation:

ConfigurationDesigner, EnvironmentConfigValidation, EnvironmentConfigBeautifier — environment configuration design, validation, automation.

### Analysis and comparison:

Checker, ComparisonPropertiesListGenerator, DetailsDump — comparison, audit, property analysis, report generation.

### M Updates and licensing:

Publisher, IncrementalUpdate, NdxUpdater, LicenseUpdater, LicensingSystem — update automation, license management, new version publishing.

### Build automation and console utilities:

CommandConsole, BuildTasks, AsposeTransformer — build automation, console commands, document conversion.

### - III Table and strategy management:

StrategyVariablesEditor, SpreadSheetEditor — strategy variable editing, spreadsheet management.

### Configuration generation and integration:

GetFunctionsList, CreateSolutionsConfiguration — configuration generation, function list retrieval for integrations.

File transfer and format handling:
 FileTransfer.Service, HDF5.PipeClient — secure file transfer, working with HDF5 and analytics pipelines.

### Benefits for the bank:

- Automation of routine tasks and reduced IT staff workload
- increased reliability and security through testing and monitoring
- Faster implementation of new features and updates
- Support for multilingualism and localization for different markets
- Flexible integration with external and internal systems
- Simplified audit, diagnostics, and incident analysis

### **Architectural features:**

- Modular structure: each tool is a separate application or service
- Support for CLI, GUI, and web interfaces
- Integration with main platform modules via API, files, queues
- Expandability and customization for bank tasks
- Use of modern automation and DevOps patterns

## Data Layer and DBMS Support

### **Purpose:**

Providing universal, flexible, and scalable access to various industrial DBMS (MSSQL, PostgreSQL, Oracle, MySQL, SQLite) for all NextGen platform modules.

### Main functions and subsystems:

- ORM and data access: NHibernate, Entity Framework Core, Dapper, Enterprise Library Data Access Application Block
- Supported DBMS:
  - MSSQL (SQL Server)
  - o PostgreSQL
  - o Oracle
  - o MySQL
  - SQLite
- Factories and abstractions: universal connection factories, DI, repository patterns, multiclient support
- Configuration: connection string support via appsettings, .config, json, environment parameters
- X Auxiliary utilities: migrations, testing, DB state monitoring

### Implementation examples:

NHibernate:

```
configuration.DataBaseIntegration(c =>
{
    c.Driver<MicrosoftDataSqlClientDriver>();
    c.Dialect<MsSql2012Dialect>();
    c.ConnectionString = connectionString;
    // ...
});
```

Entity Framework Core:

```
switch (provider)
{
    case DbProviderKeys.SqlServer:
        optionsBuilder.UseSqlServer(connectionString);
        break;
    case DbProviderKeys.PostgreSQL:
        optionsBuilder.UseNpgsql(connectionString);
        break;
}
```

Dapper:

```
using (var connection = new
SqlConnection(_configuration.GetConnectionString("default")))
{
   var orderDetails = connection.Query(sql);
}
```

- Universal DbHelper:
  - Creating connections to MsSql, Oracle, PgSql, MySql, OleDb, SQLite via factory method.
- Connection string examples:
  - o In appsettings. Database. json, App. config., . json files for different providers.

### Benefits for the bank:

- Freedom of choice and hybrid use of industrial DBMS
- Scalability and fault tolerance of infrastructure
- Fast integration with the bank's existing systems
- Reduced data maintenance and migration costs
- Flexible configuration and architecture extensibility

### **Architectural features:**

- Abstraction over providers: easy to extend support for new DBMS
- Use of factories, DI, repository patterns
- Multi-client and multi-tenant support
- Auxiliary utilities for migrations, testing, monitoring

### **Conclusion:**

The NextGen platform supports direct work with major industrial DBMS through modern ORM and data access solutions. The architecture is built for flexibility, scalability, and easy integration with the infrastructure of banks of any scale.