

Integrating OutSystems with your ecosystem

TECHNICAL WHITE PAPER



Contents

Introduction	4
Easy and seamless integration	5
Web Service integration	7
Integrating to existing systems using SOAP web services	7
Integration extensions for SOAP Web Services	8
Integrating to existing systems using REST APIs	9
Integration extensions for REST APIs	10
Integration with external databases	11
Integration extensions for external databases	11
SAP Integration	12
Integration extensions for SAP	12
Integration extensions for other SoR platforms	13
Integration Builder	13
End-user authentication integration	14
Integration extensions for authentication and identity management	15
DevOps and CI/CD integration	16
DevOps	16
Performance monitoring	16
Business activity monitoring	18
Logging	19
Analytics APIs	20
Deployment	20

Contents

DevOps extensions.....	21
CI/CD	21
OutSystems automated code testing	22
OutSystems automated integration testing.....	23
Automated deployment.....	24
CI/CD Extensions.....	25
OutSystems pipeline accelerator.....	27
 AI integration	28
AI service connectors	29
Chatbots and voice-enabled apps	30
Chatbots	30
Virtual private assistants	31
Cognitive services to create AI-enabled apps	31
Language Services.....	31
Vision Services	33
Custom machine learning	34
Machine Learning Builder	35
 Conclusion	36

Introduction

End-users of your enterprise-grade application don't need to know—don't want to know—what's going on behind the UI. How privacy is secured, where data comes from, what powers AI; all that matters is that it is safe, reliable, and smart—and it must be as attractive, quick, and useful as the most beautiful, speedy, and helpful app available in the marketplace.

If it's not, there are always competitors to check out.

Those who set business goals, design the user experience, compile the metrics, and set budgets are less interested in the nuts and bolts of the process than the end result: as long as the development team meets the goals, satisfies the users, logs good results, and comes in under budget, all is well.

Developers, as stakeholders, are invested in achieving the same goals and meeting the same targets. Development, however, is all about process, and part of that process is the constant friction between “if it ain't broke, don't fix it” and “let's see if we make it easier, faster, and more efficient.”

This paper discusses how Outsystems provides seamless integration with the full range and scope of business process management tools, data sources, and core business systems. Not only does OutSystems speed up the entire development and deployment process with its built-in tools, but they can easily be supplemented by integrating the same third-party tools your enterprise has relied on for years.

Easy and seamless integration

Developers need the ability to create seamless integration with the full range and scope of the business process management tools, data sources, and core business systems that your enterprise has developed over the years—as well as keep pace with future developments.

OutSystems provides out-of-the-box capabilities to integrate with a myriad of external systems, and this toolkit can be extended with OutSystems-supported modules, connectors, UI components, and business solutions, available from the OutSystems Forge, a repository of reusable, open code.

The Forge is also an integral part of our thriving community many of whom have shared thousands of fully working, community-supported connectors to well known external systems, such as Google Drive, Dropbox, Sharepoint, Dynamics, to name just a few.

In cases when your enterprise requires integrations to third-party libraries or non-native technologies, developers can use Integration Studio, an OutSystems desktop tool, to write custom C# and .NET code to create an extension, making its actions, entities, and structures available in Service Studio.

In all cases, once an integration has been created and added to your environment, it can be used, and reused, by every developer for every app—as if they belong to OutSystems itself.

Among the integrations available include:

- Web services to both consume and expose REST and SOAP
- External databases
- Enterprise SoR for managing sales and collaboration

- End-user authentication
- DevOps and CI/CD processes and tools for building, deploying, automating, testing, monitoring, and logging
- AI and chatbots

There are many other integrations that don't exactly fit into any particular category, including:

- Augmented reality frameworks
- Messaging formats and protocols
- IoT gateways

This document discusses the wide range of integrations made possible by OutSystems.

Web Service integration

OutSystems has out-of-the-box capabilities to simplify the task of consuming and exposing functionality as standard SOAP web services and REST APIs without any hand-coding. Once OutSystems has generated the methods and data structures to integrate with the external system, developers can use them visually in the application logic of Service Studio. From a developer's perspective, there is no difference between invoking an OutSystems method, a SOAP web service, or a REST API.

Integrating to existing systems using SOAP web services

OutSystems supports consuming SOAP web services with WSDL 1.1 and SOAP 1.1/1.2.

Developers provide the location of the Web Services Description Language (WSDL) or import directly from an WSDL file. The OutSystems IDE inspects the WSDL and generates everything needed to invoke the web service methods.

When you consume a SOAP Web Service, OutSystems does the following:

- Connects to the web service and analyses the methods and structures
- Creates the OutSystems actions matching the consumed Web Service methods with the corresponding input and output parameters
- Creates the structures and static entities to hold the complex types defined in the WSDL
- Maps the XML data types into OutSystems data types

OutSystems translates the methods exposed by a SOAP Web Service into OutSystems actions, with the same semantics as any action created in Service Studio. From a developer's perspective, there is no difference between invoking a OutSystem action or a method exposed by an external service.

Authentication options are provided without the need of complex extensibility such as basic authentication (a static username and password) or dynamic login for credentials specific to each method that can be specified at runtime.

Once a SOAP web service has been integrated into your app, it **can be exposed** to make it available to other apps in your OutSystems environment or external web applications.

Integration extensions for SOAP Web Services

When developers want customized behavior, such as access to Windows Communication Foundation (WCF) for consumed SOAP Web Services, they can create an extension by using the **SOAP Extensibility API**, which enables you to modify the SOAP connection, request and/or response message using .NET code. It is automatically imported to your environment at installation.

SOAP Extensibility Samples, a community-supported extension available from the Forge, can be used by developers as is, or adapted as required, to implement common extensibility use cases using the SOAP Extensibility API.

For any use case that is not provided out of the box, developers can use **Integration Studio**, an OutSystems desktop tool, to write custom **C# and .NET code** to create an **extension**, making its actions, entities, and structures available in Service Studio.

Integrating to existing systems using REST APIs

You can integrate your OutSystems applications with REST APIs provided by other systems, or even by other OutSystems applications. Use this functionality either to obtain data from those systems or to request them to perform some action.

When importing the REST API, Service Studio does the following:

- Connects to the web service and analyzes the methods and structures
- Creates REST API methods with the corresponding input and output parameters
- Creates the structures to hold the corresponding input and output parameters under a new tree element with the REST API name
- Maps the REST data types into OutSystems data types

OutSystems translates the methods exposed by a REST API into OutSystems actions, with the same semantics as any action created in Service Studio. From a developer's perspective, there is no difference between invoking an OutSystems method or a method exposed by an external service.

When consuming a REST API, you can add logic to customize the information that is sent in the requests or received in the responses. **Simple customizations** are readily available in Service Studio to modify both the information of the original request (such as the URL, the request text or the headers) or the information of the original response (such as the status code or the response text).

Once a REST API has been integrated into your app, the consumed API can be used across OutSystems applications within the same environment, or external web applications, to retrieve or manipulate information.

Integration extensions for REST APIs

When developers want customized behavior, they can create an extension using the REST Extensibility API. These advanced customizations then can be reused in Service Studio. These advanced customizations allow, for example, using a specific algorithm to encode part of the request content, adding support for client-certificate authentication, or handling a complex response.

The OutSystems REST Extensibility API enables you to access the content of requests and responses used by methods consumed from other REST APIs. It's automatically imported to your environment at installation.

The OutSystems Forge contains hundreds of components, free for your use, that package the REST Extensibility API and which can make it quick and easy for your developers to integrate with many applications.

Integration with external databases

OutSystems connects, out-of-the-box, to SQL Server, Azure SQL, Oracle, MySQL, and DB2 iSeries. Developers can **create a connector** to integrate these external data sources in minutes. Using the OutSystems visual editor, developers with any skill set can **query and aggregate data** to extract, load, transform and update data needed for any application.

However, developers aren't limited to manipulating data visually using aggregates. More experienced developers can also use an **SQL Node** to build complex queries or to run stored procedures.

OutSystems continually tracks all of the dependencies to the underlying database. If something changes, your SQL code is automatically refactored, or you're warned if it is no longer compatible with the changes. This extensibility feature accelerates the change process, as well as makes the application itself resilient to change.

Integration extensions for external databases

Many pre-built connectors and utilities that can be used, as is, or modified as needed, can be found on the **OutSystems Forge**, including connectors for on-premises databases that aren't natively supported by OutSystems, such as MongoDB. Big data repositories, including CouchDB and Hadoop, can be integrated using their REST APIs. For other database types, developers can create a connector in **Integration Studio** using **standard code** with ODBC/JDBC drivers.

SAP Integration

OutSystems provides native support for SAP integration, one of the most widely used SoR platforms. Developers can use Service Studio to create mobile and web applications by leveraging data and processes from any SAP instance by using SAP OData services from the **SAP Business Hub**, Service Catalog, or RFC calls (BAPI or ZBAPI). Regardless of the chosen technology (RFC or SAP OData), even developers with limited or no knowledge of SAP Integration can easily integrate both SAP ECC and SAP S/4HANA.

Once a connection has been made, and remote functions have been imported, they can be used in your application the same way you use any other server actions.

Integration extensions for SAP

If developers need to customize a SAP connection, calls to SAP remote functions, or implement different SAP authentication methods, they can create an extension using the **SAP Extensibility API**.

SAP Utilities, an OutSystems-supported extension available from the Forge, includes ready-to-use actions to perform customization of calls to SAP remote functions. Developers can use it, as is, to add advanced functionality to SAP remote functions. or adapt it, as required, to implement common extensibility use cases.

Other extensions that use the SAP Extensibility API are also available for download from OutSystems Forge.

Integration extensions for other SoR platforms

OutSystems can also integrate with other popular enterprise SoR platforms, such as Salesforce, SharePoint and Dynamics 365, using their REST APIs or SOAP methods.

Also, there are multiple connectors available on Forge which can be installed on your OutSystems environment and allow you to connect to hundreds of external systems.

Integration Builder

Integration Builder, currently available through the OutSystems **Early Access Program**, allows developers to easily plug into your CRM or ERP system and quickly access the available data to enhance your OutSystems applications.

Integration Builder removes the complexity of building integrations while allowing developers to quickly link their applications to data in existing systems by offering:

- Easy plug-and-play integration
- Dramatically reduced integration and app development lead times
- The ability for any developer to address complex integration scenarios
- The agility to easily change and augment existing integrations based on business requirements

Currently, Integration Builder allows easy integration with SAP OData services, Salesforce, Microsoft Dataverse (formerly known as CDS), and Dynamics 365, more to come in the future.

End-user authentication integration

OutSystems automatically includes built-in logic for the following end-user authentication methods:

- **Internal** authentication, the default method, stores end user information in the OutSystems database, but not the credentials. Instead, a cryptographic hash function is computed using the credentials—and only its result is stored. When the end user attempts to log in, the hash function is computed again and its result is compared with the database.
- **Active Directory** uses your Active Directory to authenticate end users, is available only in self- managed installations.
- **LDAP**, the Lightweight Directory Access Protocol method, authenticates the end user against your LDAP server, using either LDAP configured for Active Directory or standard LDAP.
- **SAML 2.0**, Security Assertion Markup Language, is used in both traditional web apps and reactive web apps to authenticate end users with Single Sign-On (SSO) provided by commercial identity provider companies, such as OneLogin and PingFederated.
- **IdP Connector** and **IdP Mobile**, community-supported connectors available from the Forge, can be downloaded and modified for use in mobile applications.
- **Azure AD** uses SAML-based authentication to authenticate end users with Single Sign-On (SSO) provided by the Azure AD Identity Provider.
- **Okta** uses SAML-based authentication to authenticate the end users with Single Sign-On (SSO) provided by Okta.

Integration extensions for authentication and identity management

The OutSystems Forge includes community-supported connectors for integration with other authentication services, such as OAuth2, JWT tokens, Basic Auth, API keys. External IdPs for identity management and access control for providers such as, IAM OAuth/SAML, and OpenID Connect can also be found on the Forge.

For integration with other authentication services and IdPs, developers can create a connector in Integration Studio using standard code.

DevOps and CI/CD integration

OutSystems includes built-in tools that unify software development and operations with automation and monitoring to facilitate the implementation of continuous integration and deployment. This allows for shorter development cycles, increased deployment frequency, more dependable releases—all in close alignment with business objectives.

DevOps

OutSystems supports the continuous process of managing applications throughout the entire lifecycle: from development to maintenance.

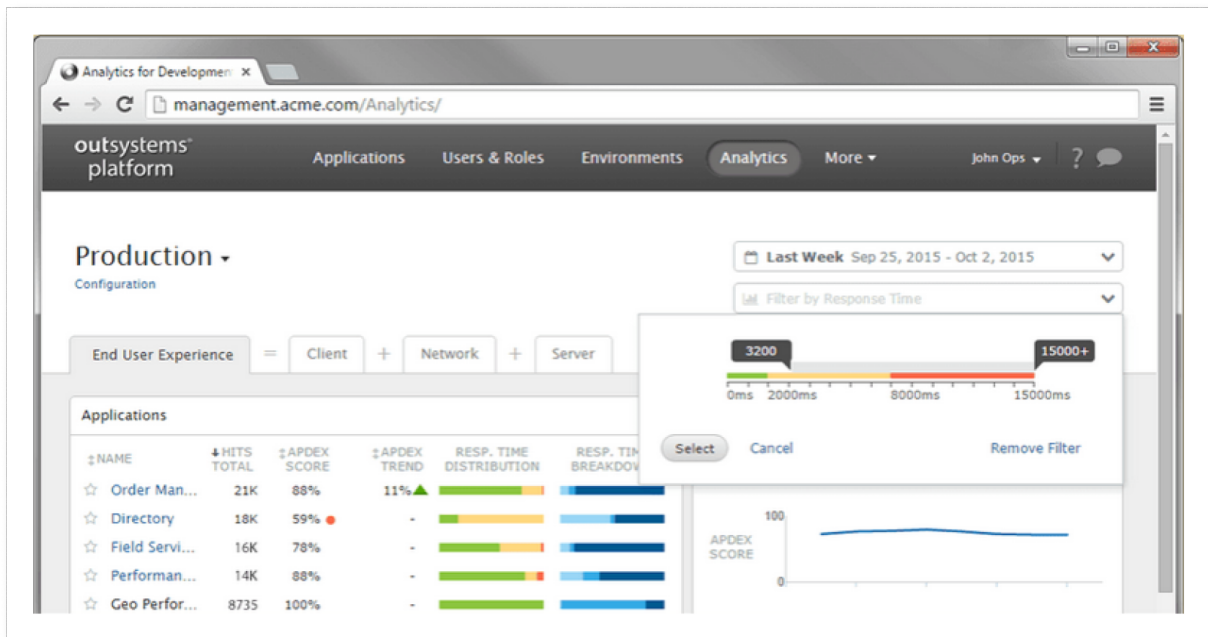
LifeTime, the OutSystems unified console, provides visibility of all environments in your infrastructure, allowing increased control in the following areas:

- IT user and team management
- Security
- Application deployment and configuration
- Performance monitoring
- Hotfix development and deployment

Performance monitoring

Comprehensive analytics and monitoring tools, built into OutSystems, enable proactive management of application performance to make it easier to detect problems by allowing the identification of performance issues in real-time.

The analytics dashboard in LifeTime cover all aspects of the end-user experience, based on industry standard indicators of user satisfaction. Developers can detect, troubleshoot, and solve performance issues even before end users report them.



The **End User Experience** dashboard displays the execution times of all queries, web service invocations, screen renderings, schedulers and invocations to adapters, external systems and custom APIs, which are all tracked asynchronously. Performance reports provide valuable tuning and optimization information for applications.

This dashboard automatically surfaces issues that a team may not be aware of, such as slow screens or extensions. This reduces the time it takes to resolve them, so teams can focus on creating new applications, not troubleshooting them.

The **Client** dashboard gives developers the ability to pinpoint performance issues per web page, and even per screen action.

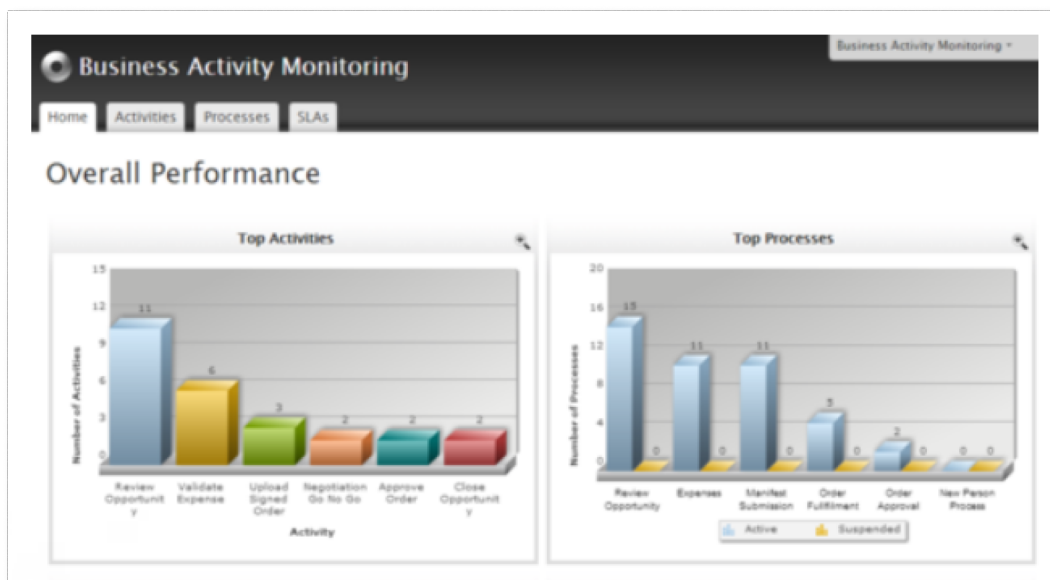
The **Network** dashboard enables teams to quickly drill down on network problems. This is useful for identifying performance issues that only manifest themselves with certain network conditions, so developers can optimize the application accordingly.

The **Server** dashboard enables investigation into server response times by logging and reporting things such as slow queries, integration responsiveness, or custom code invocation time.

Business activity monitoring

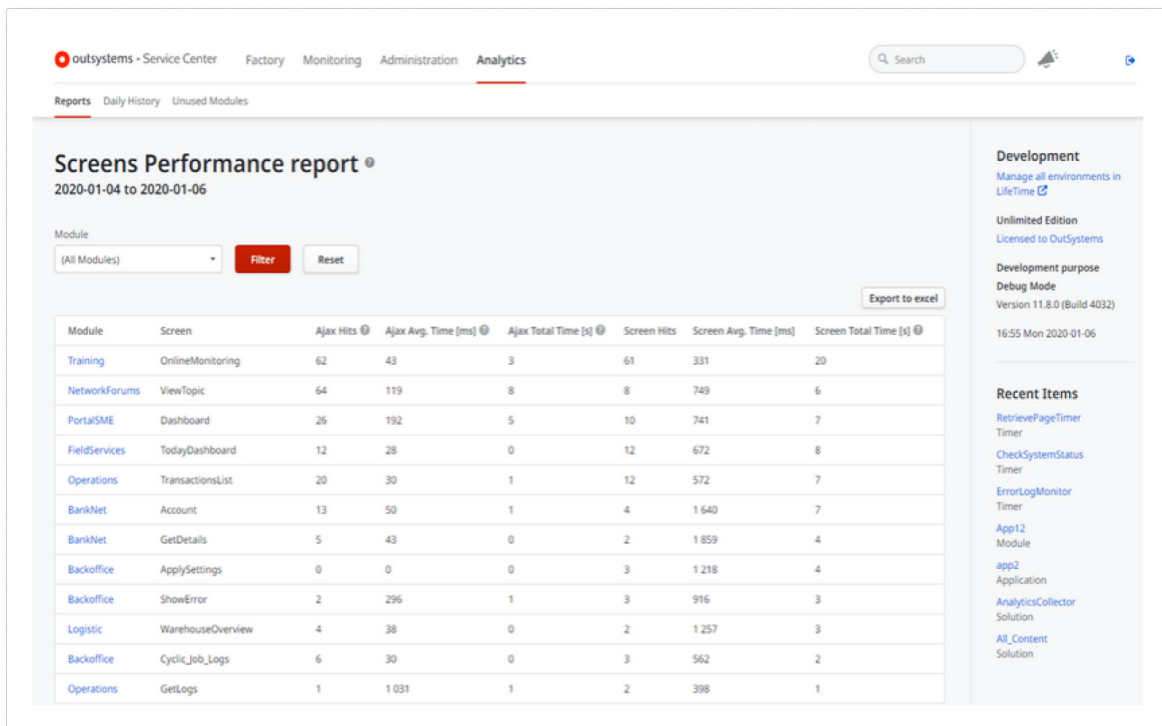
OutSystems supports business process modeling so teams can quickly create a process on top of an existing application or set of applications. This is often used to guide users throughout existing business processes, such as order approvals, claims handling and other actions.

The **Business Activity Monitoring** dashboard, an OutSystems-supported extension from the Forge, enables developers to start monitoring and optimizing their processes right after deploying them for the first time without the need for any development effort. Reports can be easily modified and extended to align with the business language of the organization using them.



Logging

OutSystems logs everything. Applications are automatically instrumented to create over a dozen types of error logs. LifeTime uses this data for its analytics dashboards, and developers have access to all of these logs, in Service Center, without having to write a single line of code.



The screenshot shows the 'Screens Performance report' in the OutSystems Service Center. The report covers the period from 2020-01-04 to 2020-01-06. It includes a table with performance metrics for various modules and screens. The table has columns for Module, Screen, Ajax Hits, Ajax Avg. Time (ms), Ajax Total Time (s), Screen Hits, Screen Avg. Time (ms), and Screen Total Time (s). The data is filtered by 'All Modules' and can be exported to Excel.

Module	Screen	Ajax Hits	Ajax Avg. Time (ms)	Ajax Total Time (s)	Screen Hits	Screen Avg. Time (ms)	Screen Total Time (s)
Training	OnlineMonitoring	62	43	3	61	331	20
NetworkForums	ViewTopic	64	119	8	8	749	6
PortalSME	Dashboard	26	192	5	10	741	7
FieldServices	TodayDashboard	12	28	0	12	672	8
Operations	TransactionsList	20	30	1	12	572	7
BankNet	Account	13	50	1	4	1 640	7
BankNet	GetDetails	5	43	0	2	1 859	4
Backoffice	ApplySettings	0	0	0	3	1 218	4
Backoffice	ShowError	2	296	1	3	916	3
Logistic	WarehouseOverview	4	38	0	2	1 257	3
Backoffice	Cyclic_Job_Logs	6	30	0	3	562	2
Operations	GetLogs	1	1 031	1	2	398	1

These metrics are captured asynchronously at run time, so application performance isn't compromised while creating a constant stream of metrics. Events captured include:

- Errors in all application layers
- Batch processing executions
- Web services invocations, both inbound and outbound
- Adapters, integration and custom API invocations
- Page accesses

- Database accesses
- External database accesses
- Client side screen access times

These logs are stored in a standard database that users can access to build their own dashboards.

Analytics APIs

All the insights from the analytics capability of OutSystems are publicly exposed in a flexible, easy-to-use, REST API that allows access to all registered events in real-time for seamless integration with other enterprise management solutions, and it also enables applications to register custom events.

For instance, the API can be used to extend the dashboard with business data, such as adoption monitoring per department. It's also possible to add a device's GPS information to the events sent by applications to OutSystems for an exact understanding of where customers or employees were when they performed a specific action.

Deployment

Lifetime provides automated deployment processes across all environments, from development to production, which simplifies processes and increases overall IT efficiency.

If a problem pops up in a deployed app, it can be debugged in a pre-production environment. When the hotfix has been tested it can then be deployed to production, with no downtime, and then propagated backwards to previous environments.

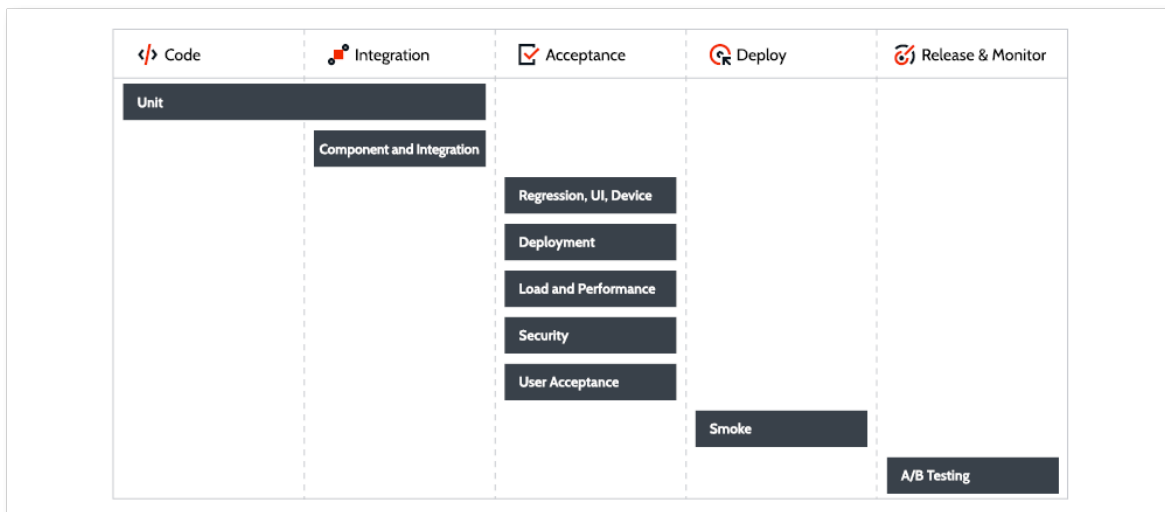
DevOps extensions

OutSystems built-in monitoring and logging features can be complimented by such third party platforms as Zabbix and HP OpenView.

CI/CD

Thorough testing is key for any type of software development, and reliable automated testing is crucial for the continual integration/continual deployment paradigm, which requires many tests, frequently performed, all along the CI/CD pipeline.

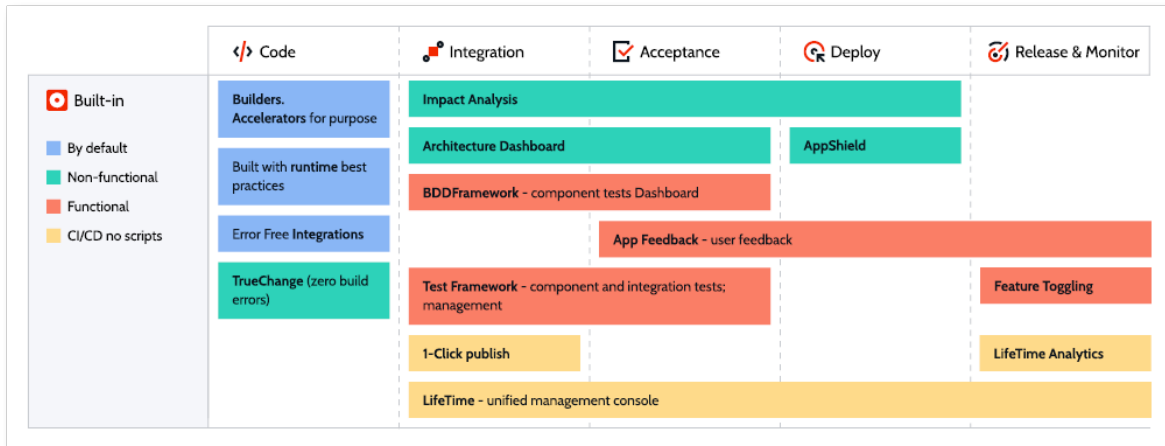
Every enterprise must find a testing solution for each stage of development, as seen in the figure below. Typical stages and testing types:



OutSystems supports a modern development approach that automatically performs validations regarding factor analysis and impact analysis, alerting developers about conflicts with existing code before new features can be merged. When staging an application to the next environment OutSystems provides feedback about missing dependencies that might cause runtime issues after deployment.

All this is provided automatically, with no need for any scripting.

OutSystems built-in automated testing:



Aside from handling many aspects of automatic testing that are required for building a continuous development pipeline, OutSystems's built-in tools and technologies facilitate easy integration with major third-party testing platforms so you can continue to work the way you are used to, or choose a tool that answers your needs, all orchestrated within the OutSystems environment.

OutSystems automated code testing

In traditional development, problematic code can only be detected during the build and compile stage. OutSystems TrueChange technology, which combines automation, AI, and analytics, provides developers with real-time feedback about every change made to the application. If errors are found in architecture, logic, or dependencies, the developer is presented with a list of bugs, each with a suggested fix and a clickable link to open the problematic component. The app can be built and published only after the correction has been made.

OutSystems automated integration testing

While traditional development models tend to test code and integration separately, they are closely linked, so OutSystems tests code and integration as a unit, checking both the validity of component functionality and its correct integration with the rest of the app. This ensures that new features do what they are supposed to do and integrate as expected. And all this is done automatically, in the background.

For example, once a database has been integrated into your app there is no further need to check if the code is working correctly. OutSystems developers can concentrate on the functional level of the app, making sure that it represents the business logic, rather than have to dig into technical details of the code.

The most powerful integration test is **1-Click Publish**, which validates all of the code, does factor analysis, merges the code, and starts the deployment of your module in your environment. However, if any errors are detected, this operation is unavailable. Instead, errors are listed in the TrueChange window, and only once they are corrected can the app be built.

Built-in impact analysis is done for integration, acceptance, and deployment, simplifying the process of deploying complex application portfolios across environments.

The **Architecture Dashboard** monitors technical debt from integration through acceptance, visualizing complex cross-portfolio architectures to identify problems while also helping developers follow best practices and avoid common pitfalls.

LifeTime, the OutSystems centralized management console, manages application environments, development, quality, production environments and version management from integration all the way through deployment and release.

Test Framework, for component and integration tests and management.

The **BDD Testing Framework**, an OutSystems-supported component available through the Forge, facilitates test automation using **Gherkin** syntax, a human-readable language for structuring and describing an app's expected behaviors in order to perform component tests, including:

- Unit level tests to map lower-level actions, like core concepts for your application
- Service level tests to map higher level actions, where the action itself can either represent or use an API call within your application
- End-to-end level tests to map higher-level end-to-end actions, where the action itself can have a tree of dependencies, like a complex business action of your application

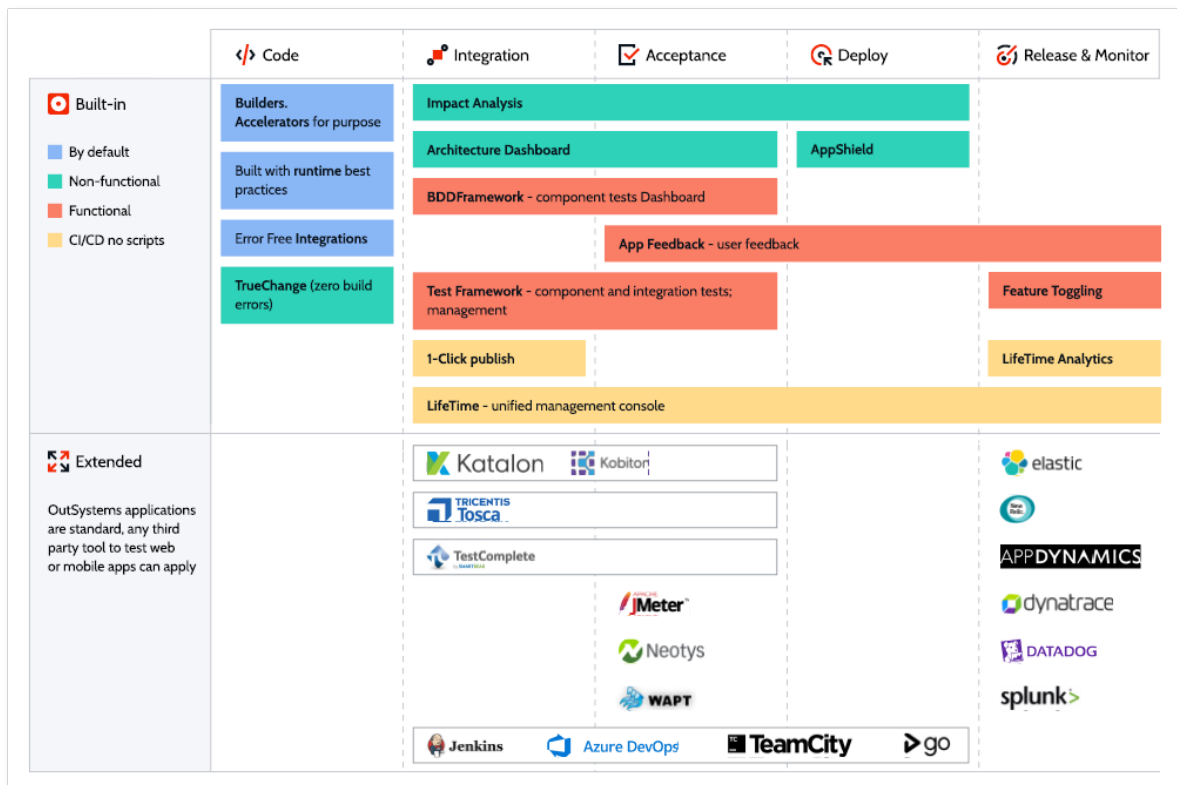
Automated deployment

OutSystems provides deployment without any need for scripting. Every time an application is staged to another environment an impact analysis gives feedback about problems that might lead to runtime issues when the app is deployed, such as missing dependencies.

OutSystems AppShield, an additional cost add-on, automatically adds additional layers of security during deployment to make mobile applications more resistant to intrusion, tampering, and reverse engineering. AppShield makes it more difficult for attackers to spoof your app, meddle with its security controls, inject malicious code, and run in an insecure environment.

CI/CD Extensions

The figure below shows the built-in test automation provided by OutSystems and the many third- party testing tools that can be integrated into the CI/CD workflow.



This [article](#) has a general discussion about how to integrate popular third-party automation tools into your OutSystems CI/CD workflow, including:

- **Jenkins**, a CI/CD automation tool for monitoring the execution of repeated tasks. It is an open- source software that you can use to run with any operating system, and it offers several built-in plugins for building CI/CD pipelines. With Jenkins CI/CD server, you can automate the various stages of your delivery pipeline.
- **Azure DevOps**, a CI/CD automation tool that covers the entire application lifecycle, providing version control, reporting, requirements management,

project management, automated builds, testing, and release management capabilities.

- **Selenium**, a portable software testing framework for web applications that provides a record/ playback tool for authoring tests without learning a test scripting language. It also provides a test domain-specific language (Selenese) to write tests in many popular programming languages, including Java, C#, Groovy, Perl, PHP, Python, and Ruby.
- **Applitools**, software designed to test and monitor your app's UI as part of your CI/CD process. It monitors critical functional and visual aspects of any web, mobile, and native app in a fully automated way.
- **Browserstack**, a cloud web and mobile testing platform that enables developers to test their websites and mobile applications across on-demand browsers, operating systems and real mobile devices, without requiring users to install or maintain an internal lab of virtual machines, devices or emulators.
- **Appium**, an open-source automation tool, where you can run scripts and test native, mobile, web, and hybrid applications on Android or iOS using a web driver.
- **Tricentis Tosca** used to automate end-to-end testing for applications. It combines multiple aspects of software testing, such as test case design, test automation, test data design and generation, and analytics to test GUIs and APIs.
- **SauceLabs**, allows users to run tests in the cloud, providing a comprehensive test infrastructure for automated and manual testing of desktop and mobile applications using Selenium, Appium, and JavaScript unit testing frameworks. It also provides a secure testing protocol, Sauce Connect, for testing applications behind customer firewalls.
- **ElasticSearch**, for monitoring and analytics, ElasticSearch is the most popular enterprise search engine that provides a distributed, multitenant-

capable full-text search engine with an HTTP web interface and schema-free JSON documents.

OutSystems pipeline accelerator

With the [outsystems-pipeline](#), an open-source accelerator developed on top of real world experience, your developers can extend the built-in capabilities of LifeTime, with such features as governance over which applications and environments you are authorized to manipulate or the built-in impact analysis that simplifies correctly deploying complex application portfolios across environments.

This accelerator includes Python scripts built on top of the LifeTime APIs, pipeline templates, and documentation to simplify integration with any CI/CD platform, including [Jenkins](#) and [Azure DevOps](#), to build an OutSystems pipeline.

Your developers can also view and modify [example scripts](#) to help bootstrap custom pipelines.

AI integration

Developers and architects can use OutSystems to create new products and services enabled by AI to help reduce costs, increase efficiency, and even find new sources of revenue.

OutSystems users and developers can enhance applications with three main aspects of AI:

- **Conversational experiences** to improve engagement in applications by allowing users to interact in their own natural language. These experiences are part of multi-channel chat and voice interfaces and in-device voice assistant integration.
- **Cognitive services** to use intelligent algorithms that see, hear, speak, understand, and interpret language and voice. These cognitive services operate on unstructured data to provide a solution to business problems that regular development algorithms can't solve.
- **Custom machine learning** use existing data to provide predictive capabilities, automating decisions in business processes, or improving application logic. Unlike cognitive services, which rely on prebuilt models for predefined use cases, custom machine learning models need to be trained and tested with business data.

With pre-built AI components for common use cases and connectors available from the Forge, OutSystems makes it easy for developers to add AI to their apps.

AI service connectors

Through OutSystems Forge, you can access connectors to multiple cognitive services that reduce the time and effort required to build chatbots, add language analysis to apps, and weave decision optimization, predictive intelligence, and machine learning and AI models throughout your final products. Among the available OutSystems-supported connectors are:

- **Azure QnA Maker Connector** to a cloud-based API service that answers natural language questions by matching them with the best possible answer from an organization's knowledge base for a virtual assistant or a chatbot, among others
- **Azure LUIS Connector** to a cloud-based API service using the cloud-based Azure Language Understanding (LUIS) to apply custom machine-learning intelligence to a user's conversational, natural language text to predict overall meaning and pull out relevant, detailed information
- **Azure Cognitive Services Connector** to enable users to have natural and contextual interactions with applications using Azure Cognitive Services
- **Azure Bot Framework Connector** to build bots that interact with users on websites, apps, Cortana, Microsoft Teams, Skype, Slack, and Facebook Messenger
- **Dialogflow Connector** to the Google Cloud platform to build voice and text-based conversational apps for user interaction for chatbots and IoT devices, among others
- **Azure ML Connector** to enables communication with an Azure Machine Learning Score Model in real time to return prediction results obtained from predictive experiments created with the tools provided by Microsoft Azure Machine Learning.

Chatbots and voice-enabled apps

OutSystems offers chatbot components and language analysis for building and adding conversational experiences to applications so they not only interact with users naturally.

Chatbots

- **OutSystems.AI Chatbot**, an OutSystems-supported component available from the Forge, can be used by developers to build, connect, deploy, and manage intelligent bots that interact naturally with users, making it much easier to implement rich conversational experiences in an app. OutSystems.AI uses Azure services to implement common patterns for chatbots, and includes a chatbot interface, a webhook template that allows quicker customization of the bot response logic, interactive cards for adding clickable thumbnail images, attachments, and buttons to automatic responses, and integration with text and speech analysis.
- **Azure Bot Framework Connector**, an OutSystems-supported connector available from the Forge, enables easy connections to the Microsoft LUIS Connector, Azure QnA Maker, and the **Azure Bot Framework**, allowing developers to use these capabilities in the OutSystems IDE and using the OutSystems visual language.
- **Dialogflow**, an OutSystems-supported connector available from the Forge, is used to build engaging voice and text-based conversational interfaces, such as voice apps and chatbots. It can be integrated with multiple channels to increase interactions and reach more customers. Some of the available channels are Google Assistant, Amazon Alexa, Facebook Messenger, and other popular platforms and devices.

Virtual private assistants

OutSystems provides developers with tools to integrate the following VPAs into their applications:

- **Siri**, Apple's virtual assistant, can be integrated into OutSystems apps using the community- supported **Siri Shortcuts** plug-in, available from the Forge, to develop seamless interaction with Apple devices and their built-in applications.
- **Google Assistant** uses a user's voice to trigger and complete activities. **Dialogflow**, an OutSystems-supported connector available from the Forge, can be used to build Google-based voice interaction.
- **Alexa**, Amazon's cloud-based voice service, can be integrated into OutSystems apps using the community-supported **Alexa** component, available from the Forge, to build natural voice experiences and intuitive interactions with the Alexa-equipped devices.
- **Cortana**, Microsoft's digital assistant, can be integrated into OutSystems apps using the OutSystems-supported **Azure Bot Framework**, available from the Forge, to build a speech bot, register it, and define your Cortana channel.

Cognitive services to create AI-enabled apps

Connectors from the OutSystems Forge enable developers and users access to cognitive services that they can add to their applications to solve business problems from unstructured data.

Language Services

Language services are used for speech recognition, intent recognition, transcription, synthesization, and content and sentiment analysis. These

connectors and components are available from OutSystems Forge for accessing language analysis:

- **OutSystems.AI Language Analysis**, an OutSystems-supported connector available from the Forge, is a set of pre-built services that developers can add to applications to determine customer sentiment, needs, and issues, and to accelerate business processes by receiving text as an input parameter for sentiment analysis, key phrase extraction, entities detection, translation, language detection, spell check, and speech-to-text conversion.
- **Azure Cognitive Services Connector**, an OutSystems-supported connector available from the Forge, is used to access services for speech and text from the OutSystems IDE and add them to their apps, including speech transcription, custom speech service, speaker verification, speaker identification, and text to speech conversion.
- **IBM Watson Services**, an OutSystems-supported connector available from the Forge, converts spoken audio into text or written text into natural sounding audio, enabling the use of voice for verification, or adding speaker recognition to an app, all in a variety of languages and voices.
- **OutSystems.AI Document Processor**, a community-supported component available from the Forge, is used to quickly capture and qualify document data, identify and classify important account-related documents, and transform forms into usable data, including form recognition to identify and extract text, key/value pairs and table data from form documents, receipt analyzer to identify and extract key information from sales receipts (using the **Azure Cognitive Services Connector**), and layout analysis, (using **this service** (link to <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/form-recognizer/>)) to extract text and table structure using high-definition optical character recognition (OCR).

Vision Services

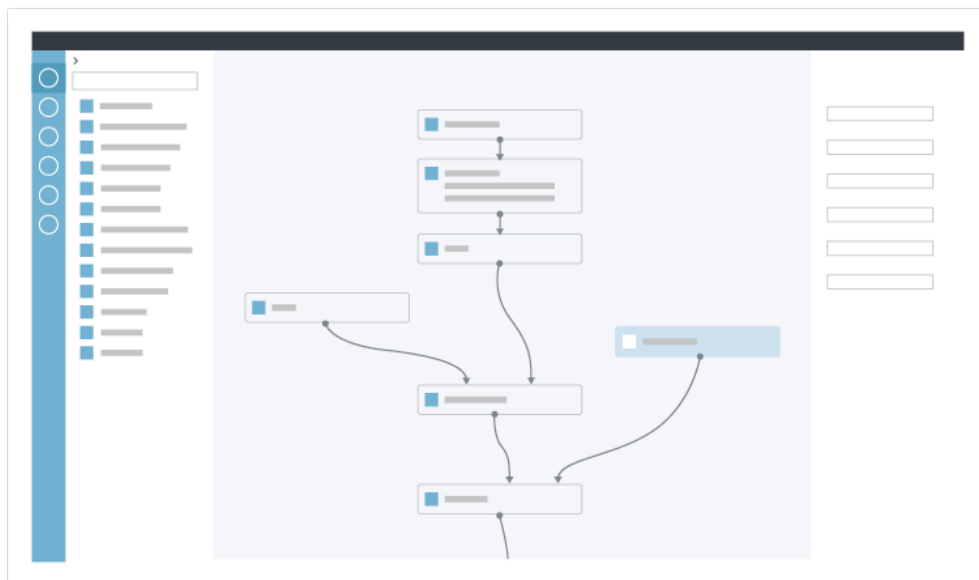
Vision services include image-processing algorithms that intelligently identify, caption, index, and moderate pictures and videos. With the help of the following Forge components and connectors, developers can make it possible to understand the content of an image, classify it, detect individual objects and faces within images, and read printed words within the images (OCR):

- **Azure Cognitive Services Connector**, an OutSystems-supported component available from the Forge, can be used to add Microsoft Azure Vision Services into the OutSystems IDE for image analysis, recognition of handwritten or printed text in image, recognition of celebrities and landmarks, thumbnail generation, face verification, face detection, emotion recognition, and custom model for image recognition.
- **Google Cloud Vision OCR**, a community-supported component available from the Forge, enables developers to use Google Cloud Vision in the OutSystems IDE to extract text, identify text in an image, and get specific data types in image.
- **IBM Watson Services**, a community-supported component available from the Forge, enables developers to use IBM Watson Visual Recognition Services for facial analysis and image recognition.
- **Amazon Rekognition Face Matching**, a community-supported component available from the Forge, enables developers to use AWS Recognition capabilities in the OutSystems IDE for face matching and adding faces to collections.

Custom machine learning

Custom machine learning automates decisions in businesses process by providing predictive capabilities or improving an application's logic on data that already exists in your system. Unlike the cognitive services that are more oriented towards pre-built models for predefined use cases, custom machine learning models must be trained and tested with actual specific business data. So, it is necessary to label and prepare your data, choose an algorithm, train the algorithm, tune it, and optimize it for deployment. Currently, there are solutions for creating machine learning models that require a minimum data science skills with a visual development language. One of these solutions is Azure ML. You can use OutSystems to integrate and leverage the application with Azure ML technology.

Azure ML, an OutSystems-supported connector available from the Forge, can be used to create an interactive, visual workspace easily and quickly build, test, and deploy models using pre-built machine learning algorithms. No programming is required; the machine learning model is constructed by connecting datasets and analysis modules on an interactive canvas, and then deploying it.



Azure Machine Learning Studio publishes models as web services that can easily be consumed by OutSystems applications.

Machine Learning Builder

Machine Learning (ML) Builder, currently available through the OutSystems **Early Access Program**, is a cloud tool to help create apps with predictive capabilities using its built-in machine learning features. ML Builder requires only basic knowledge of machine learning concepts. The user interface guides developers through the process of connecting to your data and training a model. Once a model is ready, ML Builder exposes server actions to:

- Build the logic of an app
- Create back-ends for administrators
- Design sophisticated UI
- Expose the prediction capabilities through a REST API

ML Builder currently requires Microsoft Azure as the AI provider. You need an Azure subscription and sufficient permissions to create and manage the Azure resources. Note that you need to manage the costs of your Azure subscription, as it isn't included into the pricing.

Conclusion

OutSystems provides easy integration with web systems, APIs, popular SoR platforms, end-user authentication, and AI services—to name just a few. Some of these integrations are built-in: drag and drop them into place. Others require a little assembly: locate a Forge extension, modify it for your needs, and then use it everywhere in your development environment.

Even when the integration requires some classic coding, once an extension is created it too becomes just another native OutSystems element.

