workrunner

Workrunner Platform Technical Document

1. Synopsis

Workrunner is a scalable, enterprise grade software package for developing process-based solutions with ease. It enables medium- and large-scale organizations to design and deploy unlimited processes that seamlessly interconnect people and systems.

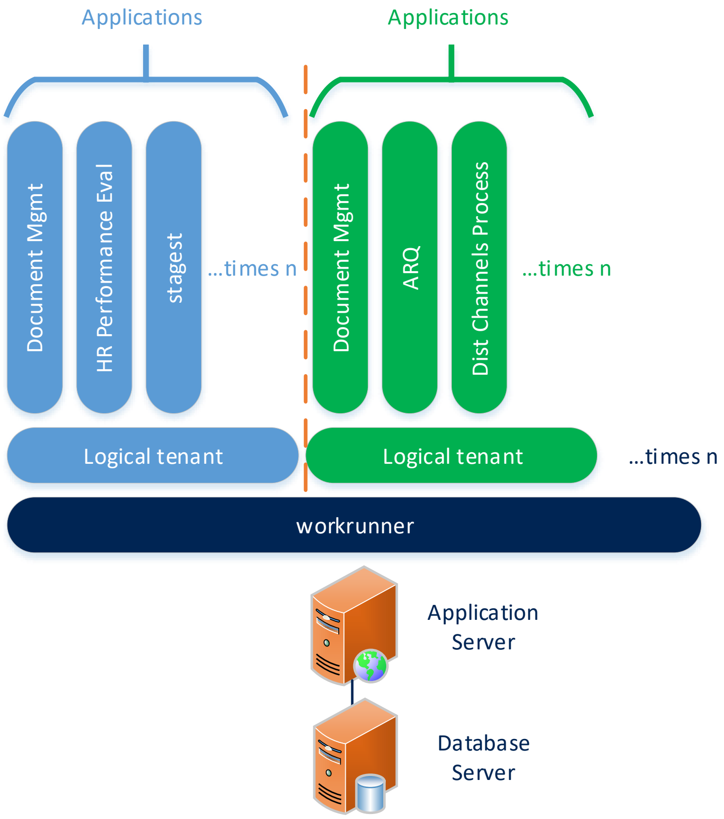
Workrunner is designed to provide a single comprehensive web interface that encompasses all facets of a process lifecycle – analysis, modeling, simulation and continuous enhancement – and to eliminate the need for people to learn and use multiple different systems and applications. With a flexible architecture, it allows for rapid and frequent modifications to active processes and makes it easy to migrate older applications into service-oriented structures.

Low-code Application Development Platform

Workrunner is a wide scale application development platform that can realize all workflows and business processes of an enterprise from the simplest as document approval to the most complex as production processes. Easily integrable with various databases and systems, **work**runner enables you to quickly meet your application development needs.

Multi-Tenant Architecture

Workrunner can host multiple tenants which are completely isolated from one another, for use multiple companies or teams with ensuring data security.

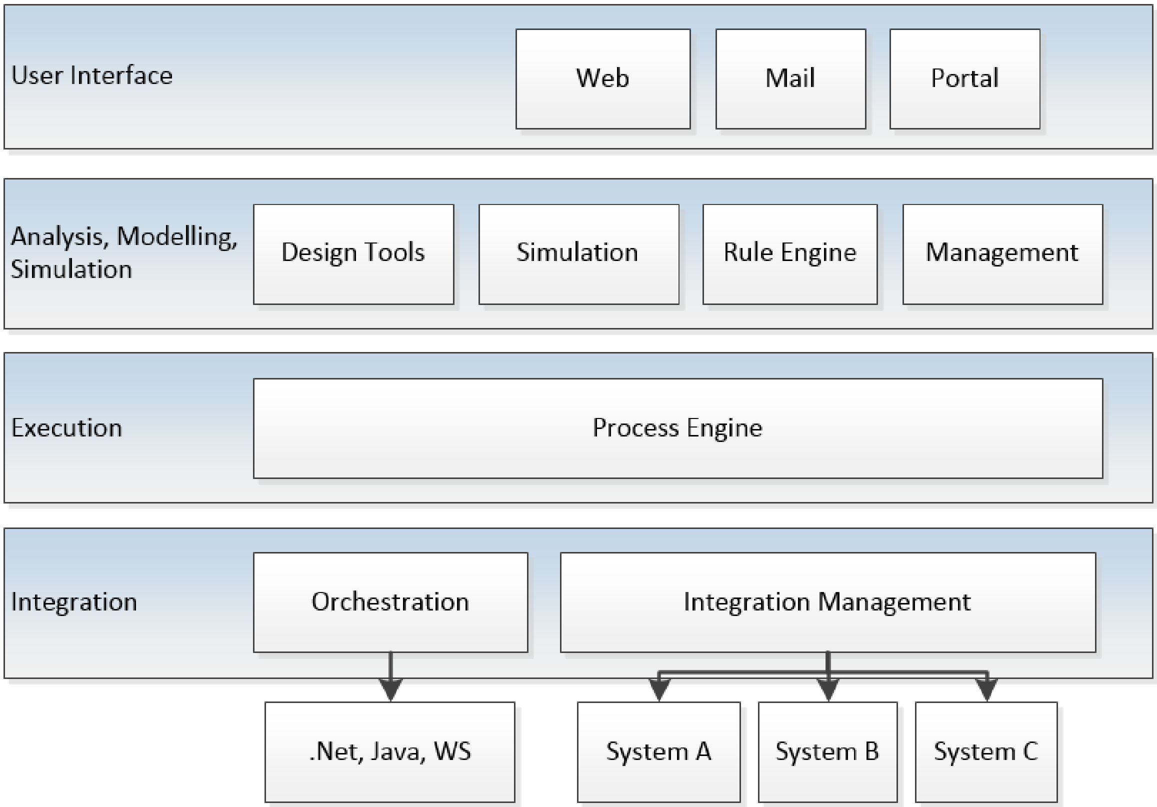


Business Process Management (BPM) Engine

At the heart of **work**runner lies a fault-tolerant and scalable Process Engine that manages and executes all processes. This engine acts as the central point of communication for every player in a designed process, be they people, applications or database systems. The engine evaluates active processes, manages authentication and authorization, and performs tasks such as dispatching notifications, assigning workflow jobs and coordinating integration endpoints.

Complex business processes usually involve the integration of several different applications, databases and legacy components. By coordinating all these subsystems, **work**runner hides the underlying complexity and makes it easy to monitor, audit and control the workflows.

2. Architecture



User Interface layer provides the user experience. It consists of a rich and user-friendly HTML5 interface that supports all modern browsers, including mobile devices.

Analysis, Modeling and Simulation layer provides the process design, change management and simulation features. Processes are designed and deployed entirely via the web-based **work**runner process designer tools. All processes can be version controlled and all business rules can be simulated on a test environment prior to deployment.

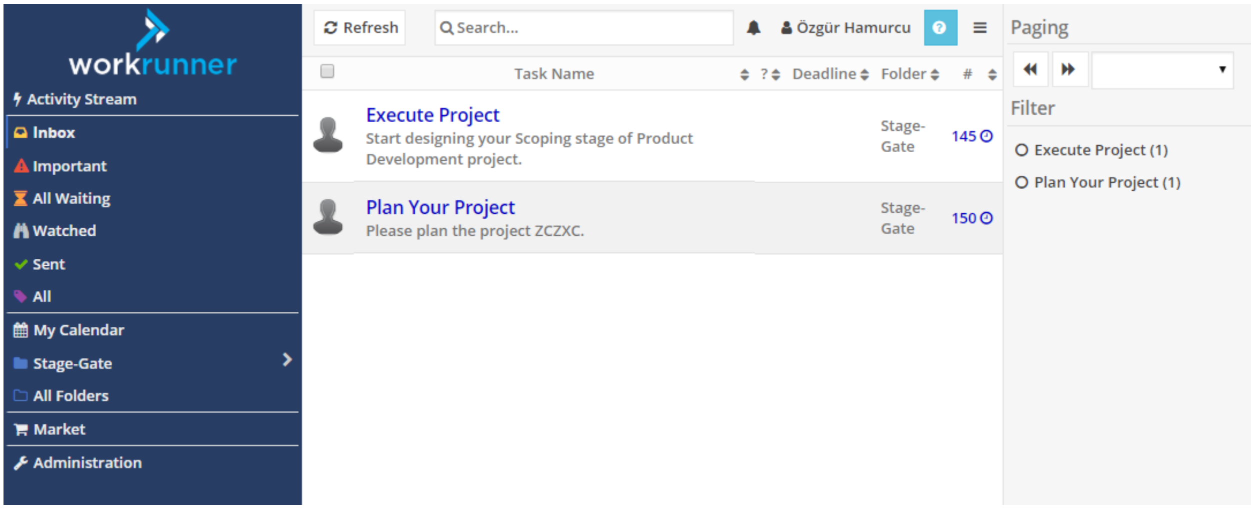
Execution layer performs the actual execution of business processes. Process Engine evaluates all activities and carries out all operations involved in workflows. Every single activity is recorded for subsequent auditing, reporting and simulation.

Integration layer coordinates all tasks requiring integration with other systems. Process Engine communicates with external databases and web services over this layer.

2.1. User Interface

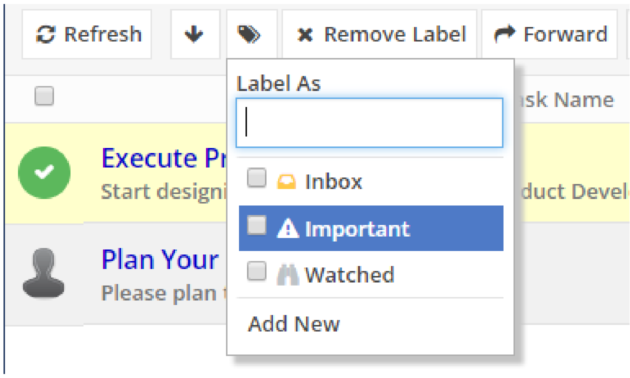
Workrunner boasts a completely browser-based interface, including all end user screens and designer tools. The interface supports all modern browsers (IE, Firefox, Chrome, Opera etc.) and is optimized for all mobile devices with JavaScript capabilities.

2.1.1. Workitem List

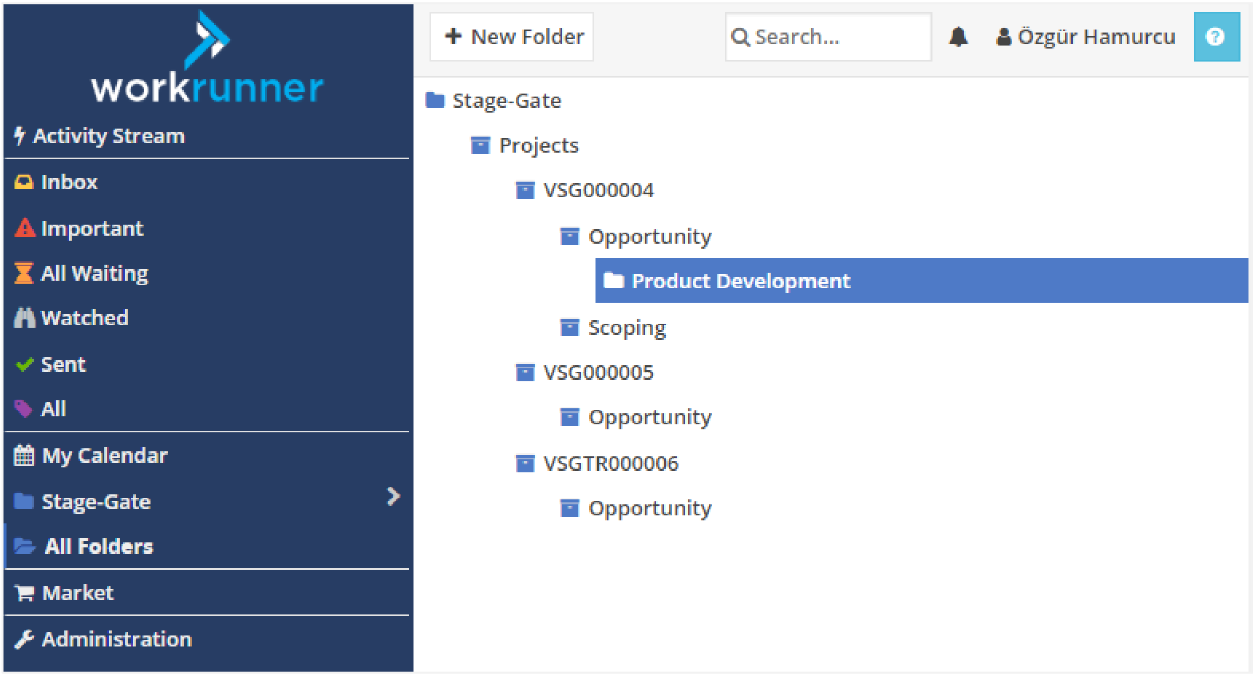


Task Name	?	Deadline	Folder	#
Execute Project Start designing your Scoping stage of Product Development project.			Stage-Gate	145
Plan Your Project Please plan the project ZCZXC.			Stage-Gate	150

Jobs assigned to the user are displayed similar to simple e-mail messages and the user can easily categorize incoming jobs under custom tags.

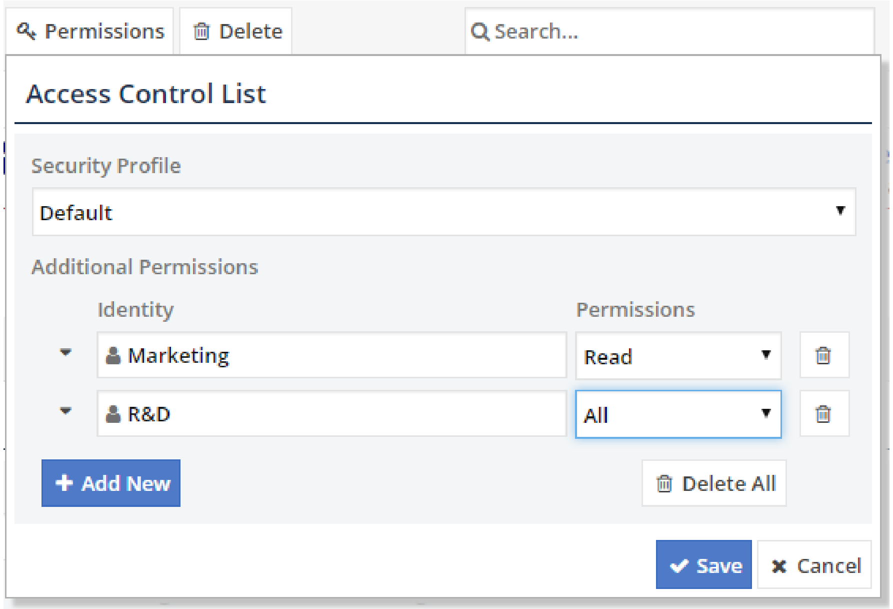


2.1.2. Folders



Workrunner processes are categorized into folders to enable quick access. Every folder has a dashboard view that allows a quick glance at the general report data for the contained processes.

Individual folders have access control lists that allow fine control over who can access the processes within.



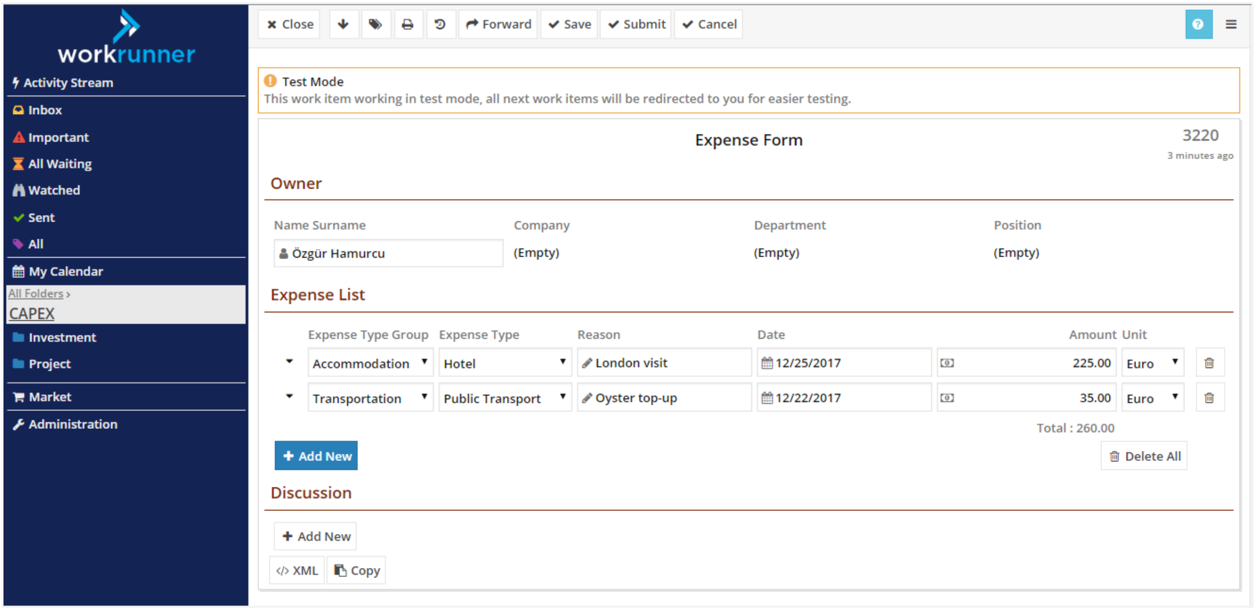
Security Profile: Default

Additional Permissions

Identity	Permissions
Marketing	Read
R&D	All

2.1.3. Forms

Workrunner uses modern HTML5 technologies in all user interfaces. Since designed to allow ease of use, there are no complex or commotion-causing processes. Form fields within user interfaces are only displayed when necessary, comprising intelligent mechanisms that direct users with no training requirements.



Test Mode
This work item working in test mode, all next work items will be redirected to you for easier testing.

Expense Form

3220
3 minutes ago

Owner

Name Surname	Company	Department	Position
Özgür Hamurcu	(Empty)	(Empty)	(Empty)

Expense List

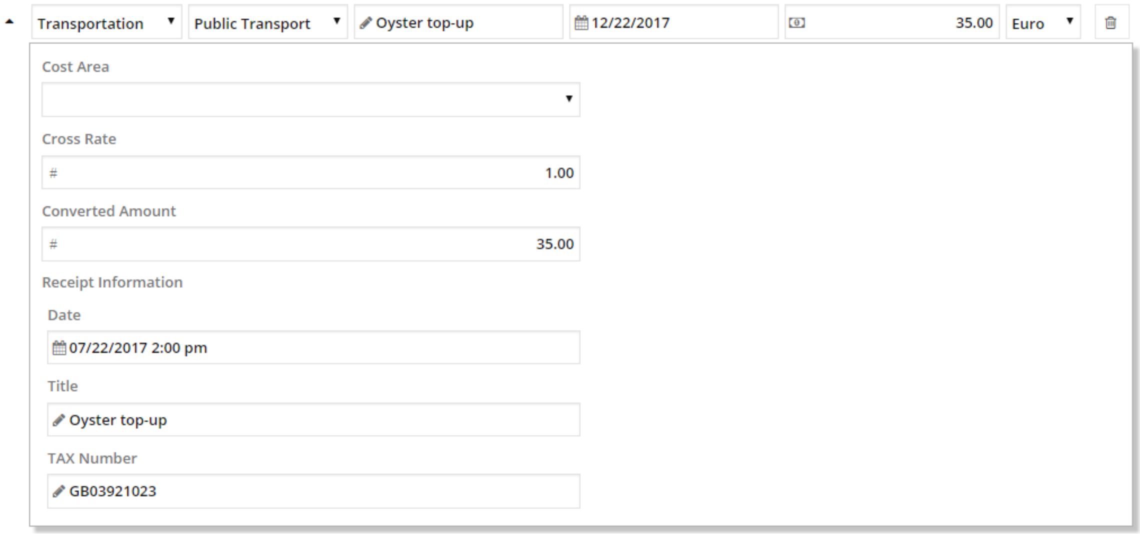
Expense Type Group	Expense Type	Reason	Date	Amount	Unit
Accommodation	Hotel	London visit	12/25/2017	225.00	Euro
Transportation	Public Transport	Oyster top-up	12/22/2017	35.00	Euro

Total : 260.00

Discussion

Workrunner forms are powered by a sophisticated HTML5 & Ajax interface engine that allows response performance to levels of milliseconds. Forms can be designed to contain subsections that are enabled or hidden according to custom conditions. Database-backed lookup lists allow for rapid and correct data entry.

In addition to simple controls such as text, numerical, date, selection boxes, it may be used for more complex controls such as tables or multiple-record entries with master/detail relationships. It also features flexibility to sections that are not used in multiple-record entries.



Transportation	Public Transport	Oyster top-up	12/22/2017	35.00	Euro

Cost Area

Cross Rate
1.00

Converted Amount
35.00

Receipt Information

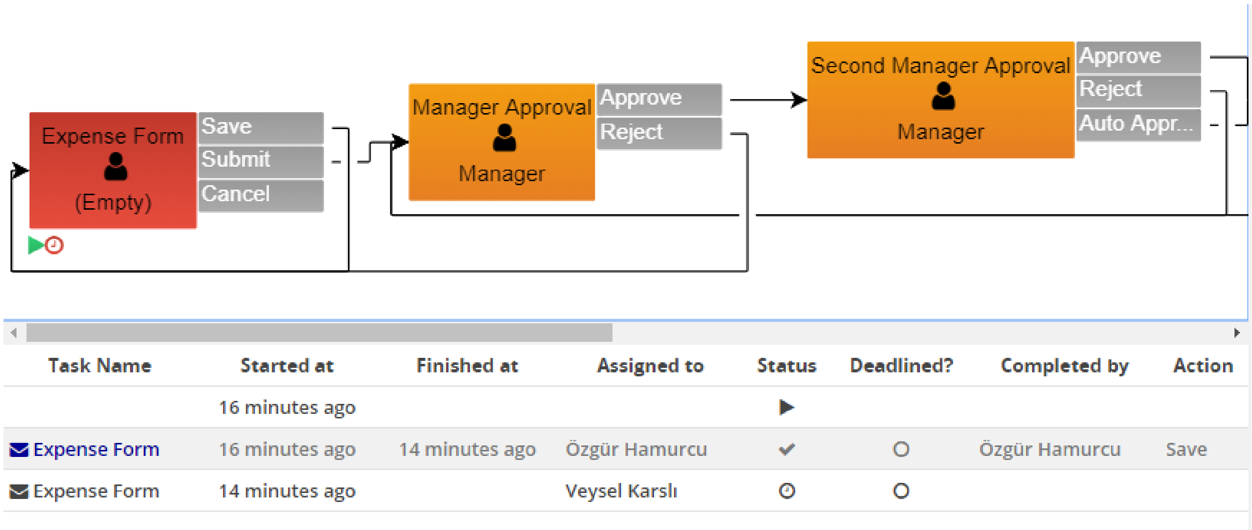
Date
07/22/2017 2:00 pm

Title
Oyster top-up

TAX Number
GB03921023

2.1.4. History

workrunner keeps all records of transactions during the execution of processes. Within History, all details, such as who started work, when the next steps are started and finished, what is being done, who owns the current step and so on, are stored as records.



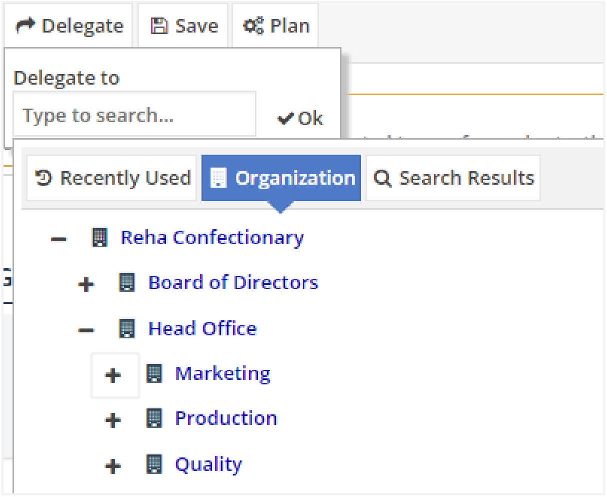
Task Name	Started at	Finished at	Assigned to	Status	Deadlined?	Completed by	Action
	16 minutes ago			▶			
✉ Expense Form	16 minutes ago	14 minutes ago	Özgür Hamurcu	✓	○	Özgür Hamurcu	Save
✉ Expense Form	14 minutes ago		Veysel Karslı	⊙	○		

History information can be displayed as both in a graphical layout based on a process map and in a classic table order.

Since history information contains process times, it enables analysis of areas in which process improvements can be identified after a certain point of use.

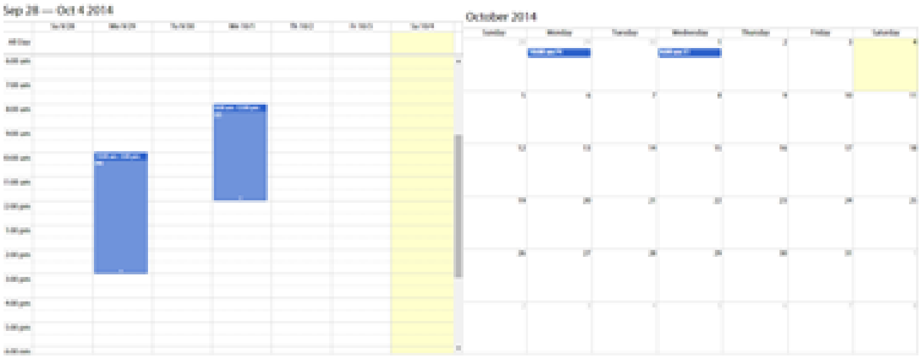
2.1.5. Delegation

Singular delegation is featured, when and if needed, to all users, excluding the shared work lists. This enables users to share the job workload of their assigned tasks. In addition, delegated tasks can be tracked from the history.



2.1.6. *Business Calendar*

workrunner has a built-in business calendar to provide a complete working environment. Via the calendar, users can manage their work days, share their work lists and temporarily delegate their tasks to other users for out-of-office periods or days off.



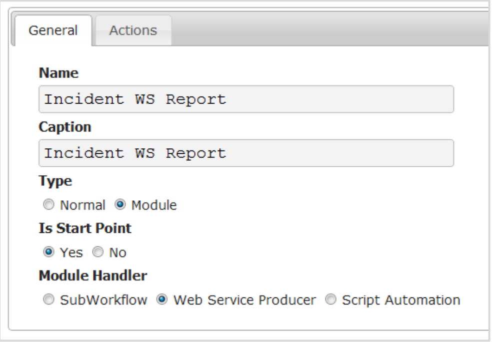
User calendars can also be accessed from within processes. For example, it is possible to select a suitable work day directly from the calendar for an appointment process. The calendar can be integrated with any external system supporting the iCAL standard.

Default office hours setting of the calendar is 09:00-18:00 and can be customizes by system administrators. For example, rules such as “each year on October, 29th” or “every 2nd week of August” can be defined.

Calculations such as due dates within the system are executed in accordance with the working hours. For example, if a task sent on a Friday at 15:00 has a due period of 6 hours, then the user will be assigned to deliver the task by Monday at 12:00 as a due date, and not Friday at 21:00, which falls out of working hours. These type of calculations can be customized by using different process based calendars.

2.2. Integration – Web Services

Workrunner, as a common part of business processes, has support for consuming SOA based web services. In addition, any task within a process can be exposed to other systems as web services. This feature is built upon Microsoft Communication Framework, which means a large number of web service standards such as WS-Transactions are supported out-of-the-box.



Name

Incident WS Report

Caption

Incident WS Report

Type

○ Normal ● Module

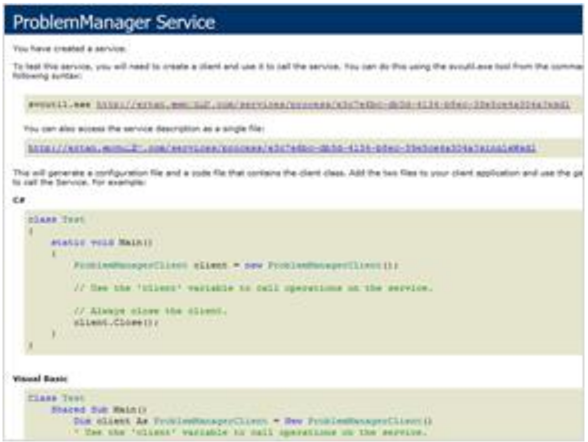
Is Start Point

● Yes ○ No

Module Handler

○ SubWorkflow ● Web Service Producer ○ Script Automation

Depending on the type of tasks, web services can also be used to initiate new processes or to finalize pending jobs for workflows in progress. Service-enabled tasks can be exposed to external systems as UDDI-based web services.



ProblemManager Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the comma
following syntax:

```
svcutil.exe [illegible]
```

You can also access the service description as a single file:

```
[illegible]
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the ge
to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        ProblemManagerClient client = new ProblemManagerClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

```
Class Test
    Shared Sub Main()
        Dim client As ProblemManagerClient = New ProblemManagerClient()
        ' Use the 'client' variable to call operations on the service.
```

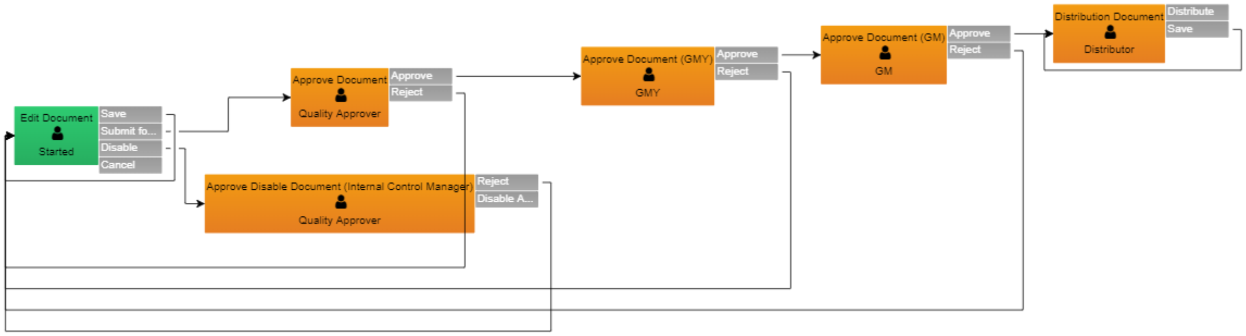
Workrunner not only allows processes to speak to one another by database, but also allows communication with each other via web services.

2.3. Process Modelling

Process Modeling in Workrunner is entirely carried out using the process designer tool provided within the same web interface. The designer tool allows the management of all the definitions involved with a process, including forms, databases and web services. Any process design can be exported as a package to be migrated into another system.

In Workrunner terminology, processes are made up of 5 major building blocks: Pools, Tasks, Actions, Routes and Roles. The Process Map provides an overhead view of the process, clearly displaying all Tasks assigned to Roles and all steps defined as automatic jobs. The resulting map is a much more straightforward and standard view of the process than one that would be produced by conventional diagramming applications.

A Workrunner process definition is a single self-contained XML file. This file can be duplicated or migrated to other servers by a simple copy-paste operation.



In order to achieve one of the primary goals of BPM applications, namely, to provide Business Process Agility, all the necessary database structure is generated automatically when an exported process definition is introduced into another server. In the cases where the database objects already exist from an older version, the whole structure is re-organized to fit the new design, without any data loss. None of these automated tasks require a restart, reboot or recompile.

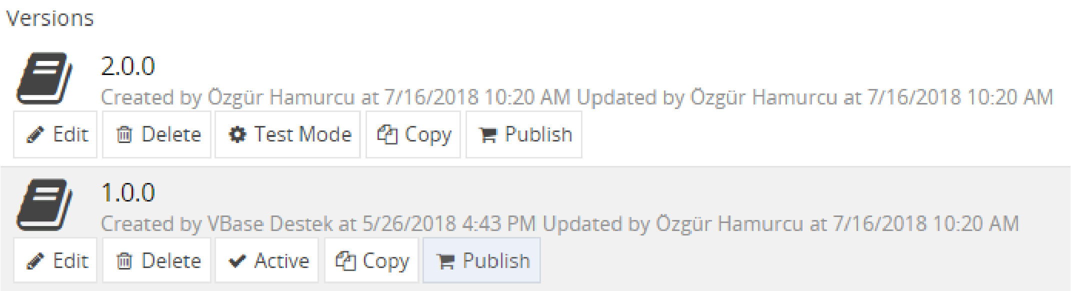
2.3.1. Data Model

All data required to be carried over between steps of processes are stored within a structure named Data Model. This Data Model is defined during the design process and is an industry standard of XSD (XML Scheme Definition). Due to the storage of data models by standard XSD definitions, XSD data types from external environments such as unlimited hierarchical models and web services can also be extracted and used.

On the other hand, workflow data is a new XML record that is produced according to Data Model when a new task is initiated within a process, and is kept till deleted from the system. All transactions within the process (form fields, scripts, role definitions, etc.) are based upon workflow data. Workflow data is accessed via XPath queries and is flexible to all modifications within XML standards.

2.3.2. Version Management

This allows storage of different versions and enables all development and change management to be tracked.



Versions

2.0.0

Created by Özgür Hamurcu at 7/16/2018 10:20 AM Updated by Özgür Hamurcu at 7/16/2018 10:20 AM

1.0.0

Created by VBase Destek at 5/26/2018 4:43 PM Updated by Özgür Hamurcu at 7/16/2018 10:20 AM

Unlimited number of versions can be created at any given time within the system. In status changes, (For example; Active version pulled to Archive status) workflows within the system do not get affected, and all active workflows are managed in accordance with the status set at the beginning.

All changes within a version are transiently reflected, for example, changes made to a script definitions on a step affects the active workflows within the system. Therefore, the change to be made should be reviewed to examine the effects on workflows; otherwise, it is highly suggested to create a new version.

When a version is deleted from the system, all subsequent active or passive workflows will be deleted as well. For this reason, it is recommended to archive versions rather than deleting, unless it is a must.

Within the system, it is possible to save different versions of the same process. The conditions can be any of the following:

Application States

Editing

This status is used when changes are still in place. It carries same features as Test Mode Status, however, changes can be undone if desired.

Test Mode

This status enables the version to be initiated only by authorized administrators, and to be redirected to the same user, regardless of role definitions within the process. By this method, it becomes less time consuming and very easy to manage control of system's functionality, without the need of logging in and out with different users.

Active

This status enables the version to be used by all authorized users.

Archive

This status places the version out of use.

2.3.3. Subprocesses

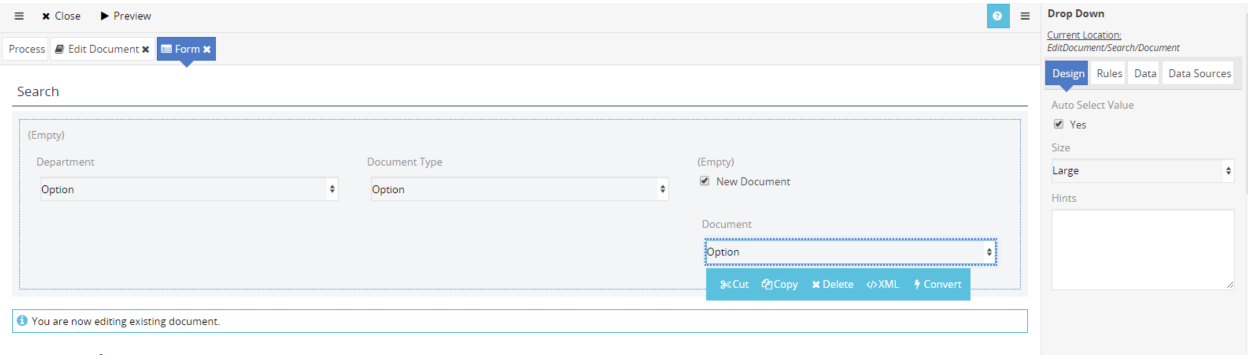
Workrunner supports sub processes, one of the major necessities in the design of large scale workflows. Depending on the form data definitions, it is possible to dynamically initiate unlimited sub processes. These sub processes can also be nested under unlimited levels. Since the sub process definitions are also packaged together with the main process, communication with other processes can be carried out with the use of web services.

2.3.4. Database

Database models defined within a process can be used jointly among multiple other processes. For example, a customer database designed for a customer management process can also be used independently in a separate help desk process. Where necessary, it is also possible to import data from external sources and treat it as a shared repository. This allows access to external data without using web services.

2.3.5. Form Design

Process interfaces on Workrunner forms are easily designed by the use of form design tools. Form design tools come with a rich interface element umbrella.



Search

(Empty)

Department: Option

Document Type: Option

(Empty)

☑ New Document

Document: Option

You are now editing existing document.

Drop Down

Current Location:
EditDocument/Search/Document

Auto Select Value

☑ Yes

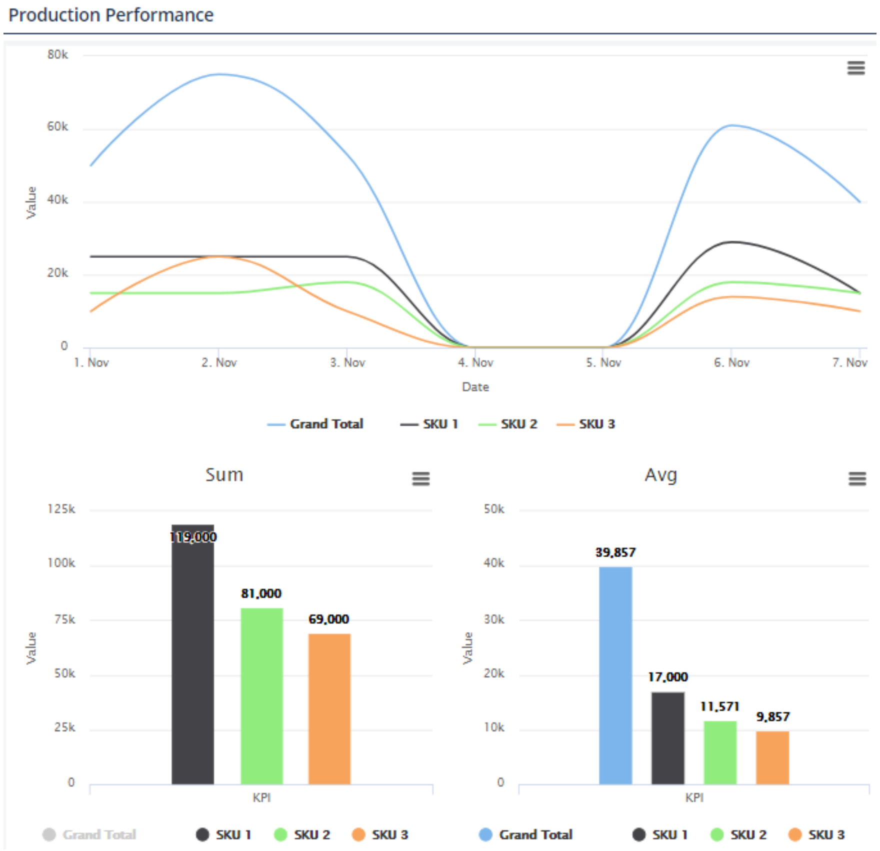
Size: Large

Hints

More than 40 form elements are available in the Form Designer. Using a combination of these elements, it is possible to create flexible forms that accept multiple-record entries with master/detail relationships. This makes fast interface development possible without the need for any coding.

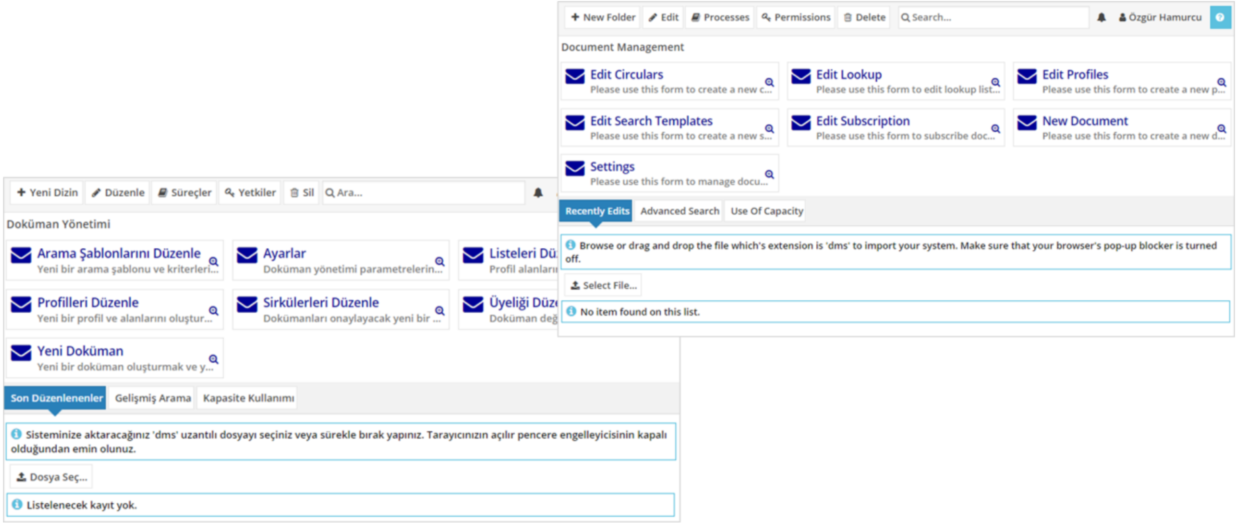
2.3.6. *Reporting*

Workrunner contains design screens for reporting purposes, in both data produced within processes and in data related to processes themselves. Conventional RDBMS and XML-based databases can be used as reporting data sources. In order to design reporting interfaces, there is an available reporting design tool just as the form design tool. When the sequence for reporting is attained, displayed panels (dashboards) or fixed reporting interface can be designed.



2.3.7. *Multi-Language Support*

Workrunner contains localization tools for the purpose of using different languages, regardless of the language a process is designed with.


+ New Folder
Edit
Processes
Permissions
Delete
Search...
Özgür Hamurcu
Document Management
Edit Circulars
Please use this form to create a new c...
Edit Lookup
Please use this form to edit lookup list...
Edit Profiles
Please use this form to create a new p...
Edit Search Templates
Please use this form to create a new s...
Edit Subscription
Please use this form to subscribe doc...
New Document
Please use this form to create a new d...
Settings
Please use this form to manage docu...
Recently Edits
Advanced Search
Use Of Capacity
Select File...
+ Yeni Dizin
Düzenle
Süreçler
Yetkiler
Sil
Ara...
Doküman Yönetimi
Arama Şablonlarını Düzenle
Yeni bir arama şablonu ve kriterleri...
Ayarlar
Doküman yönetimi parametrelerin...
Listeleri Dü
Profil alanları
Profilleri Düzenle
Yeni bir profil ve alanlarını oluştur...
Sirkülerleri Düzenle
Dokümanları onaylayacak yeni bir ...
Üyeliği Düz
Doküman değ
Yeni Doküman
Yeni bir doküman oluşturmak ve y...
Son Düzenlenenler
Gelişmiş Arama
Kapasite Kullanımı
Dosya Seç...

Browse or drag and drop the file which's extension is 'dms' to import your system. Make sure that your browser's pop-up blocker is turned off.

No item found on this list.

Sisteminize aktaracağınız 'dms' uzantılı dosyayı seçiniz veya sürekle bırak yapınız. Tarayıcınızın açılır pencere engelleyicisinin kapalı olduğundan emin olunuz.

Listelenecek kayıt yok.

Localization tools scan all objects used within the interface, and acts as a dictionary for a different language provisions. Unlimited number of dictionaries can be placed for localization purposes, and this dictionary remains valid for different systems as it will be carried on the process.

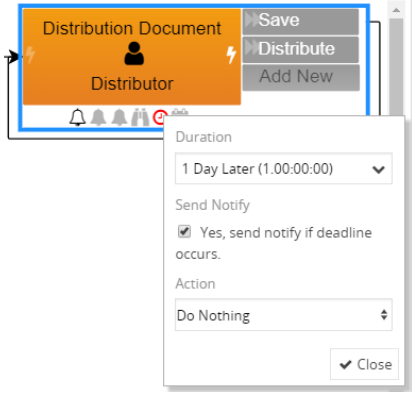
Besides localization of processes, all other interfaces are displayed according to the user's preferred language settings. For instance, areas such as date are displayed as the user's preferred language settings.

2.3.8. Time Management

Workrunner stores all date and time values annotated with proper time zone information so that users located in different time zones can cooperate with a correct time representation. All time-sensitive elements such as deadline notifications or business calendar entries employ the local time zones for individual users.

A process model can contain time-bound tasks. Any tasks assigned to users can be configured to send out periodic reminders until the task is finalized. It is possible to separately configure the delay of the initial reminder and the frequency of subsequent reminders.

A user-assigned task can also have a deadline date. Depending on the deadline definition, the task can be re-assigned to another user or it can be automatically finalized so that the workflow will proceed to the next step.



Duration

1 Day Later (1.00:00:00)

Send Notify

☑ Yes, send notify if deadline occurs.

Action

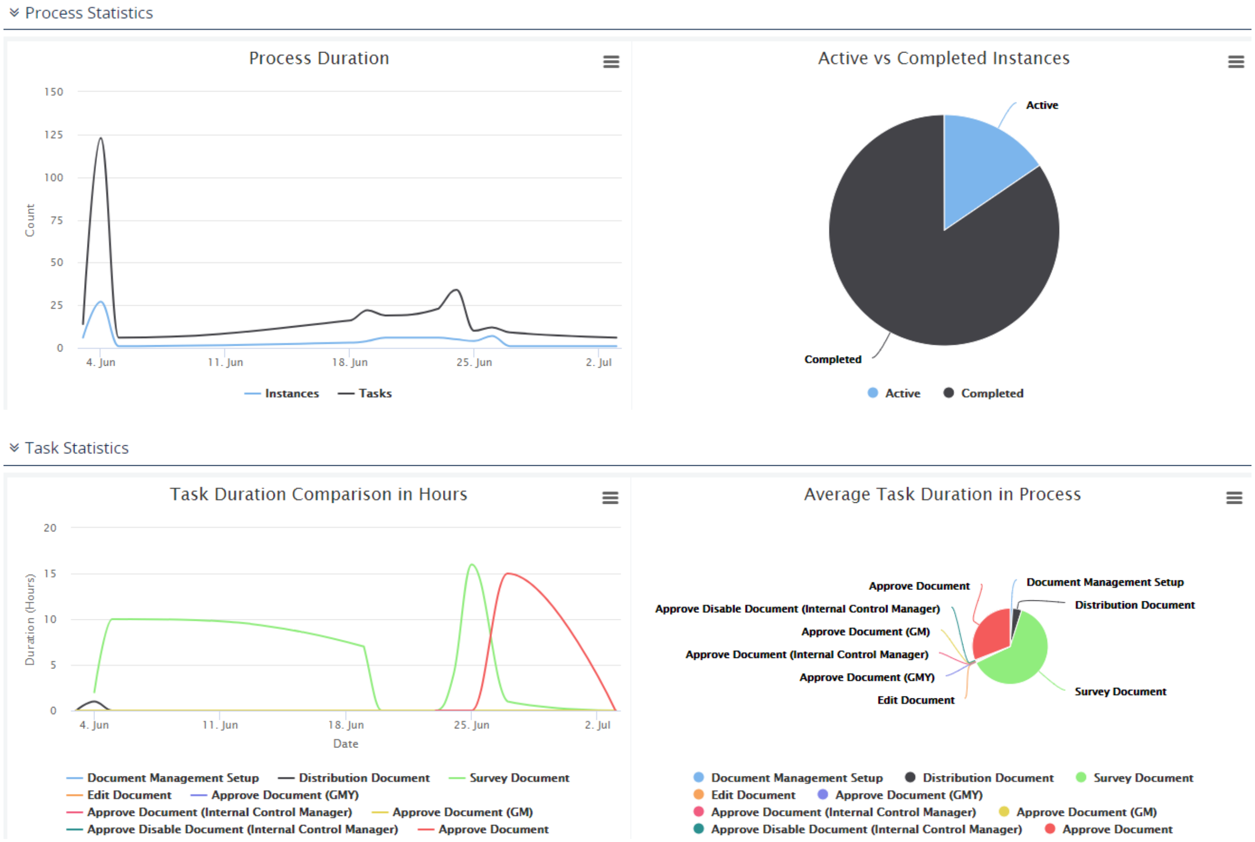
Do Nothing

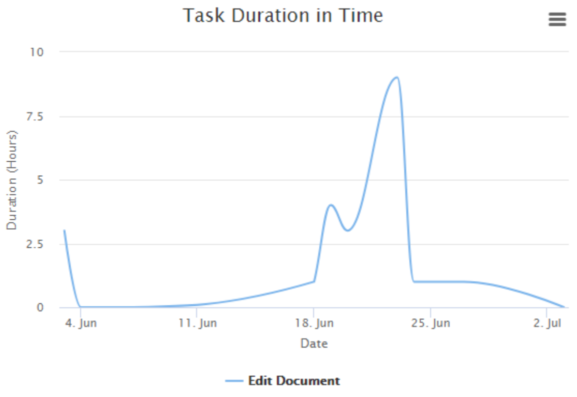
✓ Close

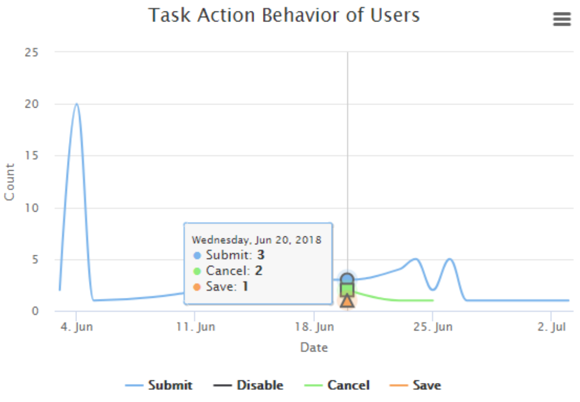
Tasks can have "evaluation calendars" that contain definitions for automatically initiating periodic processes or finalizing/re-assigning user tasks according to certain conditions.

2.3.9. *Statistical Analysis*

Workrunner keeps track of statistical data for every workflow associated with a process in addition to every assignment associated with those workflows. This data can then be used in analyzing or refining current process designs.

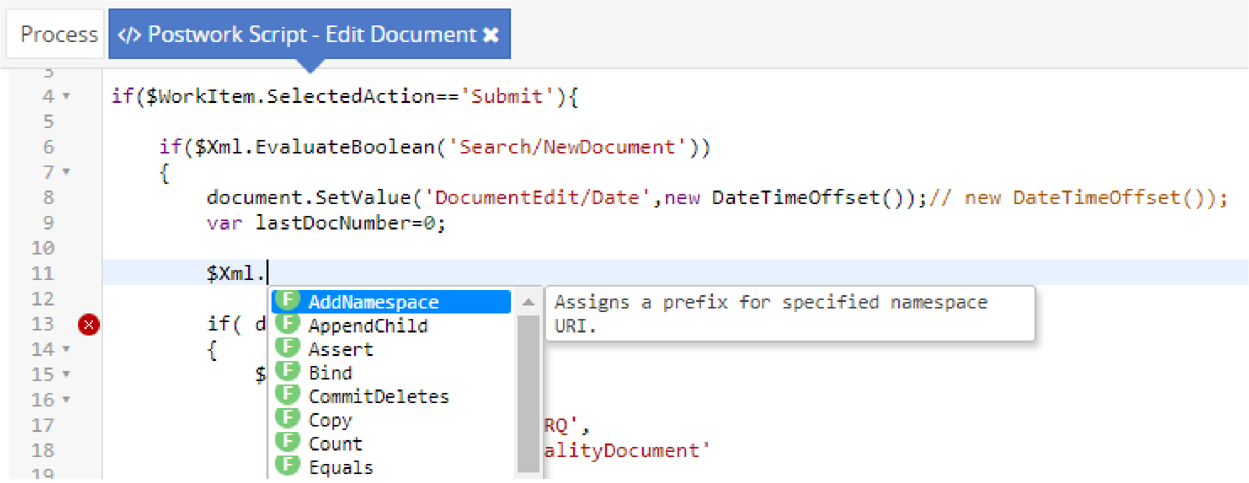






2.3.10. Scripting and Business Rules

Workrunner provides scripting support in many points within processes to allow execution of custom business rules. All scripts are written in JavaScript (ECMAScript 5) which is familiar to most developers. Access to form data from within scripts is facilitated via XPath or XQuery, two widely accessible standards.



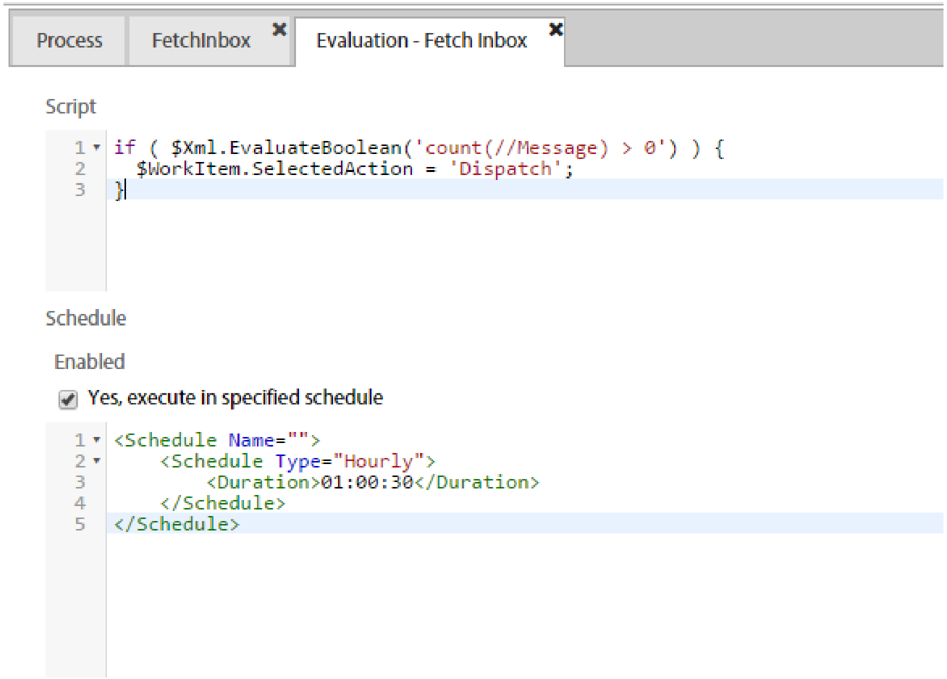
```
if($WorkItem.SelectedAction=='Submit'){

    if($Xml.EvaluateBoolean('Search/NewDocument'))
    {
        document.SetValue('DocumentEdit/Date',new DateTimeOffset());// new DateTimeOffset());
        var lastDocNumber=0;

        $Xml.

        if( d
        {
            $

                                                       RQ',
                                                       alityDocument'
```

Assigns a prefix for specified namespace URI.

Scripts also have access to work calendars, organization hierarchy and database objects. Development documentation with regard to scripting is further explained at https://workrunner.github.io/

2.3.11. Scheduled Tasks

Tasks with predetermined time priorities and rules can be appointed. Appointed task can be assigned to users under specified circumstances by starting a new workflow within the system, or can be completed by assigned tasks on users under specified circumstances.



Script

```
if ( $Xml.EvaluateBoolean('count(//Message) > 0') ) {
  $WorkItem.SelectedAction = 'Dispatch';
}
```

Schedule

Enabled

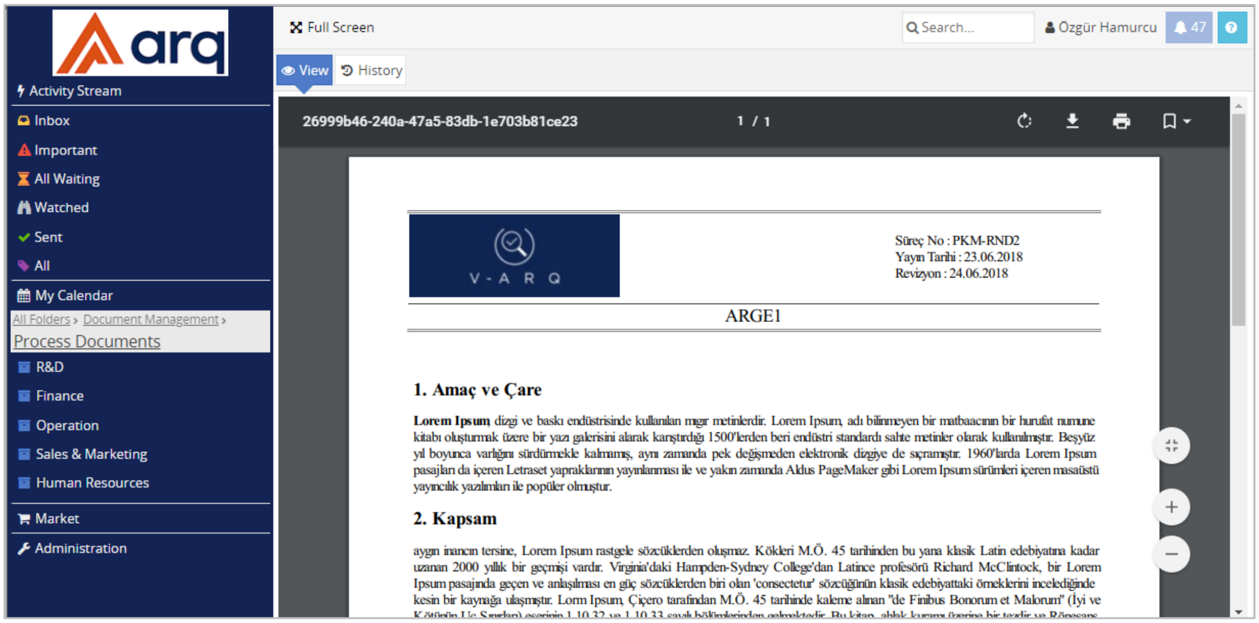
☑ Yes, execute in specified schedule

```
<Schedule Name="">
    <Schedule Type="Hourly">
        <Duration>01:00:30</Duration>
    </Schedule>
</Schedule>
```

3. Document Archive

Workrunner provides a document archive in order for a centralized and safe environment when producing data in processes and/or attaching documents in any of the forms.

Similar to the processes, documents are stored on the authorization directory tree within the system. When necessary, document based authorization is also available for usage.



26999b46-240a-47a5-83db-1e703b81ce23

Süreç No : PKM-RND2
Yayın Tarihi : 23.06.2018
Revizyon : 24.06.2018

ARGE1

1. Amaç ve Çare

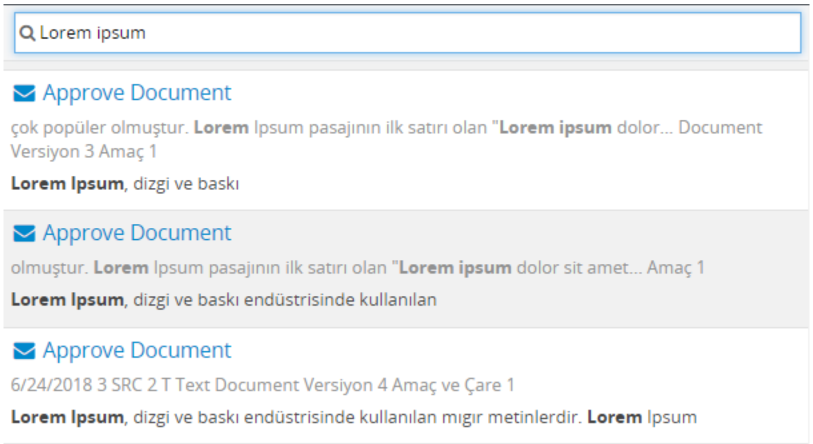
Lorem Ipsum dizgi ve baskı endüstrisinde kullanılan mıgır metinlerdir. Lorem Ipsum, adı bilinmeyen bir matbaacının bir hurufat numune kitabı oluşturmak üzere bir yazı galerisini alarak karıştırdığı 1500'lerden beri endüstri standardı sahte metinler olarak kullanılmıştır. Beşyüz yıl boyunca varlığını sürdürmekle kalmamış, aynı zamanda pek değişmeden elektronik dizgiye de sıçramıştır. 1960'larda Lorem Ipsum pasajları da içeren Letraset yapraklarının yayınlanması ile ve yakın zamanda Aldus PageMaker gibi Lorem Ipsum sürümleri içeren masaüstü yayıncılık yazılımları ile popüler olmuştur.

2. Kapsam

aygın inancın tersine, Lorem Ipsum rastgele sözcüklerden oluşmaz. Kökleri M.Ö. 45 tarihinden bu yana klasik Latin edebiyatına kadar uzanan 2000 yıllık bir geçmişi vardır. Virginia'daki Hampden-Sydney College'dan Latince profesörü Richard McClintock, bir Lorem Ipsum pasajında geçen ve anlaşılması en güç sözcüklerden biri olan 'consectetur' sözcüğünün klasik edebiyattaki örneklerini incelediğinde kesin bir kaynağa ulaşmıştır. Lorm Ipsum, Çiçero tarafından M.Ö. 45 tarihinde kaleme alınan "de Finibus Bonorum et Malorum" (İyi ve Kötünün Uç Sınırları) eserinin 1.10.32 ve 1.10.33 sayılı bölümlerinden gelmektedir. Bu kitap, ahlak kuramı üzerine bir tezdir ve Rönesans

In order to track continuously updated documents, history information on documents is available, and access is also available to old versions of documents.

Within the index sections and content, there is search available for easy access to documents at any later point. In virtue of MS Windows IFilter structure, content search is doable in different formats such as PDF, DOC, TIFF, etc.



Lorem ipsum

Approve Document

çok popüler olmuştur. **Lorem** Ipsum pasajının ilk satırı olan "**Lorem ipsum** dolor... Document Versiyon 3 Amaç 1

Lorem Ipsum, dizgi ve baskı

Approve Document

olmuştur. **Lorem** Ipsum pasajının ilk satırı olan "**Lorem ipsum** dolor sit amet... Amaç 1

Lorem Ipsum, dizgi ve baskı endüstrisinde kullanılan

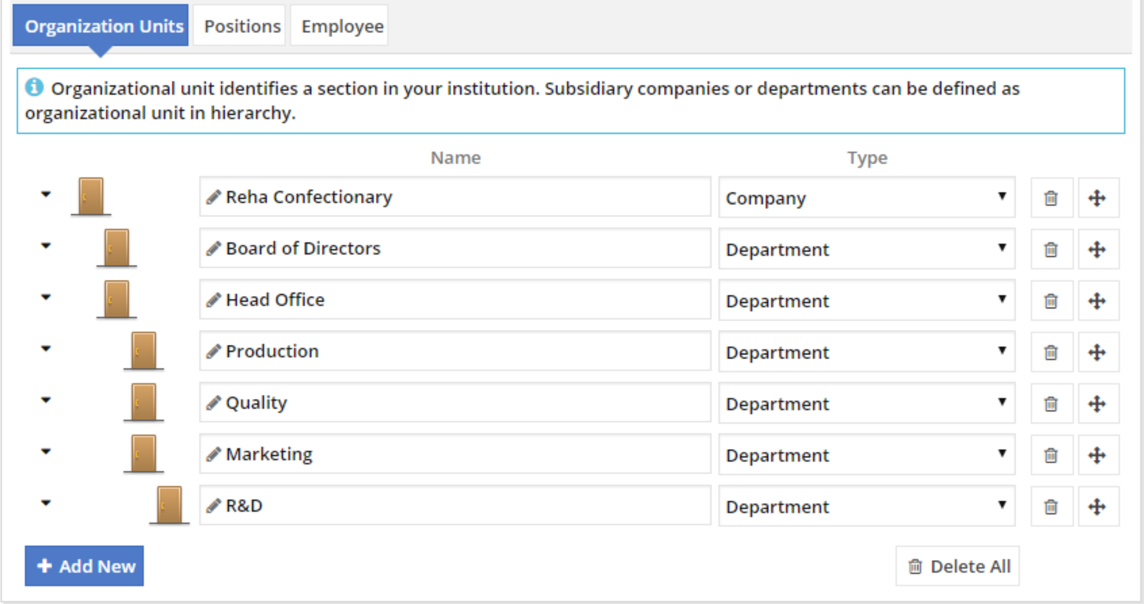
Approve Document

6/24/2018 3 SRC 2 T Text Document Versiyon 4 Amaç ve Çare 1

Lorem Ipsum, dizgi ve baskı endüstrisinde kullanılan mıgır metinlerdir. **Lorem** Ipsum

4. Organization Database

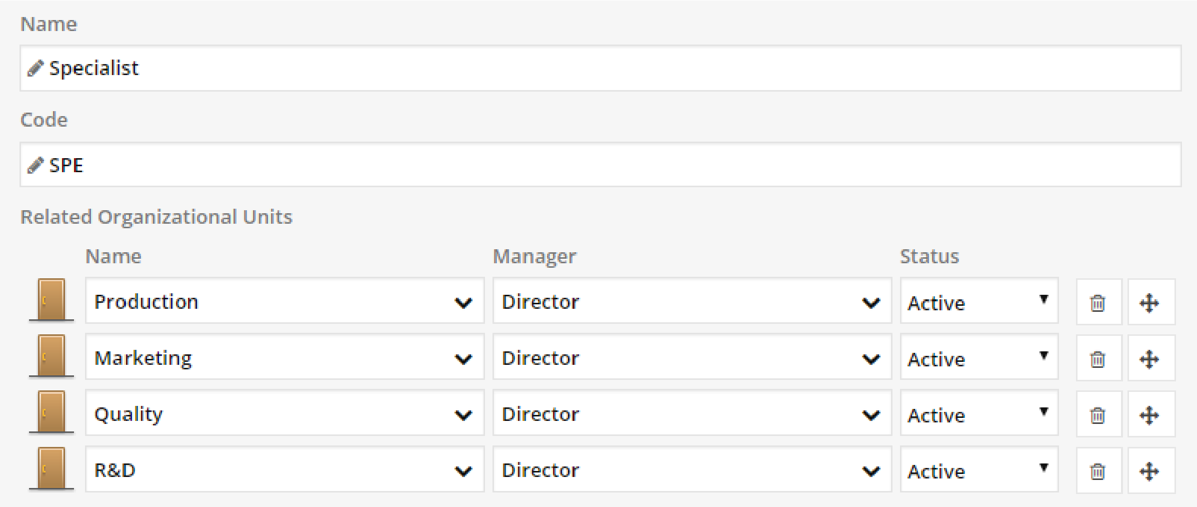
An organizational hierarchy, needed in most processes, can be designed in a flexible organization database editor. This editor allows creating a department tree that reflects the organization chart and defining employee positions within this hierarchy. Ready-made organization templates allow rapid creation of the initial hierarchy. It is possible to keep the organization database up to date via synchronization with external databases or Active Directory servers.



Organizational unit identifies a section in your institution. Subsidiary companies or departments can be defined as organizational unit in hierarchy.

Name	Type
Reha Confectionary	Company
Board of Directors	Department
Head Office	Department
Production	Department
Quality	Department
Marketing	Department
R&D	Department

Position definitions represent the duties within the hierarchy and are linked with departments within the organization.



Name

Specialist

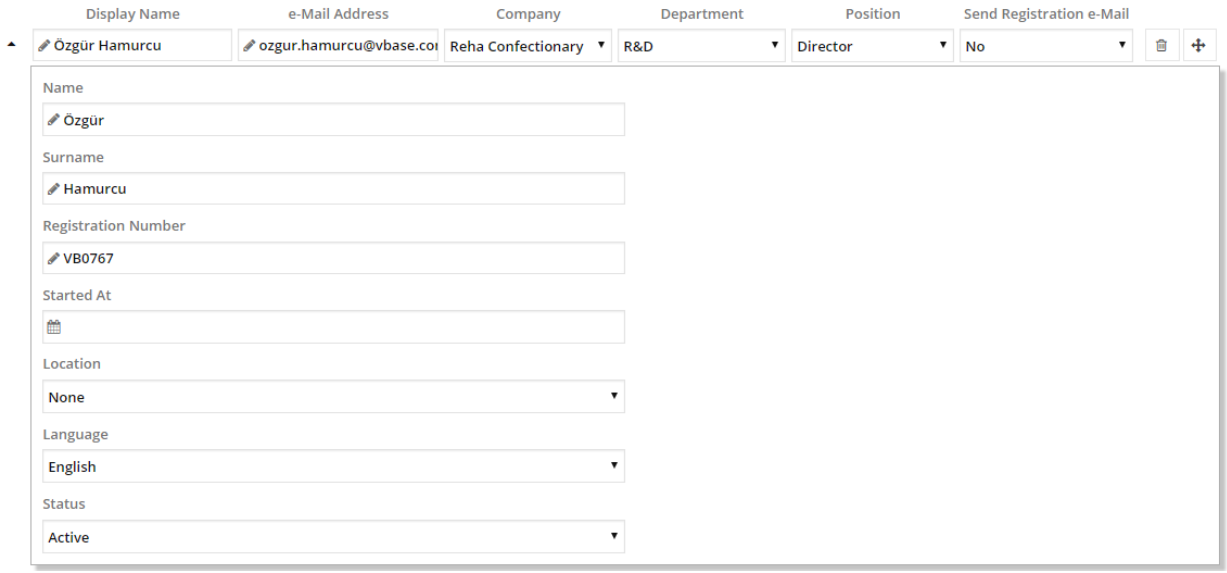
Code

SPE

Related Organizational Units

Name	Manager	Status
Production	Director	Active
Marketing	Director	Active
Quality	Director	Active
R&D	Director	Active

Employee definitions store personnel-related data such as e-mail addresses, department and position.



Display Name	e-Mail Address	Company	Department	Position	Send Registration e-Mail
Özgür Hamurcu	ozgur.hamurcu@vbase.co	Reha Confectionary	R&D	Director	No

Name

Özgür

Surname

Hamurcu

Registration Number

VB0767

Started At

Location

None

Language

English

Status

Active

5. Technology

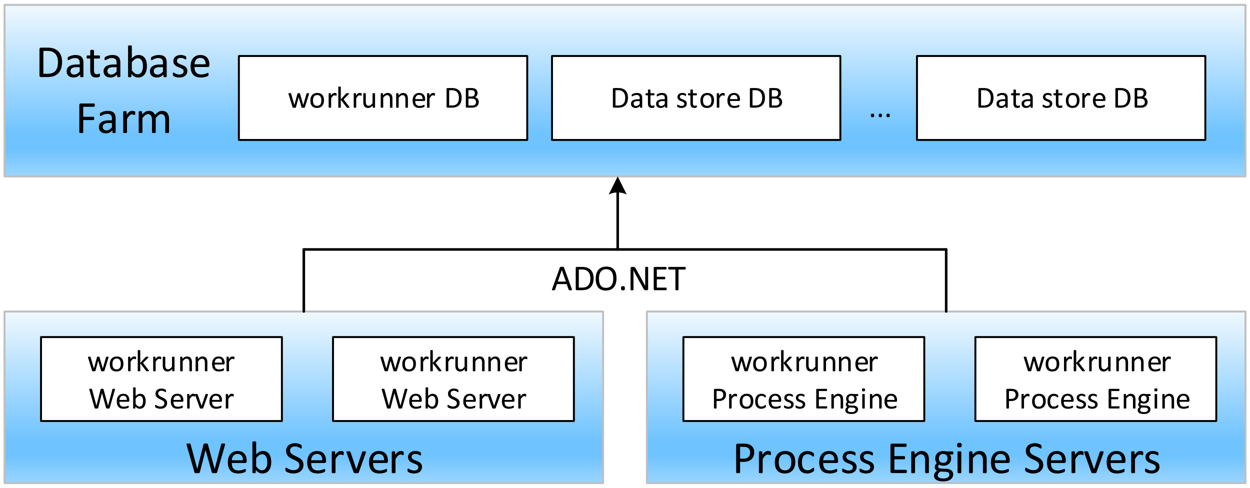
Workrunner has been developed with .NET Framework v4.0. The Process Engine and integration services require a Windows Server system, while the database layer requires MS SQL Server. The end-user interfaces and process design tools, being completely web based, can be run on any operating system (Windows/Linux/iOS). A web browser with JavaScript support is the only requirement.

6. Performance

6.1. Scalability

Workrunner is a fully stateless, enterprise class BPM system, able to cover scaling requirements of large and distributed organizations.

Platform architecture allows vertical scaling over multiple CPUs and horizontal scaling over multiple server machines. Multiple instances of web servers or process engines can work simultaneously on the same database. All database access is carried out using the .NET standard, ADO.NET. The data that belongs to the Process Engine and the processes themselves can be located in different databases, abstracted by data store components.



6.2. Load Balancing

Web Servers can be configured for hardware- or software-supported load balancing. Process Engine servers benefit from dynamic load balancing by communicating over the database.

Clients communicate with the server via transactional XML messages. The messages received from clients, before any processing, are recorded into a queue in the database by the web servers. In a multi-threaded architecture, Process Engines pull jobs from this queue and perform multiple transactions simultaneously, releasing the server resources upon completion. This allows processing of large numbers of requests with minimal resource use.

All operations carried out by Process Engines are managed by MS-DTC so every internal and external database access in addition to WS-Transaction supported web service accesses are transactional. The error conditions are handled by rolling back all operations (on local and external resources) performed by the engine. All error cases are recorded on the database and forwarded to system administrators. After corrective measures, an administrator can re-queue the task for processing.

6.3. Performance Statistics

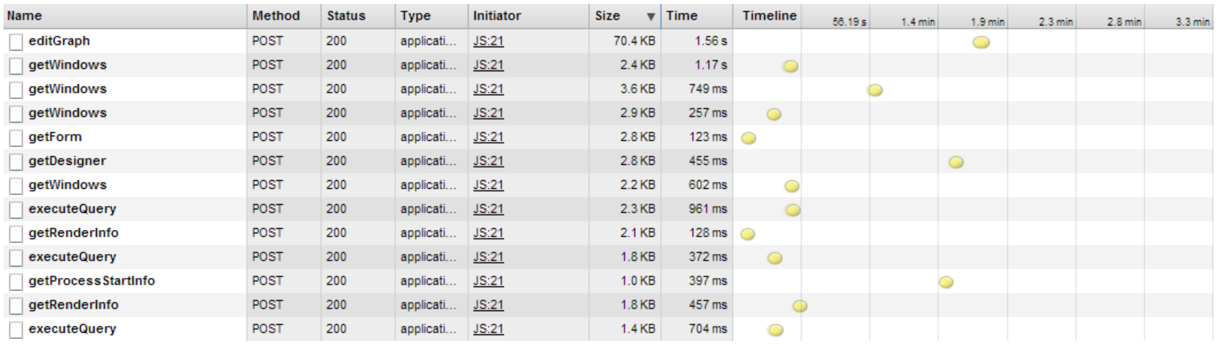
6.3.1. Web Server Traffic

Workrunner web interface is designed primarily as an SPA (Single Page Application). All the resources required by the application (JavaScript code, images, etc.) reside on the host, such as "static.workrunner.com". While the application is running, any additional resources needed by the application are loaded automatically. Browser caching ensures that these resources are downloaded to the client machine only once.

The following table lists typical bandwidth usage on the initial and subsequent runs of the application.

	Initial Run	Subsequent Runs
HTML	3 KB	3 KB
CSS	15 KB	0 KB
JavaScript	300 KB	0 KB
Image	60 KB	0 KB
AJAX	5 KB	5 KB
Total	**400 KB**	**8 KB**

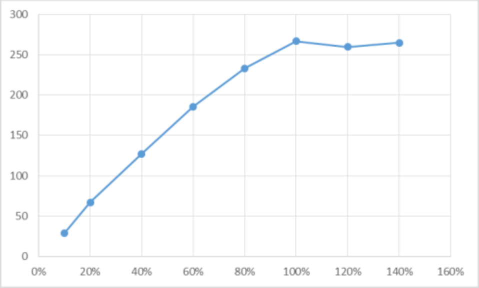
Apart from this data traffic, all the communication with server take place over RPC services. The amount of data transferred is variable among different process designs, but the average data traffic varies between 1 and 100 KBs. A typical RPC traffic is visualized below.



Name	Method	Status	Type	Initiator	Size ▼	Time
□ editGraph	POST	200	applicati...	JS:21	70.4 KB	1.56 s
□ getWindows	POST	200	applicati...	JS:21	2.4 KB	1.17 s
□ getWindows	POST	200	applicati...	JS:21	3.6 KB	749 ms
□ getWindows	POST	200	applicati...	JS:21	2.9 KB	257 ms
□ getForm	POST	200	applicati...	JS:21	2.8 KB	123 ms
□ getDesigner	POST	200	applicati...	JS:21	2.8 KB	455 ms
□ getWindows	POST	200	applicati...	JS:21	2.2 KB	602 ms
□ executeQuery	POST	200	applicati...	JS:21	2.3 KB	961 ms
□ getRenderInfo	POST	200	applicati...	JS:21	2.1 KB	128 ms
□ executeQuery	POST	200	applicati...	JS:21	1.8 KB	372 ms
□ getProcessStartInfo	POST	200	applicati...	JS:21	1.0 KB	397 ms
□ getRenderInfo	POST	200	applicati...	JS:21	1.8 KB	457 ms
□ executeQuery	POST	200	applicati...	JS:21	1.4 KB	704 ms

6.3.2. *BPM Running Time Values*

While the total execution time for a single action can vary depending on the operations performed, tasks executed by the BPM engine typically take about 30 milliseconds. An agent server machine with 8 CPUs, working at 100% performance, is able to process about 250 workflow tasks per second.



CPU Utilization

[1]Running time data has been tested on an Intel i7 1.73 GHz hardware and may vary depending on the operation system, network connections and operations performed. This data is provided only as an illustration of overall system performance.