# Organizational Units

The travel agency has expanded its business and decided to include several new branches.

Consequently, the air ticket reservation process we have developed so far will be an activity to be done at all branches. This means that, at a given time, there will be personnel from all the different offices executing the same process on the Workflow server.
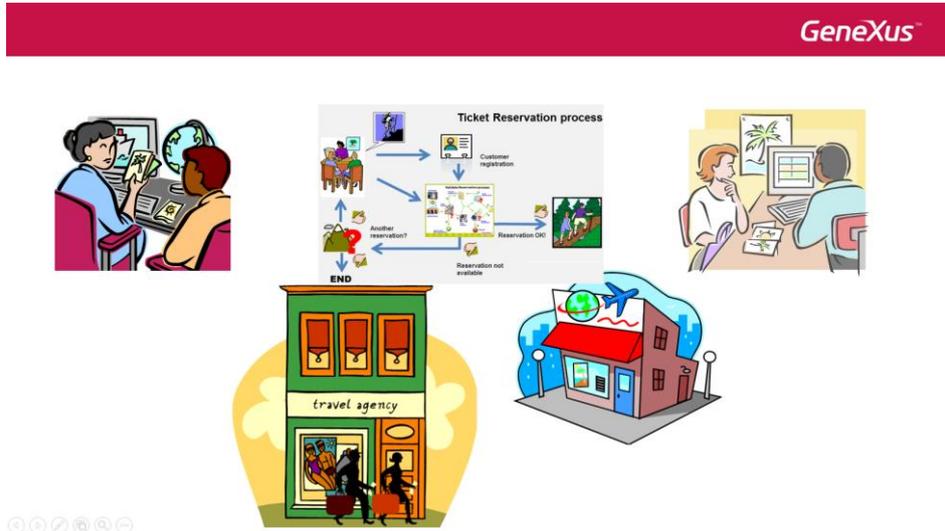
So, how what do we make it so that that a branch's processes and tasks are not available to the remaining branches? In other words, an employee at one branch should not be able to execute a task corresponding to a different branch.

The solution to this is the Organizational Unit concept.

Organizational Units enable us to define groups within an organization, such as branches, departments and so on. Once the units are created, we assign to them users from the organization.

In the case of our travel agency there are two offices, one in New York and the other one in Miami Beach.

Once the organizational units are defined we assign the corresponding organizational unit to each user. From that point on, the process will inherit the organizational unit of the user who initiates it.
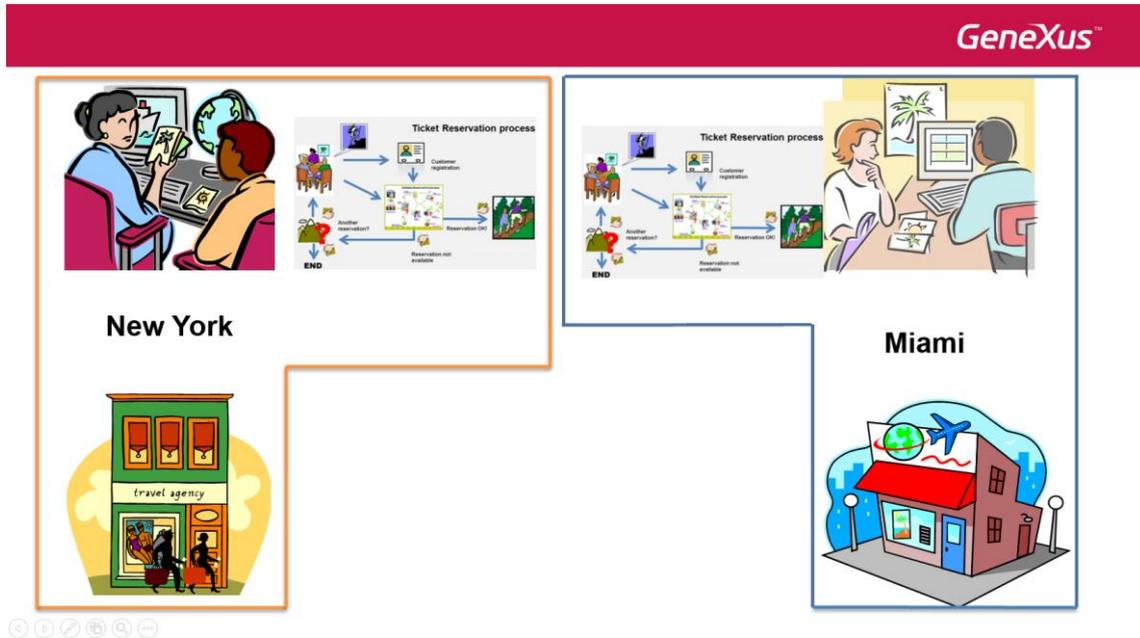
This means that users will only view in their incoming trays the tasks of the organizational unit to which each user belongs. In other words, users of one branch will not be able to view or execute tasks of another branch.

For example, if we assign the New York office to Peter Kent, when Peter initiates a ticket reservation process, the reservations (…) will be those of the New York office.
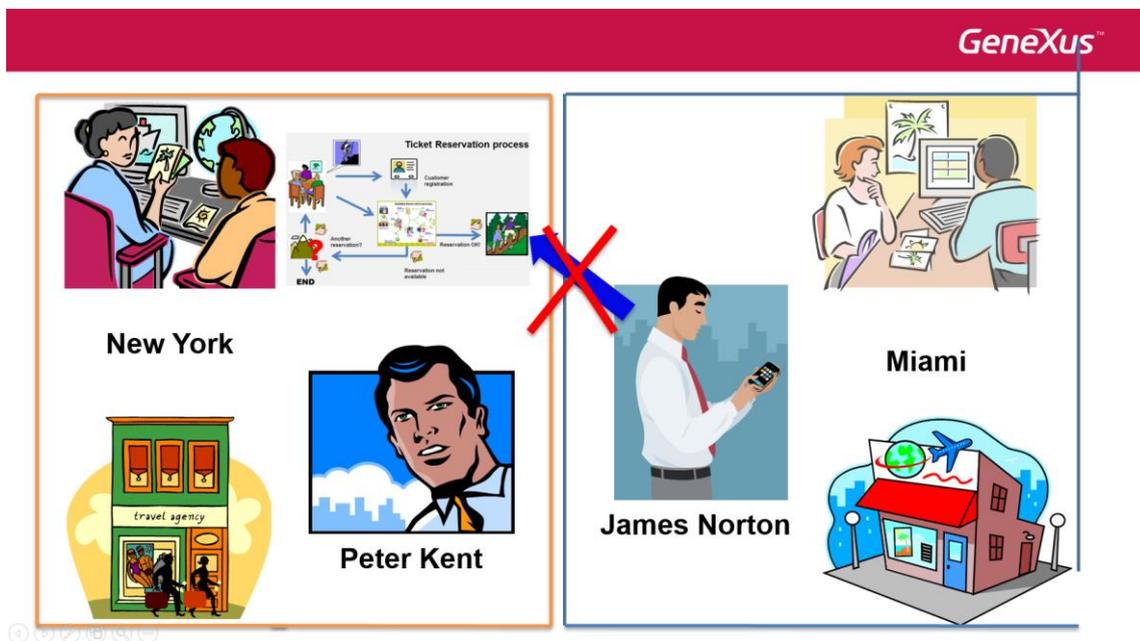
If James Norton is from the Miami office he will not be able to view the process initiated by Peter or execute the tasks that Peter can view, even whey they both access the same GXflow server. Only users with the same branch as Peter will be able to access the tasks of the process initiated by him.

In order to define the branches of the travel agency, we will open the Gxflow client, log on with the administrator user and go to Administration Console, Organizational Model, and click on Organizational Unit Definitions.

Here we define the type of organizational unit that the firm has, such as branch, department, division, and so on.

To create a definition of organizational unit we press the "New" button and in the window that opens up we write the type of organizational unit, which in our case is "Subsidiary", along with a description.


The "Propagable" option allows the process to inherit the user's organizational unit. It also indicates that subprocesses will inherit the organizational unit of the parent process.

If this is not checked, the user may initiate a process but it will not inherit the organizational unit defined for the user.

When a process has no organizational unit defined it may be viewed by users depending exclusively on their role restrictions, which is the situation which we have been working with so far.

Once we have defined the firm's type of organizational units we define the organizational units per se. To do so we click on the Organizational Units option.
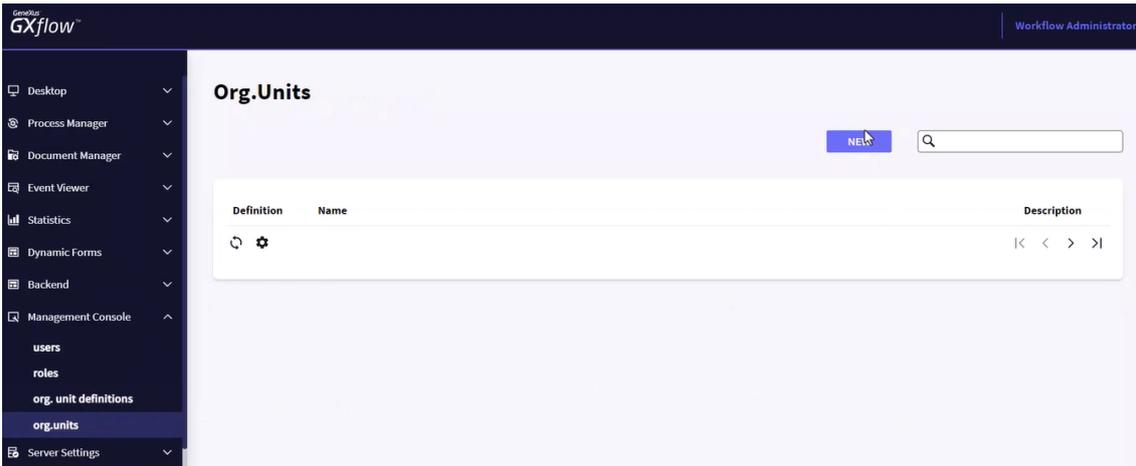
When we press New, we can create an organizational unit by entering its name, type and description. Let's now enter the data for the New York branch.

And then we do the same for the Miami branch.

This defines the two branches of the travel agency.

Org.Units

	NEW		🔍

Definition	Name	Description
Subsidiary	Miami	Travel Agency office at Miami beach
Subsidiary	New York	Travel Agency office at New York city

↻ ⚙ |< < > >|

For entering users of the New York branch we select the row and press Members.

In the window we check Anna and Peter as belonging to the New York branch.

Org. Unit Members

Users	Assigned Users
🔍	🔍
Name	**Name**
Workflow Administrator	ANNA SMITH
	PETER KENT

ADD
ADD ALL
REMOVE
REMOVE ALL

Prior to defining users of the Miami branch we must create them. To do so we go to Users and create user JNORTON, corresponding to employee James Norton, whose password will be jnorton123.

New User   ✕

General

User	JNORTON
Name	James Norton
Email	jnorton@example
Password	•••••
R. Password	•••••

User must change password at next logon
○ True
● False

OK

 We also create user PWONG, corresponding to employee Patrice Wong, with the password pwong123.

New User   ×

General
_____

User   │ PWONG   �X │

Name   │ Patrice Wong │

Email   │ pwong@example.com │

Password   │ ••••• │

R. Password   │ •••│ │

User must change password at   ○ True
next logon   ◉ False

                                              [OK]

Once we have created the users we define their roles. We select user JNORTON and press the Roles button.

Users

                                              [NEW]   [🔍   ]

👤	👤	Id	Name	Email
		ASMITH	ANNA SMITH	asmith@gmail.com
		JNORTON	James Norton	jnorton@example
		PWONG	Patrice Wong	pwong@example.com
		PKENT	PETER KENT	pkent@gmail.com
		WFADMINISTRATOR	Workflow Administrator	

🔄 ⚙ |< < > >|

In the roles application we define James as manager and press OK.

User Roles   ×

Roles   Assigned Roles

[🔍   ]   [🔍   ]

Name   Name

GXflow Administrator

GXflow Backend Administrator

GXflow Form Designer   ADD

GXflow Manager   ADD ALL

GXflow Manager Read Only   REMOVE

GXflow Prototyper   REMOVE ALL

GXflow Security Administrator

TravelAgencyEmployee

TravelAgencyManager

Then we select Patrice and define her as employee.

User Roles

Roles

🔍

Name
GXflow Administrator
GXflow Backend Administrator
GXflow Form Designer
GXflow Manager
GXflow Manager Read Only
GXflow Prototyper
GXflow Security Administrator
TravelAgencyEmployee
TravelAgencyManager

ADD
ADD ALL
REMOVE
REMOVE ALL

Assigned Roles

🔍

Name

From here we can define Patrice's and James' organizational units. Located in PWONG we press the Organizational Units button and select Miami:

User Organizational Units

Org.Units

🔍

Name
Miami
New York

ADD
ADD ALL
REMOVE
REMOVE ALL

Assigend Organizational Units

🔍

Name

Now we select JNORTON and assign him to Miami as well.

This is how we define the manager at New York to be Anna Smith, with employee Peter Kent, while the manager in Miami is James Norton and Patrice Wong is employee at that branch.
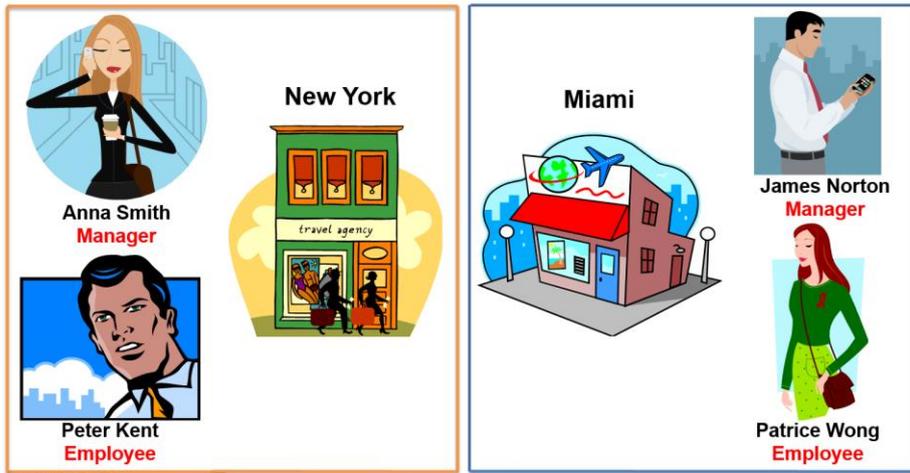
Now let's test the definitions tha we've just made in runtime.

First we Sign out and log on with user PWong. Then we press the New button and initiate an air ticket reservation process.
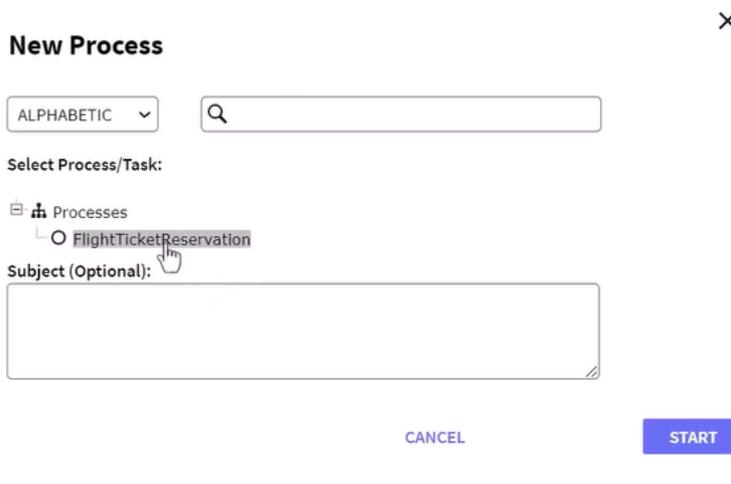
In the incoming tray we can see the TicketReservation task as pending execution.
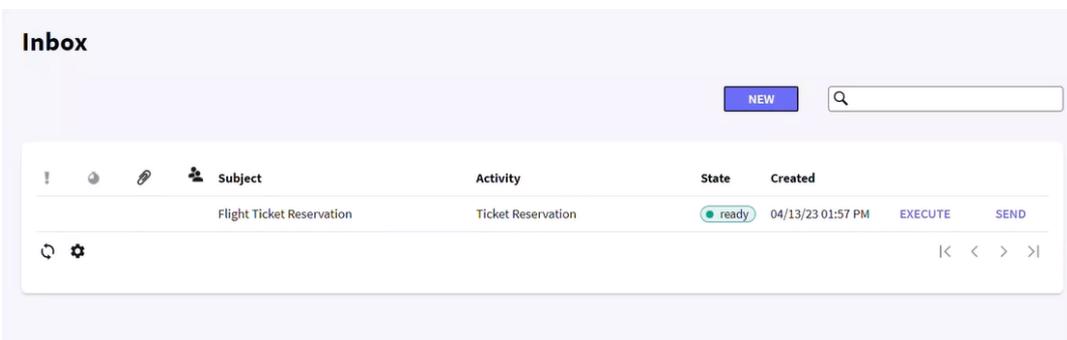
Since Patrice has the TravelAgencyEmployee role, she could initiate the process, and she has permits to execute the ticket reservation task. And because Patrice is from the Miami office, the process initiated should have inherited the Miami organizational unit.

To verify this we log on as administrator user, then select Processes in the Process Manager.

We can see that the process initiated by user Patrice Wong is being executed. If we select the process and go to More Actions we can select Organizational Units.

This way we can see that the process belongs to the Miami branch because it inherited the definition of Patrice Wong's organizational unit, as expected.

Let's now Sign Out and log on with user Peter Kent from the New York branch.

Inbox

| NEW | 🔍 |

!	⬤	✎	👥	💬 Subject	Activity	State	Created
↻ ⚙							|< < > >|

We can see that the incoming tray is empty. Even when a process has been initiated and the TicketReservation task remains pending, regardless of Peter having the adequate role, he will not be able to view it because he belongs to a different office from the one corresponding to the execution process.

Likewise, the customer may only be assessed and the reservation may only be authorized by user James Norton from the Miami branch.

In sum, only users assigned to the branch of the initiated process may interact with it, executing tasks according to their assigned roles.

We should point out that, if required, it is possible to add new organizational units or change the organizational unit in a process by using the Workflow API.

In the next video we will see how to perform analyses and obtain performance statistics for processes and tasks.