

Publishing

Focus Area: Back-end



Group Workshop

Back-end

Services
Integration

Content
Services

Advanced
Security

Targeting

Publishing

Front-end

Overview

Widget
Development

Template
Development

Portal Client

ICE

Foundation

Portal Essentials

Portal Technologies

Portal Tools

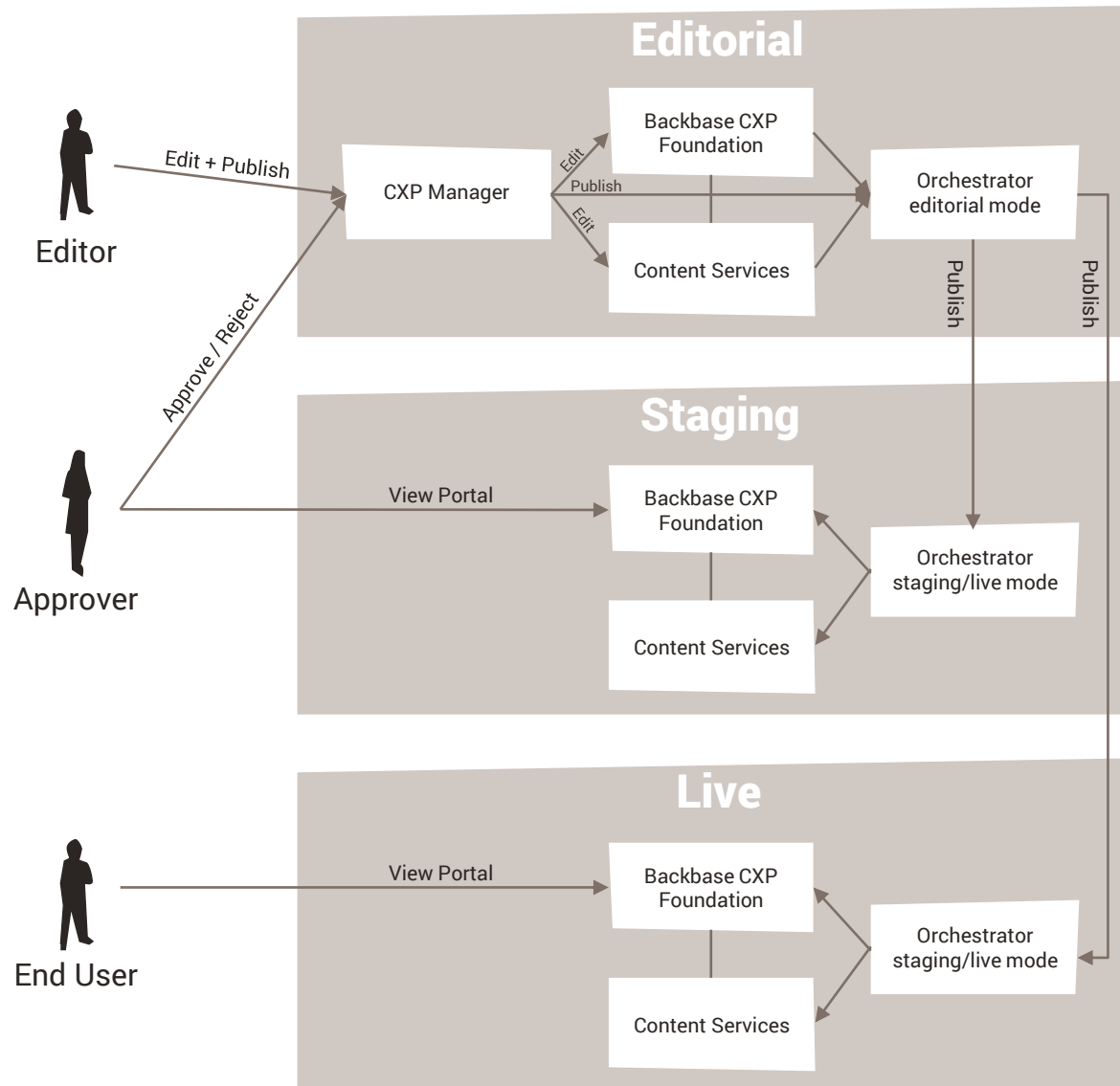
Portal APIs

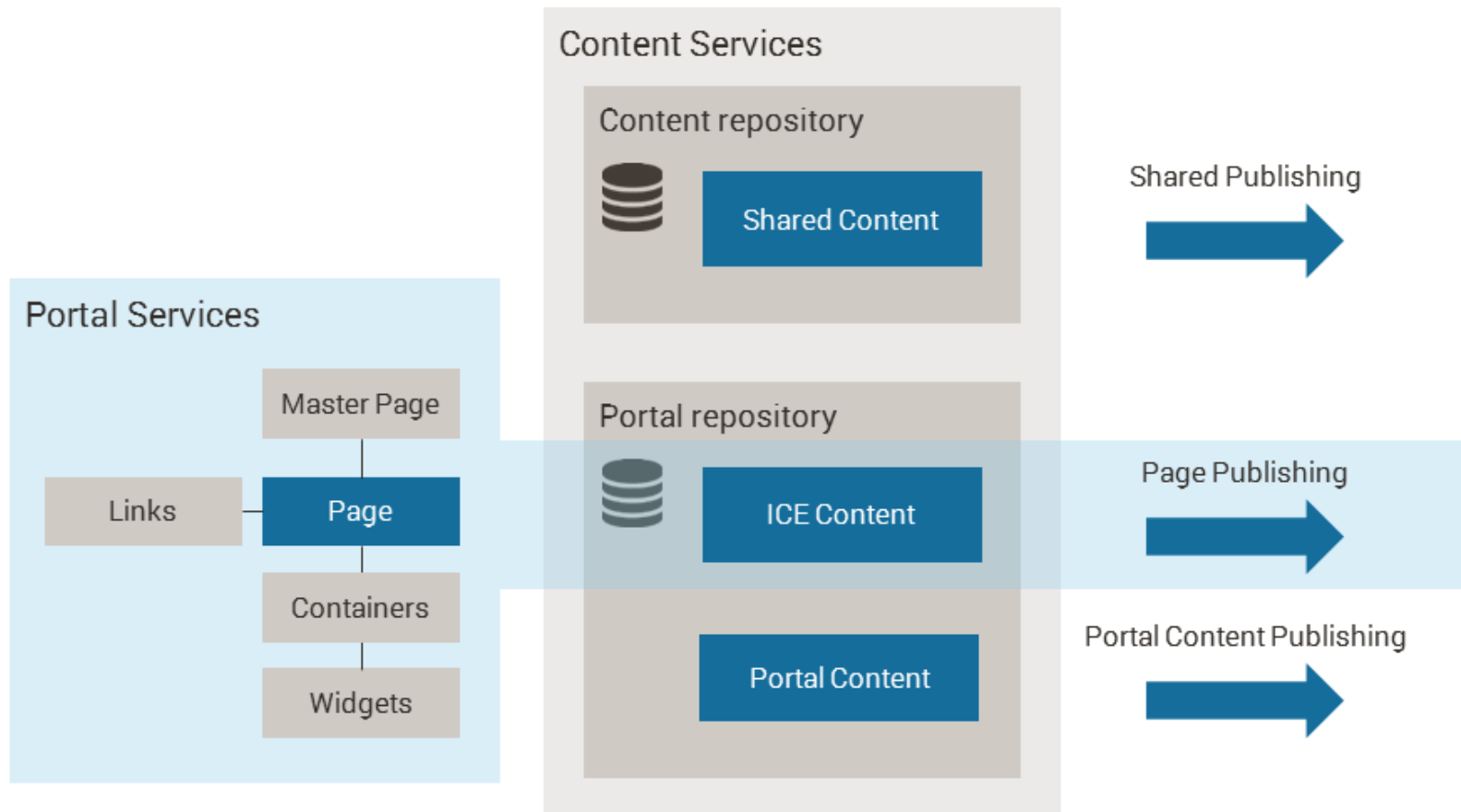
1. Understand the principles of Publishing.
2. Understand the typical environment setup.
3. Understand Publishing Chains.
4. Understand Versioning.
5. Understand Publishing Configuration.

Intro to Publishing

Publishing

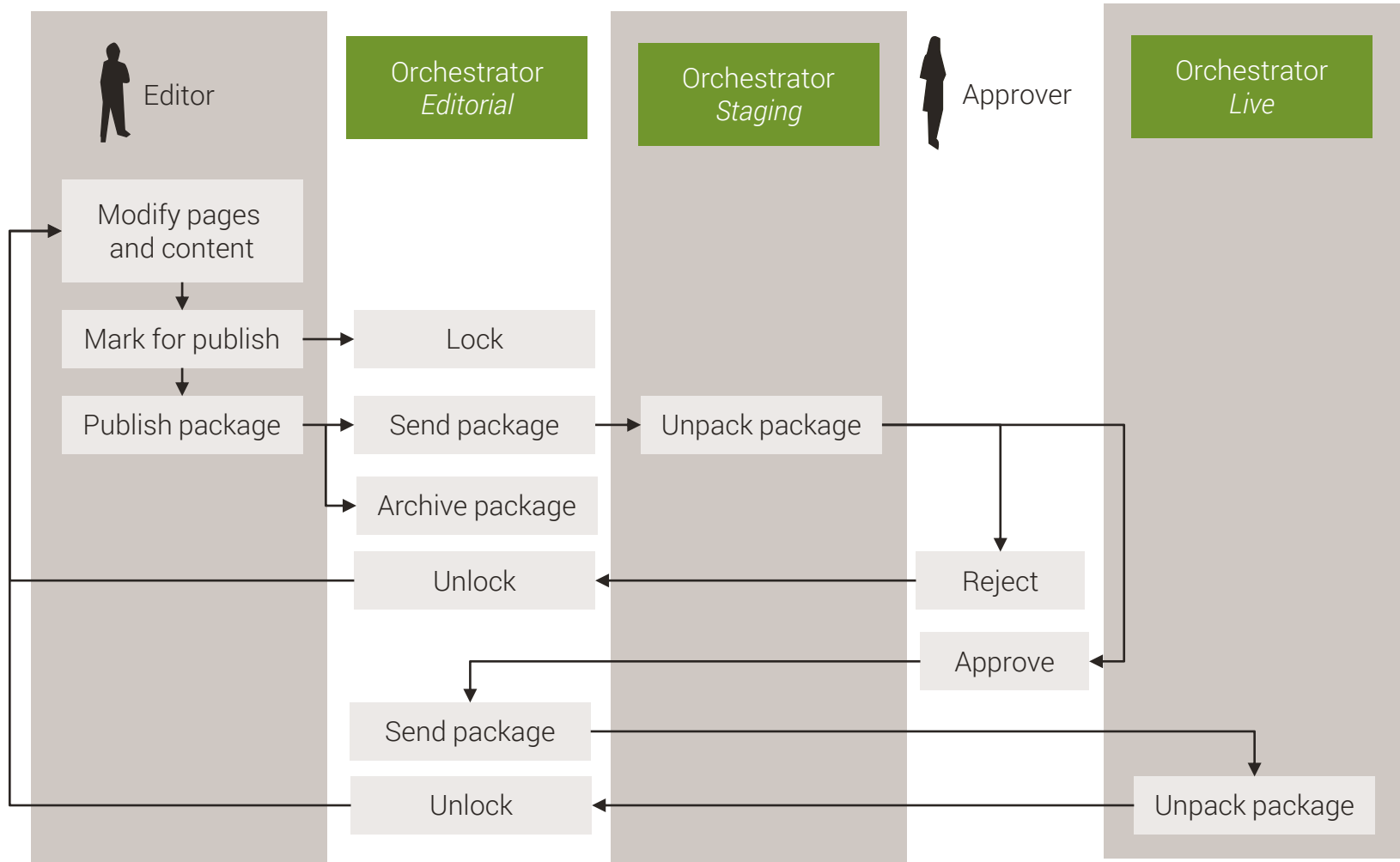
- CXP Portal supports publishing pages and content to different environments in a controlled manner:
 - **Editorial environment**
 - **Staging environment**
 - **Live environment**
- The Orchestrator module is in charge of the publishing process.
 - It must be present on all environments alongside CXP Portal Foundation.
 - Orchestrator locks data, sends and retrieves packages, manages approvals, and unlocks data.





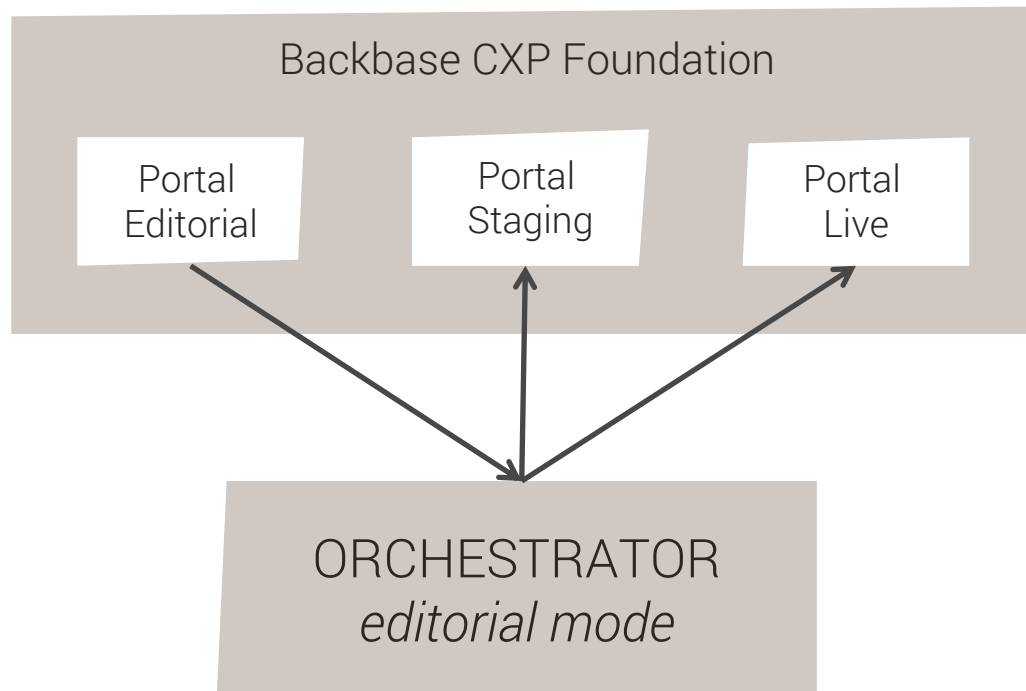
- The following items can be published:
 - Pages, their associated links and their children: containers, widgets, and ICE Content
 - Master pages
 - Reusable Content
- The following items are *not* published and need to be transferred via Import/Export:
 - Portals
 - Catalog items (Enterprise Catalog and Portal Catalog), except master pages
 - Templates
 - Template files and other resources (images, CSS, JavaScript files)
 - Configuration files

- A sequence of environments or portals an item has to be published to.
- Each publishing step may or may not require approval from an approver.
- Publishing chains can enforce approval at the following levels:
 - **Package level** — the approver must accept or reject the publish package as a whole.
 - **Item level** — the approver may accept or reject individual items (pages or Reusable Content items) within a publish package. The approver must accept each individual item before he can accept the package. Rejecting even one item in the package prevents the package from being published.



▪ Publishing Within an Environment

- publishing takes place across different portals instead of environments
- for testing and demo purposes (configurable in the Orchestrator configuration file)



- **Unpublishing**

- removing pages from a portal
- to unpublish a page, it needs to have its publishing chain completed

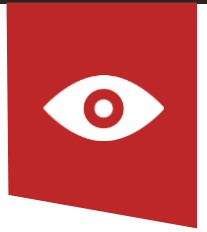
- **Republishing**

- after modifying already published pages

- **Scheduling**

- Scheduling incoming and outgoing publishing jobs
- The starting time and date of the publication can be set

- *Orchestrator* has two roles that you can map to groups in Backbase CXP Foundation:
 - Publisher – Allows users to submit publish requests.
 - Approver – Allows *Approvers* to approve or reject publish requests.
- Note that Backbase CXP enforces the **four-eye principle**: a user who submits a publish request may not approve it.

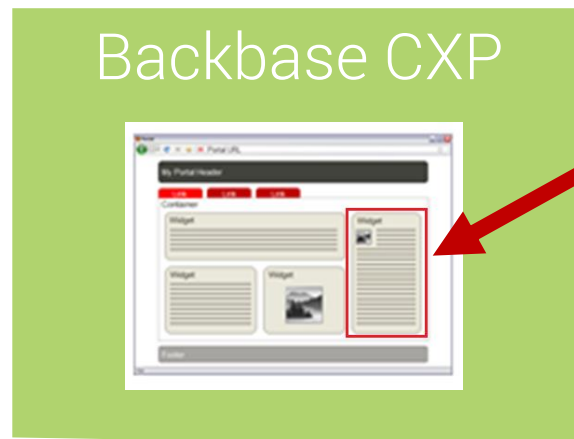


- First, let's look at the backbase.properties file.
- Now, let's see how publishing chains are configured.
- Login to CXP manager.
- Configure publisher and approver roles in CXP manager.
- Create portals required for self-publishing.
- Create a new page in the editorial portal.
- Add the page to a publish package and publish it.
- Login as an Approver.
- Approve the page.

Versioning

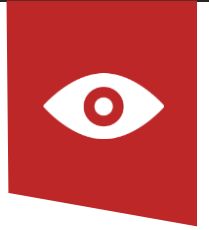
Publishing

- Versioning stores published items in Content Services and assigns them a version number.



- The following items can be versioned:
 - Pages and their children.
 - Content items.
 - Links.
- *Master pages* cannot be versioned.
- You can revert to older versions of a particular page or content item within CXP Manager.
- This feature is only available in editorial environments and requires Content Services. It is disabled by default and requires setting the *foundation.versioning.enabled* property to *true*.

```
foundation.environment.editorial=true  
foundation.versioning.enabled=true
```



- Login to CXP Manager.
- Create a new page and add a rich text widget.
- Add text to the ice widget and publish the page.
- Change the text in the ice widget and publish the page.
- Revert back to the first version of the page.
- Publish the page.
- Go to the Assets dashlet to see that the content of the rich text widget has been versioned as well.

Orchestrator Configuration

Publishing

- One can install Backbase CXP in an *editorial* or *staging/live* environments, each requiring different Publishing configurations.
- The Publishing configuration resides in the following files:
 - `backbase.properties` — contains configuration for the local environment and details about other Backbase CXP modules. Editorial environments require additional properties.
 - Publishing chains configuration — the file defining publishing chains. Only required for editorial environments.

```
#### General properties ####
orchestrator.name=EditorialOrchestrator
orchestrator.editorial=[true|false]

# The database Orchestrator is using; if not specified, defaults to h2
orchestrator.database=mysql

# Credentials to log in with on orchestrator
orchestrator.username=sys2sys
orchestrator.password=sys2sys

# Location where publishing files will be stored
orchestrator.filestorelocation=/path/to/publishing/store/location

# Thread pool configuration
orchestrator.threadpool.coresize=5
orchestrator.threadpool.maxsize=10
orchestrator.threadpool.queuesize=1000
orchestrator.auditing=false
```

```
# Portal host
orchestrator.portalhost.name=portalserver
orchestrator.portalhost.address=my.domain
orchestrator.portalhost.port=7777
orchestrator.portalhost.contextpath=portalserver
orchestrator.portalhost.username=sys2sys
orchestrator.portalhost.password=sys2sys

# Content host
orchestrator.contenthost.name=contentservices
orchestrator.contenthost.address=my.domain
orchestrator.contenthost.port=8082
orchestrator.contenthost.contextpath=contentservices/atom
orchestrator.contenthost.lockstrategy=nooplocking
orchestrator.contenthost.repositoryid=contentRepository
orchestrator.contenthost.username=admin
orchestrator.contenthost.password=admin
```

```
##### Orchestrators #####
orchestrator.orchestrators.names=EditorialOrchestrator,StagingOrchestrator,LiveOrchestrator

# Editorial
orchestrator.orchestrators.EditorialOrchestrator.address=my.domain
orchestrator.orchestrators.EditorialOrchestrator.port=8280
orchestrator.orchestrators.EditorialOrchestrator.contextpath=orchestrator
orchestrator.orchestrators.EditorialOrchestrator.username=sys2sys
orchestrator.orchestrators.EditorialOrchestrator.password=sys2sys

# Staging
orchestrator.orchestrators.StagingOrchestrator.address=my.domain
orchestrator.orchestrators.StagingOrchestrator.port=8380
orchestrator.orchestrators.StagingOrchestrator.contextpath=orchestrator
orchestrator.orchestrators.StagingOrchestrator.username=sys2sys
orchestrator.orchestrators.StagingOrchestrator.password=sys2sys

# Live
orchestrator.orchestrators.LiveOrchestrator.address=my.domain
orchestrator.orchestrators.LiveOrchestrator.port=8480
orchestrator.orchestrators.LiveOrchestrator.contextpath=orchestrator
orchestrator.orchestrators.LiveOrchestrator.username=sys2sys
orchestrator.orchestrators.LiveOrchestrator.password=sys2sys
```

```
# Allowed groups
orchestrator.approvalgroups=manager, approvers
orchestrator.publishgroups=manager, publishers

##### Import/Export #####
# Location where import files will be stored
orchestrator.import.location=/path/to/import/location
# Location where export files will be stored
orchestrator.export.location=/path/to/export/location
# Location where export log files will be stored
orchestrator.export.log.location=/path/to/export/log/location

# Publish chains
orchestrator.configuration.location=/path/to/publish-chains.xml
```



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Publish chains for this orchestrator -->
<publishChains>
  ...
  <publishChain>
    <publishChainName>Direct</publishChainName>
    <publishChainElements>
      <publishChainElement>
        <requiresApproval>false</requiresApproval>

        <targetOrchestrator>OrchestratorLive</targetOrchestrator>

        <updatePublishStates>true</updatePublishStates>
      </publishChainElement>
    </publishChainElements>
  </publishChain>
  ...
</publishChains>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Publish chains for this orchestrator -->
<publishChains>
  ...
  <publishChain>
    <publishChainName>Approval on Package</publishChainName>
    <publishChainElements>
      <publishChainElement>
        <requiresApproval>false</requiresApproval>

        <targetOrchestrator>OrchestratorStaging</targetOrchestrator>

        <publishChainElements>
          <publishChainElement>

            <targetOrchestrator>OrchestratorLive</targetOrchestrator>

            <updatePublishStates>true</updatePublishStates>
          </publishChainElement>
        </publishChainElements>

      </publishChainElement>
    </publishChainElements>
  </publishChain>
  ...
</publishChains>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Publish chains for this orchestrator -->
<publishChains>
  ...
  <publishChain>
    <publishChainName>Approval on Pages</publishChainName>
    <publishChainElements>
      <publishChainElement>
        <requiresApproval>true</requiresApproval>

        <targetOrchestrator>OrchestratorStaging</targetOrchestrator>

        <publishChainElements>
          <publishChainElement>

            <targetOrchestrator>OrchestratorLive</targetOrchestrator>

            <updatePublishStates>true</updatePublishStates>
          </publishChainElement>
        </publishChainElements>

      </publishChainElement>
    </publishChainElements>
  </publishChain>
  ...
</publishChains>
```

```
#### General properties ####
orchestrator.name=Orchestrator
orchestrator.editorial=true

...

##### Orchestrators #####
orchestrator.orchestrators.names=Orchestrator

# Editorial
orchestrator.orchestrators.Orchestrator.address=my.domain
orchestrator.orchestrators.Orchestrator.port=8280
orchestrator.orchestrators.Orchestrator.contextpath=orchestrator
orchestrator.orchestrators.Orchestrator.username=sys2sys
orchestrator.orchestrators.Orchestrator.password=sys2sys

...
```

```
<publishChain>
  <publishChainName>Self - Direct</publishChainName>
  <publishChainElements>
    <publishChainElement>
      <requiresApproval>false</requiresApproval>
      <targetOrchestrator>Orchestrator</targetOrchestrator>
      <portalMappings>
        <portalMapping>
          <sourcePortal>my-portal</sourcePortal>
          <targetPortal>my-portal-live</targetPortal>
        </portalMapping>
      </portalMappings>
      <updatePublishStates>true</updatePublishStates>
    </publishChainElement>
  </publishChainElements>
</publishChain>
```

```

<publishChain>
  <publishChainName>Self - Approval on Package</publishChainName>
  <publishChainElements>
    <publishChainElement>
      <requiresApproval>false</requiresApproval>
      <targetOrchestrator>Orchestrator</targetOrchestrator>
      <portalMappings>
        <portalMapping>
          <sourcePortal>my-portal</sourcePortal>
          <targetPortal>my-portal-staging</targetPortal>
        </portalMapping>
      </portalMappings>
    </publishChainElement>
    <publishChainElement>
      <targetOrchestrator>Orchestrator</targetOrchestrator>
      <portalMappings>
        <portalMapping>
          <sourcePortal>my-portal</sourcePortal>
          <targetPortal>my-portal-live</targetPortal>
        </portalMapping>
      </portalMappings>
      <updatePublishStates>true</updatePublishStates>
    </publishChainElement>
  </publishChainElements>
</publishChain>

```

```

<publishChain>
  <publishChainName>Self - Approval on Pages</publishChainName>
  <publishChainElements>
    <publishChainElement>
      <requiresApproval>true</requiresApproval>
      <targetOrchestrator>Orchestrator</targetOrchestrator>
      <portalMappings>
        <portalMapping>
          <sourcePortal>my-portal</sourcePortal>
          <targetPortal>my-portal-staging</targetPortal>
        </portalMapping>
      </portalMappings>
    </publishChainElement>
    <publishChainElement>
      <targetOrchestrator>Orchestrator</targetOrchestrator>
      <portalMappings>
        <portalMapping>
          <sourcePortal>my-portal</sourcePortal>
          <targetPortal>my-portal-live</targetPortal>
        </portalMapping>
      </portalMappings>
      <updatePublishStates>true</updatePublishStates>
    </publishChainElement>
  </publishChainElements>
</publishChain>

```



- Configure publishing for Edit, Staging, and Live portals on three separate portal servers.
- Setup and configure the orchestrator.
- Setup a 'Require Approval on Pages' publishing chain
- Setup the proxy (make portal aware of orchestrator).
- Deploy orchestrator, content services and portal server.
- Configure roles in portal manager.
- Test publishing.

For assistance, please follow the '**Configuring Publishing Across Multiple Environments**' and '**Install and Configure Orchestrator for Publishing**' articles on my.backbase.com

Thank you!

www.backbase.com
sales-eu@backbase.com

New York: +1 646 478 7538
Amsterdam: +31 20 465 8888