Core - Task # 67754

Status:	New	**Priority:**	Could have
Author:	Bernhard Kraft	**Category:**	File Abstraction Layer (FAL)
Created:	2015-06-25	**Assigned To:**	
Updated:	2015-07-01	**Due date:**	
TYPO3 Version:	7		
PHP Version:			
Complexity:			
Sprint Focus:			
Subject:	Cleanup "crop" implementation		

Description

For 7.3 an image "crop" functionality got added to the features of TYPO3.

The current implemntation is not really a neat one. Let me point out why.

First a rather strange construct had to get added at at least two different placed in the core:

1. line 106 of ImageViewHelper:
https://git.typo3.org/Packages/TYPO3.CMS.git/blob/refs/heads/TYPO3_7-3:/typo3/sysext/fluid/Classes/ViewHelpers/ImageViewHelper.php#l106

2. method "getImgResource" of "ContentObjectRenderer"
https://git.typo3.org/Packages/TYPO3.CMS.git/blob/refs/heads/TYPO3_7-3:/typo3/sysext/frontend/Classes/ContentObject/ContentObjectRenderer.php#l5477

Instead of this solution ContentObjectRenderer and Extbase\Service\Image\ImageService should have get changed to pass any FileReference along to the ImageProcessingService.

For this to work a few issues would have to get taken care of:

1. The "Resource\FileReference" class will require a "process" method similar to that of Resource\File

2. The "Resource\ProcessedFileRepository" will need to take care of both: Resource\File and Resource\FileReference

3. The "Resourcce\Processing\LocalCropScaleMaskHelper" class would have to get modified to take care of a FileReference and use the crop-property of the FileReference if not overruled by configuration.

When those changes get applied cropping would be implemented cleanly at one single location/class.

Related issues:

related to Core - Task # 54229: Refactor Processor Registry like the Extracto...	**On Hold**	**2014-03-11**

History

#1 - 2015-06-25 18:10 - Bernhard Kraft

Having this change integrated it would also be possible to add custom field to FileReferences and then get them handled from within any slot attaching to the "emitPreProcessFile" signal.

#2 - 2015-06-25 21:12 - Frans Saris

Maybe we can have another look at this when we proceed with #54229

#3 - 2015-07-01 13:26 - Bernhard Kraft

#54229 seems something REALLY required.

The current file processing mechanism is mostly unflexible.

For example it is almost impossible to process a file twice.

I used the "preProcess" slot of the FileProcessingService to hook in a custom method. Altough I already modified (overridden) the ImageService class of extbase to pass along the FileReference it seemed almost impossible to process a file two times. Reason:

When a file gets processed it will be put under the *processed* folder of its storage. When you then need to process the file again it is not possible to retrieve a "Resource\File" object for the processed file by any means (Except large scale xclassing). The ResourceFactory methods for retrieving a file will always return a "ProcessedFile" instance instead.

So for processing the file again I had to create a second Resource-Storage record and have it assigned a different *processed* folder (*alt_processed*). Then I could request my alternative Resource-Storage for the already processed file and got a Resource\File object back which I could then process again (and having the result put in _alt_processed).