TYPO3.Flow - Bug # 57450

Status:	New	**Priority:**	Should have
Author:	Alexander Berl	**Category:**	Validation
Created:	2014-03-31	**Assigned To:**	
Updated:	2014-04-30	**Due date:**	

PHP Version:	
Has patch:	No
Complexity:	
Affected Flow version: Git master	
Subject:	International E-Mail addresses (umlauts, etc.) are not validated correctly
Description	

Currently, Flow does not validate mail addresses that contain international special characters (non-ascii), such as german umlauts.

This is due to the PHP filter_var method not taking care of that possibility, referring to RFC 5322:
https://bugs.php.net/bug.php?id=65630&edit=3
This only deals with special chars in the domain part of an Email address, which should be handled with the IDN encoding (idn_to_ascii() on the domain part).

However, there is the more recent RFC 6531, which allows international addresses explicitly
http://tools.ietf.org/html/rfc6531#section-3.3

In detail, it allows the local part and the domain part of a mailbox address according to this definition:

The local part may be made up also of "UTF8-non-ascii" characters, i.e. all multibyte UTF8 characters (UTF8-2 / UTF8-3 / UTF8-4 according to http://tools.ietf.org/html/rfc3629#section-4) and extending from http://tools.ietf.org/html/rfc5321#section-4.1.2

The domain part may also be made up of U-Labels, where

> A "U-label" is an IDNA-valid string of Unicode characters, in
> Normalization Form C (NFC) and including at least one non-ASCII
> character, expressed in a standard Unicode Encoding Form (such as
> UTF-8).

I'm not completely sure about the consequences of this subtle difference in definition.
I see two possible solutions to deal with that within Flow:
   - fall back to regular expressions when filter_var fails OR non-ascii chars are detected in the address (Ugly, but actual support of RFC6531)
   - use idn_to_ascii on the whole address before giving it to filter_var (though I'm not sure it is formally correct to idn encode the local part, not RFC6531 conform)

Please provide your input on how to proceed, I will then take care of providing a changeset.

History

#1 - 2014-03-31 14:41 - Alexander Berl

Note: For idn_to_ascii to be usable, the PECL intl and idn extension needs to be installed. This might actually be a killer argument against it's usage as it might not be available on shared hosts.

Hence the converter method would need to be implemented in PHP (which is a lot of code or at least an external dependency, e.g. https://github.com/mabrahamde/php-idna-converter).

Alternatively, since we don't actually care for the exact IDN encoded string, all UTF8-non-ascii chars could just be stripped out before validation. This would be hacky at minimum.