

INSTALLER	<p>Language list first</p> <ul style="list-style-type: none"> Detects browser language Integrates with automated translation download Picking a language enables Language and Interface Translation
LANGUAGE	<p>Language module as base language service</p> <p>Simplified language config (no domain/prefix, etc)</p> <p>2 special instead of 1 (und: Not specified, zxx: Not applicable)</p> <p>English can be deleted (not even added if installed foreign)</p>
CONTENT LANGUAGE	<p>Content type language support more flexible (one screen setup)</p> <ul style="list-style-type: none"> Language selector field is freely positioned Same flexibility + new language support on other “content” Views to customize pages based on content language
BLOCK LANGUAGE	<ul style="list-style-type: none"> Blocks got language visibility More things are blocks
NEGOTIATION	<ul style="list-style-type: none"> Defaults to path prefix negotiation (80%), configuration there Added admin language option (move it up to first if used) Browser detection configurable + more accurate Fallback language configurable (don’t change site default)
USERS	<ul style="list-style-type: none"> Users have language preference + admin language if applicable Internally up to 3 different languages per user
NAMES	<p>Transliteration built-in for machine names</p>
INTERFACE TRANSLATION	<p>Interface translation (locale) module now only doing this</p> <ul style="list-style-type: none"> One central directory for .po files - staging/version control File import and automated update settings Import will not time out or eat up lots of memory Translation UI revamped (much quicker editing) Plural versions editable with specific UI Edited strings become custom, can be separately exported Default configuration translated English can be “translated to”
CONTENT TRANSLATION	<p>Field translation based module in core</p> <ul style="list-style-type: none"> Old content-copy translation not available anymore Configuration on the field and sub-field level (eg. image fields) Integrated with search indexing (in core) and API Node access is also language aware
CONFIG TRANSLATION	<ul style="list-style-type: none"> All new in core, replaces many of i18n’s features Not configurable, one flat permission for all translation Index page for all configuration Translation tabs, integrates with interface translation Be aware of what is content/configuration/interface
MENUS	<ul style="list-style-type: none"> Interface (built-in), configuration (views) and content (custom) Most tricky part is which one is which
APIs (picks)	<ul style="list-style-type: none"> JavaScript translation API unified with PHP API Simpler Entity API makes working with multilingual content easier More (HTML5) language markup for more accessibility/machines Configuration schema system to support configuration translation
WORKFLOW	<ul style="list-style-type: none"> Will be provided by contributed modules