Proposed Implementation:
Form Translation Via JavaScript

Engineering Creates Form Application
• text content in the application is stored in a JSON file. (en.json, es.json, etc.) • the text content for each application lives in a 'namespace' and is lazy loaded for that application to minimize file loading time • a common glossary of frequently used text is translated and available for use in any application to reduce repeated translation work • a shared translation library and corresponding translation utilities are used to interpolate strings and display the correct text for each language. • Shared application components accommodate translation functionality • Validation error messages are set up to be translated along with main text

Translators receive JSON files from engineering and translate content
• Once the translations are ready they are committed into the code repository • Revision of the text content translation require this process to be repeated

Translation file data is loaded via API call instead of via static JSON
• Eventually, it would be ideal to have translation data load for an application via an external API call from a content api like Drupal or a Translation Management Service Provider.

Application is launched live
• A language selector is displayed at the top of the form to allow changing of the language. • Form submissions are submitted with language selection data along with form data. • Form submission is potentially sent to a different location on the back end, based on language of submission

A Default Language Choice Is Automatically Detected
• The browser default language is used to changed the application language if possible. • If a user has chosen a different language via the application language selector that language choice is saved • A potential future addition to a user's profile may include a 'preferred language' that could be set and used instead