# UNIVERSITY Project Plan Asynchronize All The (Localization) Things!

### The Capstone Experience

**MICHIGAN STATE** 

#### Team Mozilla

Collin Wing Jack Smith Jack Song Jim Lennon Raza Haider

Department of Computer Science and Engineering Michigan State University

Fall 2018



From Students... ...to Professionals

# **Functional Specifications**

- Mozilla Firefox is an English (US) based web browser looking to expand in the global market.
  - Translate pages to the host countries native language not using DTD and properties files.
- This current system requires developers and localizers (translators) to edit both the DTD and .properties files.
  - Longer process for volunteer localizers.
  - Formatting a language can also cause issues with the markup language.
- The Fluent system will streamline this process.
  - Fluent is a localization framework.
  - Translations with genders and grammatical cases, date, time, and number formatting, plural categories, full bidirectionality support, custom formatters and easy-to-read syntax.
  - Stronger API between developers and localizers.
  - Platform detection, increased expressive power, and Runtime translations on client and server side.

The Capstone Experience

# **Design Specifications**

#### Localizers

- Easier to read syntax
- Easier logic construction
- Developers
  - Remove need for platform/plural checks at runtime
  - Easier to program
    - In XUL
    - o In JS

### End Users

- End goal remove series of steps and/or different build of Firefox to change languages
  - Step towards that goal

### Screen Mockup: Converting to Fluent

```
openDir = Open Directory
# LOCALIZATION NOTE (macOpenDir): This is the Mac-specific variant of openDir.
# This allows us to use the preferred"Finder" terminology on Mac.
macOpenDir = Show in Finder
# LOCALIZATION NOTE (winOpenDir2): This is the Windows-specific variant of
# openDir.
winOpenDir2 = Open Folder
```

```
profiles-opendir =
    { PLATFORM() ->
        [macos] Show in Finder
        [windows] Open Folder
        *[other] Open Directory
    }
}
```

# Screen Mockup: Cleaning up JS

let string = "openDir"; if (AppConstants.platform == "win") { string = "winOpenDir2"; } else if (AppConstants.platform == "macosx") { string = "macOpenDir";

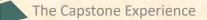
document.l10n.setAttributes(button, "profiles-opendir");

# Screen Mockup: Cleaning up XUL

<menuitem< td=""><td><pre>id="utils-installFromFile"</pre></td></menuitem<>	<pre>id="utils-installFromFile"</pre>
	<pre>label="&amp;installAddonFromFile.label"</pre>
	<pre>accesskey="&amp;installAddonFromFile.accesskey"</pre>
	<pre>command="cmd_installFromFile"/&gt;</pre>
<menuitem< td=""><td><pre>id="utils-installFromFile"</pre></td></menuitem<>	<pre>id="utils-installFromFile"</pre>
	<pre>data-l10n-id="install-addon-from-file"</pre>
	<pre>command="cmd_installFromFile"/&gt;</pre>

## Screen Mockup: End User Experience

About Profiles × +		-	đ	×
$\Theta \rightarrow C^{*} \Omega^{*}$ (1) about profiles	☺ ☆	III\	۵	
About Profiles This page helps you to manage your profiles. Each profile is a separate world which contains separate history, bookmarks, settings and add- ons. Create a New Profile Profile: Collin	Restart Restart with Add-ons Disabled, Restart normally			
Abaesist Pree/Indég × +		-	ø	×
(e) → C <sup>*</sup> (a) ① aboutprofiles	··· 🗟 🕁	hi/\	۵	
ÀĐƠỜŬŬŧ PřƠƠFTŀÉÉŞ Thiş þaagéé héélps yøøou tøø maanaagéé yøøuur prøø <i>r</i> iléés. Éaach prøøriléé is aa séépaaraatéé wøørlig which cøøntaains séépaaraatéé histøøry, bøøøøkinaarks, sééttings aang aadg-øøns. Grééaatéé aa Nééw Přøøriléé <b>Přóófīl-Éé: Collin</b>	<b>Řééştaařt</b> Řééştaařt with Ágg-dóng Dişaabl-éég Řééştaařt ndóřmaal-lý			

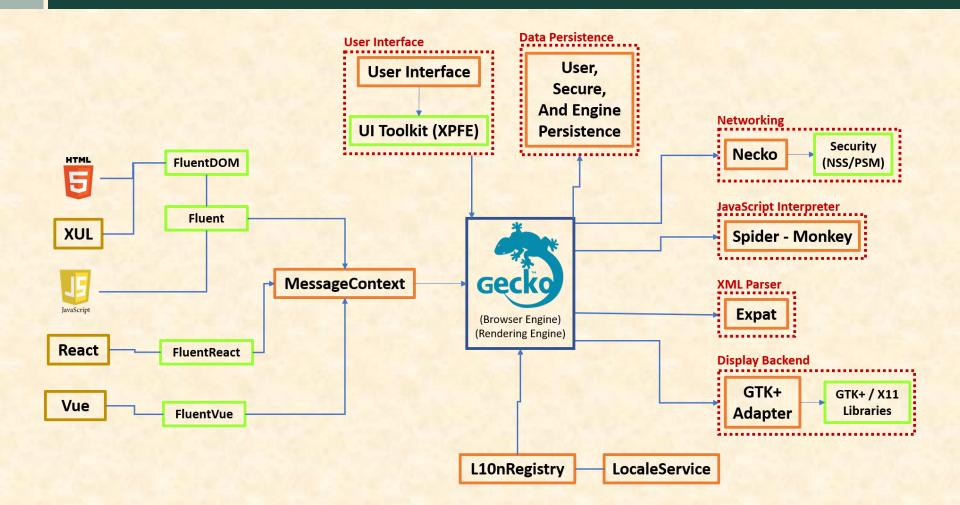


# **Technical Specifications**

#### Fluent Conversion

- Strings are moved from old-fashion files to FTL files manually.
- Localization of the browser achieved via modifying markup and Js files.
- String Migration
  - Strings are moved from old-fashion files to FTL files by migration recipes.
  - Migration test for comparison between manually and recipes-created FTL file.
- Integration
  - Mercurial, Bugzilla and Phabricator

# System Architecture



## System Components

- Software Platforms / Technologies
  - Mercurial
  - Phabricator
  - Bugzilla

## Risks

#### Localization Anti-Patterns

- Description: Translations may be violating design principles that we prefer for working with Fluent. Some of these strings will prove difficult to convert and migrate.
- Mitigation: We will identify potentially difficult strings ahead of time and determine if they will require more extensive work.

#### Limited functionality of Fluent

- Description: Fluent may not have enough expressive power to support converting some strings we encounter.
- Mitigation: We will work with the Fluent designers to add features which are on the critical path to full support. We will land patches directly if we can.

#### Migrating complex strings manually

- Description:Complex strings require laborious manual AST manipulation in the migration scripts. These prove difficult and time consuming.
- Mitigation: We will delegate migration issues to a particular team member who can gain a deep understanding of the system and efficiently solve problems

## **Questions?**

