Design Meeting with Microsoft
Week 5 – 31st/05/2018

Technology and implementation challenges are discussed with the Microsoft Tech team.

Problems discussed:

- Intellisense implementation method whether it's more effective to use ScriptCs or Roslyn as the REPL engine.
- Syntax Highlighting issues regarding the use of CSS and JavaScript files
- Error display issues regarding multi platform inconsistencies
- Azure deployment

#1 Intellisense implementation method whether it's more effective to use ScriptCs or Roslyn as the REPL engine.

After detailed discussion the use of Roslyn would be more effective as we can access to a larger scope of features however it would be more challenging that just utilizing ScriptCs.

Notes:

- Colin suggested we look at omnisharp (which is a "Visual Code plugin" kernel for C#) and suggested we might want to follow their thinking in terms of implementing our kernel's function (intellisense, errors, syntax highlighting) but those two are very different and omnisharp is coded in (typescript .ts).

Recommendation: Change to a Roslyn adaptation.

#2 Syntax Highlighting issues regarding the use of CSS and JavaScript files

1. Adaptation to Roslyn would be aid in implementing this feature.
2. Furthermore, the JavaScript file was redundant and most viable way to implement syntax highlighting is via CSS.

Notes:

- syntax highlighting is done by "webhighlighting" OR "webintellisense" file
- the .js "csharp.js" file is redundant for syntax highlighting in the ifsharp kernel but it is useful for us in icsharp
- mohammed showed (by sharing screen) that deleting the csharp.js file clearly affects our kernel because there was a difference in syntax highlighting (keywords went from their specified colours form the .js file (e.g. blue) to their default color values (e.g. green), proving that the csharp.js file was affecting our code unlike in the kernal of ifsharp where it was redundant and had no effect!
- Dino suggested we use codemirror "cm_config.mode" but Mohamed had already used it

#3 Error display issues regarding multi platform inconsistencies

1. Fix the IOPub communication to move the errors messages from the terminal to the notebook.
2. Furthermore, the ExecutionError that causes the kernel to die, check the buffer the ensure that is not storing the previous error messages and its refreshing when it needs to.

3. At this point, error display worked on Windows but not on other platforms this also required further investigation in the manner of how Mac OS user setup their development environment i.e. they may have missed a package or so therefore creating this issue seeing as Window Developers were able to see this function.

#4 Azure deployment

1. Dino provided detailed instruction that emulate the ifsharp repository deployment process onto Microsoft Azure.

Notes:

- register a custom kernal (Dino sent the link)
- start from a proof of concept by putting everything in a library (that starts up but is really slow, at least starts though)
- ifsharp is already on azure (Dino sent the link for us to refer to)
- Dino posted in slack the command to register a kernel