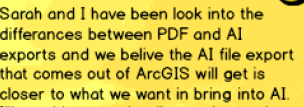
# First of all about numerated use cases.

1. Use case 1: Map layer tab. Everything is possible. Just a few questions about design.
   1. We make one toolbar for all panes in Tab panel (Found FS, Map layers, Output properties). And it is on bottom of this panel. Am I right? Sorry, not sure I understand you question. *I mean your design. But it’s not very important right now.*
   2. Now we delete selected layer using button with trash can (no drag and drop). Okay
   3. What will we move? Separate layer? Group layers corresponding to one feature service? Sorry, not sure I understand you question. *I mean layers of one feature service have to be in one group and we have to move them together keeping their order in the feature service. Or each layer is independed and can be moved separately. I think it more flexible if they come in as independent layers. It’s easier for user (and us if we chose to automatically change the layers) to group then ungroup.*
   4. Now zooming on the extent of a layer is done by double clicking on this layer. Do we remove this functionality? We can keep it, but “zoom button” I think is more AI like. A
2. Use case 2: Search tab. It’s OK too.
   1. Should we show here the content of the appropriate service (layers, as in your picture)? Yes, that what I was thinking
   2. If the answer on the previous question “yes” should we allow the user to select layers which will be added to the map? Yes, just like the existing workflow. The user searches, gets a set of result, and then they can add the layers they select to the map.*Existing workflow consider the structure of layers in the feature service as “solid” (cannot be changed). Now we add a feature service in the map with all its layers. Same as 1c above. You are asking if user can choose to only bring in some layers in the feature services, not all of them. This might be more useful, but I not sure for the prototype if it’s worth the level of effort? It would also effect the deisgn of the UI too, correct? Let’s start from simple scheme.*
3. Use case 3: Download/export. OK.
   1. You mean that there is a tool (service) that can export a Web map into an AI document? If so it would be great to get the link to such tool. No there is not automatic tool/service to do this. We could do it the same way we have it now, but instead of export to PDF, we export to AI file format (\*.AI). *I thought that you wrote about some existing tool.*
   2. 
   3. There is obviously basic functionally with ArcGIS to export a map to an AI format, but there is not an existing tool per say. Currently we construct a web map on-the-fly with the selected feature services and then call the “ConvertWebMapToMapDocument” then export the doc as PDF, correct? So would do the same, export change the last step to not export a PDF, but a AI file. *I’m not sure. Now we use “Export Web Map” tool for this purpose. But supports only following formats: PDF, PNG32, PNG8, JPG, GIF, EPS, SVG, SVGZ. The most native one for AI and PS is EPS. It support both vector and raster. I’ve read carefully the doc prepared by Sara. I have to say following. Such kind of export is available only for MXD documents but not for temporary Web Maps. Maybe we’ll find the way to feed Web map to this “tool” but it’ll help only to get file structure which we can get form PDF file using more sophisticated JSX script which we use now. If we decide to develop such JSX script then we can make even more sophisticated. Which maybe help to solve more problems than exporting directly to an AI file. Think about.*
4. Use case 4: Defining map extent.
   1. Tool tip during drawing the rectangle. Now we use the ArcGIS JS tool Draw to perform this operation. Unfortunately it has only one kind of events: draw-complete/draw-end. So if we want to display any info during draw process we have to develop our own tool. Okay let skip this part, I’ll update the UI mock up.
   2. Scale. The engine used in Adobe HTML5 extensions does not have any special restrictions for REST requests. So I think there is no problem.
5. Use case 5: Photoshop+. The Photoshop has the same possibilities for developing HTML5 extensions as AI does (maybe even more, as it is considered as more widespread product). Cool. Photoshop is their most popular product, this is the reason I wanted us to do support a workflow inside of it. Even more, there is the possibility to use one extension in several products. But it has to use different JSX code for communicating with host application. The extension can define the type of the host application. So there will be no problem Cool
6. Use case 6: Analysis. This is the subject to experiment. Yes, this is what I expected. The main problem is the result of analyzing. I’ve taken a look on the tool “Create Buffers”. It has the property “returnFeatureCollecton” which means (it is the quotation): When true, returns the result of analysis as feature collection and creates a feature service. Of course there is no problem to add a feature service in an exporting web map. But how long is the life of such service? Not sure I understand you question, “life of which service”? We can make a custom GP service and host it ourselves, let us know if we need to do that, or can you set up the GP tool/service yourself? That will allows us to define the output to match what we need. *I mean that the tool creates a feature sevice with results. But I did not find any information: is this service temporary or not?* I was envisioning this, and other services, to be permanent. I see this Esri Maps for Creative Cloud (EMACC) as SAAS driven application, at least for the most of the GIS workflows. We would host a number of services (like the CreateBuffers and the CalculateScale and more in the future) and the Adobe products would consume (and post process the result when needed). *OK, I’m not sure still. But let’s say “OK” for now.*It requires checking. Anyway it depends on the format of the result. So the exact answer can be got after defining the exact list of such operations.
7. Use case 7: Defining map extent manually. I don’t understand exactly how it will work, but if you define this more sharp I’m sure we can implement this. For example, I don’t understand how will the user define the center of the extent? Users of AI (and PS) think about dimensions in non-geograhic units (mm, in, px). So what we need to do is make this “conversion” from geographic units to page units for them. We have 3 existing services for POD that do some of these workflows.

<http://pod-ags-srv-1.esri.com:6080/arcgis/rest/services/mcs_pod/Calculators/GPServe>r

* 1. Calculate Extent, takes in a scale and page size…and returns a geographic area
  2. Calculate PageSize, takes a geographic area and scale…and returns the page size
  3. Calculate Scale, take a geographic and page size…return a scale

For this workflow, AI/PS users doesn’t know/care about map scale, they have a page size the need to fit into and know how much geographic area they want to cover (by drawing polygon on our map control). This is a “CalculateScale” service call; however we need to do extra work because the units are geographic/ground units. So to get the equivalent page units we need to multiply the unit conversion value by the scale (eg. cm to km at a scale of 50,000 is:

1cm = 0.00001km = 0.00001km x 50000 (scale) = 0.5km on a map). So I think the order of calculation is user can’t type in page size first, they must draw geographic area, then we get the scale, then we use this value to convert the distances to page units. Make sense?*I understand everything about geographic/projection coordinate system and document coordinate system. I don’t understand how users will use your the tool bar for creating map boundaries (extent) from scratch. I can imagine how it can be used to modify existing rectangle on the map. Yes, I think you are right. There is not a goodway to know before the polygon is drawn what the size (in page units would be). So it’s a two-step workflow for the user. They draw a box like they do with our workflow today, we get the scale of the map, then we enable the width and height boxes and show user what the page units are and they can then make adjustment to the size (in page units).*

1. Use case 8: Authorizing in AGOL. There are two ways to authorize: OAuth2 and token based. My suggest is to keep it the way we have it now (token based, still correct?). I just want an improved UI were I can log out and know want account I’m logged into. *OK*Token based has some restrictions but it does not require any additional means. OAuth2 is more flexible but there are some problems (solved for today but they can arise again):
   1. Adobe HTML5 extension engine uses a library which is incompatible with DoJo. They (Adobe) allow to switch off this feature now. But who can guarantee that it will last long?
   2. ArcGIS Online OAuth2 process does not allow so called trusted authorizing. So we have to run some additional web page which is used as a call-back web page. The authorizing will not work if this web page is dead by some reason. It’s not very robust.

# Now about unnumbered use cases.

* Geocoding. We can implement geocoding support. But what we should do with found places? Just mark on the map? How should we mark them? Should we include marked places in the exported map? I suggest we do it like we do with POD, return a list of places and when user click the one they want it get added to the map and we are zoomed to that location.*Just to focus. Nothing else? Add a graphic (dot/circle) with a label for the location and zoom to the location. Should those markers and labels be included in exporting map?*
* Cartographic tools. The Export Web Map tool allows using templates. And they can include everything what can be included in the ArcMap document. I’m not sure where such MXD doc has to be placed to be editable from Adobe HTML5 extension.

# Not mentioned features

* Reorganizing layers of AI (I don’t know what have to be done in PS yet). But was not implemented so I think we have to remember this use case. And more we have to investigate what happens in that case in Photoshop. Reordering layer in PS is not important at this point. The AI layer management is very important. As said, I’ll provide more details about this soon in another doc that Sarah and I are working on.
* Removing clipping property from AI layers. The same reason