Data Services —Map Services Grouped in a Configuration file To show in a Viewer→ **Context Profile** —Any Viewer that Can read this standard loads→ **Context Reader** —With Viewers on Same Config Standard, Makes easy for users with varying use Cases to use the viewer display that meets That use case need the best→ **Display** —If they make changes to the Mashup config, They need way to save that setup→ **Save Work** —The saved JSON config context profile needs to be portable to be able to share with others to use in other viewers→ **Share Work** —The portable config also needs to support a few popular In our CoUs Interoperable standards to make Usable in other viewer types→ **Reuse Work**

—————Flagship Viewer Impacts—————

—————Save Map As Module—————

🌎 **TNM 2.0 Basemap Services**

Palanterra x3 JSON (lite)

| Default Content | 1 URL added | >1 URL added |

TNM Viewer 2.0

Cascading Decisions
- Save As what Format
- option to just add what is checked or all
What Open in options are available

In USGS Module

Widget (Save As/Open In) (Option to S as or O in)

Would need to support config files on different HTTP Post (meaning not on our app server)

Current context ready only supports p=short name and assumes on our app server

If we could add in crossdomain check and checksum... shouldn't be an issue

TNM will not be using the Content Mgmt App and DB created in v18 and beyond - we'll make our own USGS Module, comment out the DB calls, and maintain our reader this way. This is more overhead because of the Px3 decisions to add a DB Server layer (which, per NGP EA, should be considered more of a design-time revision and CMS control, not run-time. Likely they won't change, and we'll have to maintain context reader changes (i.e. in v19 when they move to ESRI JSON base)

Context Reader   **TempJSON**

Post or passing an HTTP URL for the JSON As well as the calling Viewer URL

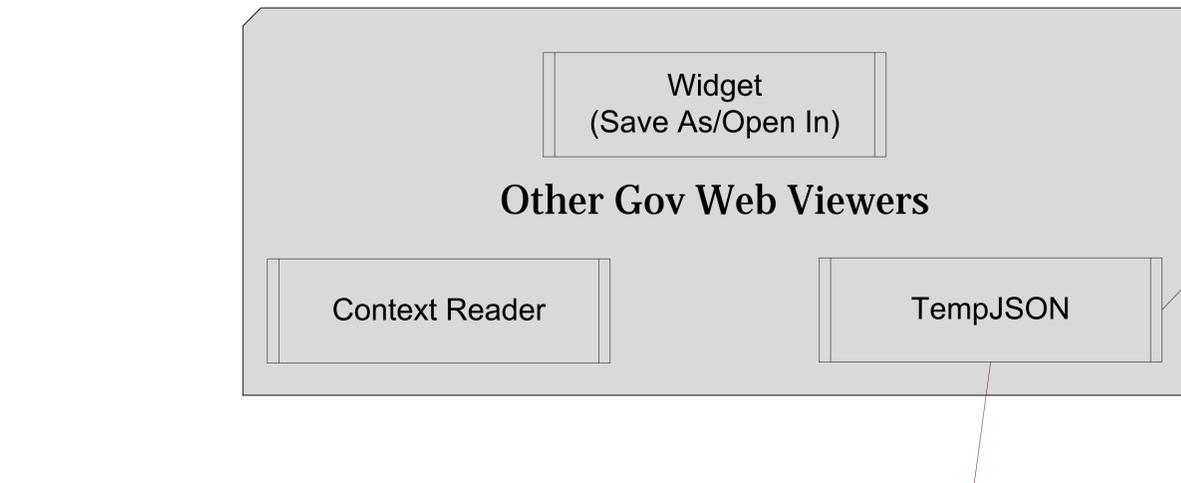The Calling Viewer will not always be the viewer.nationalmap.gov... Could be NGA, a state, FBI

This module needs to be a USGS Module in Px3

Cascading Decisions
- Save As what Format
- option to just add whats checked or all
What Open in options are available
TempJSON will need to read in whether user wanted Default + Changes Or Just Visible + Changes
Consider the Checksum is added here

Initially, v18 config schema will be your input into TempJSON that your add changes to and generate as output

V19 WILL CHANGE!!!! V19 will have the base based ESRI JSON and include Px3 JSON extensions (like service, service groups, intial extent... ALL will change to the ESRI REST Open Specification) - BUT the Px3 JSON extensions should remain the same or is irrelevant anyhow

Consider writing the TempJSON module as a way that could be copied and used by other JS Viewers to save them time

---

Other Gov Web Viewers

Widget (Save As/Open In)

Context Reader   **TempJSON**

Would need to follow same standard output that the broker can read

If Javascript, could borrow aspects of what is done in Px3

Post or passing an HTTP URL

Validator would need to
1. Security Validator (Per cross-app IT Security considerations – see NGTOC IT (Paul J./Wam) Then
2. JSON Lint technical validator Then
3. Schema validator Then
4. Business Rule/Values Exception validator (values, version control, etc.)

Validate

Save As/OpenIn

Broker/Router

Need single location that an web session calls

This broker will return/post the URL to the user, not the generators

Provide File to user (Browser "Save As" Popup comes up)

If User selects "Open In", do it

Calls appropriate generator

Temp File Location

—Below the Public Exposure—

2012 Priority Generators

Use whatever tech that supports business rules and consider future business rules like layer options as see fit

Can use for instance Px3 16.x, FBI v17, v18.8 or some ArcGIS.com print module – or something else that can read our services – KML, RSS, WMS, REST, etc.

Key for Rob:
- Use as much paper real estate as possible 8.5x11 - landscape
- Take what we learned from the Px3 16.x on handling the "byte" issues on WMS REST odd services (We won't be forcing the Service side to fix it)

Open In option: New Tab

Handle Byte issues

Open In option: New Tab

PDF Generator

JPG/PNG Generator

KML Generator

As long as the business rules are implemented
Handle the current technical "gotchas"
Don't regress, enhance agile

Px3 JSON Generator

ESRI JSON Generator

Assuming, no processing needed - take TempJSON input - done

Open In option: Calling Px3 Viewer p= parameter or TNM Viewer p= if not the calling viewer

Notice: this means the Px3 Viewer will need to handle calls that aren't on the app server

Maybe time support for doing this around receiving v19. If so, receive, pass thru, done with your code

Once the ESRI JSON v19 is implemented, technically, it's the same as the Px3 generator - so would only be 1 option (Px3/ESRI) between the 2

Open In option:Calling Viewer which may be any ESRI Viewer flavor (Flex, Px3, ESRI JS, Silverlight, iPad, iPhone, WindosPhone, ArcGIS.com, etc.) also support opening in ArcGIS.com or TNM Viewer

Will need some business rule review Possibly first release if business rule passes

Open In option: Google Maps, OpenLayers, Bing

Open In option: None

OGC Context Generator (ATOM/JSON)

Open In option: New Tab

Future: GeoPDF Generator (User MOD)

ArcGIS 10.1 MXD Generator

Audience is scientist - so whatever Scientist will be doing data editign in and needs those service mashup backgrounds, generate in that format – MXD, MSD, whatever

Explore ArcGIS.com's MXD Generator
- Is this a service call to arcgis.com hosts?
- Do we / Can we copy it down?
- Is this part of ESRI JS API library?
Need a ArcGIS 10.1 test instance to validate the MXD output
Ultimately goal: WE DON'T WANT TO WRITE THIS!