



WCS Work for OWS-4 – JPEG 2000 and WCS-T



Michael P. Gerlek
LizardTech
mpg@lizardtech.com

OGC / San Diego
11 Dec 2006



OWS-4 Projects

WCS support for geoprocessing workflows:

- support for adding coverages (“WCS-T”)
- support for JPEG 2000 (GMLJP2) output
- support for JPIP (streaming JP2) output
 - *jointly with ITT*

Today, an update on what we’ve done...



Transaction Support: WCS-T

The ability to add/modify/delete a WCS coverage

- Similar in spirit to the “WCS Harvest” proposal by Arliss...
 - ...but unbeknownst to OWS team
- small set of extensions to WCS 1.0 drafted and implemented by LizardTech
 - “OWS-4 IPR for WCS-T” (06-098)
- some changes and comments were fed back to the Harvest proposal document
 - “WCS CR for Add Transaction operation” (06-043)

WCS-T – Addition

- Insert a coverage offering into the server
 - via URL or POST

- AddCoverage operation
 - InputFormat
 - InputCRS
 - InputMetadataURL
 - InputDataURL

WCS-T – Update and Delete

- Modify existing coverage data
 - by id, via URL or POST
 - UpdateCoverageMetadata
 - UpdateCoverage

- Delete a coverage offering
 - by id
 - DeleteCoverage



WCS-T – Issues

- Partial updates?
 - we chose not to go there – can't upload just a small subscene
 - potentially requires re-encoding on server-side
 - *(we're particularly sensitive to JP2-encoded data...)*
- Versioning?
 - Wouldn't it be nice.
 - would require a big think at the WCS core level
- Atomicity and locking
 - if wanted to allow multiple operations per request
 - potentially entails commit/rollback, etc
 - (covered in the WCS 1.1 CR)



WCS + GMLJP2

The ability to return WCS coverage in JPEG 2000

- Small set of extensions to WCS 1.0
 - *“OWS-4 IPR for WCS Support for JPEG 2000” (06-128)*
- Not rocket science, just add a new return type
 - `image/jp2; subtype="gmljp2"`
- Plus some supporting stuff



The Supporting Stuff

And some additional support mechanisms

- DescribeCoverage
 - SupportsCompression
 - SupportedInterpolations (jp2wavelet)
 - SupportedJP2Profiles

- GetCoverage
 - GMLJP2Format
 - CompressionRatio
 - JP2Profile



WCS + GMLJP2 – Issues

- Server-side (re-)encoding issues
 - multi-file coverages not supported
 - reprojection not supported
 - needs a better appreciation of client-side expectations
 - e.g. one-shot/batch vs. “interactive” (JPIP)
- Encoding control
 - JP2 allows (way) many different legal encodings of a file
 - *all of which are absolutely critical, to someone or other...*
 - “profile” mechanism a reasonable hack
- Metadata / the GML boxes
 - when requesting a subscene: attempt to clip, or...?
 - Then again, this is a WCS server, not a WFS server....



WCS + JPIP

The ability to return WCS coverage as streamed JP2

- Small set of extensions to WCS 1.0
 - *also in “OWS-4 IPR for WCS Support for JPEG 2000” (06-128)*
- Again, pretty simple, just add a new return type...
 - `text/wcs-jpip-response`
- ...except for all the tricky bits.



The Tricky Bits

How to add JPIP as WCS output format?

- WCS not designed with “streaming” in mind
 - (ditto WMS and WFS...)
- we don’t want to change the basic WCS c/s model
- and we *really* don’t want to change the JPIP spec

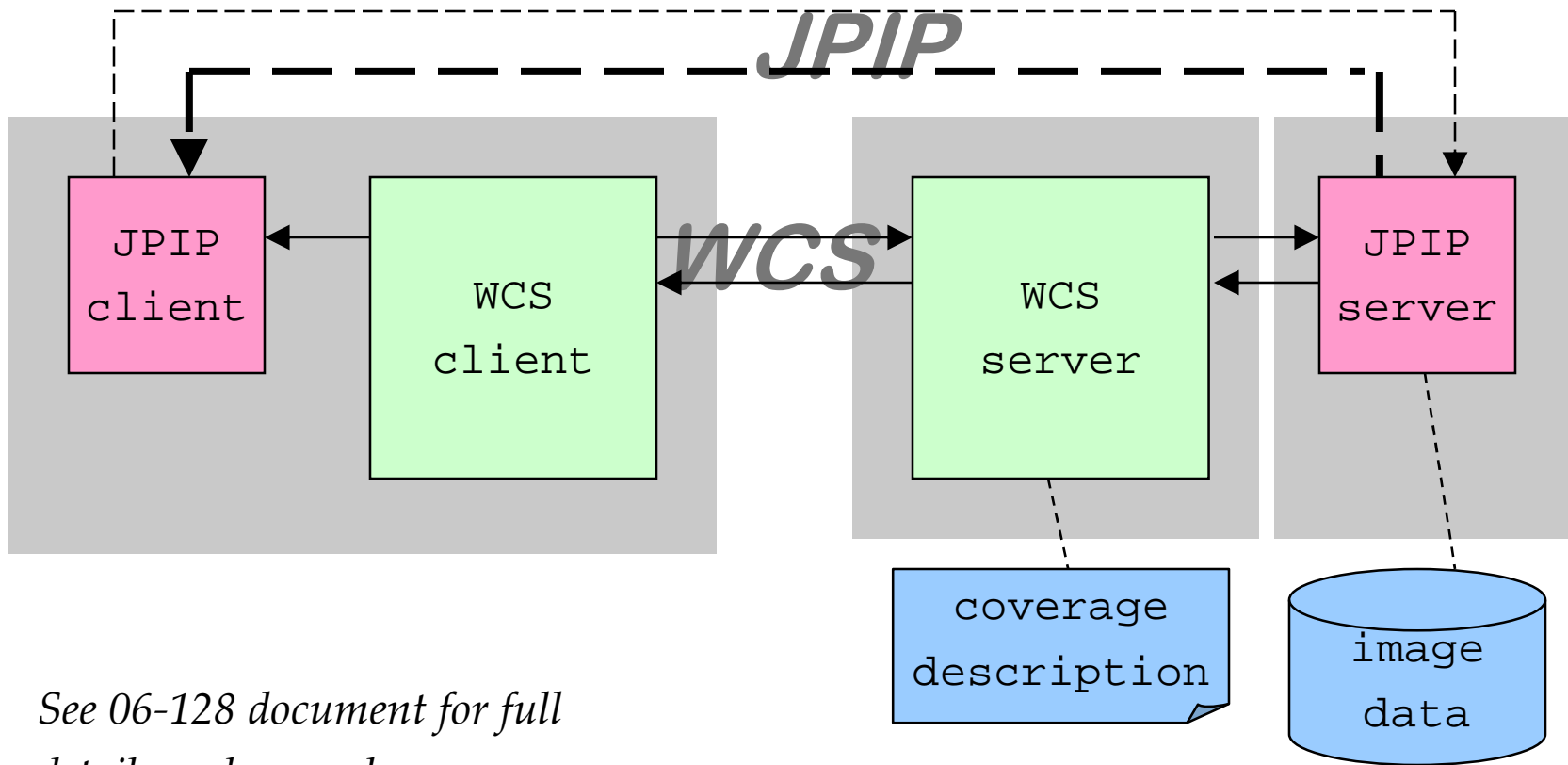
So: format is a “JPIP Response Document”

- the client side is given a handle (URL) to the appropriate JPIP server, image name, etc
- (JPIP stream is encoded via GMLJP2, of course)

From there, the client passes the handle to the JPIP decoder, and life proceeds on as normal for the JPIP world

- client requests, caching, asynchronicity, ...
- WCS client knows the geo bbox from the initial response, etc

WCS+JPIP Architecture



See 06-128 document for full details and examples.



Now What?

- WCS-T
 - feedback, WCS 1.1 CR evolved
 - no further work needed on the 1.0 extensions
- WCS with GMLJP2 & JPIP
 - 06-128 material should be recast as one or more change requests against WCS 1.1
 - present as application profiles?
- Whither the GMLJP2 RWG?
 - was led by Martin Kyle (ex-Galdos) and Sean Forde (ex-LizardTech)
 - (at least) two outstanding issues to be addressed



Thank You.

