# Distribution among GISCs via HTTP/AMQP

## A Practical Option for Distribution among GISCS

The [GISC Network and Application workflow document](http://wis.wmo.int/doc=3027) describes complications involved in the synchronization of caches amongst the GISCs, where the initial methods involving large numbers of inter-GISC channels to be configured. The complexity of these relationships can be substantially reduced by the standardization of the links and the use of protocols & methods not so far considered.

The proposal is to expose a standard file tree as an http resource, and provide a timely AMQP notification service to enable GISCS to replicate the standard tree, pulling the products themselves via HTTP.

### Components of the Option

1. MetPX

[MetPX](http://metpx.sf.net/) is a GPL licensed open source TCP/IP-only WMO MSS which began in 2004, and has been the national link for Canada to the GTS since 2007. It is distributed as a debian/ubuntu linux package. MetPX has been adopted by the civil aviation authority in Canada (NAVCANADA) and is also used in the regional weather offices of Environment Canada for data switching. At the CMC, the switching cluster distributes roughly four terabytes a day of products, including all bulletins, numerical products (such as for participation in TIGGE and NAEFS), RADAR data and products, imagery, remote sensing data, etc... acquired or distributed by the CMC.

MetPX can accept data via “receivers” which exist for GTS sockets or FTP, or SFTP (likely preferred mechanism), and then “switch” and deliver to programmed destinations, perform arbitrary transformations using relatively simple python scripting. A strategy of the application is to use the maximum number of other available tools, rather than re-implementing internally. For example, A entirely automated product driven by MetPX is: dd.weather.gc.ca a “datamart” of raw meteorological information made available to on behalf of Environment Canada. The components used to drive the datamart as as follows:

1. [OpenSSH](http://www.openssh.org/)

There is no SSH protocol driver for reception of products. The standard openssh is used and the “receiver” simply monitors a local reception directory for products.

1. [Apache](http://httpd.apache.org/)/HTTP.

Apache is among the most popular open source web servers. In this case, it is configured to show an raw directory Index, so that the products are made available via http with a minimal stack (apache alone is involved for product retrieval.) There is no specific apache functionality required, and any other web server could replace it in the deployment.

1. [AMQP](http://www.amqp.org/)/[RabbitMQ](http://www.rabbitmq.com/)

Advanced Message Queueing Protocol 0.9, is a message queuing and transfer protocol popular in the financial world as an open replacement/complement for proprietary mechanisms (such as MQ from IBM) for message exchanges. It provides some mechanisms that could implement services similar to traditional WMO sockets, but that is not how it is used here. RabbitMQ is a widely adopted free implementation of an AMQP broker, which is roughly analogous to an MSS.

There are libraries implementing amqp clients available in all major programming languages, and this service uses only a small core subset of functionality that is very easily implemented.

1. [DD Subscribe](http://sourceforge.net/projects/metpx/files/sarracenia/)

a sample client script of a few hundred lines of python demonstrating a notification client capable of reproducing the source tree on the client system.

## Integration

When a message (received by MetPX) is switched for delivery to a datamart, it is delivered and then a corresponding AMQP notification message is built. The message is then posted to the broker.

AMQP is a high speed messaging system optimized for small messages. In this plan, AMQP is used only to send notifications of products, rather than the products themselves, effectively reducing message sizes and opening the way for many http optimizations. A user of the system subscribes to the broker to receive AMQP notifications, filters for the messages of interest, and issues HTTP GET requests for the messages of interest. The provided dd\_subscribe.py client implements a client capable of reproducing the source tree on a subscribing system (much like rsync.)

In the context of the application workflow document, GISCS could uplink via SFTP to a cloud server running the described stack, and subscribe to a AMQP service from the same server, either using dd\_subscribe or a re-implementation in a preferred programming language, to reproduce the directory hierarchy of the 24-hour cache. AMQP naturally includes queues to remain active in case of disconnection, so when issues arise, the subscribtion service would queue up notifications for the GISC. The protocol is real-time and non-poll oriented, making it an excellent choice for timeliness. The protocol also includes application level acknowledgements, so that notifications are not discarded until the subscriber has assuredly received them.

Only a single (or dual) instance of an MSS is required in the cloud server for such an implementation. That configuration merely accepts data from all GISCS and posts it to the notification services. The GISCS would implement a much simpler client, which replicates the cache locally, and could be implemented in technologies matching local preferences. The configuration of these clients would be very straightforward.

There are other configurations that could be considered, such as each GISC implementing an AMQP advertised feed for the others to subscribe to, but it is probably best at this point to have a few countries participate in a technology demonstrator to investigate viability and discover (and hopefully address) concerns.

This method of data distribution is relatively new and has not been battle hardened by years of operations, so it is not appropriate to propose it immediately as the solution for distribution. More experience of more countries with the technologies are needed before it could be considered an option.

A minimum investment method of a technology trial could be done in various ways:

Method 1: an 'gisc-cache' tree could be added to the existing dd.weather.gc.ca service, some GISCS could subscribe to the service using the provided client initially for evaluation purposes. The service could be in place in a few weeks. Initially, this would be based on Canada's existing reception of GTS data from NWS Washington, but this would yield poor timing, so a follow-on step would be to configure FTP or SFTP reception for direct delivery to the gisc-cache hierarchy to gather better information about timeliness.

The configuration could be transferred to a cloud implementation if the technological approach is accepted. Using this method, the decision on the cloud server is deferred, and the option left open to

Method 2: the same experiments could be performed using the full stack implemented in a proposed cloud server. Such a solution would likely begin initially with a complete feed from one GISC, and then gradually expand to multiple GISCS over time, similar to method 1. There would likely be a slower ramp up to being able to gather feedback, but the implementation would be “complete” more quickly.

While Canada would be willing to provide participate in any such efforts should the community find this option worth investigating.

…

## References

[1] [http://dd.weather.gc.ca](http://dd.weather.gc.ca/) – Sample implementation.

[2] [http://metpx.sf.net](http://metpx.sf.net/) – WMO TCP/IP GTS MSS implementation.

[3] [http://www.rabbitmq.com](http://www.rabbitmq.com/) – most popular AMQP broker, very widely used.

[4] [http://httpd.apache.org](http://httpd.apache.org/) – popular web server, used in existing implementation.

[5] <http://www.sensorsandsystems.com/news/top-stories/corporate-news/32763-environment-canada-s-weather-datamart-experiments-with-real-time-notification-of-updates.html> Press report of this new method.

## Recommended Text

Recommend that several GISCS participate in a trial of proposed HTTP/AMQP using the existing Canadian implementation to gain experience with the technology for future evaluation.

--------------------