WEATHER CLIMATE WATER
TEMPS CLIMAT EAU

GISC Tokyo cloud project

KANNO Yoshiaki/ OZEKI Ren
Japan Meteorological Agency (JMA)

WMO OMM
World Meteorological Organization
Organisation météorologique mondiale

Project objectives

- Enable smooth migration from GTS to cloud based “shard platform” data exchange system
- Get necessary skills and knowledge for the new technology (especially for MQPs) together with AOR countries
- Demonstrate the prototype of MQPs system to AOR countries (A)
- Demonstrate the more conventional cloud storage direct download method for countries preferring a more gradual evolution (B)
- Consider ways of immediately exchanging time critical data such as warnings and advisories
- Consider interoperability, efficiency, sustainability, and techniques toward efficient migration for LDCs and other parties

WMO OMM

Project team

- KANNO Yoshiaki: project management
- OZEKI Ren: in charge of A: MQPs system
- NOYORI Tatsuya: in charge of B: Cloud storage direct downloading

And other GISC Tokyo members (total 7 people)

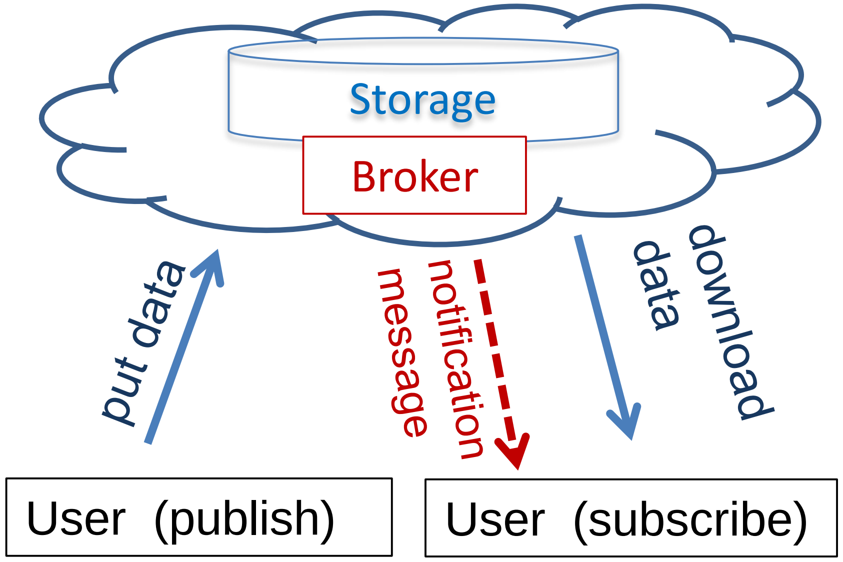
WMO OMM

Project plan

A: MQPs system

・Develop prototype of communication infrastructure

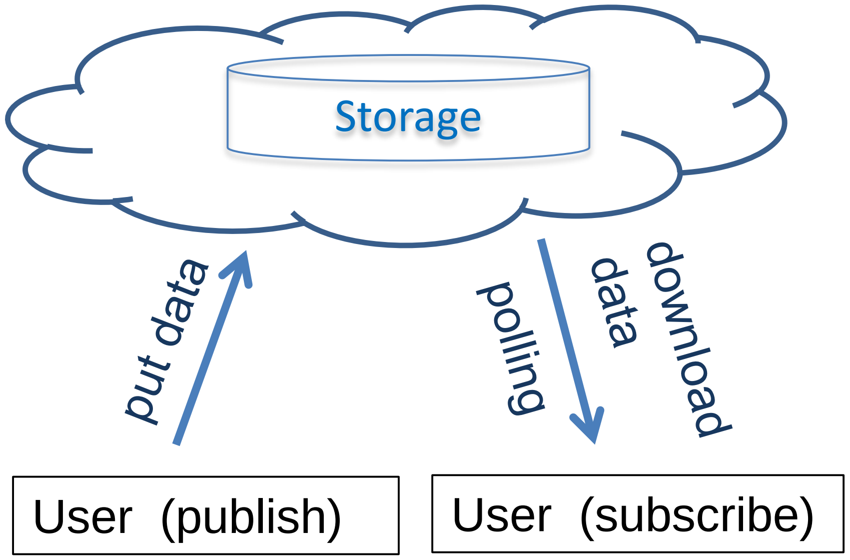
(currently using on-premise machine)



B: Cloud storage direct download

・Develop open source tools to utilize cloud storage

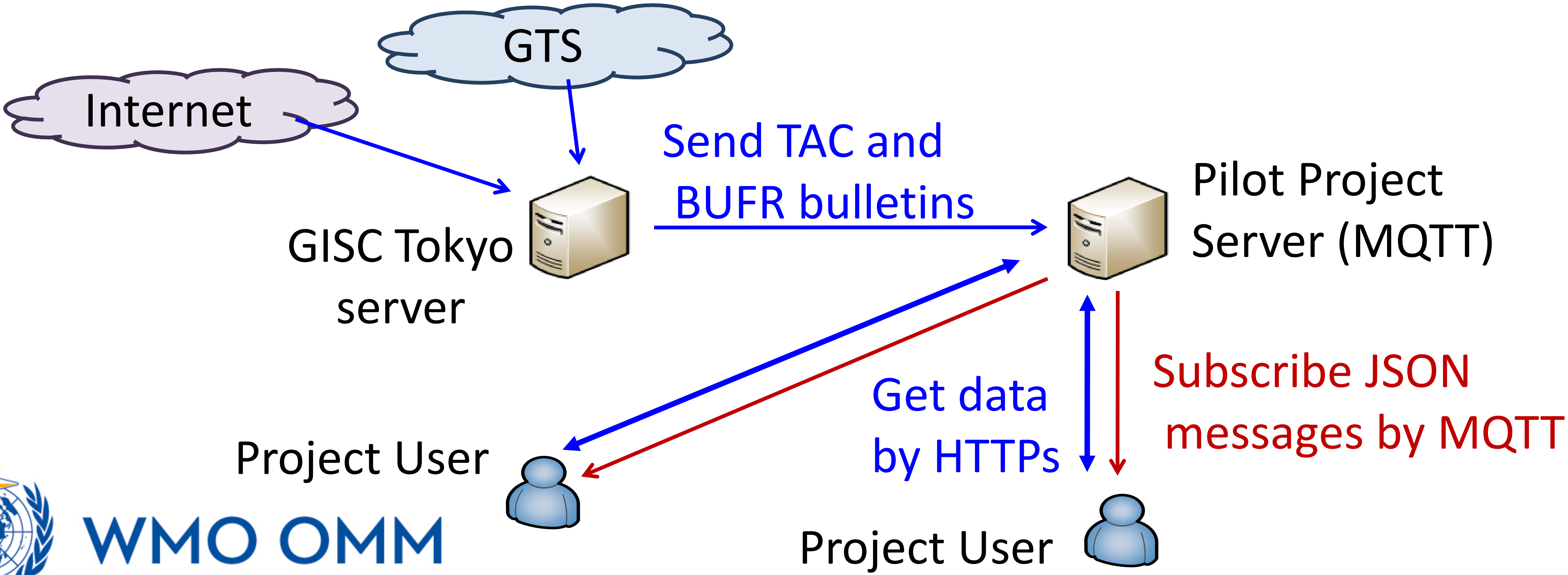
(currently using cloud on trial base)



WMO OMM

Project Overview - A: MQPs-1

- Developed initial MQP system in 2019



WMO OMM

Project Overview - A: MQPs-2

- Performance test of pub/sub architecture and implement related improvement
  - In 2020, testing of transmission for 3,000 files simultaneously (as per NWP model outputs) using MQTT revealed significant delays
  - "prefetch" (using AMQP) showed potential for significant delay reduction (details in later slide)
- Consider appropriate topic tree structure, in line with relevant WMO community

WMO OMM

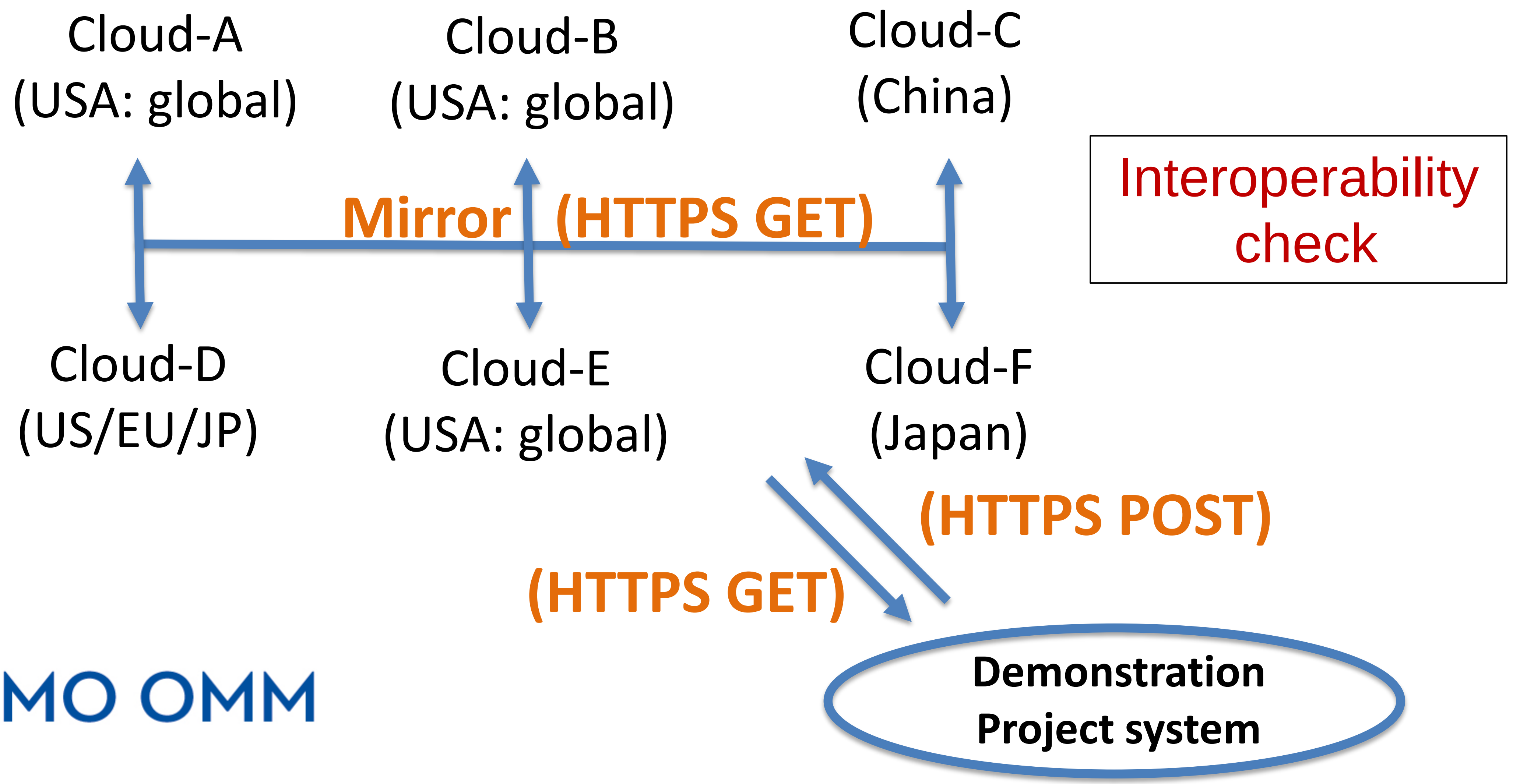
Project overview - B: Cloud storage download- 1

- Develop tools to utilize cloud storage
  - Enable user-scheduled downloading, in line with data categories
  - Make the tools open, as open-source

- Ensure interoperability among various cloud storage providers
  - Post data to cloud storage with REST (HTTPS)
  - Get data from cloud storage with REST (HTTPS)
  - Enable mirroring among cloud storages with REST (HTTPS)

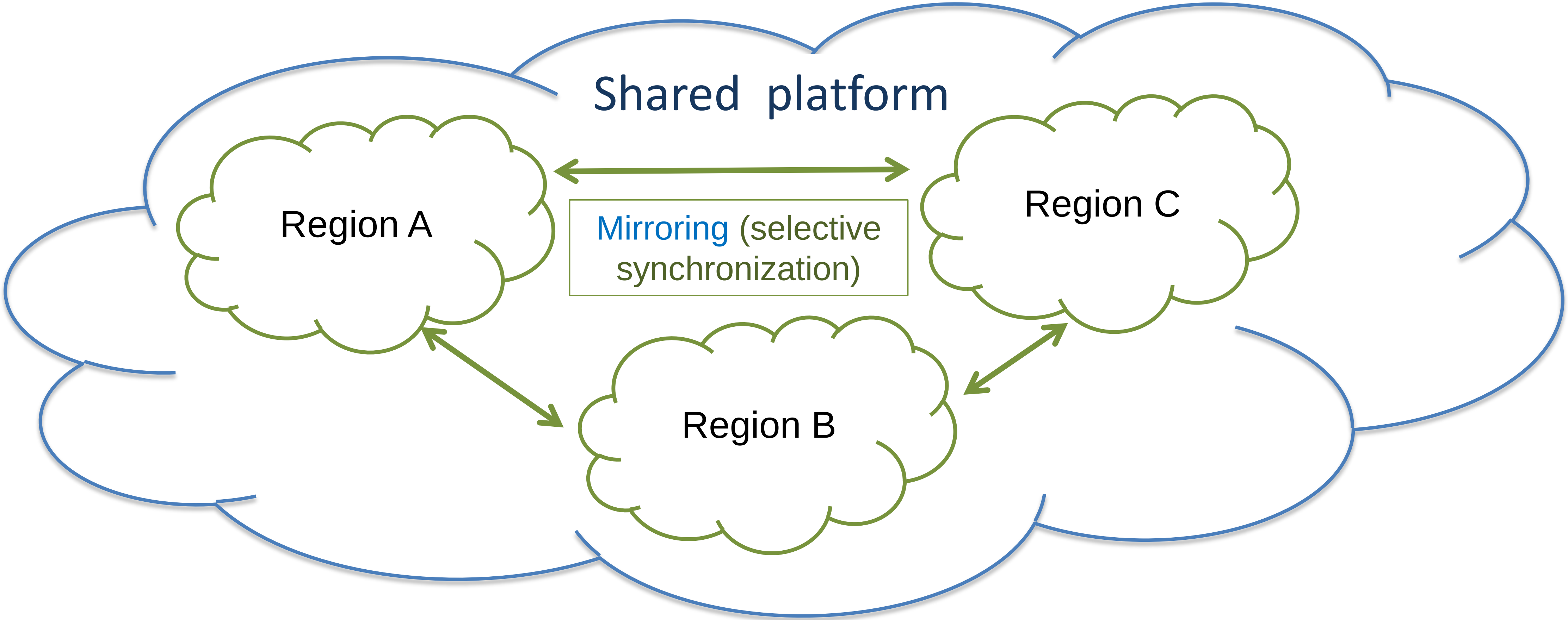
WMO OMM

Project Overview- B: Cloud storage - 2



WMO OMM

Why mirroring? - B: Cloud storage



Mirroring may be introduced in relation to federated solutions for shared platform

WMO OMM

WIS 2 Principles in the project

Principle 1: WIS 2.0 adopts Web technologies and leverages industry best practices and open standards

The project uses on

- MQPs: MQTT/AMQP and HTTPS
- Cloud storage: HTTPS and REST

Principle 2: WIS 2.0 uses Uniform Resource Locators (URL) to identify resources

The project uses URLs both on MQPs and Cloud storage

Principle 3: WIS 2.0 prioritizes use of public telecommunications networks (i.e., Internet) when publishing digital resources

The project uses Internet both on MQPs and Cloud storage

WMO OMM

WIS 2 Principles in the project

Principle 6: WIS 2.0 will add open standard messaging protocols that use the publish-subscribe message pattern to the list of data exchange mechanisms approved for use within WIS and GTS

- MQPs: use MQTT/AMQP to exchange JSON message
- Cloud storage: doesn’t use messaging. Everything are implemented with cloud storage over HTTPS.

Principle 7: WIS 2.0 will require all services that provide real-time distribution of messages to cache/store the messages for a minimum of 24-hours, and allow users to request cached messages for download

- MQPs: Real-time assured messages distribution and 24-hours data holding
- Cloud storage: Real-time data distribution and 24-hours data caching

WMO OMM

WIS 2 Principles in the project

Principle 8: WIS 2.0 will adopt direct data exchange between provider and consumer

Project enables direct data exchange both on MQPs and cloud storage

Principle 9: WIS 2.0 will phase out the use of routing tables and bulletin headers

Project doesn’t use routing tables both on MQPs and cloud storage

All direct project principles are outlined above, but harmony with other efforts will also be maintained.

WMO OMM

Project data/metadata standards

Data standards:

- This project focuses on GTS-going data. Started with BUFR & TAC. GRIB is to be included.

Protocol standards and related standards:

- Project uses MQTT(v3.1), AMQP(0-9-1) , HTTPS, JSON, and REST.

Metadata standards:

- Project doesn't directly focus on metadata at this stage. Future subject.

WMO OMM

Data exchange

- Project focuses on MQPs and direct cloud download for data exchange mechanism

- Project involves initial experimental data exchange between GISC Tokyo and GISC Offenbach.

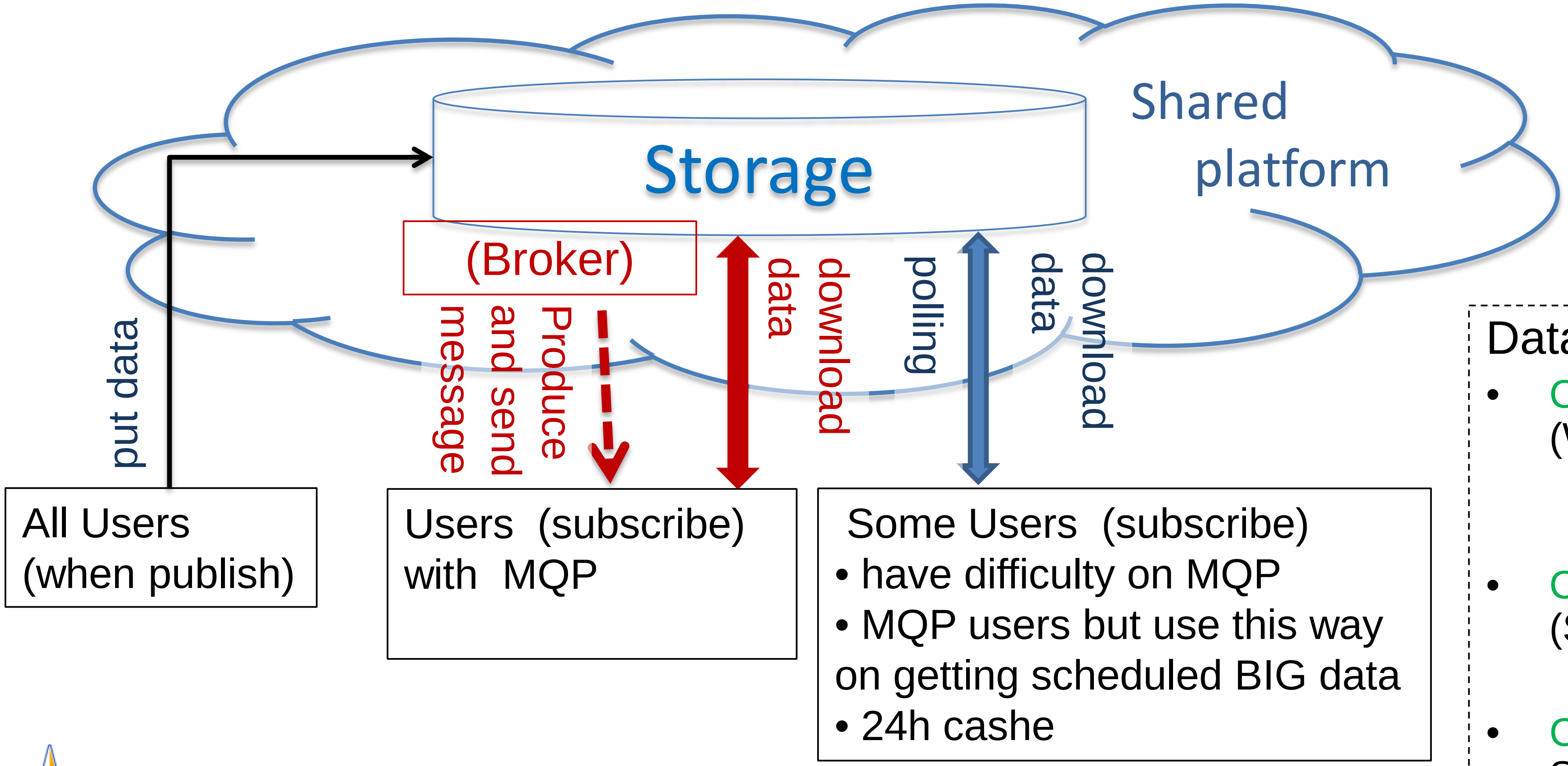
- GISC Tokyo plans to invite other countries to join the experimental data exchange.
  - Invite GISC Tokyo AoR countries and collaborating GISCs, via the online GISC Tokyo WIS workshop, to be held in late 2021.
  - Collect feedback for further improvement, and monitor system load changes.

WMO OMM

Input to WIS2

- The key MQP technology expected to replace GTS has been used by few NMHSs to date. Some countries (esp. LDCs) may struggle with related incorporation into operational systems.
- This project found some different nature between MQPs and direct cloud-storage downloading. (details later)
- MQPs system is very real-time in nature while direct downloading is simple and the hurdles to migrate maybe low.
- Big data (e.g. NWP, satellite) is likely to be suitable for simple downloading, as the data is to be produced on schedule.
- As an alternative or transitional method, combined approaches may be an option for efficient migration from GTS to WIS2.
- The next slide illustrates such a combined approach.

Potential combined approach as an alternative or transitional method



A: MQP system
B: Direct downloading (polling)

Data Category
- Cat-1: Time Critical data (Warning, Advisory, …)
  - Non-scheduled data, within 2-minute
- Cat-2: Conventional data (SYNOP, TEMP, …)
  - within 15-minute
- Cat-3: BIG size data (NWP, Satellite, …)
  - Scheduled data

WMO OMM

Speaker change to Ren

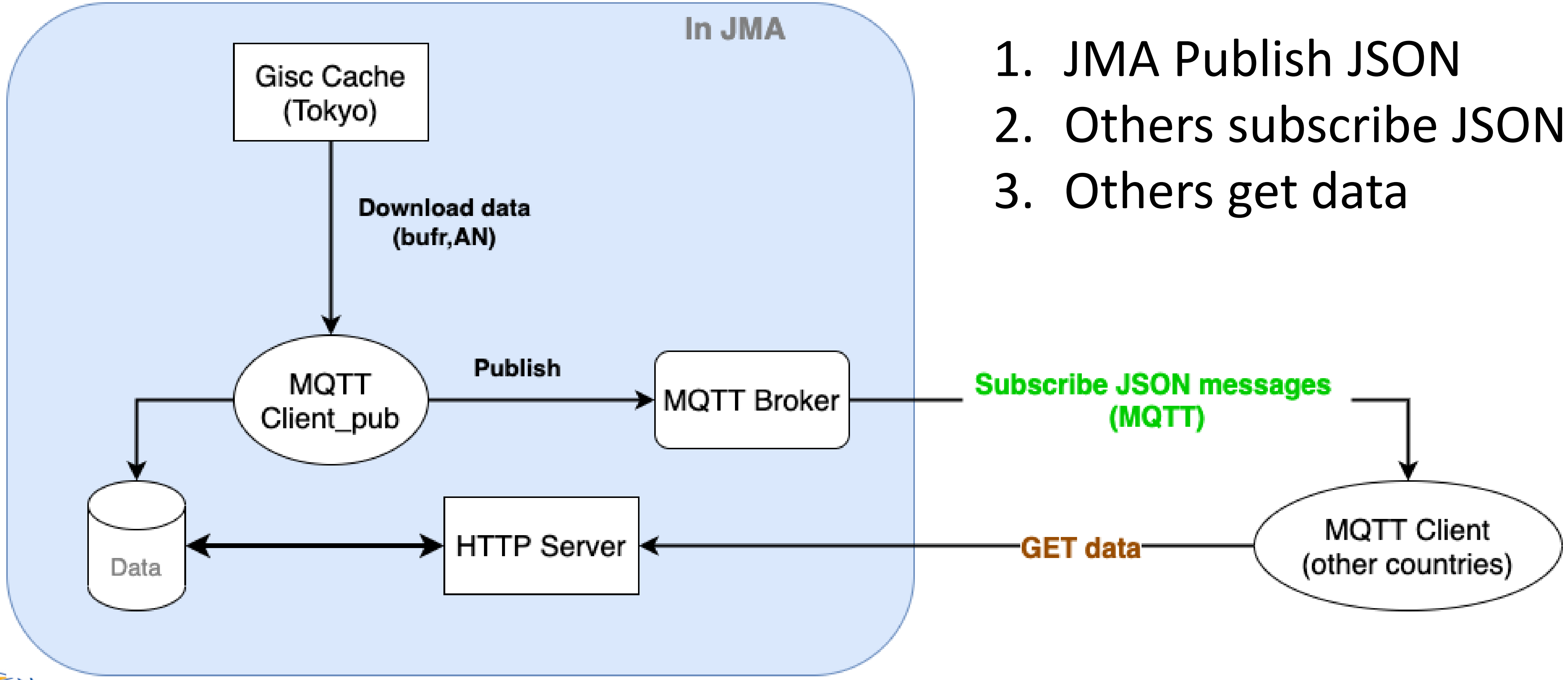
- Experiment with MQPs
- Compare MQPs system with simple cloud downloading

WMO OMM

Experiment with MQPs

WMO OMM

Pub/sub architecture



1. JMA Publish JSON
2. Others subscribe JSON
3. Others get data

WMO OMM

What subscriber is doing to get data

- Subscriber get JSON messages
- Extract URL from JSON
- GET request (to URL)


```
{
  "pubTime": "yyyymmddhhmmss",
  "baseURL": "https://example.com/",
  "relPath": "dir1/dir2/example.dat",
  "size": "xxxxxxxxxx",
  "integrity": {
    "method": "SHA512",
    "value": "zzzzzzzzzzzzzzzzzzzz(base64)"
  },
  "content": "",
  "signature": "Japan Meteorological Agency"
}
```


Message_template.json

WMO OMM

How to get messages you want

If you need JMA and Synoptic data

Subscribe following Topic("+" means all) :

WIS/+/+/RJTD/+/Synoptic/+

Topic Structure

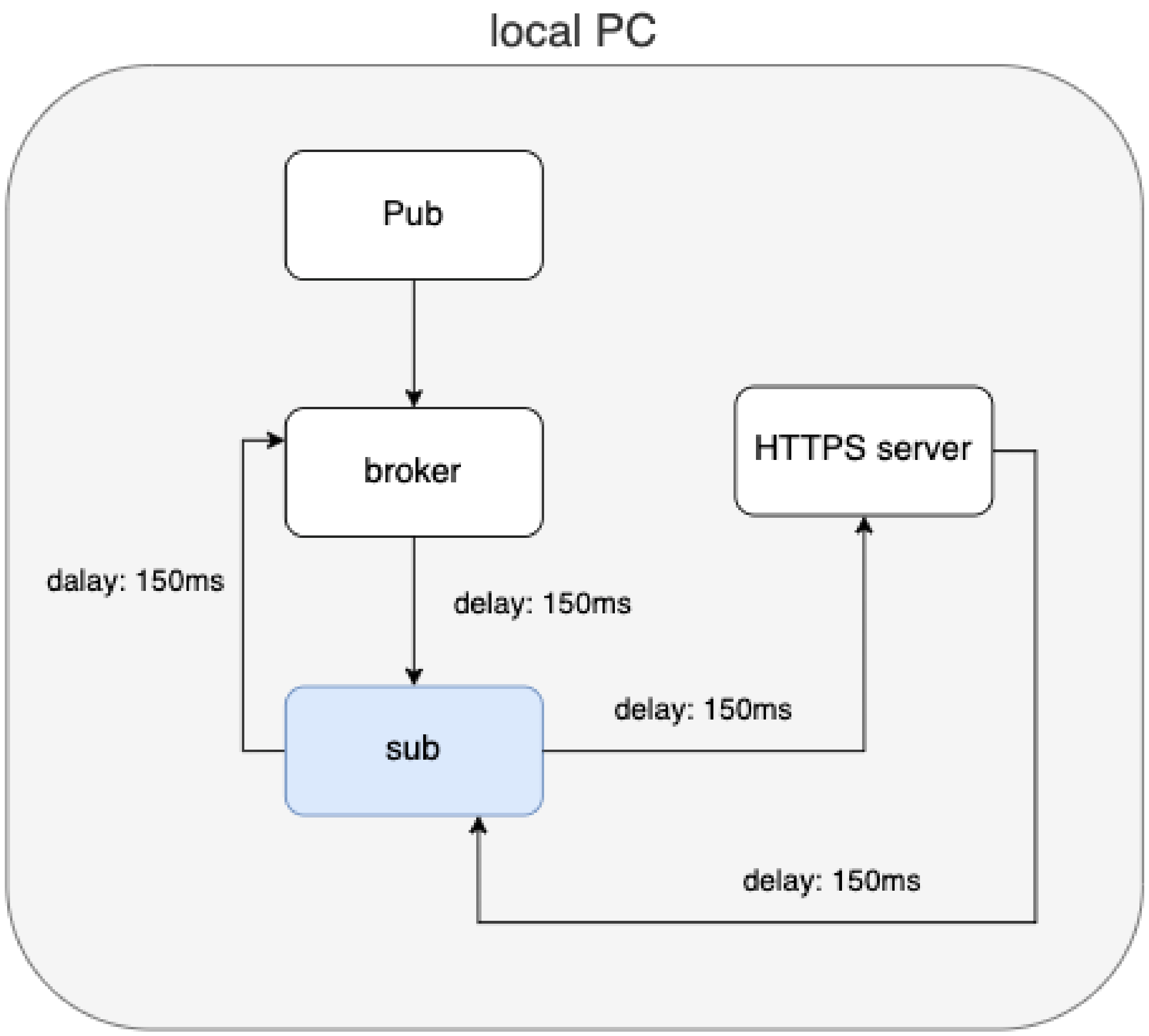
WIS/<Time>/<Country Code>/<CCCC>/<Fileformat>/<Cat1>/<Cat 2>

WMO OMM

Conditions of the experiment

- All components are built on docker
- High RTT was reproduced by "tc"
- prefetch = 1
- Consider the last ACK as the maximum delay
- use 3000 bufr file in GISC-cache
- http requests are not parallelized

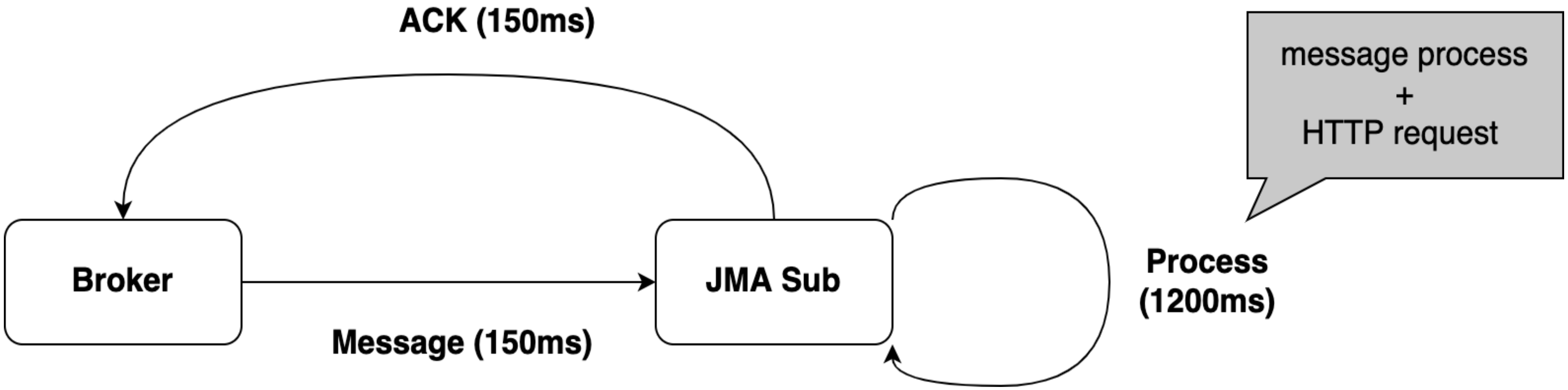
broker: rabbitMQ
sub: pika (base image python3:8)
web: httpd official image



WMO OMM

Latency (problem)

- 1 Message needs 1.5sec
- 3,000 simultaneous data => latency is about 4500s !



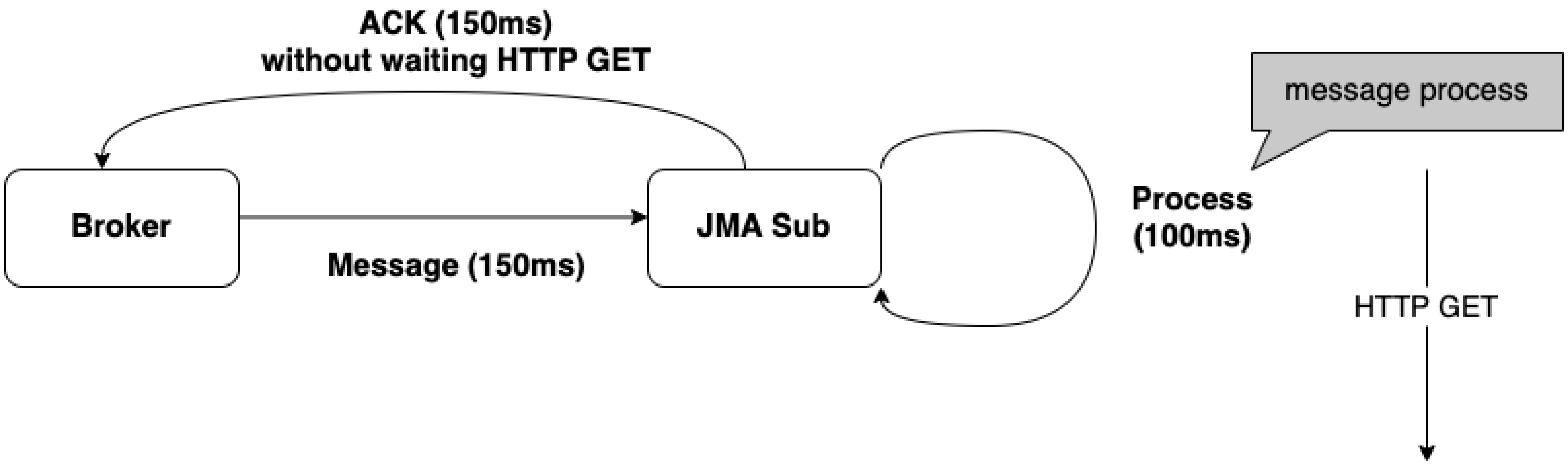
I did two things to reduce delay:

1. Asynchronization of http request
2. Use prefetch

WMO OMM

Asynchronization of http request

- Bottleneck is HTTP GET
- Asynchronization of http request
- We can receive more messages within a unit time

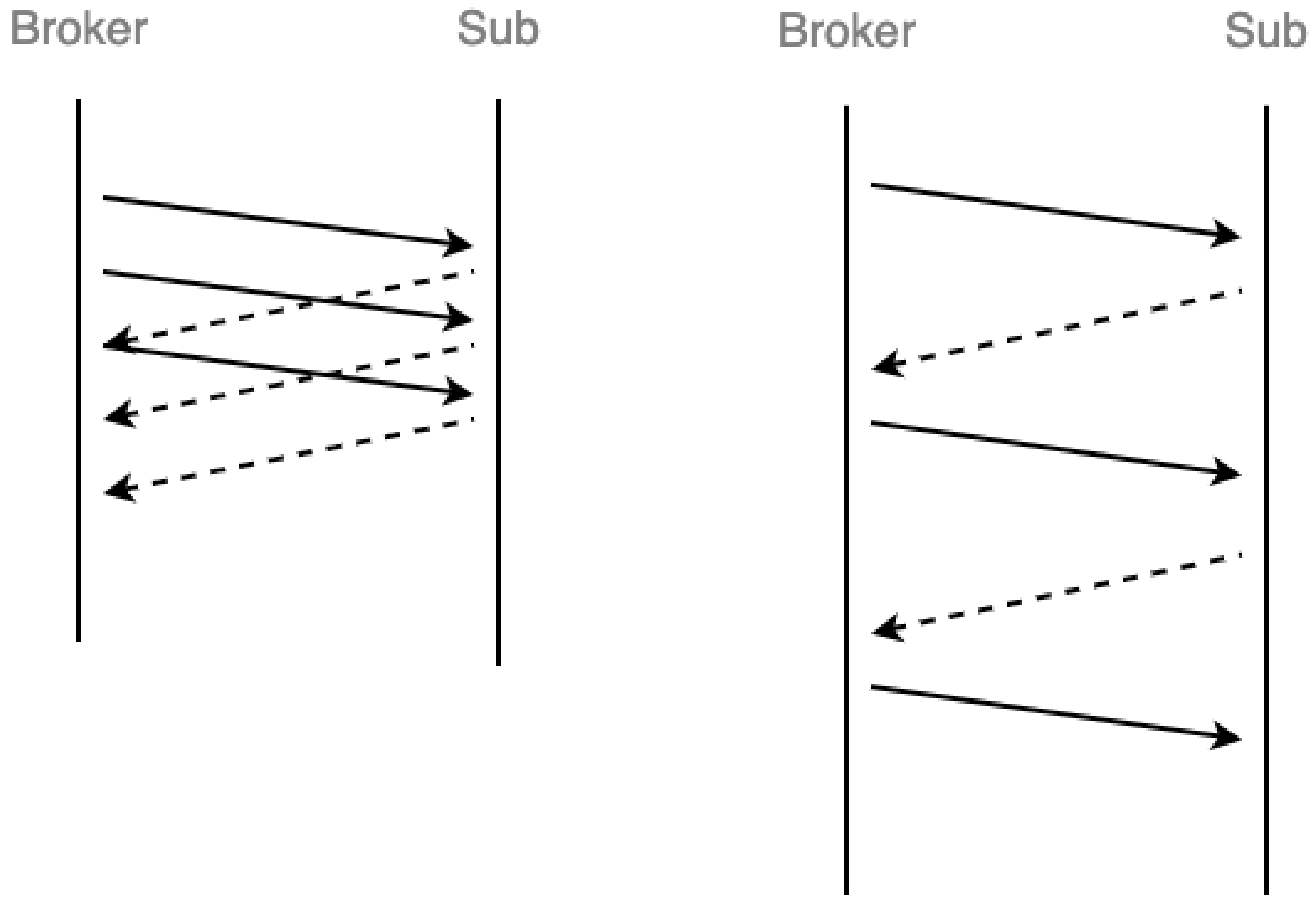


WMO OMM

What is Prefetch ?

Using Prefetch,
send a "prefetch" number of messages

Reduce latency in high RTT !



Use Prefetch

Normal

WMO OMM

Result (How much delay has been reduced?)

I did two things:

1. Asynchronization of http request
2. Use prefetch

<u>Before</u>

condition:

prefetch = 1, RTT = 300ms,
number of messages = 3000

result:

latency is about 4500s (max)

<u>After</u>

condition:

prefetch = 6, RTT = 300ms,
number of messages = 3000

result:

latency is about 180s(max)

Today we introduce two projects to you.

Here is a comparison of the research that has begun on both of them at this time.

Compare MQPs system with simple cloud downloading

WMO OMM

Feature of each architecture(A and B)

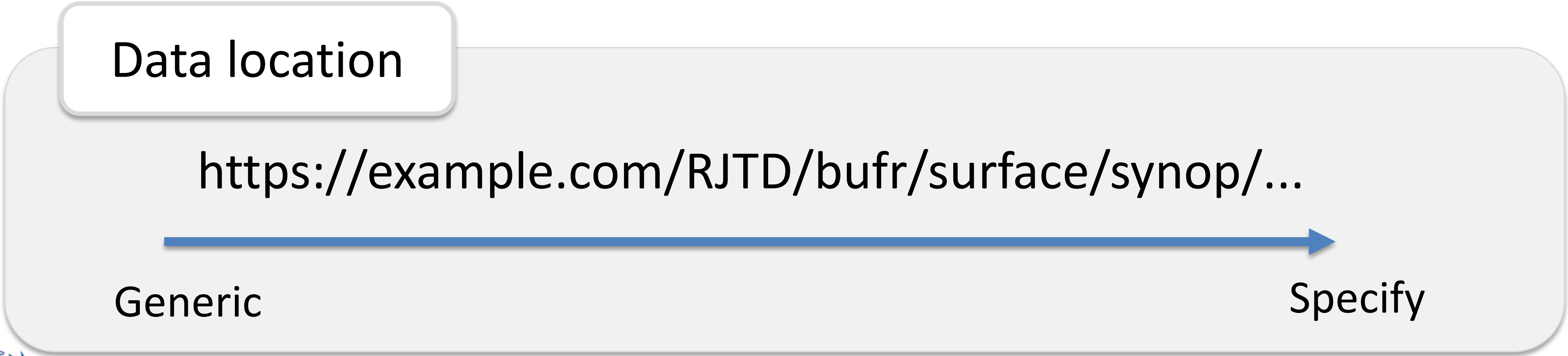
	MQPs (A)	simple cloud (B)
data description	topic tree	data path
how to get data (performance)	individually, ASAP	in batches at certain interval

WMO OMM

Data description: ("topic" vs "path")

"topic" is more flexible about Data description :

- "topic" is free (unrelated to data location)
- "path" is tree structure (data location itself)



WMO OMM

Performance: "individual (A)" vs "batch (B)"

Depends on data flow:

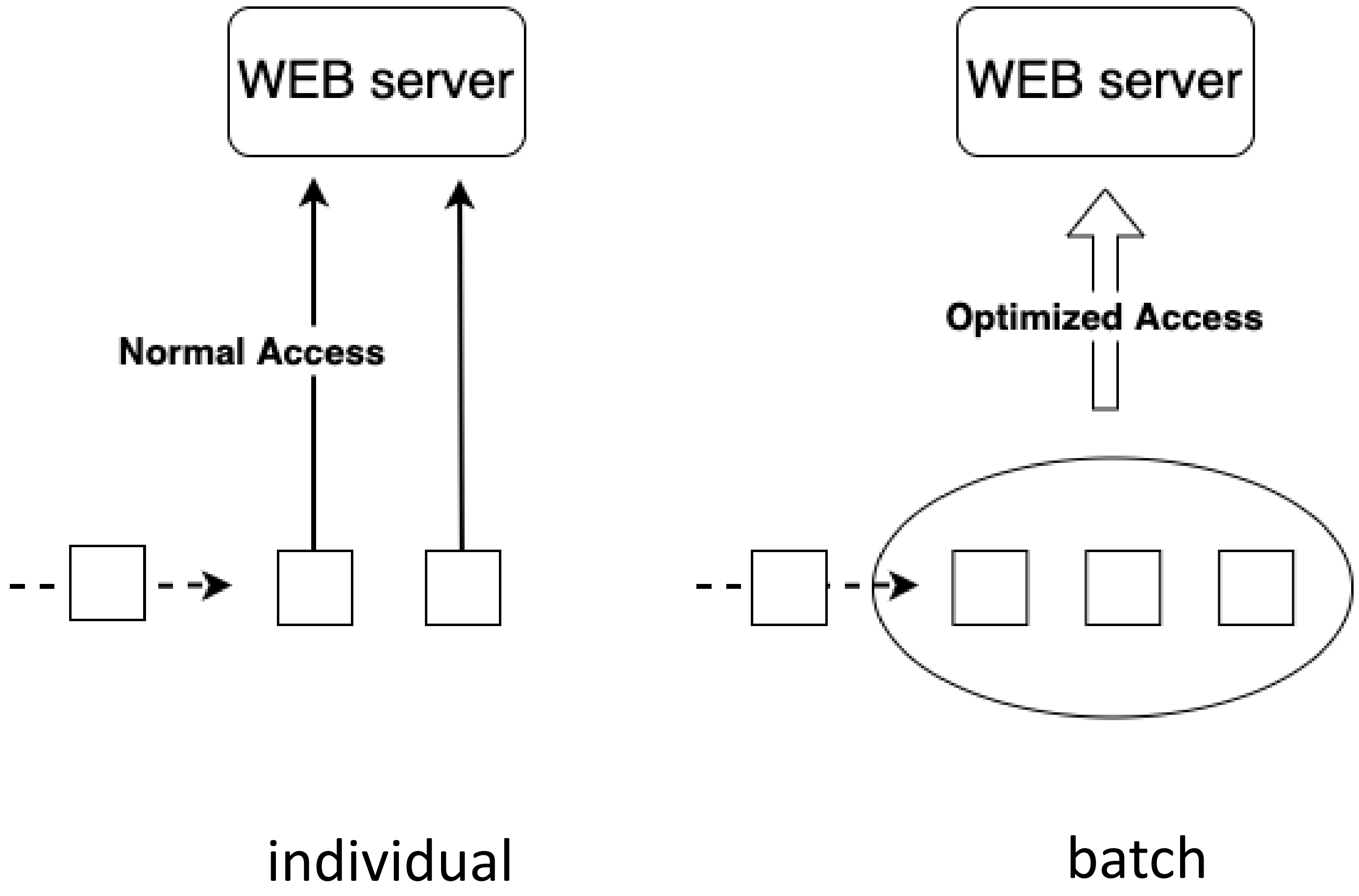
A is always connected, so it is very fast if there is no overhead
B is affected by the polling interval

situation	MQPs (A)	simple cloud (B)
small number of files	very fast (delay > 1sec)	fast (delay > 10sec)
large number of files	need optimization (large overhead)	efficient (small overhead)

Why "batch" is efficient, when the number of files is large?

- "batch" can optimize access
- How to optimize access ?
    - Lots of small files => batch
    - large file => separated, parallel

As the number of files increases, the effect of optimization becomes more significant!



WMO OMM

Survey on anticipated issues

task and issue	MQPs (A)	simple cloud (B)
Latency when number of files bursts	done	done
Implementation mirroring without routing	Not yet (use bridge ?)	done
operation of different tools	Not yet	done

Get the simple cloud downloader and try:
https://github.com/public-tatsuya-noyori/meteorological_preprocessor/blob/master/src/meteorological_preprocessor/catalog.md

WMO OMM

WEATHER CLIMATE WATER
TEMPS CLIMAT EAU

Thank you
Merci

WMO OMM

World Meteorological Organization

Organisation météorologique mondiale