

# SSEC Satellite Data Services

## Updates and Review of RabbitMQ Messaging System

Rick Kohrs

Satellite Data Services  
Space Science & Engineering Center  
University of Wisconsin – Madison



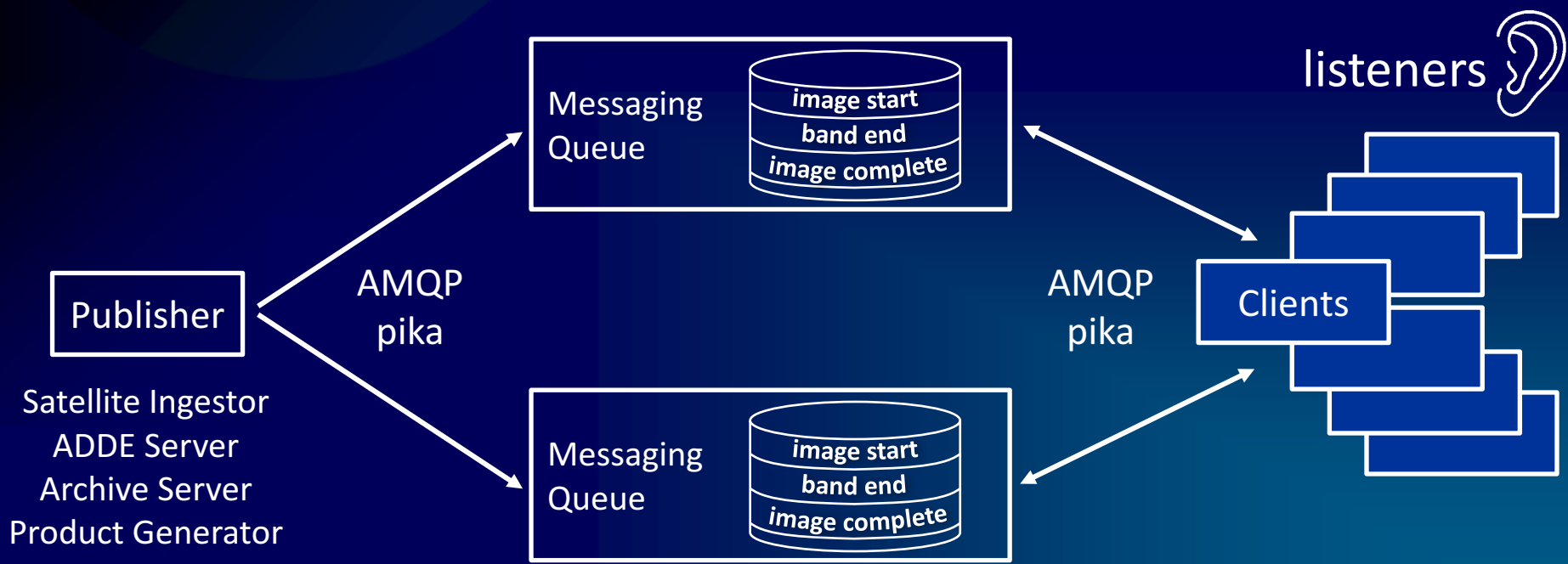
2023 McIDAS Users' Group Meeting  
25-28 September 2023  
Madison, WI



- What is RabbitMQ
- Message availability
- Data access
  - Old
  - New
- Satellite Data Services Usage
- amqpfind (tutorial)
- Example Usage

# • What is RabbitMQ

- Publisher (SDS currently ~6000 msgs/minute)
- Message Queue
- Client/Consumer/Listener
  - band end
  - image complete
- Advanced Message Queueing Protocol (AMQP)
- Pika – python implementation of AMQP



# Message Availability

- 3 Message Queue Servers
  - Two redundant servers at SSEC
  - One remote server
- ~200 user connections to single SSEC Message Queue server
- 20+ Satellites (model data under development)
- ~6000 published messages/minute
- Client side command to access messages
  - amqpfind (<https://www.ssec.wisc.edu/datacenter/amqpfind/>)
- Monitor messages
  - [Publisher problems](#)
  - [Missing Data](#)

SDS AMQP Monitor

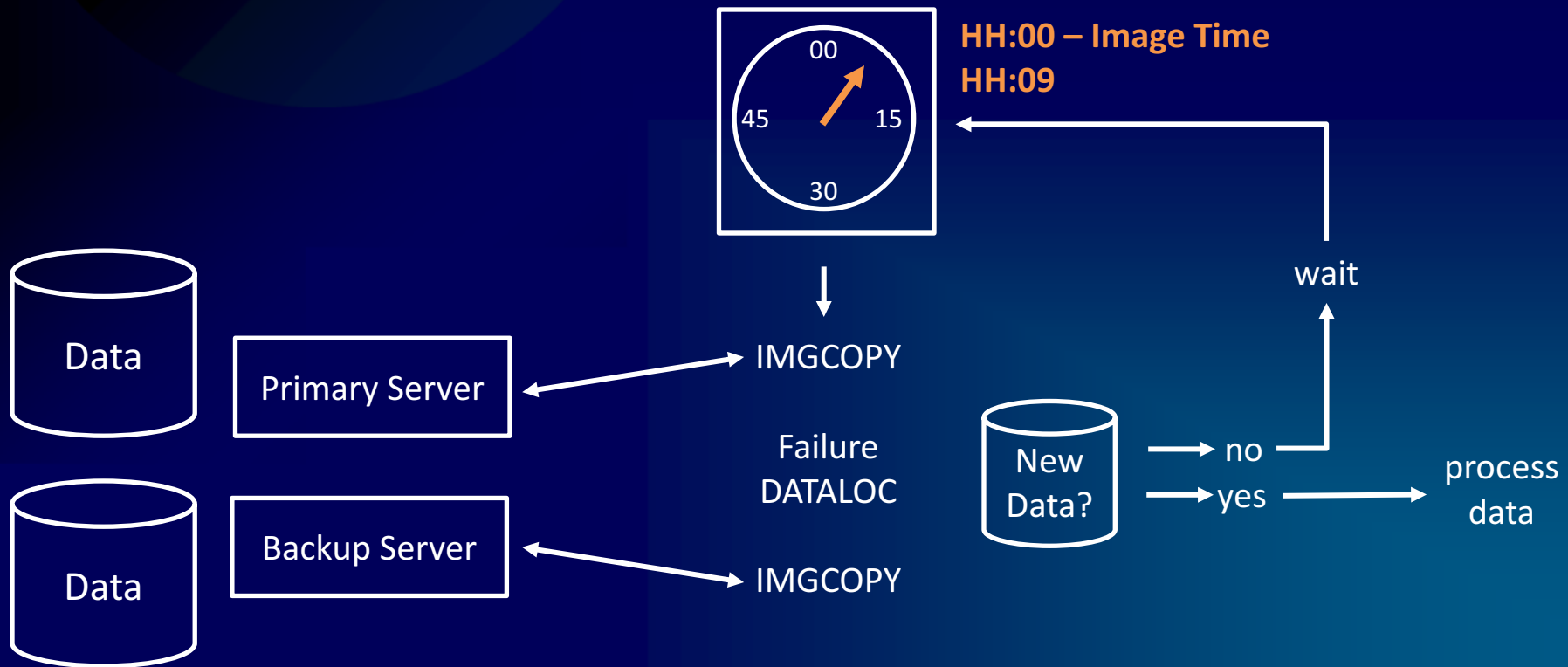
Satellite Model

amqpfind -X satellite -C \*.\*.\*.\*.\*.image.complete

Settings Clear Filters

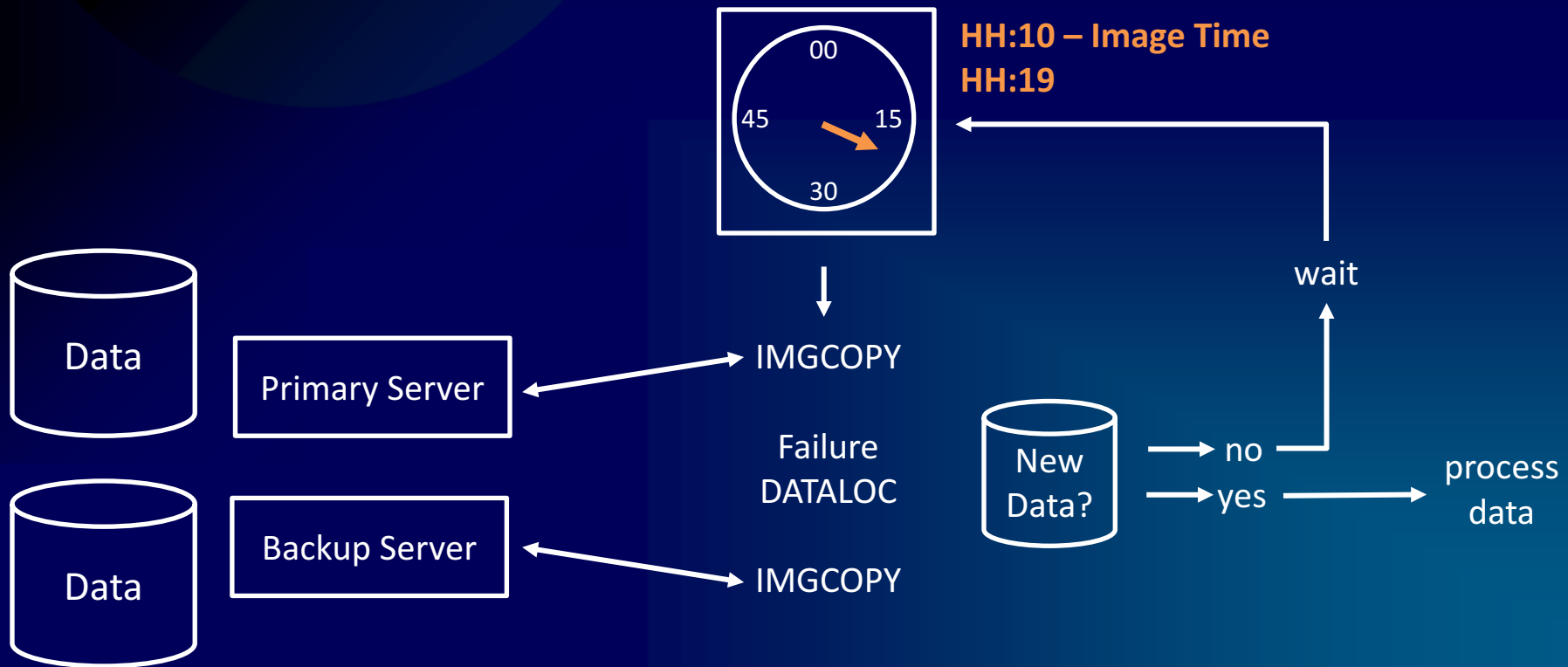
Type	Family	ID	Instrument	Medium	Server Type	Format	Class	Status	Host	Topic	Timestamp	Age	Past 3 Hours
ALL	ALL	ALL	ALL	ALL	ALL	ALL	image	complete	mq1	*.*.*.*.*.image.complete			
geo	coms	gk2a	ams	file	realtime	hdf	image	complete	mq1	geo.coms.gk2a.ams.file.realtime.hdf.image.complete	2023-09-15 19:29:42 UTC	1m	14 6 3
	evrs	g1	imager	addc	realtime	gvar	image	complete	mq1	geo.evrs.g1.imager.addc.realtime.gvar.image.complete	2023-09-15 18:55:50 UTC	36m	2 2 0
	goes	g16	abi	addc	realtime	ncdf	image	complete	mq1	geo.goes.g16.abi.addc.realtime.ncdf.image.complete	2023-09-15 19:31:14 UTC	11s	276 276 146
				file	archive	ncdf	image	complete	mq1	geo.goes.g16.abi.file.archive.ncdf.image.complete	2023-09-15 19:25:24 UTC	6m	148 143 52
			suvi	addc	realtime	ncdf	image	complete	mq1	geo.goes.g16.suvi.addc.realtime.ncdf.image.complete	2023-09-15 19:31:22 UTC	3s	1428 1440 748
				file	realtime	ncdf	image	complete	mq1	geo.goes.g16.suvi.file.realtime.ncdf.image.complete	2023-09-15 19:31:22 UTC	3s	714 720 374
		g18	abi	addc	realtime	ncdf	image	complete	mq1	geo.goes.g18.abi.addc.realtime.ncdf.image.complete	2023-09-15 19:31:14 UTC	11s	276 276 146
				file	archive	ncdf	image	complete	mq1	geo.goes.g18.abi.file.archive.ncdf.image.complete	2023-09-15 19:30:40 UTC	45s	282 284 125
			suvi	addc	realtime	ncdf	image	complete	mq1	geo.goes.g18.suvi.addc.realtime.ncdf.image.complete	2023-09-15 19:31:23 UTC	2s	1440 1432 754
				file	realtime	ncdf	image	complete	mq1	geo.goes.g18.suvi.file.realtime.ncdf.image.complete	2023-09-15 19:31:23 UTC	2s	720 716 377
	himawari	9	ahi	addc	realtime	hsf	image	complete	mq1	geo.himawari.9.ahi.addc.realtime.hsf.image.complete	2023-09-15 19:29:19 UTC	2m	105 108 58
				file	arcdata	hsf	image	complete	mq1	geo.himawari.9.ahi.file.arcdata.hsf.image.complete	2023-09-15 19:26:43 UTC	4m	54 54 27
				cluster	hsf	image	complete	mq1	geo.himawari.9.ahi.file.cluster.hsf.image.complete	2023-09-15 19:26:43 UTC	4m	54 54 27	
			hwimg	addc	realtime	hsf	image	complete	mq1	geo.himawari.9.ahi.addc.realtime.hsf.image.complete	2023-09-15 19:26:43 UTC	4m	54 54 27
			satbuf	addc	realtime	hsf	image	complete	mq1	geo.himawari.9.ahi.addc.realtime.hsf.image.complete	2023-09-15 19:29:43 UTC	1m	102 108 62
	meteosat	m10	seviri	addc	realtime	hdf	image	complete	mq1	geo.meteosat.m10.seviri.addc.realtime.hdf.image.complete	2023-09-15 19:21:09 UTC	10m	8 8 6
			seviri	addc	realtime	hdf	image	complete	mq1	geo.meteosat.m10.seviri.addc.realtime.hdf.image.complete	2023-09-15 19:15:49 UTC	15m	8 9 5
		d16	ssmi	file	realtime	hdf	image	complete	mq1	leo.dmsp.d16.ssmi.file.realtime.hdf.image.complete	2023-09-15 18:51:47 UTC	39m	2 2 0
		d17	ssmi	file	realtime	hdf	image	complete	mq1	leo.dmsp.d17.ssmi.file.realtime.hdf.image.complete	2023-09-15 19:21:50 UTC	9m	7 14 7
		d18	ssmi	file	realtime	hdf	image	complete	mq1	leo.dmsp.d18.ssmi.file.realtime.hdf.image.complete	2023-09-15 19:21:51 UTC	9m	7 14 7
		dmsp17	mrs	file	realtime	ncdf	image	complete	mq1	leo.dmsp.dmsp17.mrs.file.realtime.ncdf.image.complete	2023-09-15 18:51:26 UTC	39m	0 8 0
		dmsp18	mrs	file	realtime	ncdf	image	complete	mq1	leo.dmsp.dmsp18.mrs.file.realtime.ncdf.image.complete	2023-09-15 19:21:29 UTC	9m	2 0 2
	jps	n20	atms	file	realtime	hdf	image	complete	mq1	leo.jps.n20.atms.file.realtime.hdf.image.complete	2023-09-10 00:03:16 UTC	5d	0 0 0
			cris	file	realtime	hdf	image	complete	mq1	leo.jps.n20.cris.file.realtime.hdf.image.complete	2023-09-15 00:34:36 UTC	18h	0 0 0
			viirs	file	realtime	hdf	image	complete	mq1	leo.jps.n20.viirs.file.realtime.hdf.image.complete	2023-09-15 16:35:42 UTC	2h	663 357 239
		n21	atms	file	realtime	hdf	image	complete	mq1	leo.jps.n21.atms.file.realtime.hdf.image.complete	2023-09-13 20:17:10 UTC	1d	0 0 0
			cris	file	realtime	hdf	image	complete	mq1	leo.jps.n21.cris.file.realtime.hdf.image.complete	2023-09-14 20:16:37 UTC	23h	0 0 0
			viirs	file	realtime	hdf	image	complete	mq1	leo.jps.n21.viirs.file.realtime.hdf.image.complete	2023-09-15 19:21:05 UTC	10m	98 35 35
			atms	file	realtime	hdf	image	complete	mq1	leo.jps.n21.atms.file.realtime.hdf.image.complete	2023-09-15 16:35:42 UTC	2h	0 0 0
		snp	cris	file	realtime	hdf	image	complete	mq1	leo.jps.snp.cris.file.realtime.hdf.image.complete	2023-09-15 19:21:30 UTC	9m	38 29 10
			viirs	file	realtime	hdf	image	complete	mq1	leo.jps.snp.viirs.file.realtime.hdf.image.complete	2023-09-15 19:28:33 UTC	2m	1790 1842 370
	metop	m1	amsu	file	realtime	hdf	image	complete	mq1	leo.metop.m1.amsu.file.realtime.hdf.image.complete	2023-09-15 19:31:03 UTC	22s	27 30 28
			argps	file	realtime	hdf	image	complete	mq1	leo.metop.m1.argps.file.realtime.hdf.image.complete	2023-09-15 19:21:22 UTC	10m	1 1 2
			svhr	file	realtime	hdf	image	complete	mq1	leo.metop.m1.svhr.file.realtime.hdf.image.complete	2023-09-15 19:21:25 UTC	10m	4 8 10

- Data Access – Old Method
  - cron
    - Guess work for time of data arrival
  - Multiple ADDE request
    - Increased system load on server
    - Scripts may run multiple times
  - Potential for missing data



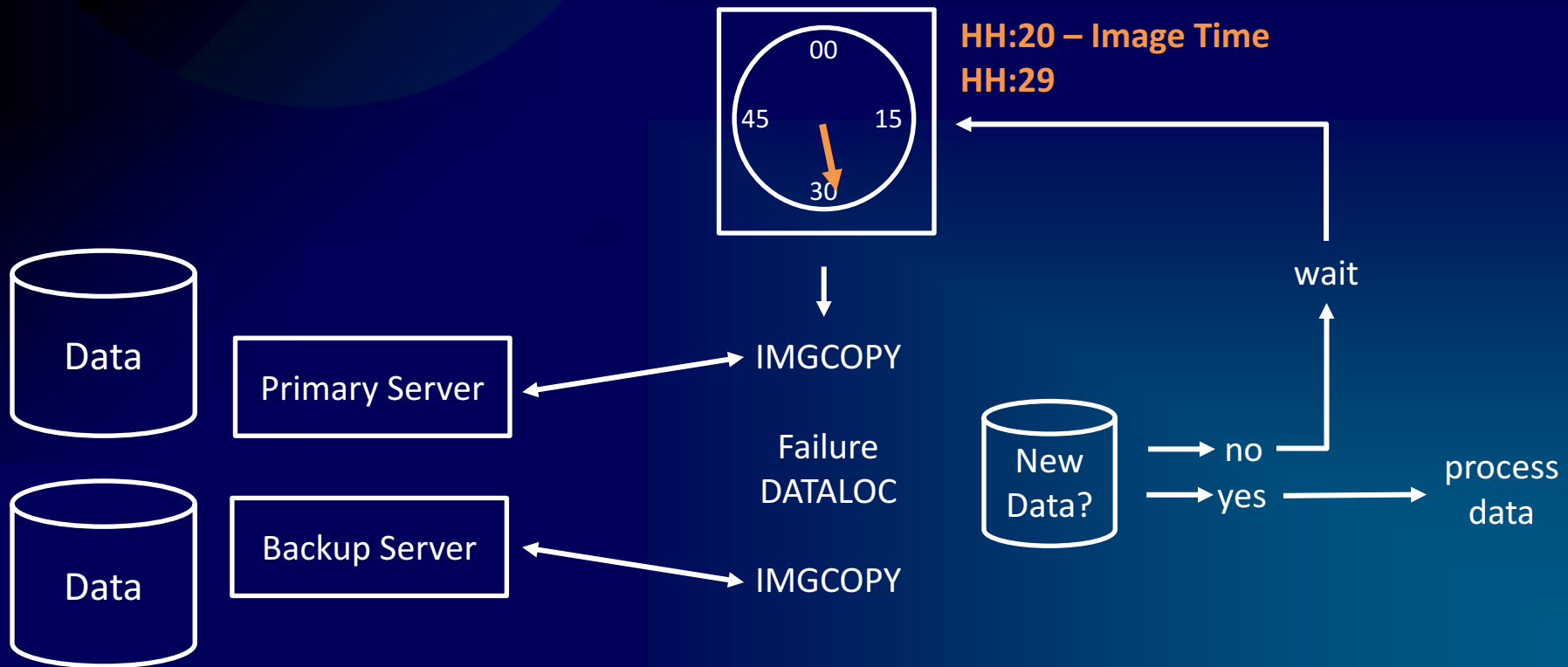
- Data Access – Old Method

- cron
  - Guess work for time of data arrival
- Multiple ADDE request
  - Increased system load on server
  - Scripts may run multiple times
- Potential for missing data

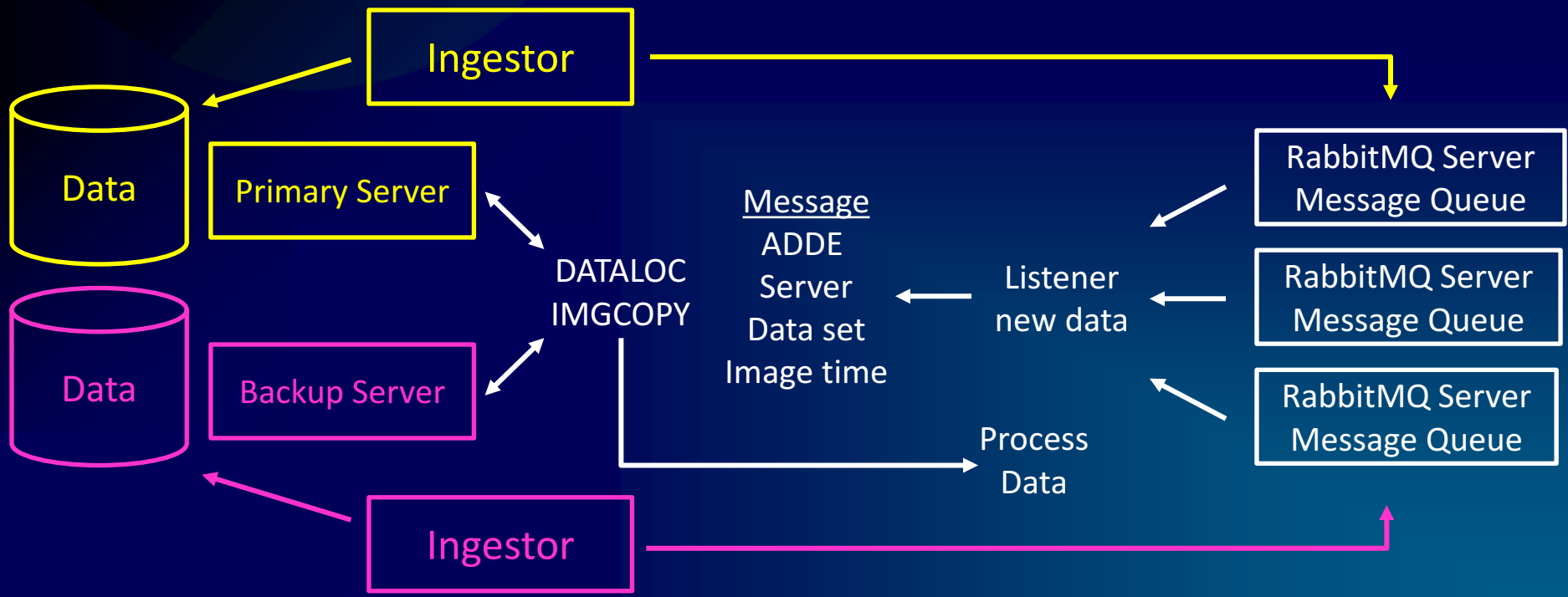


- Data Access – Old Method

- cron
  - Guess work for time of data arrival
- Multiple ADDE request
  - Increased system load on server
  - Scripts may run multiple times
- Potential for missing data

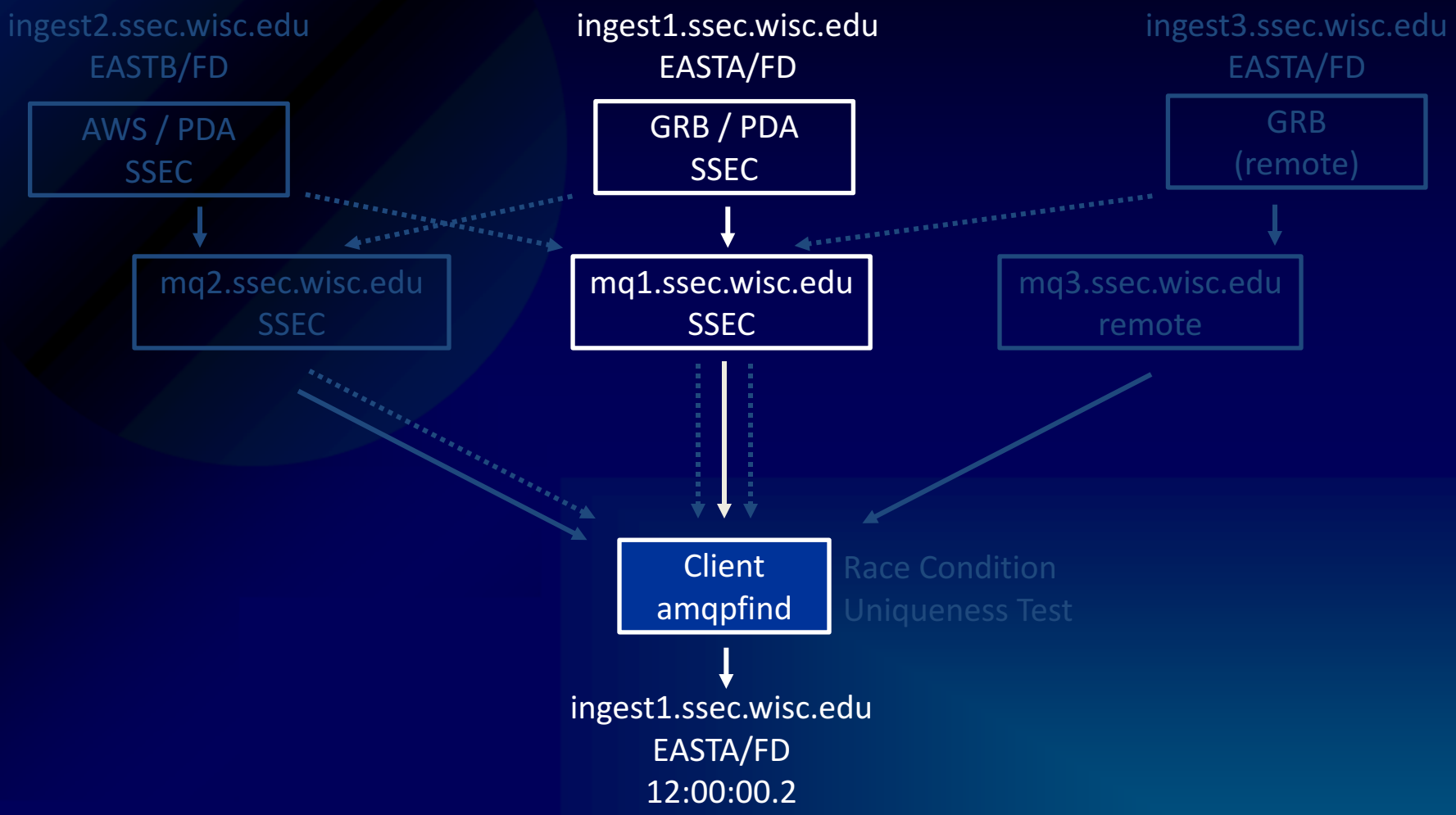


- Data Access – New Method
  - Image complete messages published
    - No guessing image frequency
  - ADDE server and dataset information available
    - Backup check obsolete
    - Exact image time known



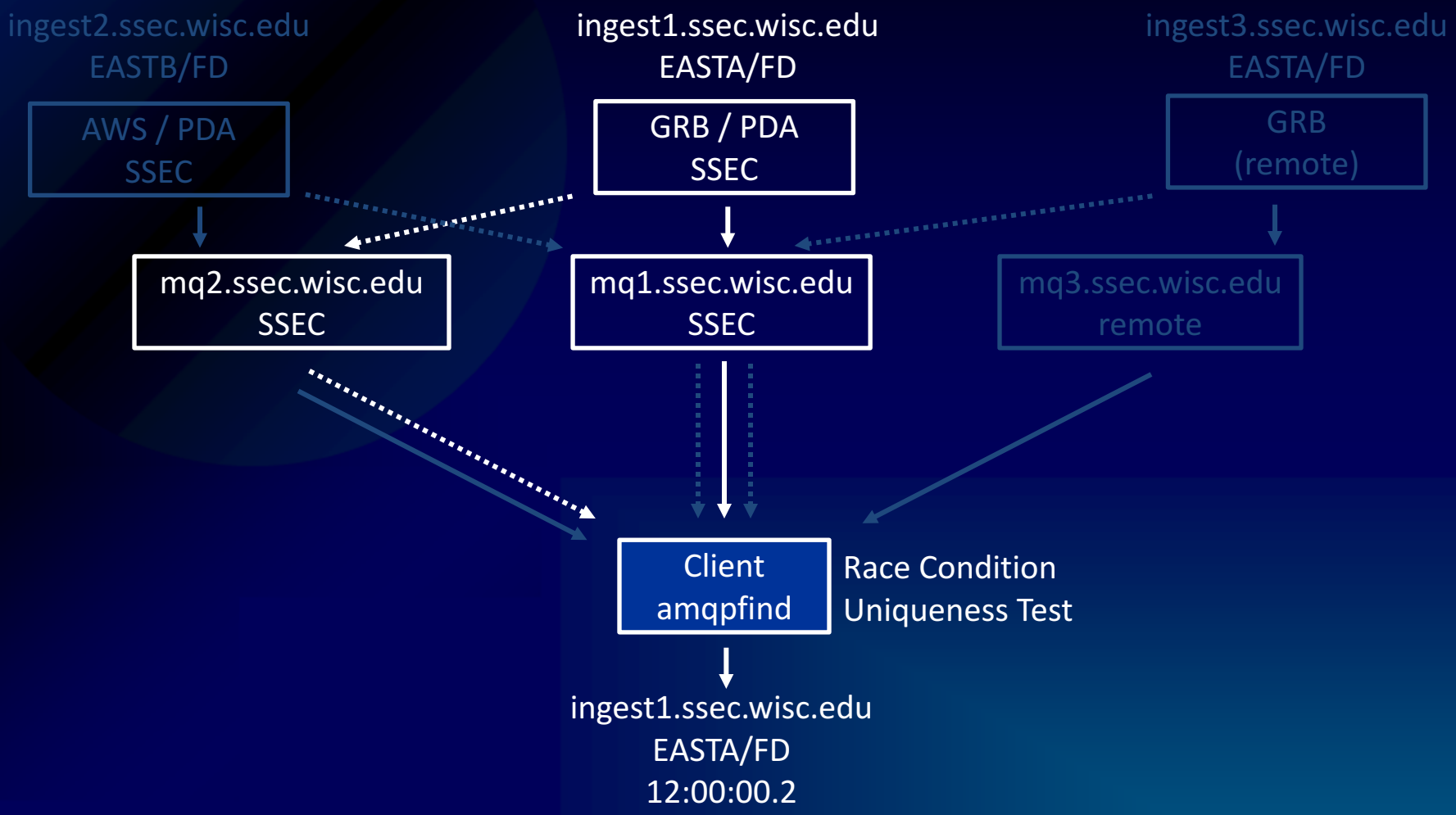


# Satellite Data Services



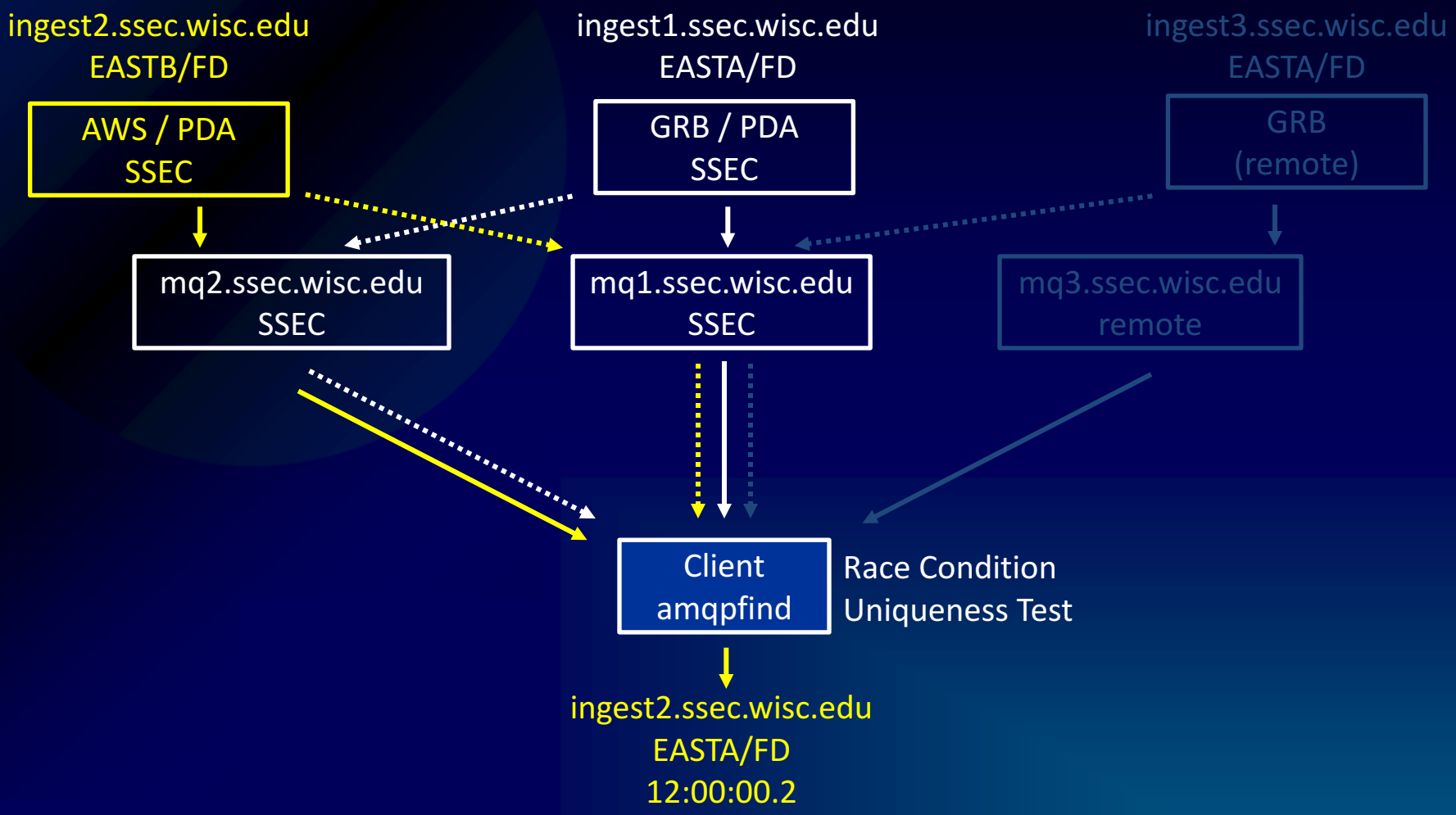
```
DATALOC ADD EASTA ingest1.ssec.wisc.edu  
IMCOPY EASTA/FD MY/AREAS.1 SIZE=SAME BAND=13 TIME=12:00:00
```

# Satellite Data Services



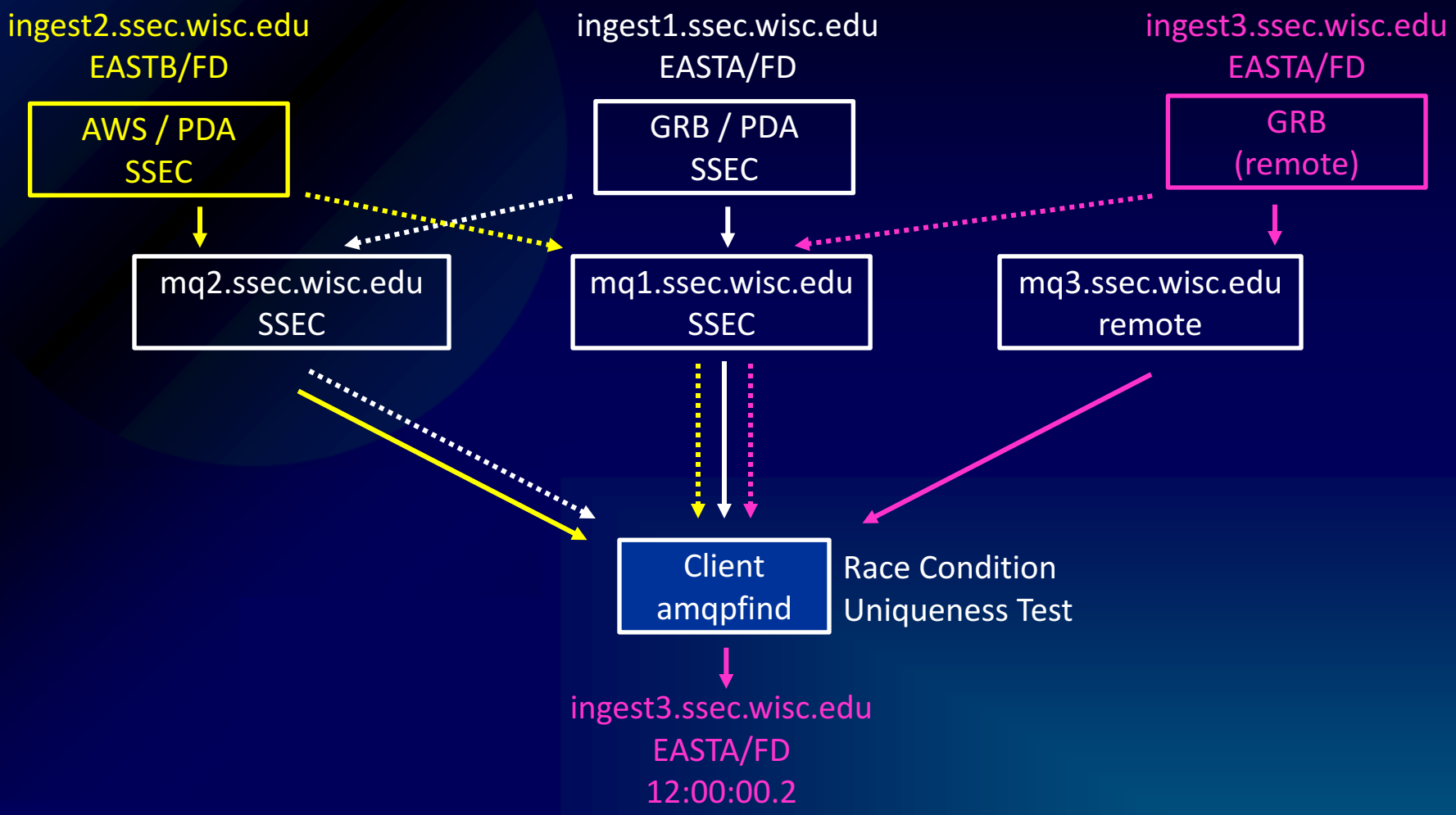
```
DATALOC ADD EASTA ingest1.ssec.wisc.edu  
IMCOPY EASTA/FD MY/AREAS.1 SIZE=SAME BAND=13 TIME=12:00:00
```

# Satellite Data Services



```
DATALOC ADD EASTA ingest2.ssec.wisc.edu  
IMCOPY EASTA/FD MY/AREAS.1 SIZE=SAME BAND=13 TIME=12:00:00
```

# Satellite Data Services



```
DATALOC ADD EASTA ingest3.ssec.wisc.edu  
IMCOPY EASTA/FD MY/AREAS.1 SIZE=SAME BAND=13 TIME=12:00:00
```

- amqpfind

- General Information
- Topic – set of subscription keys
- Payload – value pairs of message
- Monitor multiple RabbitMQ servers

```
amqpfind -H -u -p [-t] [-X] [-j'{key} {key}' | '{?}'] [-C key | key]
```

-H	host ip address of the Satellite Data Services RabbitMQ server mq1(2).ssec.wisc.edu
-u	user for SDS RabbitMQ server (sdsuser)
-p	password for SDS RabbitMQ server (sdsmq)
-j	outputs values of payload keys of a message; must be a single quoted string, each key must be surrounded by curly braces; for example: -j '{start_time} {band} {status}'
-j '?'	lists all keys and values from message, some shell types may need to use curly brackets; -j '{?}'
-k	keys checked to see if a message, sent from multiple servers, is considered a duplicate
-w	time window, in second, used to watch for duplicate messages
-t	sets timeout in seconds before amqpfind ends
-X	SDS RabbitMQ server exchange; valid options are satellite, model (development) or misc
-C	period-separated list of nine subscription keys; specific values or wild card must be specified for all nine keys;
-v	debug level

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
$HOME/amqpfind/amqpfind -X satellite\  
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \  
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \  
-j '{start_time} {satellite_ID} {adde_dataset} {coverage}' \  
-k '(start_time)' \  
-w 60.0 \  
-t 0
```

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
$HOME/amqpfind/amqpfind -X satellite\  
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \  
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \  
-j '{start_time} {satellite_ID} {adde_dataset} {coverage}' \  
-k '(start_time)' \  
-w 60.0 \  
-t 0
```

-X Exchange; common link between publisher and listener, most if not all of you will use satellite

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
$HOME/amqpfind/amqpfind -X satellite\  
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \  
-H mq3.ssec.wisc.edu -p sdsmq -u sdsuser \  
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \  
-j '{start_time} {satellite_ID} {adde_dataset} {coverage}' \  
-k '(start_time)' \  
-w 60.0 \  
-t 0
```

-H RabbitMQ servers, currently available are *mq1.ssec.wisc.edu*, *mq2.ssec.wisc.edu*, *mq3.ssec.wisc.edu*  
-p password, for SDS, you should use *sdsmq*  
-u users, for SDS, you should use *sdsuser*

To make things easier to read, most users group them together.



- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
$HOME/amqpfind/amqpfind -X satellite\  
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \  
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \  
-j '{start_time} {satellite_ID} {adde_dataset} {coverage}' \  
-k '(start_time)' \  
-w 60.0 \  
-t 0
```

-C Subscription keys. Must have 9 period separated values; use \* as a wildcard.

A set of Subscription keys is called a topic.

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
$HOME/amqpfind/amqpfind -X satellite\  
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \  
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \  
-j '{start_time} {satellite_ID} {adde_dataset} {coverage}' \  
-k '(start_time)' \  
-w 60.0 \  
-t 0
```

-j Output sent to standard out. Generally used as arguments passed to a script (discussed later)

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
$HOME/amqpfind/amqpfind -X satellite\  
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \  
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \  
-j '{start_time} {satellite_ID} {adde_dataset} {coverage}' \  
-k '(start_time)' \  
-w 60.0 \  
-t 0
```

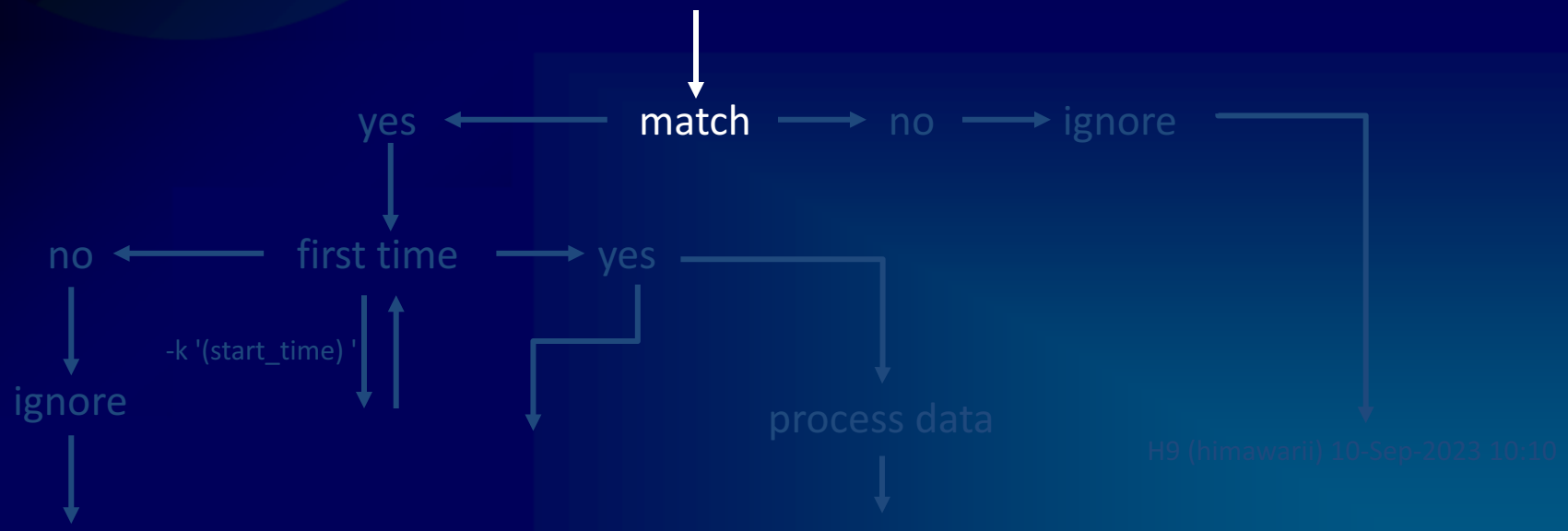
These are used to filter out duplicate messages (race condition)

- k list of parameters that must be met before a message is considered unique
- w window time (system clock); how long incoming messages will be checked for uniqueness.  
***to avoid processing duplicate messages, -w and -k should be used in tandem***
- t How long to wait for messages before stopping command; value of 0 means never stop.

- amqpfind
  - Race Condition

G16 (goes) 10-Sep-2023 10:10 (mq2)  
 M10 (meteosat) 10-Sep-2023 10:45  
 G18 (goes) 10-Sep-2023 10:10  
 G16 (goes) 10-Sep-2023 10:10 (mq1)  
 H9 (himawarii) 10-Sep-2023 10:10

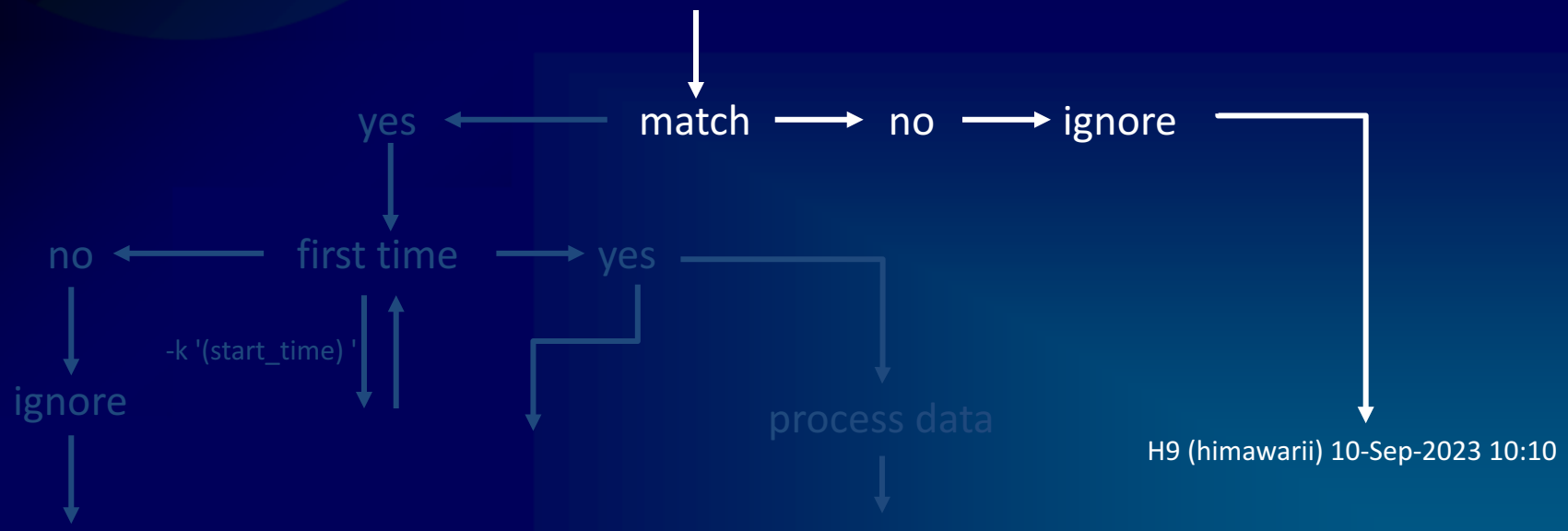
```
-C 'geo.goes.*.abi.adde.realtime.*.image.complete'
```



- amqpfind
  - Race Condition

G16 (goes) 10-Sep-2023 10:10 (mq2)  
M10 (meteosat) 10-Sep-2023 10:45  
G18 (goes) 10-Sep-2023 10:10  
G16 (goes) 10-Sep-2023 10:10 (mq1)  
H9 (himawarii) 10-Sep-2023 10:10

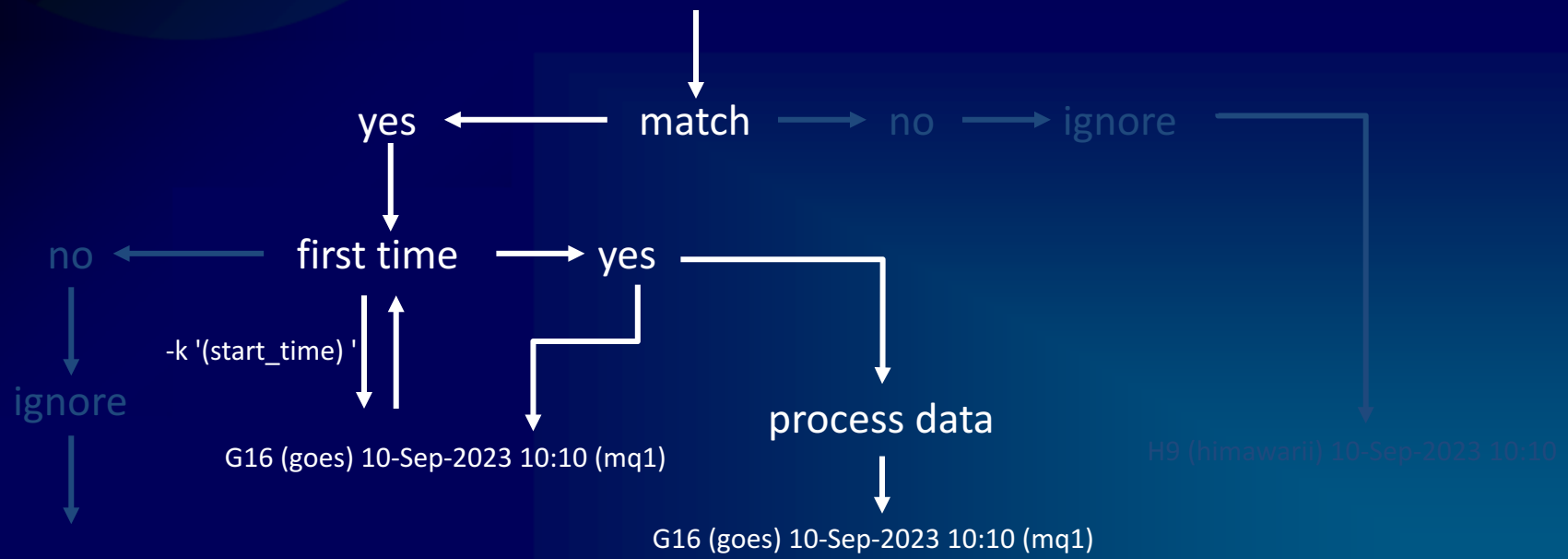
```
-C 'geo.goes.*.abi.adde.realtime.*.image.complete'
```



- amqpfind
  - Race Condition

G16 (goes) 10-Sep-2023 10:10 (mq2)  
 M10 (meteosat) 10-Sep-2023 10:45  
 G18 (goes) 10-Sep-2023 10:10:01  
 G16 (goes) 10-Sep-2023 10:10 (mq1)

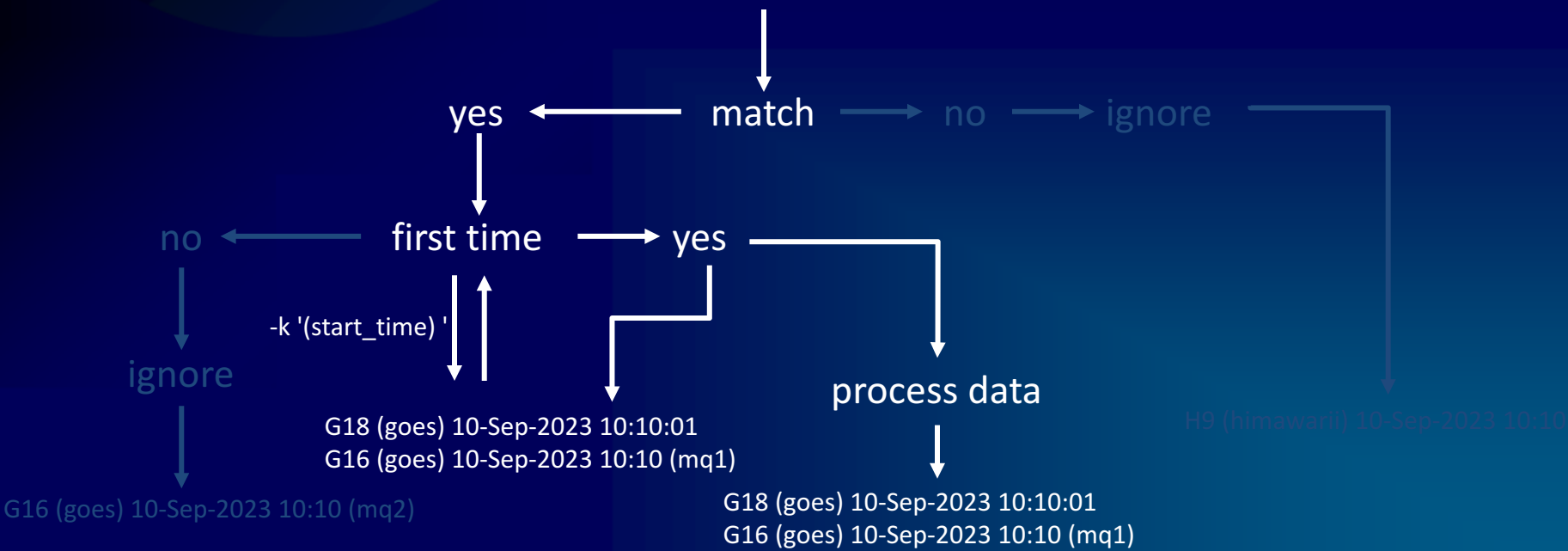
```
-C 'geo.goes.*.abi.adde.realtime.*.image.complete'
```



- amqpfind
  - Race Condition

G16 (goes) 10-Sep-2023 10:10 (mq2)  
 M10 (meteosat) 10-Sep-2023 10:45  
 G18 (goes) 10-Sep-2023 10:10:01

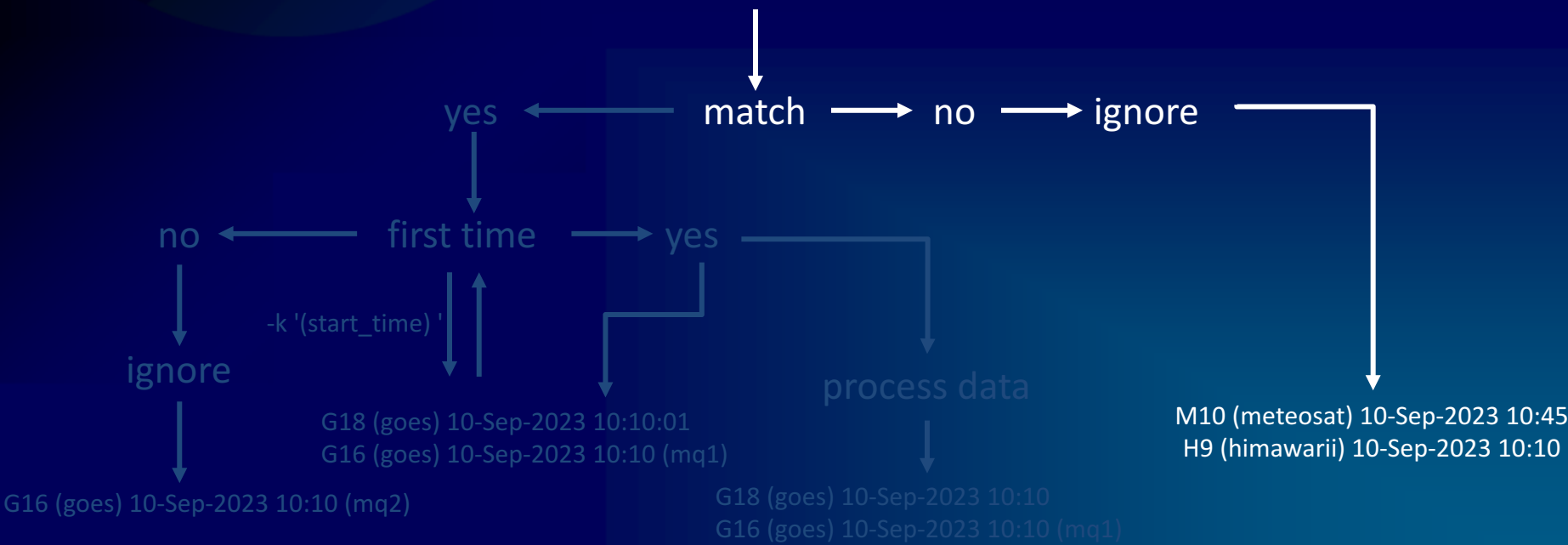
```
-C 'geo.goes.*.abi.adde.realtime.*.image.complete'
```



- amqpfnd
  - Race Condition

G16 (goes) 10-Sep-2023 10:10 (mq2)  
 M10 (meteosat) 10-Sep-2023 10:45

```
-C 'geo.goes.*.abi.adde.realtime.*.image.complete'
```

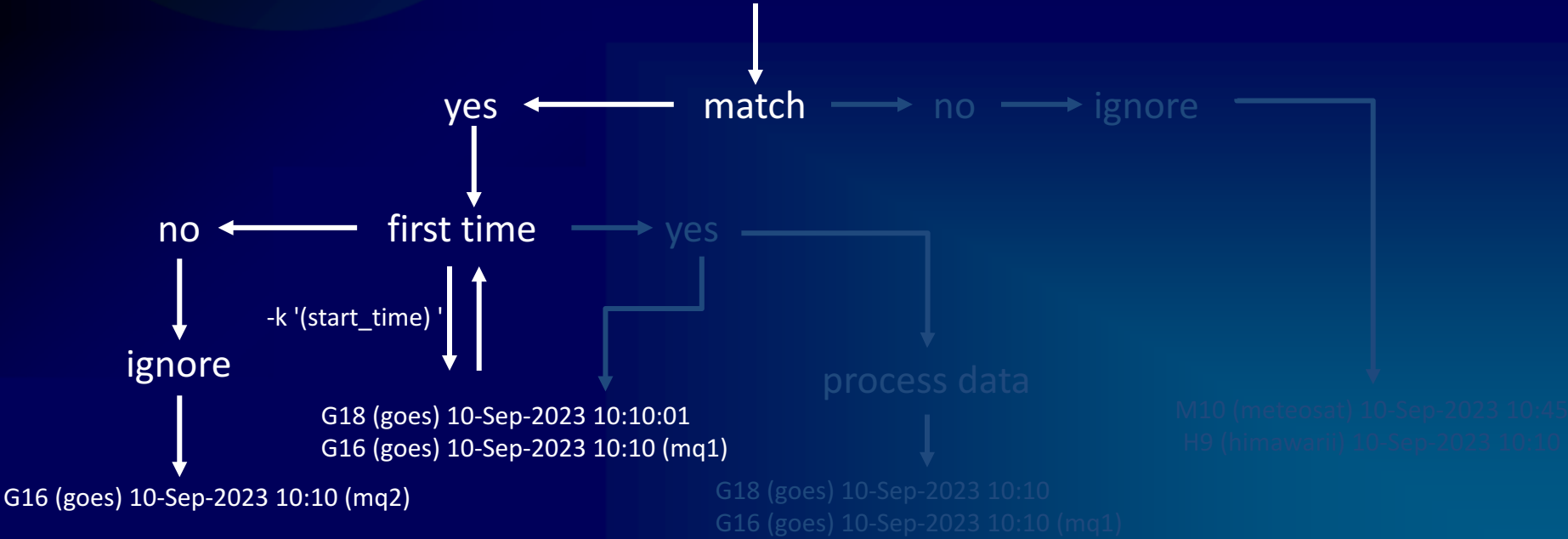




- amqpfind
  - Race Condition

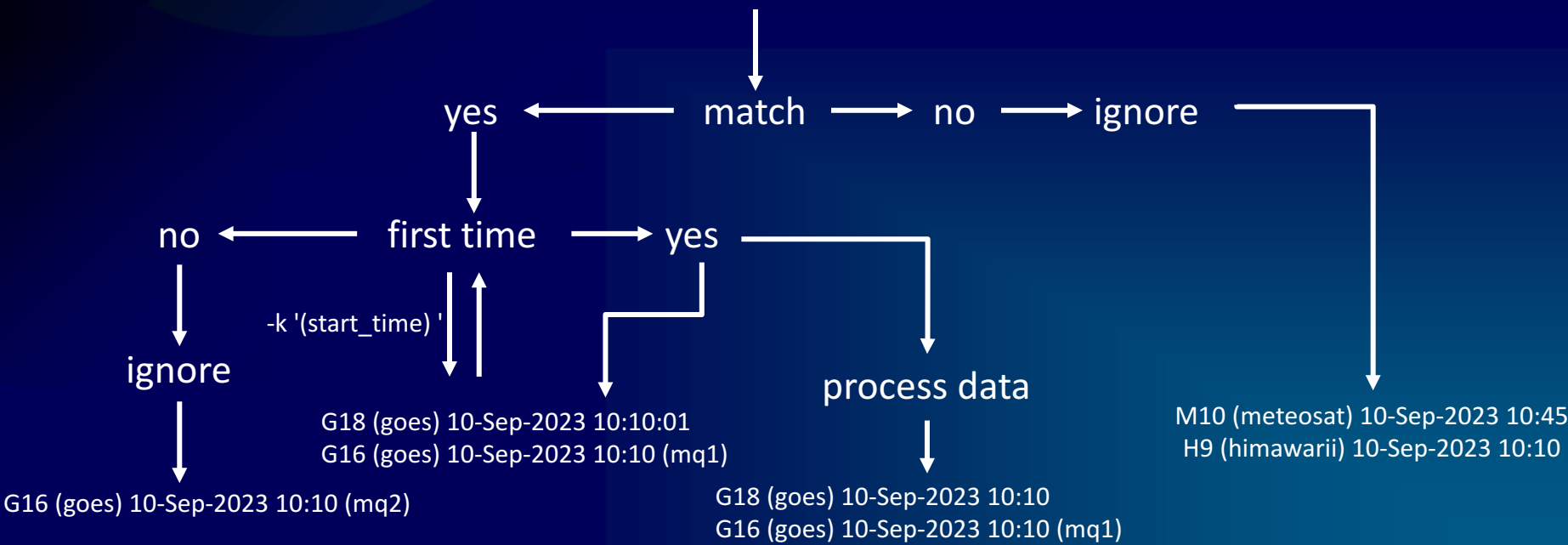
G16 (goes) 10-Sep-2023 10:10 (mq2)

```
-C 'geo.goes.*.abi.adde.realtime.*.image.complete'
```



- amqpfind
  - Race Condition

```
-C 'geo.goes.*.abi.adde.realtime.*.image.complete'
```



- amqpfnd - examples
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
$HOME/amqpfnd/amqpfnd -X satellite\
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \
-j '{start_time} {satellite_ID} {adde_dataset} {coverage}' \
-k '(start_time)' \
-w 60.0 \
-t 0
```

By default, an amqpfnd command finds all published messages. Here, only -C is used to limit the number. This may cause your script to run multiple times for the same image.

satellite_ID	start_time	coverage	received time	server
G16	10:10:01.9	Full Disk	10:20:00	mq1.ssec.wisc.edu
G16	10:10:01.9	Full Disk	10:20:20	mq2.ssec.wisc.edu
<del>M10</del>	<del>10:15:00.0</del>	<del>Full Disk</del>	<del>10:30:00</del>	<del>mq1.ssec.wisc.edu</del> Wrong satellite
G18	10:10:01.9	Full Disk	10:20:30	mq1.ssec.wisc.edu
G18	10:20:02.0	Full Disk	10:21:31	mq1.ssec.wisc.edu
G16	10:20:01.9	Full Disk	10:29:30	mq2.ssec.wisc.edu
G16	10:10:01.9	Full Disk	12:20:00	mq1.ssec.wisc.edu

- amqpfind - examples
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
$HOME/amqpfind/amqpfind -X satellite\
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \
-j '{start_time} {satellite_ID} {adde_dataset} {coverage}' \
-k '(start_time)' \
-w 60.0 \
-t 0
```

All messages would be received.  
 Adding in arguments for -k and -w

satellite_ID	start_time	coverage	received time	server	
G16	10:10:01.9	Full Disk	10:20:00	mq1.ssec.wisc.edu	
<del>G16</del>	<del>10:10:01.9</del>	<del>Full Disk</del>	<del>10:20:20</del>	<del>mq2.ssec.wisc.edu</del>	Same time inside the window
<del>M10</del>	<del>10:15:00.0</del>	<del>Full Disk</del>	<del>10:30:00</del>	<del>mq1.ssec.wisc.edu</del>	Wrong satellite
G18	10:10:01.9	Full Disk	10:20:30	mq1.ssec.wisc.edu	
G18	10:20:02.0	Full Disk	10:21:31	mq1.ssec.wisc.edu	
G16	10:20:01.9	Full Disk	10:29:30	mq2.ssec.wisc.edu	
G16	10:10:01.9	Full Disk	12:20:00	mq1.ssec.wisc.edu	

- amqpfnd - examples
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
$HOME/amqpfnd/amqpfnd -X satellite\
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \
-j '{start_time} {satellite_ID} {adde_dataset} {coverage}' \
-k '(start_time)' \
-w 14400.0 \
-t 0
```

All messages would be received.

Adding in arguments for -k and -w; increase window to 3 hours

satellite_ID	start_time	coverage	received time	server	
G16	10:10:01.9	Full Disk	10:20:00	mq1.ssec.wisc.edu	
G16	10:10:01.9	Full Disk	10:20:20	mq2.ssec.wisc.edu	Same time inside the window
M10	10:15:00.0	Full Disk	10:30:00	mq1.ssec.wisc.edu	Wrong satellite
G18	10:10:01.9	Full Disk	10:20:30	mq1.ssec.wisc.edu	
G18	10:20:02.0	Full Disk	10:21:31	mq1.ssec.wisc.edu	
G16	10:20:01.9	Full Disk	10:29:30	mq2.ssec.wisc.edu	
G16	10:10:01.9	Full Disk	12:20:00	mq1.ssec.wisc.edu	Same time inside the window

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

geo  
leo  
misc

**geo.goes.g16.abi.adde.realtime.ncdf.image.complete**

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

geo	goes
leo	himawari
<b>meteosat</b>	<b>ews</b>
<b>metop</b>	<b>jpss</b>
<b>sentinel</b>	<b>dmsp</b>

geo.goes.g16.abi.adde.realtime.ncdf.image.complete



- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

geo	goes	g16
leo	himawari	g18
	meteosat	h9
	ews	m10
	sentinel	m9
	metop	snpp
	jpss	noaa20
	dmsp	

**geo.goes.g16.abi.adde.realtime.ncdf.image.complete**

- amqpfind

- General Information
- Topic – set of subscription keys
- Payload – value pairs of message
- Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

geo	goes	g16	abi
leo	himawari	g18	glm
	meteosat	h9	exis
	ews	m10	magnetometer
	sentinel	m9	seiss
	metop	snpp	suvi
	jpss	noaa20	ahi
	dmsp		

**geo.goes.g16.abi.adde.realtime.ncdf.image.complete**

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

geo	goes	g16	abi	adde
leo	himawari	g18	glm	file
	meteosat	h9	exis	
	ews	m10	magnetometer	
	sentinel	m9	seiss	
	metop	snpp	suvi	
	jpss	noaa20	ahi	
	dmsp			

**geo.goes.g16.abi.adde.realtime.ncdf.image.complete**

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

geo	goes	g16	abi	adde	realtime
leo	himawari	g18	glm	file	archive
	meteosat	h9	exis		satbuf
	ews	m10	magnetometer		cluster
	sentinel	m9	seiss		arcdata
	metop	snpp	suvi		
	jpss	noaa20	ahi		
	dmsp				

**geo.goes.g16.abi.adde.realtime.ncdf.image.complete**

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

geo	goes	g16	abi	adde	sdi	ncdf
leo	himawari	g18	glm	file	adde	hsf
	meteosat	h9	exis			hrit
	ews	m10	magnetometer			area
	sentinel	m9	seiss			
	metop	snpp	suvi			
	jpss	noaa20	ahi			
	dmsp					

**geo.goes.g16.abi.adde.realtime.ncdf.image.complete**

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

geo	goes	g16	abi	adde	sdi	ncdf	band
leo	himawari	g18	glm	file	adde	hsf	image
	meteosat	h9	exis			hrit	product
	ews	m10	magnetometer			area	granule
	sentinel	m9	seiss				
	metop	snpp	suvi				
	jpss	noaa20	ahi				
	dmsp						

**geo.goes.g16.abi.adde.realtime.ncdf.image.complete**

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

sat\_type.sat\_family.sat\_ID.sat\_instrument.medium.server\_type.format.classification.status

geo	goes	g16	abi	adde	sdi	ncdf	band	start
leo	himawari	g18	glm	file	adde	hsf	image	end
	meteosat	h9	exis			hrit	product	complete
	ews	m10	magnetometer			area	granule	
	sentinel	m9	seiss					
	metop	snpp	suvi					
	jpss	noaa20	ahi					
	dmsp							

**geo.goes.g16.abi.adde.realtime.ncdf.image.complete**

- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

satellite_family:	GOES	satellite_location:	GOES-East	coverage:	Mesoscale-1
satellite_ID:	G16	start_time:	2018-04-30 19:52:29.2	mode:	3
instrument:	ABI	end_time:	2018-04-30 19:52:35.0	signal_type:	grb
medium:	adde	create_time:	2018-04-30 19:52:39.3	data_type:	RadM1
server_type:	sdi	server_ip:	satbuf1.ssec.wisc.edu	title:	ABI L1b Radiances
message_type:	band	adde_dataset:	EASTA/M1		
status:	end	band:	3		
path:	/data/goes/grb/goes16/2018/2018_04_30_120/abi/L1b/RadM1/ OR_ABI-L1b-RadM1-M3C03_G16_s20181201952292_e20181201952350_c20181201952393.nc				

Satellite Data Services may add to the payload as needed. Backward compatibility is maintained unless key becomes obsolete. Users can request additions to the payload.



- amqpfind
  - General Information
  - Topic – set of subscription keys
  - Payload – value pairs of message
  - Monitor multiple RabbitMQ servers

```
./amqpfind -H mq2.ssec.wisc.edu -u sdsuser -p sdsmq \  
-H mq1.ssec.wisc.edu -u sdsuser -p sdsmq \  
-X satellite \  
-w 60.0 \  
-k '(start_time, coverage, satellite_ID) '  
-C 'geo.goes.*.*.*.*.*.*' \  
-j '{start_time} {coverage} {instrument} {message_type} {status}'
```

# Example Output

```
$HOME/amqpfind/amqpfind -X satellite\  
-H mq1.ssec.wisc.edu -p sdsdq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsdq -u sdsuser \  
-w 3600.0 \  
-k '(start_time, coverage, satellite_ID) \  
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \  
-j '{start_time} {server_ip} {adde_dataset} {coverage} {__reception_host__}'
```

```
2023-08-23 14:20:21.0 satbuf2.ssec.wisc.edu WESTA/FD Full Disk mq1.ssec.wisc.edu  
2023-08-23 14:20:20.5 satbuf2.ssec.wisc.edu EASTB/FD Full Disk mq2.ssec.wisc.edu  
2023-08-23 14:29:54.9 satbuf1.ssec.wisc.edu EASTA/M2 Mesoscale-2 mq2.ssec.wisc.edu  
2023-08-23 14:29:55.4 satbuf1.ssec.wisc.edu WESTA/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:30:27.8 satbuf2.ssec.wisc.edu EASTB/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:30:28.3 satbuf2.ssec.wisc.edu WESTA/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:30:54.9 satbuf2.ssec.wisc.edu EASTB/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:30:55.4 satbuf2.ssec.wisc.edu WESTA/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:31:24.9 satbuf2.ssec.wisc.edu EASTB/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:31:25.4 satbuf2.ssec.wisc.edu WESTA/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:31:54.9 satbuf2.ssec.wisc.edu EASTB/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:31:55.4 satbuf2.ssec.wisc.edu WESTA/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:32:25.4 satbuf2.ssec.wisc.edu WESTA/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:32:24.9 satbuf2.ssec.wisc.edu EASTB/M1 Mesoscale-1 mq2.ssec.wisc.edu  
2023-08-23 14:32:55.4 satbuf2.ssec.wisc.edu WESTA/M2 Mesoscale-2 mq2.ssec.wisc.edu  
2023-08-23 14:32:54.9 satbuf2.ssec.wisc.edu EASTB/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:33:24.9 satbuf2.ssec.wisc.edu EASTB/M1 Mesoscale-1 mq2.ssec.wisc.edu  
2023-08-23 14:33:25.4 satbuf1.ssec.wisc.edu WESTA/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:31:17.5 satbuf1.ssec.wisc.edu WESTA/CONUS CONUS mq1.ssec.wisc.edu  
2023-08-23 14:31:17.0 satbuf2.ssec.wisc.edu EASTB/CONUS CONUS mq2.ssec.wisc.edu
```

# Example Output

```
$HOME/amqpfind/amqpfind -X satellite\  
-H mq1.ssec.wisc.edu -p sdsdq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsdq -u sdsuser \  
-w 3600.0 \  
-k '(start_time, coverage, satellite_ID) \  
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \  
-j '{start_time} {server_ip} {adde_dataset} {coverage} {__reception_host__}' \  
grep --line-buffered -v Mesoscale
```

```
2023-08-23 14:20:21.0 satbuf2.ssec.wisc.edu WESTA/FD Full Disk mq1.ssec.wisc.edu  
2023-08-23 14:20:20.5 satbuf2.ssec.wisc.edu EASTB/FD Full Disk mq2.ssec.wisc.edu  
2023-08-23 14:29:54.9 satbuf1.ssec.wisc.edu EASTA/M2 Mesoscale-2 mq2.ssec.wisc.edu  
2023-08-23 14:29:55.4 satbuf1.ssec.wisc.edu WESTA/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:30:27.8 satbuf2.ssec.wisc.edu EASTB/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:30:28.3 satbuf2.ssec.wisc.edu WESTA/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:30:54.9 satbuf2.ssec.wisc.edu EASTB/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:30:55.4 satbuf2.ssec.wisc.edu WESTA/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:31:24.9 satbuf2.ssec.wisc.edu EASTB/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:31:25.4 satbuf2.ssec.wisc.edu WESTA/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:31:54.9 satbuf2.ssec.wisc.edu EASTB/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:31:55.4 satbuf2.ssec.wisc.edu WESTA/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:32:25.4 satbuf2.ssec.wisc.edu WESTA/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:32:24.9 satbuf2.ssec.wisc.edu EASTB/M1 Mesoscale-1 mq2.ssec.wisc.edu  
2023-08-23 14:32:55.4 satbuf2.ssec.wisc.edu WESTA/M2 Mesoscale-2 mq2.ssec.wisc.edu  
2023-08-23 14:32:54.9 satbuf2.ssec.wisc.edu EASTB/M2 Mesoscale-2 mq1.ssec.wisc.edu  
2023-08-23 14:33:24.9 satbuf2.ssec.wisc.edu EASTB/M1 Mesoscale-1 mq2.ssec.wisc.edu  
2023-08-23 14:33:25.4 satbuf1.ssec.wisc.edu WESTA/M1 Mesoscale-1 mq1.ssec.wisc.edu  
2023-08-23 14:31:17.5 satbuf1.ssec.wisc.edu WESTA/CONUS CONUS mq1.ssec.wisc.edu  
2023-08-23 14:31:17.0 satbuf2.ssec.wisc.edu EASTB/CONUS CONUS mq2.ssec.wisc.edu
```

# Launching Wrapper Script

```
$HOME/amqpfind/amqpfind -X satellite\  
-H mq1.ssec.wisc.edu -p sdsmq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsmq -u sdsuser \  
-w 3600.0 \  
-k '(start_time, coverage, satellite_ID)\'\  
-C 'geo.goes.*.abi.adde.realtime.*.image.complete' \  
-j '{start_time} {server_ip} {adde_dataset}' |\  
grep --line-buffered -v Mesoscale |\  
xargs -P1 -n4 python $HOME/bin/mc-example.py
```

```
$HOME/bin/mc-example.py 2023-08-23 14:20:21.0 satbuf2.ssec.wisc.edu WESTA/FD
```

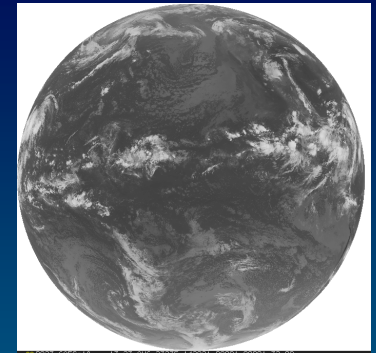
# Launching Wrapper Script

```
#!/usr/bin/python
def main(cmdArgs):
    import mcidasx
    import datetime
    imageDate = cmdArgs[1]
    imageTime = cmdArgs[2]
    server = cmdArgs[3]
    addeDataset = cmdArgs[4]
    group = addeDataset.split('/')[0]

    mcenv = mcidasx.mcenv(f='3@750x1250', i=228, g=8)

    mccmdout = mcenv.dataloc('ADD ' + group + ' ' + server)
    dateTimeStr = imageDate + ' ' + imageTime
    dateTimeObj = datetime.datetime.strptime(dateTimeStr,'%Y-%m-%d %H:%M:%S.%f')
    dateTimeKeyword = dateTimeObj.strftime(' DAY=%Y%j TIME=%H:%M:%S ')
    mcCmd = ' USER 1234'
    mcCmdOut = mcenv.logon(mcCmd)
    mcCmd = addeDataset + dateTimeKeyword + ' BAND=13'
    mcCmdOut = mcenv.imgdisp(mcCmd)
    mcCmd = ' X current-G16-band13.gif'
    mcCmdOut = mcenv.frmsave(mcCmd)

if __name__ == '__main__':
    import sys
    main(sys.argv)
```



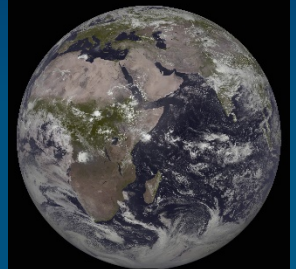
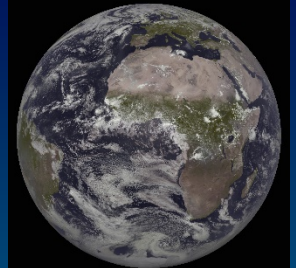
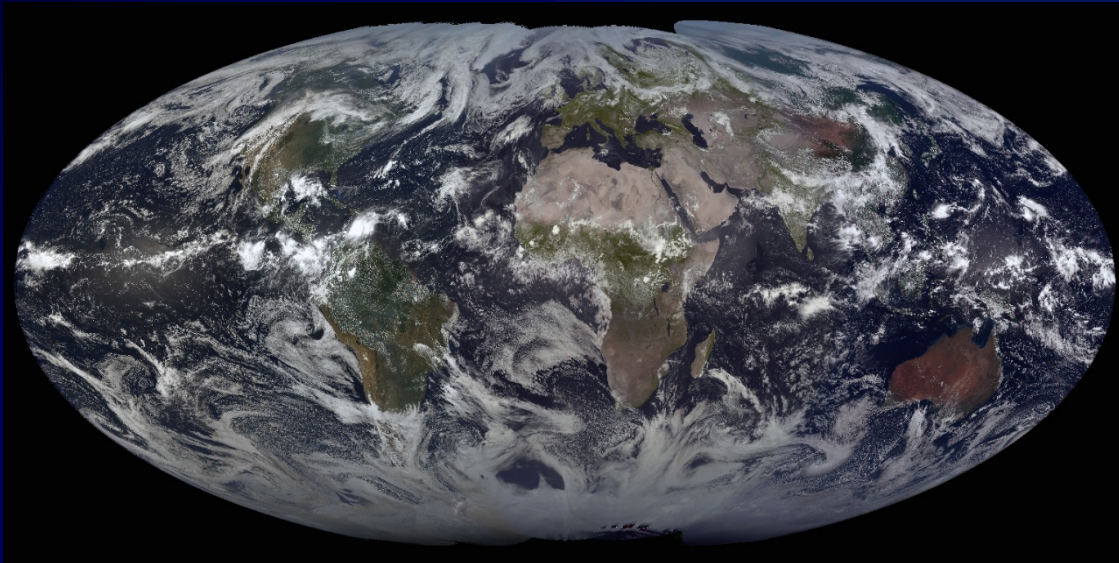
```
2023-08-23 14:20:21.0 satbuf2.ssec.wisc.edu WESTA/FD
DATALOC ADD WESTA/FD satbuf2.ssec.wisc.edu
LOGON USER 1234
IMGDISP WESTA/FD X TIME=14:20:21 DAY=2023-08-23 BAND=13
```



# Multiple satellite example

```
$HOME/amqpfnd/amqpfnd -X satellite -H mq1.ssec.wisc.edu -p sdsdq -u sdsuser \  
-H mq2.ssec.wisc.edu -p sdsdq -u sdsuser -t 0 -w 3600 -i 'Local Noon Images' \  
-C 'geo.*.*.adde.realtime.*.image.complete' -k '(start_time, satellite_ID, title)' \  
-j '{start_time} {server_ip} {adde_dataset} {satellite_ID} {title}' \  
| grep --line-buffered -i FD | xargs -P1 -n 6 rgb_script.py
```

```
2023-08-23 15:40:00.0 satbuf3.ssec.wisc.edu HIMAWARI/FD H09 Himawari-H09 AHI Full Disk  
2023-08-23 15:50:21.0 satbuf2.ssec.wisc.edu WESTA/FD G18 ABI L1b Radiances  
2023-08-23 15:50:20.5 satbuf2.ssec.wisc.edu EASTB/FD G16 ABI L1b Radiances  
2023-08-23 16:00:21.0 satbuf2.ssec.wisc.edu WESTA/FD G18 ABI L1b Radiances  
2023-08-23 16:00:20.5 satbuf2.ssec.wisc.edu EASTB/FD G16 ABI L1b Radiances  
2023-08-23 16:00:00.0 indoex.ssec.wisc.edu IODCEND/FD M9 MSG Full Disk Image complete  
2023-08-23 16:00:00.0 msg.ssec.wisc.edu MSGEND/FD M10 MSG Full Disk Image complete  
2023-08-23 15:50:00.0 satbuf3.ssec.wisc.edu HIMAWARI/FD H09 Himawari-H09 AHI Full Disk  
2023-08-23 15:30:00.0 indoex.ssec.wisc.edu IODCEND/FD M9 MSG Full Disk Image complete  
2023-08-23 16:10:20.5 satbuf2.ssec.wisc.edu EASTB/FD G16 ABI L1b Radiances  
2023-08-23 16:10:21.0 satbuf2.ssec.wisc.edu WESTA/FD G18 ABI L1b Radiances  
2023-08-23 16:00:00.0 satbuf3.ssec.wisc.edu HIMAWARI/FD H09 Himawari-H09 AHI Full Disk  
2023-08-23 15:30:00.0 msg.ssec.wisc.edu MSGEND/FD M10 MSG Full Disk Image complete  
2023-08-23 15:45:00.0 msg.ssec.wisc.edu MSGEND/FD M10 MSG Full Disk Image complete
```

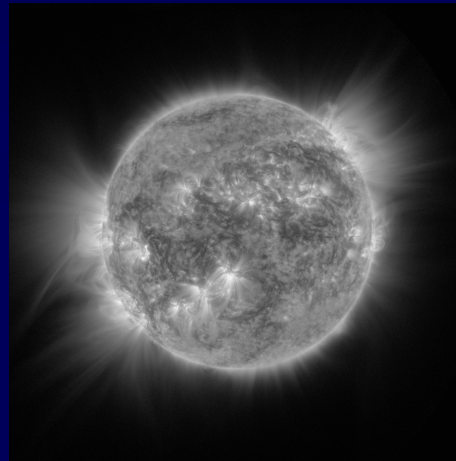


# SUVI example – Step 1 create png

```
$HOME/amqpfind/amqpfind -X satellite -H mq1.ssec.wisc.edu -u sdsuser -p sdsmq \  
-H mq2.ssec.wisc.edu -u sdsuser -p sdsmq -t 0 -w 3600 -k '(start_time, coverage)' \  
-C geo.goes.*.suvi.*.*.image.complete -j '{path}{server_ip}{exposure}' | \  
grep --line-buffered long | HereGOES_script.py
```

```
/suvi/L1b/Fe093/OR_SUVI-L1b-Fe093_G18_s20232351614391_e20232351614391_c20232351614590.nc satbuf2.ssec.wisc.edu short_flare  
/suvi/L1b/Fe171/OR_SUVI-L1b-Fe171_G18_s20232351614491_e20232351614501_c20232351615096.nc satbuf2.ssec.wisc.edu long  
/suvi/L1b/Fe195/OR_SUVI-L1b-Fe195_G16_s20232351614528_e20232351614528_c20232351615126.nc satbuf2.ssec.wisc.edu short_flare  
/suvi/L1b/Fe195/OR_SUVI-L1b-Fe195_G18_s20232351615091_e20232351615101_c20232351615294.nc satbuf2.ssec.wisc.edu long  
/suvi/L1b/Fe093/OR_SUVI-L1b-Fe093_G16_s20232351615128_e20232351615128_c20232351615320.nc satbuf1.ssec.wisc.edu short  
/suvi/L1b/Fe284/OR_SUVI-L1b-Fe284_G18_s20232351615191_e20232351615201_c20232351615395.nc satbuf1.ssec.wisc.edu long
```

HereGOES software to create png from netCDF file



When complete, a new message is published

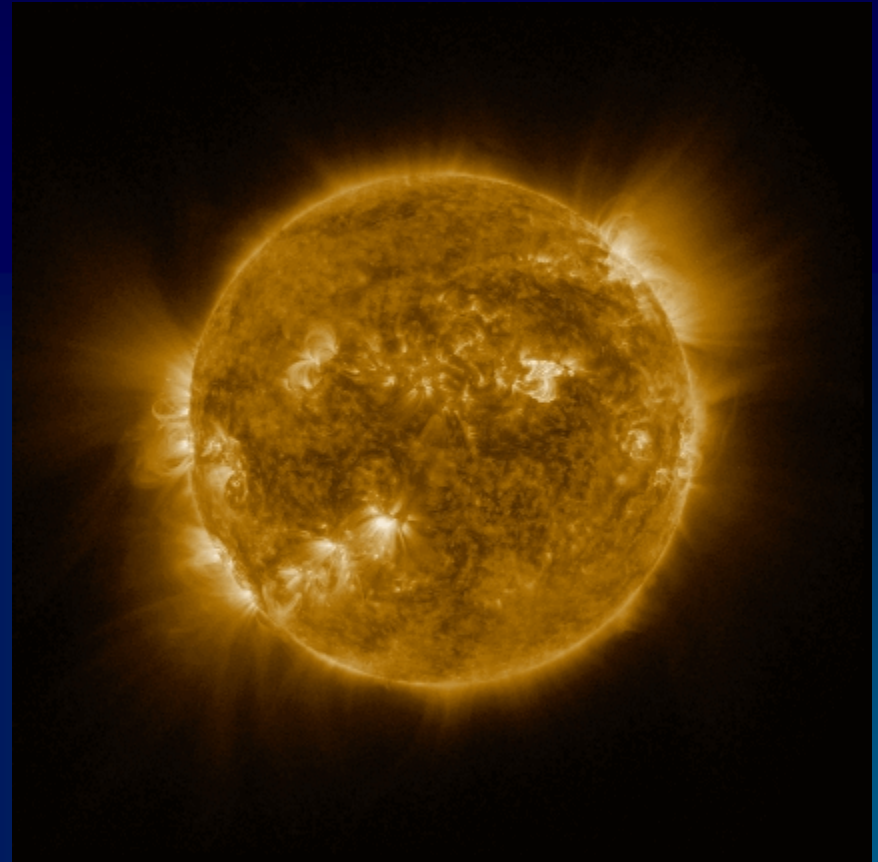
# SUVI example – Step 2 create web graphic

```
$HOME/amqpfind/amqpfind -X satellite -H mq2.ssec.wisc.edu -u sdsuser -p sdsmq \  
-H mq1.ssec.wisc.edu -u sdsuser -p sdsmq -i "SUVI Images" -w 3600 -t 0 -k '(start_time,data_type)' \  
-C geo.goes.*.suvi.*.png.*.*' | grep --line-buffered long | web_script.py
```

```
2023-08-23 17:17:32.900000 Fe195 long data_path/OR_SUVI-L1b-Fe195-L_G16_s20232351717329_e20232351717339_c20232351717531.png  
2023-08-23 17:17:59.200000 Fe195 long data_path/OR_SUVI-L1b-Fe195-L_G18_s20232351717592_e20232351718003_c20232351718202.png  
2023-08-23 17:18:02.900000 He303 long data_path/OR_SUVI-L1b-He303-L_G16_s20232351718029_e20232351718039_c20232351718237.png
```

## Scripting method:

- cp png locally (only long exposure)
- covert png to tif ([ImageMagik](#))
- tif to AREA (MAKEIMG)
- Apply wavelength specific enhancement (XRD)
- Send to SSEC [geo-browser](#)

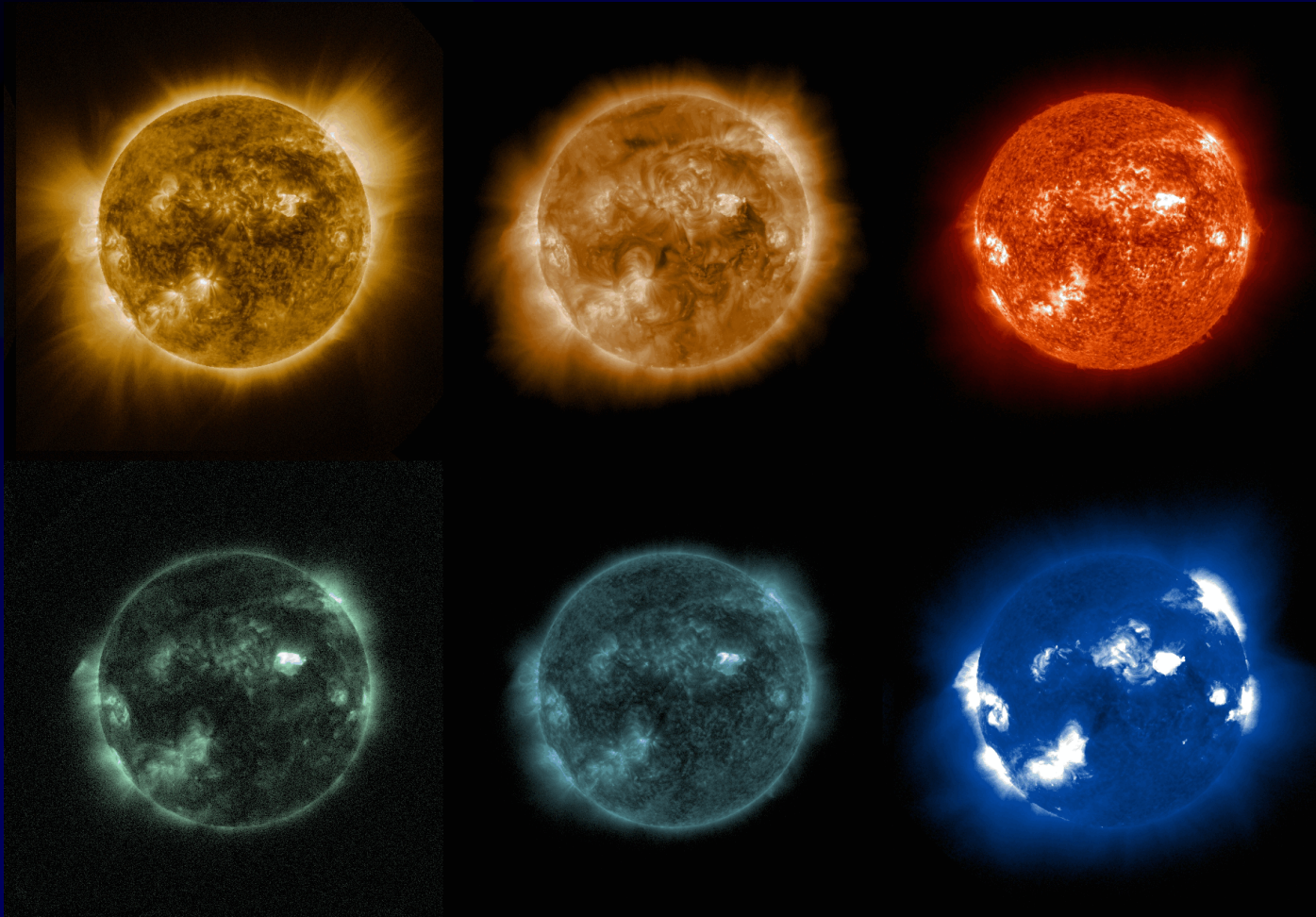




# SUVI example – Step 3 Bonus!

## Script continues:

Create RGB multi-banded AREA from original AREA and enhancement table; .ET file (XRD IMGEU)  
scp AREA files to ADDE server  
Publish another message when scp is complete





**Thank You!**

**Questions?**