

GRIB1 → GRIB2

Sebastien Villaume

Data Governance

Sebastien.Villaume@ecmwf.int



What is GRIB?

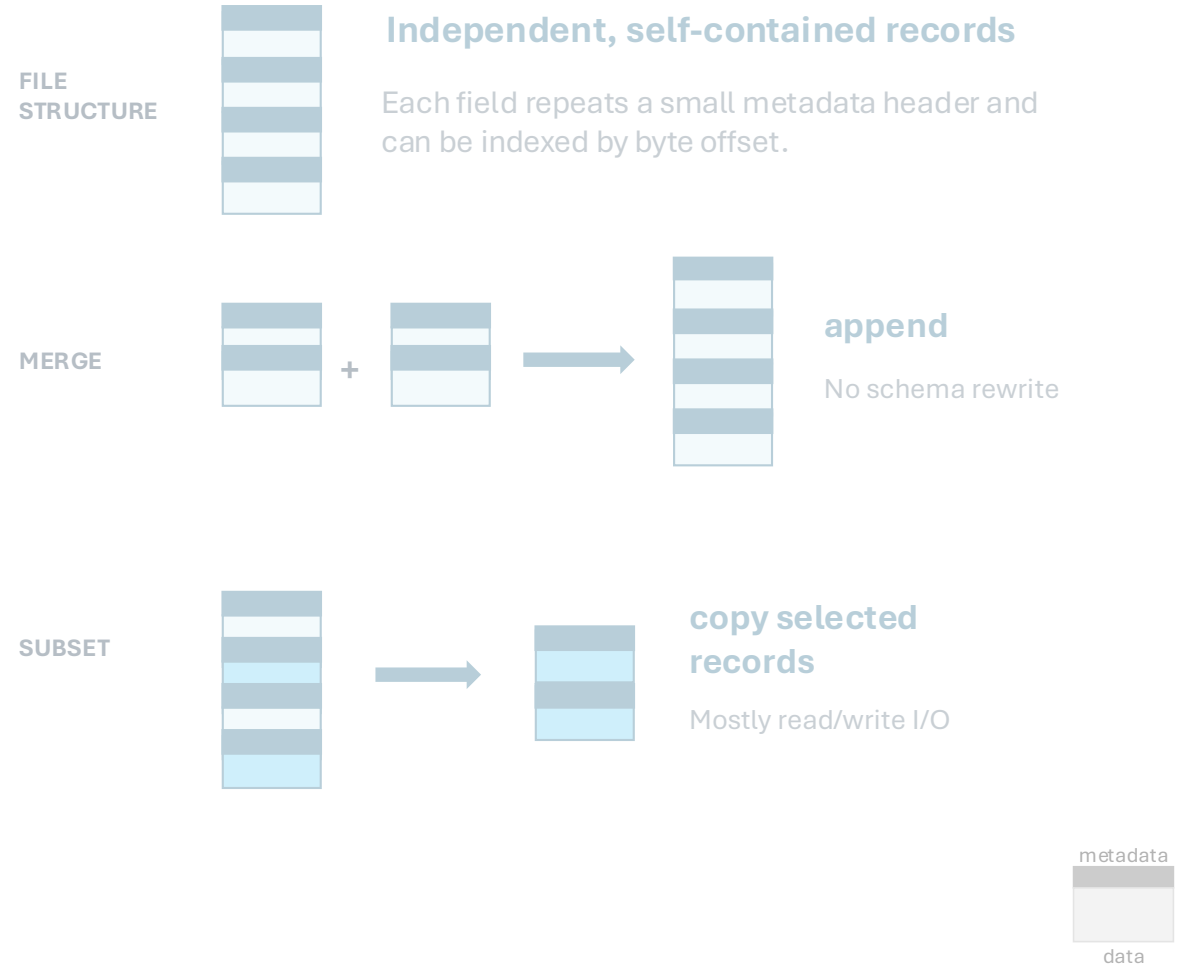
GRIB: a machine-oriented data format

- record based binary format maintained by WMO for exchange of data between Met services
- designed for meteorological data only
- **originally not meant to be used by humans:** very strict, not forgiving, table driven, complex, unfriendly
- easy to parse and process by machines and very portable
- usually aggregated by model timestep
- exists in 2 editions, GRIB1 (1990-2005) and GRIB2 (2005-)

GRIB Properties

GRIB: a machine-oriented data format

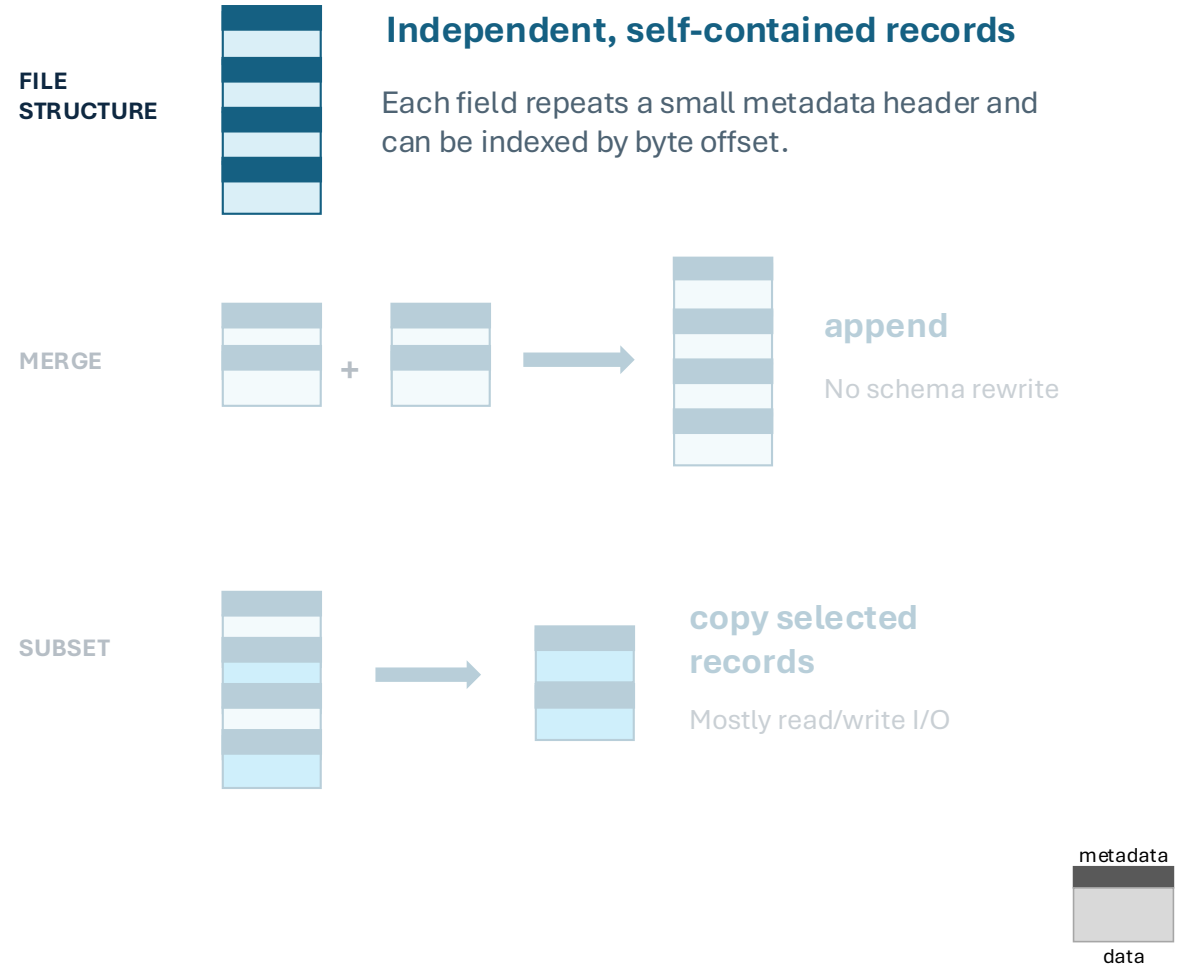
- record based binary format maintained by WMO for exchange of data between Met services
- designed for meteorological data only
- **originally not meant to be used by humans:** very strict, not forgiving, table driven, complex, unfriendly
- easy to parse and process by machines and very portable
- usually aggregated by model timestep
- exists in 2 editions, GRIB1 (1990-2005) and GRIB2 (2005-)



GRIB Properties

GRIB: a machine-oriented data format

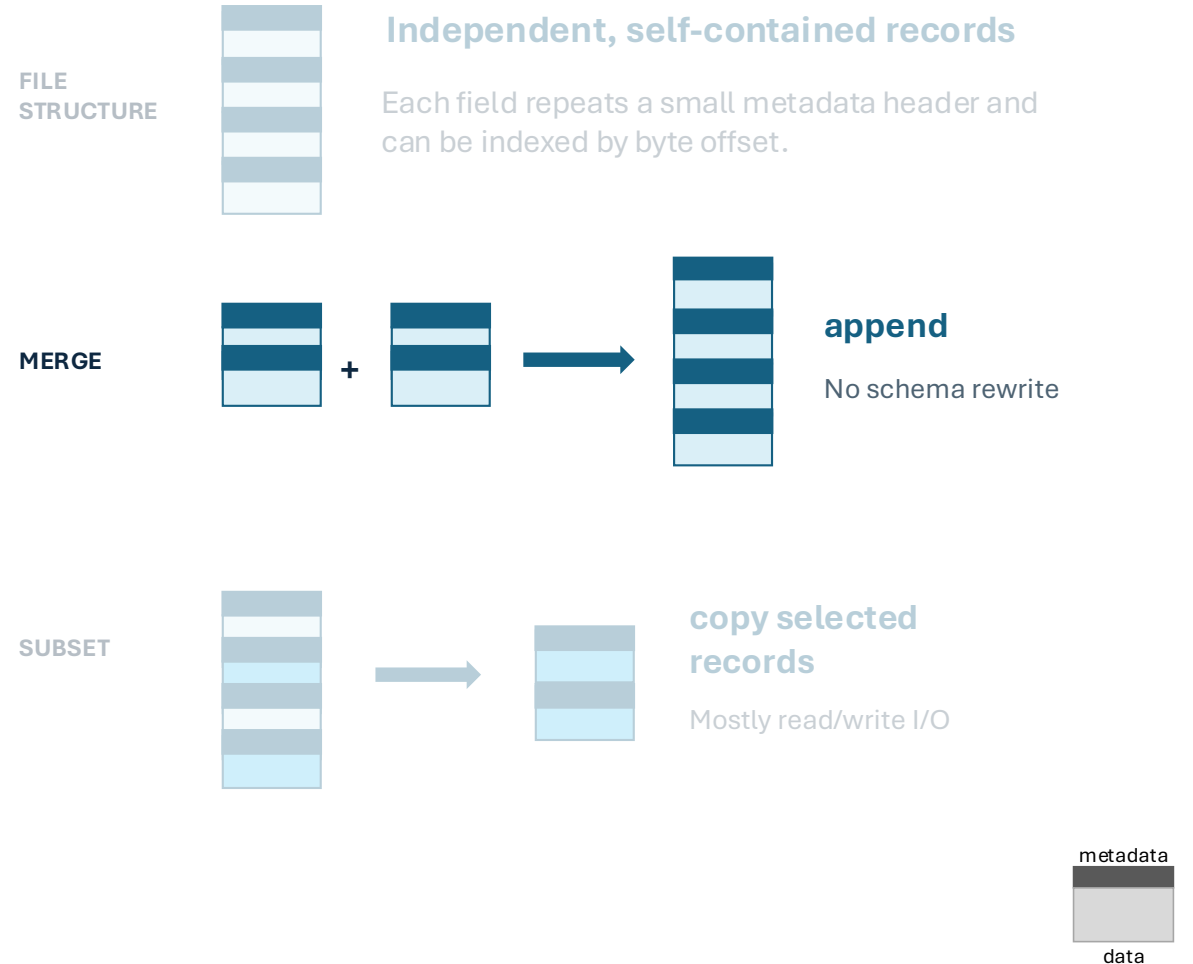
- record based binary format maintained by WMO for exchange of data between Met services
- designed for meteorological data only
- **originally not meant to be used by humans:** very strict, not forgiving, table driven, complex, unfriendly
- easy to parse and process by machines and very portable
- usually aggregated by model timestep
- exists in 2 editions, GRIB1 (1990-2005) and GRIB2 (2005-)



GRIB Properties

GRIB: a machine-oriented data format

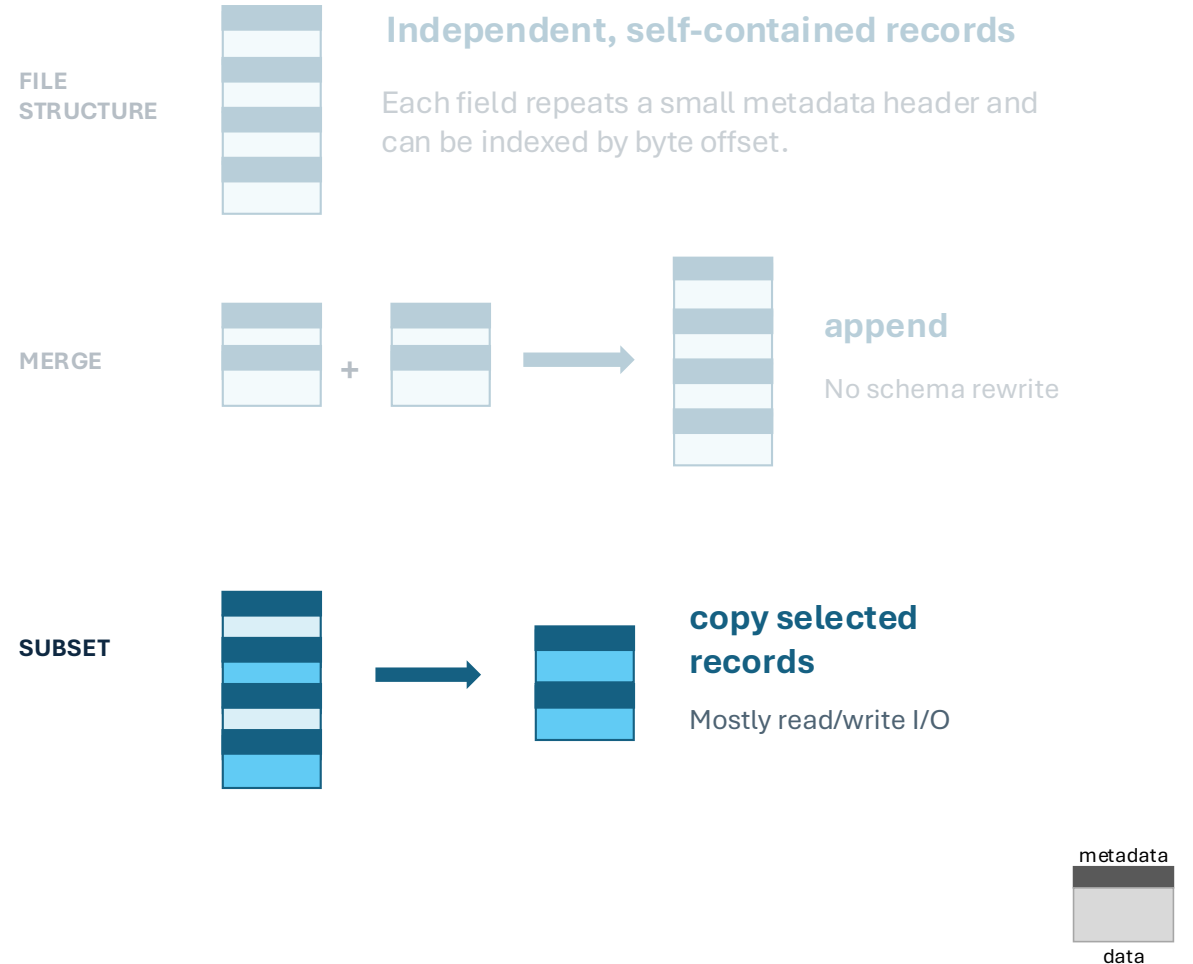
- record based binary format maintained by WMO for exchange of data between Met services
- designed for meteorological data only
- **originally not meant to be used by humans:** very strict, not forgiving, table driven, complex, unfriendly
- easy to parse and process by machines and very portable
- usually aggregated by model timestep
- exists in 2 editions, GRIB1 (1990-2005) and GRIB2 (2005-)



GRIB Properties

GRIB: a machine-oriented data format

- record based binary format maintained by WMO for exchange of data between Met services
- designed for meteorological data only
- **originally not meant to be used by humans:** very strict, not forgiving, table driven, complex, unfriendly
- easy to parse and process by machines and very portable
- usually aggregated by model timestep
- exists in 2 editions, GRIB1 (1990-2005) and GRIB2 (2005-)



GRIB1 and GRIB2: message structure

GRIB1

6 sections (0–5)

0	Indicator
1	Product definition Identification + product + optional local data
2	Grid definition [T] (optional)
3	Bitmap (optional)
4	Binary data Data Representation + values
5	End (“7777”)

WHAT CHANGES

Separate identification
and local-use sections

Packing description
separate from values

More sections are
Template-based [T]

GRIB2

9 sections (0–8)

0	Indicator
1	Identification [T]
2	Local data [T] (optional)
3	Grid definition [T]
4	Product definition [T]
5	Data representation [T] (packing)
6	Bitmap*
7	Data
8	End (“7777”)

GRIB1 limitations: resolution

Vertical model levels

127 model levels

The number of vertical-coordinate coefficients limits the hybrid-level description.

IFS's change to 137 levels required GRIB2 for model-level output. (2011)

Horizontal grids

0.001° precision

GRIB1 encodes latitude and longitude in millidegrees. This limits accurate descriptions of fine grids.

Number of grid encoded on 2 octets adds a further constraint on points along each axis.

A further increase in ECMWF's grid resolution would exceed GRIB1's encoding limits.

GRIB1 limitations: metadata and interoperability

A narrow parameter namespace

1 octet + 1 octet

The parameter identifier has only 256 code values, split between standard and local use. The versioning of that table is also encoded on 1 octet

ECMWF hacked the "table version" to encode multiple **local** tables to describe a much broader set of parameters.

Modern products need richer metadata

Local extensions

Ensembles, reforecasts and detailed processing metadata exceed the original GRIB1 design.

ECMWF adds this information inside Section 1. Other centres need the corresponding local definitions to decode the information.

Guiding principles of the migration

Scope

- Migrate all parameters
- Only new data produced, no conversion of existing data
- Migrate our entire software stack: I/O routines in models, interpolation, plotting, indexing, archiving, dissemination

Requirements

- Minimise disruption to our operations and to our users
- Maintain consistency with existing data
- Provide tools to ease the migration
- Take advantage of the rich metadata
- Support Member States
- Fully WMO compliant, avoid local encoding

Migration cycles timeline

Before the migration

Up to cycle 49r1

- Extend GRIB2 with templates and parameters
- Prepare ecCodes definitions for the necessary changes

During the migration

Cycles 49r2, 50r1 and 50r2

- Mixture of GRIB1/GRIB2 and full GRIB2 handled correctly in parallel
- Software pipeline deals with data from before/after the migration

After the migration

From cycle 51r1

- Continue to handle properly the old data

Migration preparation: New templates, new parameters

Templates

~ 100 new templates added

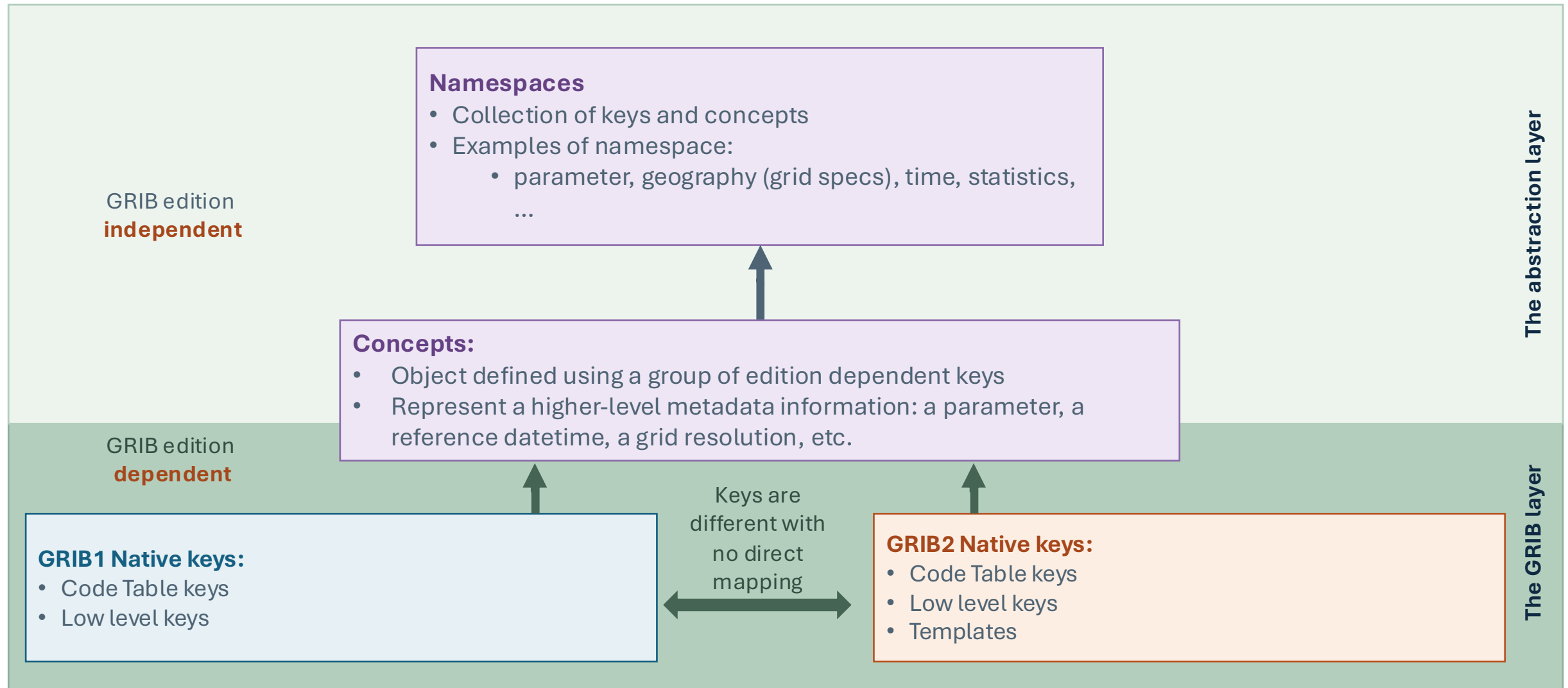
- Wave spectra (frequency/direction) and wave period templates
- Anomalies, EFI, SOT templates
- Tile templates for land surface modelling
- Probability templates with focal statistics
- Quantile based templates
- Large ensembles (>255 members)

Parameters

~200 new parameters added

- Heat stress
- Soil/snow/ice model levels/layers
- Hydrological parameters
- Ocean, ice and waves
- Destination Earth and ERA6 parameters

Mapping strategy across GRIB editions: the abstraction layer in ecCodes



Provides a unified view of GRIB messages

- The edition becomes irrelevant
- Improves systematic indexing and searchability

Hides the underlying metadata and its complexity

- Easier for users
- The design of application is easier and straightforward
 - No need for conditional handling based on the GRIB edition

Extensible to other data formats

- The low-level native GRIB keys could be replaced by
 - netCDF variables, attributes, dimensions
 - geoJSON objects and elements
 - <any other format>

Temperature

GRIB1 (ECMWF local)

table2Version = 128

indicatorOfParameter = 130 (Temperature)

- always 2 keys
- No hierarchy → an entry in a table



GRIB2 (WMO)

discipline = 0 (meteorology)

parameterCategory = 0 (temperatures)

parameterNumber = 0 (temperature)

- Minimum of 3 keys
- Hierarchical:
 - Parameters within categories and disciplines

A more Advanced example

Probability of accumulated **total precipitation** at surface of at least 25mm

GRIB1 (ECMWF local)

table2Version = 131
indicatorOfParameter = 98
timeRangeIndicator=4
(accumulation)



GRIB2 (WMO)

productDefinitionTemplateName = 9 (probability)
discipline = 0 (meteorology)
parameterCategory = 1 (Moisture)
parameterNumber = 52 (Total* precipitation rate)
typeOfFirstFixedSurface = 1 (surface)
typeOfStatisticalProcessing = 1 (Accumulation)
probabilityType = 3 (above lower limit)
scaledValueOfLowerLimit = 25
scaleFactorOfLowerLimit = 0 (no scaling)

In ecCodes, these 2 representations share:

- the same paramId: **131098**
- the same shortName: **tpg25**

What are the relevant concepts?

PROVENANCE

Producer

Dataset

Model and version

FORECAST & PROCESS

Forecasting system

Processes

others?

COORDINATES

Time

Vertical level

Horizontal grid

PARAMETER & SAMPLING

Parameter/
observable

Statistical/sampling
modifier

Additional parameter
information:
chemical,
wavelength, etc.

01

Properly define all our existing parameters in GRIB2

02

Identify and group relevant sets of GRIB keys into concepts across editions to create the abstraction layer

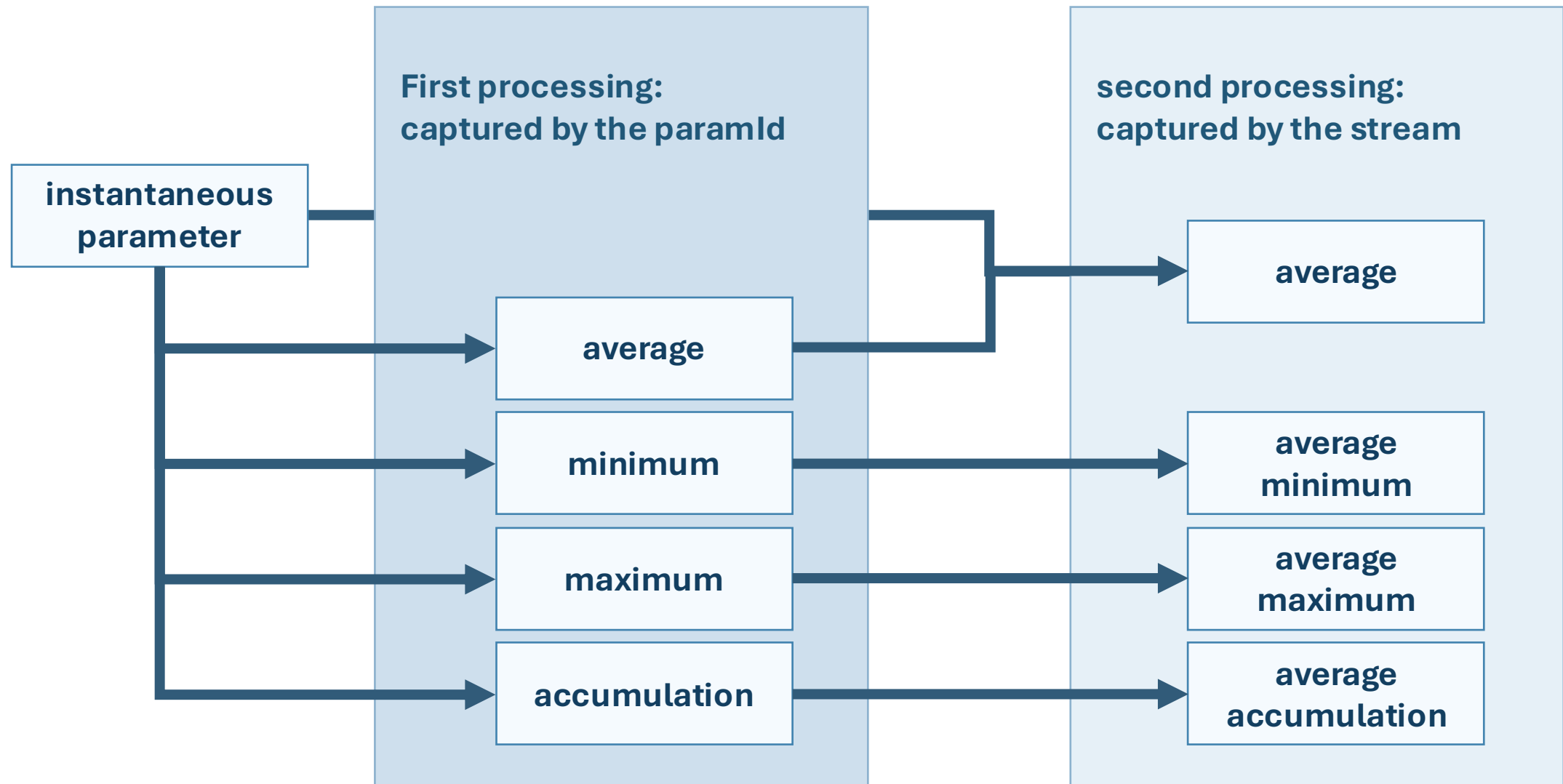
03

Implement these concepts in tools, scripts, workflows

04

improve the MARS language to enable the full potential of the metadata

The GRIB2 data model for parameters: nested time processing



Examples: 2 metres temperature and total precipitation

Temperature

discipline = 0 (meteorology)
parameterCategory = 0 (temperatures)
parameterNumber = 0 (temperature)

Maximum Temperature

discipline = 0 (meteorology)
parameterCategory = 0 (temperatures)
parameterNumber = 0 (temperature)
typeOfStatisticalProcessing = 2
(maximum)

Mean of Maximum Temperature

discipline = 0 (meteorology)
parameterCategory = 0 (temperatures)
parameterNumber = 0 (temperature)
typeOfStatisticalProcessing = [0,2]
(average, maximum)

Total precipitation rate

discipline = 0 (meteorology)
parameterCategory = 1 (moisture)
parameterNumber = 52 (total precipitation rate)

Total precipitation (Accumulated Total precipitation rate)

discipline = 0 (meteorology)
parameterCategory = 1 (moisture)
parameterNumber = 52 (total precipitation rate)
typeOfStatisticalProcessing = 1
(accumulation)

Monthly mean of total precipitation

discipline = 0 (meteorology)
parameterCategory = 1 (moisture)
parameterNumber = 52 (total precipitation rate)
typeOfStatisticalProcessing = [0,1]
(average, accumulation)



Standardisation of shortName and time post-processing

Time processing, described by three distinct fields

shortName

Processing type

A prefix identifies the first level of time processing.

2t → min_2t max_2t avg_2t

timespan

First processing interval

A new MARS keyword records the span of that first level.

1h

Hourly

3h

3-hourly

from-start

Special case for accumulations

stattype

Additional processing

A new MARS keyword captures any further processing type and length.

Monthly means or maxima

Also weekly means, daily means and 5-day maxima

Standardisation of shortName and time post-processing

Time processing, described by three distinct fields

shortName

Processing type

A prefix identifies the first level of time processing.

2t → min_2t max_2t avg_2t

timespan

First processing interval

A new MARS keyword records the span of that first level.

1h

Hourly

3h

3-hourly

from-start

Special case for accumulations

stattype

Additional processing

A new MARS keyword captures any further processing type and length.

Monthly means or maxima

Also weekly means, daily means and 5-day maxima

Standardisation of shortName and time post-processing

Time processing, described by three distinct fields

shortName

Processing type

A prefix identifies the first level of time processing.

2t → min_2t max_2t avg_2t

timespan

First processing interval

A new MARS keyword records the span of that first level.

1h

Hourly

3h

3-hourly

from-start

Special case for accumulations

stattype

Additional processing

A new MARS keyword captures any further processing type and length.

Monthly means of maxima -> momx

Also weekly means, daily means and 5-day maxima

The interval moves into timespan

Existing format

step=0-3/3-6/6-9



With timespan

step=3/6/9
timespan=3h

step=0-6/6-12/12-18



step=6/12/18
timespan=6h

step=0-6/0-12/0-18



step=6/12/18
timespan=from-start

An extensible structure for future time processing

Handling legacy encoding (ongoing work)

GRIB2 prescribes the units of parameters

- Precipitation in **kg m⁻²** in GRIB2 vs **m** in 50r1
- Cloud cover in **%** in GRIB2 vs **fraction [0-1]** in 50r1

Conversion is luckily straightforward

- 1 kg m⁻² ~ 1mm
- Conversion factor ~10³

In GRIB1, instantaneous paramIds have been used to encode monthly means -> we **break** this practice

- Special handling to continue to decode these parameters correctly
- Prevent writing more of these: some parameters become **read-only**

MARS short names: one request, alternative parameter IDs

ONGOING DESIGN

Unit changes can retain the same short name

USER REQUEST

param=tp

Without a units request,
data is returned as archived.



ALTERNATIVES EXPANSION

param=228 | 228228

228: ECMWF/legacy units (m)
228228: WMO/SI units (kg m⁻²)

The alternatives count as one requested parameter in the request

Exactly one form may exist for the same context. Both forms present is an error.

Retrieve and list requests can use the shared short name. Archive requests must use the numeric paramId.

MARS units: explicit conversion, with per-parameter control

ONGOING DESIGN

Select the data first, then choose the returned units

param=tp
units=av

Keep archived values

The default when units is omitted. No conversion.

param=tp
units=si/percent

Request standard units

Apply known SI or percentage conversions and scale values.

param=tp/z/u/v
units=si[tp]

Convert only selected parameters

Convert tp to SI. Leave z, u and v unchanged (as archived).

No conversion from SI back to legacy local units.

discontinued from 50r2

Move regular wave parameters and 2D wave spectra into respective atm streams:

- wave -> oper , waef -> enfo, etc.

Still logically separated from atmo parameters in the MARS catalogue using a **paramtype** node in the MARS catalogue

- paramtype = base -> atmospheric parameters
- paramtype = wave -> regular wave parameters
- paramtype = wave_spectra -> 2D wave spectra

Other use of paramtype: chemical , optical , tile , etc.

paramtype is not required in MARS requests.

It is used only in the MARS tree to organise the data logically.

discontinued from 50r2

Move regular wave parameters and 2D wave spectra into respective atm streams:

- wave -> oper , waef -> enfo, etc.

Still logically separated from atmo parameters in the MARS catalogue using a **paramtype** node in the MARS catalogue

- paramtype = base -> atmospheric parameters
- paramtype = wave -> regular wave parameters
- paramtype = wave_spectra -> 2D wave spectra

Other use of paramtype: chemical , optical , tile , etc.

paramtype is not required in MARS requests.

It is used only in the MARS tree to organise the data logically.

discontinued from 50r2

Move regular wave parameters and 2D wave spectra into respective atm streams:

- wave -> oper , waef -> enfo, etc.

Still logically separated from atmo parameters in the MARS catalogue using a **paramtype** node in the MARS catalogue

- paramtype = base -> atmospheric parameters
- paramtype = wave -> regular wave parameters
- paramtype = wave_spectra -> 2D wave spectra

Other use of paramtype: chemical , optical , tile , etc.

paramtype is not required in MARS requests.

It is used only in the MARS tree to organise the data logically.

discontinued from 50r2

Move regular wave parameters and 2D wave spectra into respective atm streams:

- wave -> oper , waef -> enfo, etc.

Still logically separated from atmo parameters in the MARS catalogue using a **paramtype** node in the MARS catalogue

- paramtype = base -> atmospheric parameters
- paramtype = wave -> regular wave parameters
- paramtype = wave_spectra -> 2D wave spectra

Other use of paramtype: chemical , optical , tile , etc.

paramtype is not required in MARS requests.

It is used only in the MARS tree to organise the data logically.

ChemId for atmospheric composition

The number of paramIds used in atmospheric composition is **problematic**

Conceptually these parameters are 2-dimensional:

- A physical property -> mass mixing ratio, emission rate, etc.
- chemical species and aerosols -> CH₄, CO₂

We are splitting the existing paramIds into a **pair of keywords**:

- **paramId** : now only representing the **property**
- **chemId** : representing the **chemical species and aerosols**
- Example: param=mmr, chem=CH₄

We also considered extracting the emission sources into a separate keyword but decided **not** to do so

- Example: param=emission from volcanoes vs ~~param=emission, source=volcanoes~~

Wavelength keyword for optical parameters

We are introducing a new **wavelength** keyword to capture the dependency of the **optical parameters**.

This will have an impact on the existing optical parameters, particularly in atmospheric composition:

- optical depths, etc.
- Parameters “at XXX nm”

It will support other IFS developments: simulated satellite images, etc.

- Wavelength ranges
- Colour channels
- etc.

We are introducing 2 new ecCodes helper keys to enrich the metadata:

- `modelName` (IFS, AIFS, EFAS, NEMO, etc.)
- `modelVersion` (cy50r2, v2.0, etc.)

In specific cases, these can be used as a MARS keyword (model)

- Land Data Assimilation System (LDAS)
- Hydrology models
- wildfire prediction models
- Etc.

Tools to support the migration

Reference Static Dataset

- First prototype available – will be incrementally updated to cover all operational data.
- Available on the HPC: /ec/vol/marsdev4/MTG2_sample_dataset
- Available at https://data.ecmwf.int/mtg2_sample/

MARS request translator (soon)

- Provide a tool to help users migrate their requests
- Support conversion of dissemination requests

Robust conversion tool: GRIB1 -> GRIB2 (soon)

- We will **not** support the reverse conversion

We are available for technical discussions to answer and help member states teams and commercial customers working on the migration.

Request conversion

Original request

```
retrieve,  
  class    = od,  
  type     = fc,  
  stream   = wave,  
  expver   = 0001,  
  levtype  = sfc,  
  date     = 20260901,  
  time     = 0000,  
  step     = 0/to/24/by/6,  
  param    = swh/mwp/mwd/pp1d/wind,  
  grid     = 0.25/0.25,  
  target   = wave.grib
```



Converted request

```
retrieve,  
  class    = od,  
  type     = fc,  
  stream   = oper,  
  expver   = 0001,  
  levtype  = sfc,  
  date     = 20260901,  
  time     = 0000,  
  step     = 0/to/24/by/6,  
  param    = swh/mwp/mwd/pp1d/wind,  
  grid     = 0.25/0.25,  
  target   = wave.grib,  
  timespan = none
```

Request conversion

Original request

Original

```
retrieve,
class = od,
type = pf,
stream = enfo,
expver = 0001,
levtype = sfc,
number = 1/to/50,
date = 20260901,
time = 0000,
step = 0/to/48/by/6,
param =
2t/tcc/sd/ptype/10fg6/mx2t
6/mn2t6/swv11/stl1/tp,
grid = 0.25/0.25,
target = "ens.grib"
```



Four converted requests

1 Surface, instantaneous

```
retrieve,
class = od,
type = pf,
stream = enfo,
expver = 0001,
levtype = sfc,
number = 1/to/50,
date = 20260901,
time = 0000,
step = 0/to/48/by/6,
param =
2t/tcc/sd/ptype,
grid = 0.25/0.25,
target = ens.grib,
timespan = none
```

2 Surface, time statistics

```
retrieve,
class = od,
type = pf,
stream = enfo,
expver = 0001,
levtype = sfc,
number = 1/to/50,
date = 20260901,
time = 0000,
step = 6/to/48/by/6,
param =
max_i10fg/max_2t/min_2t,
grid = 0.25/0.25,
target = ens.grib,
timespan = 6h
```

3 Soil fields, instantaneous

```
retrieve,
class = od,
type = pf,
stream = enfo,
expver = 0001,
levtype = sol,
levelist = 1,
number = 1/to/50,
date = 20260901,
time = 0000,
step = 0/to/48/by/6,
param = vsw/sot,
grid = 0.25/0.25,
target = ens.grib,
timespan = none
```

4 surface, accumulation

```
retrieve,
class = od,
type = pf,
stream = enfo,
expver = 0001,
levtype = sfc,
number = 1/to/50,
date = 20260901,
time = 0000,
step = 0/to/48/by/6,
param = tp,
grid = 0.25/0.25,
target = ens.grib,
timespan = fs
```

Thank you

