HDF5

Métadonnées et compression pour grosses données

Mathieu Gaborit

Human Talks – La Ruche Numérique – Oct. 2017

La Ruche Numérique

HAUM

$ whoami

Mathieu (matael) Gaborit

- Thésard au LAUM (Le Mans) & au KTH (Stockholm)
- Co-fondateur du HAUM (haum.org)
- Co-fondateur de sample.cat (un peu en stand-by malheureusement)
- Hacker, mangeur de données, curieux, *etc.*.

On génère des données… Plein.

On génère des données… Plein.

Parfois on les traite tout de suite,

On génère des données… Plein.

Parfois on les traite tout de suite,

souvent on revient dessus beaucoup plus tard.

Quelque soit l'usage

- expériences/manips
- simulations
- potager connecté
- logs
- *etc.*

Quelque soit l'usage

- expériences/manips
- simulations
- potager connecté
- logs
- *etc.*

Deux points compliqués :

- ré-analyse après un long moment
- reproductibilité/conditions d'obtention

Réponses ?

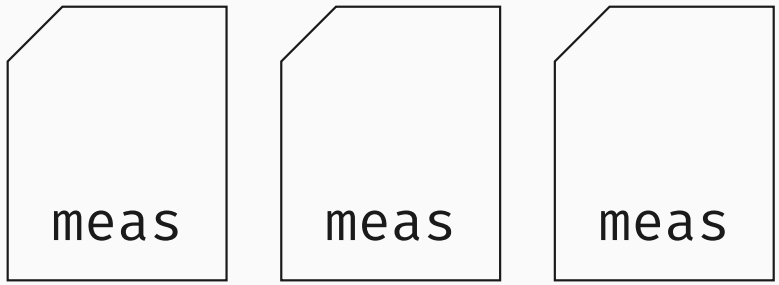
Conservation longue

Nécessité de compresser pour gagner de la place

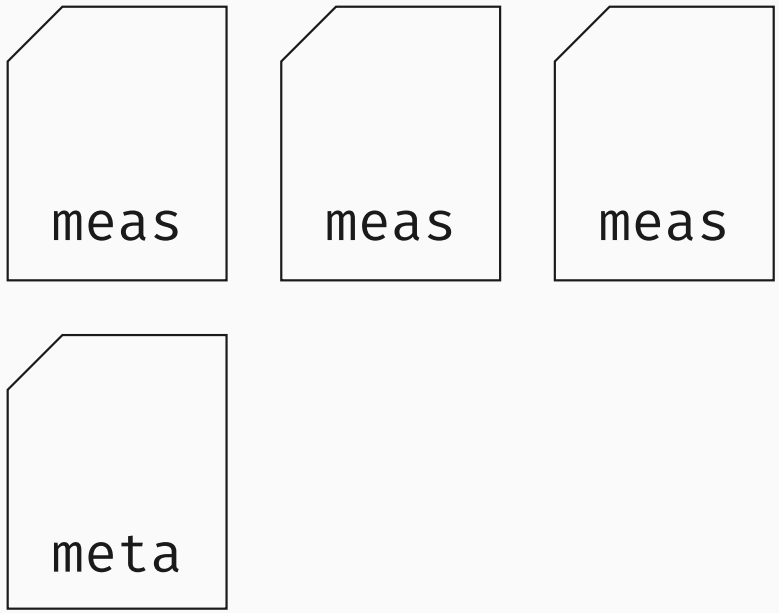
Ré-analyse & Reproductibilité

Besoin de conserver des métas données

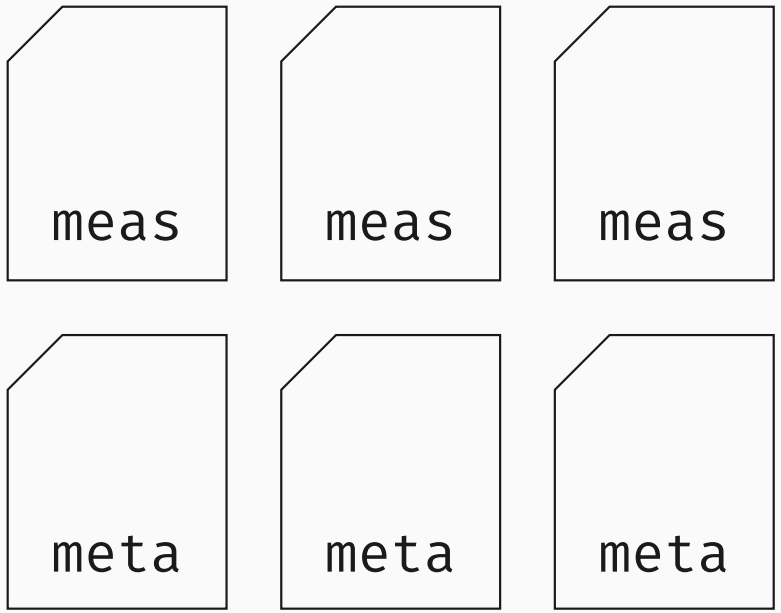
L’idée bidouille



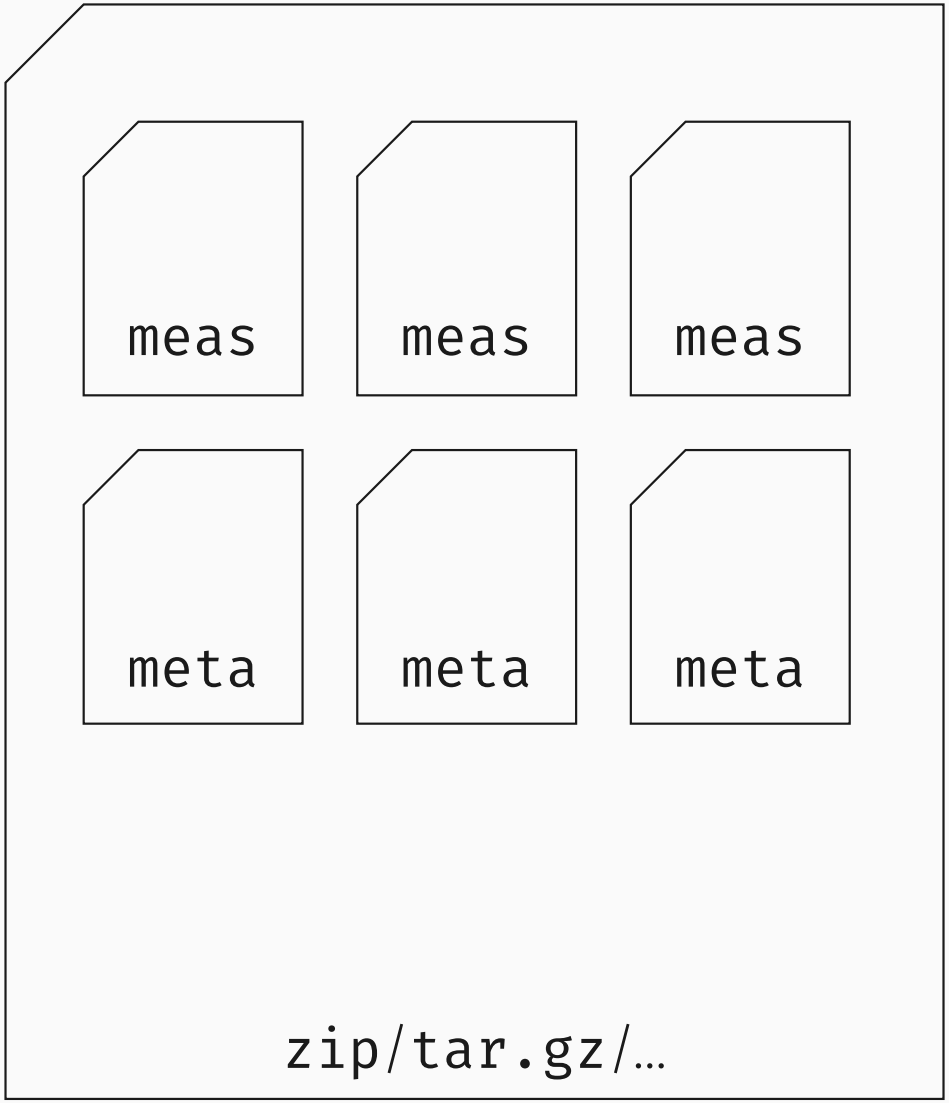
L’idée bidouille



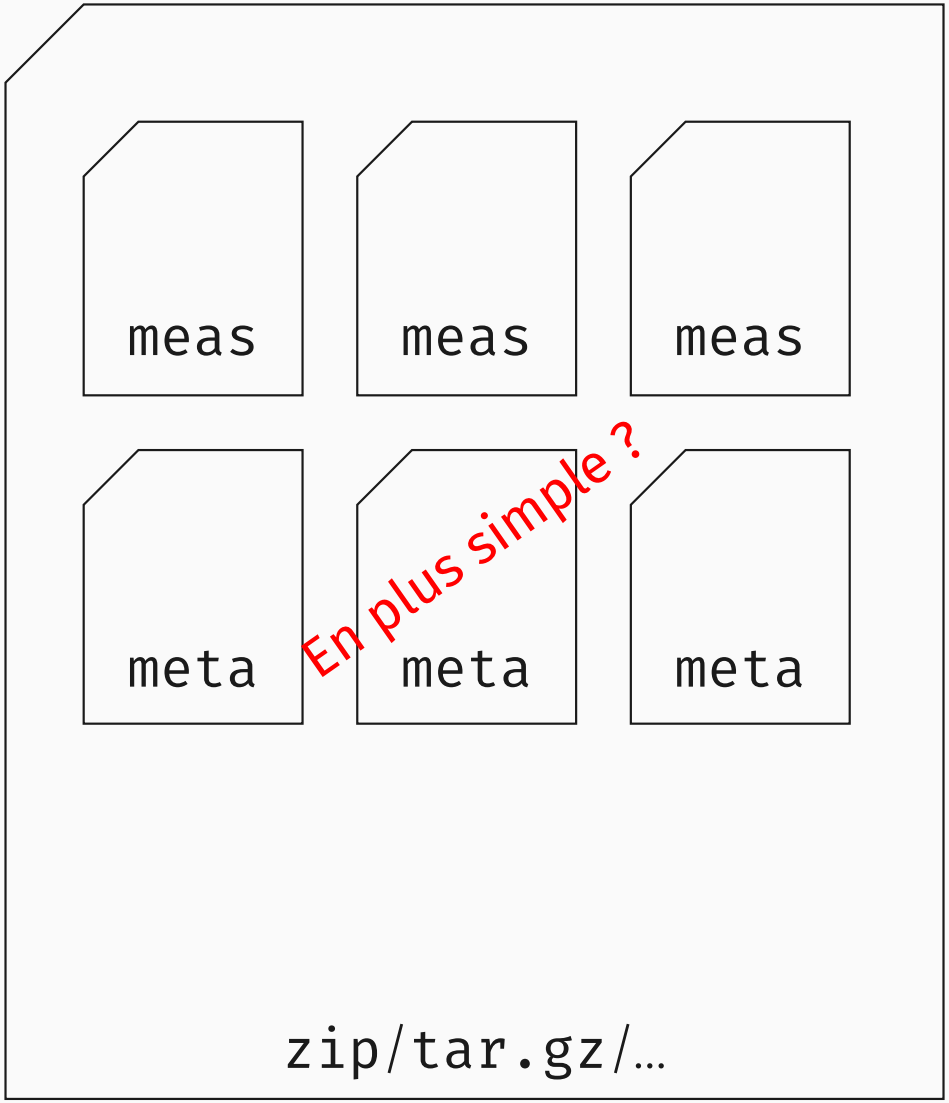
L’idée bidouille



L’idée bidouille



L’idée bidouille

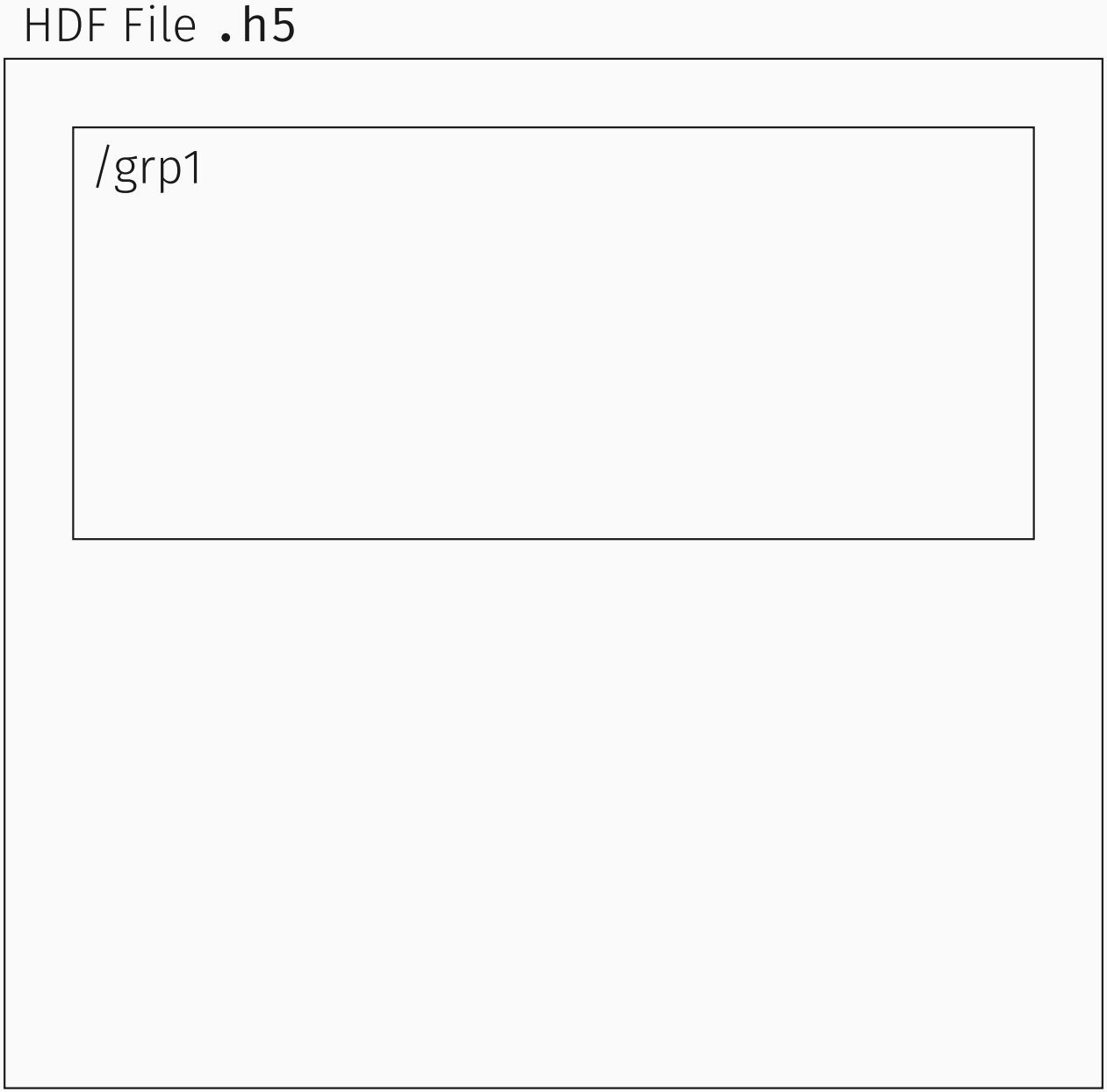


HDF5

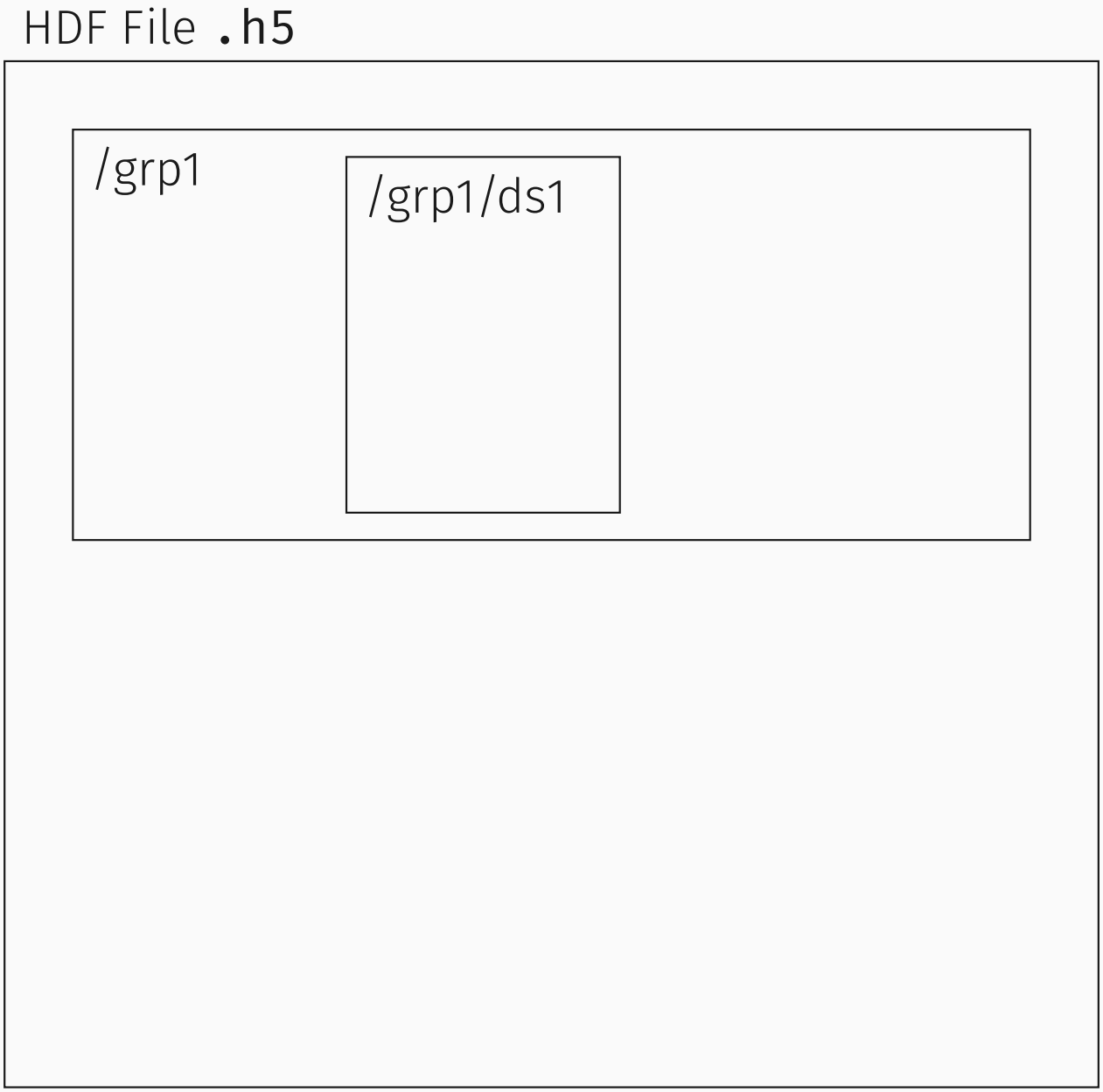
Hierarchical Data Format v5

- orienté dataset
- hiérarchie (notion de groupes, sous-groupes, *etc.*)
- métadonnées sur tous les éléments
- compressions

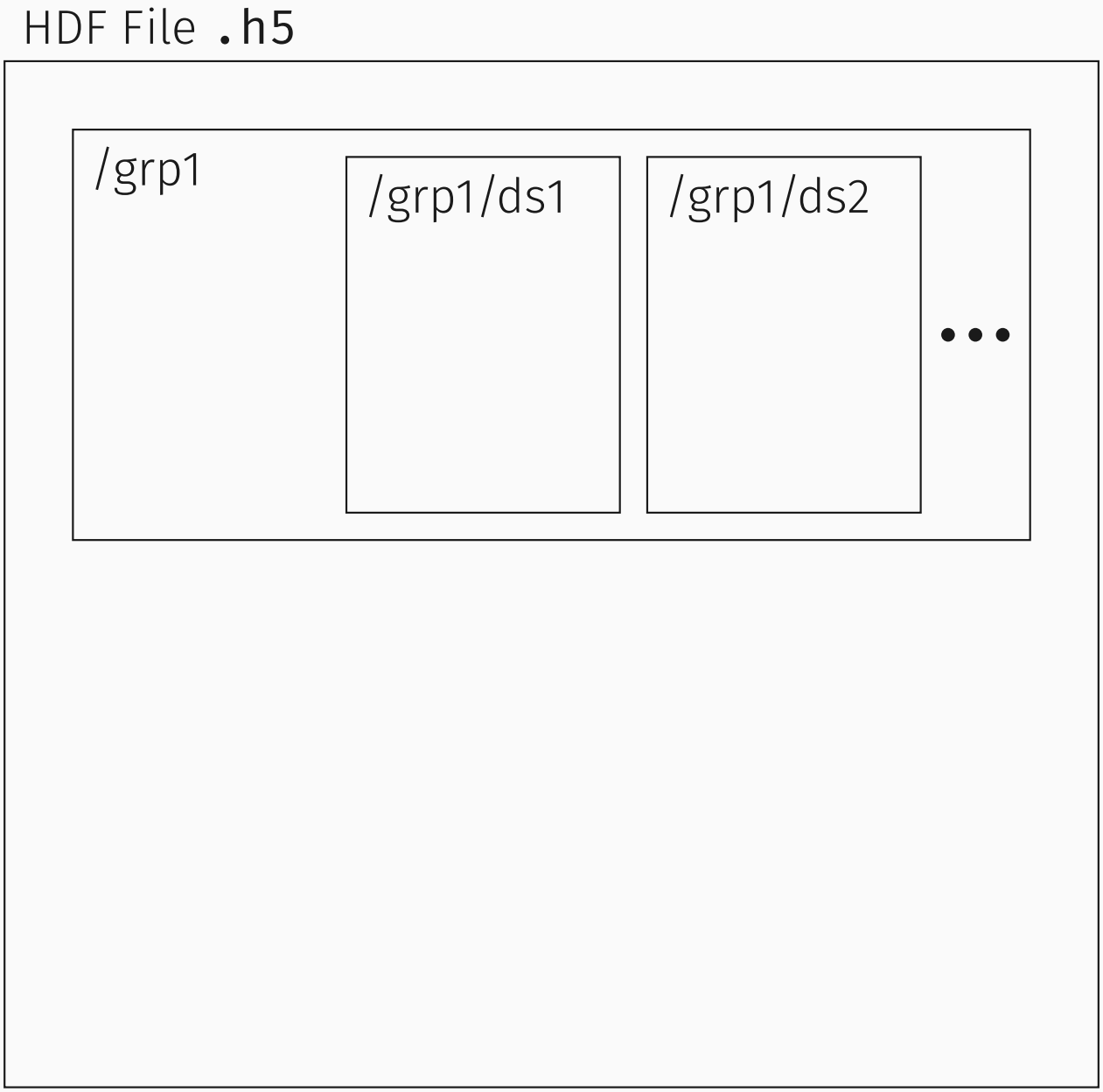
Concrètement ?



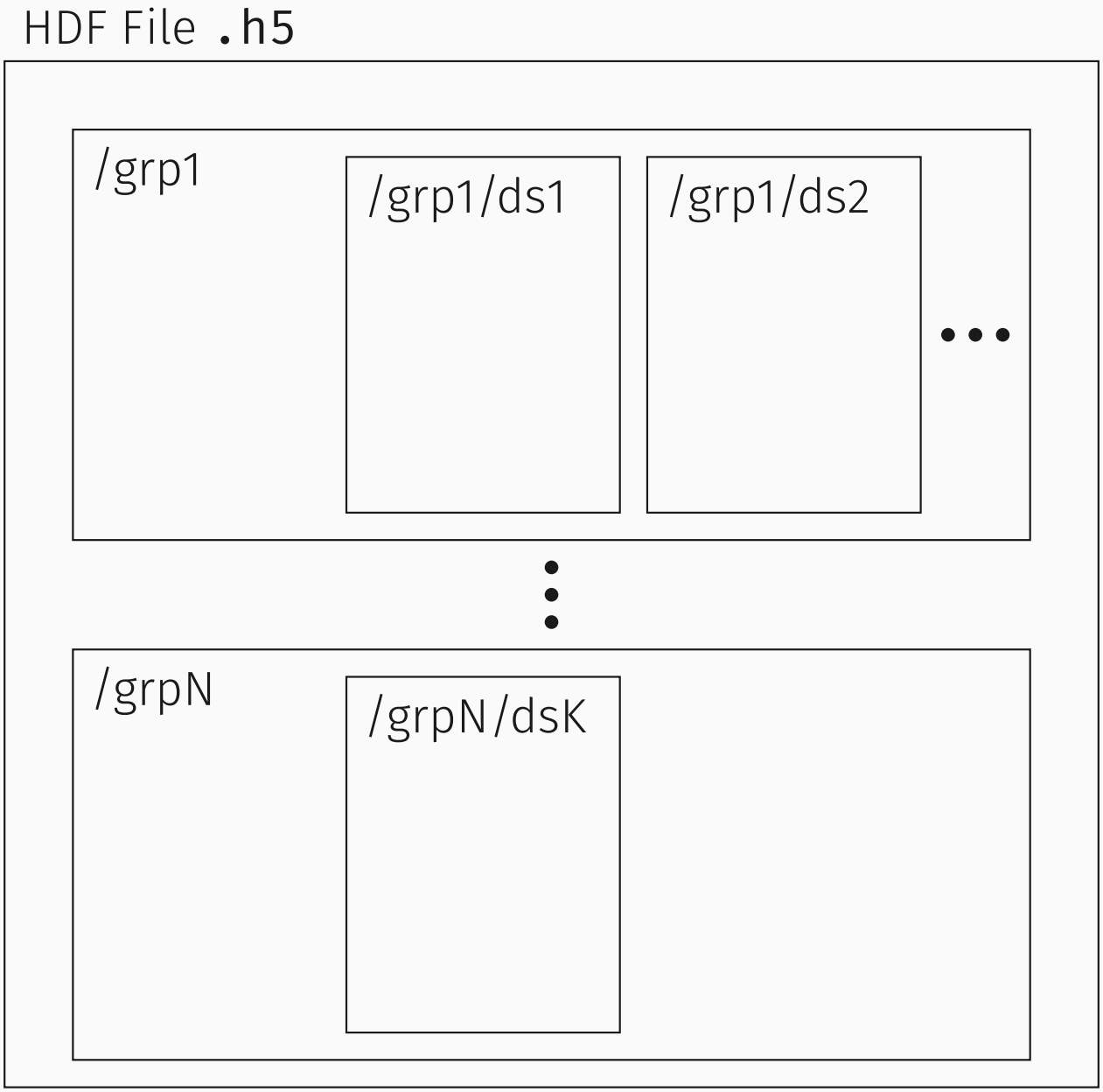
Concrètement ?



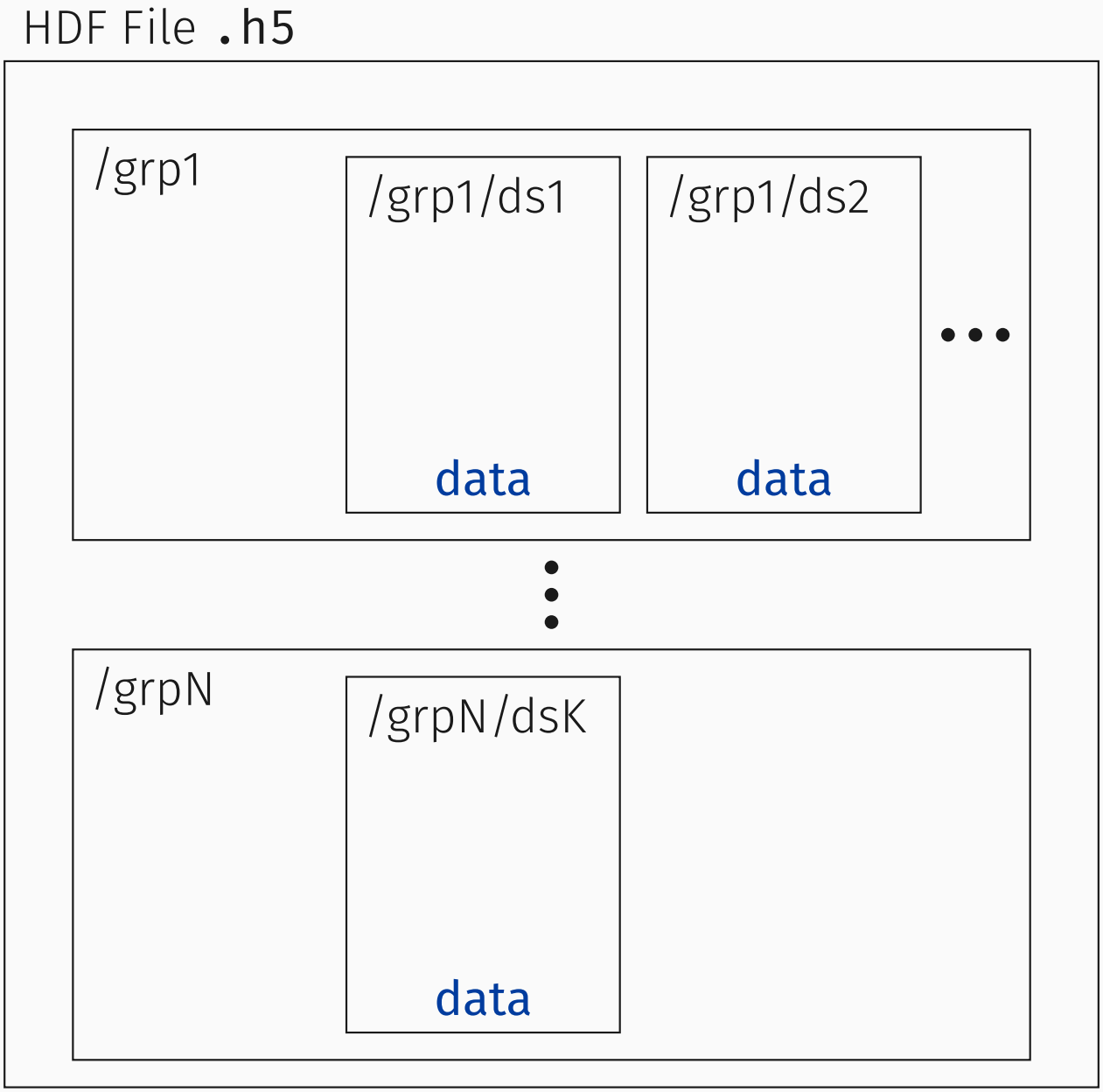
Concrètement ?



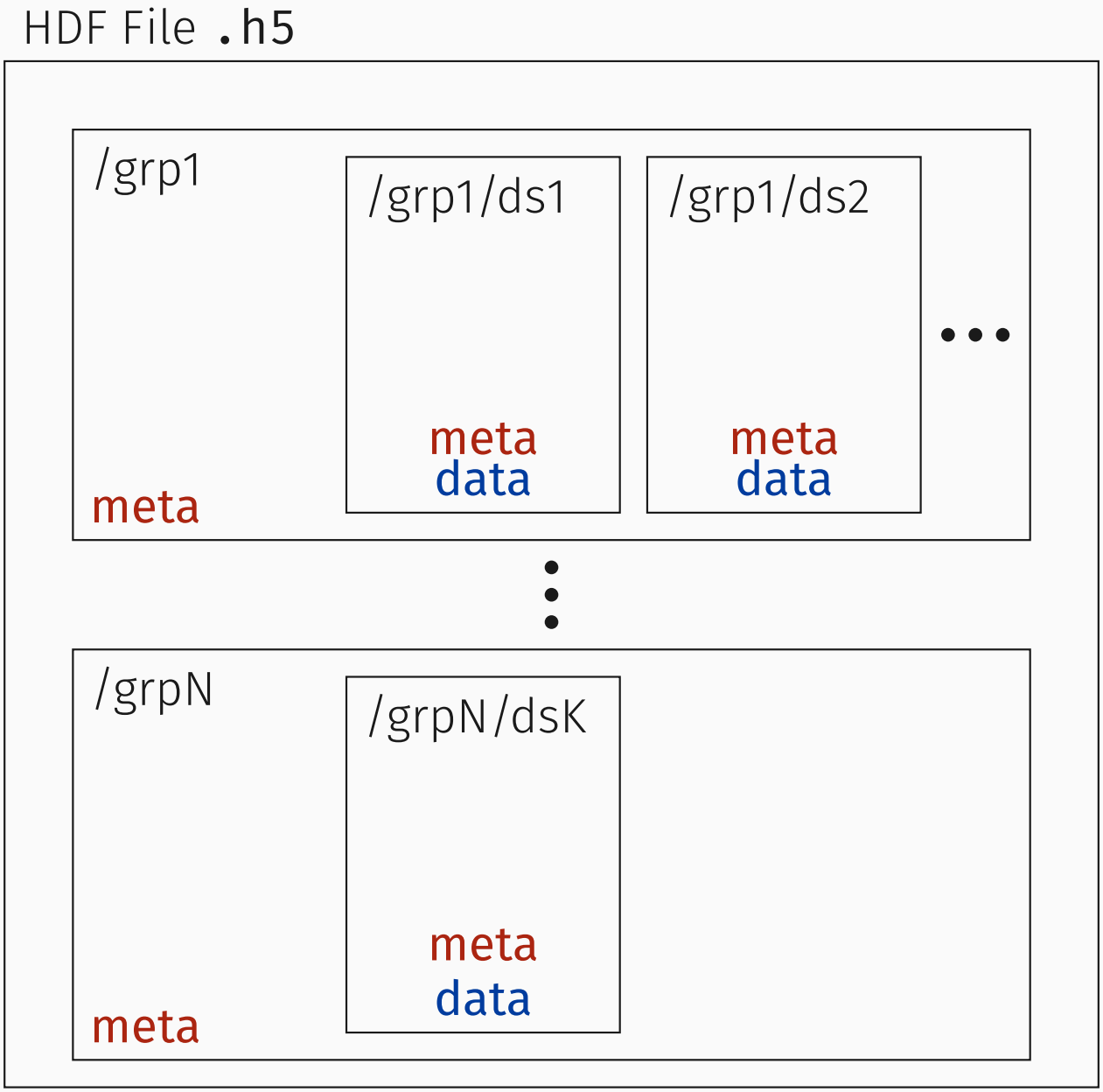
Concrètement ?



Concrètement ?



Concrètement ?



Possibilités

- Stockage de métadonnées au plus proche des données qu'elles décrivent
- Adressage depuis la racine ou en relatif (`/grp1/ds2` ou `ds2`)
- Datasets à *n* dimensions
- Compression

Interface Python

- Wrapper autour de la lib' C : h5py
- Interface plus haut niveau : PyTables

```
import h5py
import numpy as np

fh = h5py.File('fichier.h5', 'w')
grp1 = fh.create_group('grp1')
ds1 = grp1.create_dataset('ds1', data=np.ones(10,1))
grp1.attrs['date'] = "aujourd'hui"
ds1.attrs['type'] = 'uns'

fh.close()
```

Rien de bien méchant :)

Interface Python

```python
import h5py
import numpy as np

fh = h5py.File('fichier.h5', 'r')
^Ids1 = fh['grp1']['ds1']
ds1_bis = fh['/grp1/ds1']

fh.close()
```

Un monde parfait ?

Non...

- adapté aux données numériques
- mauvaise gestion des chaines (pas d'UTF8 par exemple)
- un seul gros fichier $\Rightarrow$ corruption!
- spécification horrible (150 pages) $\Rightarrow$ une seule implémentation

Du coup on fait quoi ?

Si vous avez des données :

- numériques
- complexes (besoin de métadata)
- grosses mais pas immenses

GO !

Sinon… tant pis pour vous !

Merci !

mathieu@haum.org