# Filtres et Contrôles

# Abréviations

M -> Mesure (target\_key)

(D, A) -> Douteux, Aberrant

# Type Traitement

## Aberrant

* Donnée supprimée :
  + Obs -> OK
  + Agrégation -> OK, puis cascade des niveaux sup
* Ajout dans LogFiltre

## Douteux

* Choix :
  + Garde la donnée, et on marque dans LogFiltre
    - On joue sur un indice de qualité (à définir)
  + On supprime, alors c’est pareil qu’une aberration pour nous…
* But : le moins possible d’interventions humaine…

# Contrôles au niveau observation

* M => Valeur mini, Valeur Max (D, A)
* Heure des max/min -> dans plage de l’obs (A)
* Min <= valeur <= Max (A)
* Trigger paramètres manquants
  + Ex pas de vitesse du vent => annule la direction et max/min
  + Pas de max/min -> pas d’heure de max/min (déjà fait)

# Contrôles au niveau agrégation

* Niveau d’agrégation H, D, M, Y, A
* Import d’agrégation à partir de ‘aggregations’ : Heure des max/min dans la plage de l’aggregations (A)
* Max/min sur 1H, 1D, 1M (D, A)
* Valeur max/min dans une tranche horaire (ex. UV,…)
* Min <= valeur moyenne <= Max (A)

# Contrôle cohérence par batch

* Contrôle temporel
  + (P(h-1) – p(h+1))/2 – p(h) doit être < seuil pour chaque paramètre
  + => on fait quoi ?
* Capteurs bloqués
  + Pas plus de N valeurs consécutives (Obs)
  + Vent : N = 12 (wind\_max), N=24(wind),…
  + -> on fait quoi ? (On annule les obs des valeurs identiques) + écriture dans LogFiltre
* Cohérences inter-paramètres
  + Pression mer < pression station
  + -> on fait quoi ?

# Indice de qualité

Comptage nb incidents douteux/aberrant par mesure

Json -> Obs/Aggreg

# Table de logFiltre

* + (Obs\_Id ou Aggregation\_Id)
  + Meteor
  + Measure key (target\_key)
  + Date (stop-date pour obs, start\_date pour agrégation)
  + Identifiant du filtre
  + Type filtre (D, A)
  + Json pour (Mesure : Valeur)
  + Explication

# Stockage des infos de filtres :

* Table Type\_Instrument
  + Ajout colonne Jsonb :

[{type : all / groupName :   
 { key : ‘xx’,  
 obs : {

max\_D : ‘’,   
 min\_D : ‘’,   
 max\_A :’’,   
 min\_A : ‘’,   
 },  
 agg : [{  
 niveau : ‘H/D/M/Y’,  
 t\_min : 0, // default mini de la période. Unité dépend du niveau  
 // minutes pour niveau H  
 // heures pour niveau D  
 // jour pour niveau M  
 // mois pour niveau Y  
 t\_max : 59/24/31/12, // défaut max de la période. Unité comme pour le t\_min  
 max\_D : ‘’,   
 min\_D : ‘’,   
 max\_A :’’,   
 min\_A : ‘’,  
 }, ….  
 ],

nul : [‘key1’, ‘key2’,…], // cascade de nul   
} ]

* + - On verra pour les contrôles de cohérences faits par le batch…