

# Electronique du trajectographe à fibres scintillantes (SciFi Tracker) pour l'expérience LHCb et les implications techniques du LPNHE

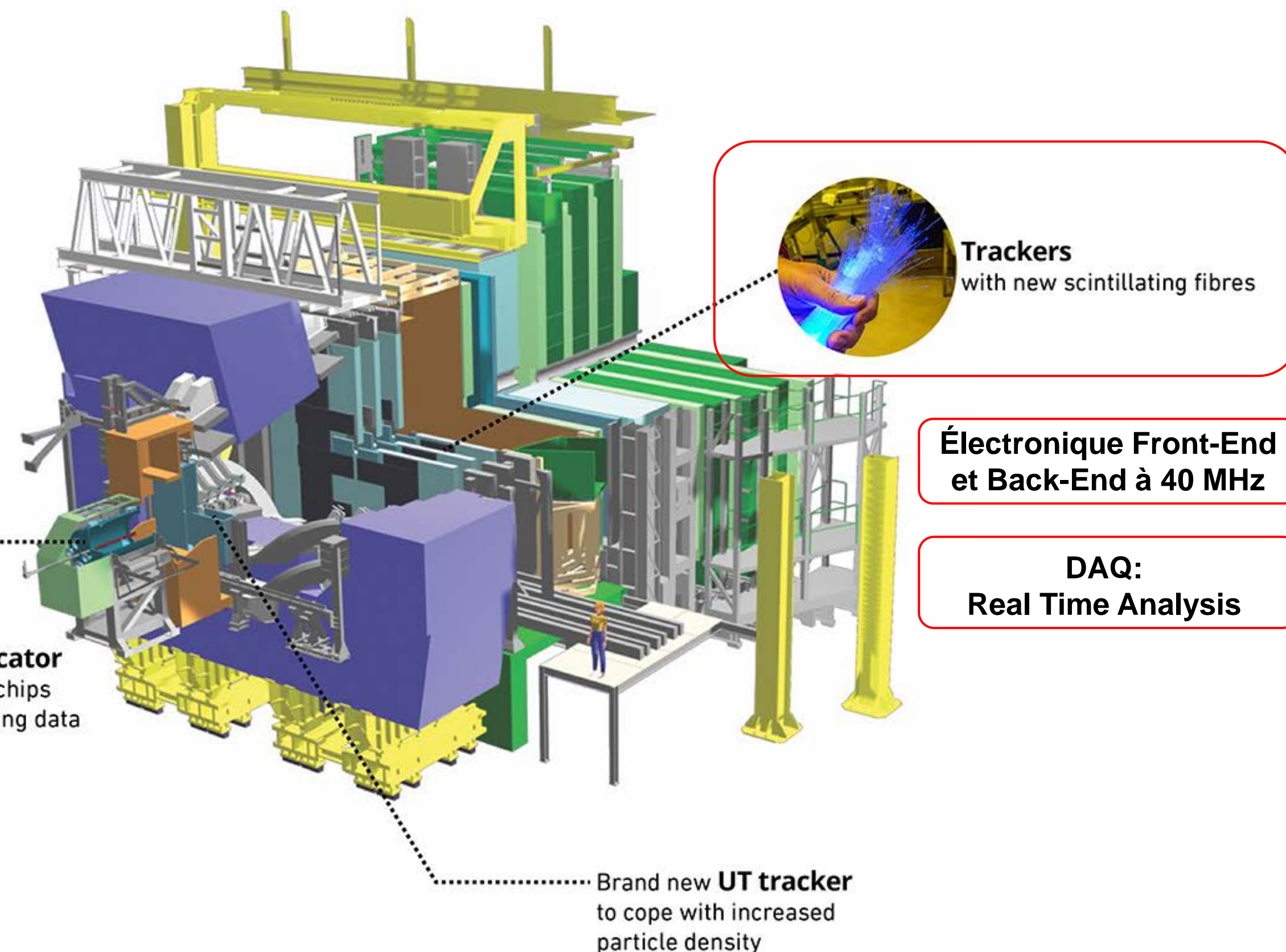


L. Del Buono, O. Le Dortz, J-L. Meunier, F. Polci

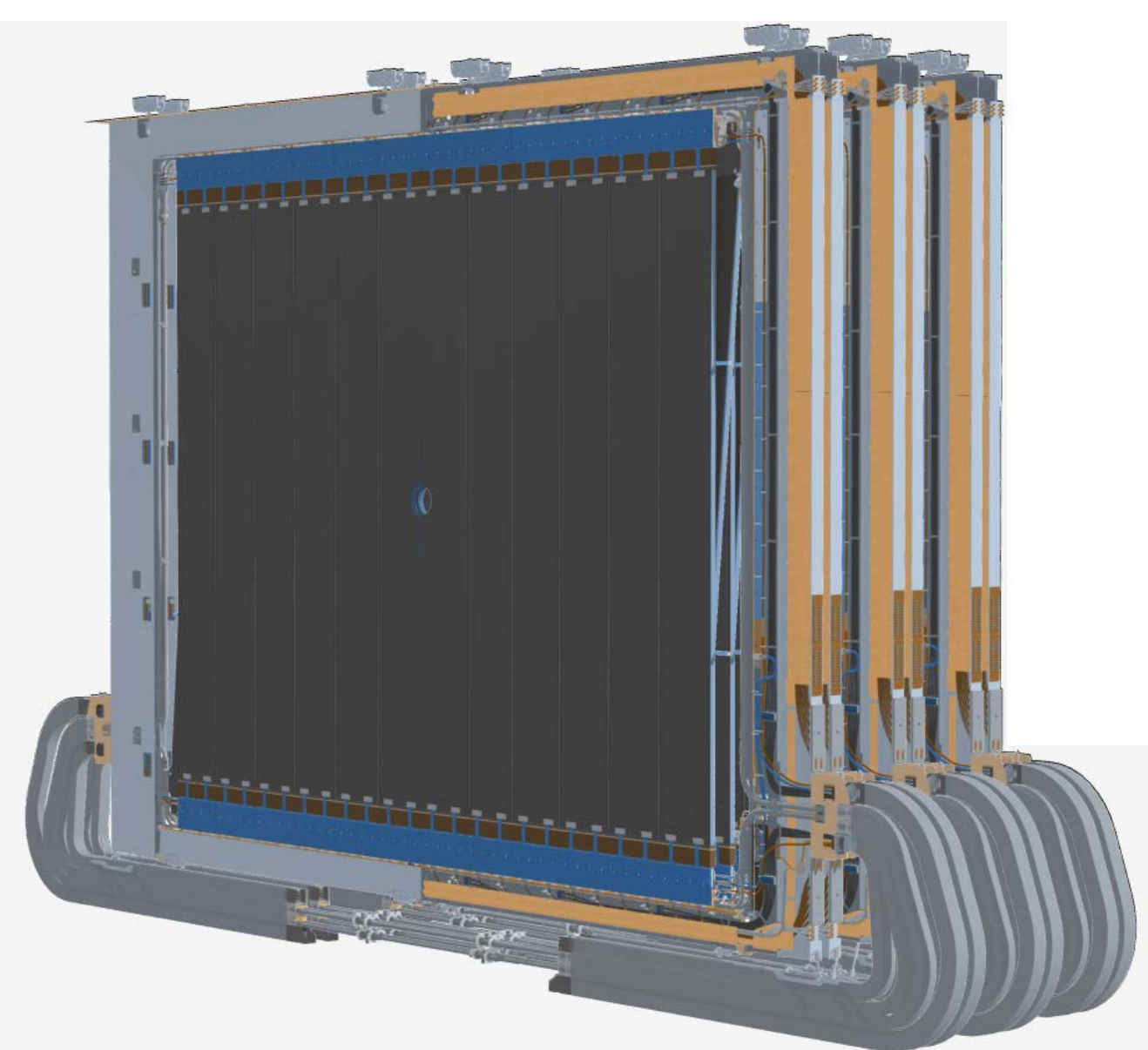
Biennale du LPNHE 2022, St-Brevin-les-Pins



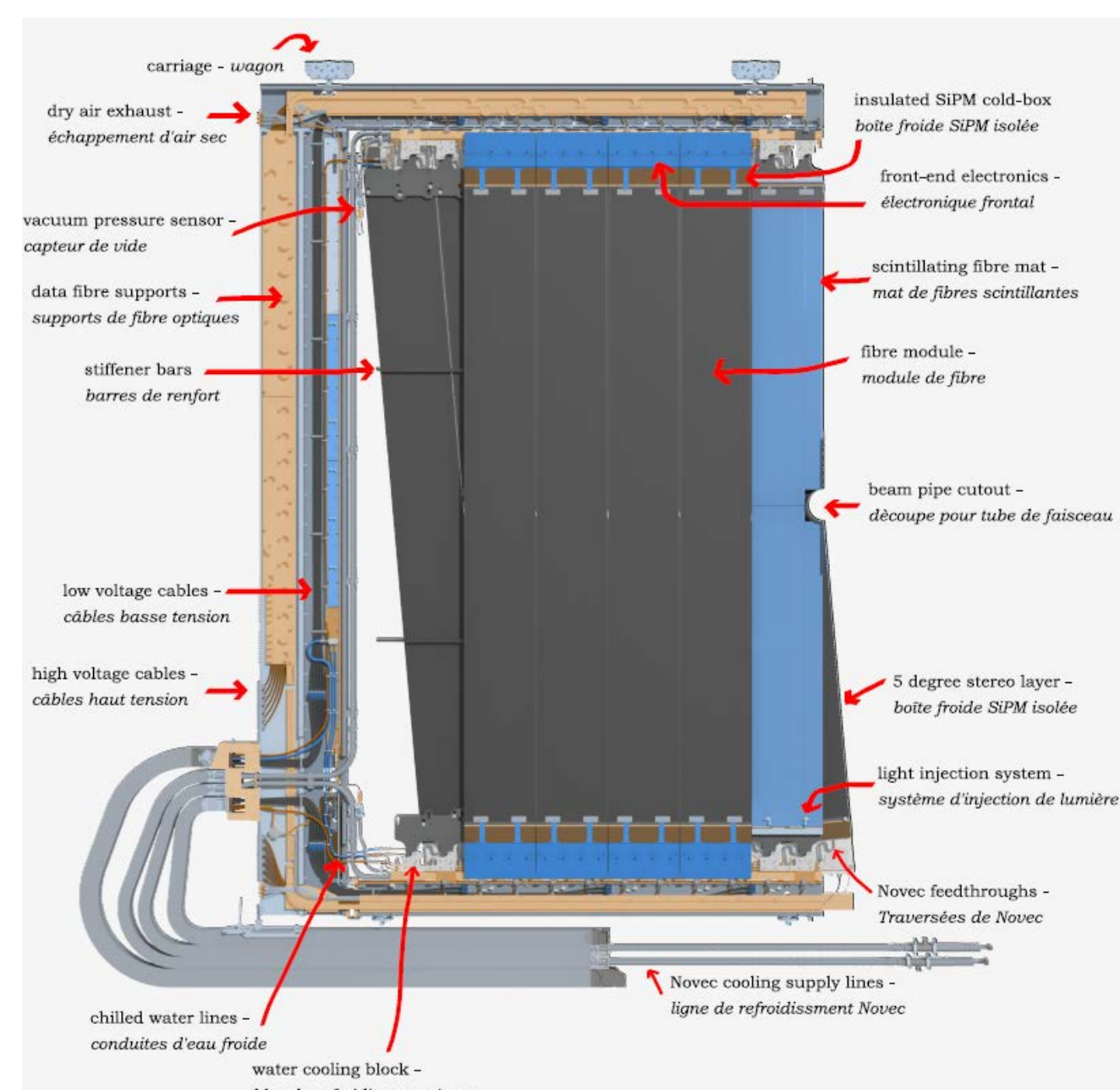
## L'Upgrade I de LHCb



L'expérience LHCb au CERN a fait l'objet, à l'occasion de l'upgrade phase I (2022-2030), de nombreuses améliorations. La contribution technique du LPNHE s'est orientée vers l'électronique back-end du « tracker SciFi » (trajectographe à fibres scintillantes).



(a) h=2x2.5m, l=2x3m, 6 plans en z

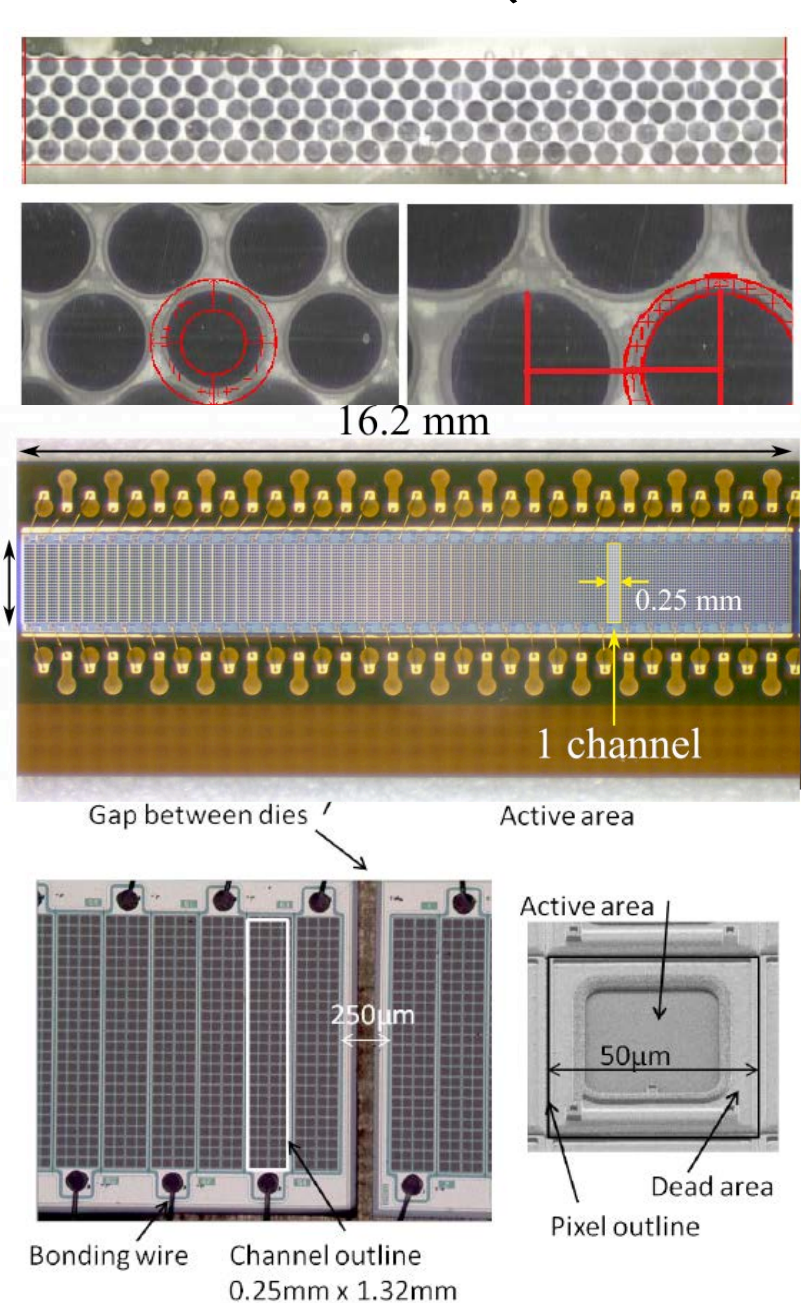


(b) C-Frame = 2 quadrants sur 2 plans

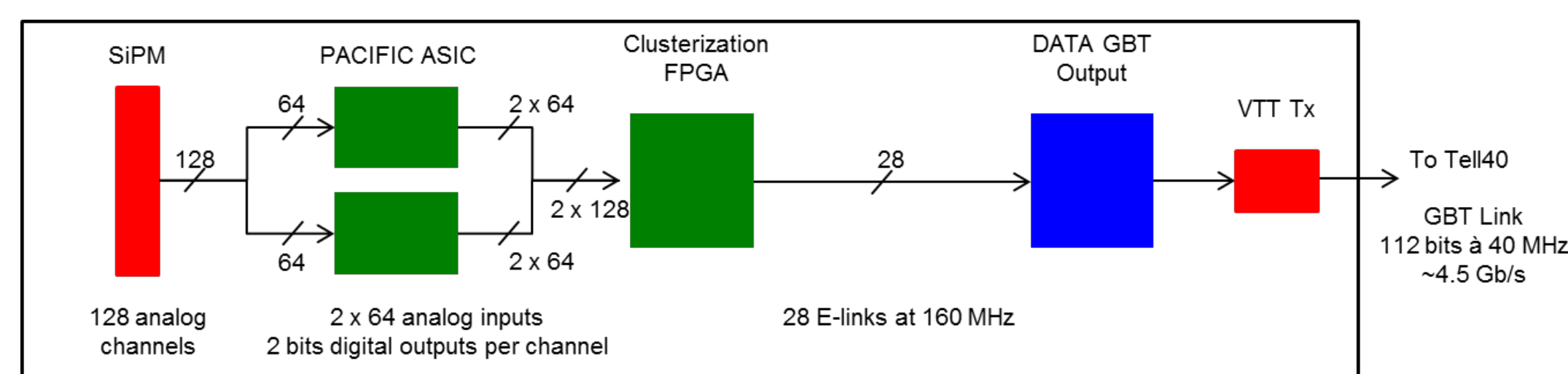
Le tracker SciFi (a) est constitué de 12 C-Frames (b). Les fibres scintillantes (en plastique, 2.5m de long, 0.25 mm de diamètre) sont disposées verticalement (ou légèrement inclinées). Les impulsions lumineuses engendrées par l'interaction des particules avec les fibres sont recueillies par l'électronique frontale, en haut et en bas.

## Electronique Frontale

Détail d'un SiPM (128 canaux)



Chaîne de lecture FE d'un bloc de SiPM



Dans les « front-end boards », proches du détecteur, lecture des signaux issus de photomultiplicateurs à silicium (SiPM).

- Mise en forme et seuillage (dans ASICs)
- Suppression de zéro, clustering (auquel nous avons contribué), dans le FPGA front-end
- Envoi des données par lien optique « GBT »

### Chiffres clés:

- 1 SiPM = 128 canaux = 1 lien optique vers back-end (4.5 Gb/s)
- 2x8 liens optiques de données par boîte front-end (2 cartes FE)
- Au total
  - 4096 SiPM = 524k canaux individuels de SiPM
  - Débit total de sortie du tracker SciFi = 18 Tbits/s

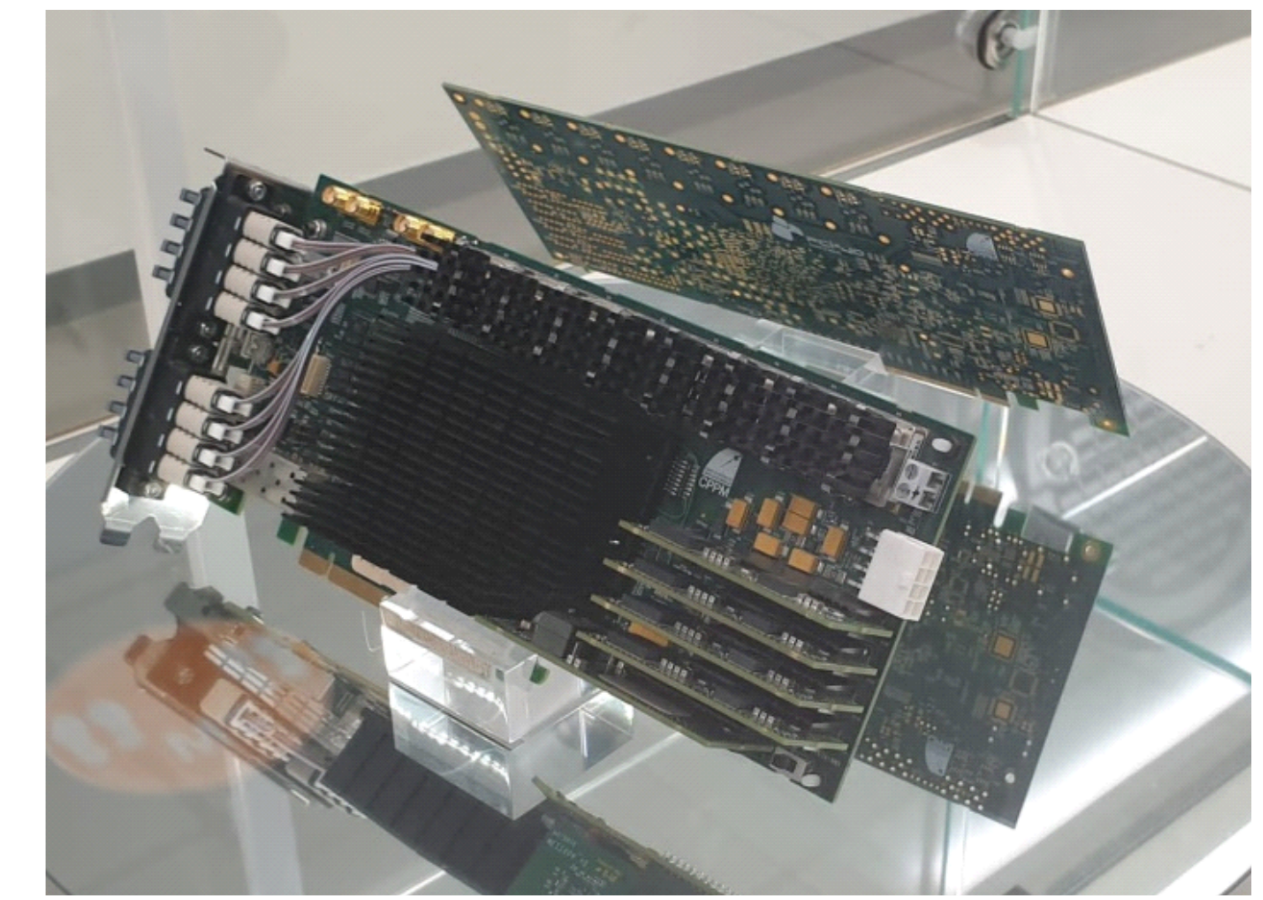
## Electronique Back-End

Concentration de données issues de plusieurs liens FE

Basée sur des cartes **PCIe40** (CPPM)

24 ou 48 liens multiGbits optiques

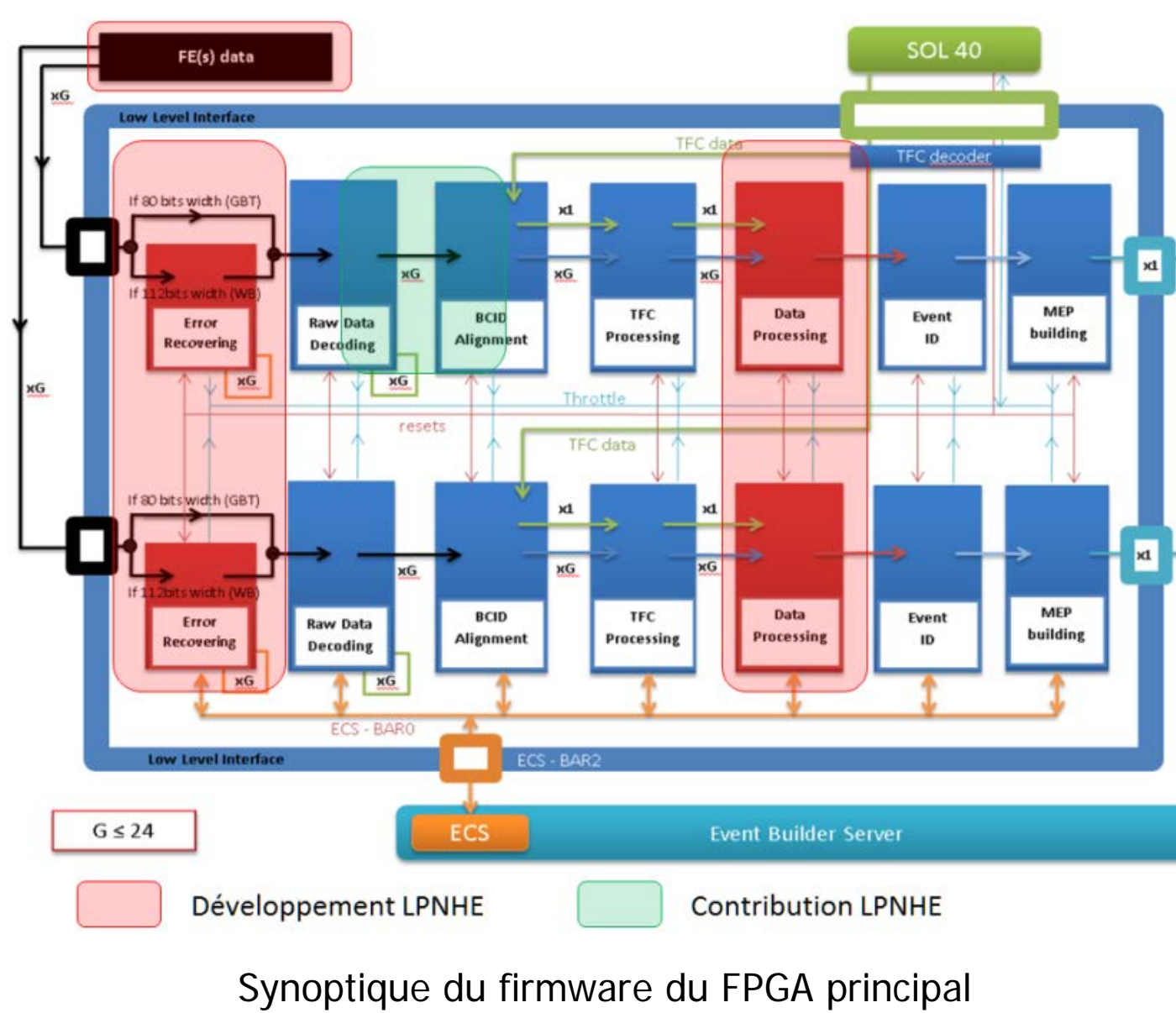
Sortie des données vers DAQ via une interface PCIexpress (100 Gb/s max, 80 Gb/s en pratique)



Carte PCIe40 (CPPM-IN2P3)

Le firmware du FPGA principal (intelFPGA Arria10) de la carte est développé en collaboration autour d'un framework commun

Le LPNHE a développé un firmware de contrôle spécifique des cartes backend du SciFi dont la fonction première est la lecture des données du front-end en synchronisation avec les signaux de contrôle rapides du système d'acquisition de LHCb et de l'accélérateur. Malgré des



contraintes sévères en terme de conception (ressources du FPGA allouées aux blocs spécifiques, horloges de 200 à 250 MHz), la conception modulaire et l'optimisation du code en VHDL ont permis d'offrir des fonctions supplémentaires de calibration et de monitoring au plus près du front-end sans compromettre les

performances du système, et en accélérant de manière significative ces opérations. Des outils d'émulation ont également été développés et ont rendu possible la mise au point du système sur des bancs de test qui ne comportaient pas tous les éléments de la chaîne d'acquisition, par exemple un utile émulateur de front-end.

### Chiffres clés:

- 144 cartes PCIe40-TELL40 pour le SciFi (4096 liens d'entrées, 1x16 ou 2x40 entrées par carte)
- Une carte:
  - 180 Gb/s en entrée, 50 Gb/s utile en sortie
  - 10.000 histogrammes effectués au vol en 300 ms

### Logiciel de contrôle de l'expérience (ECS)

La configuration, le contrôle et la surveillance des centaines de cartes back-end imposent l'utilisation d'interfaces dédiées, permettant de gérer massivement et ergonomiquement le sous-système back-end du SciFi, dans l'environnement LHCb. A cet effet, un certain nombre de panneaux et de scripts de contrôle, plutôt destinés à l'utilisation par des experts, ont été créés dans l'architecture software choisie par le CERN/LHCb (WINCC-OA + JCOP). Ces panneaux permettent par exemple d'instancier la structure software des cartes, de les connecter à la machine à états du système d'acquisition de LHCb, de faire des courbes de calibration, etc

### Expertises mises en œuvre au LPNHE:

VHDL, simulations firmware avancées, optimisation timing/ressources

Conception de firmware en équipe (git, intégration continue...)

Python et framework PyQt5 (bancs de tests / interfaces graphiques, analyse et validation de simulation firmware)

SCADA (WinCCOA): développement scripts et panneaux graphiques