

---

# Lambda+ Architecture et vérification de la conservation des propriétés avec la théorie des catégories

Annabelle Gillet, Éric Leclercq, Nadine Cullot

LIB Univ. Bourgogne Franche Comté, Dijon, France  
{prenom}.{nom}@u-bourgogne.fr

---

*RÉSUMÉ.* La Lambda Architecture a été beaucoup critiquée, principalement à cause de sa complexité mais aussi car les propriétés de temps réel et d'exactitude des traitements sont effectives chacune dans une couche distincte mais pas dans l'architecture entière. Toutefois, la Lambda présente des mécanismes intéressants. Nous proposons donc une évolution de la Lambda Architecture : la Lambda+ Architecture, qui supporte à la fois des analyses exploratoires et en temps réel sur les données. Nous proposons également d'étudier la conservation des propriétés dans les compositions de composants d'une architecture avec la théorie des catégories.

*ABSTRACT.* The Lambda Architecture has been highly criticized, mostly because of its complexity and because the real-time and correctness properties are each effective in a different layer but not in the overall architecture. Nevertheless, it proposes some interesting mechanisms. We present a renewal of the Lambda Architecture: the Lambda+ Architecture, supporting both exploratory and real-time analyzes on data. We propose to study the conservation of properties in composition of components in an architecture using the category theory.

*MOTS-CLÉS :* Patron d'architecture, Théorie des catégories, Lambda Architecture

*KEYWORDS:* Architecture pattern, Category theory, Lambda Architecture

---

Les propriétés d'exactitude des traitements, de temps réel et de tolérance aux pannes ont toujours été un enjeu majeur lors de la conception d'architectures. (Lampson, 1983) propose quelques suggestions qui sont toujours d'actualité et qui peuvent être retrouvées, entre autres, dans la Lambda Architecture (Marz, 2011). La couche *speed* de la Lambda permet d'obtenir la propriété de temps réel, tandis que la couche *batch* supporte la propriété d'exactitude des traitements. Mais lorsque les données sont rassemblées grâce à la couche *servng*, ces deux propriétés ne peuvent être supportées simultanément. Nous proposons d'étudier la conservation des propriétés dans les compositions de composants d'une architecture grâce à la théorie des catégories, mais aussi de faire évoluer la Lambda Architecture en la Lambda+ Architecture afin d'améliorer son support des propriétés et d'étendre ses cas d'utilisation pour pouvoir tirer profit des données massives. Cet article présente un résumé de l'article (Gillet *et al.*, 2021) publié dans la conférence CAiSE 2021.

La Lambda+ Architecture a deux fonctionnalités principales : 1) stocker les données d'une façon qui permet de réaliser des analyses exploratoires ; et 2) calculer en temps réel des indicateurs macroscopiques prédéfinis afin de suivre l'évolution de données d'intérêt. Pour cela, la Lambda+ est constituée de cinq composants interagissant de manière asynchrone à l'aide de messages. La dualité entre les analyses exploratoires et les indicateurs macroscopiques a une grande importance dans un contexte de données massives, dans lequel la combinaison du volume et de la variété des données empêche de révéler l'intégralité de leur valeur.

Dans le domaine de la recherche de la conception d'architectures logicielles, le développement d'une théorie adaptée accompagnée d'une formalisation est essentielle (Broy, 2011 ; Johnson *et al.*, 2012). Il est nécessaire de pouvoir prouver le maintien des propriétés dans l'ensemble d'une architecture, que ce soit au moment de sa conception ou de son évolution. La théorie des catégories (Eilenberg, MacLane, 1945) est une approche prometteuse pour répondre à ces besoins. En se concentrant sur les relations (les morphismes) et les compositions, il est possible de combiner de puissants mécanismes qui peuvent être ensuite appliqués aux architectures. De cette manière, le comportement des foncteurs couplés aux préordres permet d'étudier la conservation ou la perte des propriétés dans les compositions de composants. La formalisation que nous avons développée à partir de ces éléments a été appliquée à la Lambda Architecture ainsi qu'à la Lambda+ Architecture afin de montrer les défauts de la Lambda et la manière dont la Lambda+ les compense.

#### Remerciements

*Ce travail est soutenu par ISITE-BFC (ANR-15-IDEX-0003), piloté par Gilles Brachotte, laboratoire CIMEOS EA-4177, Université de Bourgogne.*

#### Bibliographie

- Broy M. (2011). Can practitioners neglect theory and theoreticians neglect practice? *Computer*, vol. 44, n° 10, p. 19–24.
- Eilenberg S., MacLane S. (1945). General theory of natural equivalences. *Transactions of the American Mathematical Society*, vol. 58, n° 2, p. 231–294.
- Gillet A., Leclercq É., Cullot N. (2021). Lambda+, the renewal of the lambda architecture: Category theory to the rescue. In *International Conference on Advanced Information Systems Engineering*, p. 381–396.
- Johnson P., Ekstedt M., Jacobson I. (2012). Where's the theory for software engineering? *IEEE software*, vol. 29, n° 5, p. 96–96.
- Lampson B. W. (1983). Hints for computer system design. In *Proceedings of the ninth acm symposium on operating systems principles*, p. 33–48.
- Marz N. (2011). *How to beat the cap theorem*. Consulté sur <http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html>