

cellenza

OFFRE

APP MODERNIZATION



CONTEXTE / POURQUOI CETTE OFFRE

La très grande majorité des entreprises possèdent des applications construites il y a des années. Certains les appellent le « Legacy », d'autres continuent à le maintenir et le faire évoluer constamment.

Avant l'époque SOA, on avait l'habitude de tout regrouper dans un seul gros projet pour profiter de la facilité à impacter tout le projet rapidement, ou avoir de la transactionnalité.

Aujourd'hui, le marché exige de déployer de la valeur rapidement afin de saisir les opportunités, et d'itérer grâce aux feedbacks qu'on reçoit plus rapidement. Il faut donc **livrer les applications plus fréquemment et rapidement.**

Ces exigences demandent une réappropriation et restructuration des applications « Legacy ».

Cette offre s'applique lorsque :

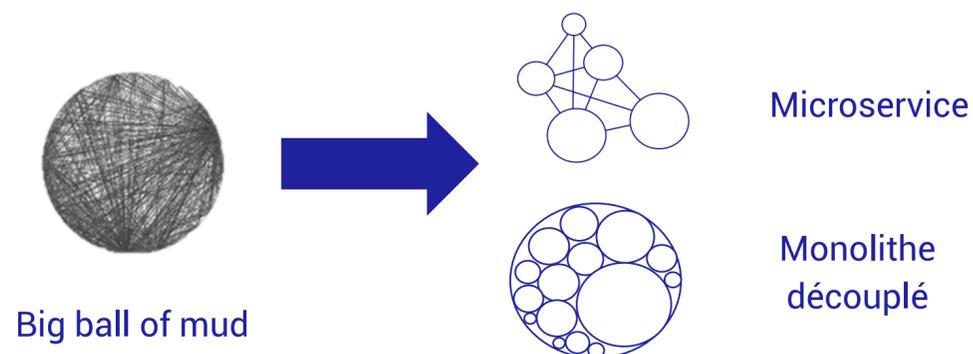
- L'application est complexe à déployer ;
- Le code est très difficile à maintenir et à faire évoluer ;
- Le Time-to-Market est de plus en plus long ;
- L'application est fragile en production.

NOTRE CONVICTION ISSUE DE NOTRE RETOUR D'EXPÉRIENCE

La cible est claire : les **micro-services dans Microsoft Azure**

Ces applications déployables individuellement, qui communiquent entre elles, sont les plus autonomes possible, et répondent chacune à un besoin spécifique (métier ou technique).

Il n'est toutefois pas toujours possible de partir d'une page blanche et de commencer une « version 2 ». L'offre App Modernization présente une façon de procéder afin de transformer un « big ball of mud » soit en monolithe découplé, soit en micro-services.



Notre conviction est qu'il est important :

- D'avoir une bonne compréhension du métier ;
- D'avoir une cible architecturale claire, tout en s'accordant de la faire évoluer au fil de l'eau ;
- De refactorer l'application petit à petit, en évitant l'effet tunnel et le « big bang » ;
- De se donner les moyens d'y parvenir avec une ou des équipes dédiées.
- D'être familiarisé avec les concepts et services du Cloud Azure

DÉMARCHE : PRÉPARATIFS

L'objectif de cette offre est de pouvoir refactorer, puis d'extraire ou d'isoler des parties de l'application originale. Au préalable, il faut toutefois effectuer ces tâches préliminaires et primordiales :

Analyse de l'existant

Objectif : comprendre l'application, son architecture actuelle, les contraintes imposées, réaliser un découpage en domaines fonctionnels. Audit de code, des outils et de l'infrastructure. Ateliers (Event Storming).

Définition de la cible

Objectif : disposer d'une structure globale qui servira de ligne directrice. Ateliers (Extreme architecture).

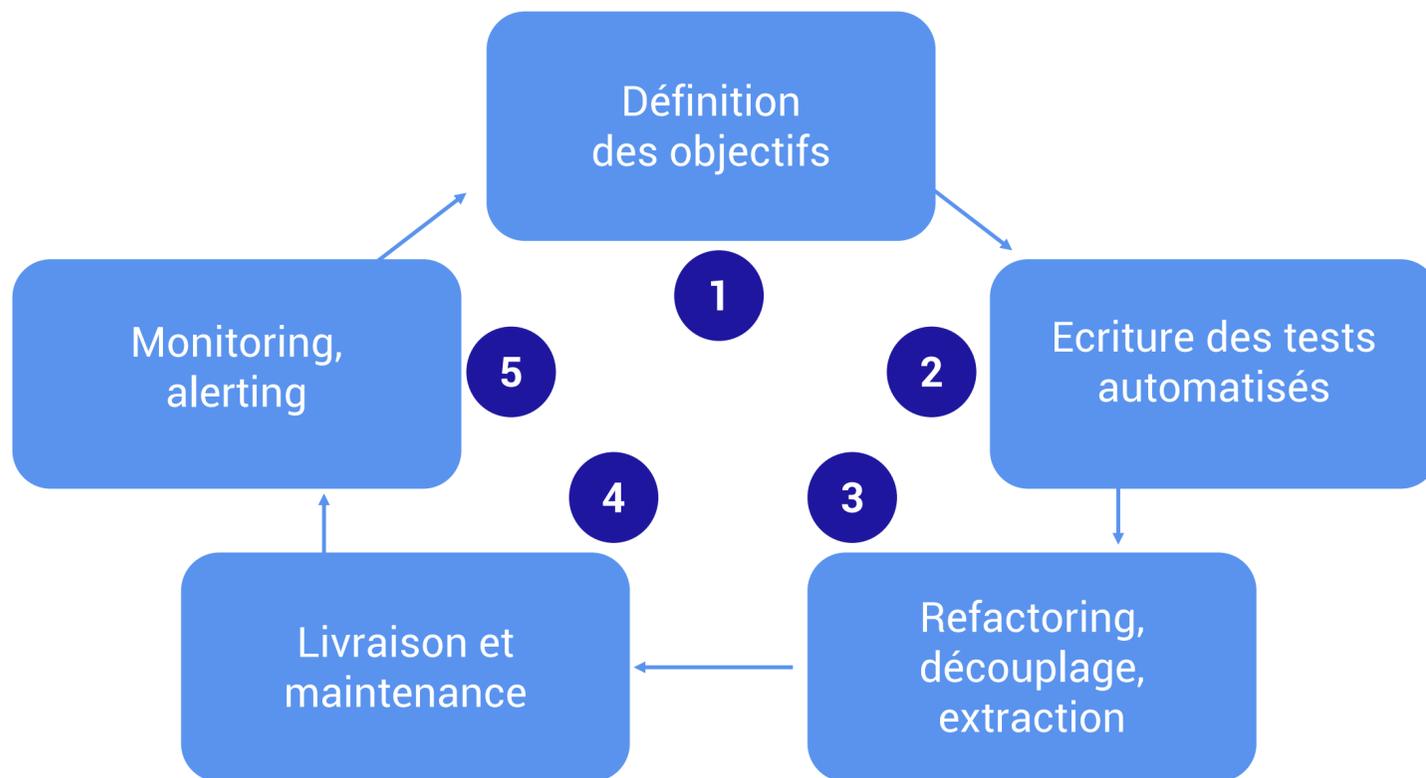
Mise en place de pratiques « DevOps »

Objectif : être capable de déployer l'application existante fréquemment, rapidement et sereinement, afin que celle-ci ne ralentisse pas la cadence de déploiement des nouvelles briques. De plus, elle permettra de monitorer la santé de l'application existante tout au long de son évolution.

Il sera également possible d'envisager de re-hoster l'application sur une autre infrastructure sur le Cloud Azure.

DÉMARCHE : PHASE ITÉRATIVE

L'effort principal consiste donc à refactorer puis à extraire ou isoler des parties l'application.



L'étape n° 2 est celle qui garantit l'absence de régression. En plus de rendre le code plus maintenable, le refactoring permet l'écriture de tests unitaires.

La phase n° 3 est une bonne occasion pour moderniser le code, lorsque c'est nécessaire ou de manière systématique.

Lors de la **phase de livraison**, un déplacement de données est nécessaire. Ce déplacement doit être inclus et testé durant la phase de développement. S'il est estimé nécessaire de s'accorder un filet de sécurité, la mise en place de Feature Flipping et de Reverse Dataflow peut être envisagée.

Cette phase peut s'arrêter à tout moment, lorsque l'on estime que le résultat est satisfaisant.

DÉMARCHE : ÉTAPES SUPPLÉMENTAIRES

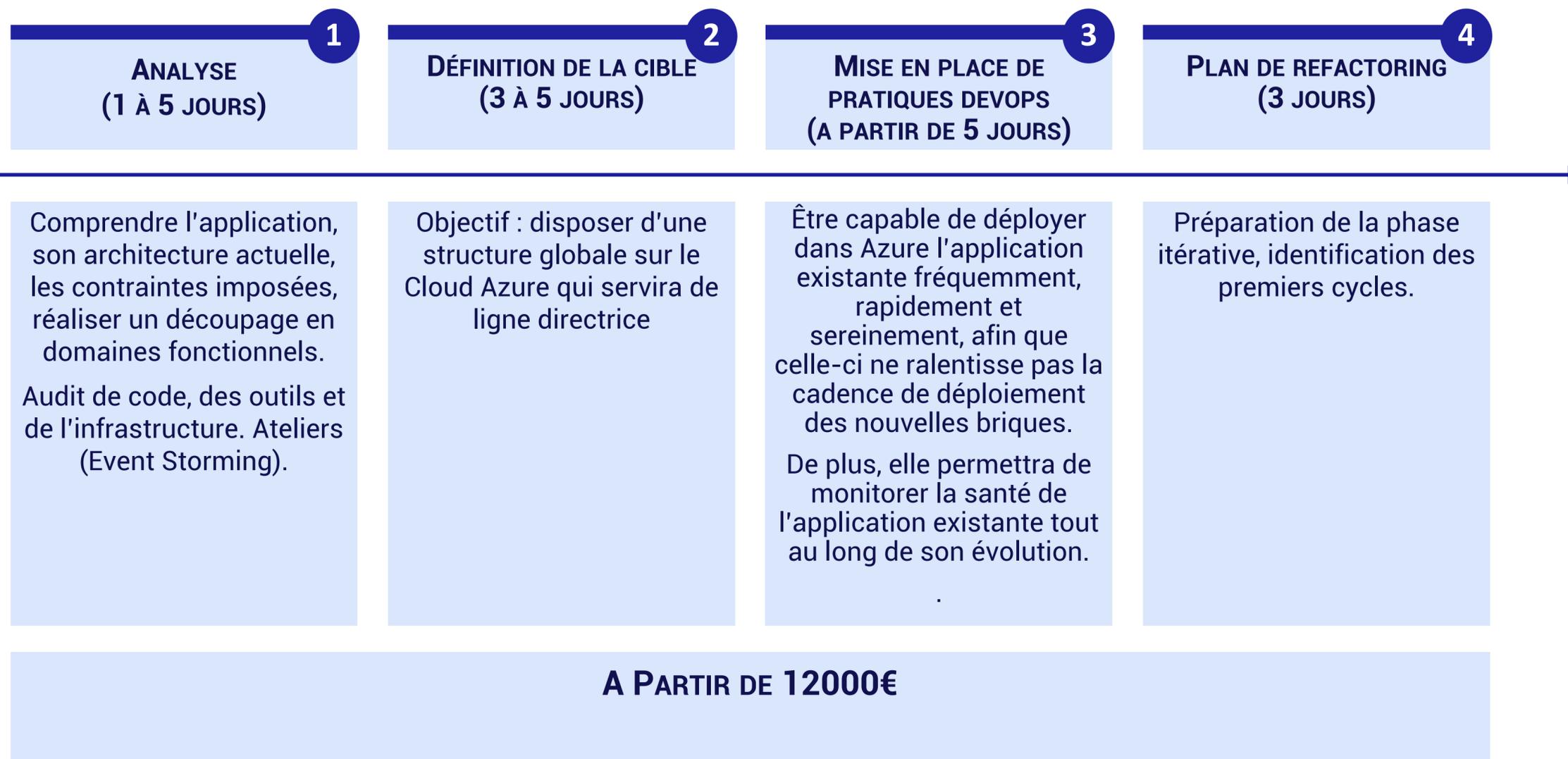
Il pourra être nécessaire de mettre en place des briques techniques nouvelles : authentification, autorisation, instrumentation.

D'autres systèmes gravitant autour de l'application existante pourraient être impactés et pourraient nécessiter des évolutions (ex : rapports).

S'il est estimé nécessaire de s'accorder un filet de sécurité supplémentaire lors des livraisons, la mise en place de [Feature Flipping](#) (activation et désactivation de fonctionnalité à la volée en production) et de [Reverse Dataflow](#) (rapatriement des données depuis le nouveau système vers l'ancien) peut être envisagée.

Afin de faciliter les communications entre l'ancien et le nouveau monde, une couche d'anti-corruption ([Anti-Corruption Layer](#)) pourrait être créée avec les services Azure. Celle-ci servira d'interprète intermédiaire lors de toutes les communications.

NOTRE PROPOSITION PRÉPARATION & AUDIT



cellenza

156 boulevard Haussmann

75008 Paris

www.cellenza.com

Rejoignez-nous sur



cellenza