

## Stratégie

### **Utiliser un logiciel standard comme solution individuelle ? La façon dont les logiciels bancaires peuvent être utilisés pour les workflows**

**La numérisation relève un conflit qui existe dans la plupart des services informatiques : quel est le meilleur choix entre utiliser un logiciel pour tous les utilisateurs, développer en interne ou appliquer un framework ? Ceux qui s'opposent à des logiciels standard, sont principalement ceux qui craignent de ne pas se distinguer suffisamment de la concurrence. Les fournisseurs de logiciels doivent par conséquent rendre possible des adaptations pour que les procédures personnalisées puissent être réalisées. Un exemple du monde bancaire.**

À première vue, ce dilemme ne peut pas être résolu. D'un côté, les entreprises souhaitent se distinguer de la concurrence numérique. Elles affirment qu'elles offrent la meilleure expérience à leurs clients. « Opérations bancaires sans absurdité » promet une institution financière sur smartphones berlinoise, qui a très rapidement fidélisé de nouveaux clients ces dernières années. De l'autre côté en revanche, cela signifie qu'une grande partie des logiciels appliqués doit être développée par les entreprises elles-mêmes, pour que l'offre se distingue réellement des autres institutions financières. Selon l'opinion de certains, les banques doivent même se convertir en fournisseurs de logiciel. Les grandes institutions financières comme ING ou Goldman Sachs sont considérées depuis longtemps comme des entreprises technologiques disposant d'un agrément bancaire. Mais tout développer en interne n'est pas toujours le plus efficace.

#### **Plus de pouvoir aux utilisateurs**

Dans les domaines où les actions sont toujours les mêmes, les logiciels développés en interne ne sont pas rentables. Cela s'applique par exemple à la comptabilité finance et souvent également aux ressources humaines. Dans la mesure où, de toute façon, les clients remarquent à peine ces processus, un logiciel personnalisé serait trop cher. Les frameworks semblent être une solution sécurisée, car ils disposent de modules prédéfinis pour les tâches répétitives et ils permettent en plus de personnaliser les services pour correspondre aux besoins individuels. Mais cela demande beaucoup d'adaptations manuelles qui comportent des risques d'incompatibilité à chaque mise à jour du framework.

La solution idéale serait de créer un logiciel standard permettant au client de définir des processus. L'idée est qu'un moteur de workflow (WFE) intégré dans le logiciel permette de le contrôler via un

langage script (voir figure 1). Le langage script distingue l'état et les demandes de qualité. Chaque statut indique ce qui doit se passer ensuite et chaque demande de qualité vérifie si les conditions définies préalablement s'appliquent, ce qui détermine par conséquent si un état individuel doit être modifié et de quelle manière. Si le workflow est entièrement défini, le moteur compile le code. Au sein du logiciel, le moteur de workflow crée des tâches (tasks), qui sont ensuite traitées par un moteur de tâches. Les modifications qui ont été formulées à l'aide du langage script peuvent ainsi être importées sans mise à jour de la version.

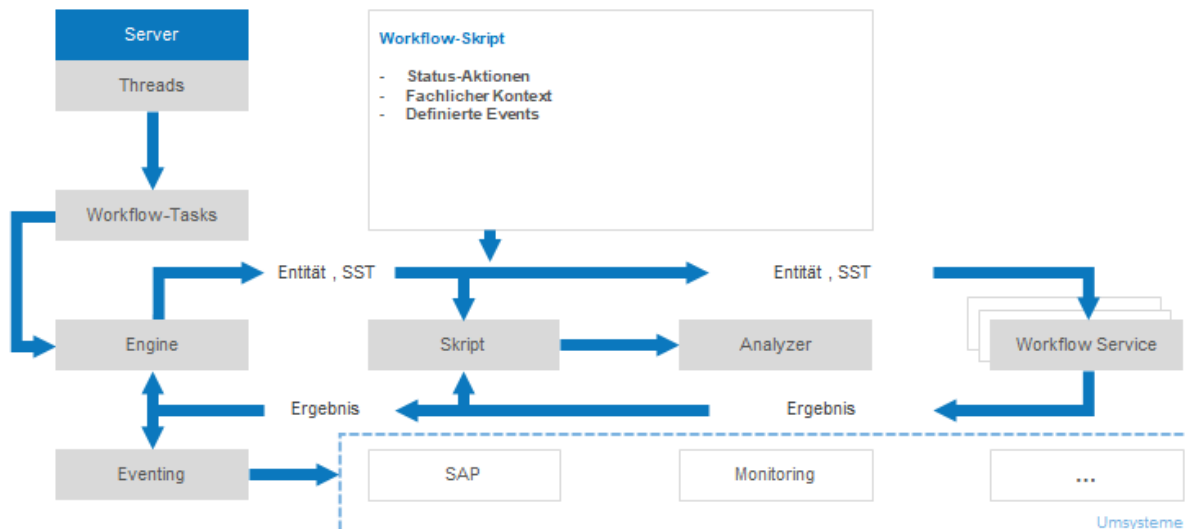


Figure 1 : Un moteur pour script de workflow intégré dans un logiciel standard.

En interne, le logiciel travaille uniquement avec des tâches, en externe il permet des adaptations de l'extérieur à l'aide du moteur de workflow et le langage script fourni. De cette façon, une grande partie du développement du logiciel initialement dédiée à l'éditeur sera transférée à l'entreprise utilisatrice. Cela peut se faire soit via les départements informatiques en interne, soit par une entreprise compétente en la matière. La technique est ainsi séparée de la fonctionnalité métier. Contrairement à de nombreux frameworks, l'utilisateur n'a pas à adapter ultérieurement le code source. L'un des plus grands dangers qui représente souvent un obstacle à de tels projets est donc écarté.

### L'individualité malgré l'uniformité

Cette architecture est particulièrement adaptée aux opérations métier qui se divisent en tâches individuelles, pouvant être traitées dans un ordre défini par l'utilisateur. Pour les institutions financières, les transactions financières sont un cas d'utilisation typique. Si de l'argent doit être transféré, les banques doivent par exemple contrôler si le compte du donneur d'ordre est suffisamment approvisionné et si l'IBAN du bénéficiaire est correct. Mais ce n'est pas tout. Les institutions financières doivent également vérifier une éventuelle suspicion de blanchiment d'argent – en particulier quand il s'agit des paiements vers l'étranger – et si le bénéficiaire est indiqué sur une liste

d'embargo ou de sanctions. Même si cette tâche ne diffère guère de banque à banque, l'ordre, les conditions et les interfaces utilisées vers d'autres systèmes informatiques sont très différentes.

Le schéma suivant montre le comportement d'un logiciel pour le transfert de paiements développé par PPI en cas d'un paiement entrant avec de frais éventuels.

```
Status CHECKINBOUNDCHRGs {
  if payment is inbound and (payment is ourCharges) then {
    if payment is ourChargesReceived then {
      just set status VALIDATERECEIVCHGS and leave step
    }
    if payment is correspondentChargeExpected then {
      if payment is debitAuthorized then {
        just set status CREATEADVICEOFCHGS and leave step
      }
      just set status CREATECHARGEREQ and leave step
    }
  }
  just set status DONE and leave step
}
```

*Figure 2 : Qui paye les frais d'un virement ?*

Dans le cas décrit, le système accepte un paiement d'une autre institution financière et vérifie si les frais sont à la charge de l'institution financière qui lance le paiement – dans le langage professionnel cette règle s'appelle OUR charges. Si ce cas s'applique, le système doit contrôler si ces frais sont compris dans la somme versée, en plus du montant du virement. Dans l'affirmative, le système génère un nouvel état, pour que le montant viré soit vérifié et retenu. Si ce n'est pas le cas, le système examine si les frais peuvent directement être décomptés des comptes de l'institution financière qui a lancé le paiement. Cette vérification conditionne le statut qui sera généré, si le système décompte les frais directement à la banque ou si une demande de paiement est créée. En fonction de la banque, presque toutes les règles peuvent être cartographiées.

Cela est notamment important dans les paiements individuels. En règle générale, les grands groupes cherchent pour chaque région du monde une institution financière pour traiter tous les paiements de manière centralisée. À cette fin, ils définissent certaines exigences auxquelles la banque doit répondre afin de remporter le contrat. Par conséquent, l'institution financière concernée – ou plutôt le logiciel utilisé – doit rapidement adapter un produit standard pour créer une offre individuelle contenant de nombreuses personnalisations. Les groupes imposent souvent les institutions financières via lesquelles le virement doit être routé et à quelle heure le montant doit être comptabilisé. Contrairement au paiement SEPA au sein de la zone euro, les paiements internationaux ne sont pas aussi simples à réaliser que les opérations de masse.

## **Tout ou rien ?**

Dans ce contexte, de nombreuses institutions financières doivent décider si les transactions financières doivent être considérées comme un domaine d'activité stratégique. Certaines institutions financières transfèrent cette activité complexe à une grande institution ou externalisent les transactions financières pour s'affranchir de l'adaptation du logiciel aux constantes nouvelles régulations, et cela sur une plateforme développée en interne. Selon un sondage de l'Association des banques européennes EBA, l'innovation n'existera plus à cause de ce processus, ce qui est surtout un problème pour les petites institutions.

Un logiciel qui peut être adapté de manière flexible et dont le fabricant libère l'utilisateur de toutes les tâches pénibles est particulièrement rentable. Cela s'applique non seulement aux institutions financières qui transfèrent l'argent d'un pays vers un autre, mais également aux secteurs d'activité dans lesquels le législateur fait des spécifications strictes, comme l'énergie, la logistique ou le commerce extérieur. Les entreprises qui participent à la circulation internationale des marchandises doivent accéder aux listes de filtre de la douane via des interfaces, pour s'informer sur les marchandises interdites ou les expéditeurs illégitimes. Si un problème survient dans ce processus, les entreprises risquent de perdre leur statut de fournisseur fiable, ce qui cause des désavantages considérables. De plus, elles ont des systèmes informatiques pour calculer des prix, faire des listes noires et pour respecter les règles internes de conformité – une procédure comparable à celle des banques.

## **Résultat**

À l'ère du digital, chaque fois qu'une entreprise souhaite se démarquer de la concurrence dans un domaine particulier, l'accent est presque automatiquement mis sur le logiciel utilisé – et sur le fait que ce logiciel a été développé en interne ou acheté en externe. Les logiciels achetés sont susceptibles de rendre la différenciation souhaitée impossible, car le développement d'un logiciel n'est souvent rentable que si un grand nombre de clients exécute beaucoup d'opérations du même type sur la même plateforme. La fonctionnalité métier est donc également standardisée. Celui qui réussira à séparer la fonctionnalité métier de la technique résoudra ce problème de base. Les clients pourront ainsi se démarquer suffisamment, même avec un logiciel standard et ils n'auront plus à s'occuper du développement technique – car les nouveautés seront intégrées dans les versions suivantes.

## Auteurs



Depuis 2015, Thomas Riedel est chef de produits senior chez PPI et responsable des logiciels pour le traitement noyau des paiements ainsi que pour la compensation et le règlement. Auparavant, le techno mathématicien a travaillé pendant presque 15 ans dans une entreprise internationale de conseil et d'informatique où il a développé des produits pour transactions financières, en dernier lieu comme chef de paiements.



Mathias Seeliger est architecte logiciel principal chez PPI et responsable de l'architecture des plateformes de paiement. L'informaticien spécialisé dispose de plus de 15 ans d'expérience dans le domaine des logiciels de paiement. Il est chargé surtout de la livraison continue, des tests avec Docker et Kubernetes ainsi que des API.

*Avec l'aimable autorisation d'IT Finanzmagazin :*

*<https://www.it-finanzmagazin.de/standardsoftware-software-workflows-106406/>*