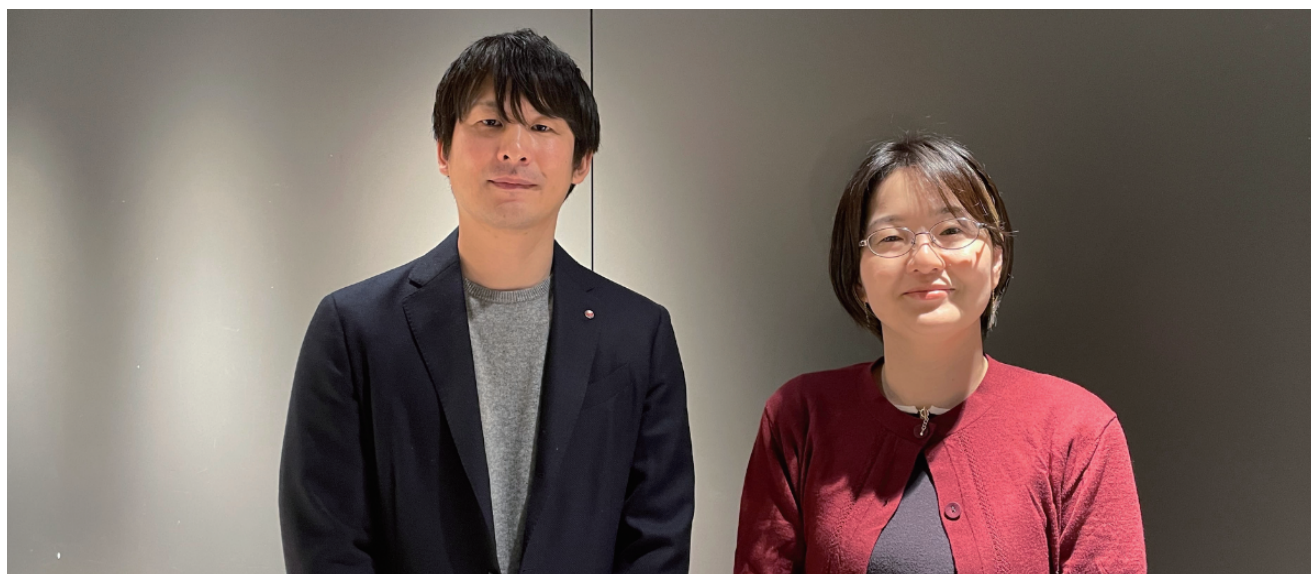


脱モノリシック構造・脱サイロ化へ！ SMBC日興証券が挑むグループ共通 「API 基盤」構築

API は事業変革の中核的テクノロジーに：Kong API Summit Japan 2025 レポート

著 森 英信 編 名須川 楓太（編集部）

金融業界のシステムは長年にわたり堅牢性と継続性を追求してきたが、その代償としてモノリシックな構造が開発の硬直化を招き、新サービスの俊敏な展開が求められるにつれ障壁となっていった。そこでSMBC日興証券は2025年、API基盤にKongを導入し、フロント・ミドル・バックを分離する次世代アーキテクチャへの転換に着手した。2025年11月12日に開催された「Kong API Summit Japan 2025」に同社が登壇し、API戦略の全容とKong導入の理由を語った。また、講演後のインタビューではシステム企画部のメンバーに、API基盤構築の流れと挑戦にあたっての課題・展望を伺うことができた。



「ビジネス戦略＝システム戦略」の時代、API 基盤がすべての中核となる

SMBCグループ傘下の総合証券会社として、従業員約1万人、国内105拠点、海外14拠点を擁するSMBC日興証券。1918年創業の川島屋商店を起源とし、100年以上の歴史を持つ。SMBCグループに参画したのは2009年だ。同社の執行役員システム担当を務める吉岡伸輔氏は、システム部門のビジョンについて次のように述べた。

「我々システム部門は、テクノロジーによってビジネス戦略をドライブしていくことを目指しています。今の時代は『ビジネス戦略＝システム戦略』といっても過言ではないからです」（吉岡氏）

かつては事業部門のサポート役のような立ち位置として見られることが多かったテクノロジー関連の部門が、今や経営戦略の遂行に不可欠な存在となり、戦略そのものになっているようだ。

APIは、そんな同社が描く戦略の中核にあると吉岡氏。APIの強固な基盤を構築することで、クラウド上のデータをAIで活用したり、セキュリティやガバナンスを確保しながらローコード開発を推進したりと、すべての分野で次世代のデジタル基盤を強化することが可能になると説明した。

また、APIと分散アーキテクチャの進化により、APIプラットフォームそのものが事業競争力の中核を担いつつある。まさにその中核として2025年に導入したのが、API

脱モノリシック構造・脱サイロ化へ！ SMB C日興証券が挑むグループ共通「API 基盤」構築

APIは事業変革の中核的テクノロジーに：Kong API Summit Japan 2025 レポート

基盤の Kong だ。

「システムによってビジネス戦略を実現する。そしてビジネスとITの共創体制を作る。これらにより、デジタルと対面営業を組み合わせたサービス品質向上を目指しています。目指すのは『金融の枠を超えたお客さま体験を提供する、先端テックカンパニーへの進化』です」(吉岡氏)

レガシーの崖を越えろ、次世代を戦うためのAPI基盤に求めた要件

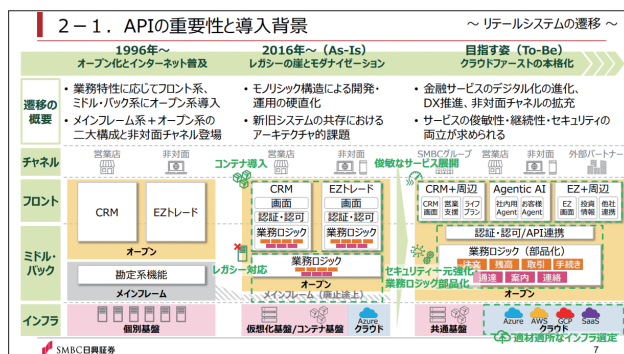
続いて同社 システム企画・IT アーキテクト課で課長を務める平間晃氏が、API 基盤導入の背景と Kong 選定の理由を解説した。

金融業界のシステムは、1980 年代にはメインフレームが中心だったが、1990 年代半ばに入るとオープン化の波が起った。ところがシステム同士が複雑につながり個別最適化が進んだ結果、2016 年頃にモノリシック構造による開発・運用の硬直化という課題に直面する。「新しいサービスの開発や修正が難しくなり、運用も硬直化した」と平間氏は話す。

新たに同社が目指すのは、フロント・ミドル・バックのシステムを分離し、APIを通じて必要な機能を柔軟かつ迅速に提供できる仕組みだ。このアーキテクチャにより、金融サービスをよりスピーディに市場へ届けられるようになる。さらに、業務ロジックとセキュリティを分離し、認証・認可を一元管理することで、安全性と俊敏性の両立を実現できるようになる。

こうした背景から、同社は「外部向けデジタルサービスの短期構築」と「API アーキテクチャ変革」という2つの課題に同時に対応できる製品を求めた。特に重要なのは、オンプレミスとクラウドにまたがって構築されていくシステムを効率的に連携させることだ。平間氏は Kong の選定で重視した5つの要件を示した。

1つ目は「多様な環境への適応性」だ。オンプレミスとクラウドが混在するハイブリッド環境でも Kong なら柔軟に対応でき、堅牢性や高可用性にも優れているため、金融業界のミッションクリティカルな業務にも安心して利用できる。



2つ目は「API ゲートウェイの分散配置・連携」だ。高いパフォーマンスや多様性を求められる金融業界にとって、複数のシステムやインフラにAPIゲートウェイを分散配置し、各システム・インフラを効率的に連携・横断管理できる点は魅力的だ。Kongはこの分散構成に強みがあった。

3つ目は「外部向けサービス対応」だ。顧客やパートナー向けに安全にAPIを公開するためには、当然だが高いセキュリティ水準が欠かせない。そして4つ目は「軽量かつ高速な動作」。オンプレミスとクラウド間の通信を、負荷を抑えつつ迅速に処理できる点が、金融業界の高いパフォーマンス要求にぴったりだった。

最後の5つ目は「機能の拡張性」だ。様々なインフラ環境やシステムにAPIを分散配置することが今回の前提にあるが、配置先ごとに非機能要件はそれぞれ異なってくる。そこで、必要最低限の機能を備えながらも、将来的なニーズや技術革新に合わせて柔軟に機能追加できる点が重視された。「Kongは、現在の課題解決だけでなく、将来的な生成AIとの連携や、社外API公開などの展望にも柔軟に対応できるプラットフォームだ」と平間氏は評価した。

2-2. Kong選定理由		製品に求める必須事項 当社が目指す姿に必要な要件に抜粋	
目指す姿 (To-Be) クラウドファーストの本格化		分類	内容
- 外部向けのデジタルサービスの短期構築と、APIアーキテクチャ変革の両方に対応できる製品が必要 - 特に、アーキテクチャ変革には、オンプレミスとクラウドに構築されていくシステムを効率的に連携させることが重要		多様な環境への 適応性	オンプレミス、クラウドなど多様な環境に導入・適応できること
フロント	EZ画面	API-GWの 分散配置・連携	API-GWを分散配置し、 分散したインフラシステム間を 効率的に連携できること
	投資情報	外部向け サービス対応	外部向けAPI公開に 対応できること
ミドル・ バック	外部API	軽量・高速な 動作	オンプレミスとクラウド間の通信を 軽量かつ高速に行えること
	チャネル 拡充	機能の拡張性	必要最低限の機能を持ちつつ、 柔軟に機能追加できること
インフラ	API-GW① 制御・セキュリティ 実装・可視化		
	API-GW② 制御・セキュリティ 実装・可視化		
	API-GW③ 制御・セキュリティ 実装・可視化		
	API-GW④ 制御・セキュリティ 実装・可視化		
	API-GW⑤ 制御・セキュリティ 実装・可視化		
	API-GW⑥ 制御・セキュリティ 実装・可視化		
	API-GW⑦ 制御・セキュリティ 実装・可視化		
	API-GW⑧ 制御・セキュリティ 実装・可視化		
	API-GW⑨ 制御・セキュリティ 実装・可視化		
	API-GW⑩ 制御・セキュリティ 実装・可視化		
	API-GW⑪ 制御・セキュリティ 実装・可視化		
	API-GW⑫ 制御・セキュリティ 実装・可視化		
	API-GW⑬ 制御・セキュリティ 実装・可視化		
	API-GW⑭ 制御・セキュリティ 実装・可視化		
	API-GW⑮ 制御・セキュリティ 実装・可視化		
	API-GW⑯ 制御・セキュリティ 実装・可視化		
	API-GW⑰ 制御・セキュリティ 実装・可視化		
	API-GW⑱ 制御・セキュリティ 実装・可視化		
	API-GW⑲ 制御・セキュリティ 実装・可視化		
	API-GW⑳ 制御・セキュリティ 実装・可視化		
	API-GW㉑ 制御・セキュリティ 実装・可視化		
	API-GW㉒ 制御・セキュリティ 実装・可視化		
	API-GW㉓ 制御・セキュリティ 実装・可視化		
	API-GW㉔ 制御・セキュリティ 実装・可視化		
	API-GW㉕ 制御・セキュリティ 実装・可視化		
	API-GW㉖ 制御・セキュリティ 実装・可視化		
	API-GW㉗ 制御・セキュリティ 実装・可視化		
	API-GW㉘ 制御・セキュリティ 実装・可視化		
	API-GW㉙ 制御・セキュリティ 実装・可視化		
	API-GW㉚ 制御・セキュリティ 実装・可視化		
	API-GW㉛ 制御・セキュリティ 実装・可視化		
	API-GW㉜ 制御・セキュリティ 実装・可視化		
	API-GW㉝ 制御・セキュリティ 実装・可視化		
	API-GW㉞ 制御・セキュリティ 実装・可視化		
	API-GW㉟ 制御・セキュリティ 実装・可視化		
	API-GW㊱ 制御・セキュリティ 実装・可視化		
	API-GW㊲ 制御・セキュリティ 実装・可視化		
	API-GW㊳ 制御・セキュリティ 実装・可視化		
	API-GW㊴ 制御・セキュリティ 実装・可視化		
	API-GW㊵ 制御・セキュリティ 実装・可視化		
	API-GW㊶ 制御・セキュリティ 実装・可視化		
	API-GW㊷ 制御・セキュリティ 実装・可視化		
	API-GW㊸ 制御・セキュリティ 実装・可視化		
	API-GW㊹ 制御・セキュリティ 実装・可視化		
	API-GW㊺ 制御・セキュリティ 実装・可視化		
	API-GW㊻ 制御・セキュリティ 実装・可視化		
	API-GW㊼ 制御・セキュリティ 実装・可視化		
	API-GW㊽ 制御・セキュリティ 実装・可視化		
	API-GW㊾ 制御・セキュリティ 実装・可視化		
	API-GW㊿ 制御・セキュリティ 実装・可視化		
	API-GW① 制御・セキュリティ 実装・可視化		
	API-GW② 制御・セキュリティ 実装・可視化		
	API-GW③ 制御・セキュリティ 実装・可視化		
	API-GW④ 制御・セキュリティ 実装・可視化		
	API-GW⑤ 制御・セキュリティ 実装・可視化		
	API-GW⑥ 制御・セキュリティ 実装・可視化		
	API-GW⑦ 制御・セキュリティ 実装・可視化		
	API-GW⑧ 制御・セキュリティ 実装・可視化		
	API-GW⑨ 制御・セキュリティ 実装・可視化		
	API-GW⑩ 制御・セキュリティ 実装・可視化		
	API-GW⑪ 制御・セキュリティ 実装・可視化		
	API-GW⑫ 制御・セキュリティ 実装・可視化		
	API-GW⑬ 制御・セキュリティ 実装・可視化		
	API-GW⑭ 制御・セキュリティ 実装・可視化		
	API-GW⑮ 制御・セキュリティ 実装・可視化		
	API-GW⑯ 制御・セキュリティ 実装・可視化		
	API-GW⑰ 制御・セキュリティ 実装・可視化		
	API-GW⑱ 制御・セキュリティ 実装・可視化		
	API-GW⑲ 制御・セキュリティ 実装・可視化		
	API-GW⑳ 制御・セキュリティ 実装・可視化		
	API-GW㉑ 制御・セキュリティ 実装・可視化		
	API-GW㉒ 制御・セキュリティ 実装・可視化		
	API-GW㉓ 制御・セキュリティ 実装・可視化		
	API-GW㉔ 制御・セキュリティ 実装・可視化		
	API-GW㉕ 制御・セキュリティ 実装・可視化		
	API-GW㉖ 制御・セキュリティ 実装・可視化		
	API-GW㉗ 制御・セキュリティ 実装・可視化		
	API-GW㉘ 制御・セキュリティ 実装・可視化		
	API-GW㉙ 制御・セキュリティ 実装・可視化		
	API-GW㉚ 制御・セキュリティ 実装・可視化		
	API-GW㉛ 制御・セキュリティ 実装・可視化		
	API-GW㉜ 制御・セキュリティ 実装・可視化		
	API-GW㉝ 制御・セキュリティ 実装・可視化		
	API-GW㉞ 制御・セキュリティ 実装・可視化		
	API-GW㉟ 制御・セキュリティ 実装・可視化		
	API-GW㊱ 制御・セキュリティ 実装・可視化		
	API-GW㊲ 制御・セキュリティ 実装・可視化		
	API-GW㊳ 制御・セキュリティ 実装・可視化		
	API-GW㊴ 制御・セキュリティ 実装・可視化		
	API-GW㊵ 制御・セキュリティ 実装・可視化		
	API-GW㊶ 制御・セキュリティ 実装・可視化		
	API-GW㊷ 制御・セキュリティ 実装・可視化		
	API-GW㊸ 制御・セキュリティ 実装・可視化		
	API-GW㊹ 制御・セキュリティ 実装・可視化		
	API-GW㊺ 制御・セキュリティ 実装・可視化		
	API-GW㊻ 制御・セキュリティ 実装・可視化		
	API-GW㊼ 制御・セキュリティ 実装・可視化		
	API-GW㊽ 制御・セキュリティ 実装・可視化		
	API-GW㊾ 制御・セキュリティ 実装・可視化		
	API-GW㊿ 制御・セキュリティ 実装・可視化		
	API-GW① 制御・セキュリティ 実装・可視化		
	API-GW② 制御・セキュリティ 実装・可視化		
	API-GW③ 制御・セキュリティ 実装・可視化		
	API-GW④ 制御・セキュリティ 実装・可視化		
	API-GW⑤ 制御・セキュリティ 実装・可視化		
	API-GW⑥ 制御・セキュリティ 実装・可視化		
	API-GW⑦ 制御・セキュリティ 実装・可視化		
	API-GW⑧ 制御・セキュリティ 実装・可視化		
	API-GW⑨ 制御・セキュリティ 実装・可視化		
	API-GW⑩ 制御・セキュリティ 実装・可視化		
	API-GW⑪ 制御・セキュリティ 実装・可視化		
	API-GW⑫ 制御・セキュリティ 実装・可視化		
	API-GW⑬ 制御・セキュリティ 実装・可視化		
	API-GW⑭ 制御・セキュリティ 実装・可視化		
	API-GW⑮ 制御・セキュリティ 実装・可視化		
	API-GW⑯ 制御・セキュリティ 実装・可視化		
	API-GW⑰ 制御・セキュリティ 実装・可視化		
	API-GW⑱ 制御・セキュリティ 実装・可視化		
	API-GW⑲ 制御・セキュリティ 実装・可視化		
	API-GW⑳ 制御・セキュリティ 実装・可視化		
	API-GW㉑ 制御・セキュリティ 実装・可視化		
	API-GW㉒ 制御・セキュリティ 実装・可視化		
	API-GW㉓ 制御・セキュリティ 実装・可視化		
	API-GW㉔ 制御・セキュリティ 実装・可視化		
	API-GW㉕ 制御・セキュリティ 実装・可視化		
	API-GW㉖ 制御・セキュリティ 実装・可視化		
	API-GW㉗ 制御・セキュリティ 実装・可視化		
	API-GW㉘ 制御・セキュリティ 実装・可視化		
	API-GW㉙ 制御・セキュリティ 実装・可視化		
	API-GW㉚ 制御・セキュリティ 実装・可視化		
	API-GW㉛ 制御・セキュリティ 実装・可視化		
	API-GW㉜ 制御・セキュリティ 実装・可視化		
	API-GW㉝ 制御・セキュリティ 実装・可視化		
	API-GW㉞ 制御・セキュリティ 実装・可視化		
	API-GW㉟ 制御・セキュリティ 実装・可視化		
	API-GW㊱ 制御・セキュリティ 実装・可視化		
	API-GW㊲ 制御・セキュリティ 実装・可視化		
	API-GW㊳ 制御・セキュリティ 実装・可視化		
	API-GW㊴ 制御・セキュリティ 実装・可視化		
	API-GW㊵ 制御・セキュリティ 実装・可視化		
	API-GW㊶ 制御・セキュリティ 実装・可視化		
	API-GW㊷ 制御・セキュリティ 実装・可視化		
	API-GW㊸ 制御・セキュリティ 実装・可視化		
	API-GW㊹ 制御・セキュリティ 実装・可視化		
	API-GW㊺ 制御・セキュリティ 実装・可視化		
	API-GW㊻ 制御・セキュリティ 実装・可視化		
	API-GW㊼ 制御・セキュリティ 実装・可視化		
	API-GW㊽ 制御・セキュリティ 実装・可視化		
	API-GW㊾ 制御・セキュリティ 実装・可視化		
	API-GW㊿ 制御・セキュリティ 実装・可視化		

脱モノリシック構造・脱サイロ化へ！ SMB C日興証券が挑むグループ共通「API 基盤」構築

APIは事業変革の中核的テクノロジーに：Kong API Summit Japan 2025 レポート

個別最適から「全体最適」への転換、 グループ内 SIer も製品選定に関与

講演後には平間氏と、同じくシステム企画部の小川志保氏にインタビューをする機会があった。両氏に対し、まずは講演で語られた API 戦略の具体的な経緯と課題について尋ねた。

SMB C日興証券での API 基盤導入の構想は、一朝一夕に生まれたわけではない。「システムのモダナイズについては、2016 年頃から『レガシーの課題に対応しなければ』という機運が高まり、そこから全体構想を描き始めた」と平間氏は振り返る。

構想を進める中で、API の必要性が明確となったのは 2020 年～2021 年頃だった。そこで、2022 年の中期経営計画策定時に現在のアーキテクチャ構想を描いたという。その後、同社はスモールスタートで API 活用への道のりを歩み出した。

「まずは小規模で API を作ってみて、フロントで利用して本当に価値あるものなのかを検証し始めました。そして 2023 年頃に、本格的に API の重要性を確信しました」（平間氏）

同社をはじめ金融業界において、API 連携そのものは決して新しい概念というわけではない。ただし、従来の連携方法には課題があった。小川氏は、「現在はチャンネルごとにシステムが存在し、それぞれがモノリシックな構造になっている」と話す。

顧客向け、社内向け、さらにはその中でも営業向け、な



▶ SMB C日興証券株式会社
システム企画部 IT アーキテクト課長 平間晃氏

どの領域ごとに管理が分かれているという。これでは全体最適の視点に欠けるほか、さらなるスピードや柔軟性の追求といったデジタル時代の要求に応えるのが難しい。

「デジタルチャネルが増え、お客さま一人ひとりの行動様式も多様化する中では、チャンネル単位でシステムが分かれているのは非効率です。機能を共通化・API 化し、それらの API を API ゲートウェイを通してフロント側で用途に合わせて柔軟に使いこなせる環境へとシフトする必要があったのです」（小川氏）

同社で Kong の導入検討が本格化したのは 2024 年だ。当初は別の製品も検討していたが、分散配置の要件に応えられるかが不透明だったため選定に悩んでいたという。「外部公開向けをコンセプトに設計されている製品だと、どうしても機能が重たくなりがちだった」と平間氏は振り返る。

こうした背景もあり、軽量性に優れる Kong の導入を決定したとのことだ。必要最低限の機能でコンパクトに導入できるため、試しやすかったそうだ。選定にあたっては、グループ会社でシステムの開発・運用を手掛ける日興システムソリューションズ（以下、NKSOL）の評価も重視された。

「NKSOL は、長年にわたり当社のシステム開発を担ってきた会社です。グループ外の金融機関や証券会社などに向けた SI 事業も展開しています。特に、堅牢性については本当に目利きのレベルが高いんです。NKSOL の担当者においても Kong を見てもらい、そこでマルがついたことも導入決定を後押ししました」（平間氏）

共通基盤の構築は、 組織のサイロを打破する挑戦

Kong の最初の導入先として選ばれたのは、同社が展開する顧客向けオンライン取引サービス「日興 EZ トレード（イージートレード）」だった。

「EZ トレードは、我々がお客さま向けに提供している取引のプラットフォームシステムです。今後、サードパーティの金融系アプリケーションとも連携することで、お客さま体験のさらなる向上が見込めます。そこを見越して、まずはここで API 活用のための環境整備に向け着手し、成果を出していきたいと考えました」（小川氏）

脱モノリシック構造・脱サイロ化へ！ SMBC日興証券が挑むグループ共通「API 基盤」構築

APIは事業変革の中核的テクノロジーに：Kong API Summit Japan 2025 レポート

外部サービスとの連携というのは、ユーザーのニーズや社会の変化次第でいつ求められるかわからない。しかし必要に迫られてから対応するのでは、そこから2～3年と長い時間がかかってしまう。そのため、将来への備え、すなわち先行投資の意味も含んだ戦略的判断により導入プロジェクトが推進されている。背景には、Kongの技術サポートのほか、企画をSMBC日興証券が担い、実行をNKSOLが担当するという役割分担の最適化もある。

なお、障壁や懸念がなかったわけではない。平間氏は技術面だけでなく組織面でもそれなりのハードルはあったと率直に語る。

「先ほど申し上げました通り、長年にわたって個別最適で堅牢性や品質を維持してきたため、システムは複雑にサイロ化しています。その中で『共通基盤を作る』と一言でいっても、各部門との調整や合意形成をとりながら進めていかなければなりません。これはサイロ化を破る行為となりますので、組織的なチャレンジとしては非常に難易度が高くなります」（平間氏）

ただし、このプロジェクトを機に本当にサイロ化が打破できれば、API基盤の導入以上に大きな効果が得られる。平間氏の発言を踏まえ、小川氏も組織変革の必要性を強調した。

「共通基盤の導入によりアーキテクチャが大きく変わりますから、それに合わせて組織体制も変えていかなければいけないとグループ全体で考えるようになったんです。システム単位で組織が形成されてきた従来の体制から、共通基盤を各システムのSLA (Service Level Agreement) を考慮した形で管理していくような、新たな体制への転換が求められているのだと思います」（小川氏）

開発のアプローチにも変化が起こる。APIによって開発、特にフロントの開発に柔軟性が増し、スピードアップもしてくると、必然的にアジャイルのような開発が増えていくだろう。また、ゆくゆくは内製化の動きも加速することが予想される。

変革は事業部門にも及ぶ。「金融サービスのユーザー側、システム側から、サービスを提供しお客さまとの接点を持つ部門、開発に携わるNKSOLまで、すべての組織変革をテクノロジーの変化に合わせて遂行しなければいけない」と平間氏は語る。小川氏も、今回の共通基盤の導入が良いきっかけになったのではと前向きな見解を示した。

最後には、AI活用を見据えたコメントも出てきた。小川氏は、「AIとの接続は急いでいきたい。その際は、Kongの『AI Gateway』としての機能にも期待したい」と述べる。AI利用が今後増えていく中では、組織内で誰が、何をどう使っているのかを一元的に管理する必要が出てくる。自社製品も含め、サードパーティのAIサービスも様々なものが使われるようになる。そうしたAIとの接続はAPI経由になることが予想されるため、その一元管理もKongのゲートウェイが提供する機能で実現できないか検討する予定とのこと。目指すのは、開発者の自由度を妨げない、かつ安全な環境づくりだ。

平間氏もこれに同意し、「KongのAI Gatewayは比較的新しい仕組みであるため、まずはその技術を着実にキャッチアップして、あくまでも現場のニーズに即したものを実装していきたい」と語る。加えて、「SaaSの製品や各種基盤製品は定期的にアップデートされるため、単純に導入して終わりではなく、現場での運用実態を把握しながら随時アップデートしていく文化を根付かせていきたい」と結んだ。

2026年5月8日 発行

企画・編集◆EnterpriseZine編集部

発行所◆株式会社翔泳社 [<https://www.shoeisha.co.jp>]

本書は著作権上の保護を受けています。

本書の一部または全部について（ソフトウェアおよびプログラムを含む）、株式会社翔泳社からの文書による許諾を得ずに、いかなる方法においても無断で複写、複製することは禁じられています。