

Fintech 2025: Transforming Banks with End-to-End API Factory

Mass Produce Composable System APIs – Last Mile Integration to your Core Backend

In June 2021, The Hong Kong Monetary Authority (HKMA) unveiled “Fintech 2025”. It is a new strategy to give further opportunities to fintech development in Hong Kong. One of the five key focus areas under “Fintech 2025” is the “All banks go fintech” initiative. This initiative aims to promote the all-round adoption of fintech by Hong Kong banks and encourage them to fully digitalise their operations from front-end to back-end. [Read the full details launched by HKMA here](#) 

Bank's Challenges of Digitalisation

According to Everest Group¹, 73% of digital transformation projects fail where they are often driven by challenges with core system modernization. In banks, updating or enhancing legacy applications can prove to be difficult:

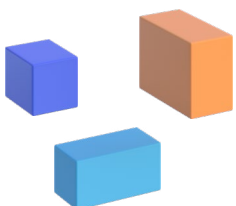
- Limited legacy technical expertise
- Lack of automation and standardization
- Risk of vendor lock-in
- Complex monolithic legacy applications and middleware
- Focusing too much on technology and not enough on the problem

How can OpenLegacy's API Factory help accelerate your digitalisation journey?

APIs enable institutions to securely and seamlessly share data with one another. They form the building block of new product offerings and customer experiences. When it comes to generating APIs from monolithic core systems, not all approaches are equal. Most successful organizations have three key capabilities:

1. Composable system APIs (APIs with business logic)

Financial institutions that create composable system APIs have an easier time supporting the demands of the open banking API ecosystem. Composable APIs are abstract higher-level system APIs derived from the core systems underneath. This makes the APIs easier to reuse across systems. It ensures that if you change the underlying technology, the APIs shouldn't need to change.



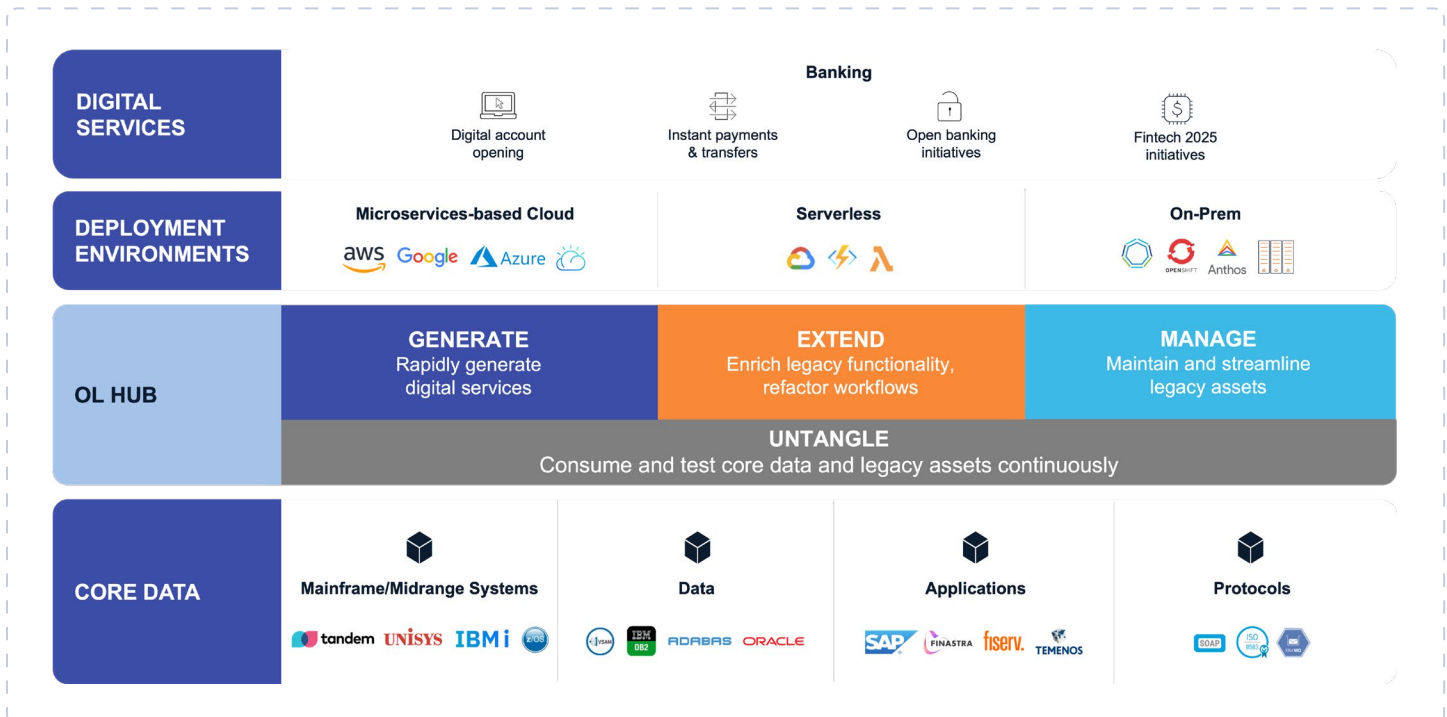
2. An API factory (the ability to rapidly assemble assets into APIs)

A whole group of assets is required to support all APIs, including open ones. An API factory approach is critical for staying on time and on budget.

Building an API factory requires a combination of technology, practices, and organizational mindset.

3. Modern architecture

Microservice are modern and easy to test and deploy. Multiple microservices working to achieve a design is called a microservice architecture. These sets are easy to put into containers. Containers allow for easy deployment into any cloud platform (AWS, Google, Microsoft Azure, etc) using any toolsets (Kubernetes, OpenShift).



Key Features

OpenLegacy enables enterprises to take a cloud-first approach, empower their digital teams and get results quickly and easily.

- Quickly parse legacy assets and create a digital assets repository powered by continuous, automated testing
- Accelerate development and time-to-market with fully reusable “modules,” automated templating, and built-in cloud deployments
- Flexible digital service creation, ranging from no-code to full-code development options
- Avoid cloud and vendor lock-in with multi-cloud interoperability
- Future-proof against obsolescence with language, vendor, and environment-agnostic integration assets
- Ensure IT governance with centralized asset management
- Support for API-First and API-Last methodologies

Business Value

Reduced Time-to-Market

- Accelerate the API generation in hours/days vs. weeks/months
- Fast delivery of new functionality and/or services

Lower Total Cost of Ownership

- 75% lower Total Cost of Ownership (TCO) by automating the entire process of generating services and eliminating the need for a complex technology stack

Resources

- Developers can build legacy integrations by using no-code to full-code options, simplifying the efforts needed. Save time and money

Minimum Risk of Failure

- Requires no changes to on-premise system

Technical Benefits

Automation

- Generate automated microservices directly from legacy systems
- Production ready APIs in hours
- Enable and align with DevOps

Simplification

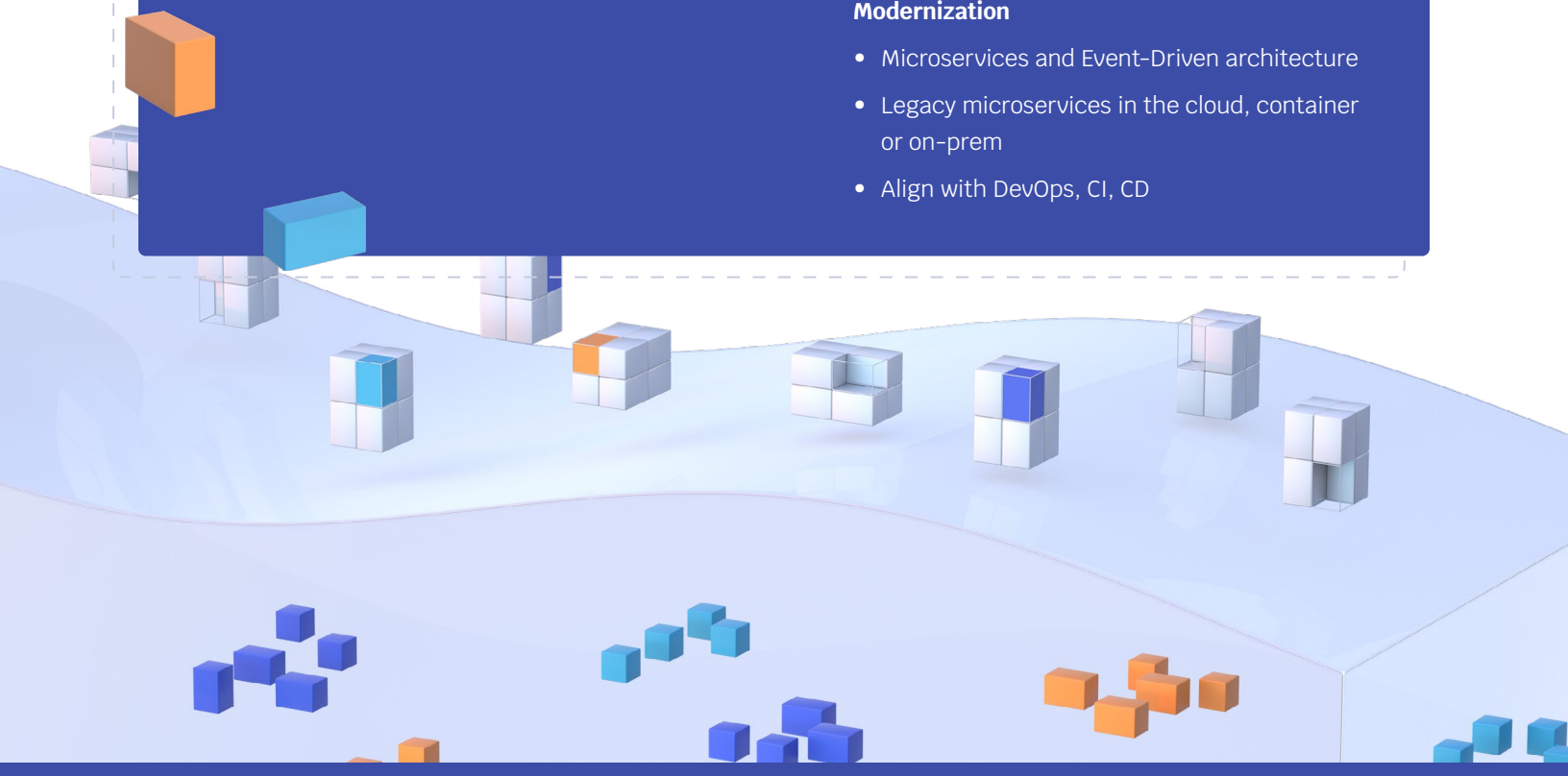
- Independent microservices simplify future legacy migration
- Bypass existing complex middleware and reduce IT stack
- Deployment agnostic. Full flexibility in designing target deployment and security

Standardization

- 100% open standards and code-based platform
- APIs can be built from open contracts

Modernization

- Microservices and Event-Driven architecture
- Legacy microservices in the cloud, container or on-prem
- Align with DevOps, CI, CD



Capabilities

Platforms

IBM Z/OS, IBM i (AS400), CICS,
IMS, Tandem, etc.

Languages Parsed

Cobol, PL1, RPG, Adabas,
Assembler, Natural, etc.

Data Formats

Oracle, VSAM, Adabas, DB2, IMS, etc.

Applications

SAP, Oracle, HOGAN, Temenos,
Finastra, etc.

Design Time Parsers

XML, SQL, JSON, Swagger,
Tuxedo, JDBO, gRPC

Generate into Multiple Formats

No code and low-code (Java, C#, nodeJS)
microservices, serverless functions, REST,
Kafka, RabbitMQ, MQ, SOAP, OData

Successful Transformation Projects in Asia Pacific



About OpenLegacy

OpenLegacy pioneered Digital-Driven Integration as a way to handle the most complex part of application integrations—the connections with legacy systems, mainframes, applications and databases. Based on dozens of years in the legacy space, we realized this need was not adequately met by current vendors. Most often, vendors can handle integrations well except for this critical, complex part of the process. Yet most established organizations rely on legacy technology to run their business. We found a way to do these integrations far faster than any other current method so companies can be agile and innovative today and more easily migrate off the legacy system in the future. It's a solution for today—and tomorrow.