OpenLegacy

Make core systems
as light as the cloud.

eBook

Reduce Your Reliance on Tibco Integration Middleware

www.openlegacy.com

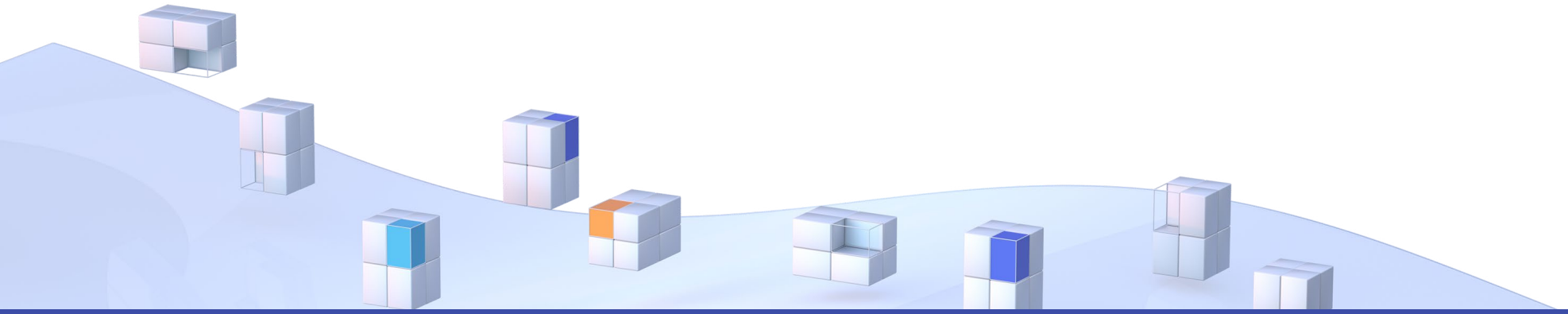
Introduction

The move from using “smart pipes, dumb endpoint” to “dump pipes, smart endpoints” has been happening in software for a number of years now. This is driven by the need to deploy functionality more frequently in small pieces without having to test the whole system on each release.

Tibco is middleware/ESB that acts as a smart pipe and can’t adapt to this new paradigm. This doesn’t mean you should remove all of your Tibco ESB, including complex orchestrations. Just be judicious about using it only when there is a need or benefit. One place you should look to remove Tibco is for building system-level APIs that connect to legacy systems.

Why Legacy Integrations are Different Than Cloud integrations

Cloud systems expose APIs as part of their architectural design. In contrast, legacy systems primarily expose transactions. Transactions are designed to achieve a specific task, not expose a set of assets directly, which makes them tough to use directly as RESTful APIs. To build truly RESTful APIs that fulfill a business case, you need to combine transactions. The combination is best done by development teams with experience handling REST APIs. OpenLegacy parses and creates a language-agnostic repository of the transactions. This way development teams can quickly build APIs from different combinations of the assets without having to go build additional APIs or request more resources from legacy development teams. Tibco treats legacy systems the same as cloud systems and exposes the transactions directly as REST APIs. Tibco’s approach creates additional work combining the APIs. This ebook will explain the differences.



Time to market

OpenLegacy generates usable integration components quickly

Tibco requires the installation of middleware that all connections leverage. This slows down time to market since the middleware typically needs customization to work with many types of assets. Users have to go through the code and figure out the connections. The OpenLegacy platform simplifies this through automatic generation of services based on the specific legacy system types. The platform also supports a variety of end generation options—no code, low code, and full code. All of these can be deployed in days rather than weeks. The no code option allows for rapid prototyping and building while the other options allow for greater flexibility in design. In all cases, the automation makes the system fast to design and build since no manual coding is needed. The digital design teams have all the assets to

> *OpenLegacy helped us become truly responsive by letting us build APIs and microservices in a single sprint—five microservices in two weeks, instead of one in many months.*
>
> bank hapoalim

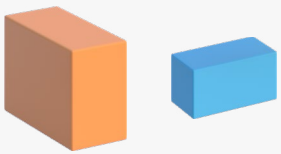
Figure 1.

work without waiting for other teams to get assets to them. The information about the legacy system (COBOL copybooks, VSAM information, screen scenarios, etc) all get automatically parsed using easily modified scripts that leverage a powerful CLI.

Building Multiple Legacy Transactions into a Single API

The OpenLegacy platform builds multiple transactions, screens, or even systems into a single API. This combination allows users to build Smart System APIs instead of low-level APIs that then need to be combined to build anything useful. Tibco requires building low-level system APIs first. These low-level APIs support only a single mainframe transaction or screen. The low-level system APIs waste time and resources since they require a rigorous testing process during deployment. Furthermore, hosting the additional APIs takes up valuable core resources. At run-time, queueing data just to pass among APIs wastes run-time resources as well. A much better way to develop APIs is to build them based on a business need or purpose.

An example of such integration is combining a mainframe transaction with data stored in a VSAM record. Instead of requiring an API for both the transaction and the VSAM retrieval why not build one that can handle both? Tibco requires individual APIs both transactions and VSAM files.



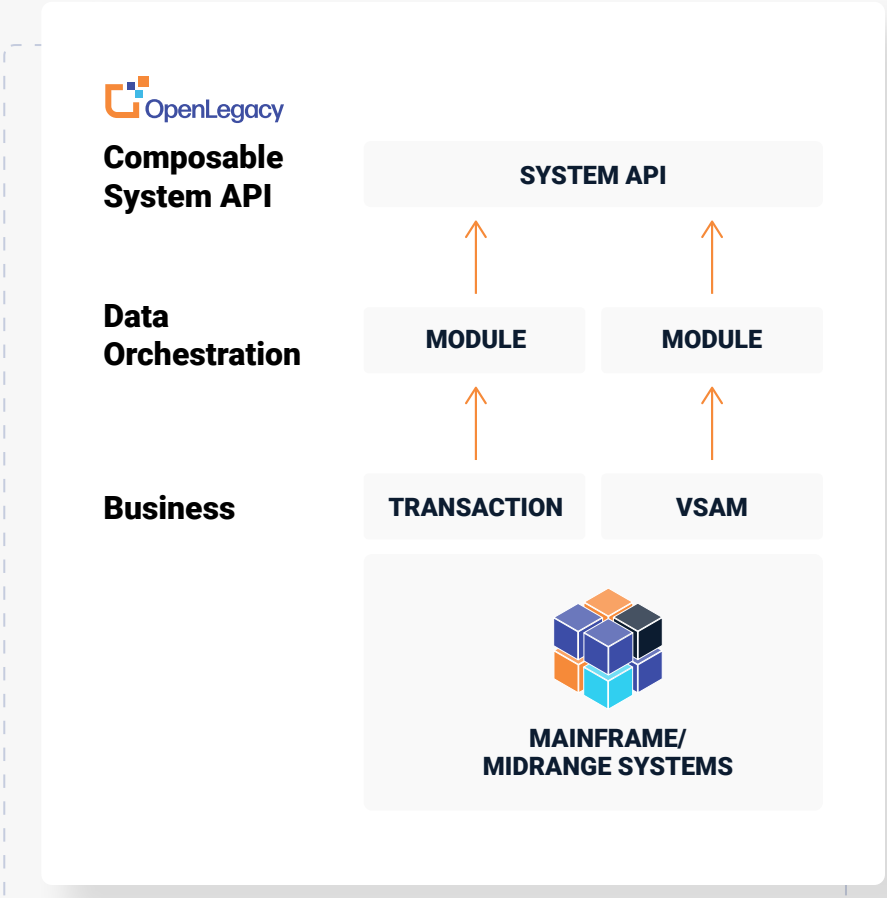


Figure 2. A mainframe transaction and a VSAM data record combining into a System API

Leveraging Citizen Integrators

Many enterprises want citizen integrators, domain experts with programming expertise, to lead their integration processes. Citizen Integrators can help speed a product to market since typically they are a more available resource and understand the business needs. A common way to allow citizen integrators to build the design is by using a graphical flow engine. The issue is most flow engines, such as the Tibco Cloud Integration Engine, only support building flows between APIs. This means citizen integrators have to wait until low-level APIs are built, which interrupts the workflow. OpenLegacy's Flow Engine supports building flows as part of an API project (intra-service orchestration).

The OpenLegacy Flow Engine uses a mapper to pull data from legacy assets and connect them into a design flow with the inputs and outputs of the REST APIs. You can combine fields from multiple legacy assets into one API. Once the services are created by the Flow Engine, you could use the Tibco Cloud Integration Engine to orchestrate data and logic between multiple services or you may use a service mesh and choreography instead, the choice is yours.

Other API Management platforms also have their own interservice orchestration engines that work as well. The OpenLegacy Flow Engine saves time by allowing the creation of Smart System APIs. The key in any development effort is enabling a small team to

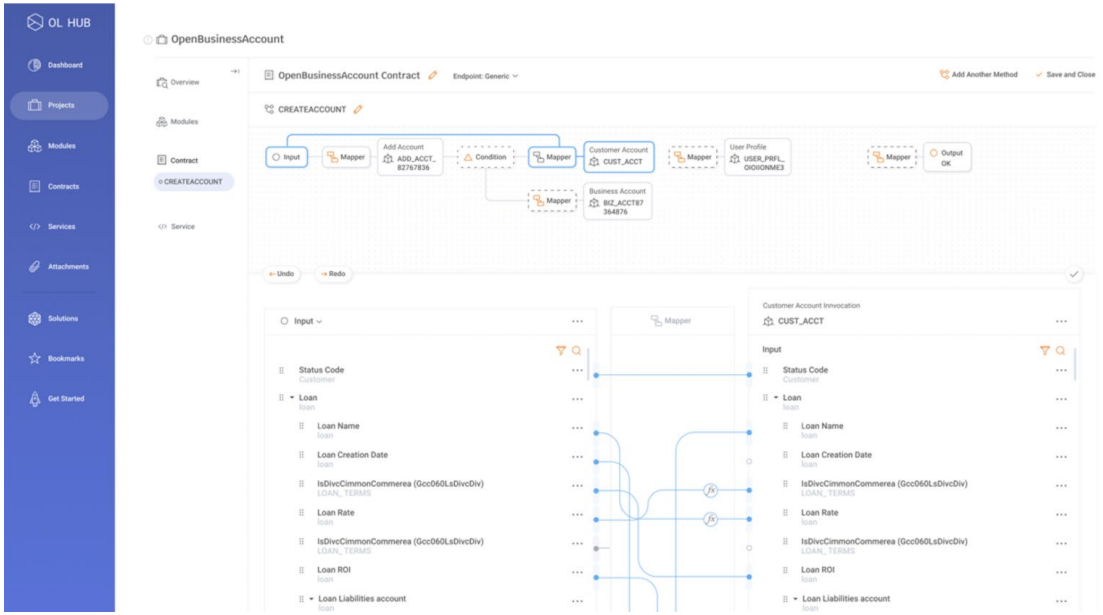
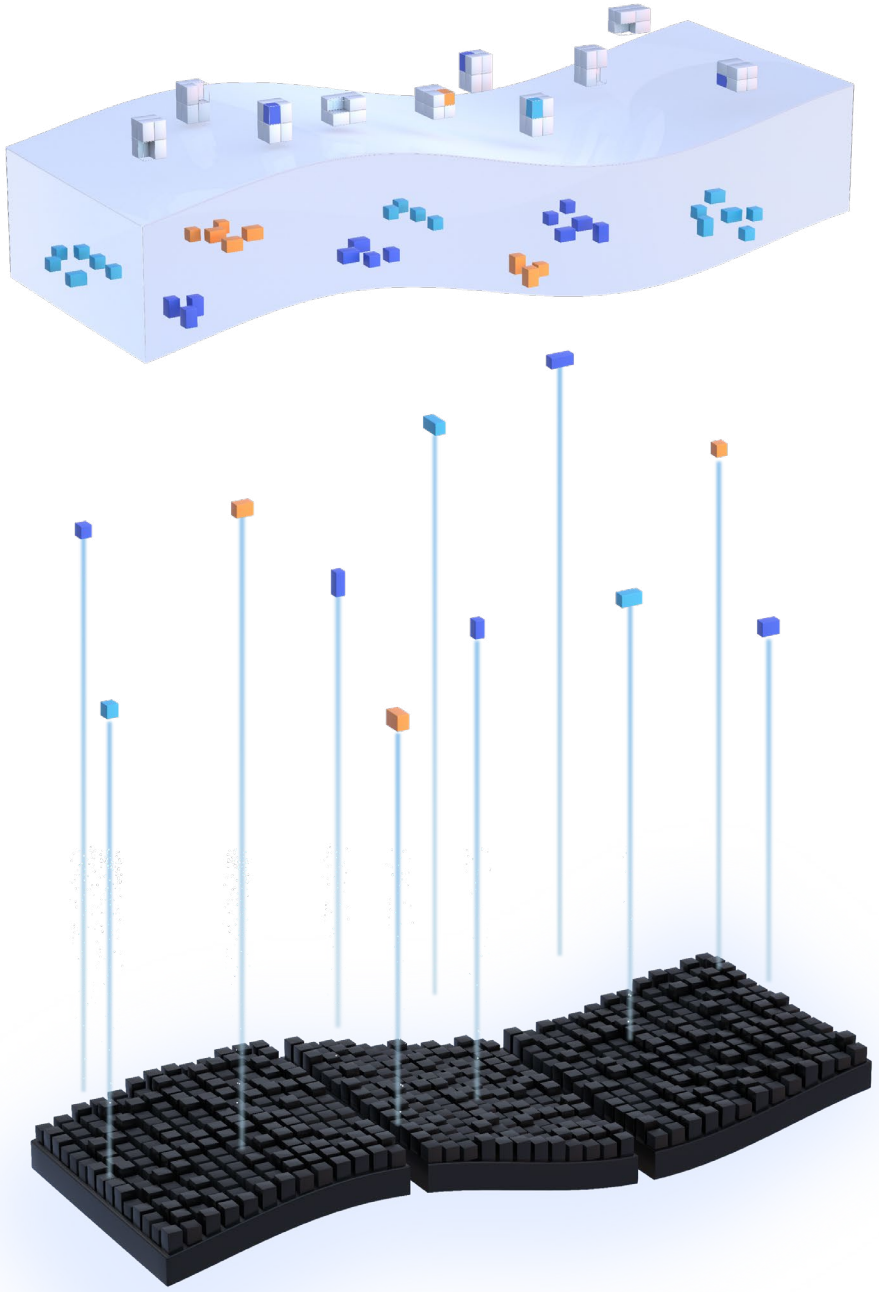


Figure 3. OpenLegacy Flow Engine used for intra-service orchestration

build what they need to without having to wait on others to complete a task first. OpenLegacy's Flow Engine empowers citizen integrators to build the Smart System APIs without waiting for others. It is all there for them to use. This is because the legacy assets already are converted to an easy to leverage language-agnostic format instead of storing those assets in COBOL files citizen integrators don't understand.

DevOps Processes: Building Small Testable Units

DevOps shows the importance of small buildable units. You can't test a whole monolithic system at the frequency required for DevOps since the build takes too long. Additionally, to test a monolith you have to run through every test, which is a painstakingly long process. The Tibco Integration solution turns every system into a monolith since all transactions go through the same set of queues. OpenLegacy is different. Each API is a standalone microservice with a direct connection to the legacy system. It doesn't default to using a queue and all of the system doesn't rely on the rest. This design means it is easy to test individual APIs.



> *With a foundation of DevOps methodology, OpenLegacy helped us step into the new digital age and deliver the access and speed we need to meet today's need and head into tomorrow.*
>
> **Eldad Omer, CTO**
>
> AYALON

Lower the total cost of ownership (TCO) when connecting to legacy systems

Companies can lower their TCO by automating the entire process of generating services. Enterprises automate to turn their API creation into a factory churning out all the needed APIs and free up their development teams. API development teams can't be waiting at each stage in the process for different people to sign off to build their specific pieces as part of the Tibco ESB workflow. Many enterprises are looking to reduce their reliance on ESBs. Integration is a good place to phase out ESBs.

Traditional API Creation

	Design to Deployment (Actual Client Data)
Design	576.5
Build	1296
Test	632
Deploy	374
END TO END BUILD TOTAL LABOR HOURS	2878.5 hours
COST	$179,906
TIME TO MARKET	7.6 Months

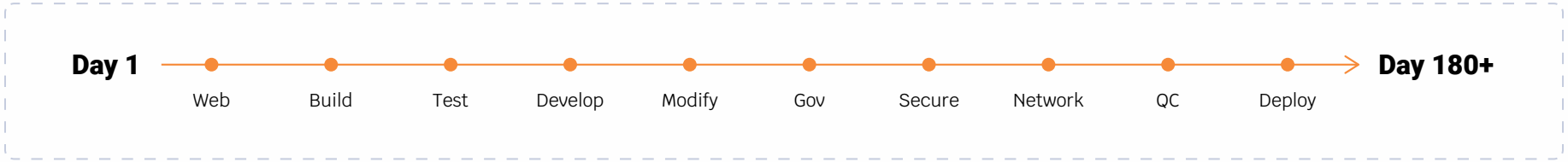
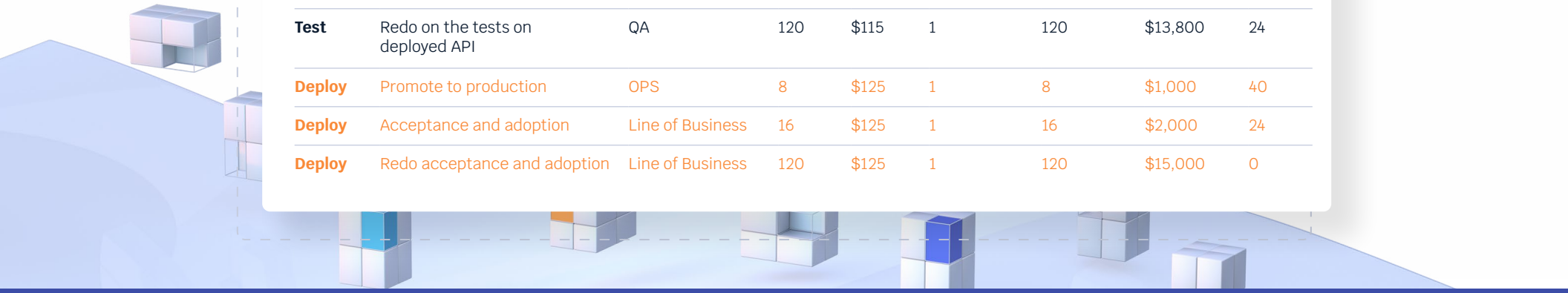


Figure 4. Typical legacy system Integration process

DevOps Stage	Customer As Is Tasks (Manual, Specialty Skill Specific, Testing Intensive)	Team	Man Hours	Hourly Rate	Number of Resources	Calculated Man Hours	Calculated Cost	Delay Time (Hrs)
Design	Project Kickoff	All	8	$100	5	40	$4,000	0
Design	Requirement	Line of Business	8	$125	2	16	$2,000	0
Design	Create specs	API Team	40	$125	1.5	60	$7,500	0
Design	Approve API	Security Team	2	$135	1	2	$270	16
Design	Data Discovery	IT Team	40	$100	2	80	$8,000	16
Design	Architecture Definition	Architecture Team	40	$125	1	40	$5,000	0
Build	Façades and driver programs	IT Team	80	$100	2	160	$16,00	0
Build	Connector configurations	IT Team	24	$100	2	48	$4,800	0
Build	Create low level integration	Integration Team	40	$120	2	80	$9,600	0
Build	Create cross cutting concerns	Integration Team	24	$120	2.25	54	$6,480	0
Build	Iterate on integration	IT Team	40	$100	2	80	$8,000	0
Build	Dynamic and Static Data Validation	Integration Team	40	$120	1.5	60	$7,200	0
Build	Build the API Workflow	Integration Team	16	$120	1.5	24	$2,880	0
Test	Code revies workflow for vulnerabilities	Security Team	40	$135	1	40	$5,400	24
Test	Deploy to test environment	Integration Team	16	$120	1	16	$1,920	0
Test	Run tests on deployed API	QA	40	$115	1.5	60	$6,900	0
Test	Redo on the tests on deployed API	QA	120	$115	1	120	$13,800	24
Deploy	Promote to production	OPS	8	$125	1	8	$1,000	40
Deploy	Acceptance and adoption	Line of Business	16	$125	1	16	$2,000	24
Deploy	Redo acceptance and adoption	Line of Business	120	$125	1	120	$15,000	0

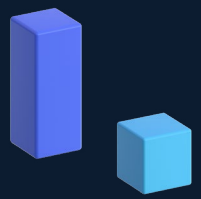


Flexibility in design, development, and deployment

Companies need to build their integrations quickly but also be flexible enough to adapt to any changes that come in the future. Many enterprises with legacy systems chose a MuleSoft 3-tiered approach since process APIs are required when building with low-level systems APIs that heavily leverage ESBs. New digital-only companies typically use a more flexible 2-tiered approach by having Smart System APIs. Now with OpenLegacy enterprises with legacy systems can also build with Smart system APIs no matter whether you choose a 2-tiered or 3-tiered approach.

Smart System APIs Help Developers Design in Flexibility

Smart System APIs allow the building of truly composable systems no matter what approach you use. Service meshes typically are 2 tiered approaches with just a Smart System API that supports your business processes and an experience API to enhance the end-user experience. This is why digital enterprises are so good at building complex systems that adapt quickly based on user preferences. But legacy integrations typically get left out of these systems since they are built using low-level system APIs. With OpenLegacy you can build Smart System APIs to support service mesh-based 2 tiered approaches.



Avoid Proprietary Run-Times to Enhance Your Flexibility

Proprietary runtimes and ESBs also limit your deployment flexibility. Enterprises choose their cloud vendor, on-premise deployments, and how to deploy (microservices, serverless functions, etc) based on whatever criteria makes sense for them. The goal is to support potential future architectures without giving up on the best current options. As part of this process, enterprises also need support for different API Management platforms since times and systems may change. OpenLegacy supports integration to all API Management platforms.

> *Time to market isn't dealing with my backlog—it's when the market changes today, how do I react to it* tomorrow.

BANORTE

20X faster end-to-end API lifecycle design process

Enterprises Need Flexibility in the Legacy Assets They Support

Another important type of flexibility is in what legacy systems a company needs to support (CICS, Tandem, Unisys, IBM i), what languages need parsing, whether support is needed for screens or even hard to reach data types like VSAM, Adabas, etc. The goal is to support all systems required from one platform so your enterprise can build composable assets in the same manner every time. Most integration platforms only support 25% of legacy platforms, OpenLegacy supports 3X more than those platforms.

Below is a partial list of the platform OpenLegacy supports.

Platforms
IBM Z/OS, IBM i, Tandem, etc.

Languages Parsed
COBOL, RPG, PL1, Adabas, Natural, etc.

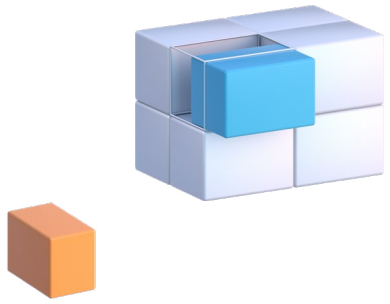
Applications
SAP, Oracle, HOGAN, Temenos, Finastra, etc.

Design Time Parsers
XML, SQL, JSON, Swagger, Tuxedo, JDBC, gRPC

Future-Proof for Ever-Changing Enterprise Needs

Companies periodically change from one API Management to another or choose a different cloud provider. Sometimes enterprises even decide some of their systems should switch from one design paradigm to another. Throughout the history of software development companies switched development languages or models (from SOA to ESBs and now microservices and service meshes). Companies need to standardize their approach and rapidly adjust to new requests in the future.

OpenLegacy does this by creating a language-agnostic repository of metadata from legacy systems. The platform then generates the system in no code, low code, or full code formats in the language of choice for the current situation. It also can generate microservice-based APIs, serverless functions, and whatever your future needs will be. These can be deployed easily into any cloud, or on-premise system. It also supports all different flavors of Kubernetes and other deployment and run-time methodologies around its API Management platform, an ESB-based architecture, and a 3 tier API architecture. This way you aren't tied to one design paradigm. With Tibco, companies are locked into these methodologies.



About OpenLegacy

OpenLegacy's Digital-Driven Integration enables organizations with legacy systems to release new digital services faster and more efficiently than ever before. It connects directly to even the most complex legacy systems, bypassing the need for extra layers of technology. It then automatically generates APIs in minutes, rapidly integrating those assets into exciting new innovations. Finally, it deploys them as standard microservices or serverless functions, giving organizations speed and flexibility while drastically cutting costs and resources. With OpenLegacy, industry-leading companies release new apps, features, and updates in days instead of months, enabling them to truly become digital to the core.

OpenLegacy
Make core systems as light as the cloud.

www.openlegacy.com
sales@openlegacy.com

eBook_Tibco_28.2022