

API TECHNOLOGY SPECIAL

CIOReview

The Navigator for Enterprise Solutions

SEPTEMBER 21 • 2015 CIOREVIEW.COM

IN MY OPINION

Matt Minetola,
EVP & Global CIO, Travelport

CIO INSIGHTS

Judi Flournoy,
CIO, Kelley Drye & Warren LLP

Apigee: Accelerating the Digital Business Transformation

Chet Kapoor,
CEO, Apigee



CIO REVIEW
44790, S Grimmer Blvd.
#202, Fremont, CA-94538

APIs as a Foundation for Systems of Engagement

By Victor Olex, Founder and President, SlashDB

As CIO your job is to design and execute business information flows. While technology and methods change, the underlying objective remains the same: to support business operations and accelerate growth. In the past, fax machines, inter-office mail, and telephone were the tools of the trade. More recently it has been email, shared folders, intranets, and a whole gamut of database-driven information systems. But are those techniques still sufficient in today’s increasingly digitally inter-connected global marketplace?

In this article we taking a look at the REST API and how it can and ought to be leveraged for improving information flows both within ones’ organization and to engage with prospects, clients and partners on the outside.

Transition to the Systems of Engagement

“Systems of Engagement” is a term coined in 2011 by Geoffrey Moore in his paper “Systems of Engagement and the Future of Enterprise IT”. The term refers to the transition from enterprise systems designed around discrete pieces of information ("records") to systems which are more decentralized, incorporate technologies which encourage peer interactions, and which often leverage cloud technologies to enable those interactions. Examples of such systems are the easiest to point out in consumer space: Facebook, Twitter, or WhatsApp, but also – increasingly – in business to business Software as a Service

offers like Intercom, Brand24, and throes of CRM and marketing apps.

But, no matter how compelling and easy to use those apps are, they are disconnected from your business’ systems of record. Working in isolation they cannot fully support custom business processes nor is it feasible to replace all custom-built software with the cloud.

Enterprise's mission critical data are predominantly stored in relational databases such as Oracle, Microsoft SQL Server, and IBM DB2 etc. Business



solutions have been developed on top of those databases conforming to client/server and service oriented architectures (SOA). Those are the vital systems of record, indispensable operations in all departments, but they were never designed for beyond enterprise scale. They are notoriously hard and expensive to integrate with even internally, let alone with cloud-based systems of engagement.

REST API and Resource-Oriented Architecture

Exposing database systems for direct online connectivity is both insecure and not very robust, while batch-oriented data synchronization is almost no better than manual re-keying of data. Thus the typical solution has been a web application built on top of a database-connected backend such as Java Server Pages or ASP.NET. This approach is not extensible. Server-side templates generate HTML, which is only good for human interaction and only through that website.

We need to separate the application data interface from the presentation layer into an API, which then can be utilized by alternative GUIs and for integration with third party systems in a resource-oriented manner.

REST stands for Representational State Transfer and is defined as “a software architecture style for building scalable web services”. The World Wide Web is the largest RESTful system, and

while it is mostly used for distributing human readable content, the underlying protocol (HTTP) is agnostic to the kind of data transmitted. Putting technicalities aside, we think of Resource Oriented Architecture (ROA) as a uniform access layer to all data assets in their unobstructed form for reading and writing in various representations.

SOA helps engineers abstract processing elements of their systems. ROA democratizes access to data. Service oriented systems hide the data

Implement API, and Experience Progress

Resource Oriented APIs preserve and multiply the return on investments in systems of record.

Technical elements of resource-oriented API implementation include: a web server, a web application framework , custom logic for querying databases, enforcing authorization to data, and serializing results to requested format. For web and mobile facing use cases, JavaScript Object Notation (JSON) is

Service Oriented	Resource Oriented
SOAP/XML	Plain HTTP/JSON
Complex programming interfaces	Easy to inspect
Endpoint represents Action	Endpoint Represents State
Transaction, Unit of Work	Addressable Resource
Message (function call)	Update to Resource
API controlled by functional design	API evolves with data
Harder to adapt and scale beyond “enterprise”	Harder to support multi-step transactions
Harder to deprecate functionality	Clients are expected to be resilient to change

behind function façade. Under ROA data resources should be uniformly accessible to both software engineers and domain knowledge workers (data scientists, business intelligence, quantitative analysts, and salespeople). Here’s how we contrast the two approaches:



a must-have representation, while for internal facing systems XML and comma-separated values may be preferred.

SlashDB, of which I am the CEO, is an off-the-shelf component, which saves up to 90% in API development time. It automatically exposes database tables to authorized users and applications as online resources for reading and writing in alternative representations. Related records are hyperlinked forming a web façade over relational data.

Regardless of how your REST API will be built around your systems of record, the benefits will be profound:

- Accelerate time to market for mobile business apps
- Engage and interoperate with clients, partners, or vendors on API level

“REST APIs fit right in with existing HTTP infrastructure such as reverse proxies for caching and search engines for indexing available resources

- Add internal search engine, and find anything in your hyperlinked data cloud
- At last triumph over data silos and unlock productivity

Sharing and engaging nature of the web can now work for your business in full context of its data resources. For example, instead of sending your client a new copy of a price list every month in an Excel report, one could easily produce a self-updating spreadsheet, which calls secured API to obtain the latest prices. That same API could be used as a backend for a mobile app or internally for data analysis or reporting.

REST APIs fit right in with existing HTTP infrastructure such as reverse proxies for caching and search engines for indexing available resources. Other commonly required features such as keys issuance, developer portal generation, call rate throttling are often handled by API management services. Leading vendors in this space are 3Scale, Mashery (TIBCO), Apiphany (Microsoft), and Layer7 (Computer Associates).

Web APIs are now crossing the chasm. Businesses, which cannot evolve their systems in that direction risk loosing their market position to those that can. [FR](#)