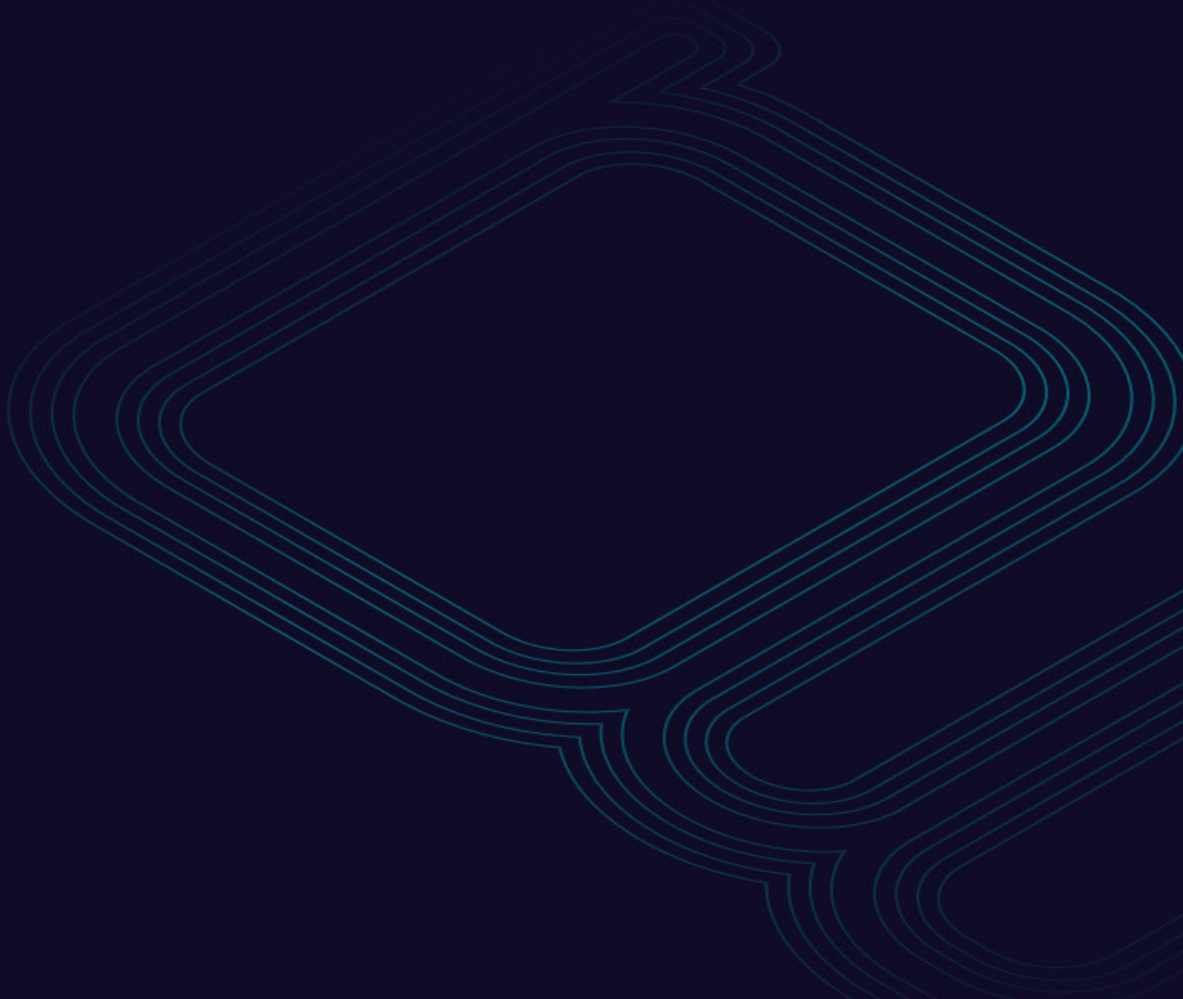
gravitee.io

The Future of APIs:
11 trends you need to know

11 trends that we are watching around API modernization, security, and governance, and more.



Introduction: **ever-increasing complexity**

The API ecosystem is evolving rapidly. The availability of a myriad of new data sources, coupled with a growing demand for real time data is accelerating the adoption of protocols, API styles, and Pub/Sub technologies (i.e. event management and streaming) that support streaming data and event-driven APIs such as gRPC, Kafka, Websocket, Webhooks, SSE, MQTT, Solace, etc.

The dominance of REST is being challenged by asynchronous APIs in the quest for more efficient microservices, governable IoT, and more profitable uses of productized APIs and data. And, while technically outside of the asynchronous universe, GraphQL—while adopted slower than some may have thought—is still taking portions of the universe by storm as a solution that might replace the REST request with the arguably more efficient (for certain use cases) GraphQL query.

The result? The API ecosystem in your organization is becoming ever more complex.

Controlling API complexity is critical for businesses to accelerate deployment, make infrastructure more reliable, strengthen security posture, and discover new revenue streams.

In this ebook, we'll dive into 11 trends that we see unfolding as answers to the API complexity problem:

1. The need for event-native API Management
2. The rise of the sync/async machine
3. API-first as the norm
4. IoT, IoT, IoT
5. Securing systems down to the API level
6. APIs for all: democratizing the API lifecycle
7. Automation of API standards, governance, and control
8. Machine learning evolving from API consumer To API controller and security enforcer
9. The death of the API key, As nuanced API security enables new architectures
10. The demise of direct database access
11. Monetisation of APIs internally and externally

1. The need for Event-Native API Management

Mentioned in the introduction, asynchronous APIs, the AsyncAPI spec, and event-driven architectures are taking the world by storm.

Again.

While asynchronous communication and event-driven systems and APIs aren't new, they are uniquely positioned to solve for very present (and future) demands: real-time data, real-time user experiences, IoT-driven revenue streams, and lower infrastructure costs—just to name a few.

Powering these systems is the asynchronous, streaming, and/or event-driven API.

And these APIs need to be managed, secured, and governed.

The challenge? In order for API Management and Security solutions to support this different communication paradigm (asynchronous communication) and various new kinds of APIs and backends, those API Gateways and Management consoles also need to be built using approaches (like reactive programming) that natively support asynchronous communication, persistent connection, and streamed data.

"API agility means looking beyond REST."

-David Mooter, Senior Analyst at Forrester

We call this approach "event-native," and it's very similar to the need for "cloud-first" companies to find and implement "cloud-native" vendors and solutions in order for utmost interoperability.

"We now enter the next frontier of event-driven architecture (or EDA) – where the once-humble pub/sub feed is now a nexus, and an API is no longer considered a mere enabling mechanism for requesting and retrieving data between services.

Event-driven architectures are really starting to explode on the market. A recent industry survey reported that while 85% of respondents say they are 'on the journey' to EDA, only 13% claimed to have arrived at the 'promised land' of a reliable, scalable EDA approach."

-Jason English, Principal Analyst at Intellyx

2. The rise of sync/async machine

Called out in our whitepaper on event-native API Management, we expect most organizations to take a hybrid approach to implementing synchronous and asynchronous APIs. This approach will take two major forms:

1. Some lines of the business powered by synchronous-API-powered client applications and backends and some powered entirely by asynchronous APIs at both client and backend layers
2. Client applications powered by REST APIs that must "shake hands" securely and reliably with evented backends (i.e. REST-based apps that need Kafka backends to ingest data via HTTP Post and then must be able to consume events from that Kafka backend via HTTP Get)

This hybrid approach will be the result of certain use cases requiring it: either the need to keep critical, synchronous systems intact, and/or the need for organizations to make the move from synchronous systems and APIs to asynchronous systems and APIs slowly while keeping the critical "lights on."

"The most sophisticated organizations I've talked to–in addition to unified platform teams and unified governance, and unified taxonomy–also unify the lifecycle for both REST APIs and Events."

-David Mooter, Senior Analyst at Forrester

3. API first as the norm

Adopting an API first strategy will be increasingly common in the future. Postman's 2020 State of The API report found that 39.2% of teams already design and define APIs and schema before beginning development.

By focusing on building APIs that can serve all applications, governed by a contract between services that all teams follow, significant efficiencies can be gained. By adopting this approach, teams can confidently mockup APIs and test dependencies based on the agreed API definitions, meaning wait time for APIs to be completed can be minimized, making it easier for multiple development teams across an organization to work in parallel.

Well-designed, fully documented, consistent APIs significantly reduce the risk of failure. They encourage API use and facilitate code reuse, leading to a significant reduction in delivery times. The use of mock APIs can help the identification and resolution of many problems before time is expended on coding.

"We have API only platforms and we layer various applications on top of them."

– John Duffie, Senior Principal Software Engineer, Xylem

4.IoT, IoT, IoT

No, IoT isn't new.

Yes, IoT is becoming more and more prominent. And more and more API-relevant.

Whether it be IoT connected refrigerators running on MQTT or sensors in a factory that stream data to utilities companies, organizations are finding more and more ways to mine, consume, stream, and (often times) sell the data to vendors that can can use that data to provide awesome customer experiences.

This data is streamed and exposed via APIs, oftentimes asynchronous APIs. These APIs could be accessed through an API Developer Portal in order to expand the footprint of this service as far as possible while also making it self-service. And they'd use an API Gateway and authentication policies to make sure that only the right partners and customers could consume this data.

5.Securing systems all the way down to the API and communication level

According to Salt's State of API Security report, organizations are up against a "681% Increase in API Attacks." As APIs become a main way of exposing data and services to external consumers, this will only become more and more relevant.

This trend ushers in the need for forward-thinking organizations to implement and enforce security measures that go beyond the application and log-in layers. This requires access control and access management all the way down to the API call, event, message, and/or communication methods that govern the brokering of these kinds of data.

Securing systems down to the API level allows you the flexibility to enforce strict security (i.e. FIDO2 as an added MFA factor) in areas of applications where sensitive data is housed and brokered (i.e. the payments portion of a banking app) and less strict security measures where less sensitive or non-sensitive data is housed or brokered (i.e. the "Market overview" of that same banking app).

"Among the most sobering findings:

- *95% of the more than 250 survey respondents said they've experienced an API security incident in the past 12 months*
- *Only 11% of respondents have an API security strategy that includes dedicated API testing and protection – 34% lack any security strategy at all for APIs*
- *Shift-left tactics are falling short, with more than 50% of respondents saying developers, DevOps, or DevSecOps teams are responsible for API security while 85% acknowledge their existing tools are not very effective in stopping API attacks*
- *When asked their biggest concern about their company's API program, 40% of respondents highlighted gaps in security as their top worry*
- *94% of API exploits are happening against authenticated APIs, according to Salt customer data*
- *Stopping attacks tops the list of most valuable attributes of an API security platform*
- *40% of respondents are grappling with APIs that change at least every week, with 9% saying their APIs change daily"*

-Salt State of API Security Report, Q1 2022

6. APIs for all: Democratizing the API lifecycle

Maximizing the success of an API first strategy will mean equipping the whole business, not just developers, with the tools they need to join in. FinTech challenger Tide is using the Gravitee.io API Design studio to close the gap between product management and development. Product managers working on a value proposition can create a mind map style API shell that can be rolled up into a Swagger document for an engineer to build from. Having this ability deepens shared understanding of the product and enables earlier cooperation and technical discussion, accelerating code delivery.

Roche pharmaceuticals uses a similar pattern to automate data collection and sharing in their laboratories to reduce the time to scientific discovery. For the past two years the company has been training research scientists in using Python, not just for graphing results, but to build integrations with APIs. Previously only their developers worked on creating bridges between the physical equipment running experiments and data stores. Due to the scale of automation which Roche are aiming to bring to their research, they decided to enable scientists to build, consume and perform unit testing on their own APIs. Roche business analysts, who previously would have talked to the business and then written specifications, are now embedded in hardware integration teams, and generating APIs. Analysts are now creating mock APIs together with the scientists, using the Gravitee.io API Design Studio and creating mock data; meaning even before a developer starts work the team can have something running and can test against it.

The API definition files created in this process can be used to automate the documentation and application of security policies and effective governance of the API, significantly speeding development and deployment.

"Using the API gateway, even before we have a single developer looking at the API, we already have something running and we can start testing against it."

- Matthieu Croissant, Scientific Solution Architect Roche

"Blurring the lines of traditional roles is going to be ever more important. If you need an organisation to be API first, you really need everyone to be API first, so you need to give non-traditionally tech types the tools so they can join in. If they can mind map into a skinny API shell and that can get consumed as a swagger doc for an engineer to iterate on, fantastic! That means you can go faster."

- Guy Duncan, CTO Tide

7. Automation of API standards, governance and control

API Gateways with integrated Identity and Access Management (IAM) will become common core infrastructure elements to enable the API curation, observability, consistency, and governance required by an API for all approach and the secure integration of APIs across internal and external boundaries.

Water technology company Xylem has gone through a series of successful acquisitions, and as a result it has numerous development teams, each with their own legacy code and processes. Often these teams are building streams of data ingested from millions of sensors for leak detection and water quality monitoring. Typically streaming data is done within a group owned security boundary. As systems are integrated across Xylem these streams are shifting from single producer single consumer, to having multiple consumers across security boundaries. Addressing the challenges of crossing these security boundaries, controlling access to message brokers, and managing permission for access to data which does not have a predefined end point will require the tight integration between API Gateway and IAM.

Ensuring consistency of schemas is essential to efficiency when working with multiple development groups. This is particularly important with streaming technologies using technologies like Apache Avro or Protobuf, where tweaking latency performance is required. Using a schema registry is an effective pattern to build controls for these challenges. The schema registry is a central authority of schemas and versioning, designed with up-front rules of who owns what schema, and who can evolve the Schema. When this is coupled with the message broker, leveraging the registry teams can be confident of the version of the schema, and see how to interpret it.

Tide, working with numerous banks as data providers, manages significant variability in API maturity, through clarity of its own API standards which it rigorously enforces through processes and automating governance, security, and hygiene in their API Management system. Where third party APIs are found, the team dedicates engineering resource to working with the partner to bring APIs up to the required quality level.

This combination of control, collaboration, and automation is critical to maintaining development velocity.

"An API is useless unless it is delivered with consistency and quality."

- Guy Duncan, CTO Tide

8. Machine learning evolving from API consumer to API controller and security enforcer

APIs are an essential part of the data pipelines feeding most machine learning systems, with the machine learning algorithms increasingly calling APIs directly.

Less common, but seen as a valuable emerging tool, is the application of machine learning technology to manage the provisioning and control of APIs based on consumption data. For example, using the ML to identify anomalous connection or consumption patterns, then working with the API gateway and IAM to throttle or withdraw API access and trigger alerts so the behavior can be investigated.

Further into the future expect ML systems to drive increased resilience by enabling self-healing systems to work round bottlenecks and halted processes.

"In the next two to three years we will see a massive increase in applying ML for API management and provisioning. With decomposed federated architectures, it becomes a zoo pretty quickly, ML is something that can help us there."

- Guy Duncan, CTO Tide

9. The death of the API key, as nuanced API security enables new architecturs

Gartner have predicted that by 2022 the most frequent vector for enterprise web applications breaches will be API security issues.

As the complexity and extent of the API ecosystem grows, and with it the attack surface, API security must become more nuanced.

API keys as a simple text string are notoriously accident prone; left in comments to source code, embedded in firmware, accidentally emailed, and historically have tended to be left active for months if not years.

The adoption of fully integrating Identity and Access Management (IAM) with API management (APIM) and API design tools is accelerating. Combining IAM and APIM enables token generation for API access based on current permissions, for flexible durations, with fine grained control over data flows based on role or other attributes. This lends itself to application on serviced mesh architectures. By using a single identifier for all activity, auditing is enhanced, and enforcing security programmatically with things like step up authentication is simplified.

Consumer expectations of using biometrics and social media accounts to log in are easily met via common frameworks such as OAuth2, as is supporting legacy authentication technologies.

The security and usability case for combined IAM and APIM is compelling and will soon consign API keys to a niche history.

"In increasingly complex systems you have a mixing of concerns, you have business logic code, you have access management code. What would be nice to do is separate those concerns, making it so even as you enforce it everywhere you can isolate those actual actions of access management. Certain technologies like service mesh could be a good possibility in that regard. You could decouple microservices from transport between the microservices, that might be a possibility."

– John Duffie, Senior Principal Software Engineer, Xylem

"It is interesting to see that people view API security as one of the main fault points around APIs, as there are well-established protocols around authentication and authorization."

- David Brassely, Chief Architect and Co-Founder, Gravitee.io

"We provide APIs with multiple authentication methods from the API gateway, but behind the scenes on service mesh we only support one gold standard."

- Matthieu Croissant, Scientific Solution Architect Roche

10. The demise of direct database access

Nuanced API security will see the demise of direct database access being provided in many API first organizations. Tide, for example, has already implemented an OpenID based IAM system for its customers. The IAM, an integrated part of the API manager, ties individual permissions to APIs, which in turn unlock the relevant API features and functionality that unlocks the data required. So instead of directly accessing data, it all flows via the APIs, which are the only route to the endpoint.

"Traditionally you would always talk about database permissions, we are now talking API permissions, you have to provision API provision which then gets you to the data, as opposed to directly having access data. At Tide we say it all comes through API, through the endpoint and data can only accessed via that. This is a huge seismic change compared to 5 years ago." -

Guy Duncan, CTO Tide

11. Monetization of APIs internally and externally

Monetization of APIs is moving from aspiration to implementation as API Gateway technology matures.

The visibility into API consumption that the latest generation of gateways provide enables APIs to be evaluated in terms of the ROI they deliver through the data they enable to be consumed or the business use cases they deliver against. Access to this visibility of value enables the justification of investment in API development.

In combination with IAM to control access, advanced API gateways can now also enable APIs to become directly revenue generating rather than relying on traditional logical flows via applications.

"We want to accelerate the monetisation of those APIs that have the greatest value to us. Monetizing them either through monitoring and just making the business aware of the importance of that API, or by wrapping up data, creating data marts etc. We think that is incredibly important."

- Guy Duncan, CTO Tide

Wrapping up

SOAP is not entirely washed up, REST is not sleeping, but the future of APIs lies in exploiting new dynamic protocols. The teams that take advantage of the advances in APIs, API Management, and API Security to keep up with the trends outlined in this ebook can look forward to more secure, more reliable systems, increased delivery velocity, and—potentially—more revenue via exposed and monetized APIs.

You can do this with Gravitee.

Gravitee is the world's first event-native API Management platform that enables teams to manage, secure, and expose synchronous and asynchronous APIs–effortlessly.

"Firewalls are ubiquitous, they are everywhere, we think API gateways are going to become the next core infrastructure like a firewall."

- Guy Duncan, CTO Tide, Gravitee API Management and Access Management customers

To find out more, feel free to book a demo. We hope to hear from you soon!

How to **Contact Us**

gravitee.io**/contact-us**

If you're interested,
and want to reach out,
you can contact us here

gravitee.io**/demo**

If you'd like to skip (some of)
the Sales pitch and see a demo,
you can book one of those here

community.gravitee.io

If you want to give OSS a go,
check out our community forum,
where you can find links to our
github repo and connect with
the folks who have driven over
350,000 Docker pulls / month