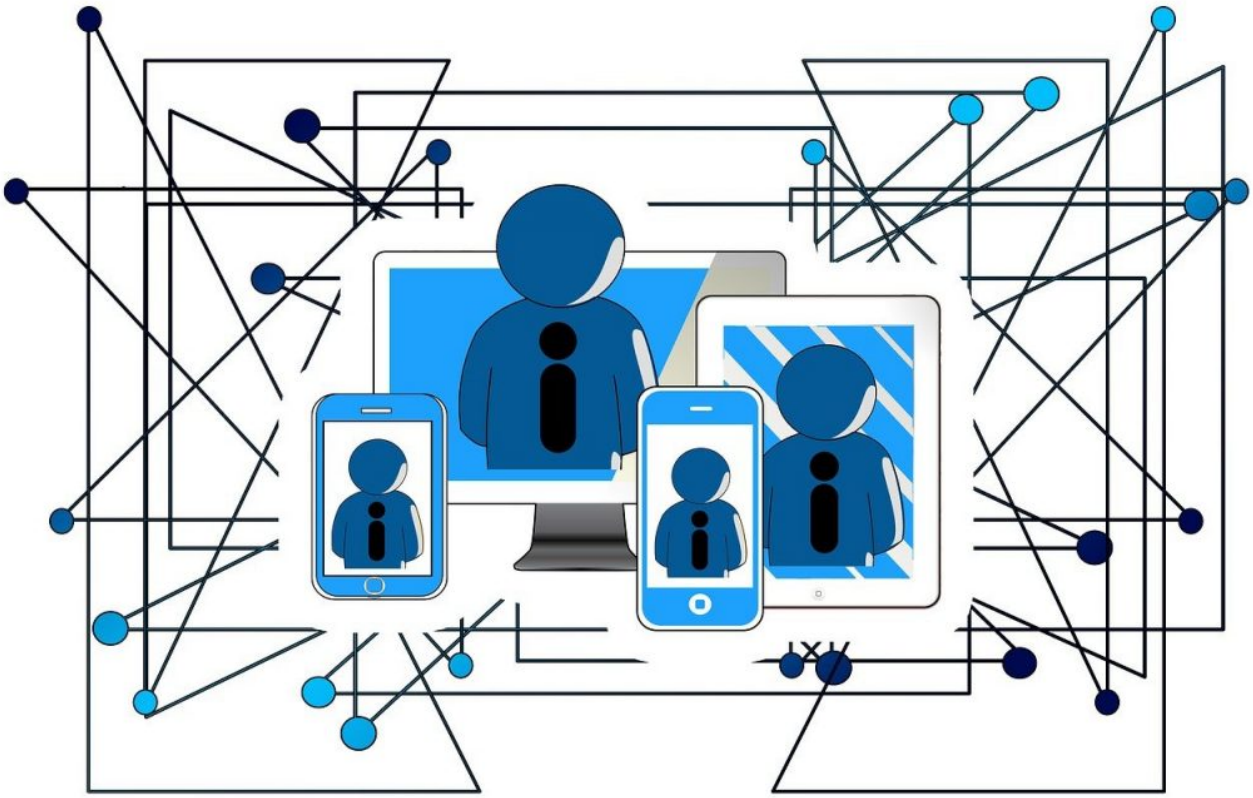
API MANAGEMENT / CLOUD NATIVE / MICROSERVICES / CONTRIBUTED

API Technology Trends in 2021

12 Aug 2021 3:00am, by Milap Neupane



Milap Neupane

Milap Neupane is a Senior Software Engineer and writer for Draft.dev. He unity and is the organizer of Golang ng street photography.

hey are increasingly turning to provide core services to internal specific user interface.

oviding secure and reliable real- QL and gRPC are giving

Keeping pace with the many different technology trends in software development is a big job in itself. Understanding these emerging API trends and how they will affect your next product is one of the many things you need to worry about as you progress through your career as a developer.

In the first part of this article, I'll share a few of the key API technology trends you should watch for in 2021. I'll offer some insight into each and how you can lean into these trends to build better, faster, more flexible software. In the second part of the article, I'll share some of the emerging technologies being used to realize these trends. You'll learn more about how the next generation of APIs is being built, tested, and scaled.

1. Real-Time Data Synchronization

API consumers increasingly want immediate results, so real-time data access is essential. Some tasks, like booking an Uber or ordering a pizza, are relatively easy to build with traditional REST-focused patterns, but other problems like fraud detection and reporting can be more complicated to solve in real-time.

For some industries, real-time API delivery is a mission-critical service. In inventory management, providing real-time notifications when an item is available to buy might be the difference between making the sale or losing a customer.

Finally, real-time data synchronization is essential for many internal processes too.

updated in seconds to ensure

ple except for the fact that API

around the world. This means

ing introduces the likelihood of

rmance optimization.

techniques such as caching, payload limits, rate limiting, and others to optimize the API.

Stress testing your API with real-world scenarios can help you catch performance issues in the development phase, minimizing problems when it eventually goes live. This is why load testing tools are becoming so popular and varied, especially in the API-centric Kubernetes space.

3. Improved Security and Better Third-Party Options

As APIs are more widely used and distributed, their attack surface areas tend to grow. The complexity and variety of APIs used by many organizations only increase the security risk:

> "Simply put, it's impossible to secure what you don't know about. But all too often, organizations simply don't have any idea what's running on their networks. Companies added security tools in a one-off fashion… What's more, traditional tools were not built with the modern, decentralized enterprise in mind and often cause more problems than they solve."
> — **Sean Leach, Fastly**

issues compared to others, but
rds increased awareness and

its as baseline security
mplex DDoS attacks,

tect your API from common
icial intelligence to provide even

4. Widespread Adoption of Microservices

Microservices are now a core focus when building APIs. While once a buzzword in the tech industry, they're clearly here to stay. In an ideal world, microservices will help with scalability, flexibility, and security. Of course, they also introduce their own challenges for API developers.

For example, API gateways are an essential part of microservices. They will forward requests to the respective services based on routing rules you configure. Authentication and security checks are often done in the API gateway so that duplication is not required for each service. If not implemented correctly, this can introduce a single point of failure into your API, so it's not something to be adopted without care.

Microservices also need to communicate with each other internally. Historically, this communication has been done using REST APIs. Nowadays, microservice communication is trending toward gRPC and event-driven architectures, given their speed and reliability (as we'll discuss in the next section). Latency can be a major issue for microservices, so the transition away from REST makes sense, but these new networking protocols are less established and supported.

Technologies Powering These Trends

As API developers focus on real-time data synchronization, global performance, security, and the adoption of microservices, open source and proprietary tools are being

e most notable API development

lopment, you must create
ded in each call. Using GraphQL,
ust define the types and fields. In
s and get the data they need

back as JSON. In the REST model of fetching data, multiple round trips to the server are often needed to get all of the data an application requires — so it can be slow, especially on a mobile device."

— **Richard MacManus, The New Stack**

e. If they also want to get the
est. This saves development
ndpoints with different formats,
ne request. GraphQL is also
werful developer tools (like

gRPC is a modern, open source, high-performance Remote Procedure Call framework. It can be run on any environment and can be used to communicate across backend, frontend, and mobile devices.

gRPC's speed and cross-platform support are key strengths, particularly given the rising popularity of microservices. In many microservice architectures, a request needs to be made to multiple services to compile the response. Response times can get slow because of network calls, and gRPC helps to reduce this latency.

gRPC uses protocol buffers for binary serialization instead of JSON. This makes gRPC fast — even with a large payload of data. In fact, gRPC is roughly seven times faster than REST API calls, plus it has built-in HTTP/2 protocol support. Finally, instead of relying on individually defined API endpoints like REST, gRPC provides an auto-generated SDK that clients can use to make calls to the service.

REST: Tried and Tested, But What's New?

st, REST isn't going away
most common problems have
few years thanks to innovations

request could be processed at a
ultiplexing, where multiple
time. In HTTP/1, multiple
HTTP/2 can achieve parallelism

This is mostly used for time-sensitive data that you want to be reflected in the user interface as soon as possible (like notifications or alerts). Users don't have to refresh the browser just to see the latest updates.

The Web Push API can even deliver messages to users who are temporarily offline. While often used in conjunction with traditional asynchronous protocols (like webhooks or email notifications), widespread use of the Web Push API is opening up new avenues for real-time communication to API consumers.

Conclusion

The software industry evolves quickly, and every developer has to find ways to keep up. New technologies allow API developers to build performant, highly available, secure applications more easily. They also offer API consumers more options for integrating them.

More than ever, the big challenge is picking the right technology for the right use case. For example, a real-time chat API might be a great candidate for gRPC, while a social network API might lean on GraphQL. Similarly, a notifications API might use push notifications to alert users, while a simple two or three endpoint application might limit its offering to a more traditional REST API.

As the variety of APIs being offered grows, API technology will continue to evolve accordingly. The rise in video streaming, cloud native technology, and edge computing ard to predict.

Email Address

Subscribe

We don't sell or share your email. By continuing, you agree to our Terms of Use and Privacy Policy.

RELATED STORIES

DATA / DEVELOPMENT / MACHINE LEARNING / CONTRIBUTED

Machine Learning Still Struggles to Extract Meaning from Language

10:00am, by Walt Mayo

SEPTEMBER 01, 2021

solo.io

How to get Istio traceability with Grafana Tempo and Grafana Loki

SEPTEMBER 01, 2021

portworx by Pure Storage

Using Smart Storage Management with Pure1 for Portworx-enabled IBM Cloud Pak for Data on OpenShift

SEPTEMBER 01, 2021

SYNOPSYS®

New banking paradigm requires better security paradigm

SEPTEMBER 01, 2021

anynines

Welcome Wednesday: Ankita Sharma

SEPTEMBER 01, 2021

dynatrace

Dynatrace extends automatic and intelligent observability to applications in Azure Spring Cloud

SEPTEMBER 01, 2021

kasten by Veeam

CyberPeace Institute Replaces Open Source Backup Tool for Kasten K10

SEPTEMBER 01, 2021

Tricentis

Transforming testing: Is it worth it?

SEPTEMBER 01, 2021

styra

Meet our new Vice President of Product, Nils Swart!

SEPTEMBER 01, 2021

Check Point SOFTWARE TECHNOLOGIES LTD

Check Point Harmony: A secure solution for remote users

SEPTEMBER 01, 2021

RED HAT OPENSHIFT

The Advanced Cluster Security Operator Is Here. What You Need to Know and How to Get Started.

SEPTEMBER 01, 2021

GitLab

Google Summer of Code at GitLab – some intern highlights
AUGUST 31, 2021

TIGERA

Calico integration with WireGuard using kOps
AUGUST 31, 2021

STORJ

Using Storj DCS with GitHub Actions
AUGUST 31, 2021

snowflake

Data Science: The Future of Corporate Finance
AUGUST 31, 2021

CONFLUENT

Announcing Elastic Data Streams Support for Confluent's Elasticsearch Sink Connector
AUGUST 31, 2021

armory

Git Pull Support in Spinnaker

harness

Tutorial: [Monitoring and Observability] How to Send Kubernetes Delegate Logs to Splunk
AUGUST 31, 2021

TRIGGERMESH
CLOUD NATIVE INTEGRATION PLATFORM

Serverless COBOL in Knative
AUGUST 31, 2021

PRISMA
BY PALO ALTO NETWORKS

Introducing SASE Converge 2021: The Summit for What's Next in SASE
AUGUST 31, 2021

Vates
Solutions Open Source

Xen Orchestra 5.62
AUGUST 31, 2021

bridgecrew

Multi-Cloud Drift Detection powered by Yor: Automated infrastructure drift monitoring and fixes
AUGUST 31, 2021

NS1.

Life as a Frontend Engineer
AUGUST 31, 2021

redis

Nearshore Delivery Solutions
AUGUST 30, 2021

MIRANTIS

Reduce Network Latency: How to Beat the Internet at its Own Game
AUGUST 30, 2021

SENTRY

Guest Post: Performance Monitoring in GraphQL
AUGUST 30, 2021

torq

Automating Identity Lifecycle Management
AUGUST 30, 2021

MINIO

Continuous Data Protection with MinIO Versioning and Rewind
AUGUST 30, 2021

AUGUST 27, 2021

circleci

Deploying your Gradle Build Cache Node using GCP

AUGUST 27, 2021

WSO2

The Evolution of Cloud Native Engineering

AUGUST 26, 2021

JFrog

Scanning Dependencies in your sources using JFrog CLI and Xray

AUGUST 26, 2021

ASPEN MESH

Get App-focused Security from an Enterprise-class Service Mesh | On-demand Webinar

AUGUST 26, 2021

SAP

ASYNC/Parallel ABAP in a OO way

AUGUST 26, 2021

NGINX

Why You Need an Enterprise-Grade Ingress Controller on OpenShift

AUGUST 25, 2021

Hewlett Packard Enterprise

Future-proof your data and analytics investment with HPE Ezmeral

AUGUST 16, 2021

honeycomb.io

Why Observability Requires a Distributed Column Store

AUGUST 12, 2021

ThousandEyes

Coffee Chat with Michael, Customer Engineering Leader>

AUGUST 12, 2021

shipa

GitOps Workshop

AUGUST 11, 2021

StepZen

Build a Mobile Login with React Native.

JULY 29, 2021

CLOUD FOUNDRY

CVE-2021-22098: Open redirect vulnerability in UAA server
JULY 27, 2021

CITRIX

App delivery for complex environments made simple, flexible, cost effective
JULY 14, 2021

mongoDB.

We Replaced an SSD with Storage Class Memory. Here is What We Learned.
AUGUST 27, 2020

vmware

Join us at our new blog home
JULY 02, 2020

IBM Cloud

CubeSat Space Mission "ENDURANCE": Democratizing Access to Space
DECEMBER 31, 1969

Development
Machine Learning
Security

OPERATIONS

CI/CD
Culture
DevOps
Kubernetes
Monitoring
Service Mesh
Tools

THE NEW STACK

Ebooks
Podcasts
Events
Newsletter
About / Contact
Sponsors
Sponsorship