EP33: Twitter Architecture 2022 vs 2012

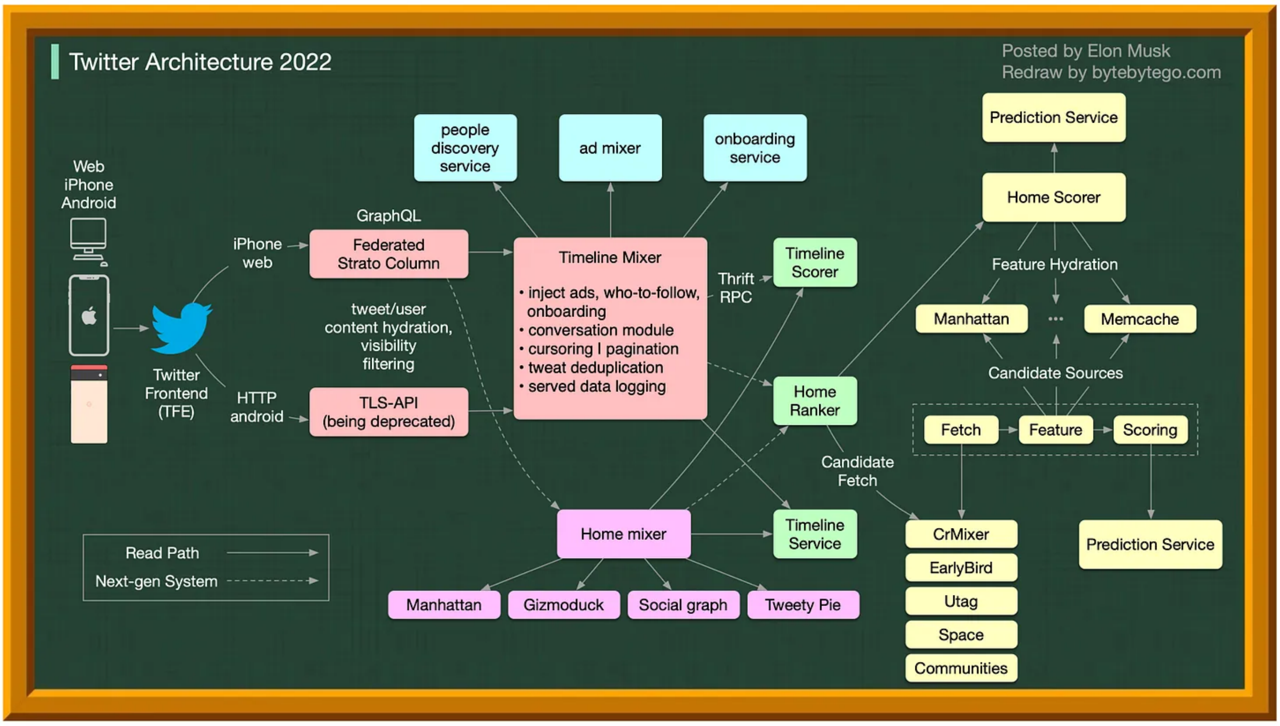
ALEX XU
NOV 19, 2022

187 2 Share

This week’s system design refresher:

- Twitter Architecture 2022
- What is Single Sign-On? (Youtube video)
- TCP/IP Encapsulation
- Code First v.s. API First
- The Ping command
- API architectural styles
- Join ByteByteGo Talent Collective

Twitter Architecture 2022

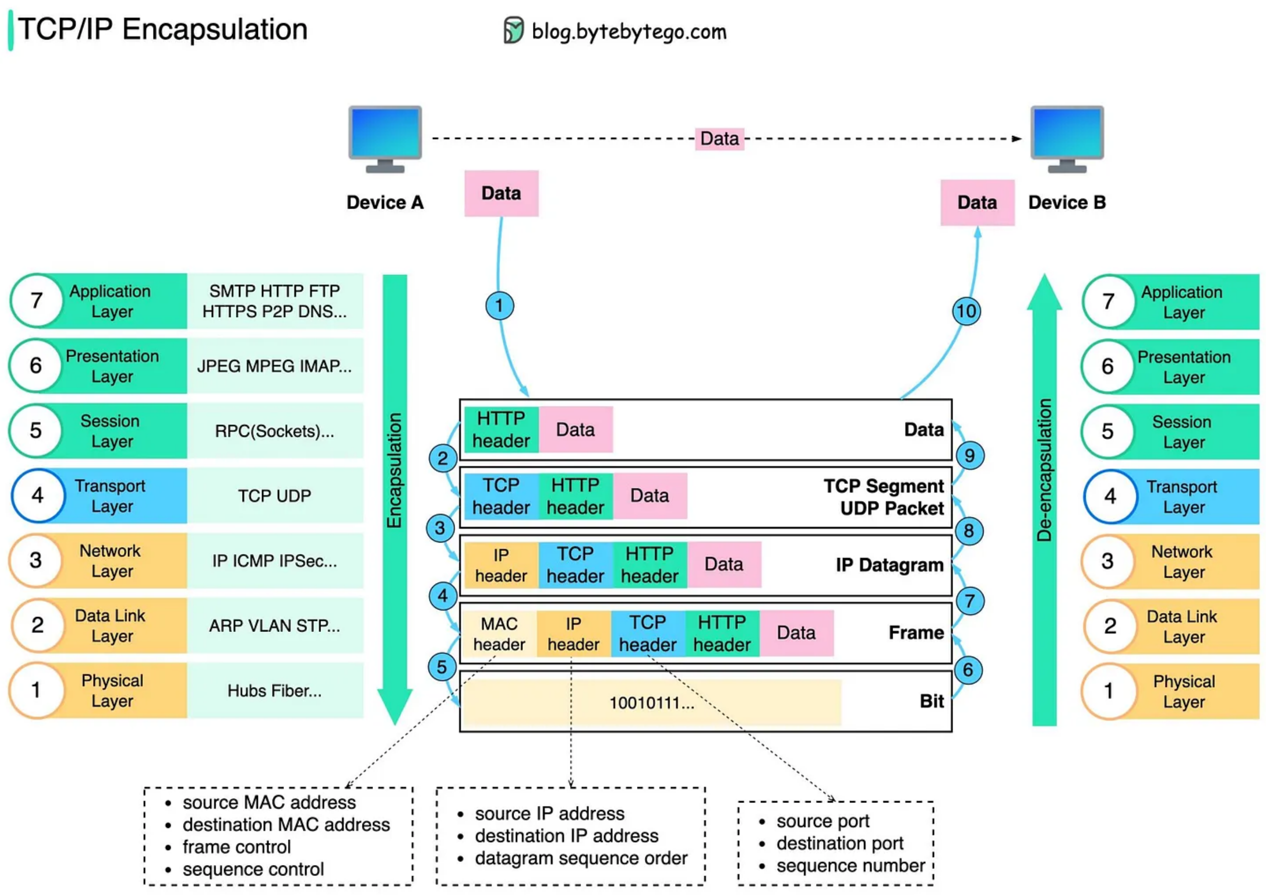


What is SSO, or Single Sign-On?

What Is Single Sign-on (SSO)? How It Works

How is data sent over the network? Why do we need so many layers in the OSI model?

The diagram below shows how data is encapsulated and de-encapsulated when transmitting over the network.



Step 1: When Device A sends data to Device B over the network via the HTTP protocol, it is first added an HTTP header at the application layer.

Step 2: Then a TCP or a UDP header is added to the data. It is encapsulated into TCP segments at the transport layer. The header contains the source port, destination port, and sequence number.

Step 3: The segments are then encapsulated with an IP header at the network layer. The IP header contains the source/destination IP addresses.

Step 4: The IP datagram is added a MAC header at the data link layer, with source/destination MAC addresses.

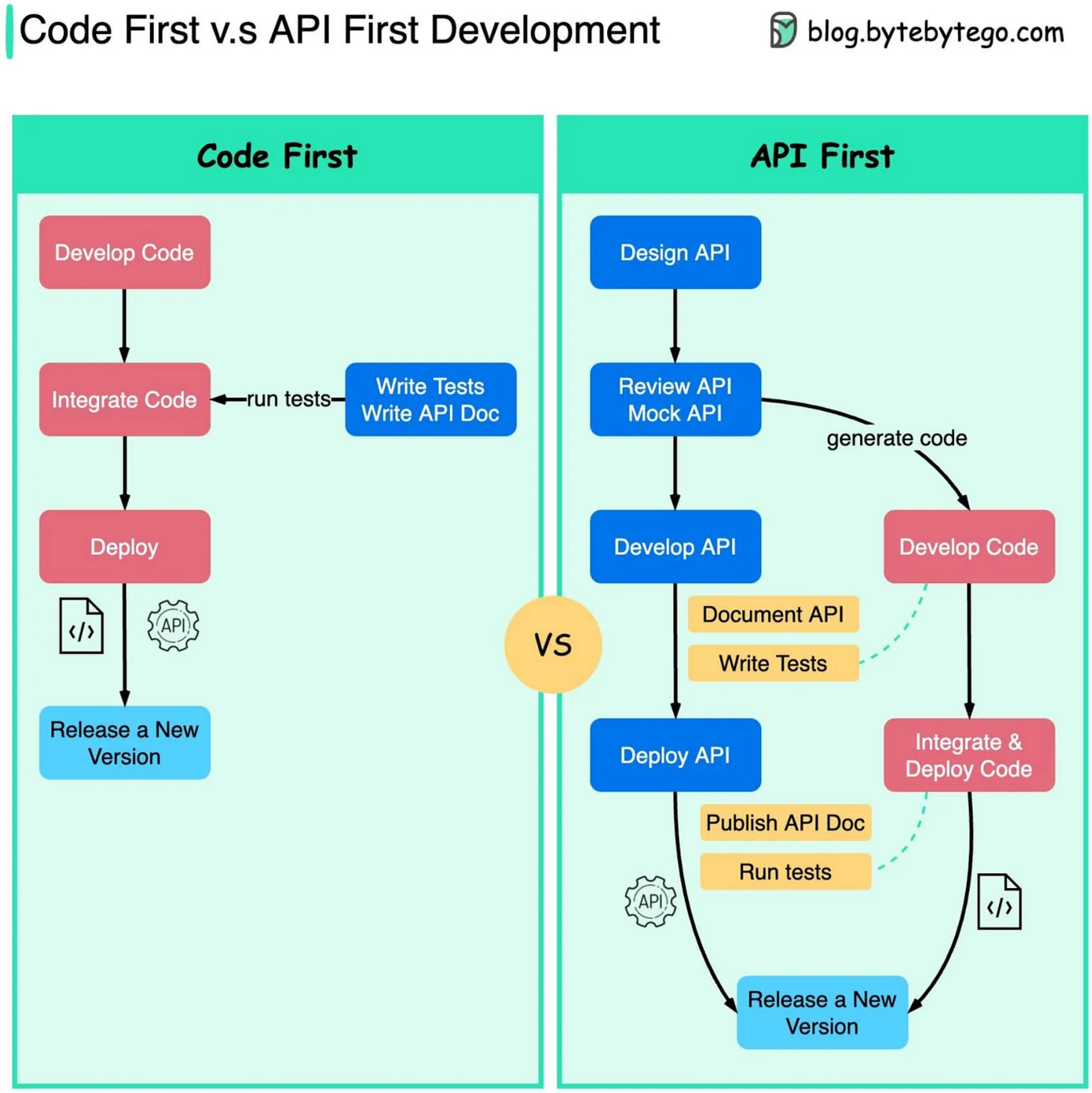
Step 5: The encapsulated frames are sent to the physical layer and sent over the network in binary bits.

Steps 6-10: When Device B receives the bits from the network, it performs the de-encapsulation process, which is a reverse processing of the encapsulation process. The headers are removed layer by layer, and eventually, Device B can read the data.

We need layers in the network model because each layer focuses on its own responsibilities. Each layer can rely on the headers for processing instructions and does not need to know the meaning of the data from the last layer.

Code First v.s. API First - A change of software development philosophy

The diagram below shows the differences between code-first development and API-first development. Why do we want to consider API first design?



- **Microservices increase system complexity.**
  We have separate services to serve different functions of the system. While this kind of architecture facilitates decoupling and segregation of duty, we need to

handle the various communications among services.
It is better to think through the system's complexity before writing the code and carefully defining the boundaries of the services.

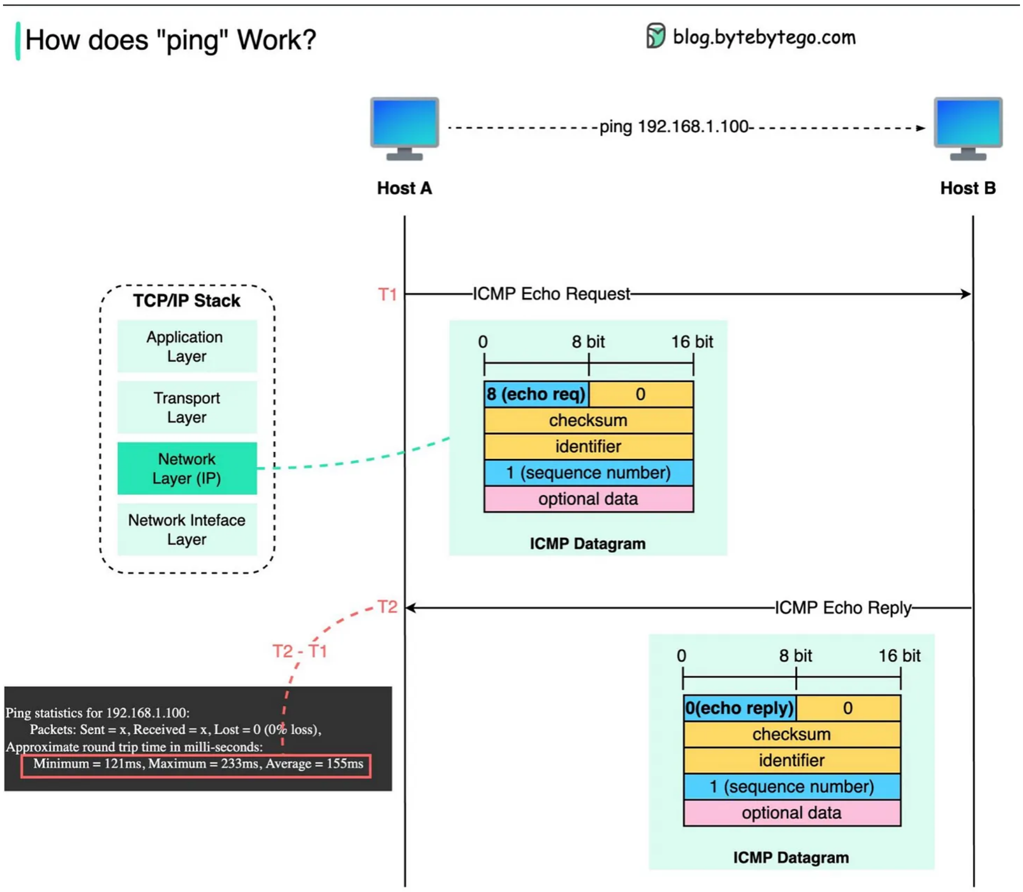
- **Separate functional teams need to speak the same language.**
  The dedicated functional teams are only responsible for their own components and services. It is recommended that the organization speak the same language via API design.
  We can mock requests and responses to validate the API design before writing code.

- **Improve software quality and developer productivity**
  Since we have ironed out most of the uncertainties when the project starts, the overall development process is smoother, and the software quality is greatly improved.
  Developers are happy about the process as well because they can focus on functional development instead of negotiating sudden changes.
  The possibility of having surprises toward the end of the project lifecycle is reduced.

Because we have designed the API first, the tests can be designed while the code is being developed. In a way, we also have TDD (Test Driven Design) when using API first development.

What happens when we “ping” a server?

The diagram below shows how the “ping” command works.



```
Ping statistics for 192.168.1.100:
    Packets: Sent = x, Received = x, Lost = 0 (0% loss),
Approximate round trip time in milli-seconds:
    Minimum = 121ms, Maximum = 233ms, Average = 155ms
```

The “ping” command runs on ICMP (Internet Control Message Protocol), which is a network layer protocol.

There are 6 common types of messages in ICMP. For the ping command, we mainly use “echo request” and “echo reply”.

Host A sends an ICMP echo request (type = 8) with a sequence number 1. The request is encapsulated with an IP header to specify the source and destination IP addresses.

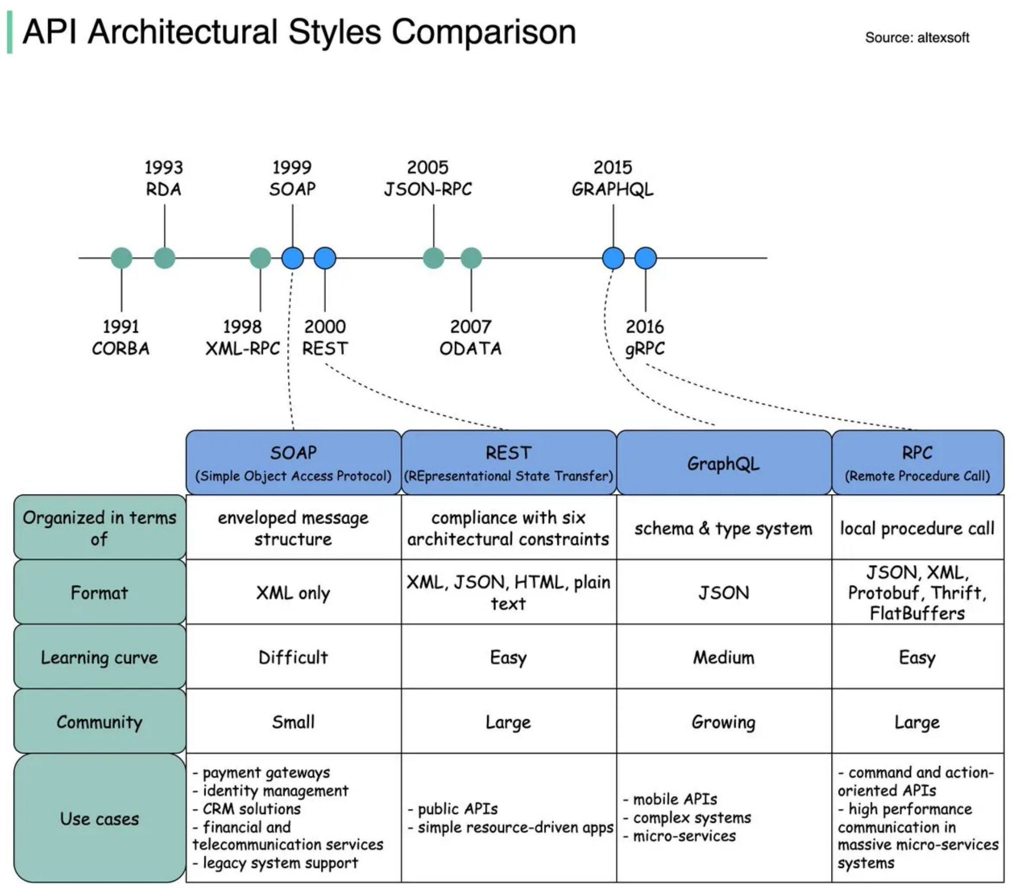
When host B receives the data, it sends back an ICMP echo reply (type = 0) with sequence number 1 to host A.

When host A receives the echo reply, it correlates the request and reply with the sequence number, and uses T1 and T2 to calculate the round trip time. That’s how we see the ping statistics.

API architectural styles

Since RPC has become a hot topic, let's briefly review its history.

The diagram below illustrates the API timeline and API style comparison.



API Architectural Styles Comparison

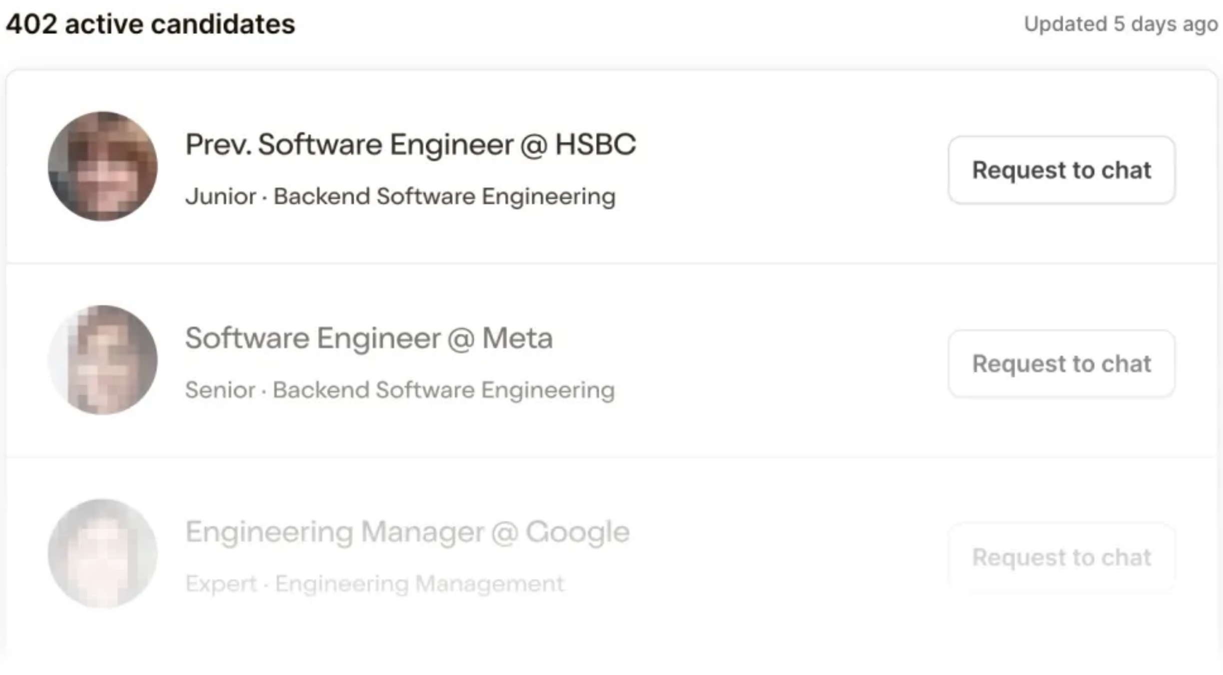
Source: altexsoft

	SOAP (Simple Object Access Protocol)	REST (REpresentational State Transfer)	GraphQL	RPC (Remote Procedure Call)
Organized in terms of	enveloped message structure	compliance with six architectural constraints	schema & type system	local procedure call
Format	XML only	XML, JSON, HTML, plain text	JSON	JSON, XML, Protobuf, Thrift, FlatBuffers
Learning curve	Difficult	Easy	Medium	Easy
Community	Small	Large	Growing	Large
Use cases	- payment gateways - identity management - CRM solutions - financial and telecommunication services - legacy system support	- public APIs - simple resource-driven apps	- mobile APIs - complex systems - micro-services	- command and action-oriented APIs - high performance communication in massive micro-services systems

Over time, different API styles are released. Each of them has its own patterns of standardizing data exchange.

Join ByteByteGo Talent Collective

If you're hiring, join ByteByteGo Collective to start getting weekly drops of world-class hand-curated engineers who are open to new opportunities.



402 active candidates

Updated 5 days ago

Prev. Software Engineer @ HSBC
Junior · Backend Software Engineering

Software Engineer @ Meta
Senior · Backend Software Engineering

Engineering Manager @ Google
Expert · Engineering Management

HIRE CANDIDATES FROM TOP COMPANIES

If you're looking for a new job, join to get personalized opportunities from hand-selected companies. You can join publicly or anonymously, and leave anytime.

Featured job openings

Openedges: Chief Architect (San Jose, Austin, Remote)

Thanks for reading ByteByteGo Newsletter! Subscribe for free to receive new posts and support my work.

187 Likes

2 Comments

Write a comment...

Pratistha Srivastav Writes Pratistha's Newsletter Nov 20, 2022

Loved the SSO explanation and the OSI model information. Thank you! 🙌

♡ LIKE (1) REPLY SHARE ...

Francis Ibok Nov 26, 2022

This is an overview of architecture system design of Twitter, which describes the holistic view of the components interacts.

♡ LIKE REPLY SHARE ...