EP 39: Accounting 101 in Payment Systems

ALEX XU

DEC 31, 2022

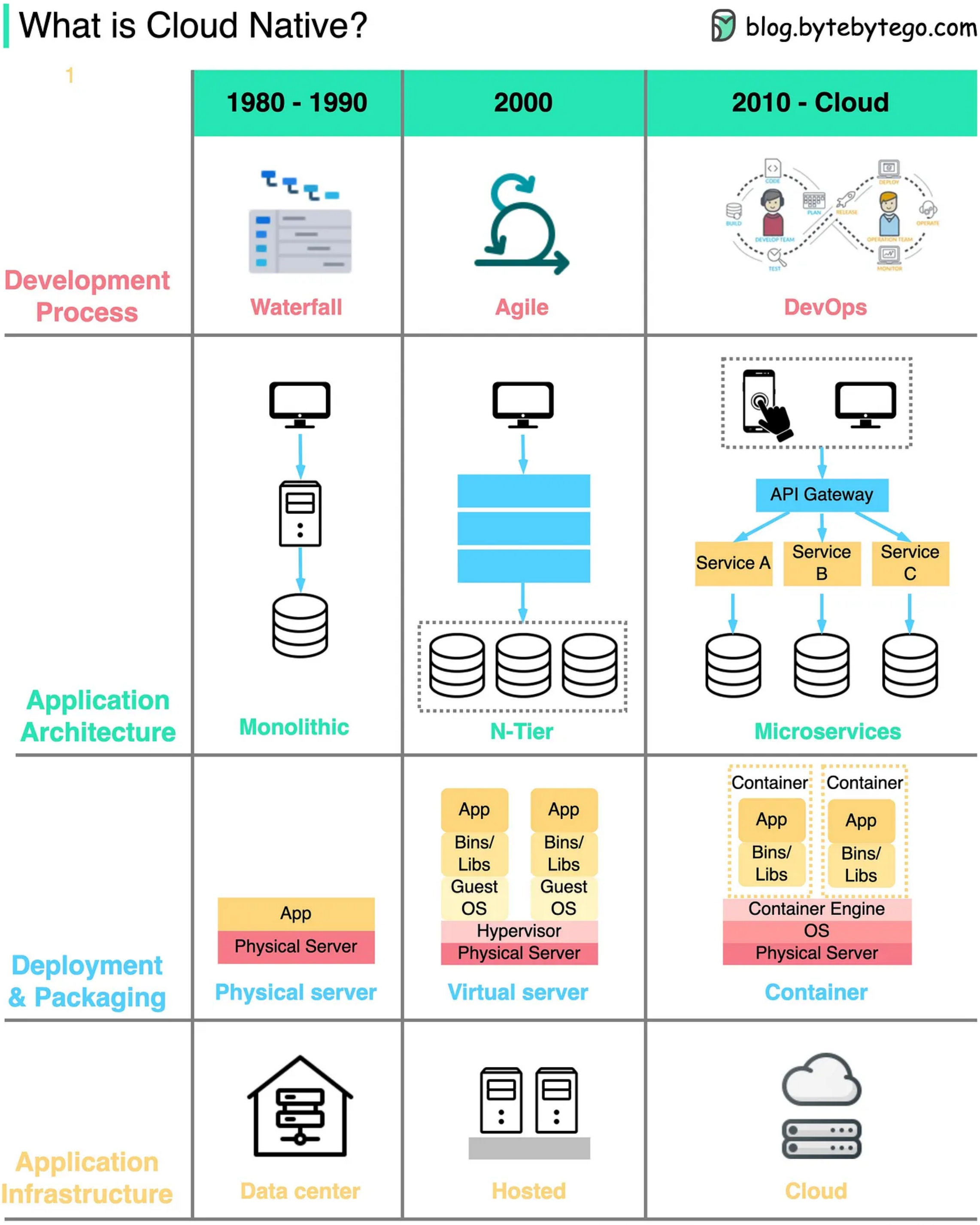
♡ 160 5 Share •••

This week's system design refresher:

- Cloud Native
- Accounting 101 in Payments
- Evolution of Uber's API Layer
- Short/Long Polling, SSE, Websocket
- 17 Equations That Changed the World

What is cloud native?

Below is a diagram showing the evolution of architecture and processes since the 1980s.



What is Cloud Native?

blog.bytebytego.com

	1980 - 1990	2000	2010 - Cloud
Development Process	Waterfall	Agile	DevOps
Application Architecture	Monolithic	N-Tier	Microservices
Deployment & Packaging	Physical server	Virtual server	Container
Application Infrastructure	Data center	Hosted	Cloud

Reference: https://www.oracle.com/cloud/cloud-native/what-is-cloud-native/

Organizations can build and run scalable applications on public, private, and hybrid clouds using cloud native technologies.

This means the applications are designed to leverage cloud features, so they are resilient to load and easy to scale.

Cloud native includes 4 aspects:

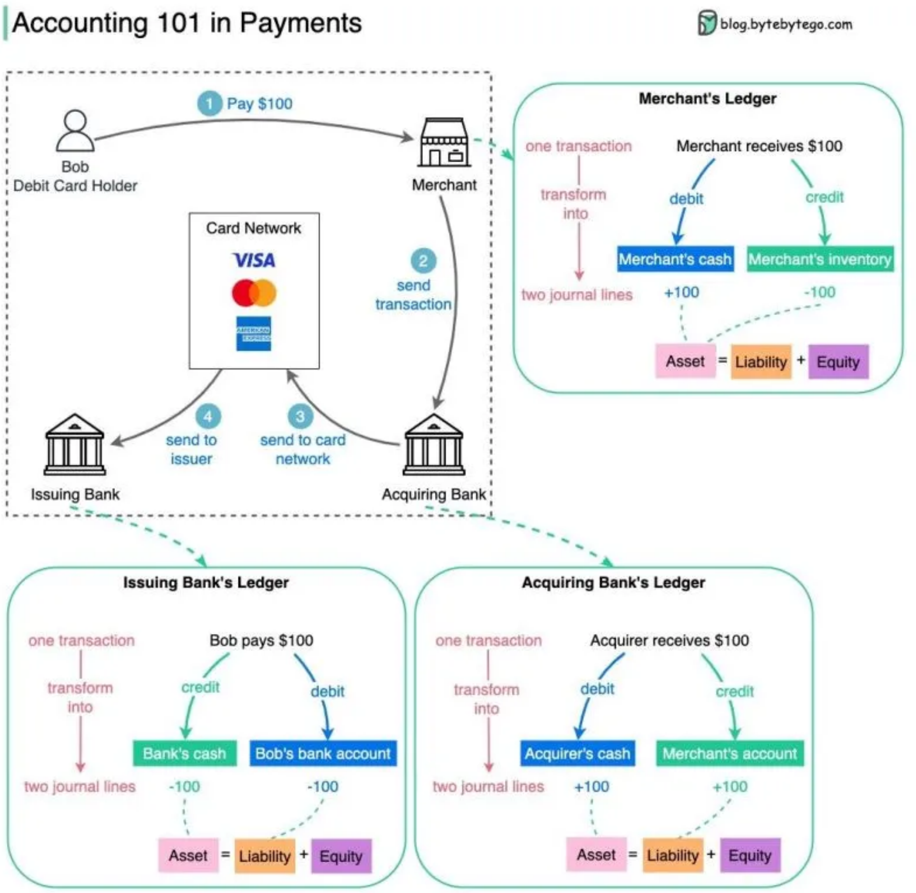
1. **Development process**
   This has progressed from waterfall to agile to DevOps.
2. **Application Architecture**
   The architecture has gone from monolithic to microservices. Each service is designed to be small, and adaptive to the limited resources in cloud containers.
3. **Deployment & packaging**
   The applications used to be deployed on physical servers. Then around 2000, the applications that were not sensitive to latency were usually deployed on virtual servers. With cloud native applications, they are packaged into docker images and deployed in containers.
4. **Application infrastructure**
   The applications are massively deployed on cloud infrastructure instead of self-hosted servers.

👉 Over to you: what comes into your mind when people talk about “cloud native”?

Accounting 101 in Payments

Why is a credit card called a “**credit**” card?
Why is a debit card called a “**debit**” card?

An example of a debit card payment is shown in the diagram below.



- Each transaction in the business system is transformed into at least two journal lines in the ledger system. This is called **double-entry** accounting, where every transaction must have a **source** account and a **target** account.
- Each journal line is booked to an account.
- Each account belongs to one of the three components in the balance sheet:

> 💡 Asset = Liability + Equity

Let’s look at the issuing bank’s ledger as an example:

Bob pays $100 to the merchant with a debit card. We have two accounts involved in this transaction:

1. Journal line 1 - From the issuing bank’s point of view, Bob’s bank account is a **liability** (because the bank owes Bob money). Bob’s bank account is deducted $100.
   This is a **debit** record.

2. Journal line 2 - Bank’s cash is an **asset** and the bank’s cash is deducted by $100. This is a **credit** record.

The balance sheet equation still balances with the two journal lines recorded in the ledger.

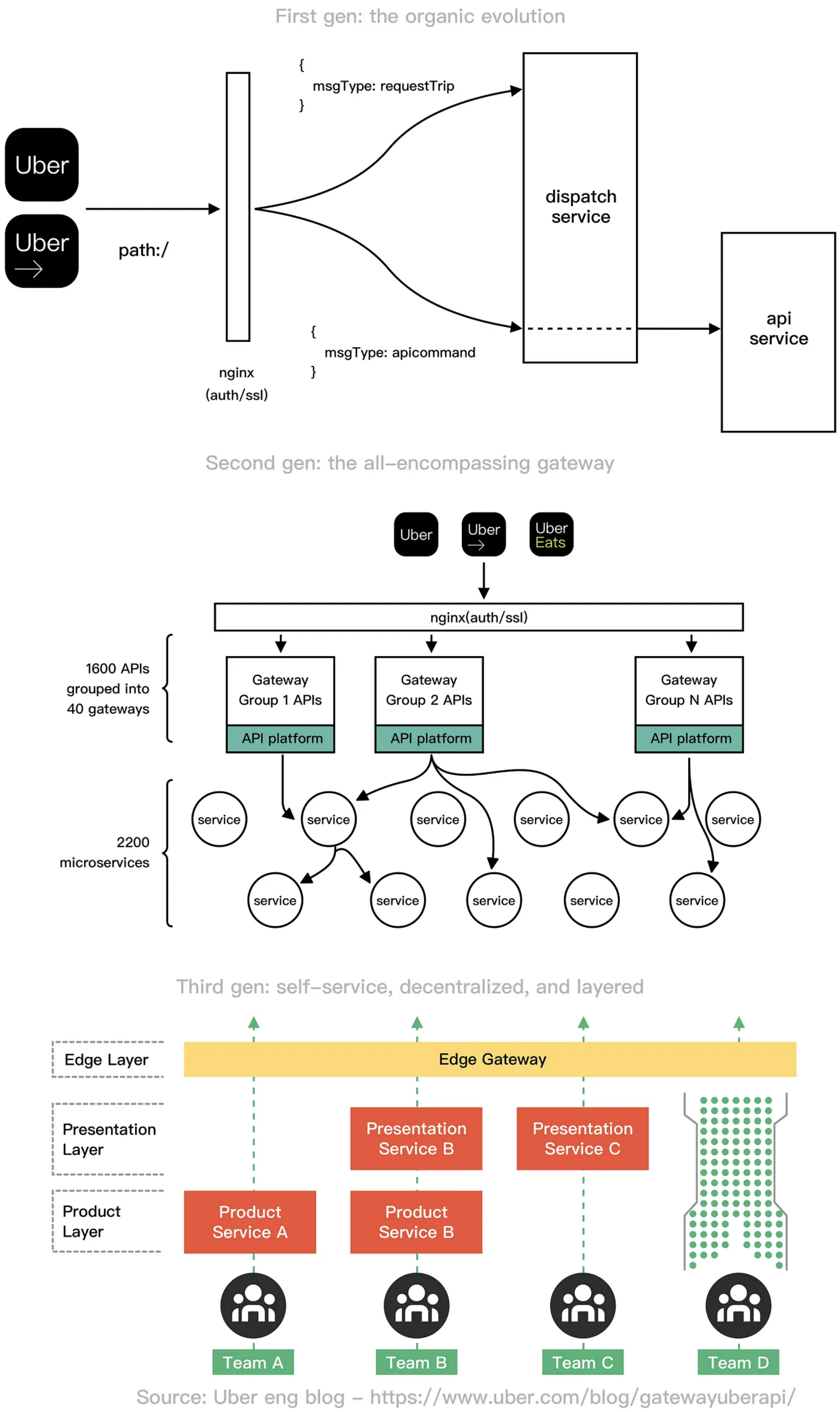
Bob’s card is called a “debit” card because it is a **debit record** when paying with a debit card.

👉 Why is this important?
This is how a ledger system is designed, only a real ledger is more complicated. Applying these strict accounting rules makes reconciliation much easier!

Evolution of Uber’s API layer

Uber’s API gateway went through 3 main stages.

Evolution of Uber's API Layer



Source: Uber eng blog – https://www.uber.com/blog/gatewayuberapi/

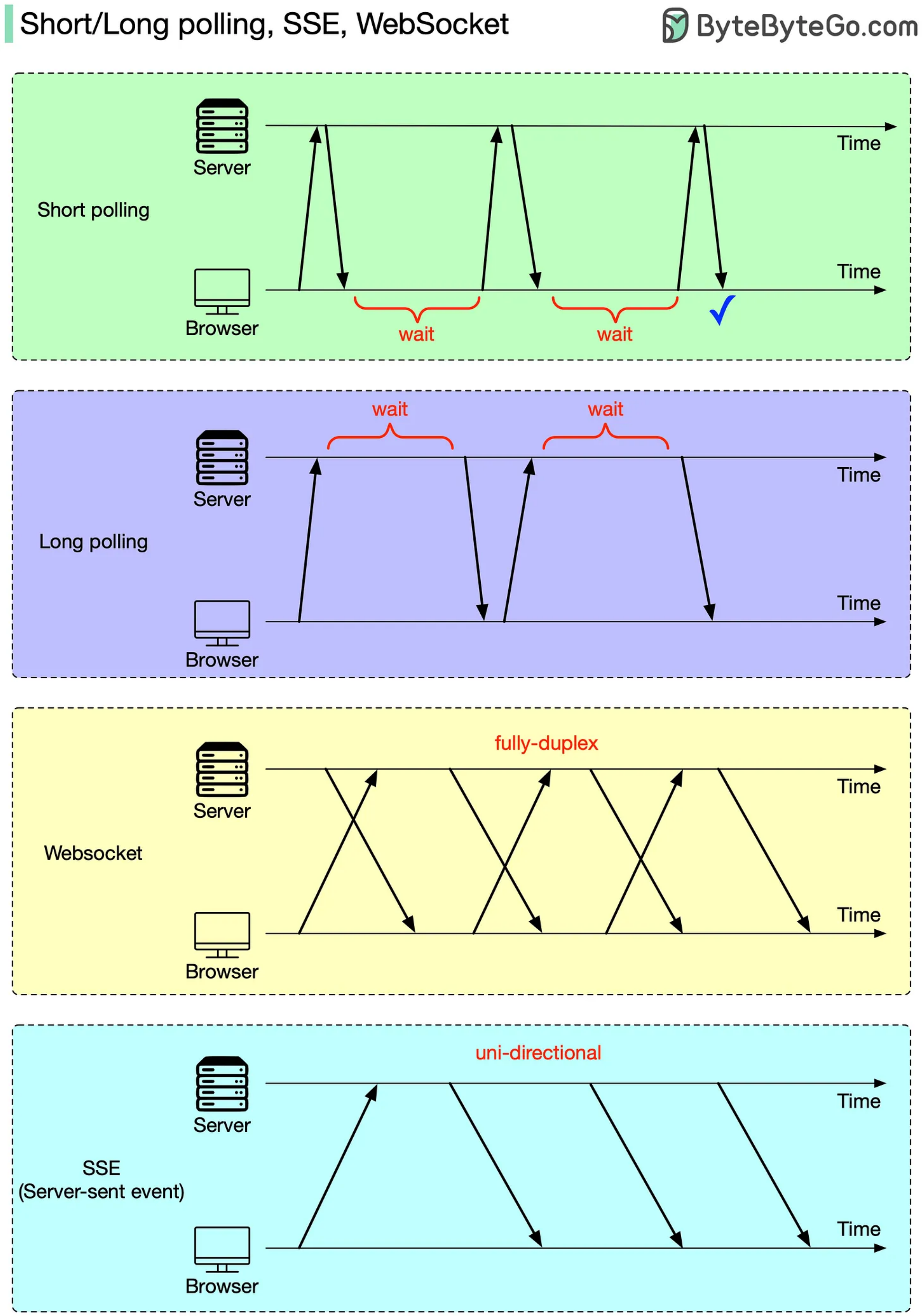
First gen: the organic evolution. Uber's architecture in 2014 would have two key services: dispatch and API. A dispatch service connects a rider with a driver, while an API service stores the long-term data of users and trips.

Second gen: the all-encompassing gateway. Uber adopted a microservice architecture very early on. By 2019, Uber's products were powered by 2,200+ microservices as a result of this architectural decision.

Third gen: self-service, decentralized, and layered. As of early 2018, Uber had completely new business lines and numerous new applications. Freight, ATG, Elevate, groceries, and more are among the growing business lines. With a new set of goals comes the third generation.

Short/long polling, SSE, WebSocket

An HTTP server cannot automatically initiate a connection to a browser. As a result, the web browser is the initiator. What should we do next to get real-time updates from the HTTP server?



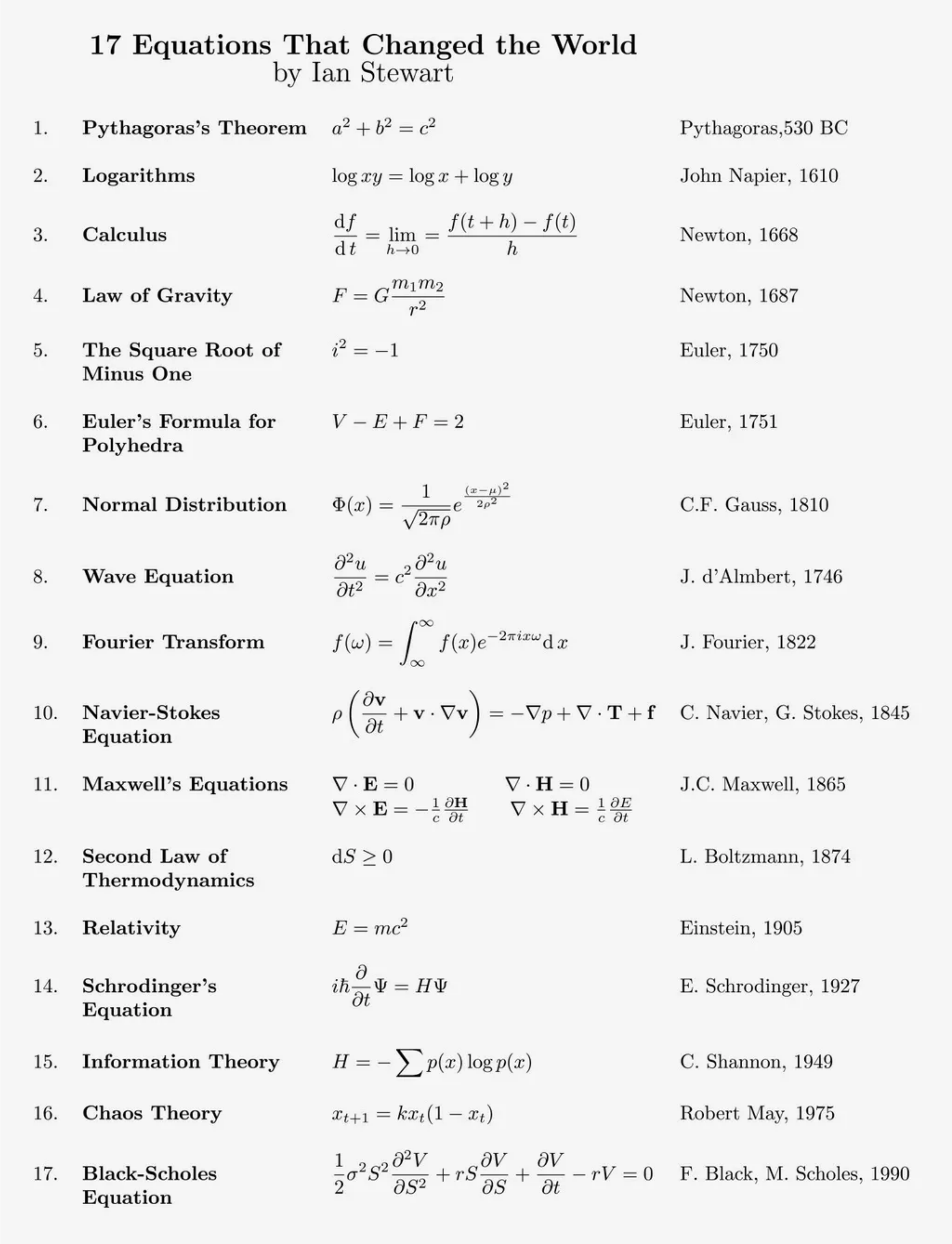
Both the web browser and the HTTP server could be responsible for this task.

- Web browsers do the heavy lifting: short polling or long polling. With short polling, the browser will retry until it gets the latest data. With long polling, the HTTP server doesn't return results until new data has arrived.
- HTTP server and web browser cooperate: WebSocket or SSE (server-sent event). In both cases, the HTTP server could directly send the latest data to the browser after the connection is established. The difference is that SSE is uni-directional, so the browser cannot send a new request to the server, while WebSocket is fully-duplex, so the browser can keep sending new requests.

👉 Over to you: of the four solutions (long polling, short polling, SSE, WebSocket), which ones are commonly used, and for what use cases?

17 Equations That Changed the World

What else should be included?



17 Equations That Changed the World
by Ian Stewart

1.	**Pythagoras's Theorem**	$a^2 + b^2 = c^2$	Pythagoras,530 BC
2.	**Logarithms**	$\log xy = \log x + \log y$	John Napier, 1610
3.	**Calculus**	$\frac{\mathrm{d}f}{\mathrm{d}t} = \lim_{h\to 0} = \frac{f(t+h)-f(t)}{h}$	Newton, 1668
4.	**Law of Gravity**	$F = G\frac{m_1 m_2}{r^2}$	Newton, 1687
5.	**The Square Root of Minus One**	$i^2 = -1$	Euler, 1750
6.	**Euler's Formula for Polyhedra**	$V - E + F = 2$	Euler, 1751
7.	**Normal Distribution**	$\Phi(x) = \frac{1}{\sqrt{2\pi\rho}}e^{\frac{(x-\mu)^2}{2\rho^2}}$	C.F. Gauss, 1810
8.	**Wave Equation**	$\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2}$	J. d'Almbert, 1746
9.	**Fourier Transform**	$f(\omega) = \int_{\infty}^{\infty} f(x)e^{-2\pi i x\omega}\,\mathrm{d}x$	J. Fourier, 1822
10.	**Navier-Stokes Equation**	$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \nabla\cdot\mathbf{T} + \mathbf{f}$	C. Navier, G. Stokes, 1845
11.	**Maxwell's Equations**	$\nabla\cdot\mathbf{E} = 0$ $\quad \nabla\cdot\mathbf{H} = 0$ $\nabla\times\mathbf{E} = -\frac{1}{c}\frac{\partial\mathbf{H}}{\partial t}$ $\quad \nabla\times\mathbf{H} = \frac{1}{c}\frac{\partial E}{\partial t}$	J.C. Maxwell, 1865
12.	**Second Law of Thermodynamics**	$\mathrm{d}S \geq 0$	L. Boltzmann, 1874
13.	**Relativity**	$E = mc^2$	Einstein, 1905
14.	**Schrodinger's Equation**	$i\hbar\frac{\partial}{\partial t}\Psi = H\Psi$	E. Schrodinger, 1927
15.	**Information Theory**	$H = -\sum p(x)\log p(x)$	C. Shannon, 1949
16.	**Chaos Theory**	$x_{t+1} = kx_t(1-x_t)$	Robert May, 1975
17.	**Black-Scholes Equation**	$\frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} - rV = 0$	F. Black, M. Scholes, 1990

Featured job openings

HEIR: Senior Software Engineer, Full Stack (United States)

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

160 Likes

5 Comments

Write a comment...

Jono Stiansen Writes Everyday Leader Dec 31, 2022 ♥ Liked by Alex Xu

Thank you for the great content, it's really useful even for a seasoned engineer.

♡ LIKE (3) REPLY SHARE ...

Agastia Jan 1 ♥ Liked by Alex Xu

Thank you!

♡ LIKE (2) REPLY SHARE ...

3 more comments...