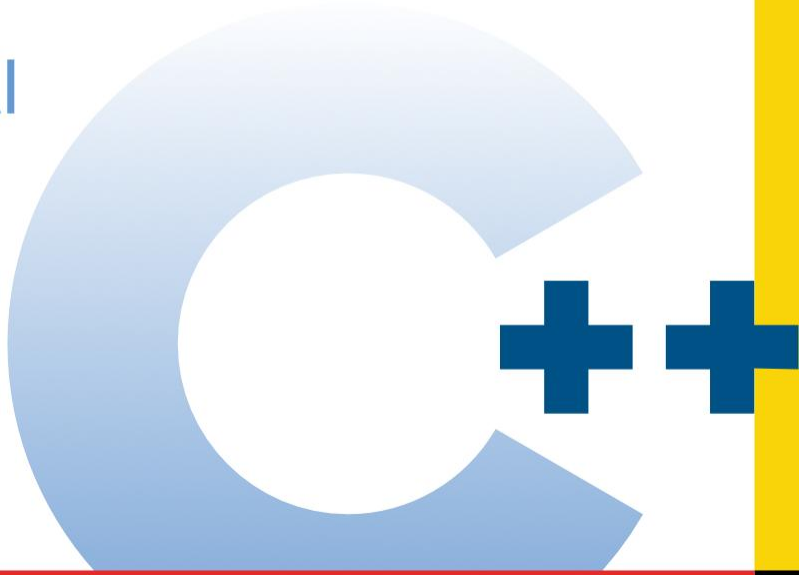
BeC++ SYMPOSIUM

Herb Sutter

C++: Growing in a world of competition, safety, and AI



BECPP SYMPOSIUM 2026 | March 30th | Kortrijk

C++ in 2026

C++ growth, fueled by our insatiable hunger for compute

C++ perspective on structural global changes now in progress

Power	Record-shattering 2026 CapEx
Security	Toto, we're not in 2021 anymore
AI	Software engineering cost impact (and non-impact)

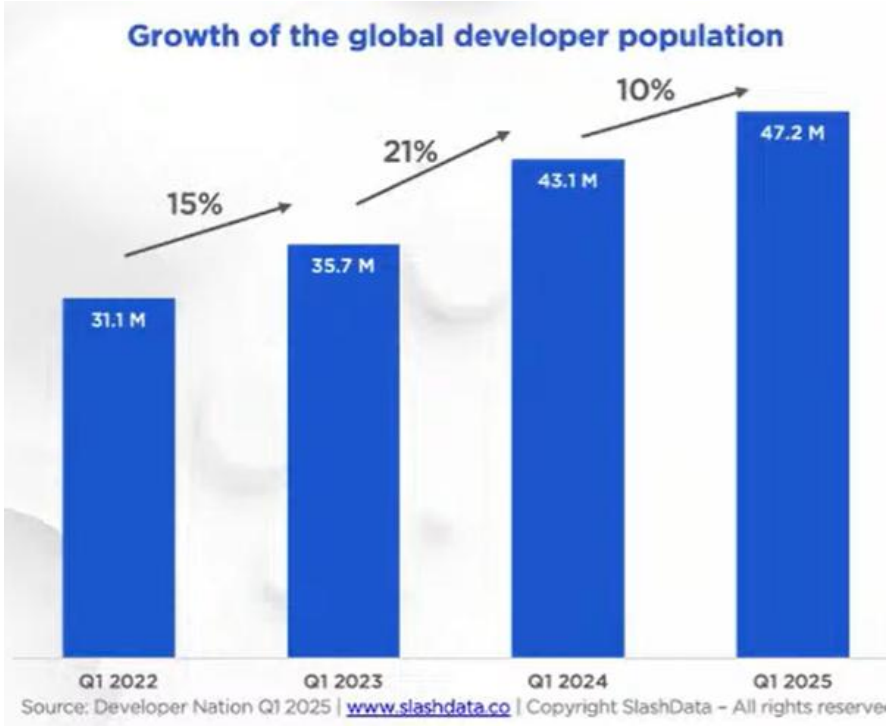
Programming is a hot market (SlashData, 2025)

Human programmers are in long-term high-growth demand

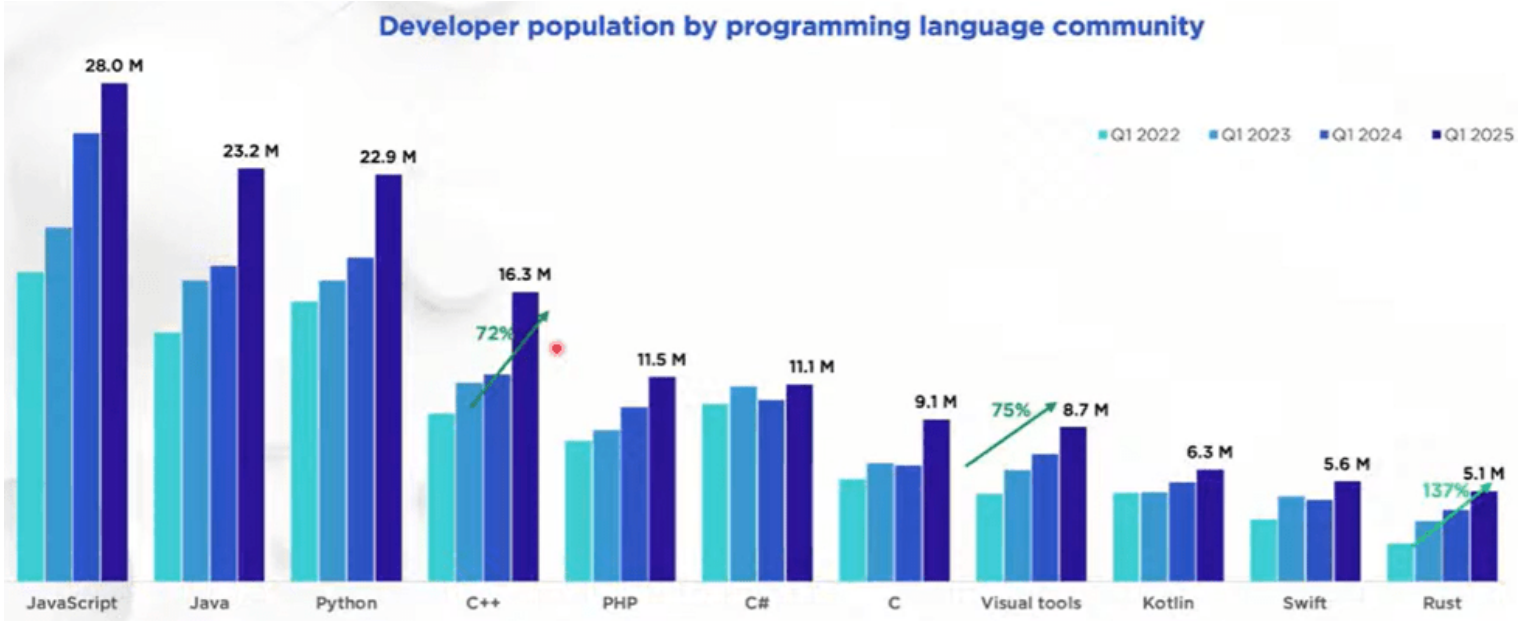
Other sources have consistent numbers

- IDC: forecasts >57M devs by 2028
- JetBrains: similar numbers counting only professional devs (not students or hobbyists)

AI is not changing this;
more on that later



Programming is a hot market (SlashData, 2025)



There are more C++ developers than the #1 language had 4 years ago

C++ (and Python and Java each) added as many devs in 1 year than total Rust devs

Why?

Q: Why have **C++ and Rust** been the fastest-growing major programming languages from 2022 to 2025?

A: Throughout the history of computing, **our demand for solving ever-larger computing problems consistently outstrips our ability to build greater computing capacity**, with no end in sight

$\Rightarrow$ long-term demand for “most performance from available hardware”

C++ in 2026

C++ growth, fueled by our insatiable hunger for compute

C++ perspective on structural global changes now in progress

Power	Record-shattering 2026 CapEx
Security	Toto, we're not in 2021 anymore
AI	Software engineering cost impact (and non-impact)

#2 compute constraint: Chips (CPUs, GPUs, RAM)

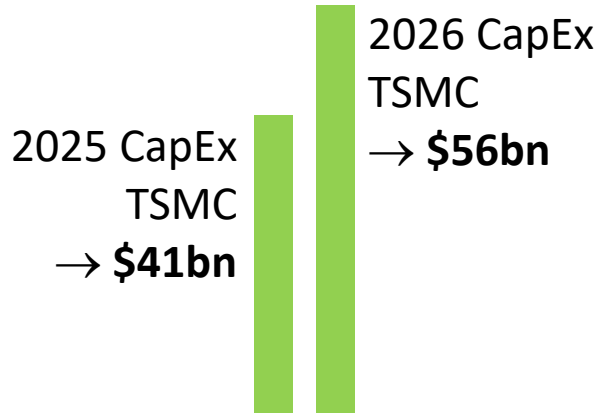
Leading-edge chip manufacturers can sell every part they can make

NVIDIA Now the world's most valuable company & TSMC top customer

TSMC Now the world's greatest single point of failure (and "brake")

C.C. Wei, TSMC Chairman & CEO (TSMC earnings call, Jan 15, 2026):

I spent a lot of time in the last three-four months talking to my customers and then customers' customers. … ***Can the semiconducting industry [keep strong growth]*** *for three four five [more] years … ? I will tell you the truth – I do not know. But I look at the AI,* ***it looks like, it is going to be, like, endless. I mean, that's for many years to come.***

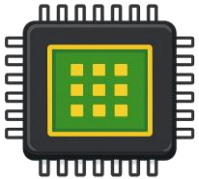


But…

… did you notice the OpenAI deal announcements pattern?

2023 - 2024:

expressed in terms of access to **NVIDIA GPUs**



mid/late 2025:

all expressed in terms of access to **gigawatts**

‖ datacenter capacities primarily measured in **GW**

#1 compute constraint: Power

Amy Hood, Microsoft CFO (MSFT earnings call, October 29, 2025):

*"[Azure] were short the … **power** … Those are very long-lived assets… 15 to 20 years"*

Andy Jassy, Amazon CEO (AMZN earnings call, October 30, 2025):

*"[AWS added] more than 3.8 GW of **power** in the past 12 months … now double the power capacity that AWS was in 2022, and we're on track to double again by 2027."*

Jensen Huang, NVIDIA CEO (NVDA earnings call, November 19, 2025):

*"1 GW data centers [means] 1 GW of **power**. … That 1 GW translates directly. Your **performance per watt** translates directly, absolutely directly, to your revenues."*

Rene Haas, ARM CEO (Stratechery interview, March 26, 2026):

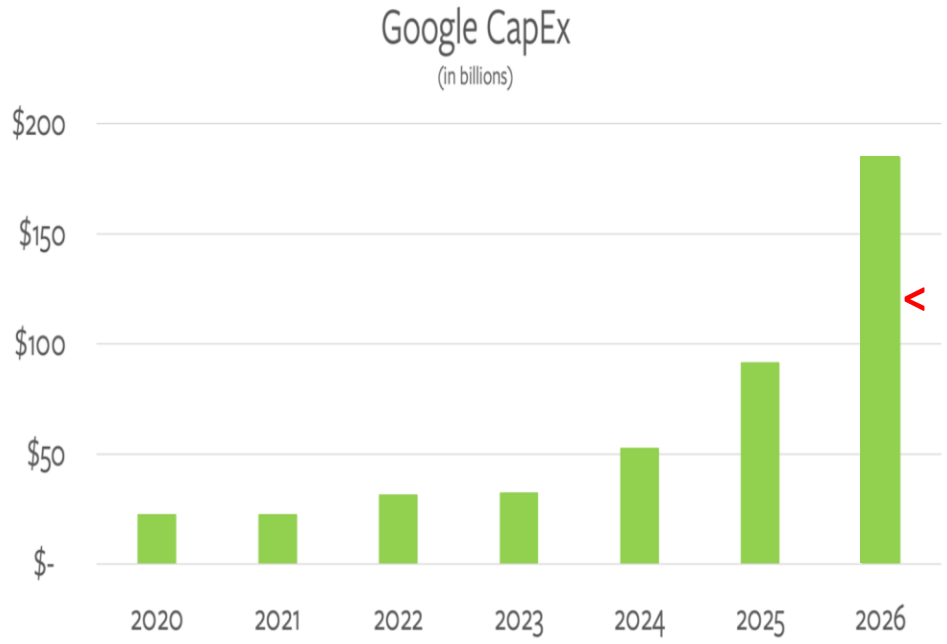
"Performance-per-watt, period."

News flashes: February 2026

2026 CapEx projection (GOOGL earnings call, February 4, 2026)

Ben Thompson, Stratechery:
*"The headline here is the massive CapEx guide, the top end of which is **a stunning $65.5 billion more than Wall Street expected**; for context that $65.5 billion "miss" is $13 billion more than Google spent on CapEx in 2024"*

And **more than…** 2026 CapEx TSMC → **$56bn**

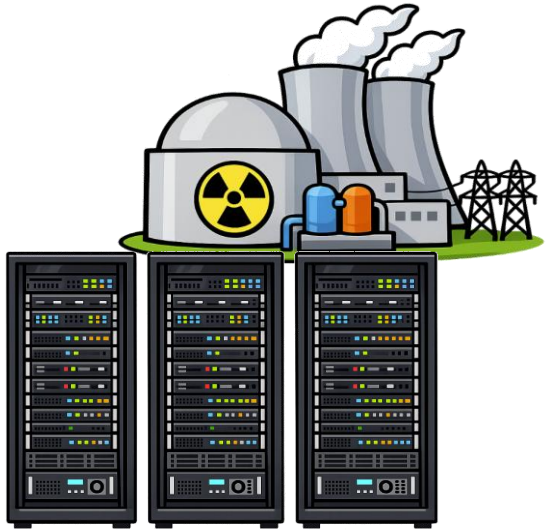


News flashes: February 2026

2026 CapEx projection (GOOGL earnings call, February 4, 2026)

2026 CapEx
GOOGL
→ **$185bn**

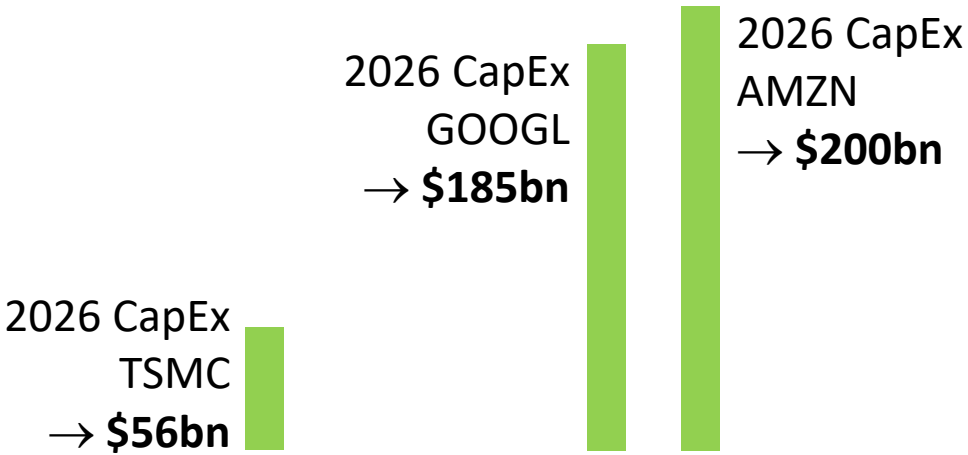
2026 CapEx
TSMC
→ **$56bn**

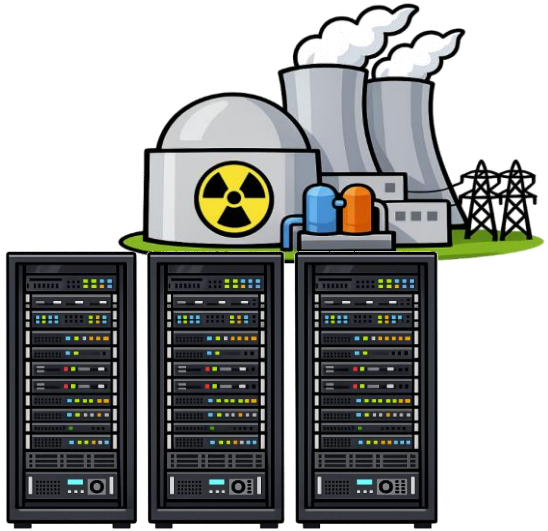


News flashes: February 2026

2026 CapEx projection (GOOGL earnings call, February 4, 2026)

2026 CapEx projection (AMZN earnings call, February 5, 2026)





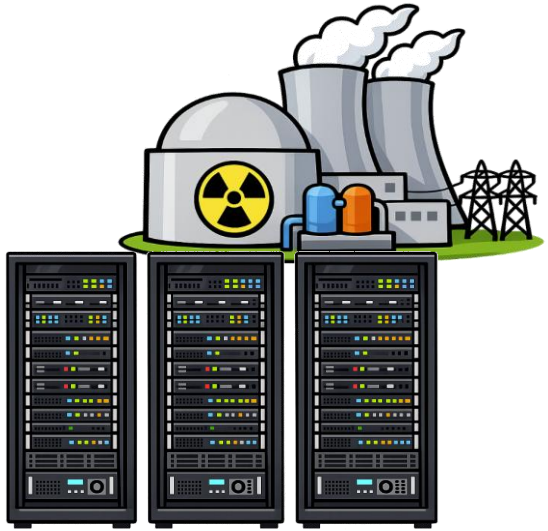
2026 CapEx projections: Side by side

2026 CapEx
TSMC
→ **$56bn**

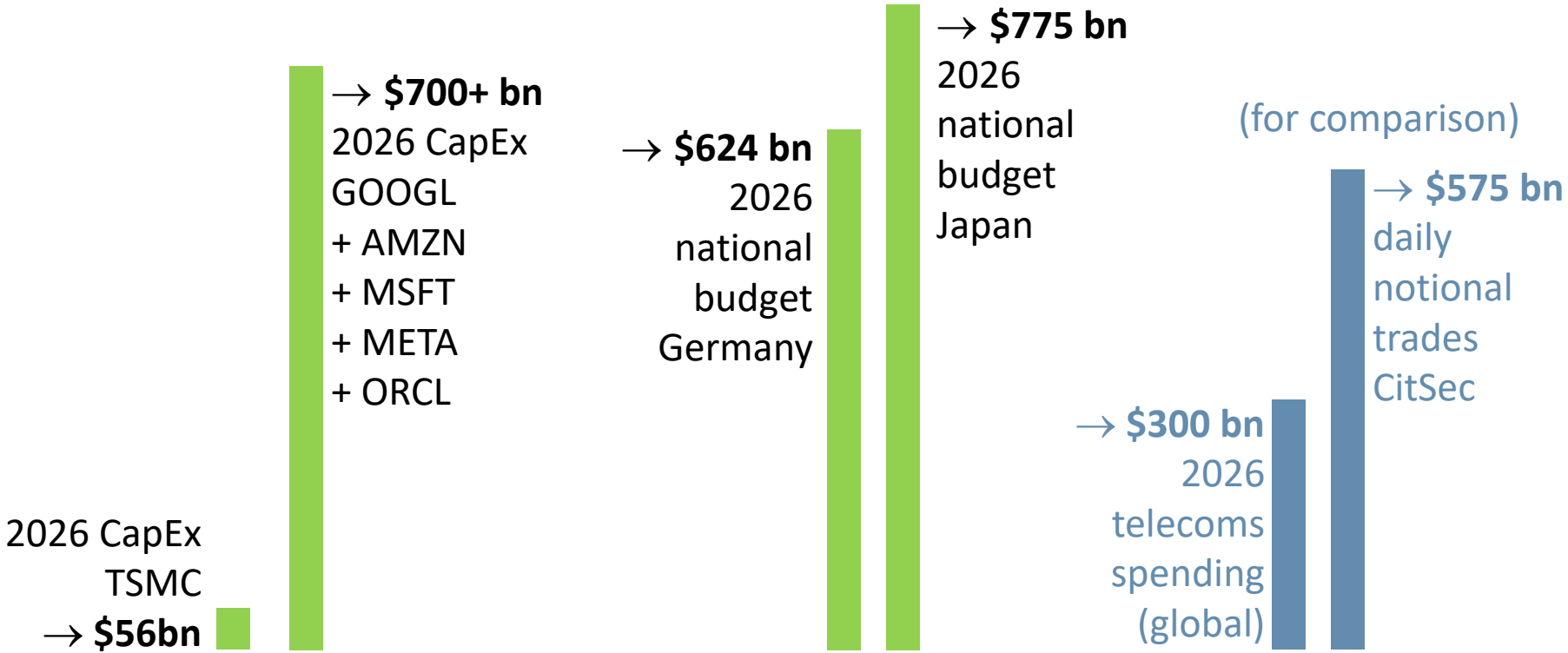
→ **$700+ bn**
2026 CapEx
GOOGL
+ AMZN
+ MSFT
+ META
+ ORCL

→ **$624 bn**
2026
national
budget
Germany

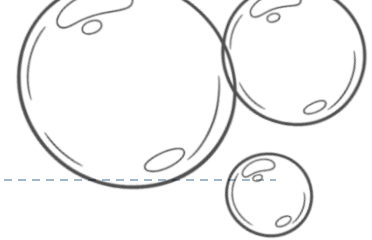
→ **$775 bn**
2026
national
budget
Japan



2026 CapEx projections: Side by side

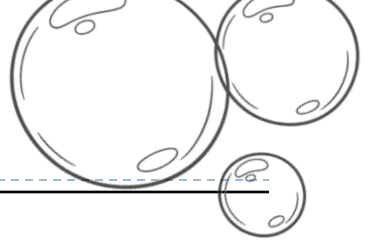


Aside: Good & bad bubbles



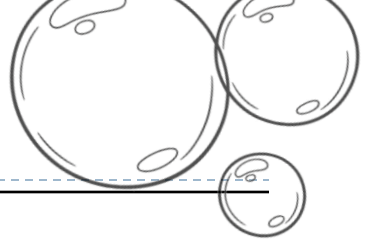
	Losers
1630s: Tulip mania	Speculators
1800s: U.S. railroads	Panics of 1873 & 1893
1920s: Stock mania	Decade-long recession
1990s: Dot-com	Speculators
2000s: Subprime, MBS	Homeowners, pensioners

Aside: Good & bad bubbles



	Losers	Durable gains
1630s: Tulip mania	Speculators	None
1800s: U.S. railroads	Panics of 1873 & 1893	
1920s: Stock mania	Decade-long recession	None
1990s: Dot-com	Speculators	
2000s: Subprime, MBS	Homeowners, pensioners	None

Aside: Good & bad bubbles



	Losers	Durable gains
1630s: Tulip mania	Speculators	None
1800s: U.S. railroads	Panics of 1873 & 1893	Transcontinental freight rail
1920s: Stock mania	Decade-long recession	None
1990s: Dot-com	Speculators	Land & cell internet buildout
2000s: Subprime, MBS	Homeowners, pensioners	None

Key Is there **coordinated building of durable value** that wouldn't be built otherwise?

AI Might be a bubble: If so, losers would be stuck with chips @ ~5yr depreciation

Either way: Building durable **power capacity** @ ~15-20yr depreciation, especially **clean nuclear** *(caveat: short-term it's driving a gas boom)*

So: Why is C++ still so relevant?

Perf/W is an enduring metric

Also: C++ keeps evolving to stay relevant to modern hardware and software

- **C++26: More SIMD types** for intra-CPU vector parallelism
- **C++26: std::execution** for general CPU/GPU concurrency and parallelism

Including important workloads

- **Datacenter/cloud**: Power and compute have been the major costs for cloud software for as long as there has been cloud software
- **Low-latency** (not just games): The world's financial systems are largely built on C++
- **AI**: If you're doing AI you're using CUDA (or TF or similar), directly or indirectly, and therefore you're probably writing or running C++
  - For general-purpose AI, most high-perf deployment and inference is implemented in C++, even if accessed via higher-level code in other languages

C++ in 2026

C++ growth, fueled by our insatiable hunger for compute

C++ perspective on structural global changes now in progress

Power	Record-shattering 2026 CapEx
Security	Toto, we're not in 2021 anymore
AI	Software engineering cost impact (and non-impact)

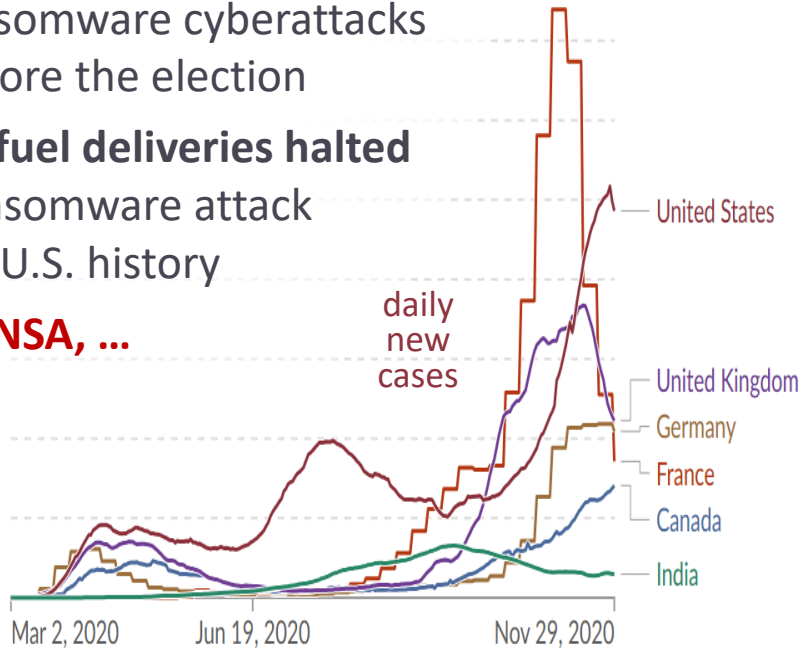
Recall: The winter of 2020-2021

"C (and C++) code has vulnerabilities" known for decades.
Regulators stared caring because of ransomware

Oct 2020: 100s of U.S. hospitals targeted by ransomware cyberattacks — in the fall 2020 COVID spike + weeks/days before the election

May 7, 2021: Colonial Pipeline gasoline and jet fuel deliveries halted & emergencies declared in 17 states + DC — ransomware attack is the largest cyberattack on oil infrastructure in U.S. history

May 12, 2021: Exec Order 14028 → NIST, CISA, NSA, …



That was 5 years ago

We *have* been hardening our software

Now: memory safety exploits *less* severe, *less* common, and *more* $$$ for attackers…

MITRE "**Most Dangerous** Software Weaknesses"

2023 Top 10

cwe.mitre.org/top25/archive/2023/2023_top25_list.html#tableView

1	Out-of-bounds Write
2	Cross-site Scripting
3	SQL Injection
4	Use After Free
5	OS Cmd Injection
6	Improper Input Validation
7	Out-of-bounds Read
8	Restricted Path Traversal
9	Cross-Site Req. Forgery
10	Dangerous File Upload

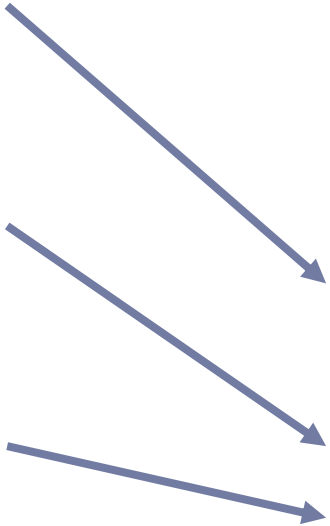
Q: Which of these are related to **programming language safety**?

MITRE "**Most Dangerous** Software Weaknesses"

2023 Top 10

cwe.mitre.org/top25/archive/2023/2023_top25_list.html#tableView

1	**Out-of-bounds Write**
2	Cross-site Scripting
3	SQL Injection
4	**Use After Free**
5	OS Cmd Injection
6	Improper Input Validation
7	**Out-of-bounds Read**
8	Restricted Path Traversal
9	Cross-Site Req. Forgery
10	Dangerous File Upload



2025 Top 10

cwe.mitre.org/top25/archive/2025/2025_cwe_top25.html#tableView

1	Cross-site Scripting
2	SQL Injection
3	Cross-Site Req. Forgery
4	Missing Authorization
5	**Out-of-bounds Write**
6	Restricted Path Traversal
7	**Use After Free**
8	**Out-of-bounds Read**
9	OS Cmd Injection
10	Code Injection

Attackers go after the **slowest** in the herd

Yes, some vulnerabilities are about language memory safety…

Only 3 of 10 MITRE “Most Dangerous,” and dropping

C++26’s hardened standard library addresses 2 of the 3:
bounds-checks the most widely used bounded operations

… **but most attacks are not about language weaknesses**…

See the **other 7 of 10** MITRE’s “Most Dangerous”

We *have* been hardening our software… 0-day cost: $000s → **$000,000s**

CrowdStrike 2025 Global Threat Report: ***“79% of detections were malware-free…***

… **and increasingly not about software at all**

*“… **442% growth in vishing** operations between the first and second half of 2024.”*

Sad fact: Hundreds of thousands of people in forced labor in criminal call centers

Attackers go after the **slowest** in the herd

Yes, some vulnerabilities are about language memory safety.

- **Only 3 of 10** MITRE "Most Dange
- C++26's hardened standard librar
- **bounds-checks** the most widely u

… **but most attacks are not abou**

- See the **other 7 of 10** MITRE's "M
- We *have* been hardening our soft
- CrowdStrike 2025 Global Threat R

… **and increasingly not about so**

- *"… **442% growth in vishing** opera*
- Sad fact: Hundreds of thousands

Cybercriminal with $10M budget, 2026 edition:

*"Why **buy 10 0-days** …*

*… when I can **spend way less** exploiting XSS / SQL injection / XSRF / Missing Auth …*

*… or could **spend the same to staff a SE Asia call center** to do voice phishing + text/chat scams + romance/investment scams + live password reset request calls + … ?"*

Attackers go after the **slowest** in the herd

Changing a cost by 1000× structurally changes the whole market

The whole point of memory safety hardening has always been **to raise the cost of memory safety exploits**

… which pushes the need to harden to the next slowest animals, so better go work on those next

Cybercriminal with $10M budget, 2026 edition:

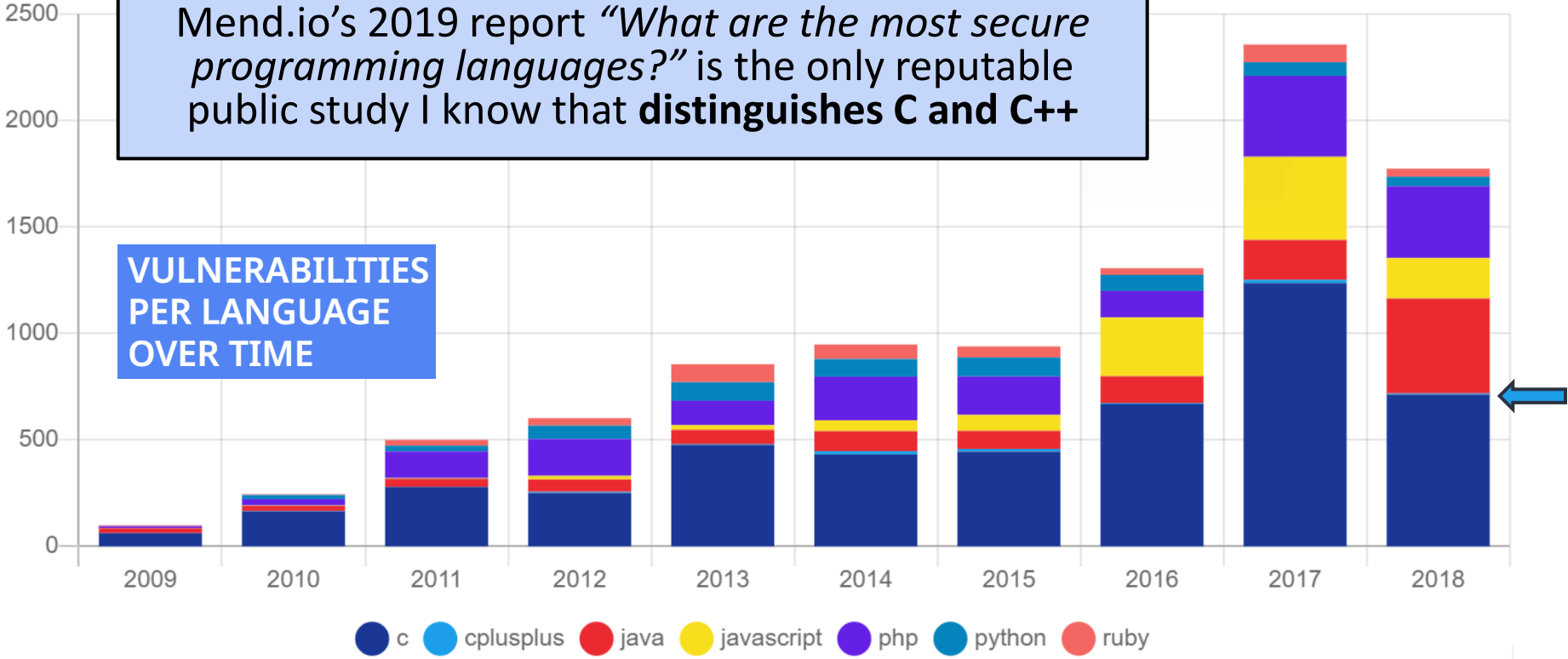
*"Why **buy 10 0-days** …*

*… when I can **spend way less** exploiting XSS / SQL injection / XSRF / Missing Auth …*

*… or could **spend the same to staff a SE Asia call center** to do voice phishing + text/chat scams + romance/investment scams + live password reset request calls + … ?"*

Even for memory safety, **the problem is with C**

Mend.io's 2019 report *"What are the most secure programming languages?"* is the only reputable public study I know that **distinguishes C and C++**



www.mend.io/most-secure-programming-languages/
www.mend.io/blog/is-one-programming-language-more-secure/

C++26 contains even more improvements…

More **init memory safety**: Reading uninit locals is not undefined

- To opt out, use [[indeterminate]]

More **bounds memory safety**: Hardened standard library

- Covers most popular bounded ops (vector, string, span, string_view, …)
- ACM Queue, November 2025: *queue.acm.org/detail.cfm?id=3773097*
- Deployed on Apple platforms (incl. WebKit), nearly all Google services (100s MLOC, incl. Chrome), avg. +0.5%/+0.3% space/speed overhead

"Actually it's going to be super easy, barely an inconvenience"

Rebuild as C++26 ⇒ code is X% safer

More **functional safety**: Contracts (pre, post, contract_assert)

C++ in 2026

C++ growth, fueled by our insatiable hunger for compute

C++ perspective on structural global changes now in progress

Power	Record-shattering 2026 CapEx
Security	Toto, we're not in 2021 anymore
AI	Software engineering cost impact (and non-impact)

AI: New tools (always) change the cost landscape

Drudgery is much cheaper: Good commit messages, refactoring to use a similar API, already-known data structures and algorithms

Long-tail shallow bugs/features are cheaper + deterministic: Things we wouldn't fix because of opportunity cost + uncertainty

- Rare bug can be feasible if 10min prompt + 2h bkgd AI generates a repro

But: Caveats and tradeoffs

- **Creative work isn't cheaper:** Don't know how to automate with AI
- **Strategic work can be choked out:** Diving into 100s of long-tail shallow backlog items is time not spent on "big rocks" AI can't deliver
- **Evaluating generated code has a human cost:** Good enough? scalable? correct? usable? submarine issues? — Fundamental + not automatable
- **Most ongoing cost is in code maintenance:** Generating lots of "good enough" code faster piles up the hidden cost of long-term technical debt

Predicted

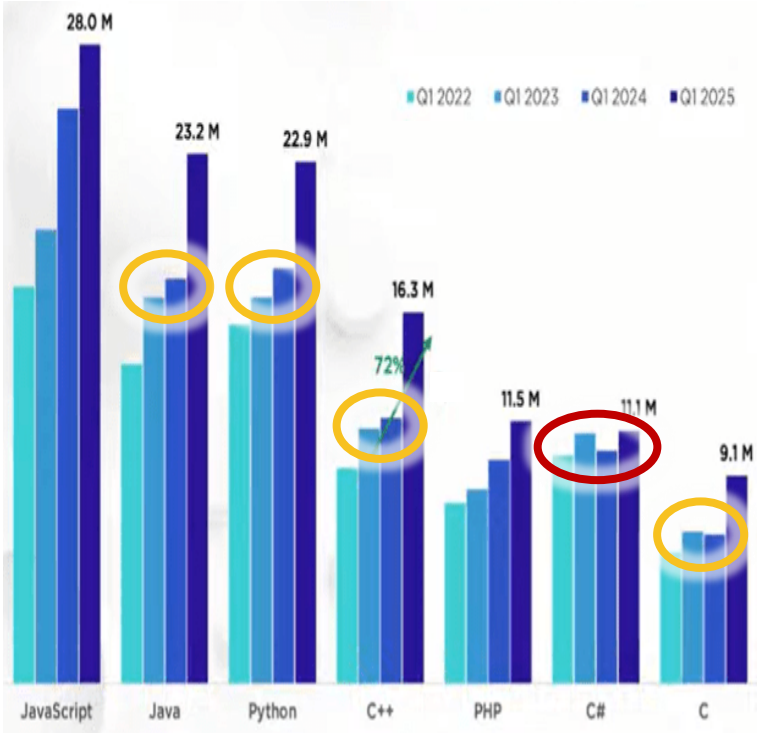
Jul 2023: Emad Mostaque, Stability.AI CEO

*There will be **no programmers** in five years.*

Mar 2025: Dario Amodei, Anthropic CEO

*I think we will be there in three to six months, where **AI is writing 90% of the code**. And then, in 12 months, we may be in a world where **AI is writing essentially all of the code***

Actual



Predicted

Jul 2023: Emad Mostaque, Stability.AI CEO

*There will be **no programmers** in five years.*

Mar 2025: Dario Amodei, Anthropic CEO

*I think we will be there in three to six months, where **AI is writing 90% of the code**. And then, in 12 months, we may be in a world where **AI is writing essentially all of the code***

What seems likely

Durable long-term demand for human programmers

- Esp. who use tools well (incl. AI)
- Esp. who get most perf/Watt

Humans won't be replaced en masse, we'll be **more efficient + numerous**

- ⇒ **Produce more/better software**
- Incl. new software that **wouldn't otherwise have been written** (often special-purpose / small / custom)
- … Just like with every new tool (remember VB? Excel macros? …)

But wait! Disconfirming data … ?

Feb 27, 2026: Block **cutting 40%** (4k) jobs
(across company, not specifically programmers)

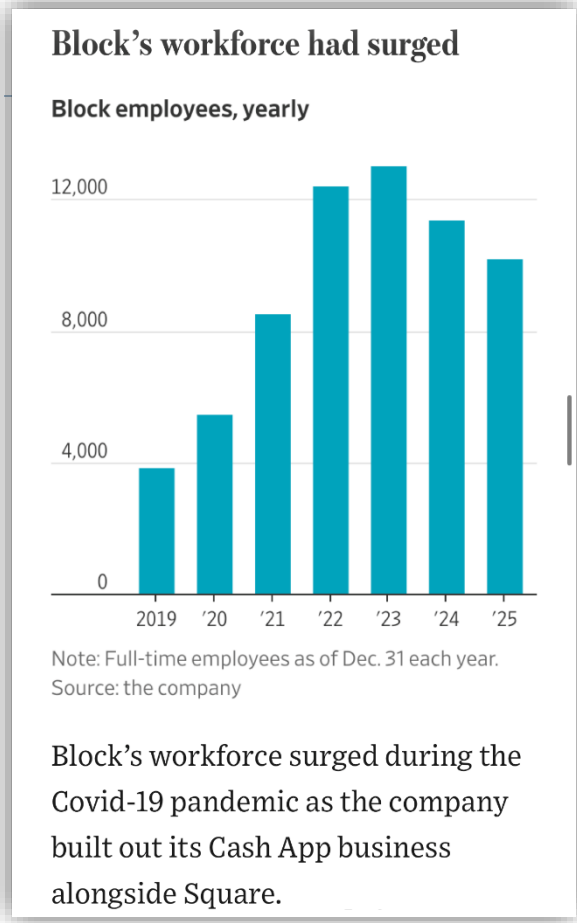
Jack Dorsey, shareholder letter:

> ***"[Due to Artificial] Intelligence tools … a significantly smaller team … can do more and do it better. …***
>
> *I don't think we're early to this realization. I think most companies are late. Within the next year, I believe the majority of companies will reach the same conclusion and make similar structural changes. "*

Reality check: **"AI-washing"**

Big tech has just been correcting absurd over-hiring

(Except – congrats!)



Survey of 1,000 U.S. hiring managers

Resume.org, “The Great Turnover: 9 in 10 Companies Plan to Hire in 2026, Yet 6 in 10 Will Have Layoffs”
tinyurl.com/resume-org-2026

“59% admit they emphasize AI when explaining hiring freezes or layoffs because it plays better *with stakeholders than citing financial constraints”*

Some top of mind Qs

Q: How does AI affect programming language design?

A: Directly expressing intent helps AI (too)

- We already do best by evolving languages/libraries to **directly express intent — elevate common patterns** to native operations
  - Examples: range-for, lambdas, concepts, contracts, metaclasses
- What already helps humans & tools … drum roll … also helps AI (a tool!)
  - Consider: **range-for** and **pre/post on function decl** ⇒ human / static analyzer / optimizer / AI doesn't need to reason about the body (e.g., to hoist a bounds check)

Q: Is AI less helpful for new standards / new languages?

A: Not anymore, especially since "o1" reasoning models

- Example: Sender-receiver

C++ in 2026

C++ growth, fueled by our insatiable hunger for compute

C++ perspective on structural global changes now in progress

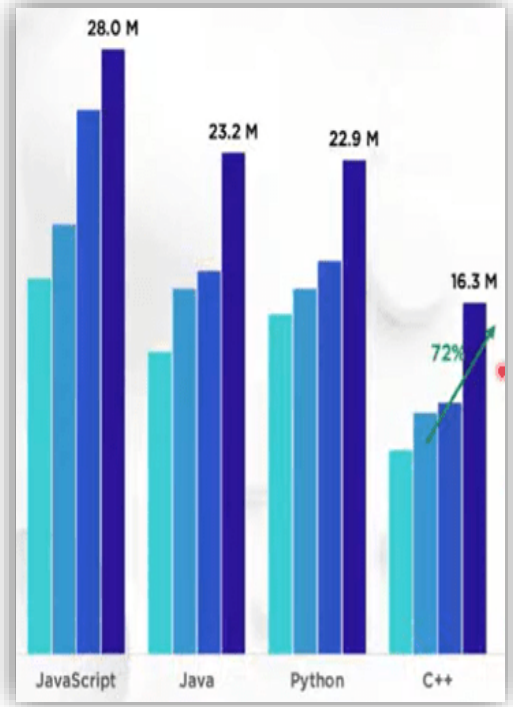
Power	Record-shattering 2026 CapEx
Security	Toto, we're not in 2021 anymore
AI	Software engineering cost impact (and non-impact)

We're growing gangbusters for a reason

The future is enduringly bright for languages that value "performance per Watt"… and for human developers

The size of new computing problems we want to solve has routinely **outstripped our computing capacity and human developer supply** for the past 80 years — I know of no reason why that would change in the next 80 years

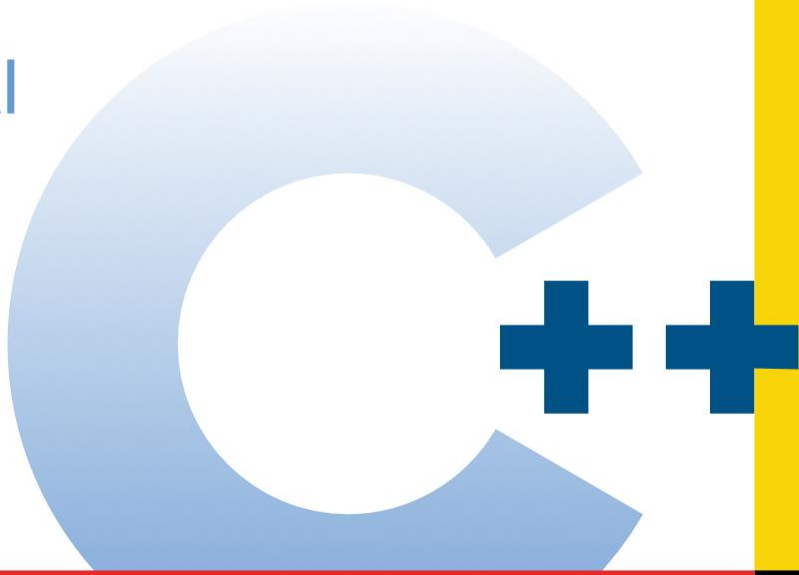
The list of major **general-purpose** languages that target the durable perf/Watt metric is short: ***C, C++, Rust***
(DSLs/ASICs/FPGAs will always be used on leading edge)



BeC++ SYMPOSIUM

Herb Sutter

C++: Growing in a world of competition, safety, and AI



BECPP SYMPOSIUM 2026 | March 30th | Kortrijk