FRED TALKS

11th April 2026

CONTENTS

Scaling Managed Agents: Decoupling the brain from the hands

ANTHROPIC.COM · 09 APR 2026 · SOURCE

Get started with Claude Managed Agents by following our docs.

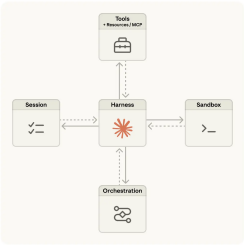
A running topic on the Engineering Blog is how to build effective agents and design harnesses for long-running work. A common thread across this work is that harnesses encode assumptions about what Claude can't do on its own. However, those assumptions need to be frequently questioned because they can go stale as models improve.

As just one example, in prior work we found that Claude Sonnet 4.5 would wrap up tasks prematurely as it sensed its context limit approaching—a behavior sometimes called "context anxiety." We addressed this by adding context resets to the harness. But when we used the same harness on Claude Opus 4.5, we found that the behavior was gone. The resets had become dead weight.

We expect harnesses to continue evolving. So we built Managed Agents: a hosted service in the Claude Platform that runs long-horizon agents on your behalf through a small set of interfaces meant to outlast any particular implementation—including the ones we run today.

Building Managed Agents meant solving an old problem in computing: how to design a system for "programs as yet unthought of." Decades ago, operating systems solved this problem by virtualizing hardware into abstractions—*process*, *file*—general enough for programs that didn't exist yet. The abstractions outlasted the hardware. The `read()` command is agnostic as to whether it's accessing a disk pack from the 1970s or a modern SSD. The abstractions on top stayed stable while the implementations underneath changed freely.

Managed Agents follow the same pattern. We virtualized the components of an agent: a session (the append-only log of everything that happened), a harness (the loop that calls Claude and routes Claude's tool calls to the relevant infrastructure), and a sandbox (an execution environment where Claude can run code and edit files). This allows the implementation of each to be swapped without disturbing the others. We're opinionated about the shape of these interfaces, not about what runs behind them.



We started by placing all agent components into a single container, which meant the session, agent harness, and sandbox all shared an environment. There were benefits to this approach, including that file edits are direct syscalls, and there were no service boundaries to design.

But by coupling everything into one container, we ran into an old infrastructure problem: we'd adopted a *pet*. In the pets-vs-cattle analogy, a pet is a named, hand-tended individual you can't afford to lose, while cattle are interchangeable. In our case, the server became that pet; if a container failed, the session was lost. If a container was unresponsive, we had to nurse it back to health.

Nursing containers meant debugging unresponsive stuck sessions. Our only window in was the WebSocket event stream, but that couldn't tell us *where* failures arose, which meant that a bug in the harness, a packet drop in the event stream, or a container going offline all presented the same. To figure out what went wrong, an engineer had to open a shell inside the container, but because that container often also held user data, that approach essentially meant we lacked the ability to debug.

A second issue was that the harness assumed that whatever Claude worked on lived in the container with it. When customers asked us to connect Claude to their virtual private cloud, they had to either peer their network with ours, or run our harness in their own environment. An assumption baked into the harness became a problem when we wanted to connect it to different infrastructure.

The solution we arrived at was to decouple what we thought of as the "brain" (Claude and its harness) from both the "hands" (sandboxes and tools that perform actions) and the "session" (the log of session events). Each became an interface that made few assumptions about the others, and each could fail or be replaced independently.

The harness leaves the container. Decoupling the brain from the hands meant the harness no longer lived inside the container. It called the container the way it called any other tool: `execute(name, input) → string`. The container became cattle. If the container died,

the harness caught the failure as a tool-call error and passed it back to Claude. If Claude decided to retry, a new container could be reinitialized with a standard recipe: `provision({resources})`. We no longer had to nurse failed containers back to health.

Recovering from harness failure. The harness also became cattle. Because the session log sits outside the harness, nothing in the harness needs to survive a crash. When one fails, a new one can be rebooted with `wake(sessionId)`, use `getSession(id)` to get back the event log, and resume from the last event. During the agent loop, the harness writes to the session with `emitEvent(id, event)` in order to keep a durable record of events.

Component	Interface (pseudocode)	Satisfied by

Session	`getSession(session_id) -> [Session, Event[]]`
	`getEvents(session_id) -> PendingEvent[] // not yet processed`
	`emitEvent(id, event)`
		Any append-only log that can be coalesced in order from any event point and accepts idempotent appends—Postgres, SQLite, in-memory array, etc.

Orchestration	`wake(session_id) -> void`	Any scheduler that can call a function with an ID and retry on failure—a cron job, a queue consumer, a while-loop, etc.
Harness	`yield Effect<T> -> EffectResult<T>`	Any loop that yields effects and appends progress to the Session.
Sandbox	`provision([resources]) execute(name, input) -> string`	Any executor that can be configured once and called many times as a tool—a local process, a remote container, etc.
Resources	`[{source_ref, mount_path}]`	Any durable store the container can fetch from by reference—Filestore, GCS, a git remote, S3.
Tools	`[name, description, input_schema]`	Any capability describable as a name and an input shape—MCP server, custom tool, etc.

The security boundary. In the coupled design, any untrusted code that Claude generated was run in the same container as credentials—so a prompt injection only had to convince Claude to read its own environment. Once an attacker has those tokens, they can spawn fresh, unrestricted sessions and delegate work to them. Narrow scoping is an obvious mitigation, but this encodes an assumption about what Claude can't do with a limited token—and Claude is getting increasingly smart. The structural fix was to make sure the tokens are never reachable from the sandbox where Claude's generated code runs.

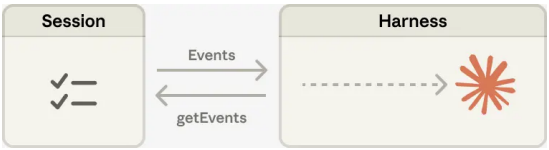
We used two patterns to ensure this. Auth can be bundled with a resource or held in a vault outside the sandbox. For Git, we use each repository's access token to clone the repo during sandbox initialization and wire it into the local git remote. Git `push` and `pull` work from inside the sandbox without the agent ever handling the token itself. For custom tools, we support MCP and store OAuth tokens in a secure vault. Claude calls MCP tools via a dedicated proxy; this proxy takes in a token associated with the session. The proxy can then fetch the corresponding credentials from the vault and make the call to the external service. The harness is never made aware of any credentials.

The session is not Claude's context window

Long-horizon tasks often exceed the length of Claude's context window, and the standard ways to address this all involve irreversible decisions about what to keep. We've explored these techniques in prior work on context engineering. For example, compaction lets Claude save a

summary of its context window and the memory tool lets Claude write context to files, enabling learning across sessions. This can be paired with context trimming, which selectively removes tokens such as old tool results or thinking blocks.

But irreversible decisions to selectively retain or discard context can lead to failures. It is difficult to know which tokens the future turns will need. If messages are transformed by a compaction step, the harness removes compacted messages from Claude's context window, and these are recoverable only if they are stored. Prior work has explored ways to address this by storing context as an object that lives *outside* the context window. For example, context can be an object in a REPL that the LLM programmatically accesses by writing code to filter or slice it.



In Managed Agents, the session provides this same benefit, serving as a context object that lives outside Claude's context window. But rather than be stored within the sandbox or REPL, context is durably stored in the session log. The interface, `getEvents()`, allows the brain to interrogate context by selecting positional slices of the event stream. The interface can be used flexibly, allowing the brain to pick up from wherever it last stopped reading, rewinding a few events before a specific moment to see the lead up, or rereading context before a specific action.

Any fetched events can also be transformed in the harness before being passed to Claude's context window. These transformations can be whatever the harness encodes, including context organization to achieve a high prompt cache hit rate and context engineering. We separated the concerns of recoverable context storage in the session and arbitrary context management in the harness because we can't predict what specific context engineering will be required in future models. The interfaces push that context management into the harness, and only guarantee that the session is durable and available for interrogation.

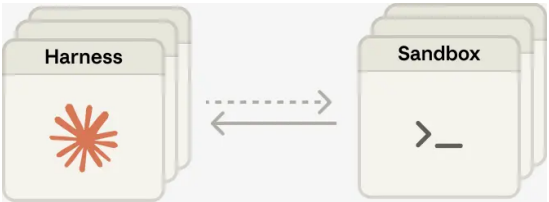
Many brains. Decoupling the brain from the hands solved one of our earliest customer complaints. When teams wanted Claude to work against resources in their own VPC, the only path was to peer their network with ours, because the container holding the harness assumed every resource sat next to it. Once the harness was no longer in the container, that assumption went away. The same change had a performance payoff. When we initially put the brain in a container, it meant that many brains required as many containers. For each brain, no inference could happen until that container was provisioned; every session paid the full container setup cost up front. Every session, even ones that would never touch the sandbox, had to clone the repo, boot the process, fetch pending events from our servers.

That dead time is expressed in time-to-first-token (TTFT), which measures how long a session waits between accepting work and producing its first response token. TTFT is the latency the user most acutely *feels*.

Decoupling the brain from the hands means that containers are provisioned by the brain via a tool call `(execute(name, input) → string)` only if they are needed. So a session that didn't need a container right away didn't wait for one. Inference could start as soon as the orchestration layer pulled pending events from the session log. Using this architecture, our p50 TTFT dropped roughly 60% and p95 dropped over 90%. Scaling to many brains just meant starting many stateless harnesses, and connecting them to hands only if needed.

Many hands. We also wanted the ability to connect each brain to many hands. In practice, this means Claude must reason about many execution environments and decide where to send work —a harder cognitive task than operating in a single shell. We started with the brain in a single container because earlier models weren't capable of this. As intelligence scaled, the single container became the limitation instead: when that container failed, we lost state for every hand that the brain was reaching into.

Decoupling the brain from the hands makes each hand a tool, `execute(name, input) → string`: a name and input go in, and a string is returned. That interface supports any custom tool, any MCP server, and our own tools. The harness doesn't know whether the sandbox is a container, a phone, or a Pokémon emulator. And because no hand is coupled to any brain, brains can pass hands to one another.



The challenge we faced is an old one: how to design a system for "programs as yet unthought of." Operating systems have lasted decades by virtualizing the hardware into abstractions general enough for programs that didn't exist yet. With Managed Agents, we aimed to design a system that accommodates future harnesses, sandboxes, or other components around Claude.

Managed Agents is a meta-harness in the same spirit, unopinionated about the *specific* harness that Claude will need in the future. Rather, it is a system with general interfaces that allow many different harnesses. For example, Claude Code is an excellent harness that we use widely across tasks. We've also shown that task-specific agent harnesses excel in narrow domains. Managed Agents can accommodate any of these, matching Claude's intelligence over time.

Meta-harness design means being opinionated about the interfaces around Claude: we expect that Claude will need the ability to manipulate state (the session) and perform computation (the sandbox). We also expect that Claude will require the ability to scale to many brains and many hands. We designed the interfaces so that these can be run reliably and securely over long time horizons. But we make no assumptions about the number or location of brains or hands that Claude will need.

Written by Lance Martin, Gabe Cemaj and Michael Cohen. Special thanks to the Agents API team and Jake Eaton for their contributions.

[Daily article] April 11: Relief of Douglas MacArthur

ENGLISH WIKIPEDIA ARTICLE OF THE DAY · 11 APR 2026 · SOURCE

On 11 April 1951, General Douglas MacArthur was relieved of command by U.S. president Harry S. Truman (both pictured) after making statements that contradicted the administration's policies. MacArthur was a popular hero of World War II, achieving the rank of General of the Army, and his relief remains a controversial topic in civil–military relations. After North Korea invaded South Korea in June 1950, starting the Korean War, MacArthur won the Battle of Inchon, but the following invasion of North Korea on Truman's orders led to China inflicting a series of defeats. MacArthur was compelled to withdraw from North Korea and, after the military situation had stabilized, Truman relieved him, creating a constitutional crisis. The United States Senate held an inquiry into the military situation and the circumstances surrounding MacArthur's relief, and concluded that "the removal of General MacArthur was within the constitutional powers of the President but the circumstances were a shock to national pride". (Full article...).

Read more: https://en.wikipedia.org/wiki/Relief_of_Douglas_MacArthur

---

Today's selected anniversaries:

1885:

Luton Town Football Club was founded in England. https://en.wikipedia.org/wiki/Luton_Town_F.C.

1951:

The Stone of Scone, removed months earlier by Scottish students, was found on the altar of Arbroath Abbey. https://en.wikipedia.org/wiki/Stone_of_Scone

1973:

On the Art of the Cinema, a treatise on film propaganda in support of the ruling Workers' Party of Korea written by the future North Korean leader Kim Jong Il, was published. https://en.wikipedia.org/wiki/On_the_Art_of_the_Cinema

2011:

A bomb exploded at Kastryčnickaja station on the Minsk Metro in Belarus, killing 15 people and injuring more than 200. https://en.wikipedia.org/wiki/2011_Minsk_Metro_bombing

---

Wiktionary's word of the day:

befriend: 1. (also figurative) To become a friend of (someone), to make friends with (someone). 2. (dated except social work) To act as a friend to (someone, especially a client) by providing companionship and support; to assist, to help. 3. (figurative) To benefit or favour (someone or something); also, to advocate or promote (someone or something). 4. About Word of the Day 5. Nominate a word 6. Leave feedback https://en.wiktionary.org/wiki/befriend

---

Wikiquote quote of the day:

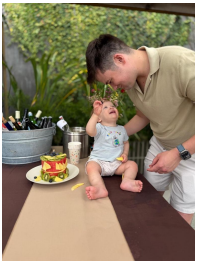
She entered heaven; she climbed the stair And knelt down at the mercy- seat. Seraphs and saints with one great voice Welcomed that soul that knew not fear.Amazed to find it could rejoice, Hell raised a hoarse, half-human cheer. --John Davidson https://en.wikiquote.org/wiki/John_Davidson

---

-

SAM ALTMAN · 10 APR 2026 · SOURCE

Here is a photo of my family. I love them more than anything.



Images have power, I hope. Normally we try to be pretty private, but in this case I am sharing a photo in the hopes that it might dissuade the next person from throwing a Molotov cocktail at our house, no matter what they think about me.

The first person did it last night, at 3:45 am in the morning. Thankfully it bounced off the house and no one got hurt.

Words have power too. There was an incendiary article about me a few days ago. Someone said to me yesterday they thought it was coming at a time of great anxiety about AI and that it made things more dangerous for me. I brushed it aside.

Now I am awake in the middle of the night and pissed, and thinking that I have underestimated the power of words and narratives. This seems like as good of a time as any to address a few things.

First, what I believe.

*Working towards prosperity for everyone, empowering all people, and advancing science and technology are moral obligations for me.

*AI will be the most powerful tool for expanding human capability and potential that anyone has ever seen. Demand for this tool will be essentially uncapped, and people will do incredible things with it. The world deserves huge amounts of AI and we must figure out how to make it happen.

*It will not all go well. The fear and anxiety about AI is justified; we are in the process of witnessing the largest change to society in a long time, and perhaps ever. We have to get safety right, which is not just about aligning a model—we urgently need a society-wide response to be resilient to new threats. This includes things like new policy to help navigate through a difficult economic transition in order to get to a much better future.

*AI has to be democratized; power cannot be too concentrated. Control of the future belongs to all people and their institutions. AI needs to empower people individually, and we need to make decisions about our future and the new rules collectively. I do not think it is right that a few AI labs would make the most consequential decisions about the shape of our future.

*Adaptability is critical. We are all learning about something new very quickly; some of our beliefs will be right and some will be wrong, and sometimes we will need to change our mind quickly as the technology develops and society evolves. No one understands the impacts of superintelligence yet, but they will be immense.

Second, some personal reflections.

As I reflect on my own work in the first decade of OpenAI, I can point to a lot of things I'm proud of and a bunch of mistakes.

I was thinking about our upcoming trial with Elon and remembering how much I held the line on not being willing to agree to the unilateral control he wanted over OpenAI. I'm proud of that, and the narrow path we navigated then to allow the continued existence of OpenAI, and all the achievements that followed.

I am not proud of being conflict-averse, which has caused great pain for me and OpenAI. I am not proud of handling myself badly in a conflict with our previous board that led to a huge mess for the company. I have made many other mistakes throughout the insane trajectory of OpenAI; I am a flawed person in the center of an exceptionally complex situation, trying to get a little better each year, always working for the mission. We knew going into this how huge the stakes of AI were, and that the personal disagreements between well-meaning people I cared about would be amplified greatly. But it's another thing to live through these bitter conflicts and often to have to arbitrate them, and the costs have been serious. I am sorry to people I've hurt and wish I had learned more faster.

I am also very aware that OpenAI is now a major platform, not a scrappy startup, and we need to operate in a more predictable way now. It has been an extremely intense, chaotic, and high-pressure few years.

Mostly though, I am extremely proud that we are delivering on our mission, which seemed incredibly unlikely when we started. Against all odds, we figured out how to build very powerful AI, figured out how to amass enough capital to build the infrastructure to deliver it, figured out how to build a product company and business, figured out how to deliver reasonably

safe and robust services at a massive scale, and much more.

A lot of companies say they are going to change the world; we actually did.

Third, some thoughts about the industry.

My personal takeaway from the last several years, and take on why there has been so much Shakespearean drama between the companies in our field, comes down to this: “Once you see AGI you can’t unsee it.” It has a real "ring of power” dynamic to it, and makes people do crazy things. I don’t mean that AGI is the ring itself, but instead the totalizing philosophy of “being the one to control AGI”.

The only solution I can come up with is to orient towards sharing the technology with people broadly, and for no one to have the ring. The two obvious ways to do this are individual empowerment and making sure democratic system stays in control.

It is important that the democratic process remains more powerful than companies. Laws and norms are going to change, but we have to work within the democratic process, even though it will be messy and slower than we’d like. We want to be a voice and a stakeholder, but not to have all the power.

A lot of the criticism of our industry comes from sincere concern about the incredibly high stakes of this technology. This is quite valid, and we welcome good-faith criticism and debate. I empathize with anti-technology sentiments and clearly technology isn’t always good for everyone. But overall, I believe technological progress can make the future unbelievably good, for your family and mine.

While we have that debate, we should de-escalate the rhetoric and tactics and try to have fewer explosions in fewer homes, figuratively and literally.

---

mist is now open source and looking for interop

INTERCONNECTED · 10 APR 2026 · SOURCE

A brief update on **mist,** my ephemeral Markdown editor with Google Docs-style comments and suggested edits:

mist is now open source with an MIT license, and the mist repo is here on GitHub.

(Try mist now and here's my write-up from February.)

What I love about Markdown is that it's document-first. The formatting travels with the doc. I can't tell you how many note-taking apps I've jumped between with my exact same folder of Markdown notes.

The same should be true for collaboration features like suggested edits. If somebody makes an edit to your doc, you should be able to download it and upload to a wholly different app before you accept the edit; you shouldn't be tied to a single service just because you want comments.

(And of course the doc should still be human-readable/writeable, and it's cheating to just stuff a massive data-structure in a document header.)

So mist mixes Markdown and CriticMarkup – and I would love it if others picked up the same format. If apps are cheap and abundant in the era of vibing, then let's focus on interop!

With mist itself:

Several people have asked for the ability to self-host it. The README says how (it's all on Cloudflare naturally). You can add new features to your own fork, though please do share upstream if you think others could benefit.

And yes, contributions welcome! We've already received and merged our first pull request – thank you James Adam!

No, a document editor is not what we're building at Inanimate. But it's neat to release small useful projects that get made along the way. btw subscribe to our newsletter.

If you enjoyed this post, please consider sharing it by email or on social media. Here's the link. Thanks, —Matt.

Issue #706

POINTER · 10 APR 2026 · SOURCE

April 10th, 2026 | Read online

Pointer

Friday 10th April's issue is presented by SerpApi

Add Real-Time Web Search To Your AI Apps

AI is smarter when it can search the web. How are AI companies doing it? Meet the worst-kept secret in the AI industry: the SerpApi web search API:

- 👉 Scrape Google and other search engines (including AI overviews, Maps, Shopping, Amazon, and more)
- 👉 Integrate with a dead-simple GET request that any agent can make.
- 👉 Used by NVIDIA, Uber, Adobe, and even the United Nations.

Give it a try now; no sign-up needed .

Explore Our Playground
Finding Comfort In The Uncertainty

— Annie Vella

tl;dr : A retreat of engineering leaders and researchers highlights growing uncertainty around AI's impact on software development. Key themes include shifting bottlenecks, rising cognitive load, evolving roles, and the need for new abstractions, governance, and systems to manage agents.

Leadership Management

The Alarm That Went Silent

— Mike Fisher

tl;dr : Teams optimize for metrics that are visible while quietly accumulating risk in areas that are harder to quantify: team burnout, customer frustration that hasn't yet surfaced as churn, growing operational fragility, cultural erosion, or decision latency caused by process overhead. Just as critically, teams rarely ask a harder question: How would we know if our metrics stopped telling the truth?

Leadership Management

Add Real-Time Web Search To Your AI Apps

tl;dr : AI is smarter when it can search the web. How are AI companies doing it? Meet the worst-kept secret in the AI industry: the SerpApi web search API: (1) Scrape Google and other search engines (including AI overviews, Maps, Shopping, Amazon, and more). (2) Integrate with a dead-simple GET request that any agent can make. (3) Used by NVIDIA, Uber, Adobe, and even the United Nations.

Promoted by SerpApi

Search AI Tools

Say The Thing You Want

— Matheus Lima

tl;dr : "When people know what you want, they can help you get it (so obvious, and yet…). This is especially true for your manager. A big part of their job is putting people in positions where they can grow. But they can't send opportunities your way if they don't know which direction you're heading. They have seven other reports and their own fires to deal with."

CareerGrowth Communication

“There are three essentials to leadership: humility, clarity and courage.”

— Fuchan Yuan

Feedback Flywheel

— Rahul Garg

tl;dr : “Every AI interaction generates signal: prompts that worked, context that was missing, patterns that succeeded, failures worth preventing. Most teams discard this signal. I propose a structured feedback practice that harvests learnings from AI sessions and feeds them back into the team's shared artifacts, turning individual experience into collective improvement.”

AI Process

Registration Now Open: IaCConf 2026

tl;dr : IaCConf brings together the engineers and platform leads who've already figured it out - and are willing to show their work. Catch live demos, practitioner talks, and candid discussion you won't find anywhere else. No theory or vendor pitches.

Promoted by Spacelift

Infrastructure Scale

DHH’s New Way Of Writing Code

— David Heinemeier Hansson, Gergely Orosz

tl;dr : David discusses his approach to building software, how it’s changed in the last six months, and why he now takes an agent-first approach, and how he barely writes any code by hand. We go into how he uses AI agents: which alter how he builds and explores ideas, but also how his standards of quality and craft remain the same.

Productivity Agents Scale

Components Of A Coding Agent

— Sebastian Raschka

tl;dr : “In this article, I want to cover the overall design of coding agents and agent harnesses: what they are, how they work, and how the different pieces fit together in practice.”

Agents Guide

The Git Commands I Run Before Reading Any Code

— Alex Piechowski

tl;dr : “The first thing I usually do when I pick up a new codebase isn’t opening the code. It’s opening a terminal and running a handful of git commands. Before I look at a single file, the commit history gives me a diagnostic picture of the project: who built it, where the problems cluster, whether the team is shipping with confidence or tiptoeing around land mines.”

BestPractices Git

Null Pointer

Pick Zero

Hand-drawn by Manu . View the archives here

Most Popular From Last Issue

Why Your Engineering Team Is Slow -Alex Piechowski

Notable Links

Clicky : AI teacher that next to your cursor.

Defuddle : Get the content of any page as markdown.

Interview Coach : Covers the full job search lifecycle.

JAI : Jail your AI agent.

Onyx : OS AI platform.

How did you like this issue of Pointer?

1 = Didn't enjoy it all // 5 = Really enjoyed it

1 | 2 | 3 | 4 | 5

[Daily article] April 10: Ojos del Salado

ENGLISH WIKIPEDIA ARTICLE OF THE DAY · 10 APR 2026 · SOURCE

Ojos del Salado is a dormant complex volcano in the Andes on the Argentina–Chile border. It is the highest volcano on Earth and the highest peak in Chile. The complex extends over an area of 70 to 160 square kilometres (27 to 62 sq mi) and its highest summit reaches an elevation of 6,893 metres (22,615 ft) above sea level. The mountain has extremely dry conditions, which prevent the formation of substantial glaciers and a permanent snow cover. Despite the arid climate, there is a permanent crater lake about 100 metres (330 ft) in diameter at an elevation of 6,480 to 6,500 metres (21,260 to 21,300 ft) within the summit crater and east of the main summit. This is the highest lake of any kind in the world. An international highway between Argentina and Chile crosses north of the mountain. During the middle of the 20th century, there was a debate on whether Ojos del Salado or Aconcagua was the highest mountain in South America that was eventually resolved in favour of Aconcagua. (Full article...).

Read more: https://en.wikipedia.org/wiki/Ojos_del_Salado

Today's selected anniversaries:

1741:

War of the Austrian Succession: Prussian forces defeated Austrian troops at the Battle of Mollwitz in present-day Małujowice, Poland, cementing Frederick II's authority over the newly conquered territory of Silesia. https://en.wikipedia.org/wiki/Battle_of_Mollwitz

1815:

Mount Tambora in Indonesia began the most powerful volcanic eruption in recorded history, killing at least 71,000 people and affecting temperatures worldwide. https://en.wikipedia.org/wiki/Mount_Tambora

2019:

Scientists from the Event Horizon Telescope project released the first image of the black hole at the center of the galaxy M87. https://en.wikipedia.org/wiki/Messier_87

Wiktionary's word of the day:

whimberry: 1. (UK, dialectal) The bilberry or whortleberry (Vaccinium myrtillus). 2. About Word of the Day 3. Nominate a word 4. Leave feedback https://en.wiktionary.org/wiki/whimberry

---

Wikiquote quote of the day:

We will explore, we will build. We will build ships, we will visit again. We will construct science outposts, we will drive rovers, we will do radio astronomy. We will found companies, we will bolster industry, we will inspire. But ultimately, we will always choose Earth, we will always choose each other. --Christina Koch https://en.wikiquote.org/wiki/Christina_Koch