

Google Chrome



Brian Rakowski,
Product Manager

TODAY, MOST OF WHAT WE USE THE WEB FOR ON A DAY-TO-DAY BASIS AREN'T JUST WEB PAGES, THEY'RE APPLICATIONS.

A stylized illustration of a woman with long dark hair and glasses, wearing a patterned sweater. She is surrounded by various icons representing web applications: a video player, a calendar, a speaker, a list of items, and a grid. Two speech bubbles are positioned above her, and a navigation bar is visible in the top right corner.

PEOPLE ARE WATCHING AND UPLOADING VIDEOS, CHATTING WITH EACH OTHER, PLAYING WEB-BASED GAMES...

ALL THESE THINGS THAT DIDN'T EXIST WHEN THE FIRST BROWSERS WERE CREATED.

Azula88: ROFL ^^
DctrHrrbl: LOL
Trg: OMFG

Pam Greene, Software Engineer

WOULDN'T IT BE GREAT, THEN, TO START FROM SCRATCH --



-- AND DESIGN SOMETHING BASED ON THE NEEDS OF TODAY'S WEB APPLICATIONS AND TODAY'S USERS?

2008



FIRST, BROWSERS NEED TO BE MORE **STABLE**. WHEN YOU'RE WRITING AN IMPORTANT EMAIL OR EDITING A DOCUMENT, A BROWSER CRASH IS A BIG DEAL.



Darin Fisher, Software Engineer

BROWSERS ALSO NEED TO BE **FASTER**. THEY NEED TO START FASTER, LOAD PAGES FASTER --


-- AND FOR WEB APPS, JAVASCRIPT ITSELF CAN BE A LOT FASTER.



Lars Bak, Software Engineer

Kasper Lund, Software Engineer

THEY NEED TO BE MORE **SECURE**. GIVEN WHAT'S KNOWN ABOUT MASS BROWSER EXPLOITS, BROWSERS NEED ARCHITECTURAL CHANGES TO DISADVANTAGE MALWARE.



Ian Fette, Product Manager

AND WE WANT BROWSERS TO FIND THAT SWEET SPOT BETWEEN TOO MANY FEATURES AND TOO FEW, WITH A **CLEAN, SIMPLE, AND EFFICIENT** USER INTERFACE.



Ben Goodger, Software Engineer

FINALLY, GOOGLE CHROME IS A FULLY **OPEN SOURCE** BROWSER.

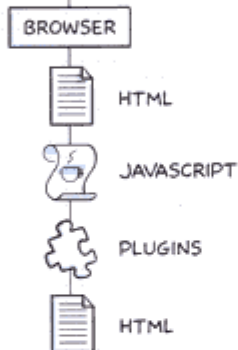
WE WANT OTHERS TO ADOPT IDEAS FROM US --




-- JUST AS WE'VE ADOPTED GOOD IDEAS FROM OTHERS.



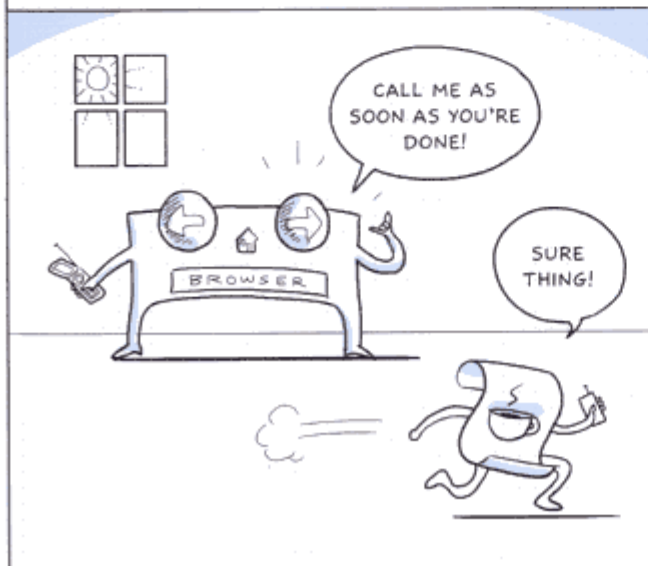
WHEN WE STARTED THIS PROJECT, THE GEARS GUYS WERE SAYING THAT ONE OF THE PROBLEMS WITH BROWSERS IS THAT THEY'RE INHERENTLY SINGLE-THREADED.



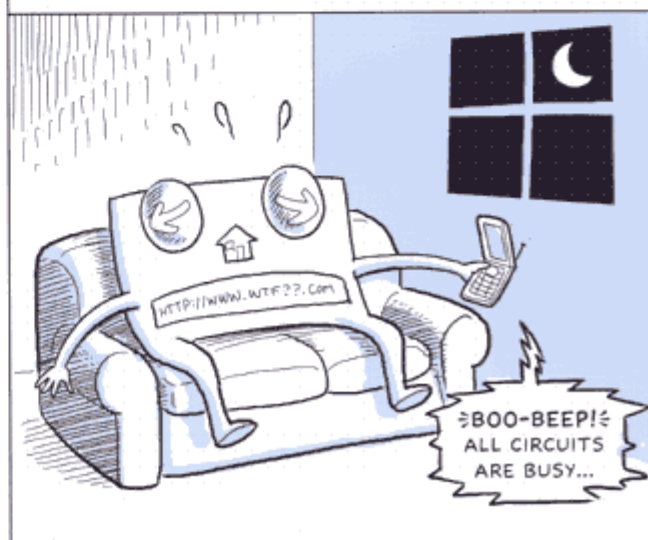
FOR EXAMPLE, ONCE YOU HAVE JAVASCRIPT EXECUTING, IT'S GOING TO KEEP GOING, AND THE BROWSER CAN'T DO ANYTHING ELSE UNTIL JAVASCRIPT RETURNS CONTROL TO THE BROWSER.



SO DEVELOPERS WRITE APIS THAT ARE ASYNCHRONOUS --



-- AND EVERY NOW AND THEN THE BROWSER LOCKS UP BECAUSE JAVASCRIPT IS HUNG UP ON SOMETHING.



THE GEARS GUYS WERE THINKING ABOUT A MULTI-THREADED BROWSER, BUT THAT LED US TO TALK ABOUT, WELL, INSTEAD OF MULTIPLE THREADS --

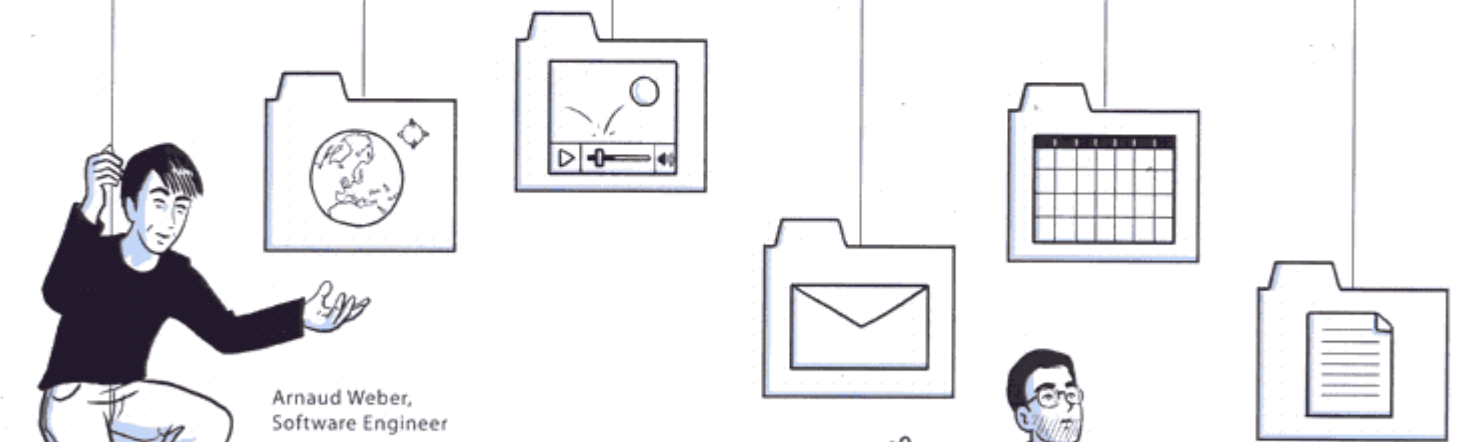
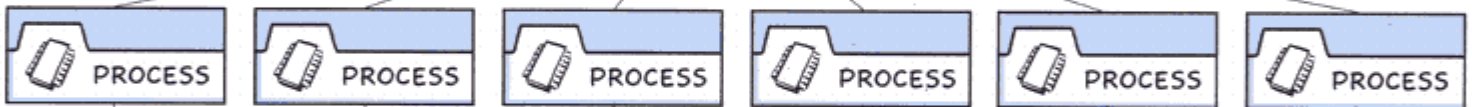
BROWSER PROCESS



-- WHAT IF WE HAVE MULTIPLE PROCESSES? EACH HAVING ITS OWN MEMORY AND ITS OWN COPY OF THE GLOBAL DATA STRUCTURES.



CHROME PROCESS MANAGER



Arnaud Weber, Software Engineer

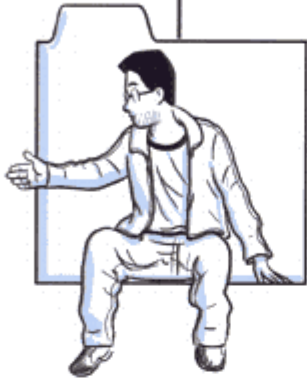
WE'RE APPLYING THE SAME KIND OF PROCESS ISOLATION YOU FIND IN MODERN OPERATING SYSTEMS.

SO, SEPARATE PROCESSES RENDERING SEPARATE TABS.

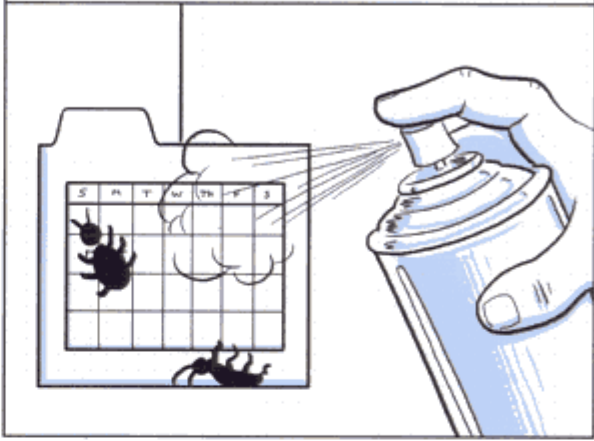


AND NOW YOU HAVE SEPARATE JAVASCRIPT THREADS AS WELL.

ONE TAB CAN BE BUSY, WHILE YOU'RE STILL USING ALL THE OTHERS.



AND IF THERE'S A BROWSER BUG IN THE RENDERER (AND OUR EXPERIENCE IS THAT IT'S ALMOST IMPOSSIBLE TO ELIMINATE ALL BUGS), WE STILL ONLY LOSE THE ONE TAB.



WHEN ONE TAB GOES DOWN YOU GET A 'SAD TAB' BUT IT DOESN'T CRASH THE WHOLE BROWSER.

AND YES, IT REALLY LOOKS LIKE THIS.

A MULTI-PROCESS DESIGN MEANS USING A BIT MORE MEMORY UP FRONT. EACH PROCESS HAS A FIXED ADDITIONAL COST.

BUT OVER TIME, IT WILL ALSO MEAN LESS MEMORY BLOAT.

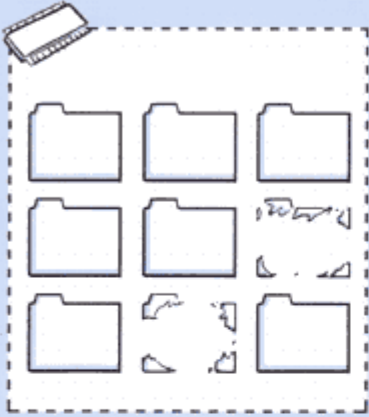
LAUNCH

IN A TRADITIONAL BROWSER, YOU ONLY HAVE ONE PROCESS AND ONE ADDRESS SPACE THAT YOU KEEP LOADING WEB PAGES INTO.

WHEN YOU HAVE TOO MANY TABS OPEN, YOU CAN CLOSE SOME TO FREE UP MEMORY.

WHEN YOU BRING IN ANOTHER TAB, YOU USE THE MEMORY THAT WAS PREVIOUSLY USED.

BUT AS TIME GOES ON, FRAGMENTATION RESULTS -- LITTLE BITS OF MEMORY STILL GET USED EVEN WHEN A TAB GETS CLOSED.

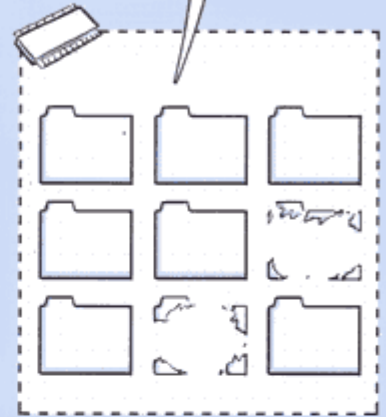


EITHER WE HAVE MEMORY THAT NOTHING CAN REFER TO AGAIN, OR THERE'S A PIECE OF DE-ALLOCATED MEMORY WE STILL HAVE POINTERS TO.



Mike Belshe,
Software Engineer

SO WHEN THE BROWSER WANTS TO OPEN A NEW TAB, IT CAN'T FIT IT IN THE EXISTING SPACE --



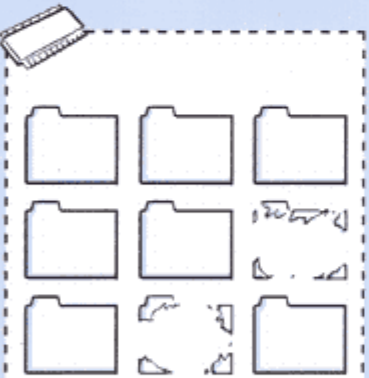
AND THIS PROBLEM GROWS ALL DAY, AS THE LIFETIME OF THE BROWSER EXTENDS.

HURRY UP, DAMMIT!

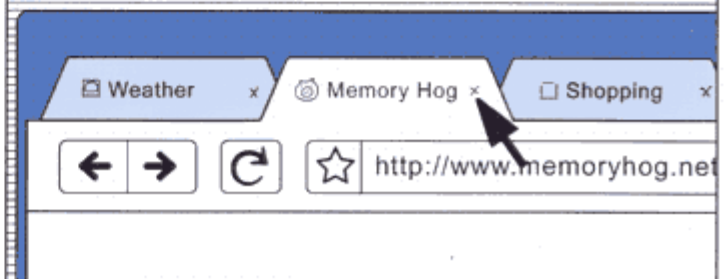
TRY CLOSING SOME TABS.

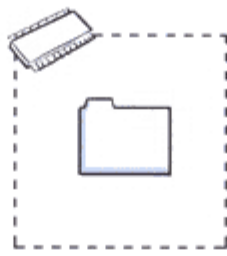
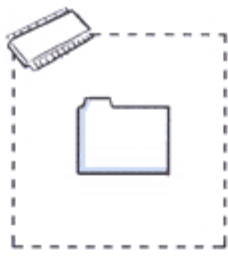
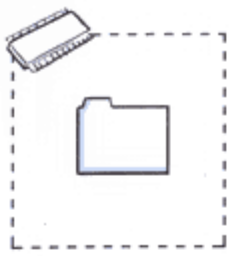


-- AND SO THE OS HAS TO GROW THE BROWSER'S ADDRESS SPACE.

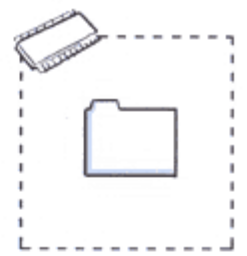


BUT WHEN A TAB IS CLOSED IN GOOGLE CHROME, YOU'RE ENDING THE WHOLE PROCESS --

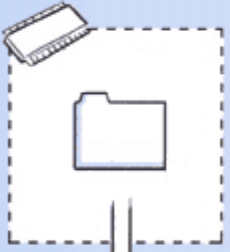




-- AND ALL THAT MEMORY GETS RECLAIMED.



OPEN A NEW TAB NOW, AND YOU'RE STARTING FROM SCRATCH.



SO AS YOU BROWSE, WE'RE CREATING AND DESTROYING PROCESSES ALL THE TIME. IF THERE'S A CRAZY MEMORY LEAK IT WON'T AFFECT YOU FOR THAT LONG BECAUSE YOU'LL PROBABLY CLOSE THE TAB AT SOME POINT AND GET THAT MEMORY BACK.



Brett Wilson, Software Engineer

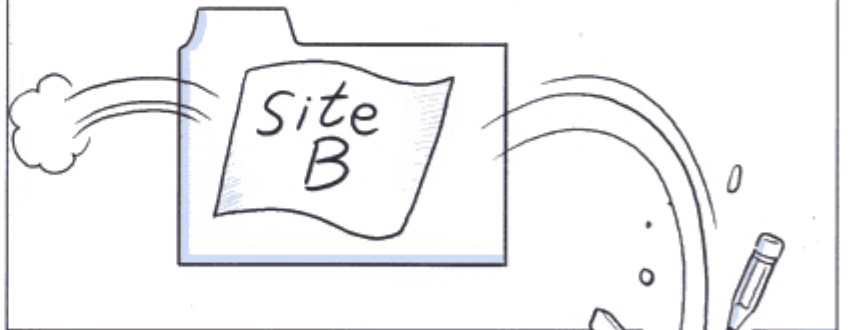
AND WE'RE TAKING IT ONE STEP FURTHER. SUPPOSE YOU NAVIGATE FROM DOMAIN A TO DOMAIN B. THERE'S NO NEED FOR ANY RELATIONSHIP BETWEEN THE TWO SITES --

Site A

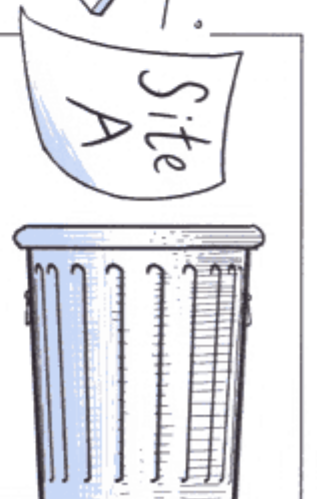
→ Site B



-- SO NOW WE CAN THROW AWAY THE OLD RENDERING ENGINE, THE OLD DATA STRUCTURES, THE OLD PROCESS.



SO, EVEN WITHIN A TAB, WE CAN BE COLLECTING AND TOSsing OUT THE GARBAGE, RECYCLING THE WHOLE PROCESS.



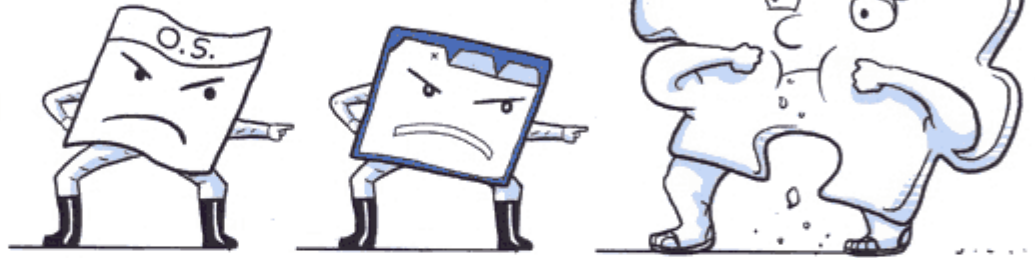
AND JUST LIKE WITH YOUR OS, YOU CAN LOOK UNDER THE HOOD WITH GOOGLE CHROME'S TASK MANAGER TO SEE WHAT SITES ARE USING THE MOST MEMORY, DOWNLOADING THE MOST BYTES, AND ABUSING YOUR CPU.

WHY IS THIS APPLICATION DOWNLOADING THE ENTIRE INTERNET?

YOU CAN EVEN SEE PLUG-INS WITHIN THE TAB, SINCE THEY APPEAR IN CHROME'S TASK MANAGER AS SEPARATE PROCESSES.

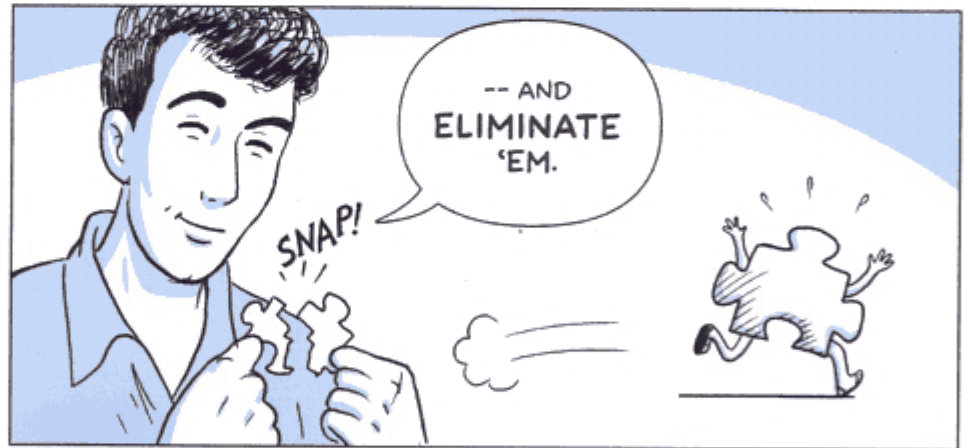
	Memory	CPU	Network
nnivore	74,000K	0	0
nes from Mail	0	0	0
ogle.com	14,768K	0	0
y 2008	0	0	0
s - All items	17,200K	1	0
adsheets	0	0	0

SO, WHEN THINGS START FREAKING OUT, YOU'LL FINALLY HAVE SOME INSIGHT INTO WHO'S MISBEHAVING AND WHY --

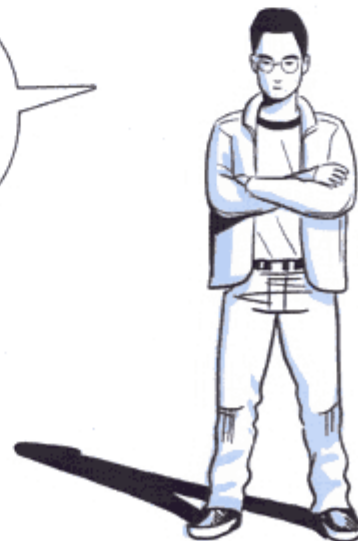


-- AND ELIMINATE 'EM.

SNAP!



PLACING BLAME WHERE BLAME BELONGS.



	Memory	CPU	Network
nnivore	74,000K	0	0
nes from Mail	0	0	0
ogle.com	14,768K	0	0
y 2008	0	0	0
s - All items	17,200K	1	0
adsheets	0	0	0



GOOGLE CHROME IS A MASSIVE, COMPLICATED PRODUCT THAT WILL NEED TO LOAD BILLIONS OF DIFFERENT WEB PAGES, SO TESTING IS CRITICAL.

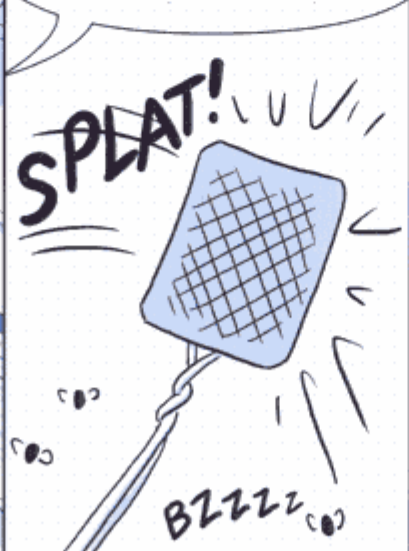
Huan Ren, Software Engineer

FORTUNATELY, HERE AT GOOGLE, WE HAVE AN EQUALLY MASSIVE INFRASTRUCTURE FOR CRAWLING WEB PAGES.

WITHIN 20-30 MINUTES OF EACH NEW BROWSER BUILD, WE CAN TEST IT ON TENS OF THOUSANDS OF DIFFERENT WEB PAGES.

EACH WEEK, "CHROME BOT" TESTS MILLIONS OF PAGES, GIVING OUR DEVELOPERS EARLY RESULTS THEY'D OTHERWISE HAVE TO WAIT UNTIL EXTERNAL BETA FOR.

THE KEY IS CATCHING PROBLEMS AS EARLY AS POSSIBLE. IT IS LESS EXPENSIVE AND EASIER TO FIX THEM RIGHT AWAY. AFTER A FEW DAYS IT IS HARDER TO TRACK THEM DOWN.



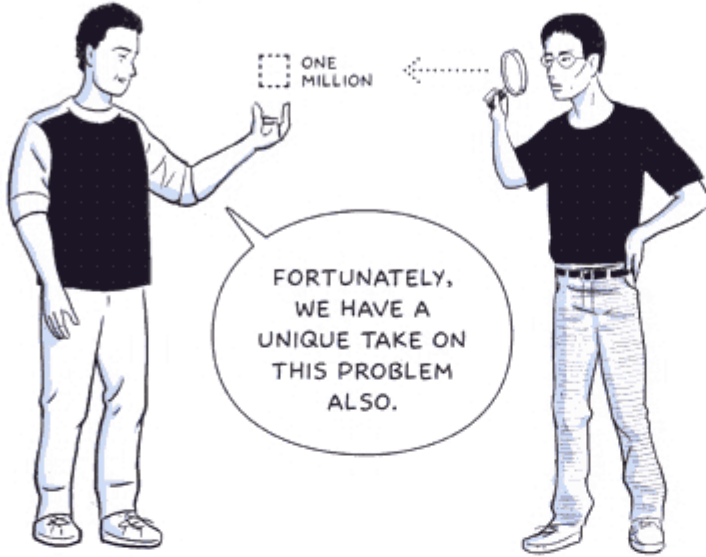
AND CATCHING THEM EARLY HELPS ENGINEERS WRITE BETTER CODE. THEY SAY, "OH, THIS MISTAKE IS PART OF A PATTERN" AND THE NEXT TIME, THEY'RE LESS LIKELY TO MAKE IT.



Erik Kay, Software Engineer

ONE BILLION

OF COURSE, THERE ARE BILLIONS, MAYBE TRILLIONS OF WEBPAGES OUT THERE. IF EACH BUILD IS TESTED AGAINST A MILLION SITES, WHICH MILLION DO WE USE?



FORTUNATELY, WE HAVE A UNIQUE TAKE ON THIS PROBLEM ALSO.

Web Images

We already rank pages based on which pages the average user is most likely to visit.

www.alreadyrank.com - Similar Pages

At the very least, we'll make sure we won't be broken on the kinds of sites people use on a day-to-day basis.

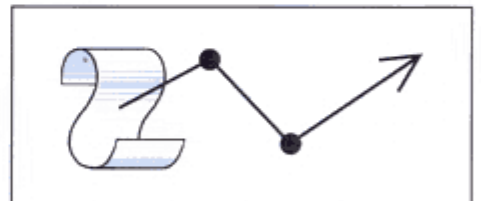
www.attheveryleast.com - Similar Pages



THERE ARE SEVERAL WAYS WE TEST EACH CHECK-IN, FROM UNIT TESTS OF INDIVIDUAL PIECES OF CODE --

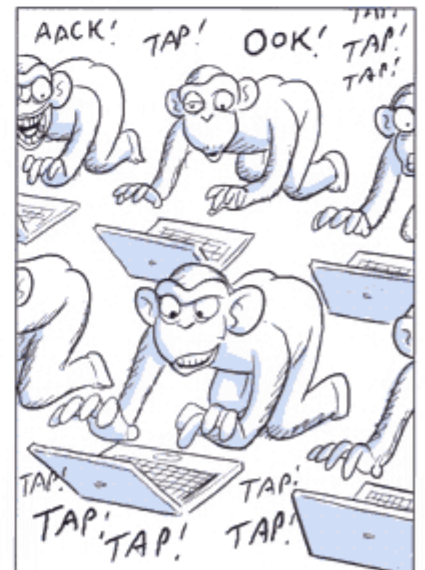


-- TO AUTOMATED UI TESTING OF SCRIPTED USER ACTIONS LIKE "CLICKED BACK BUTTON... WENT TO PAGE..." --

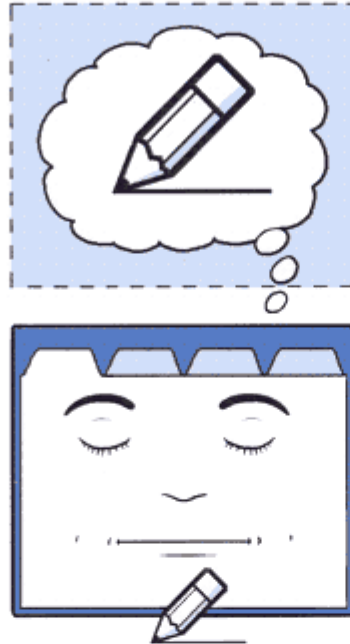


-- TO FUZZ TESTING: SENDING YOUR APPLICATION RANDOM INPUT.

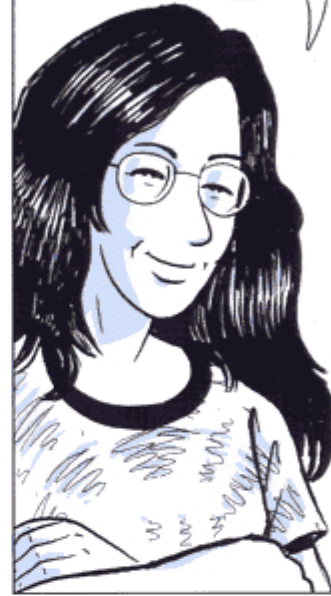
Pam Greene, Software Engineer



IN LAYOUT TESTING, WEBKIT FOUND THAT PRODUCING A SCHEMATIC OF WHAT THE BROWSER THINKS IT'S DISPLAYING IS A MORE PRECISE WAY TO COMPARE LAYOUTS THAN TAKING SCREENSHOTS AND CREATING A CRYPTOGRAPHIC HASH.



WHEN WE STARTED WE WERE PASSING 23% OF WEBKIT'S LAYOUT TESTS. MOVING FROM THERE TO 99% HAS BEEN A FUN CHALLENGE AND AN INTERESTING EXAMPLE OF TEST-DRIVEN DESIGN.



THERE ARE LIMITS TO WHAT WE CAN DO WITH AUTOMATED TESTING.



Humans Only

Name

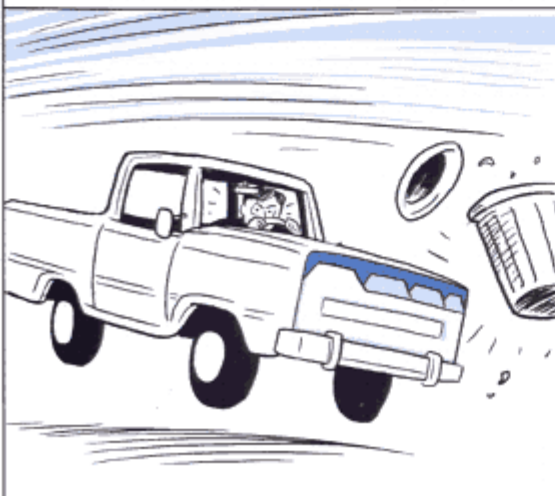


●●●●●●

Password

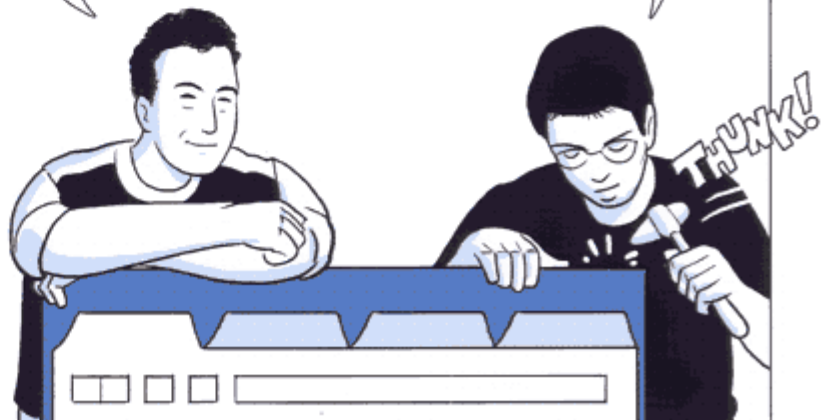
WE CAN'T TEST WEBSITES THAT REQUIRE A PASSWORD, FOR EXAMPLE.

AND IT'S NOT THE SAME AS A HUMAN BEING WALKING AROUND AND MISUSING THINGS. WE ARE USING THE BROWSER IN THE WAY WE'VE DESIGNED IT TO BE USED.



IT'S HARD TO COVER 100%, BUT THAT'S WHAT WE'RE TRYING TO DO.

I DON'T CARE IF THERE'S ONE FEWER COOL FEATURE. I JUST WANT THIS PRODUCT TO BE ROCK SOLID.



Part Two



Speed: WebKit and V8



WEBKIT IS THE OPEN SOURCE RENDERING ENGINE WE USED FOR GOOGLE CHROME.



WE WERE IMPRESSED BY HOW FAST IT IS.



WE ALSO KNEW THERE WAS A TEAM AT GOOGLE WORKING ON ANDROID AND WE ASKED THEM, "WHY DID YOU GUYS USE WEBKIT?"



THEY SAID IT USES MEMORY EFFICIENTLY, WAS EASILY ADAPTED TO EMBEDDED DEVICES, AND IT WAS EASY FOR NEW BROWSER DEVELOPERS TO LEARN TO MAKE THE CODE BASE WORK.



BROWSERS ARE COMPLEX. ONE OF THE THINGS DONE WELL WITH WEBKIT IS THAT IT'S KEPT SIMPLE.



BECAUSE JAVASCRIPT IS SO IMPORTANT TO THE WEB TODAY --



-- WE DECIDED IT WAS IMPORTANT TO WORK ON BUILDING A JAVASCRIPT VIRTUAL MACHINE --



-- WHICH IS EXACTLY WHAT THE V8 TEAM IN DENMARK DID.

THE V8 TEAM ARE EXPERTS AT VIRTUAL MACHINES. WHATEVER LANGUAGE YOU WANT TO PUT INTO A VM, THEY CAN TELL YOU HOW TO WRITE IT.



VIRTUAL MACHINES PROVIDE SAFETY AND PLATFORM INDEPENDENCE.

BUT PREVIOUS VIRTUAL MACHINES FOR JAVASCRIPT WERE DESIGNED FOR SMALL PROGRAMS, WHERE THE PERFORMANCE AND INTERACTIVITY OF THE SYSTEM WEREN'T THAT IMPORTANT.

THEY JUST WANTED TO RUN SOME VERY BASIC STUFF ON A WEBPAGE.



Lars Bak,
Software
Engineer, V8



Kasper
Lund,
Software
Engineer, V8



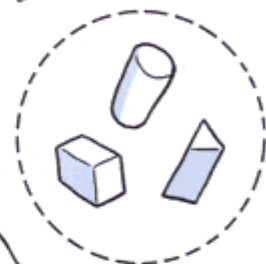
BUT NOW,
YOU HAVE WEB APPLICATIONS LIKE
GMAIL THAT ARE USING THE WEB
BROWSER TO ITS FULLEST WHEN IT COMES
TO DOM MANIPULATIONS AND
JAVASCRIPT --

-- AND THAT
SIMPLISTIC APPROACH TO
JAVASCRIPT ENGINES ISN'T
ENOUGH ANYMORE.

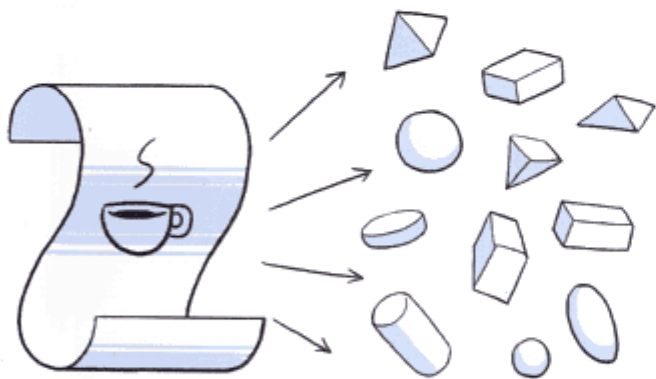


SO WE STARTED WITH
NO CODE, JUST SOME WILD
IDEAS ABOUT HOW TO MAKE
IT GO REALLY FAST --

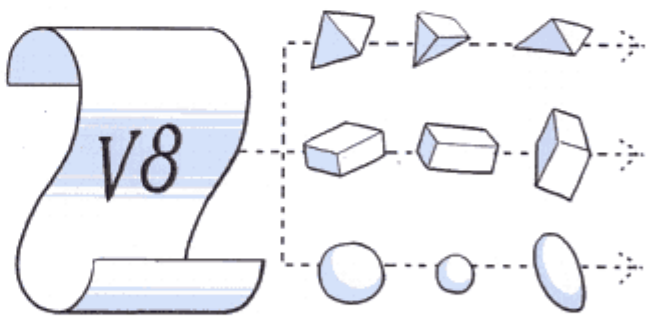
-- SUCH AS
INTRODUCING
HIDDEN CLASS
TRANSITIONS.



JAVASCRIPT ITSELF IS **CLASSLESS**.
YOU CAN CREATE A NEW OBJECT,
DYNAMICALLY ADD PROPERTIES TO
IT AND GO ON.



BUT IN **V8**, AS EXECUTION GOES ON,
OBJECTS THAT END UP WITH THE SAME
PROPERTIES WILL SHARE THE SAME HIDDEN
CLASS AND WE CAN START APPLYING
DYNAMIC OPTIMIZATIONS BASED ON THAT.

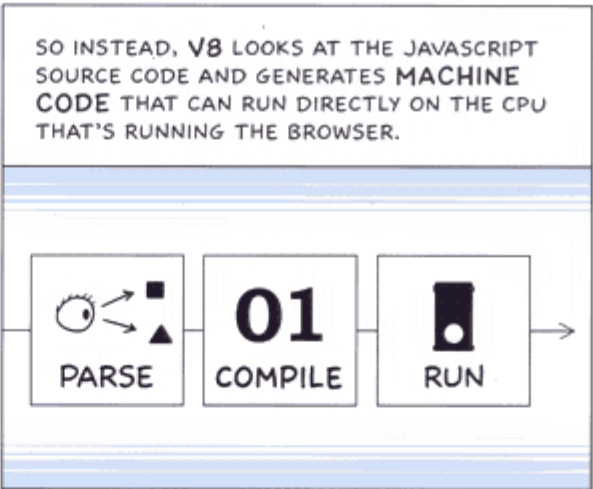
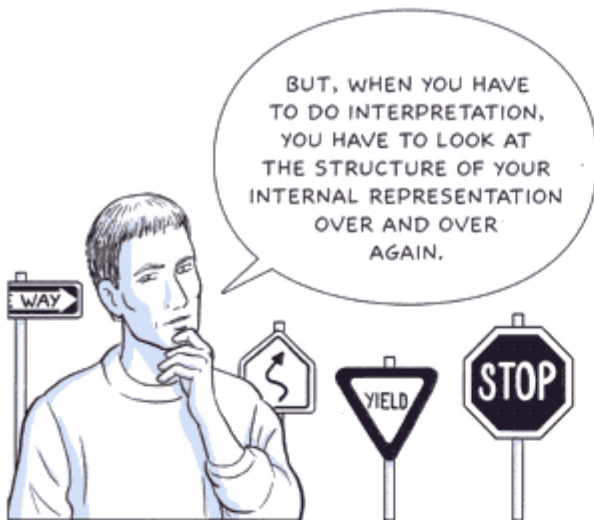


ANOTHER FACTOR
IN V8'S SPEED IS
DYNAMIC CODE
GENERATION.



WHEN OTHER JAVASCRIPT ENGINES RUN, THEY
LOOK AT THE JAVASCRIPT SOURCE CODE AND
GENERATE AN INTERNAL REPRESENTATION OF IT
THEY CAN INTERPRET.

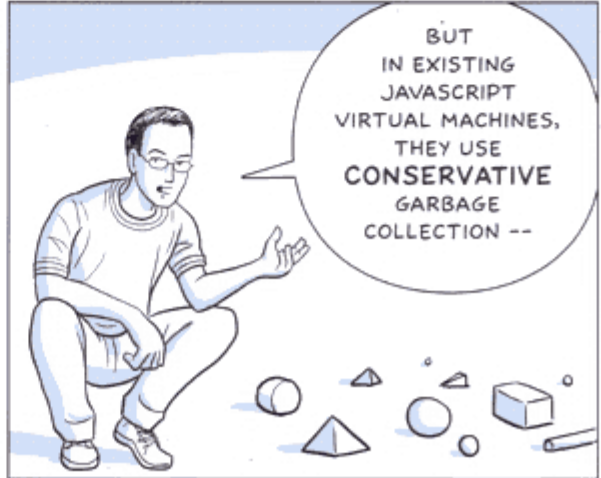
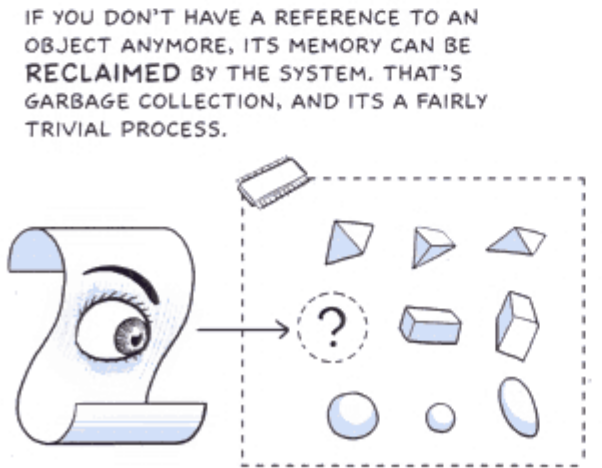
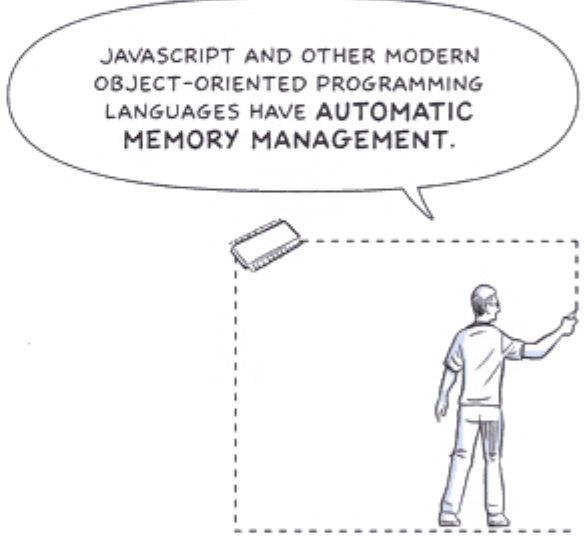
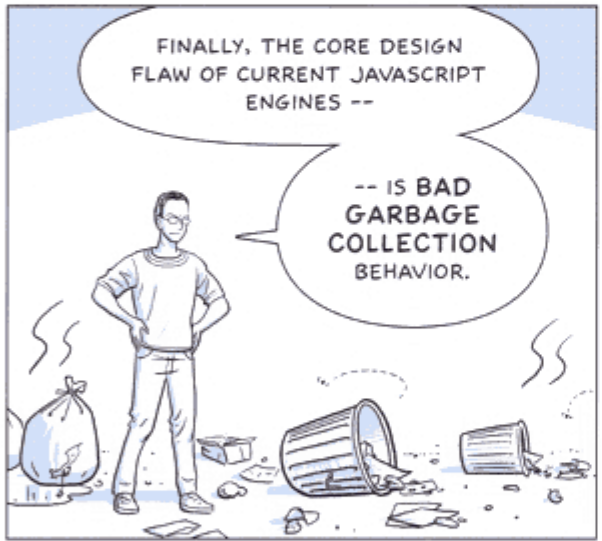




WHEN YOU INTERPRET ONCE AND COMPILE MACHINE CODE, THEN THAT CODE *IS* YOUR REPRESENTATION OF THE JAVASCRIPT SOURCE CODE AND IT DOESN'T NEED TO BE INTERPRETED, IT JUST RUNS.



101010001010001010100101010000101010000101010000101010



-- WHICH MEANS THAT BECAUSE YOU DON'T KNOW EXACTLY WHERE ALL THE POINTERS ARE --



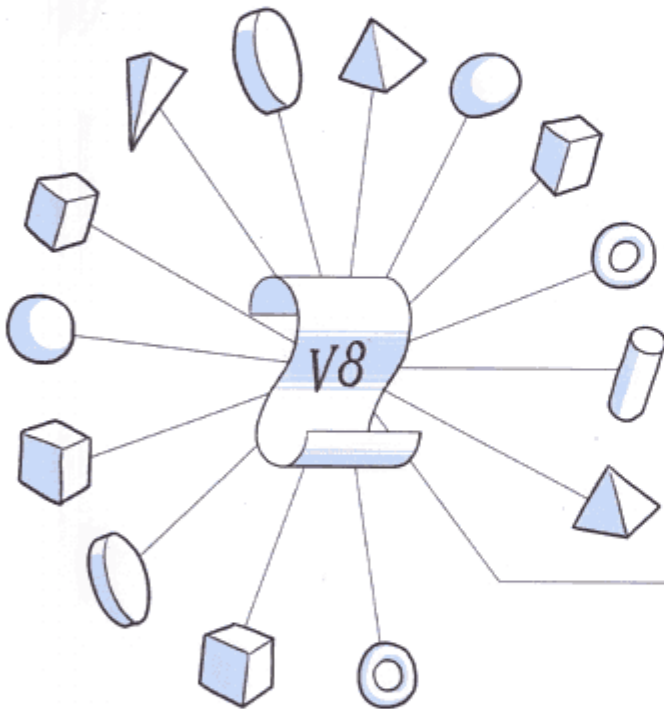
-- YOU START SEARCHING THROUGH THE EXECUTION STACK TO SEE WHICH WORDS LOOK LIKE POINTERS.

BUT THE ONES THAT SORT OF LOOK LIKE POINTERS COULD ALSO BE INTEGERS THAT JUST HAPPEN TO HAVE THE SAME ADDRESS AS AN OBJECT IN THE OBJECT HEAP.



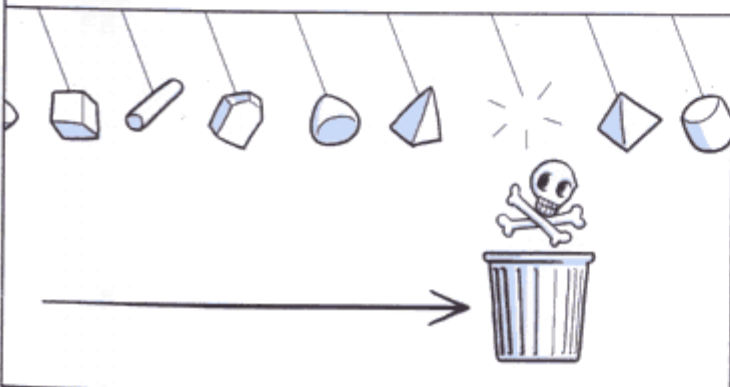
IN V8, WE ARE USING **PRECISE** GARBAGE COLLECTION, SO WE KNOW PRECISELY WHERE **ALL** OF THE POINTERS ARE ON THE STACK AND THIS GIVES US SEVERAL ADVANTAGES.

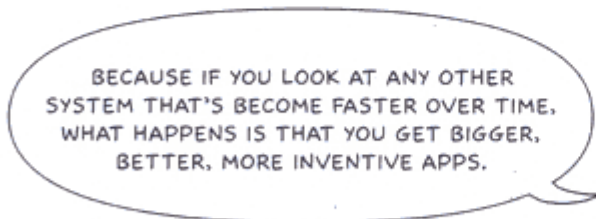
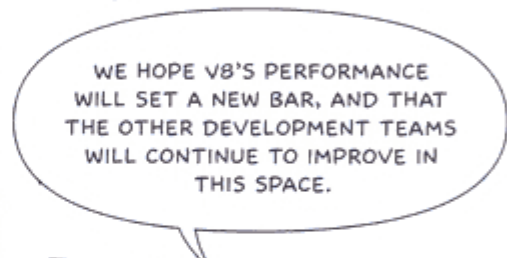
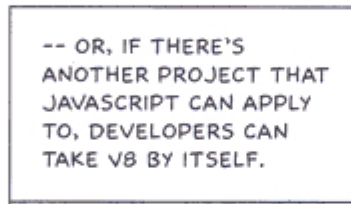
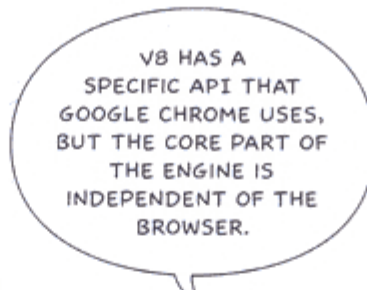
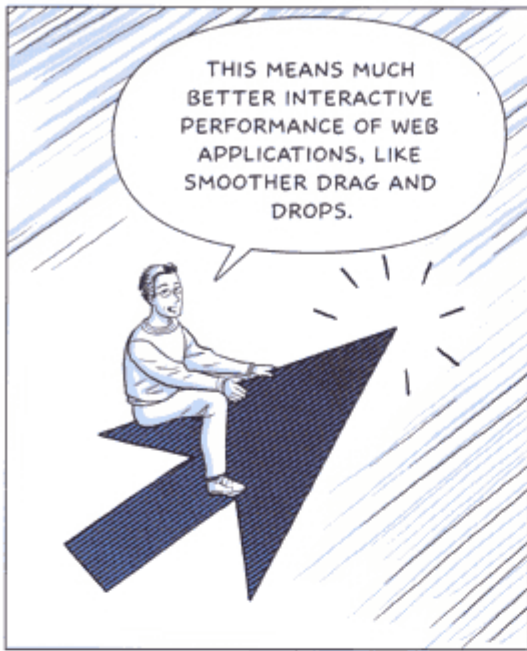
ONE IS THAT WE CAN MIGRATE AN OBJECT TO ANOTHER PLACE AND JUST REWIRE THE POINTER.



AND, BECAUSE WE KNOW PRECISELY WHERE ALL THE POINTERS ARE, WE CAN ALSO IMPLEMENT **INCREMENTAL** GARBAGE COLLECTION.

MEANING QUICK GARBAGE COLLECTION ROUND-TRIPS THAT ARE CLOSE TO A FEW MILLISECONDS, COMPARED TO PROCESSING ALL 100MB OF DATA WHICH COULD CAUSE SECOND-LONG PAUSES.





Part Three



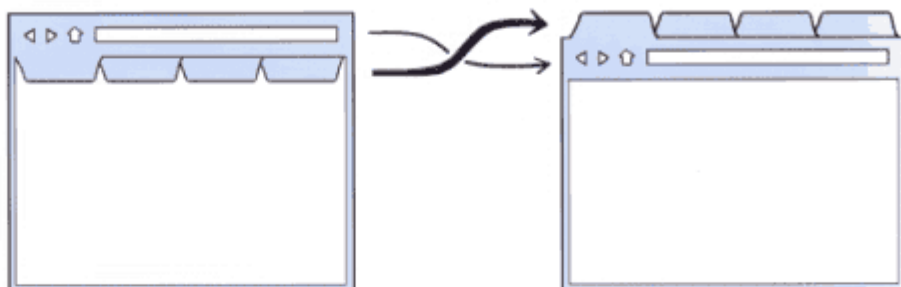
Search and the User Experience



IN GOOGLE CHROME, THE PRIMARY PIECE OF THE USER INTERFACE IS THE TAB.



Ben Goodger,
Software Engineer

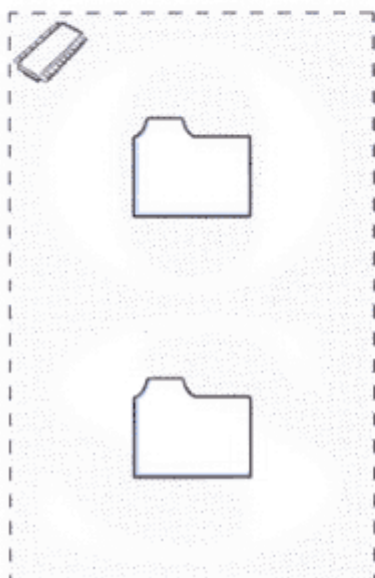


AS SOON AS WE STARTED THINKING ABOUT IT THAT WAY, THE DESIGN NATURALLY FOLLOWED.



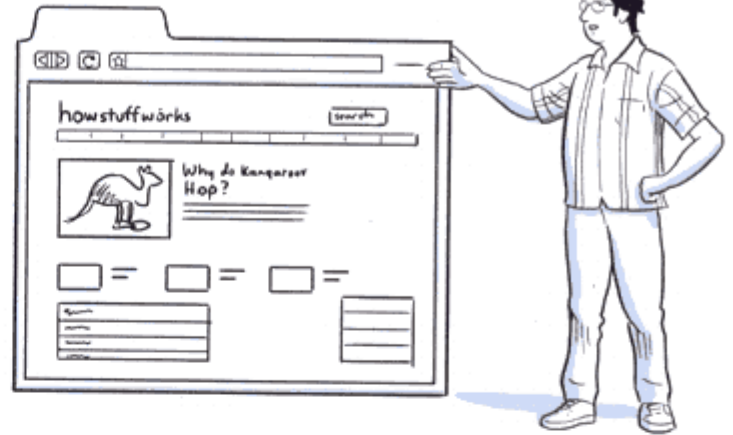
WE BEGAN REBUILDING THE UI SO THE TABS WERE ON TOP.

WE COULD DETACH THE TABS EASILY BECAUSE OF THE SEPARATION OF THE BROWSER AND TAB PROCESSES.



AND BECAUSE THE TABS ARE THE MOST IMPORTANT PART OF THE UI, EACH TAB HAS ITS OWN CONTROLS.

AND ITS OWN URL BOX.

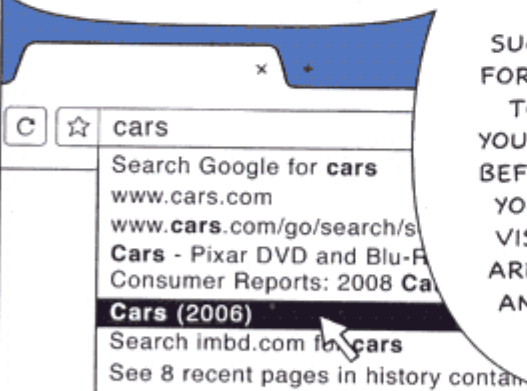


WHICH AROUND HERE WE'VE BEEN CALLING THE 'OMNIBOX.'



THE OMNIBOX HANDLES FAR MORE THAN JUST URLS.

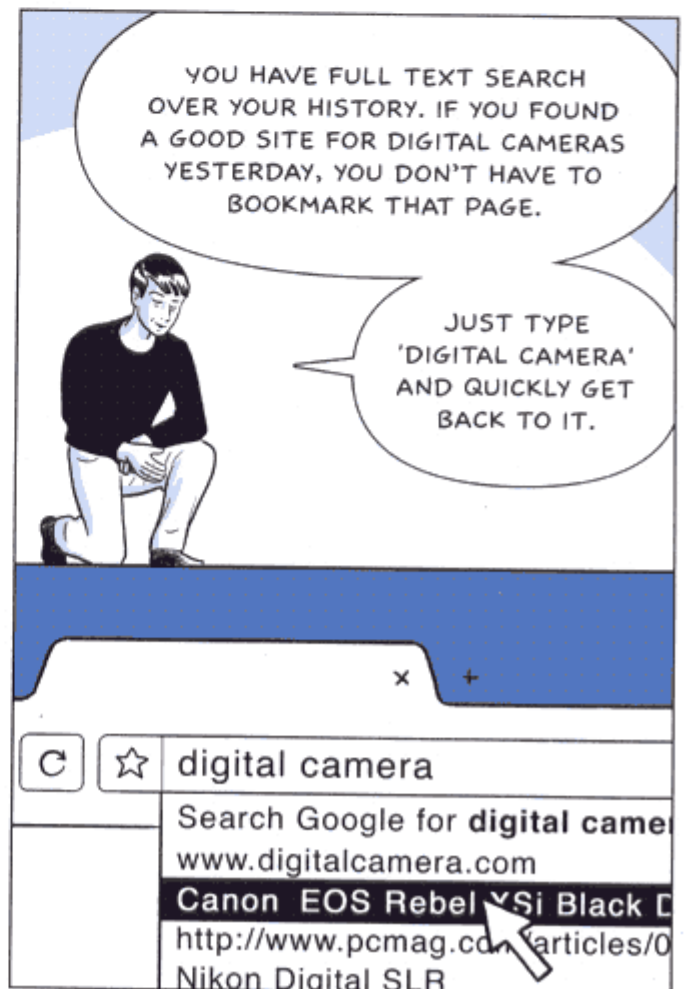
IT ALSO OFFERS SUGGESTIONS FOR SEARCHES, TOP PAGES YOU'VE VISITED BEFORE, PAGES YOU HAVEN'T VISITED BUT ARE POPULAR AND MORE...



Glen Murphy, Software Engineer

YOU HAVE FULL TEXT SEARCH OVER YOUR HISTORY. IF YOU FOUND A GOOD SITE FOR DIGITAL CAMERAS YESTERDAY, YOU DON'T HAVE TO BOOKMARK THAT PAGE.

JUST TYPE 'DIGITAL CAMERA' AND QUICKLY GET BACK TO IT.



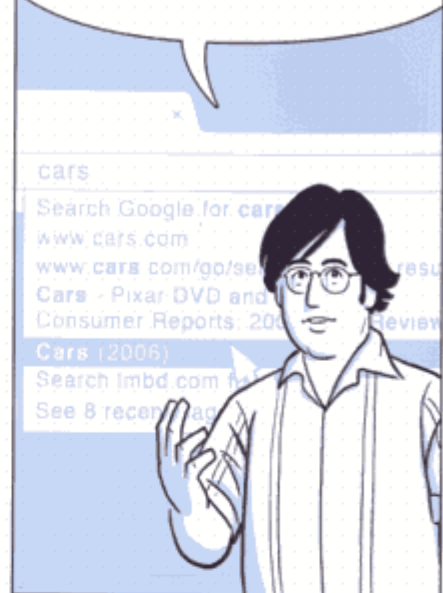
WHEN THE TEAM SUGGESTED AUTOCOMPLETION IN LINE, I SAID I HATED IT WHEN BROWSERS STICK ALL THIS CRAP INTO A LOCATION BAR AS I'M TYPING. IT'S NEVER WHAT I WANT.



BUT, THEY SAID, NO, NO, IT'LL BE FINE, TRUST US -- AND THEY WENT ON AND MADE IT SOMETHING REALLY COMPELLING...



INLINE COMPLETIONS WILL NEVER FLICKER, NEVER FLASH. IT'S PERFECTLY, AESTHETICALLY NON-DISTRACTING.

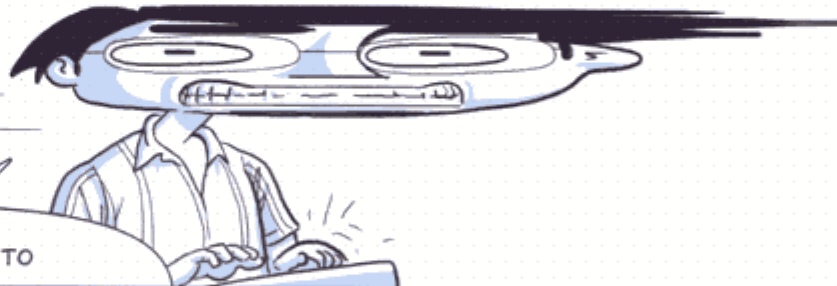


PLUS, IT'LL ONLY AUTOCOMPLETE TO SOMETHING YOU'VE EXPLICITLY TYPED BEFORE.

TYPE

AND YOU MIGHT GO STRAIGHT TO

--



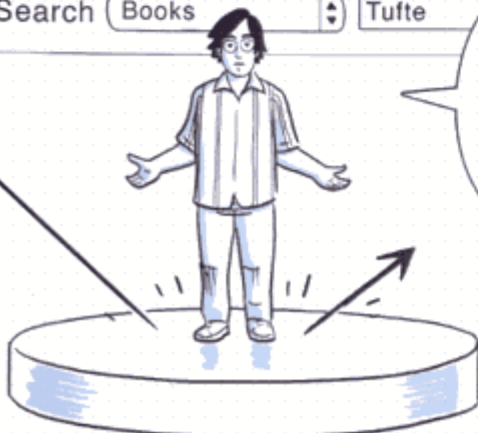
-- BUT NEVER TO

cnn.com/2008/politics/07/27/campaign.wrap/index.html?iref=mpstor

AND WHEN YOU SEARCH ON SITES LIKE AMAZON, WIKIPEDIA OR EVEN GOOGLE --

Search

-- THE SEARCH BOXES ON THOSE PAGES ARE CAPTURED ON YOUR LOCAL SYSTEM --



-- SO YOU CAN SEARCH THOSE SAME SITES WITH DIFFERENT TERMS LATER ON, STRAIGHT FROM THE ADDRESS BAR, BY STARTING THE SITE'S NAME AND PRESSING 'TAB'.

Search Amazon:

CLICK!

OPEN A NEW TAB IN MOST BROWSERS TODAY, AND YOU'LL GET YOUR HOMEPAGE.

SOME USERS HAVE A BLANK PAGE BECAUSE IT OPENS QUICKLY.



BUT THE ACTION OF OPENING A TAB IS A STATEMENT OF INTENT: YOU WANT TO GO SOMEPLACE!



MAYBE YOU KNOW WHERE. MAYBE YOU DON'T KNOW AND NEED TO SEARCH.

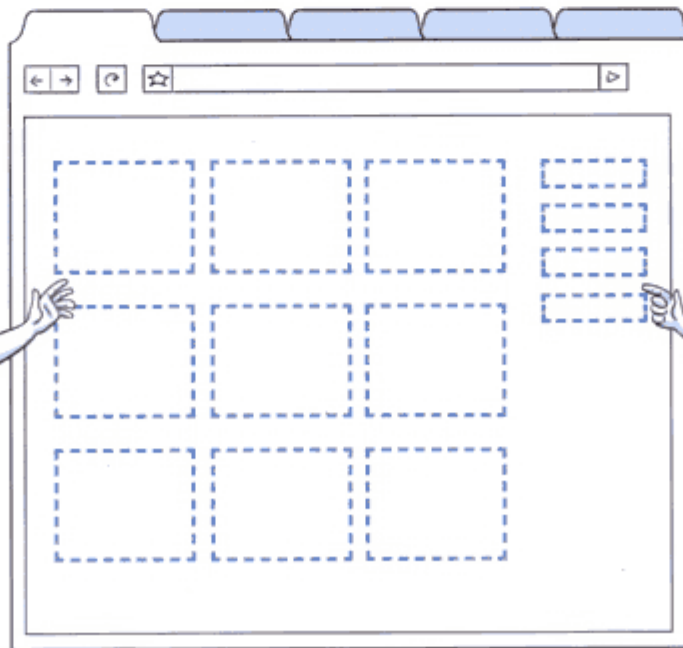


WE'RE GOING TO SHOW A PAGE THAT IS DESIGNED TO BE FAST, BUT ALSO HELPS YOU COMPLETE THAT ACTION.

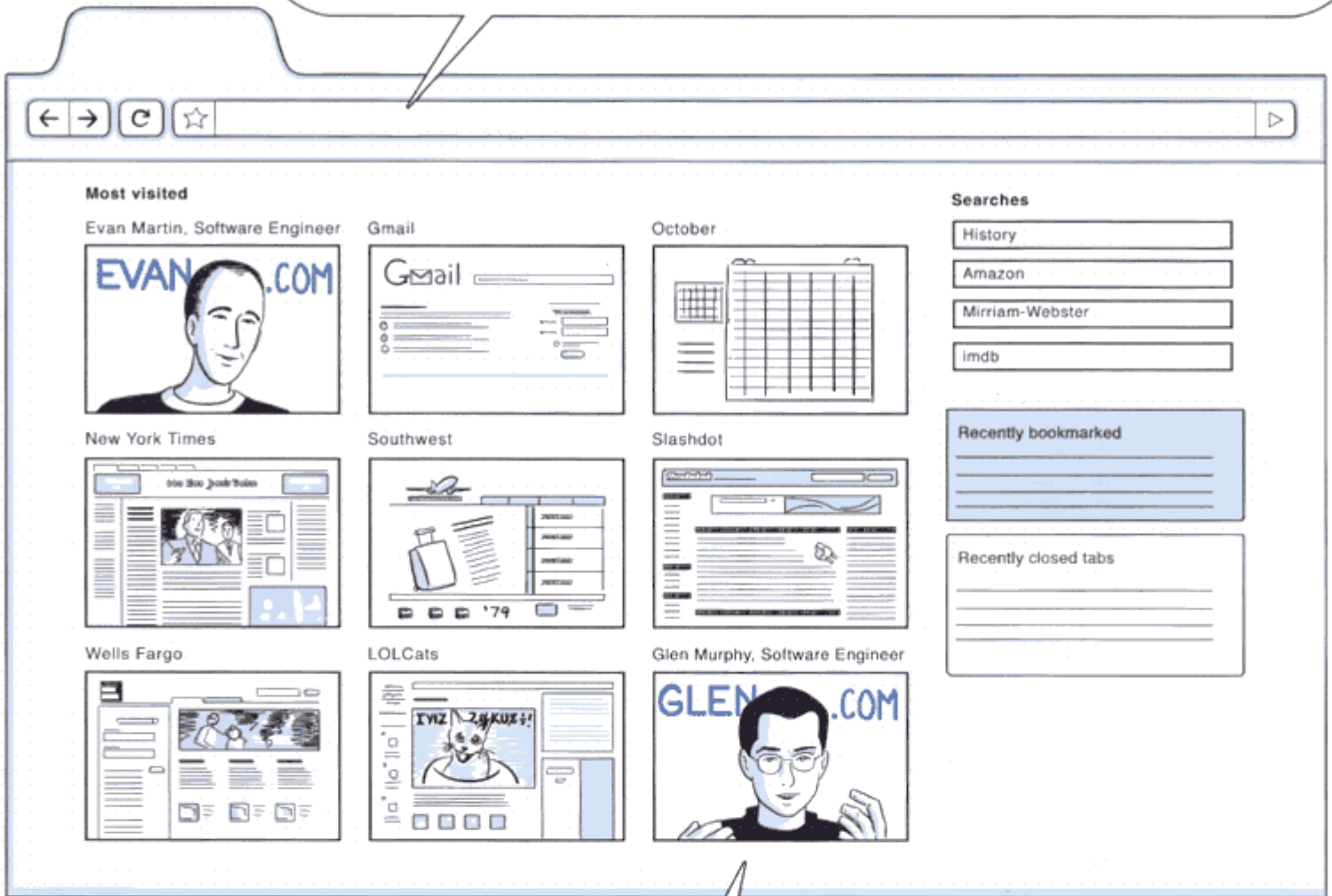


OUR DEFAULT EXPERIENCE, THEN, IS THE NEW TAB PAGE WITH YOUR NINE MOST VISITED PAGES HERE --

-- AND THE SITES YOU SEARCH ON MOST HERE.



IT'S THE PAGES YOU WERE GOING TO TYPE INTO THE URL BOX ANYWAY. GOOGLE CHROME USES YOUR BEHAVIOR IN THE OMNIBOX TO FEED INTO THAT PAGE.



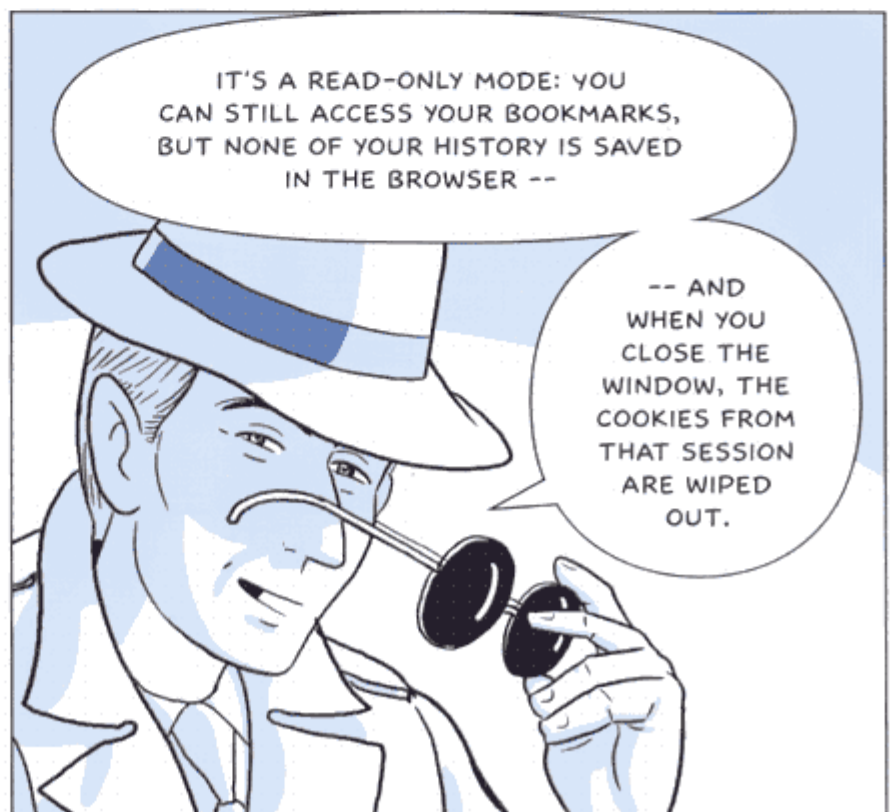
YOU MIGHT OPEN IT AND BE, LIKE, WHAT'S ALL MY STUFF DOING HERE? BUT AFTER A WHILE, YOU SEE THIS PAGE AND IT'S JUST YOU, IT'S **YOUR** BROWSER.

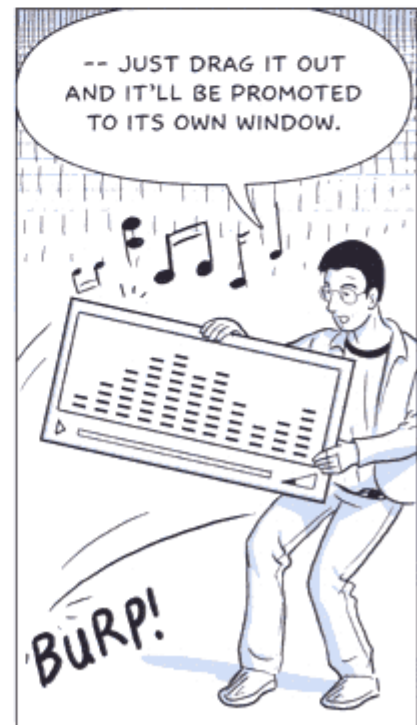
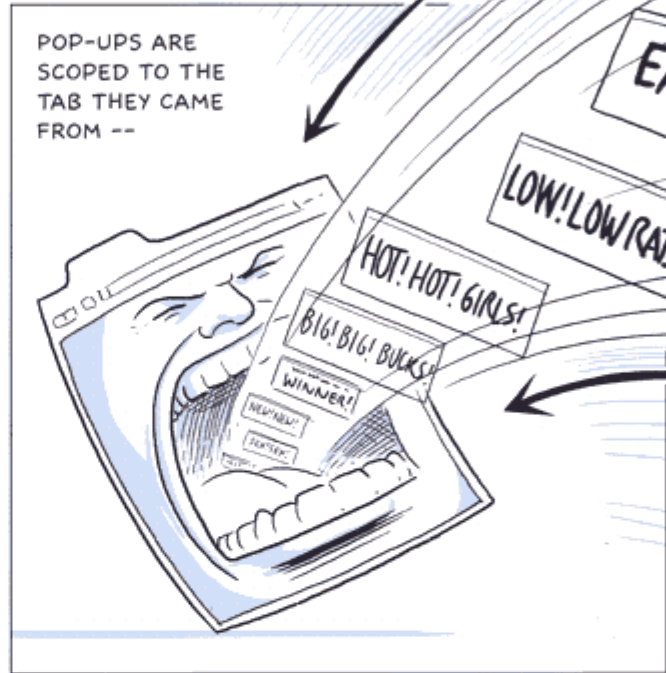
GOOGLE CHROME HAS A PRIVACY MODE. YOU CAN CREATE AN 'INCOGNITO' WINDOW AND NOTHING THAT OCCURS IN THAT WINDOW IS EVER LOGGED ON YOUR COMPUTER.

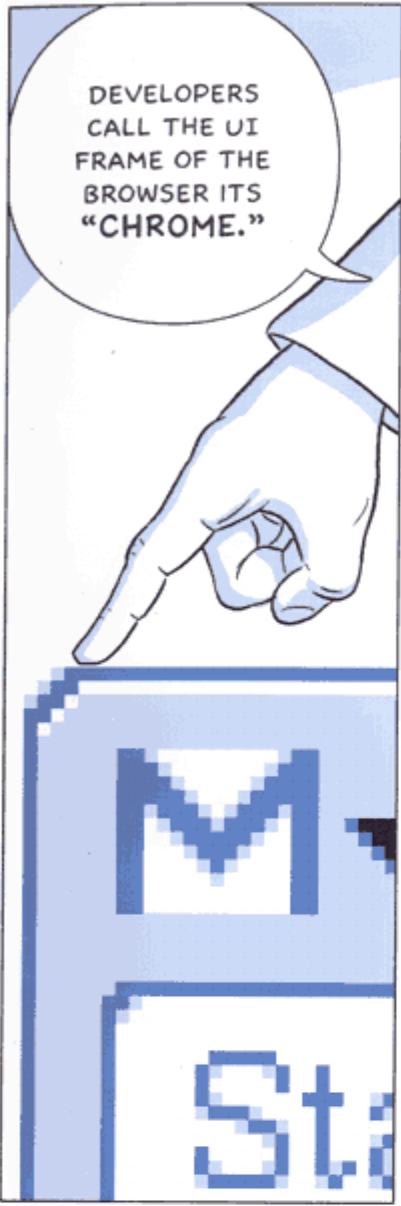


IT'S A READ-ONLY MODE: YOU CAN STILL ACCESS YOUR BOOKMARKS, BUT NONE OF YOUR HISTORY IS SAVED IN THE BROWSER --

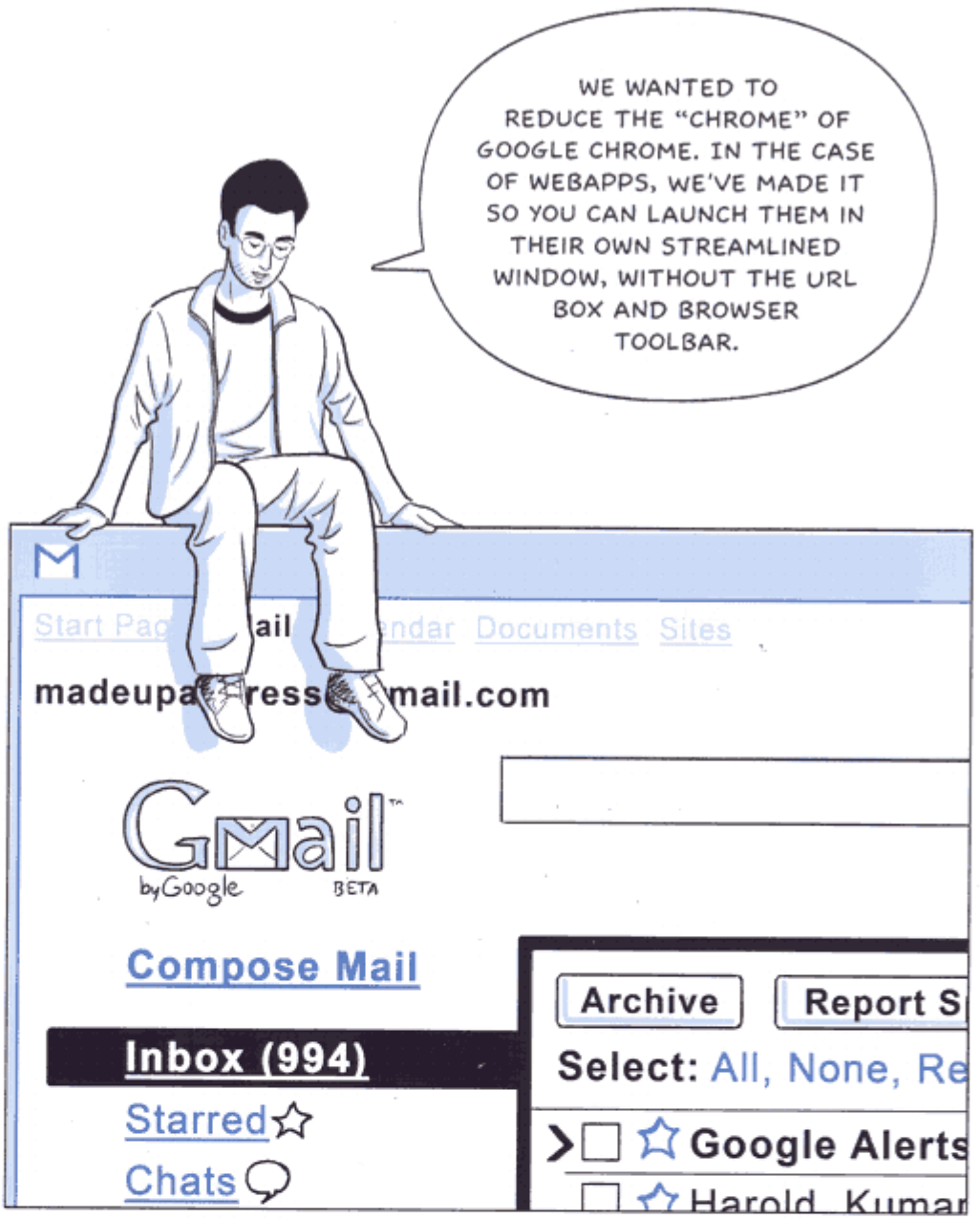
-- AND WHEN YOU CLOSE THE WINDOW, THE COOKIES FROM THAT SESSION ARE WIPED OUT.







DEVELOPERS CALL THE UI FRAME OF THE BROWSER ITS "CHROME."



WE WANTED TO REDUCE THE "CHROME" OF GOOGLE CHROME. IN THE CASE OF WEBAPPS, WE'VE MADE IT SO YOU CAN LAUNCH THEM IN THEIR OWN STREAMLINED WINDOW, WITHOUT THE URL BOX AND BROWSER TOOLBAR.

WE DON'T WANT TO INTERRUPT ANYTHING THE USER IS TRYING TO DO.

IF YOU CAN JUST IGNORE THE BROWSER, WE'VE DONE A GOOD JOB.





MALWARE AND PHISHING ARE A HUGE PROBLEM FOR USERS, AFFECTING TRUST AND CONFIDENCE IN THE WEB.

WHEN WE STARTED THIS PROJECT, IT WAS A VERY DIFFERENT LANDSCAPE FROM WHEN THE OTHER BROWSERS STARTED.



Ian Fette, Product Manager

John Abd-El-Malek, Software Engineer

BACK THEN, IT WAS ABOUT RENDERING THE PAGE AND GETTING THE COOL THINGS WORKING. THERE WAS NO MONETARY INCENTIVE TO PUT MALWARE ON USERS' MACHINES.



NOW, MALWARE IS VERY FINANCIALLY DRIVEN. IT'S ALL ABOUT STEALING PASSWORDS AND MOVING MONEY AROUND.



IN THINKING ABOUT SECURITY, WE BEGAN WITH THE ASSUMPTION THAT YOUR BROWSER WOULD GET COMPROMISED.

YOU WILL EVENTUALLY ENCOUNTER MALWARE.

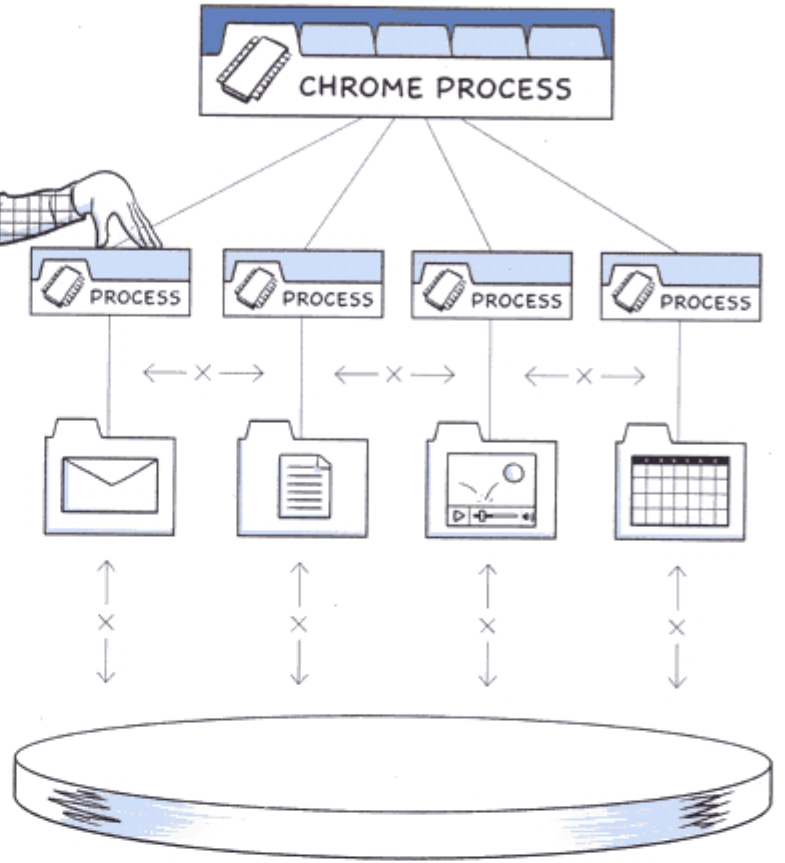


Carlos Pizano, Software Engineer

WITH **SANDBOXING**, OUR GOAL IS TO PREVENT MALWARE FROM INSTALLING ITSELF ON YOUR COMPUTER OR USING WHAT HAPPENS IN ONE TAB TO AFFECT WHAT HAPPENS IN ANOTHER.

SO, FOR EACH OF THESE PROCESSES WE'VE STRIPPED AWAY ALL OF THEIR RIGHTS.

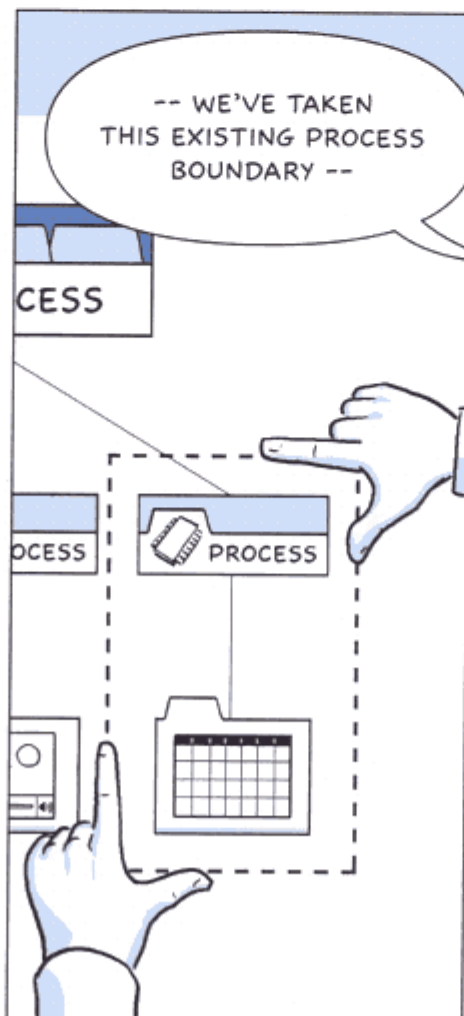
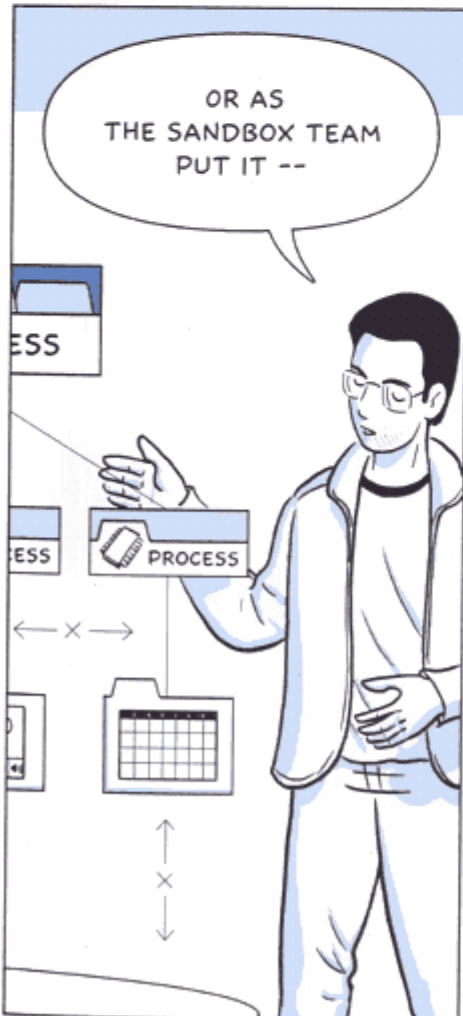
THEY CAN COMPUTE BUT THEY CAN'T WRITE FILES TO YOUR HARD DRIVE OR READ FILES FROM SENSITIVE AREAS LIKE YOUR DOCUMENTS OR DESKTOP.



OR AS THE SANDBOX TEAM PUT IT --

-- WE'VE TAKEN THIS EXISTING PROCESS BOUNDARY --

-- AND MADE IT INTO A **JAIL**.



THAT MEANS NO WATCHING YOU TYPE YOUR CREDIT CARD NUMBER.

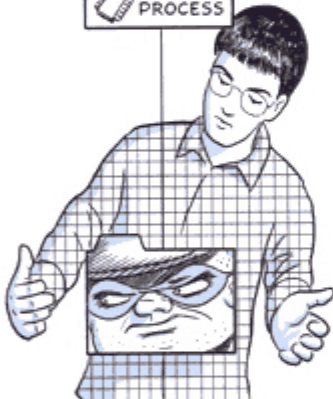
NO INTERACTING WITH MOUSE OPERATIONS.

NO READING YOUR TAX RETURNS.

NO TELLING WINDOWS TO RUN AN EXECUTABLE AT START-UP.



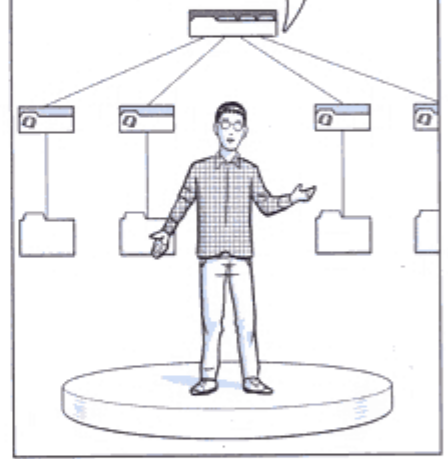
SOMETHING BAD COULD BE RUNNING IN THIS TAB --



-- BUT AS SOON AS YOU CLOSE IT, IT'S GONE.



NO EFFECT ON YOUR MACHINE AND NO EFFECT ON OTHER PROCESSES.



THE PERIMETER OF THE SANDBOX IS LARGELY BASED ON PERMISSIONS.



Mark Larson, Program Manager

VISTA USES A MODIFIED VERSION OF THE BIBA SECURITY MODEL WHICH HAS THREE LEVELS.

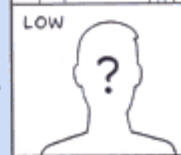
VERY TRUSTED.



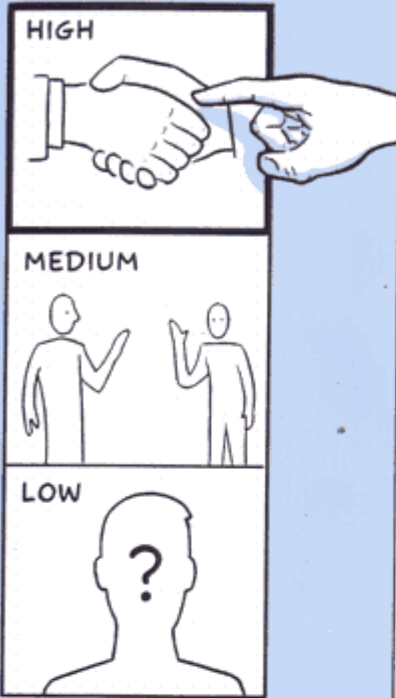
SOMEWHAT TRUSTED.



NOT TRUSTED AT ALL.



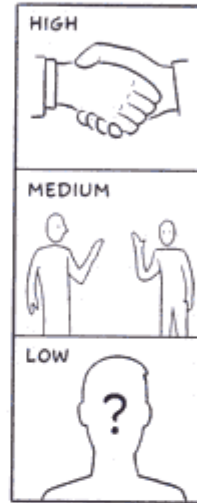
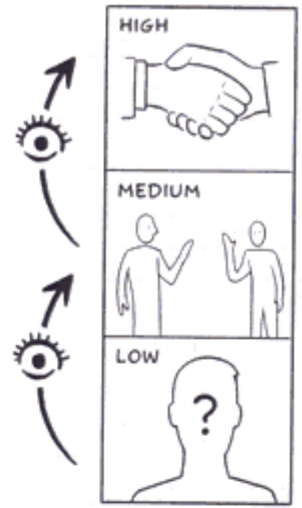
THIS LEVEL IS FOR
BACKUP SYSTEMS,
PROGRAMS THAT
UPDATE, ETC.



THIS LEVEL IS FOR
EVERYTHING THE USER
RUNS NORMALLY.
NOTEPAD, SOLITAIRE,
CALCULATOR...

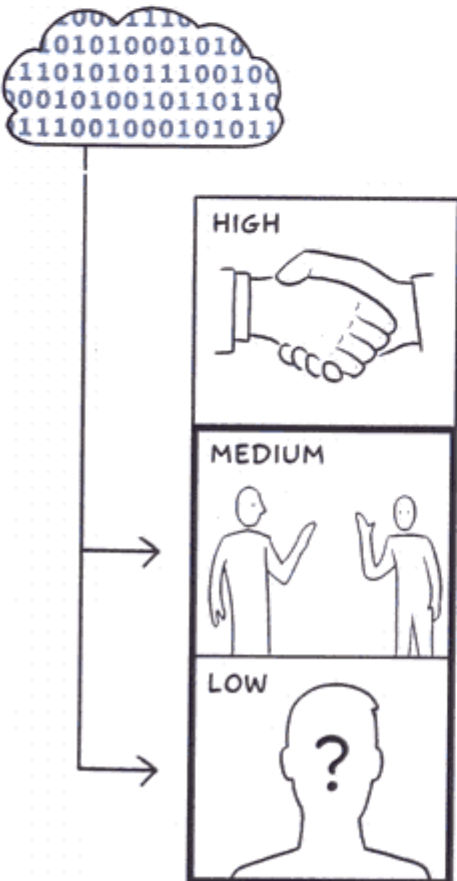


READING
IS
ALLOWED
FROM
LOW
TO
HIGH --

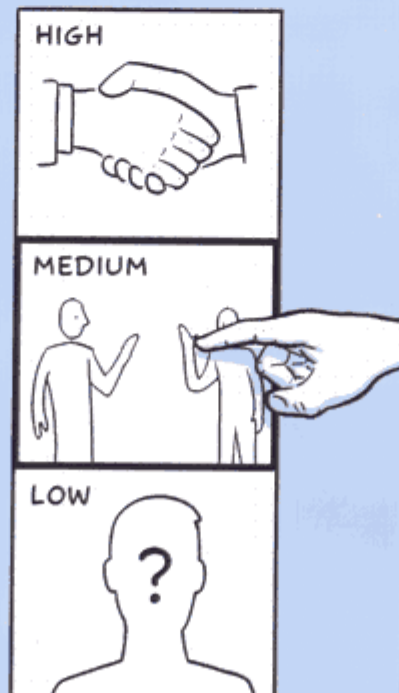


-- BUT
WRITING
IS
ALLOWED
ONLY
FROM
HIGH
TO
LOW.

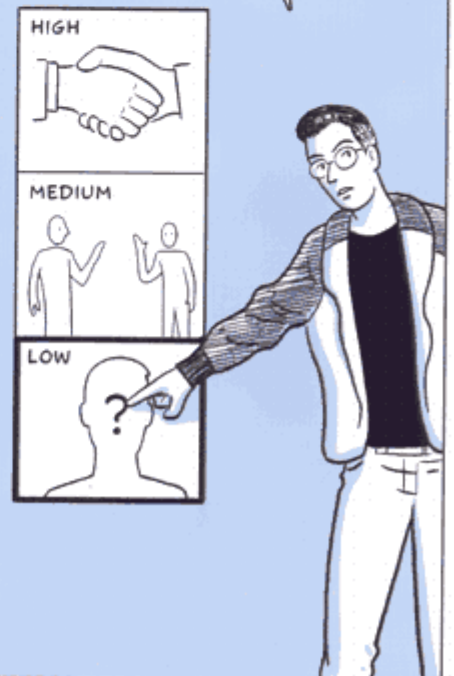
TYPICALLY, APPLICATIONS
RECEIVING AND PROCESSING
DATA FROM THE INTERNET
ARE SPLIT INTO THE TWO
LOWER LEVELS.

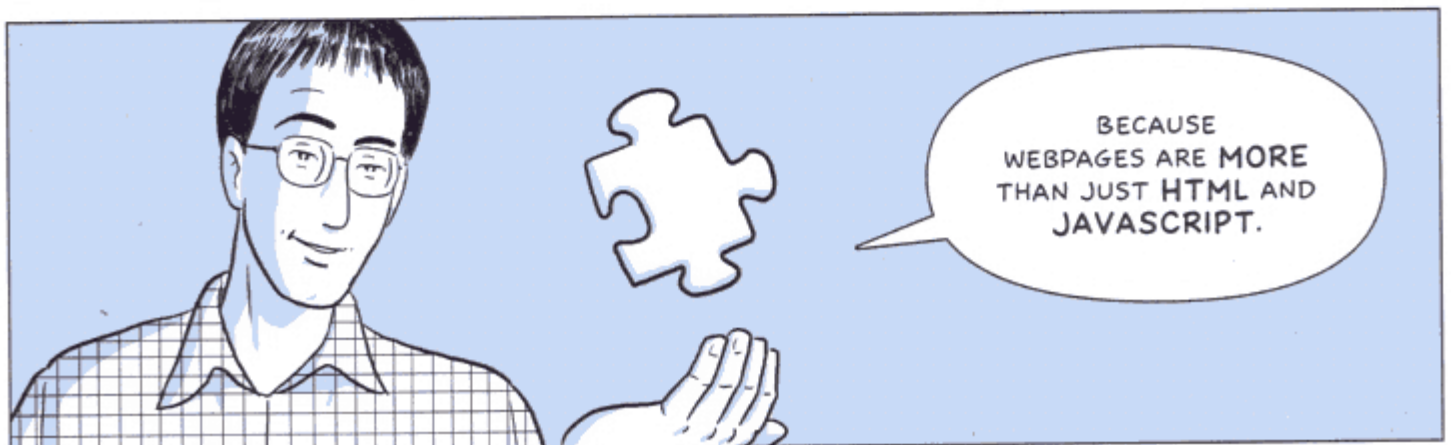
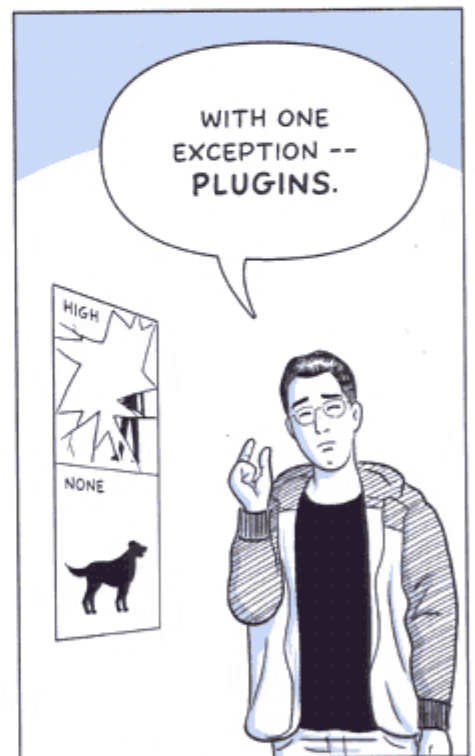


THE PROBLEM IS THAT
UNLIKE THE HIGH LEVEL,
THERE IS A LOT OF
SENSITIVE INFO HERE --



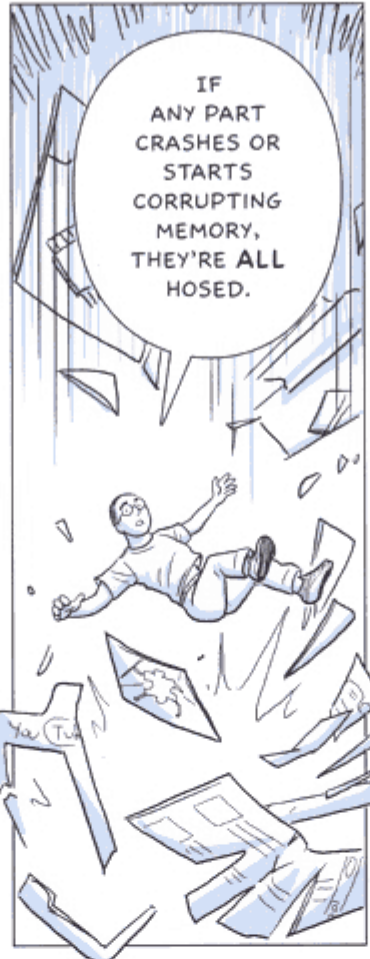
-- THAT
THIS
LEVEL SHOULD
NOT BE
ALLOWED TO
READ!





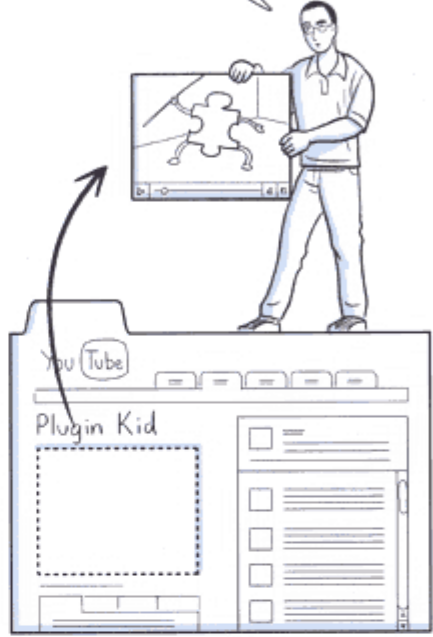
 **PROCESS**

WHEN A
PLUGIN
COMBINES
WITH HTML
AND
JAVASCRIPT, IT
ALL RUNS IN
THE SAME
PROCESS.



IF
ANY PART
CRASHES OR
STARTS
CORRUPTING
MEMORY,
THEY'RE ALL
HOSED.

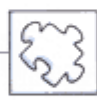
SO I
WORKED
ON
RIPPING PLUGINS OUT
OF THE RENDERING
PROCESS AND PUTTING
THEM IN A SEPARATE
PROCESS ALL THEIR
OWN.



 **CHROME PROCESS**



 **PROCESS**

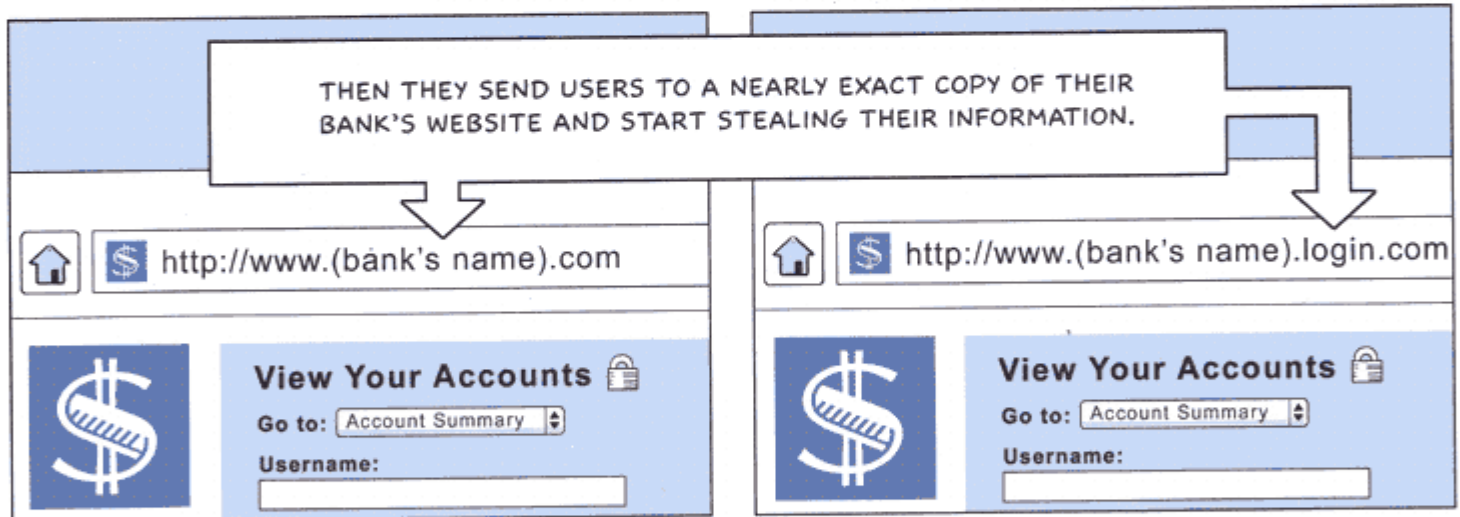
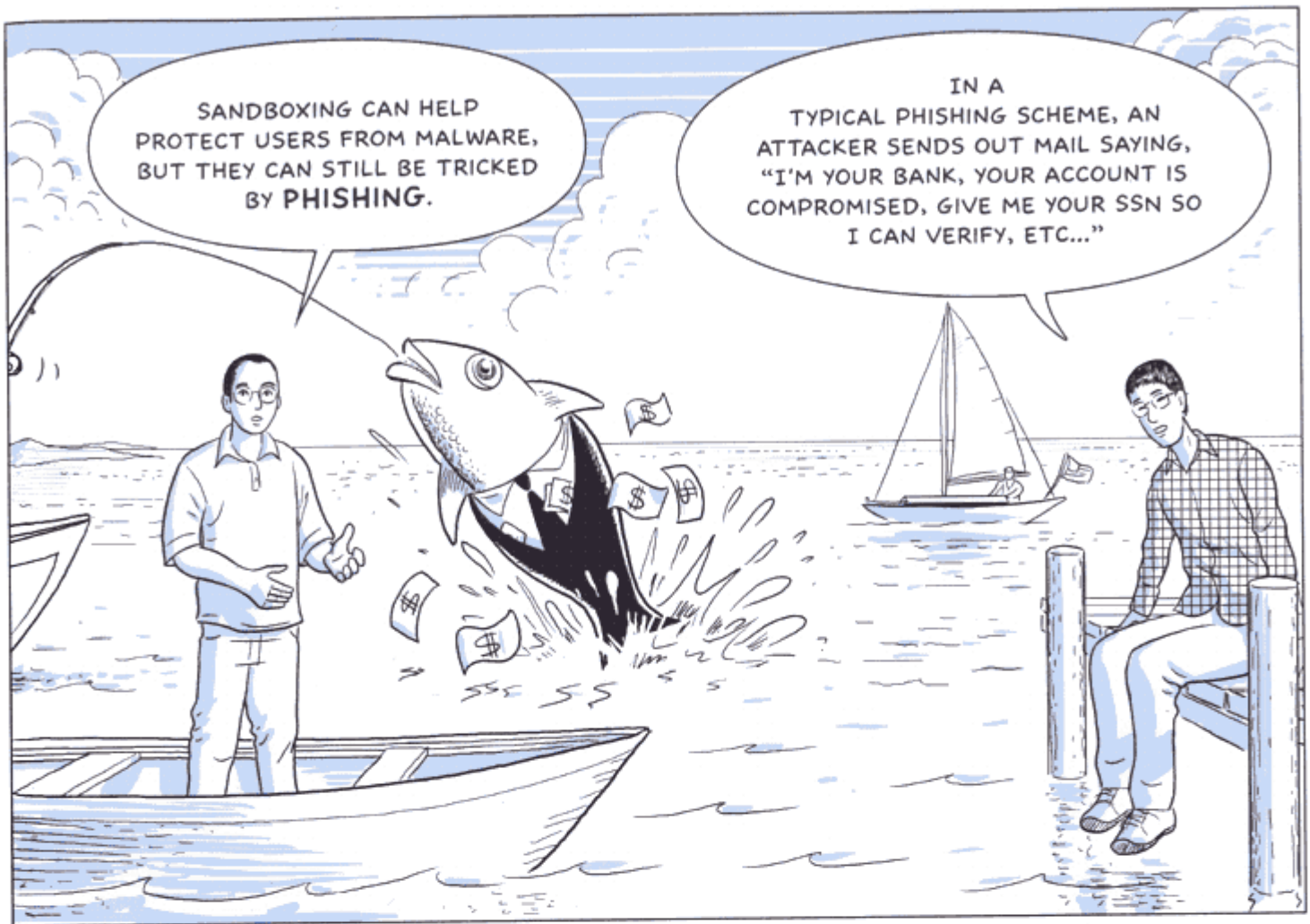


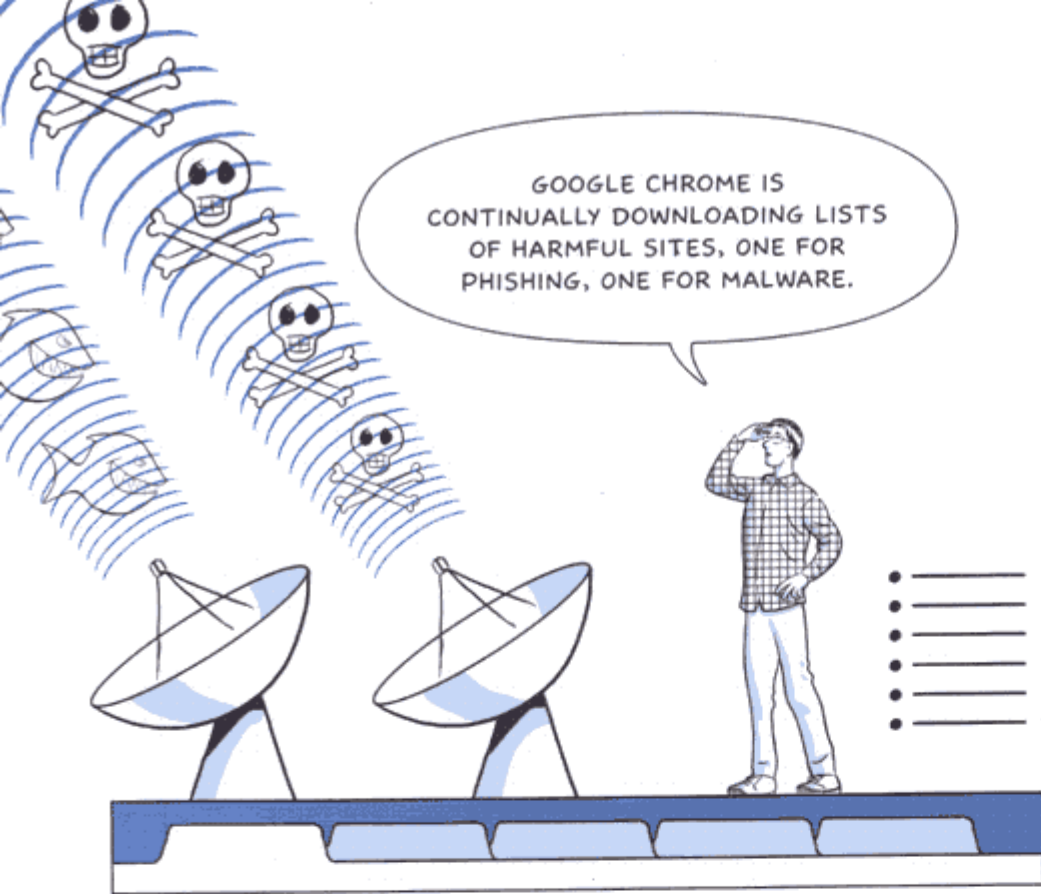
 **PROCESS**



IN
THAT WAY,
THE REST OF THE
PAGE CAN STILL BE
SANDBOXED, EVEN
IF THE PLUGIN
CAN'T BE.







GOOGLE CHROME IS CONTINUALLY DOWNLOADING LISTS OF HARMFUL SITES, ONE FOR PHISHING, ONE FOR MALWARE.

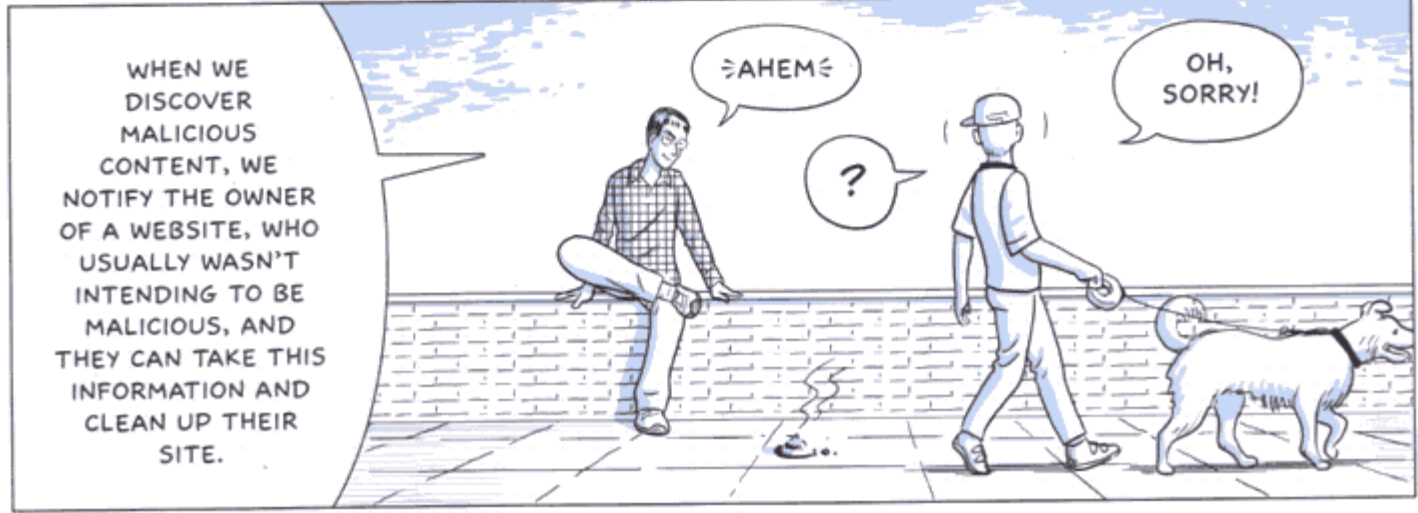
IF YOU GO TO A WEBSITE THAT MATCHES THE LIST, YOU'LL GET A WARNING.



WE'VE MADE THIS SERVICE FREELY AVAILABLE. WE'RE HAPPY TO GIVE IT AWAY. IT'S A PUBLIC API.



THERE'S A SECOND LIST OF MALWARE WEBSITES. WEBSITES WHERE A TON OF BAD THINGS MIGHT HAPPEN TO YOUR COMPUTER, JUST ON ARRIVAL.



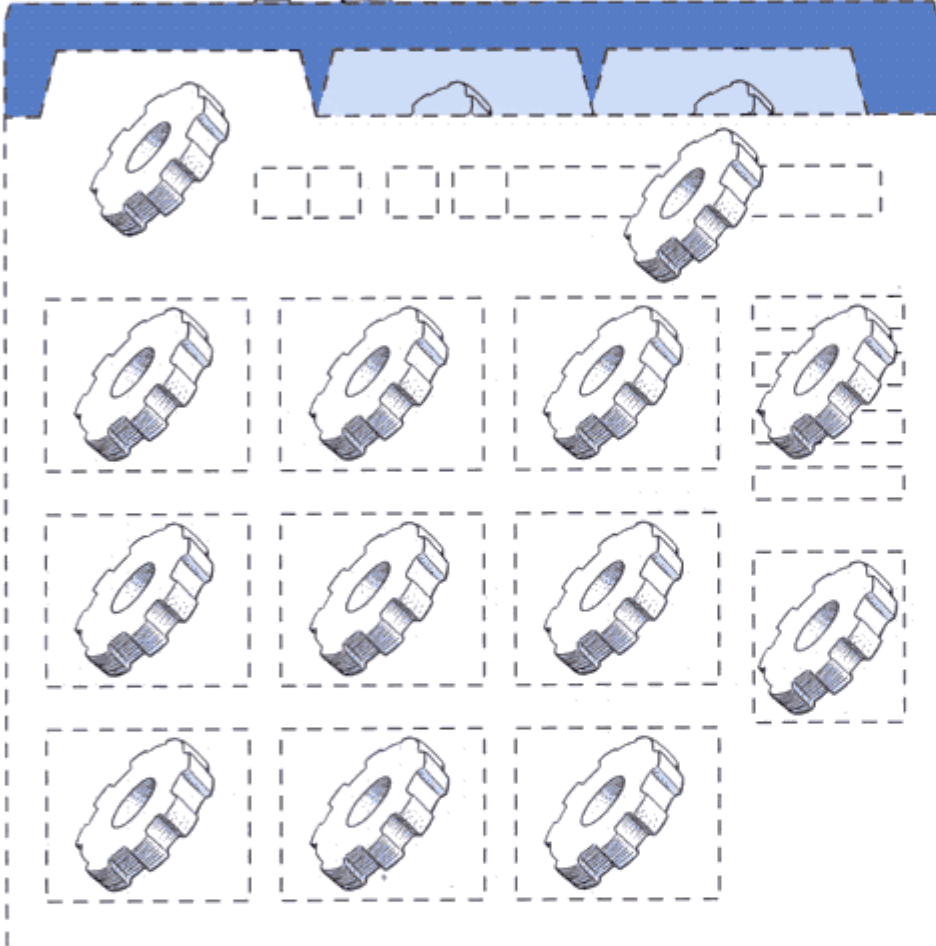
WHEN WE DISCOVER MALICIOUS CONTENT, WE NOTIFY THE OWNER OF A WEBSITE, WHO USUALLY WASN'T INTENDING TO BE MALICIOUS, AND THEY CAN TAKE THIS INFORMATION AND CLEAN UP THEIR SITE.



ANOTHER THING WE BUILT INTO GOOGLE CHROME IS **GEARS**.

Aaron Boodman,
Software Engineer

GEARS BASICALLY ADDS AN API TO YOUR BROWSER -- AN EXTENSION THAT IMPROVES ITS CAPABILITIES.



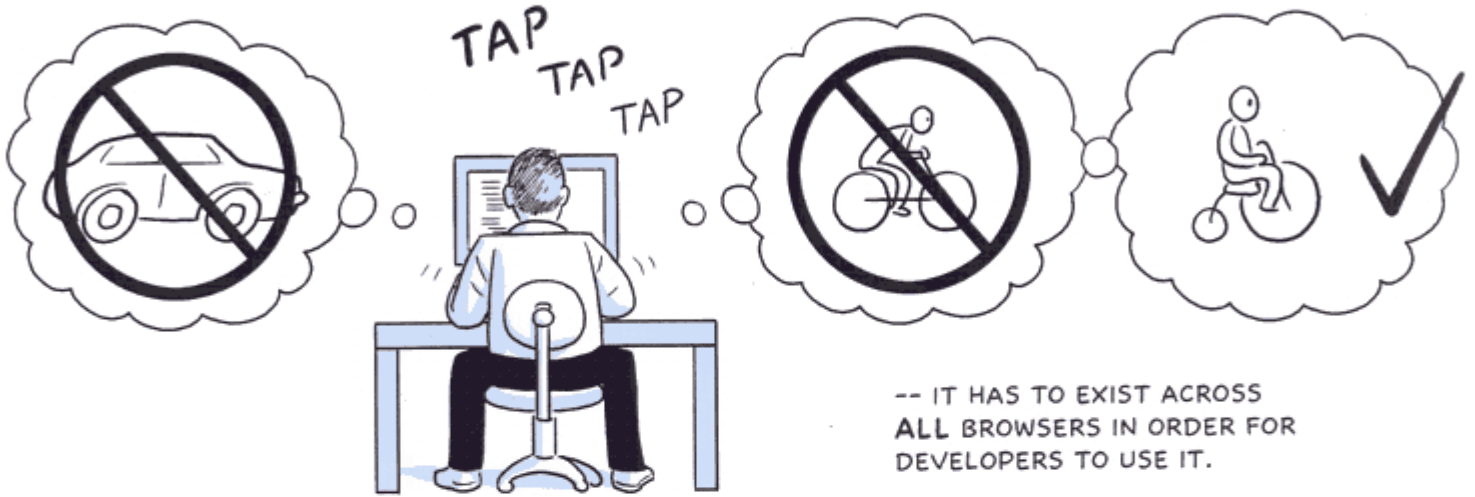
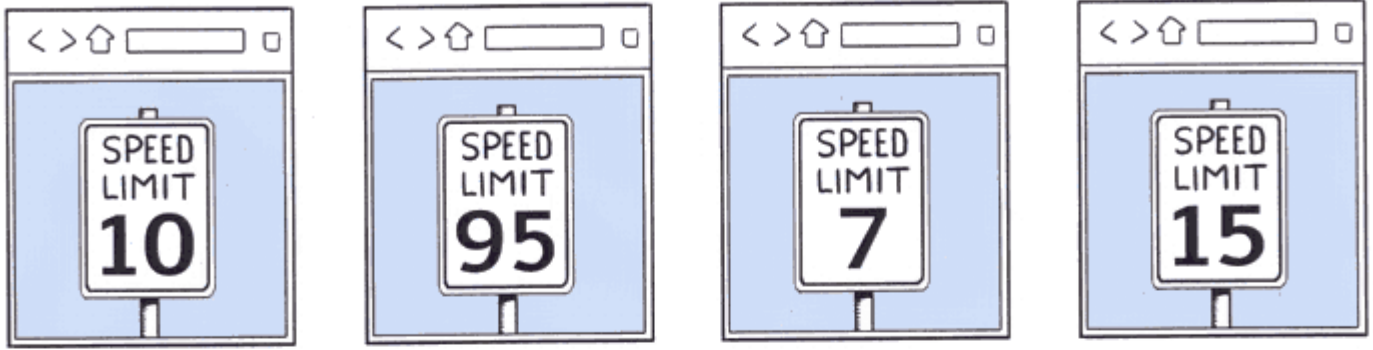
FROM MY PERSPECTIVE, GOOGLE CHROME AND GEARS ARE ENTERING THE WEB FROM TWO DIRECTIONS.

THE BROWSER PROJECT IS AN EFFORT TO MAKE THE WEB BETTER FOR **USERS**.

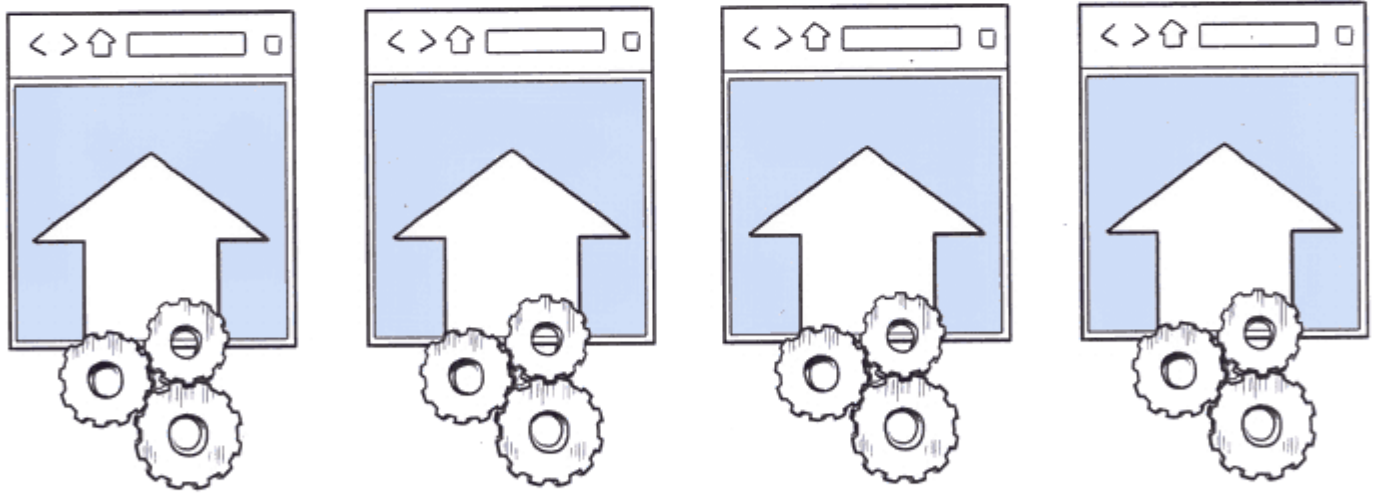
THE GEARS TEAM WANTS TO MAKE THE WEB BETTER FOR **DEVELOPERS**.



THERE ARE A LOT OF LIMITATIONS TO THE KINDS OF APPLICATIONS THAT YOU CAN BUILD TODAY WITH WEB BROWSERS, AND THE SUBSET OF THINGS YOU CAN DO IS DIFFERENT FOR EACH BROWSER. IF ONE BROWSER HAS A COOL FEATURE, THAT DOESN'T HELP --



-- IT HAS TO EXIST ACROSS ALL BROWSERS IN ORDER FOR DEVELOPERS TO USE IT.



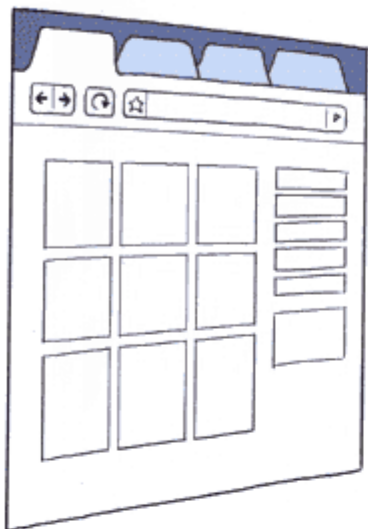
GEARS IS TRYING TO IMPROVE THE BASE FUNCTIONALITY OF ALL BROWSERS, INCLUDING GOOGLE CHROME.

WHATEVER THE ADVANTAGES OF BUILDING NATIVE APPS OVER WEB APPS, WE WANT TO BUILD THOSE ENHANCEMENTS THROUGH GEARS --

-- AND HELP THEM MAKE THEIR WAY INTO NEW STANDARDS ACROSS THE WEB.

SO, OPEN STANDARDS ARE ONE WAY TO HELP ALL BROWSERS GET BETTER.

THE TEAM HAS ALSO DONE SOME INTERESTING THINGS WITH SPEED, STABILITY AND THE UI, LIKE THE NEW TAB PAGE.



SOME OF THESE MIGHT BECOME STANDARDS --



-- SOME MIGHT NOT.



Chris DiBona,
Open Source Programs Manager

BUT --



-- SINCE IT'S OPEN SOURCE --

-- OTHER BROWSER DEVELOPERS CAN TAKE WHAT THEY WANT OUT OF IT.



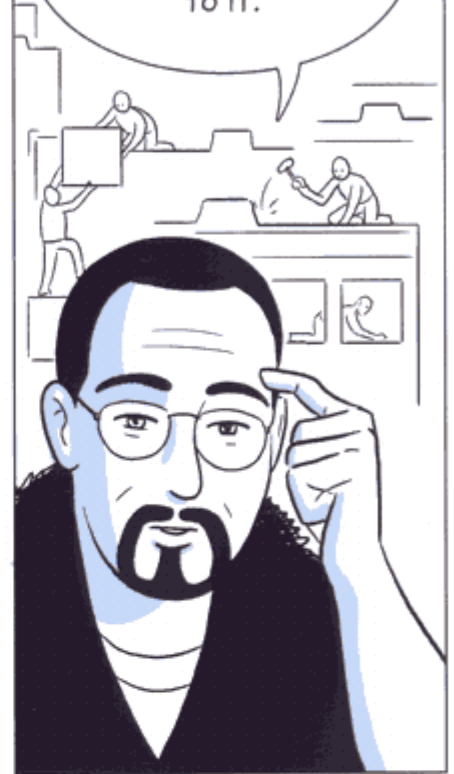
THEY DON'T HAVE TO PAY US. THEY DON'T HAVE TO ASK OUR PERMISSION.

THEY DON'T HAVE TO SHARE PATCHES OR REPORT BUGS.*



* THOUGH, IF THEY LIKE, WE HAVE SYSTEMS IN PLACE FOR THAT.

BUT THEY CAN BUILD ON WHAT WE'VE DONE AND BRING THEIR OWN CREATIVITY TO IT.



SURE, WE COULD SHIP A PROPRIETARY BROWSER AND HOLD IT IN.



BUT GOOGLE LIVES ON THE INTERNET.

IT'S IN OUR INTEREST TO MAKE THE INTERNET BETTER AND WITHOUT COMPETITION WE HAVE STAGNATION.

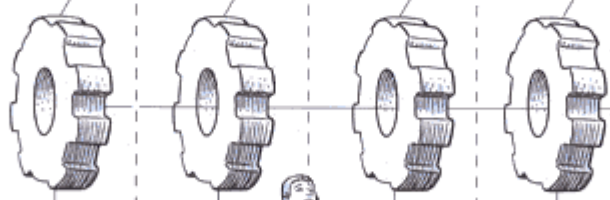
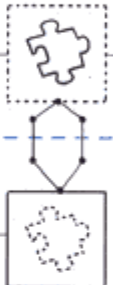
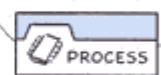


THAT'S WHY WE'RE OPEN SOURCING THE WHOLE THING. WE NEED THE INTERNET TO BE A FAIR, SMART, SAFE PLACE.



AS EXCITED AS WE ARE ABOUT BUILDING GOOGLE CHROME, IT'S IMPORTANT TO HELP ALL BROWSERS BECOME MORE POWERFUL --

-- TO KEEP EVOLVING WITH THE WEB AND CONTINUING TO BUILD A SOLID FOUNDATION FOR MODERN WEB APPLICATIONS.



WE OWE A GREAT DEBT TO OTHER OPEN SOURCE BROWSER PROJECTS -- ESPECIALLY, MOZILLA AND WEBKIT.

THIS IS OUR CONTRIBUTION, AND WE HOPE PEOPLE WILL TAKE SOME OF THESE IDEAS, TOO; CHALLENGE THEM, BUILD ON THEM, AND KEEP MOVING THE WEB FORWARD.

