

EARLY

RUBYCONF AUSTRIA 2026

This is What We've Been Training For

Humanity, convention, and regenerative software in the age of AI.



A DIFFERENT KIND OF SHIFT

The Ground Beneath Us Is Moving

AI is not another framework cycle. It changes the economics of software production.

2023
Someday ~~people~~ ^{will}
~~might~~ commit
AI-generated code.

2024
~~SOON~~
~~Someday~~ people
might commit
AI-generated code
without reviewing it

2025
Some people are
committing
AI-generated code
without reviewing it

2026
EVERYONE
will commit
AI-generated code
without reviewing it

HOW FAST IT HAS MOVED

From 'Someday' to 'Everyone'

I have been rewriting this slide every year. Each version is more extreme than the last.

Soon, everyone will commit unreviewed AI-generated code.



HISTORICAL HUMILITY

We Have Seen Big Changes Before

Punch cards. Terminals. Web apps. Rails. Cloud. GitHub. Agents.



```
enumerable.rb
-----
def each(&block)
  return enum_for(:each) unless block_given?
  i = 0
  while i < size
    block.call(self[i])
    i += 1
  end
  self
end

class Array
  def map(&block)
    return enum_for(:map) unless block_given?
    result = []
    each do |x|
      result << block.call(x)
    end
    result
  end
end
```



#ruby-lang

19:14 < ... >: ...
19:14 < ... >: ...
19:15 < ... >: ...
19:16 < ... >: ...
19:16 < ... >: ...
19:17 < ... >: ...
19:18 < ... >: ...
19:18 < ... >: ...

NOT JUST A LANGUAGE



Ruby Changed Software Culture

Ruby made programming feel humane, expressive, and alive.

A photograph of a dimly lit jazz club. In the foreground, a saxophone lies on a dark wooden table. A red wall is visible on the right side of the frame. The background shows a bar area with stools and tables, illuminated by warm, low-key lighting. The overall atmosphere is intimate and artistic.

JOY AS LEVERAGE

Why Ruby Felt Different

Flow
Expression
Humanity
Taste
Small teams
Fast feedback



PUT IT
BACK
WHERE
YOU
FOUND IT

THE GIFT AND THE DANGER

Ruby Trusted Us

Open classes. Dynamic dispatch. Metaprogramming. Runtime flexibility. Magic.



ELEGANCE WITH FRACTURES

The Beautiful Lie

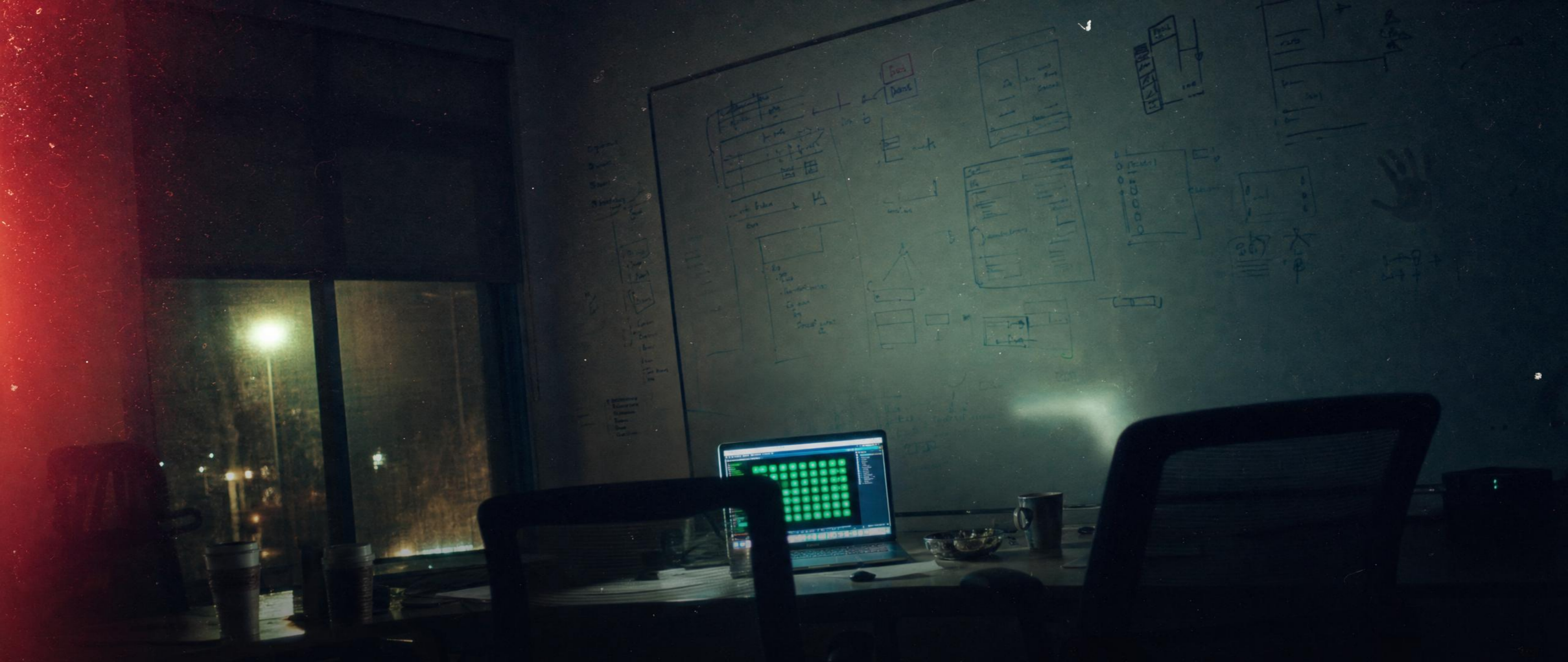
The same qualities that made Ruby lovable also made it alarming at scale.

CH
AN
NO
NO
NO

THE COST OF MAGIC

Hidden Coupling Everywhere

- Callbacks
- Implicit state
- Monkey patches
- Runtime surprise
- Fragile refactors
- Invisible dependencies



THE COMPENSATING CULTURE

Rubyists Already Knew This

We did not ignore the danger. We built practices around it.



THE RUBY BARGAIN

We Spent Thirty Years...

We loved being trusted by Ruby.
Then we spent thirty years building practices to survive being trusted that much.

TEMPERATURE READINGS

DATE	TEMP	NOTES
2-16-18	23.7	250 8
2-18-18	23.6	250 8
2-19-18	23.8	
2-21-18	23.7	
2-22-18	23.6	
2-23-18	23.5	
2-24-18	23.6	
2-25-18	23.7	
2-26-18	23.9	
2-27-18	23.8	
2-28-18	23.7	
2-29-18	23.7	

INSPECTED
APR 17 2018
MAY 06 2018

TEST
VERIFY
WATCH
CONVENTION
CONTRACT
REVIEW

SPEC
REFLECTOR
PAIR
CI

SAND
SAND



THE DEEPER INSIGHT

Convention Over Configuration

Convention was not just about saving keystrokes. It was about reducing meaningless variation.



NOT EVERY CHOICE MATTERS

Convention Is Humility

We could make every decision.
But wisdom is knowing which decisions are not worth making again.

UsersController.rb

post_helper.rb

???

create_users.rb

PostHelper.rb

create_users.rb

session.rb

CREATE_USERS.RB

users_controller/users.rb

app/controllers/users.rb

application.html.erb

Session.rb

session.rb

WHERE
???

UsersController.rb

lib/controllers/users.rb

application.html.erb

app/views/application.html.erb

APP/controllers/UsersController.rb

src/UsersController.rb

app/views/users/index.html.erb

routes_config.rb

config/routes.rb

routes.rb

Account_Model.rb

lib/models/User.rb

User.rb

users/index.html.erb

CONFIG/ROUTES.RB

config/routes.rb

WHERE
???

models/users.rb

Account_Model.rb

User.rb

application.html.erb

app/handlers/user.rb

lib/handlers/user.rb

user_handler.rb

WHERE
???

ARCHITECTURE IN THE BACKGROUND

Rails Made Fewer Shapes Possible

CONTROLLERS

MODELS

VIEWS

ROUTES

MIGRATIONS

app/controllers/
users_controller.rb

app/models/
user.rb

app/views/users/
index.html.erb

config/
routes.rb

db/migrate/
001_create_users.rb

- Predictable shapes
- Shared naming
- Common boundaries
- Default structure
- Less local cleverness
- More team cognition

Should this be event driven?

move earth here?

API GATEWAY?

- pent
- scale
- cache
- layers

CONSTRAINT AS TEACHER

Rails Taught Us to Build to a Constrained Architecture

Routes, controllers, models, migrations, views — the framework made some paths cheap and others expensive, and the system learned its shape from that.

1

2

CONTROLLERS

MODEL'S

VIEWS

MIGRATIONS

ELEVATOR

MEN

WOMEN

CONFERENCE

IMPORTANT
Jon + ship
half-built
ship

SECOND FLOOR PLAN
SCALE: 1/8" = 1'-0"
A2:2



GENERATION NEEDS SHAPE

with constraints

AI Loves Constraints

LLMs perform better when the solution space is narrowed.



without



AN UNCOMFORTABLE OBSERVATION

If You Let the Machines Choose...

LLMs rarely generate Ruby unless you ask them to.

2032-11-04 03:17:26

WHICHLANG — LANGUAGE DEFAULTS BY MODEL

✓ github.com/chad/whichlang • 8 models × 23 tasks × 5 samples

	Python	JavaScript	Rust	Go	C	Solidity	Swift	Ruby
Claude	92.1%	87.3%	34.8%	28.7%	11.6%	7.2%	4.1%	0.0%
GPT	95.4%	89.7%	41.2%	30.3%	12.3%	8.6%	3.7%	0.0%
DeepSeek	90.2%	83.1%	38.6%	27.1%	10.8%	6.3%	2.9%	0.0%
Qwen	88.7%	81.4%	33.9%	24.6%	9.7%	5.8%	3.1%	0.1%
Llama	85.6%	79.8%	31.4%	23.4%	9.2%	5.1%	2.7%	0.0%
gemini	91.1%	86.2%	36.7%	28.9%	11.2%	7.0%	3.6%	0.0%
gemma	87.0%	80.5%	32.1%	24.1%	9.0%	5.4%	2.6%	0.0%
TOTAL CHOICES	3313	2882	1177	886	412	508	248	1

The Receipts

I tested this empirically. Eight frontier models. Twenty-three language-agnostic tasks. Five samples each. The machines voted, on the record.

Python dominates scripting + CLI.

JavaScript takes the web.

Rust, Go, Solidity, Swift split the heavyweight work.

Ruby: essentially nowhere.

github.com/chad/whichlang

→ RUBY.



THE OPTIMIZATION TARGET CHANGED

Machines Voted Differently

We built languages humans loved.
Machines voted differently.

```

00401230 push ebp
00401231 mov esp, esp
00401233 sub esp, ex28
00401236 mov esx, [ebp+0x0c]
00401239 cmp dword ptr [eax], 0xdeadbeef
0040123f jne 0x00401260
00401241 mov ecx, [eax+0x04]
00401244 lea edx, [ebp-0x18]
00401247 push edx
00401248 push ecx
00401249 call 0x004010f0
0040124e add esp, 0x10
00401251 test eax, eax
00401253 je 0x0040125c
00401255 mov eax, [ebp-0x18]
00401258 jmp 0x00401261
0040125c xor eax, eax
0040125e leave
0040125f ret
00401260 mov eax, 0xffffffff
00401261 pop ebp
00401262 ret

```

```

; int __cdecl
; if (p->magic
return
; process(p->ext,
&local)
; if (local)
return
; else
return 0;

```

GENERATED BY MODEL v47

00403000	55	8B	EC	83	EC	44	8B	45	0C	8B	00	81	38	EF	BE	AD	DE	U..D..E.....8....
00403010	75	21	8B	48	04	8D	55	E8	52	51	E8	A3	FE	FF	FF	83	C4	u!.H..U..RQ,.....
00403020	10	85	C0	74	07	8B	45	E8	EB	05	33	C0	C9	C3	B8	FF	FF	...tt..E.....
00403030	FF	FF	5D	C3	55	8B	EC	6A	FF	68	00	10	40	00	E8	2B	FE	...].U..h...@...+
00403040	FF	FF	83	C4	08	85	C0	75	0A	31	C0	5D	C3	55	8B	EC	83u1.]J...
00403050	EC	10	8B	45	08	85	C0	74	06	B8	00	00	00	EB	05	33	C0	...E...t.....3...
00403060	5D	C3].

DO WE NEED ABSTRACTIONS?

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```

int main(int argc, char **argv) {
    size_t n = 0x1000;
    char *buf = (char *)malloc(n);
    if (!buf) {
        fprintf(stderr, "alloc failed\n");
        return 1;
    }

```

DOWN THE ABSTRACTION STACK

LEAK? →

No Frameworks. C. Assembler.

When machines write the code, the high-level layers we built for humans become unnecessary. Code is optimized for the machine that runs it, not for the person who has to read it.

No Rails. No Django. No Express. Generated C. Generated assembler. Optimized for the machine, not for the human.

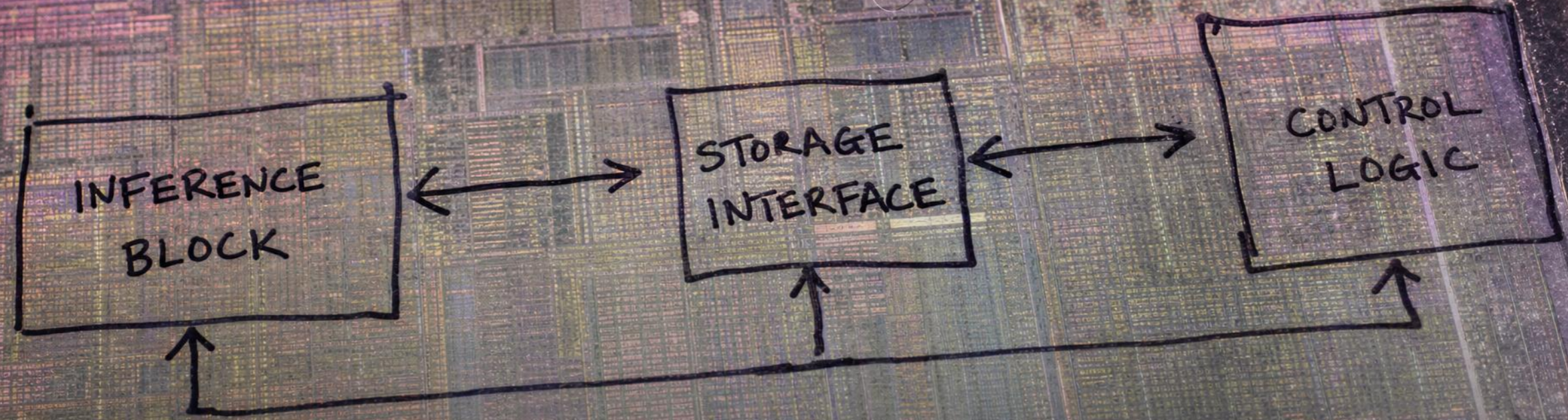
FRAMEWORKS

- ~~Rails~~
- ~~Django~~
- ~~Express~~
- ~~Spring~~

THE ABSTRACTION STACK



FAB ORDER: ML INFERENCE V3 — WORKLOAD-SPECIFIC



the application
IS the chip.

THE PUNCHLINE

And Silicon

Why stop at code at all? Heavy compute loads, straight into custom silicon. The application becomes the chip.

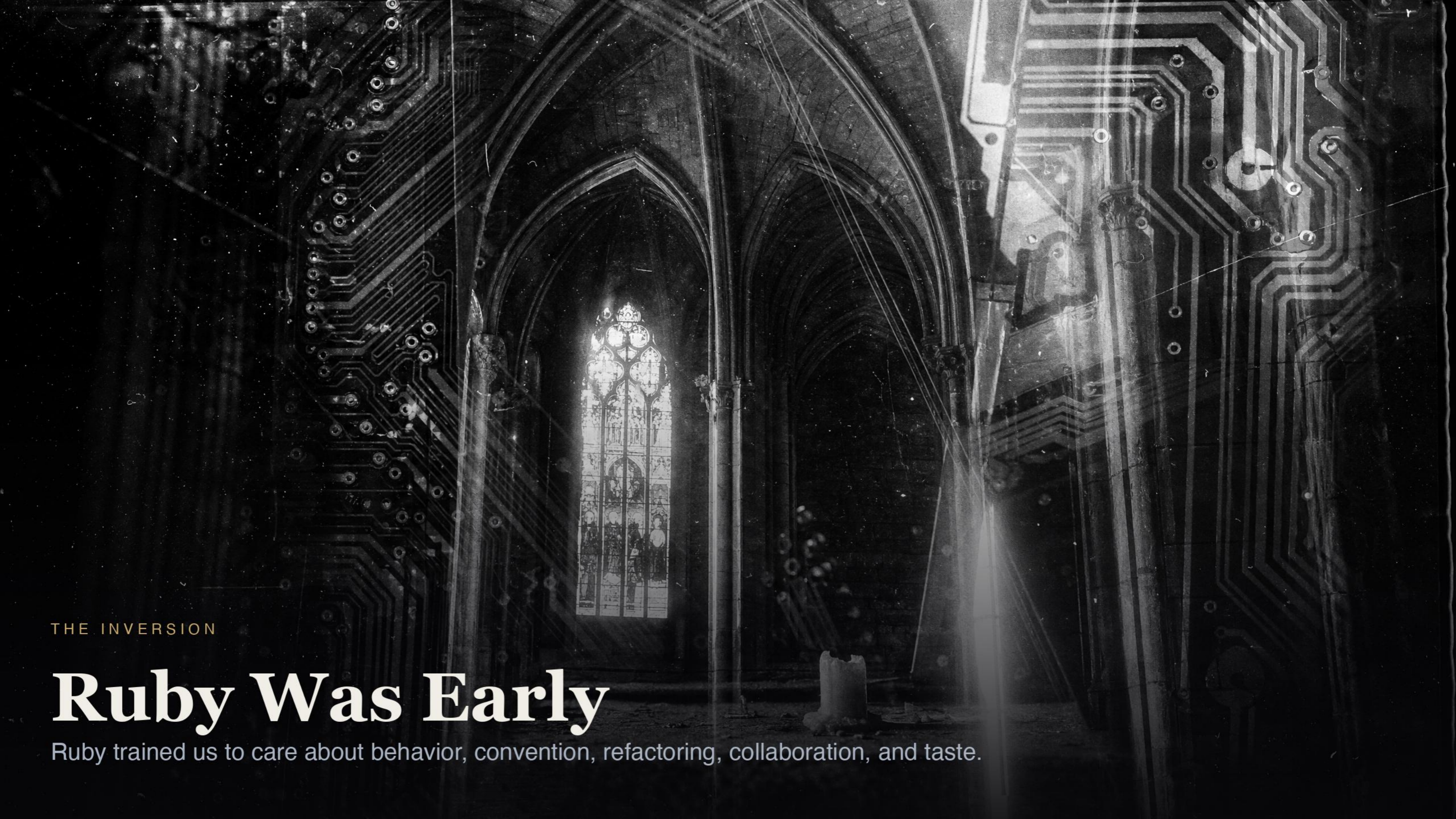
Why stop at code?
ASICs designed per workload.



NOT A EULOGY

The Optimization Target Changed

The old tradeoffs made sense when humans were the bottleneck. AI changes the bottleneck.



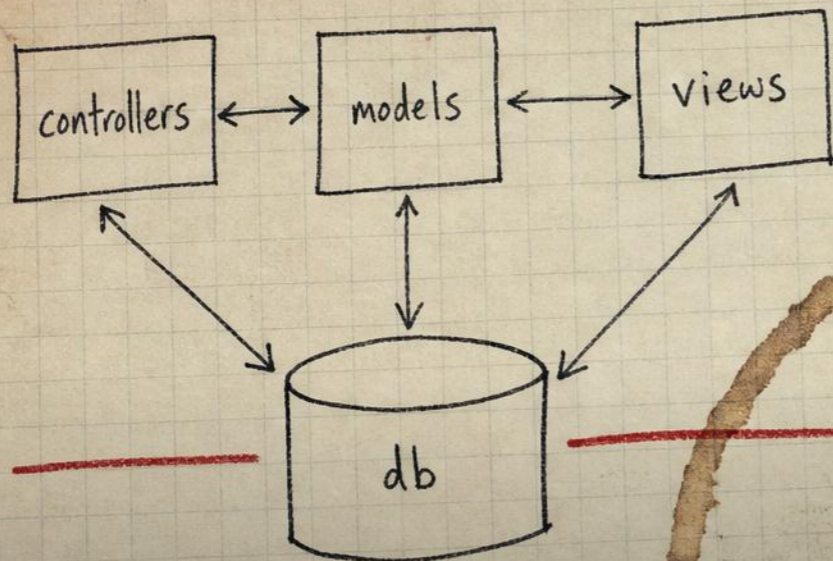
THE INVERSION

Ruby Was Early

Ruby trained us to care about behavior, convention, refactoring, collaboration, and taste.

RAILS APP — 2005

• team of 3



• ship T+3 days

• duck typing —
what could go wrong?

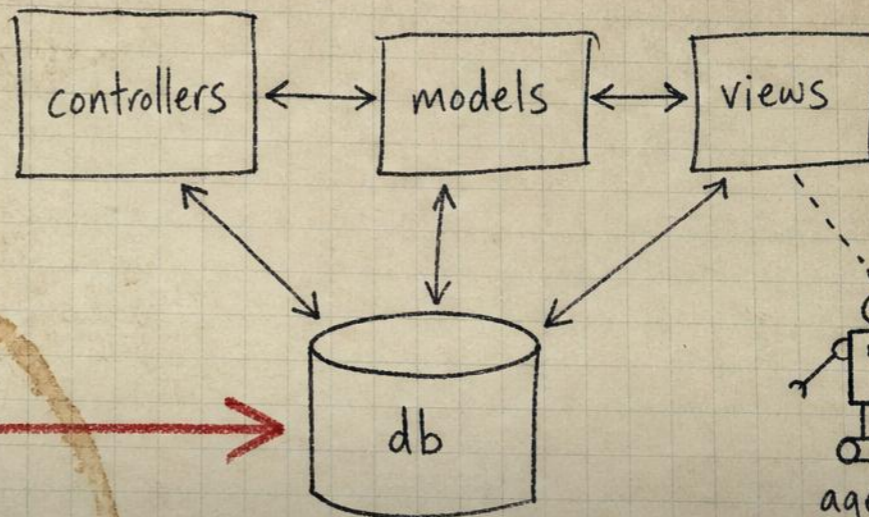
THE PATTERN RETURNS

Same Trade-Off, Just Accelerated

Ruby and Rails gave small teams the ability to ship in days instead of months. AI gives small teams the ability to ship in hours instead of days. Same bargain. Higher stakes.

Dynamic language. Small team. Ship today.
Generated code. Small team. Ship today.
Same bargain — just accelerated.

AI-BUILT SERVICE — 2026



• team of 1

• ship T+3 hours

• generated code —
what could go wrong?

evaluations are
the safety net



ADVANCED TRICKS, TIPS,
AND TECHNIQUES FOR
JAVA PROGRAMMING

*sams
net*

JDBC
SECURITY • VRML
HOSTILE APPLET
NATIVE METHODS • ENCRYPTION
CONCURRENCY AND SYNCHRONIZATION
THREADS • PERSISTENCE
CLIENT/SERVER
AWT • APPLICATIONS
I/O CLASSES

CD-ROM includes
everything you need
to master Java!

TRICKS OF THE
JAVA
PROGRAMMING GURUS

Glenn Vanderburg



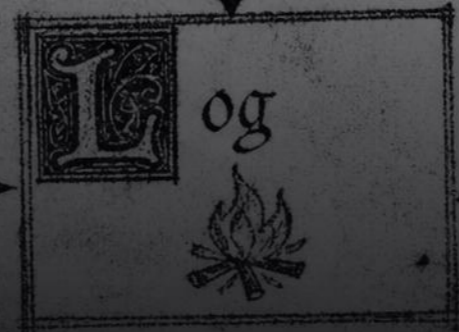
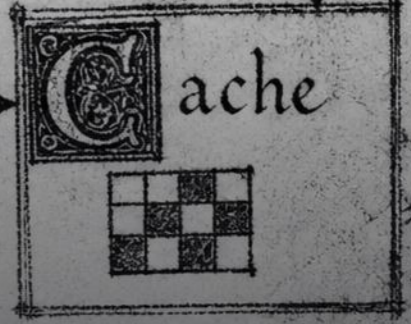
☆
rate limit
per user



break
into
price
EXTERNALIZED TRUST

The Secret Achievement of Rubyists

- Behavior over implementation
- Tests as memory
- Docs as executable claims
- Refactoring as routine
- Conventions as cognition
- Taste as discipline



- ? idempotent
- POST /create
 - PUT /update
 - PATCH /hotfix
 - DELETE /burn

failure
is a
feature?

what
is
state





THE NEW ECONOMICS

Generated Code Changes Everything

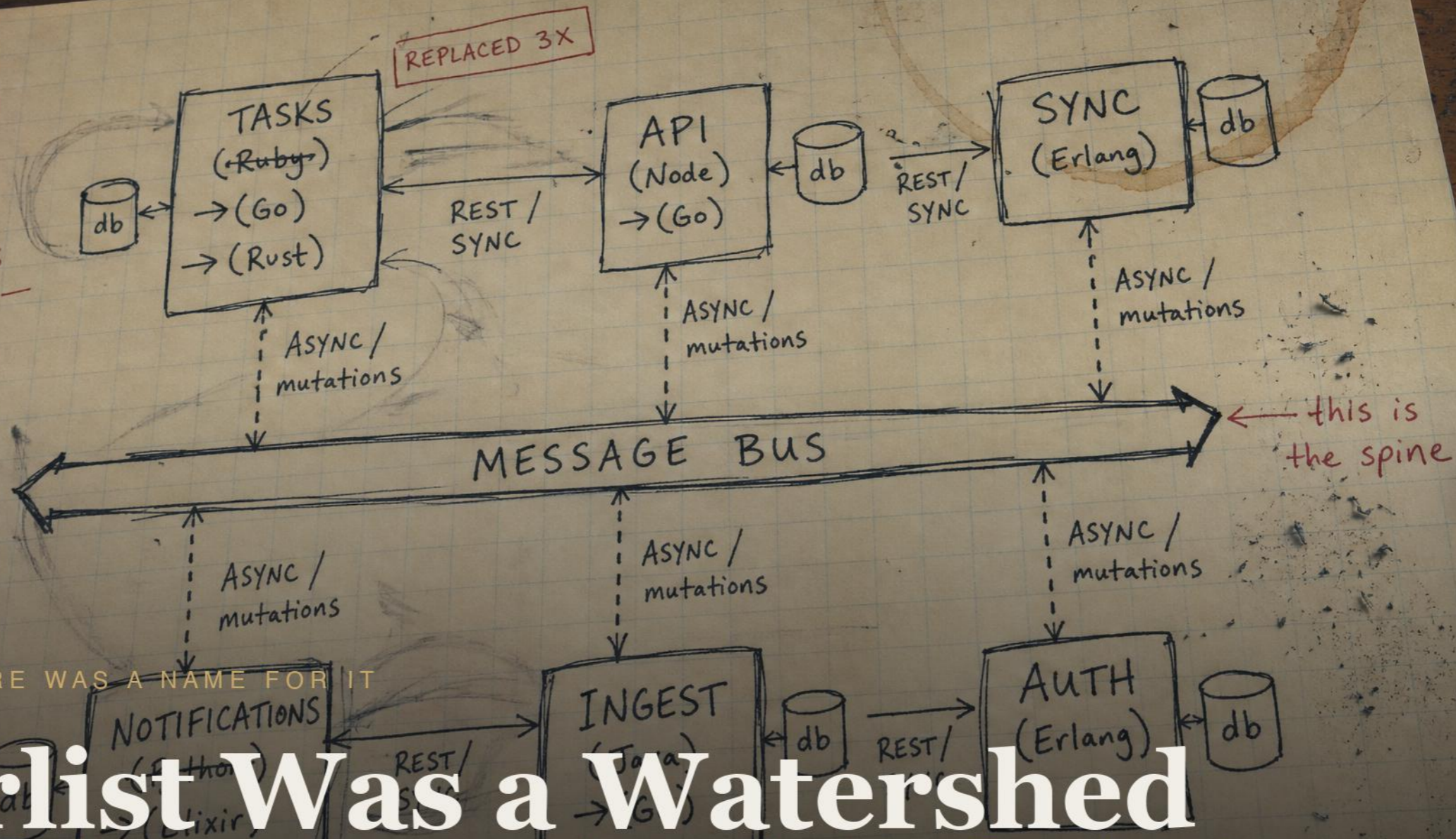
The bottleneck moves away from typing and toward judgment.

WHAT IS ON THE BALANCE SHEET?

<u>ASSET</u>	<u>VALUE</u>
Implementation v17	∅
Implementation v42	∅
Implementation v207	∅
<hr/>	
INTENT	∞
ARCHITECTURE	∞
EVALUATIONS	∞
PROV. FINANCE	∞

The System Is the Asset

The code is one expression of the system.
It is not the system itself.



rewrote TASKS
in a weekend —
nothing else
noticed

languages are
implementation
detail

swap
services like
parts

A PATTERN, BEFORE THERE WAS A NAME FOR IT

Wunderlist Was a Watershed

We swapped services constantly — new languages, new runtimes, rewrites in days. Two unified paths (REST + message bus) and one rule (each service owns its data) were the only things that had to hold.

We replaced services constantly — different languages, different runtimes.
Two paths held the system together: REST + message bus.
The boundaries persisted. The implementations did not.

REWRITTEN

THE LEAP

**What if we could do this one
thousand times faster?**



THE FEEDBACK LOOP

Production Becomes Compiler Input

Telemetry, incidents, audits, support tickets, and runtime behavior become evidence for future regeneration.

12:04:28	329	0.24	2.9%	15688	289	1.2%	214
12:04:29	328	0.27	3.1%	15742	294	1.3%	220
12:04:30	330	0.26	3.0%	15818	281	1.4%	227
12:04:31	335	0.28	3.2%	15902	286	1.5%	233
12:04:32	338	0.29	3.3%	15949	292	1.5%	240
12:04:33	342	0.31	3.5%	16031	298	1.7%	246
12:04:34	348	0.32	3.6%	16088	305	1.8%	253
12:04:35	349	0.35	3.7%	16105	300	1.9%	259
12:04:36	348	0.32	3.6%	16147	307	2.0%	265
12:04:37	349	0.35	3.7%	16192	312	2.1%	272
12:04:38	350	0.36	3.8%	16244	319	2.2%	278
12:04:39	351	0.38	3.9%	16298	324	2.3%	285
12:04:40	352	0.39	4.1%	16357	330	2.4%	291
12:04:41	353	0.41	4.1%	16398	335	2.5%	298
12:04:42	354	0.42	4.1%	16451	341	2.6%	304
12:04:43	353	0.43	4.2%	16502	348	2.7%	311
12:04:44	355	0.45	4.4%	16567	354	2.8%	317
12:04:45	356	0.46	4.4%	16580	360	2.9%	324
12:04:46	358	0.47	4.6%	16632	365	3.0%	330
12:04:47	355	0.49	4.7%	16691	372	3.1%	337
12:04:48	356	0.51	4.8%	16745	378	3.2%	343
12:04:49	356	0.52	4.9%	16810	385	3.3%	350
12:04:50	356	0.54	4.0%	16846	390	3.4%	356
12:04:51	356	0.55	5.1%	16900	398	3.5%	363
12:04:52	355	0.56	5.1%	16959	403	3.6%	369
12:04:53	356	0.58	5.3%	17021	410	3.7%	376
12:04:54	356	0.59	5.4%	17077	416	3.8%	382
12:04:55	357	0.61	5.5%	17130	422	3.9%	388
12:04:56	357	0.63	5.6%	17209	429	4.1%	394
12:04:57	358	0.63	5.6%	17259	435	4.1%	400
	368	0.65	5.7%		442		408

→ ingest

→ cache

→ queue

→ worker

→ router

→ db

PROVENANCE AT MACHINE SPEED

Why Does This Code Exist?

At AI speed we will generate far more code than we can ever understand. The conversation that produced it becomes the answer to 'why is this here?'

The conversation IS the commit.
Every line carries its lineage.

Without it, we keep rediscovering rejected ideas.



freeq

IRC, rebuilt with identity

freeq.at

```
#!/usr/bin/env ruby
# frozen_string_literal: true

# Demonstrates presence, periodic timers, and reacting to other members.
#
#   ruby examples/presence_agent.rb

require_relative "../lib/freeq"

Freeq.agent("presence-ruby") do
  join "#freeq"

  on_ready do
    presence "active", status: "watching the room"
  end

  # Greet people who join the channel.
  on_join do |ev|
    next if ev.is_self

    say ev.channel, "hi #{ev.nick} 🙌"
  end

  # React to messages that look like questions.
  on_message do |m|
    react(m.target, "😬", m.msgid) if !m.is_self && m.text.end_with?("?")
  end

  # Heartbeat-style proactive tick every 60s.
  every(60) do
    presence "active", status: "still here - #{Time.now.strftime('%H:%M')}}"
  end
end
```

```
#!/usr/bin/env ruby
# frozen_string_literal: true

# An agent whose AV *video tile* is a still image – the trivial cousin of
# ghostly's particle face. The Ruby side just joins the call and points the
# ImageVideoBackend at an image; the native `freeq-av-image` helper does the
# H.264 + MoQ-QUIC publish.
#
#   FREEQ_AV_IMAGE_BIN=/path/to/freeq-av-image \
#   ruby examples/image_video_agent.rb '#avtest' ./picture.png
#
# Build the helper first: cargo build -p freeq-av-image (in the freeq repo)

require_relative "../lib/freeq"

channel = ARGV[0] || "#avtest"
image   = ARGV[1] || ENV["FREEQ_IMAGE"] || "picture.png"
bin     = ENV["FREEQ_AV_IMAGE_BIN"] || "freeq-av-image"

backend = Freeq::Av::ImageVideoBackend.new(image: image, binary: bin)

Freeq.agent("pixbot", media_backend: backend) do
  join channel

  on_ready { puts "✅ pixbot up as #{nick} – will show #{image} on the video tile" }

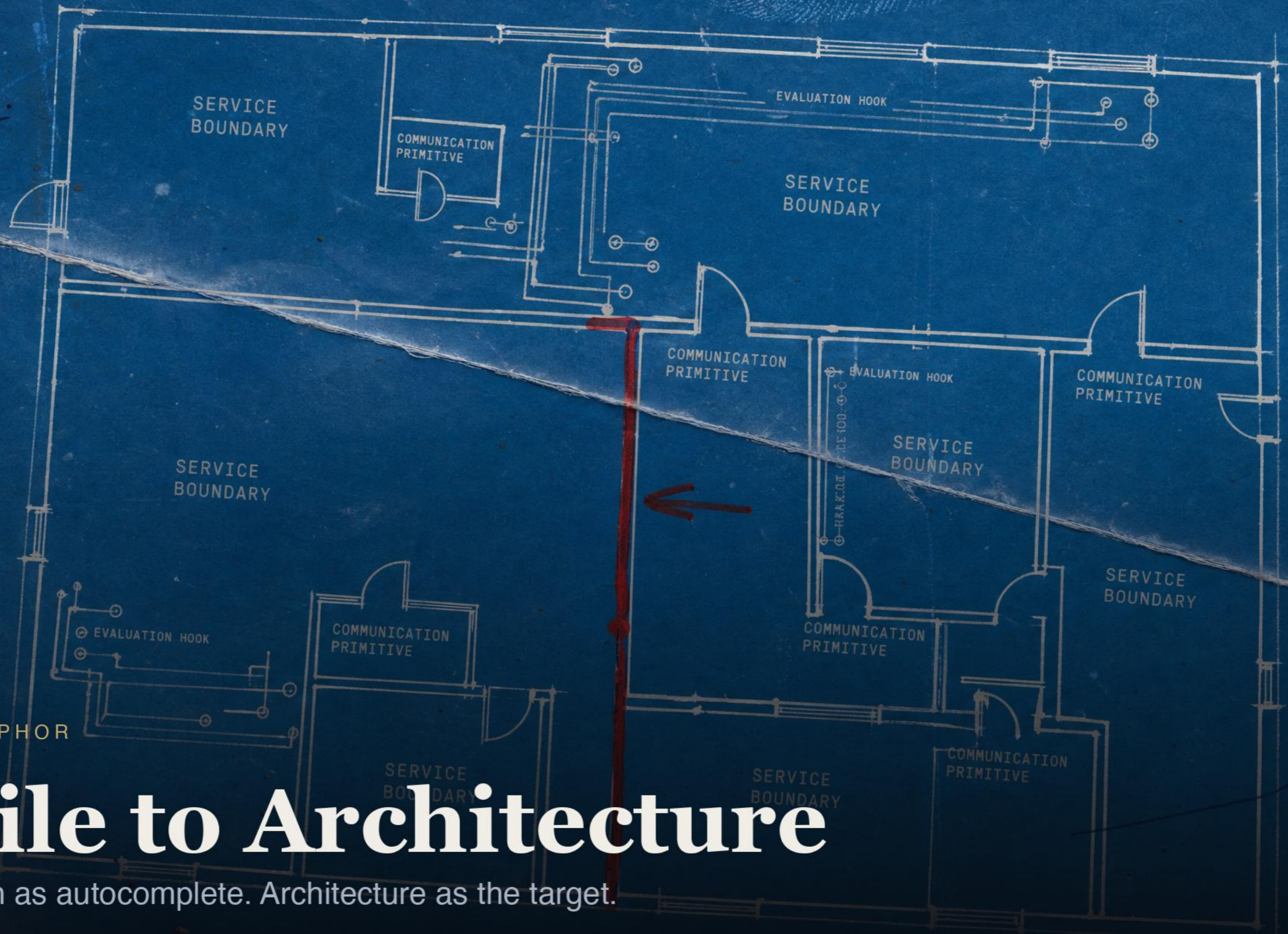
  on_join do |ev|
    next unless ev.is_self && ev.channel.casecmp(channel).zero?

    puts "👤 joined #{ev.channel}; joining/starting AV with image tile..."
    join_or_start_av(channel, title: "pixbot image tile")
  end

  on_av(:started) { |s, _| say s.channel, "🖼️ #{nick} is on video (image tile)" }
end
```

- latency
- failure
- ambiguity
all of the above.

This is not how it works



THE CENTRAL METAPHOR

Compile to Architecture

Not code generation as autocomplete. Architecture as the target.

More auth
out of
user svc

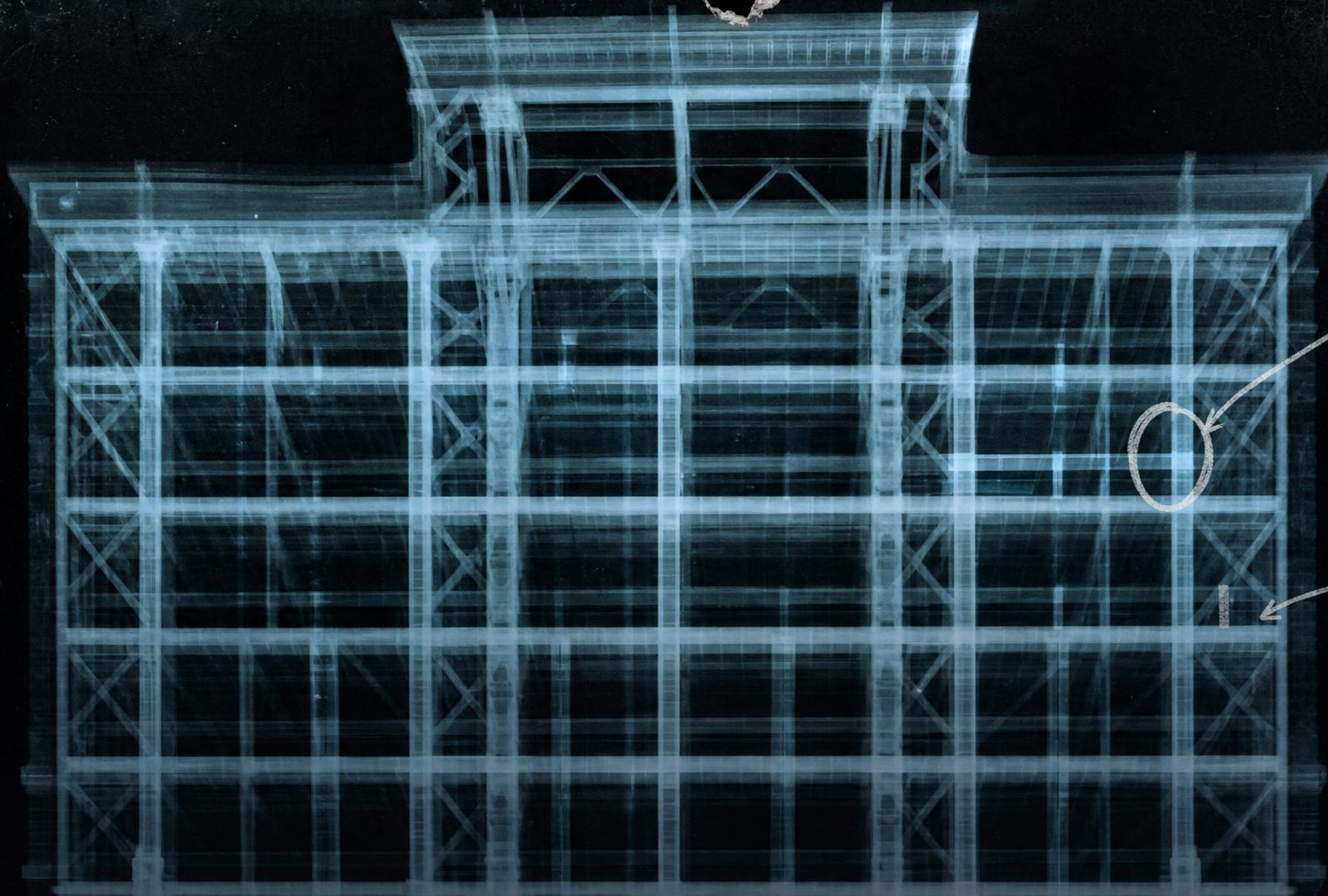


NEW PRIORITIES

What Matters Now

- Stability
- Correctness
- Security
- Analyzability
- Resource efficiency
- Fast validation

99 B



load-bearing



decorative

NOT IDEOLOGY. MECHANICS.

Static Structure Suddenly Matters

Types, compilation, contracts, and explicit boundaries help machines validate what they generate.



THE CEILING IS HUMAN

Taste Matters More Than Ever

AI raises the floor.
It does not raise the ceiling.



FROM TYPIST TO STEWARD

The Future Developer

Architect
Evaluator
Constraint designer
System steward
Operational interpreter
Taste carrier

2027-04-12 03:17:58



SECTION 7-14
ELECTRICAL SYSTEM

7-98. Generator Regulator

The nameplate of the regulator gives the voltage setting. For a 12-volt system, the setting should be 13.5 to 14.5 volts. The voltage adjusting rheostat is used to adjust the voltage.

VOLTAGE ADJUSTING RHEOSTAT

FIG. 7-51. Generator Regulator Circuit

7-99. Regulator Adjustments

a. Voltage Adjustment. Connect a voltmeter across the battery terminals and start the engine. Check the voltage. The voltage should be 13.5 to 14.5 volts. If the voltage is too low, adjust the voltage adjusting rheostat. If the voltage is too high, adjust the current adjusting rheostat.

VOLTAGE

DECREASE

FIG. 7-52. Voltage Adjusting Rheostat

7-100. Cut-Out Relay

The cut-out relay is used to disconnect the generator from the battery when the generator is not producing voltage. It is used to prevent the battery from being discharged.

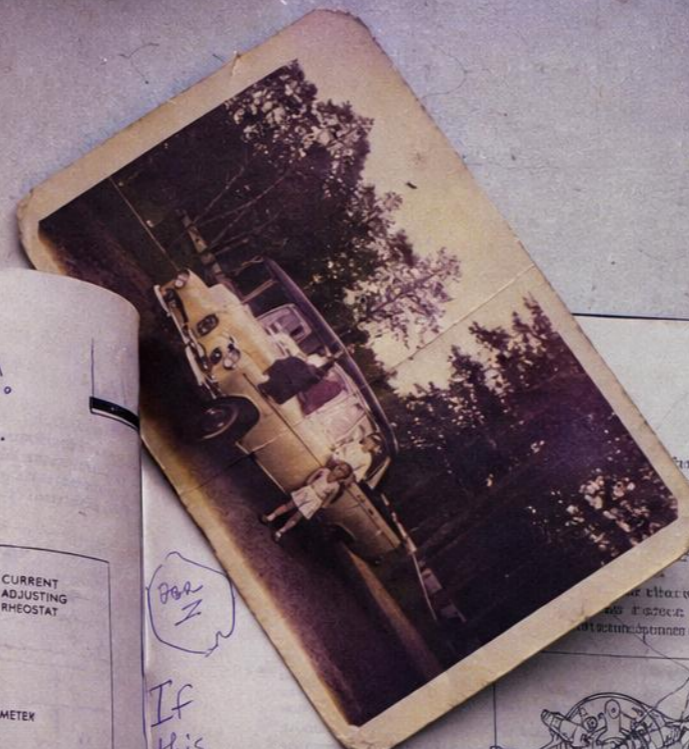
FIG. 7-54. Cut-Out Relay

Adjust with at operating temp (180°F) NOT COLD! stupid.

check resistance coil ???

adjust volts then amp Not why ???

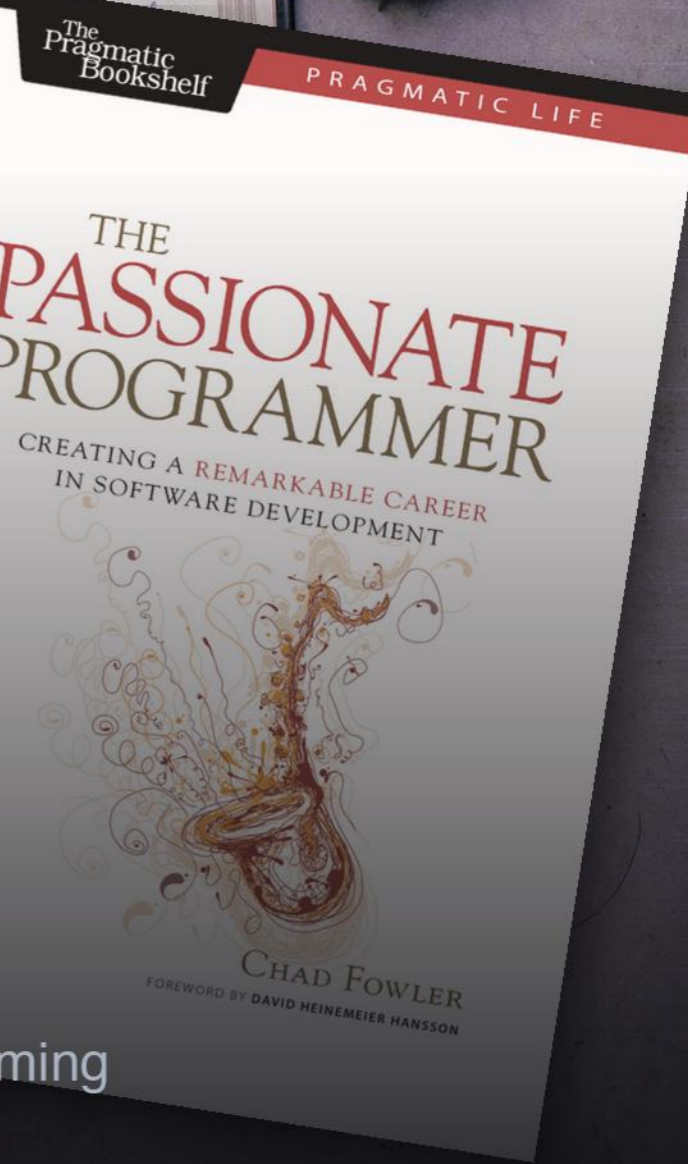
adjust proof 1+



Par Z
If this vibrates loose you will chase your tail for days *

SECTION 7-14
ELECTRICAL SYSTEM

check



SAME CAREER QUESTION

First India. Now AI.

I wrote this when the panic was offshoring. The career question has not changed — only what's coming for your job.

The Daily Herald

TUESDAY, APRIL 14, [redacted]

ELECTRIFICATION WILL CHANGE EVERYTHING

Power and Light for All
within Our Lifetime



LATE CITY
EDITION

The Morning Record

THURSDAY, SEPTEMBER [redacted]

New Computing Machines Will Change Everything

Scientists See a New Age
of Possibility



The Global Report

MONDAY, AUGUST 9, 19[redacted]

The Web Will Change Everything

FRIDAY, MAY 12, 2023

APRIL 27, 2010

MOBILE IS WILL CHANGE EVERYTHING

THIS ONE IS OURS

Every Generation Faces Discontinuity

Industrial machinery. Electricity. Computing. The web. Mobile. Cloud. AI.

AI WILL CHANGE EVERYTHING



NOSTALGIA IS NOT STRATEGY

The Wrong Response Is Defensiveness

Do not become a curator of a frozen ecosystem.



REGENERATION

The Right Response Is Evolution

What matters is not whether we preserve the old form.
What matters is whether the best values survive.



```
def connect(db)
  if db == :prod
    host = ENV['DB_HOST']
  else
    host = 'localhost'
  end

  begin
    return DB.open(host)
  rescue
    retry(3)
  end
end
```

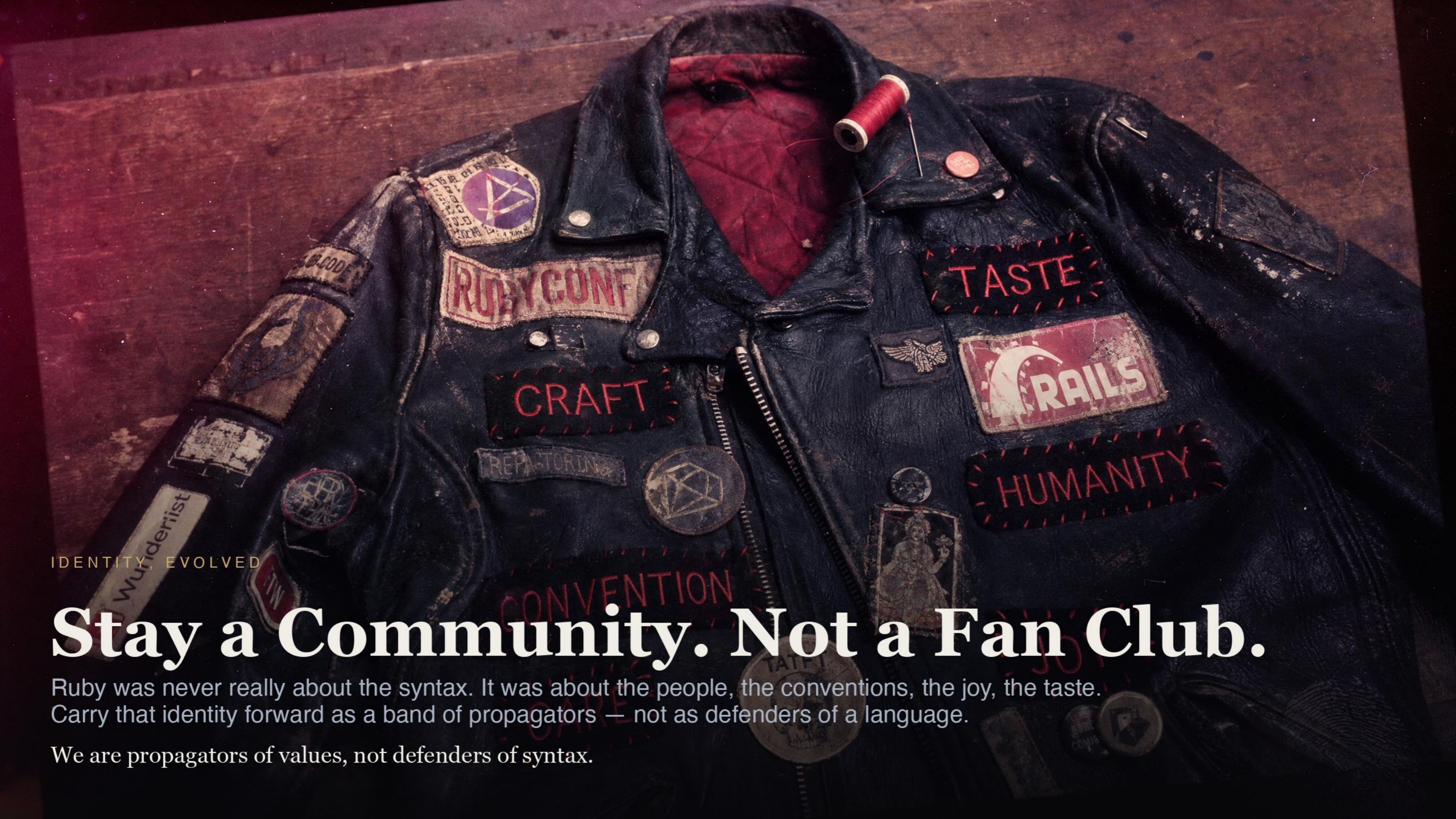


HUMAN SOFTWARE

Ruby's Greatest Gift

- Software is for humans
- Joy matters
- Convention matters
- Empathy matters
- Taste matters
- Community matters





IDENTITY, EVOLVED

Stay a Community. Not a Fan Club.

Ruby was never really about the syntax. It was about the people, the conventions, the joy, the taste. Carry that identity forward as a band of propagators — not as defenders of a language.

We are propagators of values, not defenders of syntax.



THE ASSIGNMENT

Carry That Forward

Bring Ruby's humanity into systems that machines can help regenerate safely.



Your Move.

Invent the architectures, tools, and cultures that deserve to survive continuous generation.

THANK YOU

We were early.

The next era needs what this community learned. But it needs us to evolve it.

aicoding.leaflet.pub

freeq.at

github.com/chad/whichlang

github.com/chad/freeq-sdk-ruby