

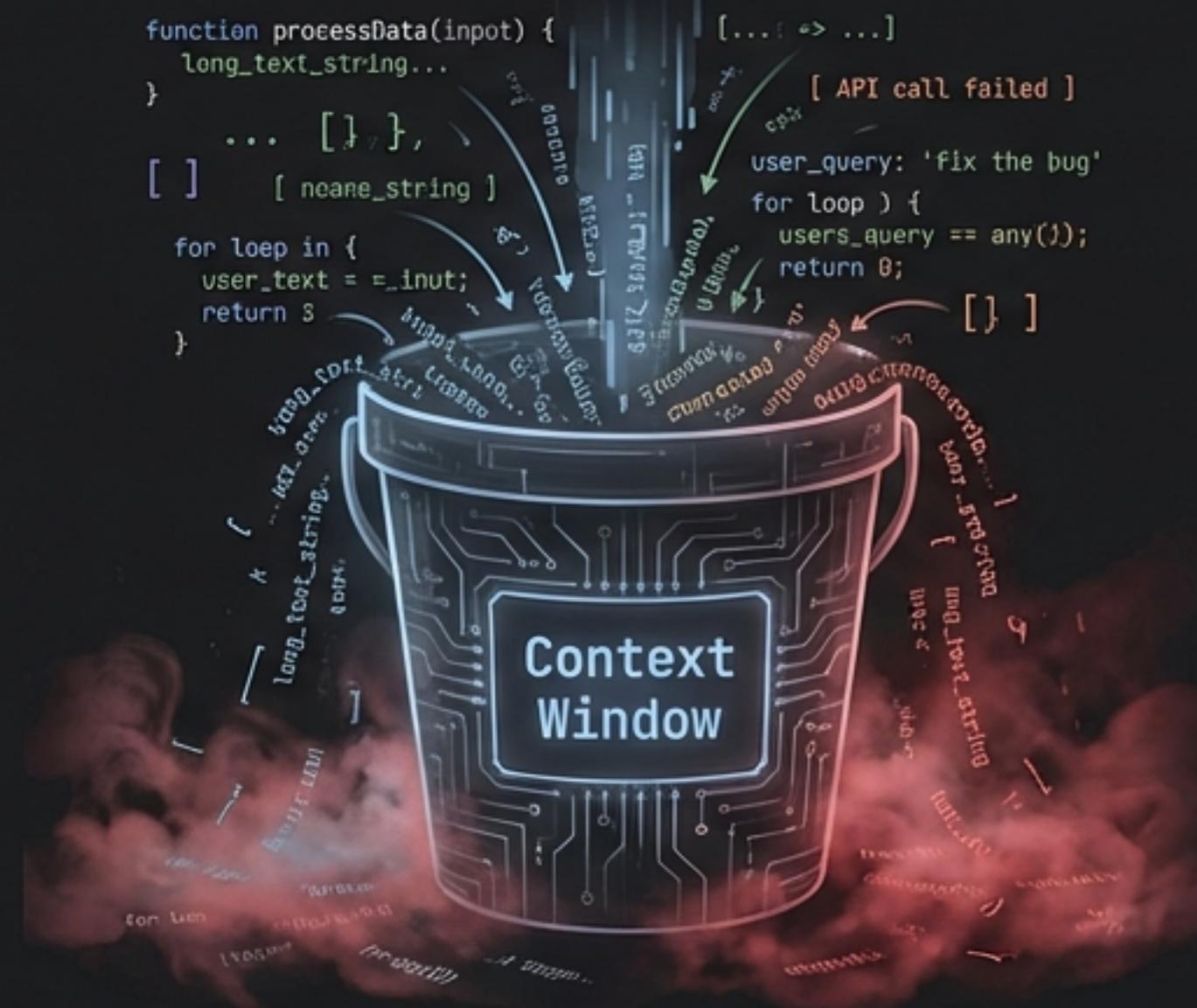
The Long-Running Agent Harness

Automated Feature-Driven Development for the Claude CLI

A deterministic orchestration system for converting natural language goals into verified codebases.

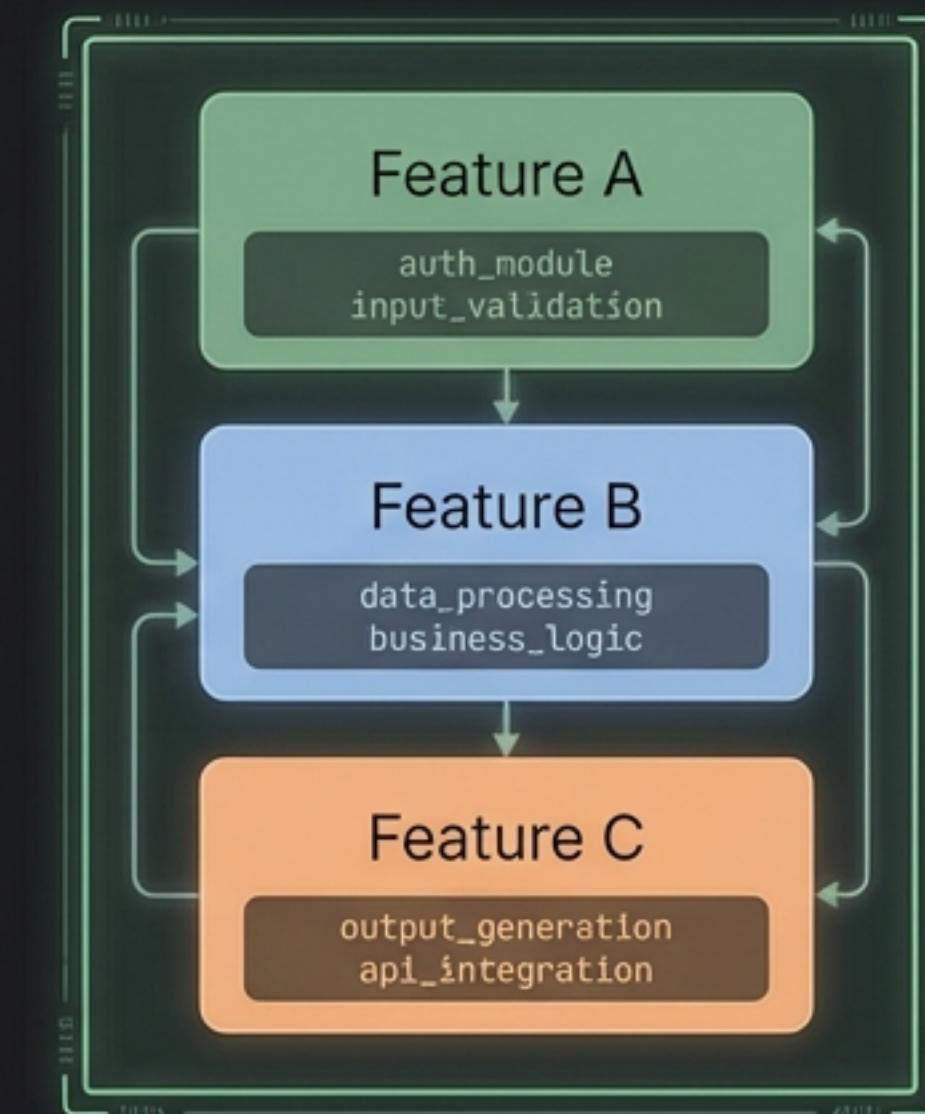
The Context Window is a Trap for Complex Software

Standard LLM Usage



Context Saturation & Amnesia

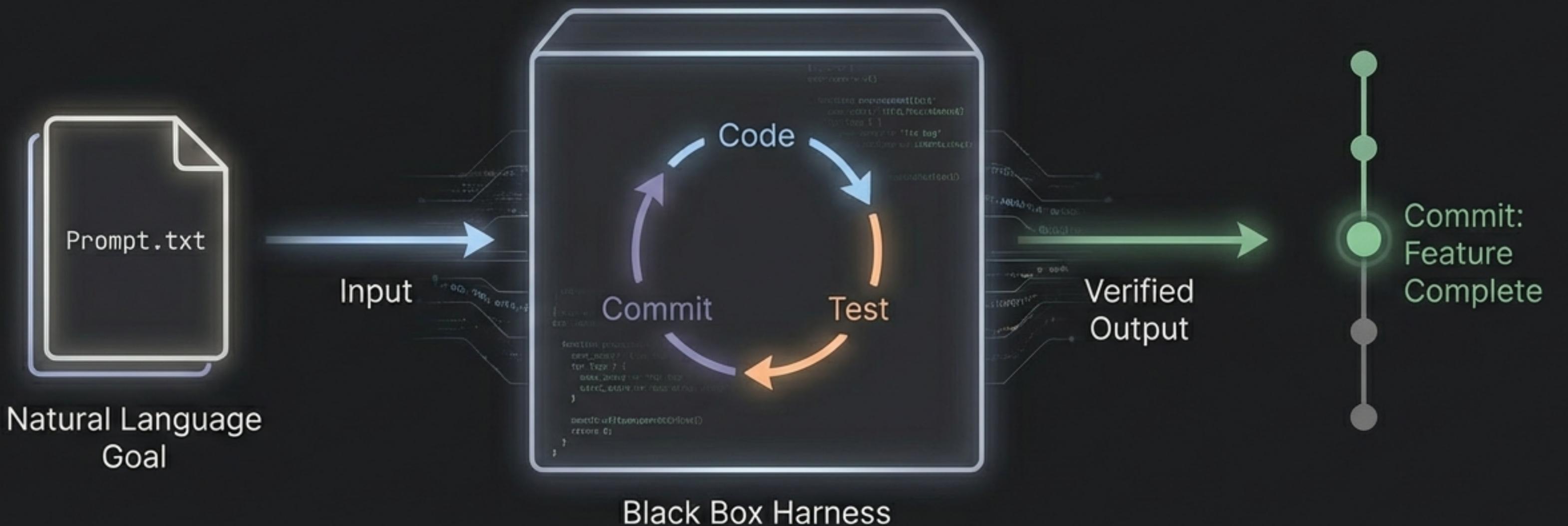
The Harness Approach



Modular Architecture

Without structure, long-running tasks degrade into hallucination.
To build real software, we need a harness that enforces architecture over conversation.

Turning Natural Language into Verified Git Commits



Tests Are The Only Specifications That Matter

Theoretical Pillar 1: Eval-Driven Development (EDD)

```
[...]
[ API call failed ]

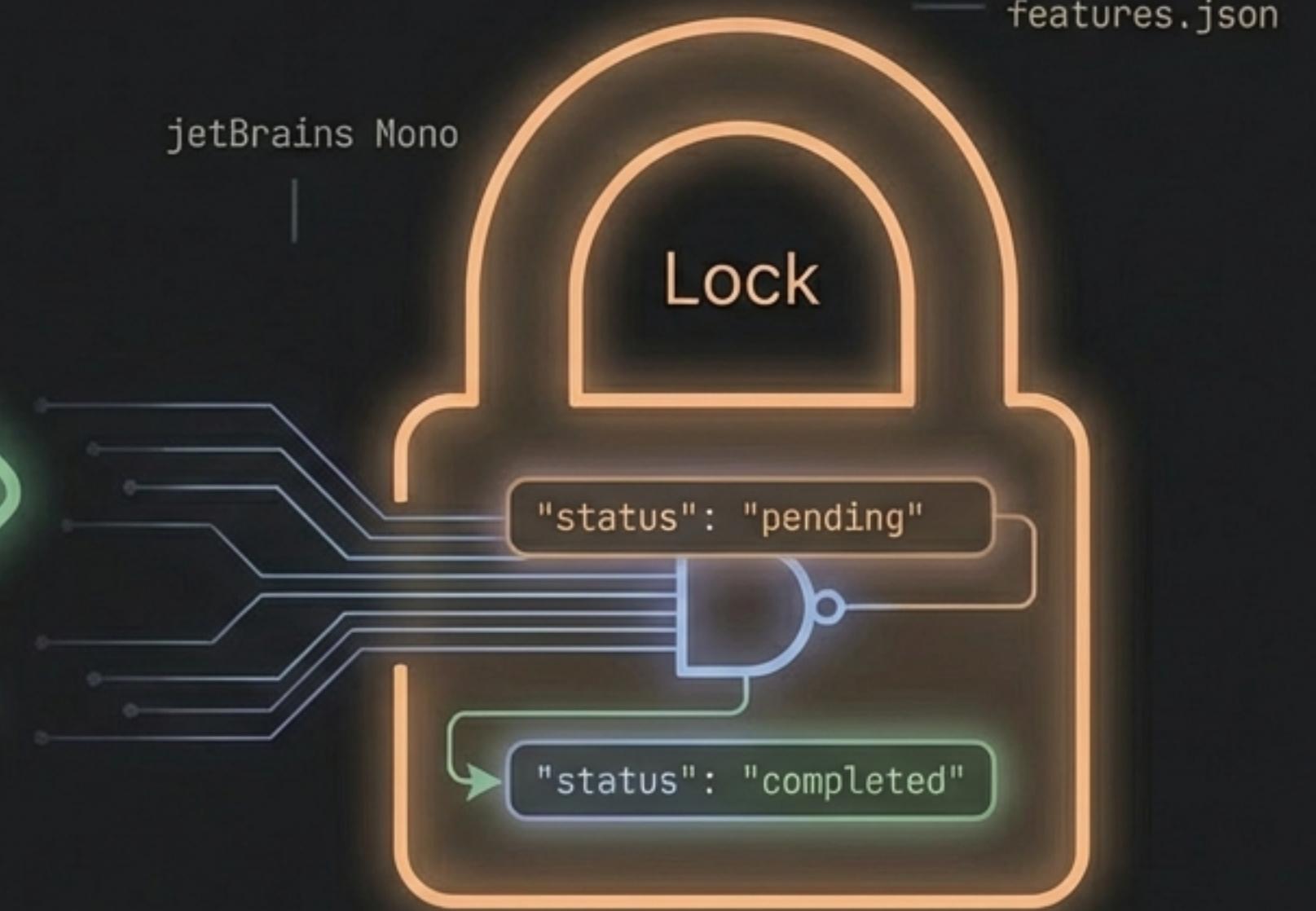
user_query: "fix the bug"
for loop ) {
    users_query == any();
    return 0;
}
}

}

// Some code

[ Lows?
cese_poor.

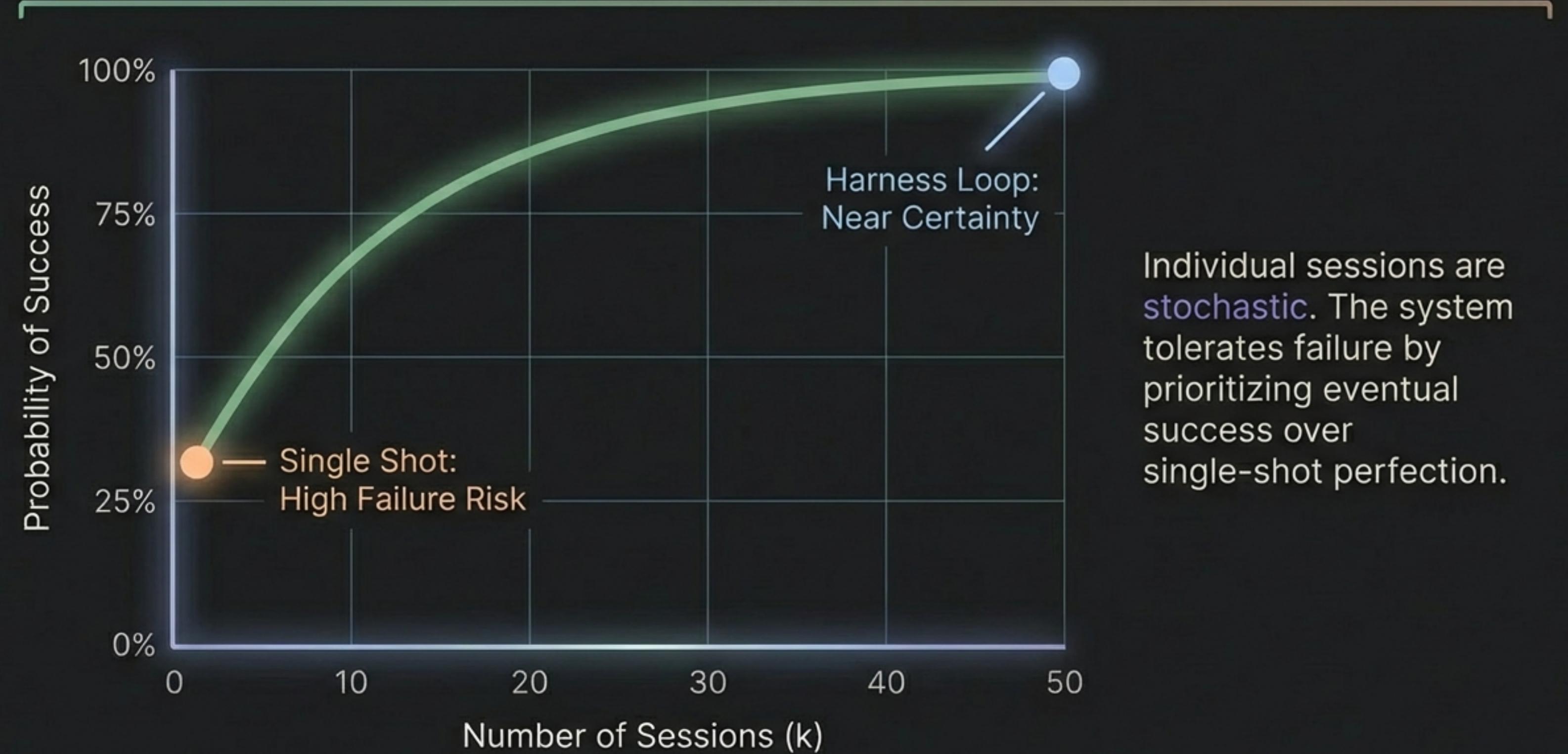
Funct2en precess()
Sess_ondis ~ TIVID...
for Topo ? {
    eval_omert: = "Fix bog"
    evali_receit: = "out_PessoL_Pocher"
}
d8ECCT:utFIleepentCricLeee()
return 0;
}
```



The 'passes' field in features.json acts as a deterministic gate.
Nothing is marked 'done' until the code satisfies the test.

Reliability Through Redundancy (Pass@k)

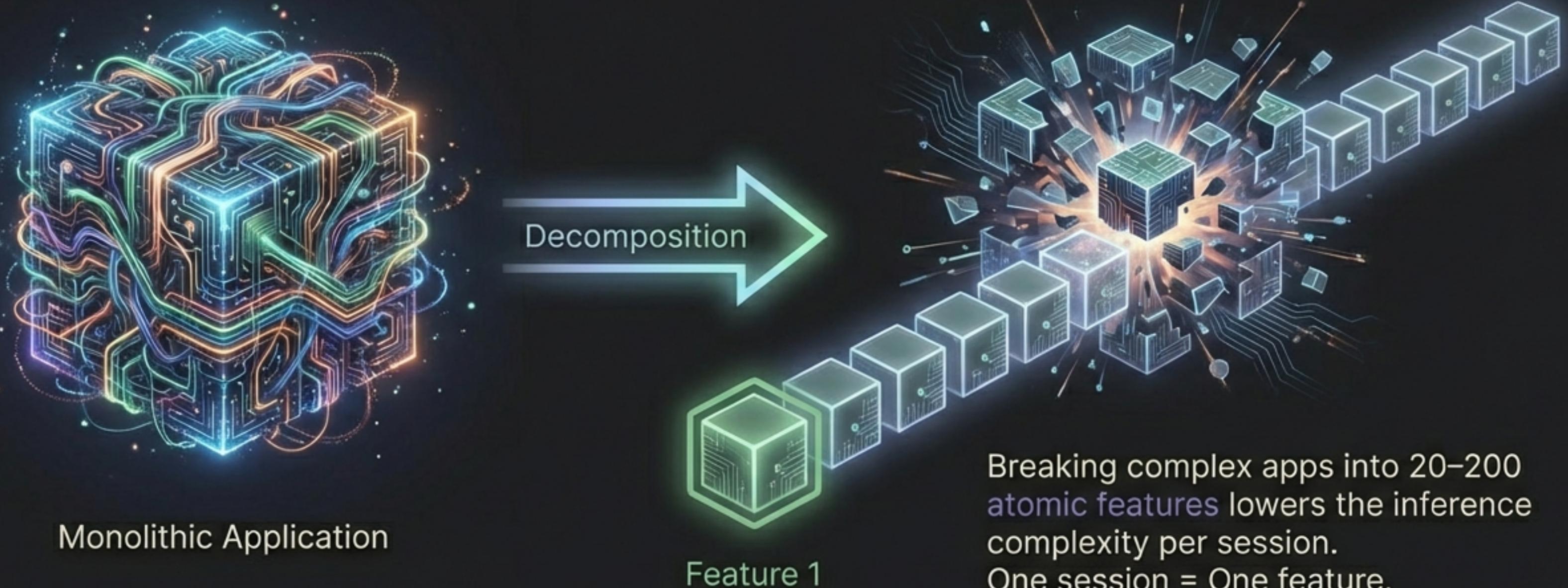
Theoretical Pillar 2: Based on Chen et al. (2021)



Individual sessions are stochastic. The system tolerates failure by prioritizing eventual success over single-shot perfection.

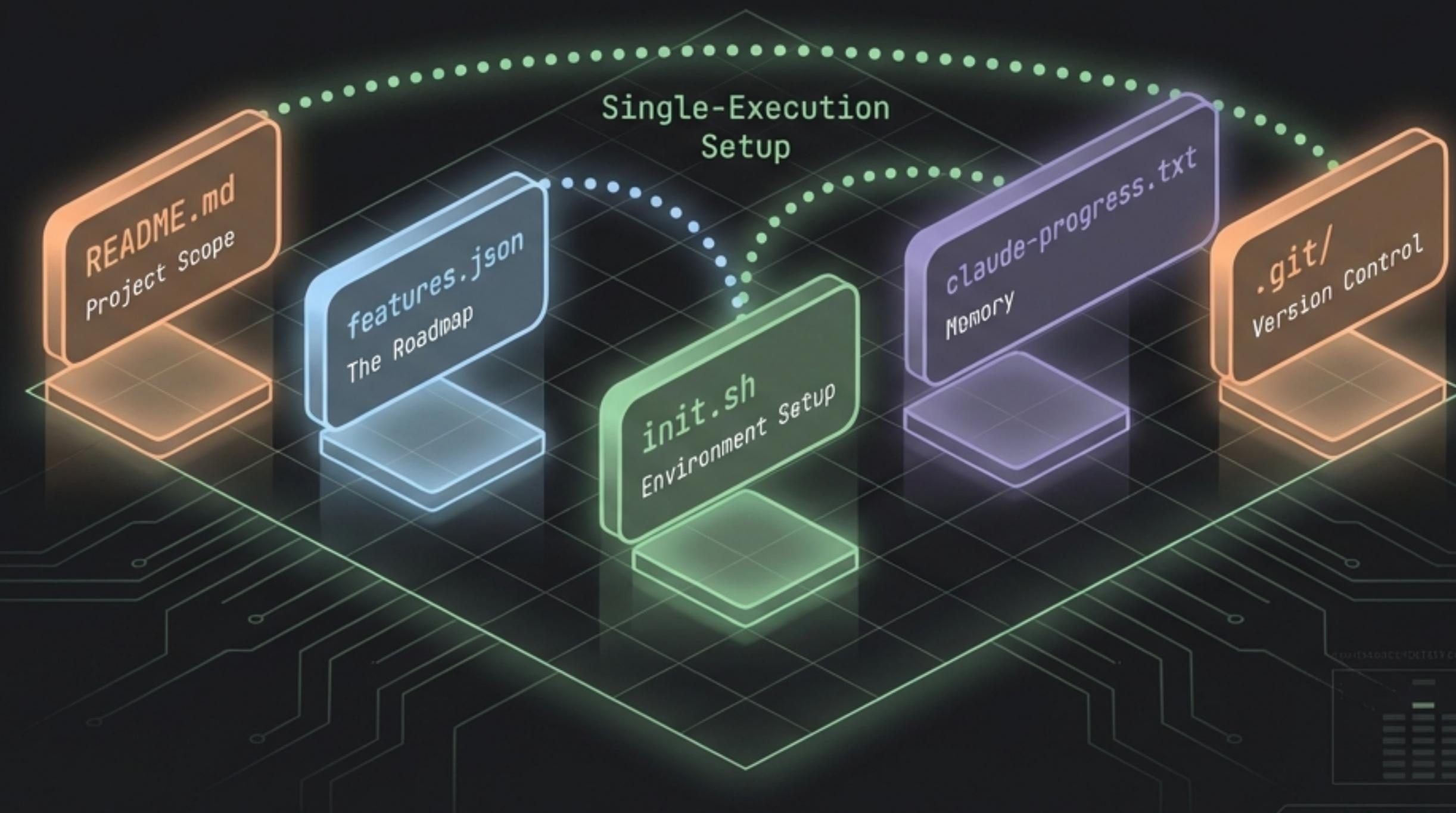
Atomic Decomposition Reduces Complexity

Theoretical Pillar 3: Inspired by SWE-bench / Jimenez et al. (2024)



Phase 1: The Initializer

Laying the Foundation

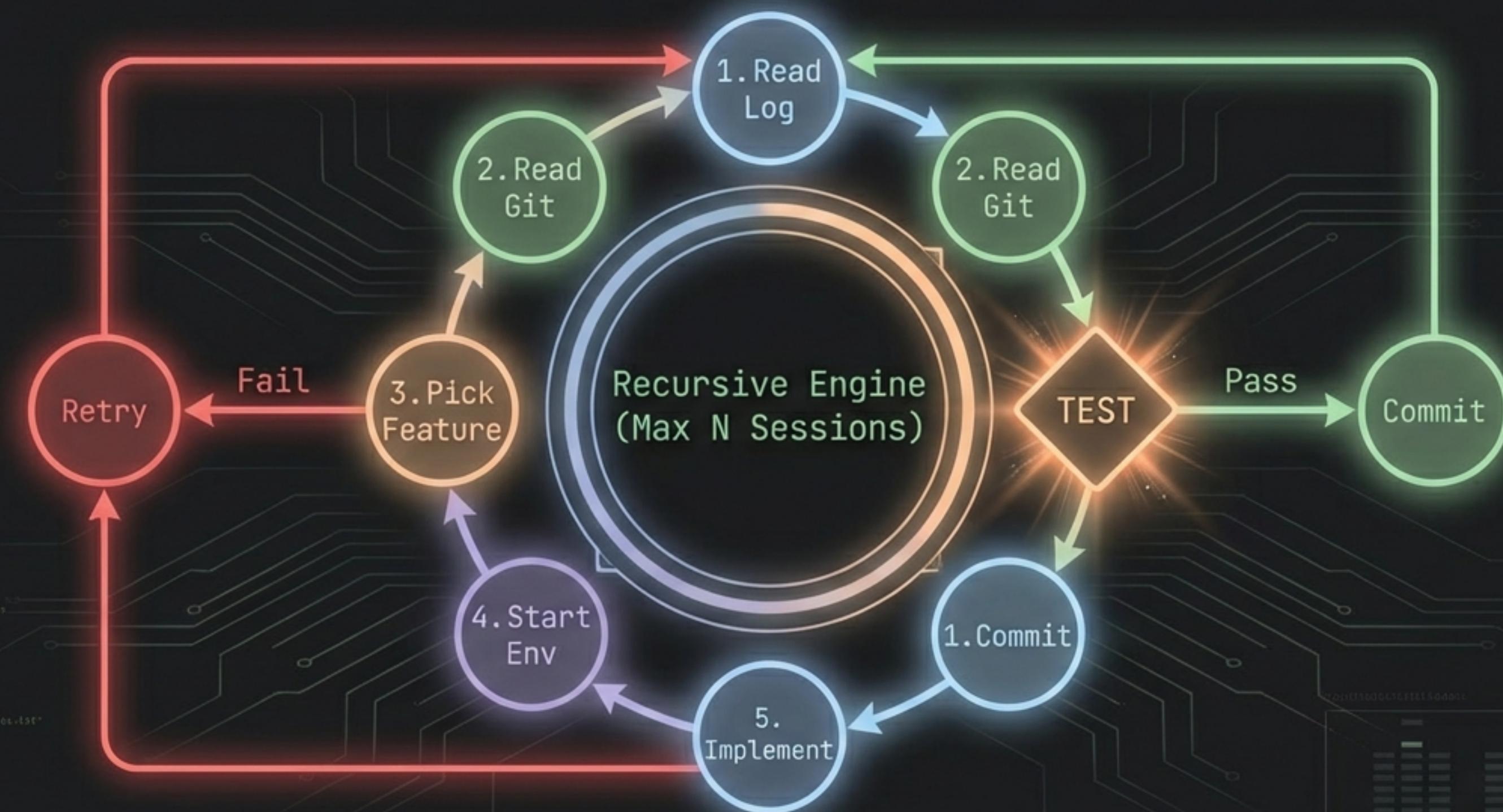


```
1 //  
2 //  
3 //  
4 //  
5 //  
6 //  
7 //  
8 //  
9 //  
10 //  
11 //  
12 //  
13 //  
14 //  
15 //  
16 //  
17 //  
18 //  
19 //  
20 //  
21 //  
22 //  
23 //  
24 //  
25 //  
26 //  
27 //  
28 //  
29 //  
30 //  
31 //  
32 //  
33 //  
34 //  
35 //  
36 //  
37 //  
38 //  
39 //  
40 //  
41 //  
42 //  
43 //  
44 //  
45 //  
46 //  
47 //  
48 //  
49 //  
50 //  
51 //  
52 //  
53 //  
54 //  
55 //  
56 //  
57 //  
58 //  
59 //  
60 //  
61 //  
62 //  
63 //  
64 //  
65 //  
66 //  
67 //  
68 //  
69 //  
70 //  
71 //  
72 //  
73 //  
74 //  
75 //  
76 //  
77 //  
78 //  
79 //  
80 //  
81 //  
82 //  
83 //  
84 //  
85 //  
86 //  
87 //  
88 //  
89 //  
90 //  
91 //  
92 //  
93 //  
94 //  
95 //  
96 //  
97 //  
98 //  
99 //  
100 //  
101 //  
102 //  
103 //  
104 //  
105 //  
106 //  
107 //  
108 //  
109 //  
110 //  
111 //  
112 //  
113 //  
114 //  
115 //  
116 //  
117 //  
118 //  
119 //  
120 //  
121 //  
122 //  
123 //  
124 //  
125 //  
126 //  
127 //  
128 //  
129 //  
130 //  
131 //  
132 //  
133 //  
134 //  
135 //  
136 //  
137 //  
138 //  
139 //  
140 //  
141 //  
142 //  
143 //  
144 //  
145 //  
146 //  
147 //  
148 //  
149 //  
150 //  
151 //  
152 //  
153 //  
154 //  
155 //  
156 //  
157 //  
158 //  
159 //  
160 //  
161 //  
162 //  
163 //  
164 //  
165 //  
166 //  
167 //  
168 //  
169 //  
170 //  
171 //  
172 //  
173 //  
174 //  
175 //  
176 //  
177 //  
178 //  
179 //  
180 //  
181 //  
182 //  
183 //  
184 //  
185 //  
186 //  
187 //  
188 //  
189 //  
190 //  
191 //  
192 //  
193 //  
194 //  
195 //  
196 //  
197 //  
198 //  
199 //  
200 //  
201 //  
202 //  
203 //  
204 //  
205 //  
206 //  
207 //  
208 //  
209 //  
210 //  
211 //  
212 //  
213 //  
214 //  
215 //  
216 //  
217 //  
218 //  
219 //  
220 //  
221 //  
222 //  
223 //  
224 //  
225 //  
226 //  
227 //  
228 //  
229 //  
230 //  
231 //  
232 //  
233 //  
234 //  
235 //  
236 //  
237 //  
238 //  
239 //  
240 //  
241 //  
242 //  
243 //  
244 //  
245 //  
246 //  
247 //  
248 //  
249 //  
250 //  
251 //  
252 //  
253 //  
254 //  
255 //  
256 //  
257 //  
258 //  
259 //  
260 //  
261 //  
262 //  
263 //  
264 //  
265 //  
266 //  
267 //  
268 //  
269 //  
270 //  
271 //  
272 //  
273 //  
274 //  
275 //  
276 //  
277 //  
278 //  
279 //  
280 //  
281 //  
282 //  
283 //  
284 //  
285 //  
286 //  
287 //  
288 //  
289 //  
290 //  
291 //  
292 //  
293 //  
294 //  
295 //  
296 //  
297 //  
298 //  
299 //  
300 //  
301 //  
302 //  
303 //  
304 //  
305 //  
306 //  
307 //  
308 //  
309 //  
310 //  
311 //  
312 //  
313 //  
314 //  
315 //  
316 //  
317 //  
318 //  
319 //  
320 //  
321 //  
322 //  
323 //  
324 //  
325 //  
326 //  
327 //  
328 //  
329 //  
330 //  
331 //  
332 //  
333 //  
334 //  
335 //  
336 //  
337 //  
338 //  
339 //  
340 //  
341 //  
342 //  
343 //  
344 //  
345 //  
346 //  
347 //  
348 //  
349 //  
350 //  
351 //  
352 //  
353 //  
354 //  
355 //  
356 //  
357 //  
358 //  
359 //  
360 //  
361 //  
362 //  
363 //  
364 //  
365 //  
366 //  
367 //  
368 //  
369 //  
370 //  
371 //  
372 //  
373 //  
374 //  
375 //  
376 //  
377 //  
378 //  
379 //  
380 //  
381 //  
382 //  
383 //  
384 //  
385 //  
386 //  
387 //  
388 //  
389 //  
390 //  
391 //  
392 //  
393 //  
394 //  
395 //  
396 //  
397 //  
398 //  
399 //  
400 //  
401 //  
402 //  
403 //  
404 //  
405 //  
406 //  
407 //  
408 //  
409 //  
410 //  
411 //  
412 //  
413 //  
414 //  
415 //  
416 //  
417 //  
418 //  
419 //  
420 //  
421 //  
422 //  
423 //  
424 //  
425 //  
426 //  
427 //  
428 //  
429 //  
430 //  
431 //  
432 //  
433 //  
434 //  
435 //  
436 //  
437 //  
438 //  
439 //  
440 //  
441 //  
442 //  
443 //  
444 //  
445 //  
446 //  
447 //  
448 //  
449 //  
450 //  
451 //  
452 //  
453 //  
454 //  
455 //  
456 //  
457 //  
458 //  
459 //  
460 //  
461 //  
462 //  
463 //  
464 //  
465 //  
466 //  
467 //  
468 //  
469 //  
470 //  
471 //  
472 //  
473 //  
474 //  
475 //  
476 //  
477 //  
478 //  
479 //  
480 //  
481 //  
482 //  
483 //  
484 //  
485 //  
486 //  
487 //  
488 //  
489 //  
490 //  
491 //  
492 //  
493 //  
494 //  
495 //  
496 //  
497 //  
498 //  
499 //  
500 //  
501 //  
502 //  
503 //  
504 //  
505 //  
506 //  
507 //  
508 //  
509 //  
510 //  
511 //  
512 //  
513 //  
514 //  
515 //  
516 //  
517 //  
518 //  
519 //  
520 //  
521 //  
522 //  
523 //  
524 //  
525 //  
526 //  
527 //  
528 //  
529 //  
530 //  
531 //  
532 //  
533 //  
534 //  
535 //  
536 //  
537 //  
538 //  
539 //  
540 //  
541 //  
542 //  
543 //  
544 //  
545 //  
546 //  
547 //  
548 //  
549 //  
550 //  
551 //  
552 //  
553 //  
554 //  
555 //  
556 //  
557 //  
558 //  
559 //  
560 //  
561 //  
562 //  
563 //  
564 //  
565 //  
566 //  
567 //  
568 //  
569 //  
570 //  
571 //  
572 //  
573 //  
574 //  
575 //  
576 //  
577 //  
578 //  
579 //  
580 //  
581 //  
582 //  
583 //  
584 //  
585 //  
586 //  
587 //  
588 //  
589 //  
590 //  
591 //  
592 //  
593 //  
594 //  
595 //  
596 //  
597 //  
598 //  
599 //  
600 //  
601 //  
602 //  
603 //  
604 //  
605 //  
606 //  
607 //  
608 //  
609 //  
610 //  
611 //  
612 //  
613 //  
614 //  
615 //  
616 //  
617 //  
618 //  
619 //  
620 //  
621 //  
622 //  
623 //  
624 //  
625 //  
626 //  
627 //  
628 //  
629 //  
630 //  
631 //  
632 //  
633 //  
634 //  
635 //  
636 //  
637 //  
638 //  
639 //  
640 //  
641 //  
642 //  
643 //  
644 //  
645 //  
646 //  
647 //  
648 //  
649 //  
650 //  
651 //  
652 //  
653 //  
654 //  
655 //  
656 //  
657 //  
658 //  
659 //  
660 //  
661 //  
662 //  
663 //  
664 //  
665 //  
666 //  
667 //  
668 //  
669 //  
670 //  
671 //  
672 //  
673 //  
674 //  
675 //  
676 //  
677 //  
678 //  
679 //  
680 //  
681 //  
682 //  
683 //  
684 //  
685 //  
686 //  
687 //  
688 //  
689 //  
690 //  
691 //  
692 //  
693 //  
694 //  
695 //  
696 //  
697 //  
698 //  
699 //  
700 //  
701 //  
702 //  
703 //  
704 //  
705 //  
706 //  
707 //  
708 //  
709 //  
710 //  
711 //  
712 //  
713 //  
714 //  
715 //  
716 //  
717 //  
718 //  
719 //  
720 //  
721 //  
722 //  
723 //  
724 //  
725 //  
726 //  
727 //  
728 //  
729 //  
730 //  
731 //  
732 //  
733 //  
734 //  
735 //  
736 //  
737 //  
738 //  
739 //  
740 //  
741 //  
742 //  
743 //  
744 //  
745 //  
746 //  
747 //  
748 //  
749 //  
750 //  
751 //  
752 //  
753 //  
754 //  
755 //  
756 //  
757 //  
758 //  
759 //  
760 //  
761 //  
762 //  
763 //  
764 //  
765 //  
766 //  
767 //  
768 //  
769 //  
770 //  
771 //  
772 //  
773 //  
774 //  
775 //  
776 //  
777 //  
778 //  
779 //  
780 //  
781 //  
782 //  
783 //  
784 //  
785 //  
786 //  
787 //  
788 //  
789 //  
790 //  
791 //  
792 //  
793 //  
794 //  
795 //  
796 //  
797 //  
798 //  
799 //  
800 //  
801 //  
802 //  
803 //  
804 //  
805 //  
806 //  
807 //  
808 //  
809 //  
8010 //  
8011 //  
8012 //  
8013 //  
8014 //  
8015 //  
8016 //  
8017 //  
8018 //  
8019 //  
8020 //  
8021 //  
8022 //  
8023 //  
8024 //  
8025 //  
8026 //  
8027 //  
8028 //  
8029 //  
8030 //  
8031 //  
8032 //  
8033 //  
8034 //  
8035 //  
8036 //  
8037 //  
8038 //  
8039 //  
8040 //  
8041 //  
8042 //  
8043 //  
8044 //  
8045 //  
8046 //  
8047 //  
8048 //  
8049 //  
8050 //  
8051 //  
8052 //  
8053 //  
8054 //  
8055 //  
8056 //  
8057 //  
8058 //  
8059 //  
8060 //  
8061 //  
8062 //  
8063 //  
8064 //  
8065 //  
8066 //  
8067 //  
8068 //  
8069 //  
8070 //  
8071 //  
8072 //  
8073 //  
8074 //  
8075 //  
8076 //  
8077 //  
8078 //  
8079 //  
8080 //  
8081 //  
8082 //  
8083 //  
8084 //  
8085 //  
8086 //  
8087 //  
8088 //  
8089 //  
8090 //  
8091 //  
8092 //  
8093 //  
8094 //  
8095 //  
8096 //  
8097 //  
8098 //  
8099 //  
80100 //  
80101 //  
80102 //  
80103 //  
80104 //  
80105 //  
80106 //  
80107 //  
80108 //  
80109 //  
80110 //  
80111 //  
80112 //  
80113 //  
80114 //  
80115 //  
80116 //  
80117 //  
80118 //  
80119 //  
80120 //  
80121 //  
80122 //  
80123 //  
80124 //  
80125 //  
80126 //  
80127 //  
80128 //  
80129 //  
80130 //  
80131 //  
80132 //  
80133 //  
80134 //  
80135 //  
80136 //  
80137 //  
80138 //  
80139 //  
80140 //  
80141 //  
80142 //  
80143 //  
80144 //  
80145 //  
80146 //  
80147 //  
80148 //  
80149 //  
80150 //  
80151 //  
80152 //  
80153 //  
80154 //  
80155 //  
80156 //  
80157 //  
80158 //  
80159 //  
80160 //  
80161 //  
80162 //  
80163 //  
80164 //  
80165 //  
80166 //  
80167 //  
80168 //  
80169 //  
80170 //  
80171 //  
80172 //  
80173 //  
80174 //  
80175 //  
80176 //  
80177 //  
80178 //  
80179 //  
80180 //  
80181 //  
80182 //  
80183 //  
80184 //  
80185 //  
80186 //  
80187 //  
80188 //  
80189 //  
80190 //  
80191 //  
80192 //  
80193 //  
80194 //  
80195 //  
80196 //  
80197 //  
80198 //  
80199 //  
80200 //  
80201 //  
80202 //  
80203 //  
80204 //  
80205 //  
80206 //  
80207 //  
80208 //  
80209 //  
80210 //  
80211 //  
80212 //  
80213 //  
80214 //  
80215 //  
80216 //  
80217 //  
80218 //  
80219 //  
80220 //  
80221 //  
80222 //  
80223 //  
80224 //  

```

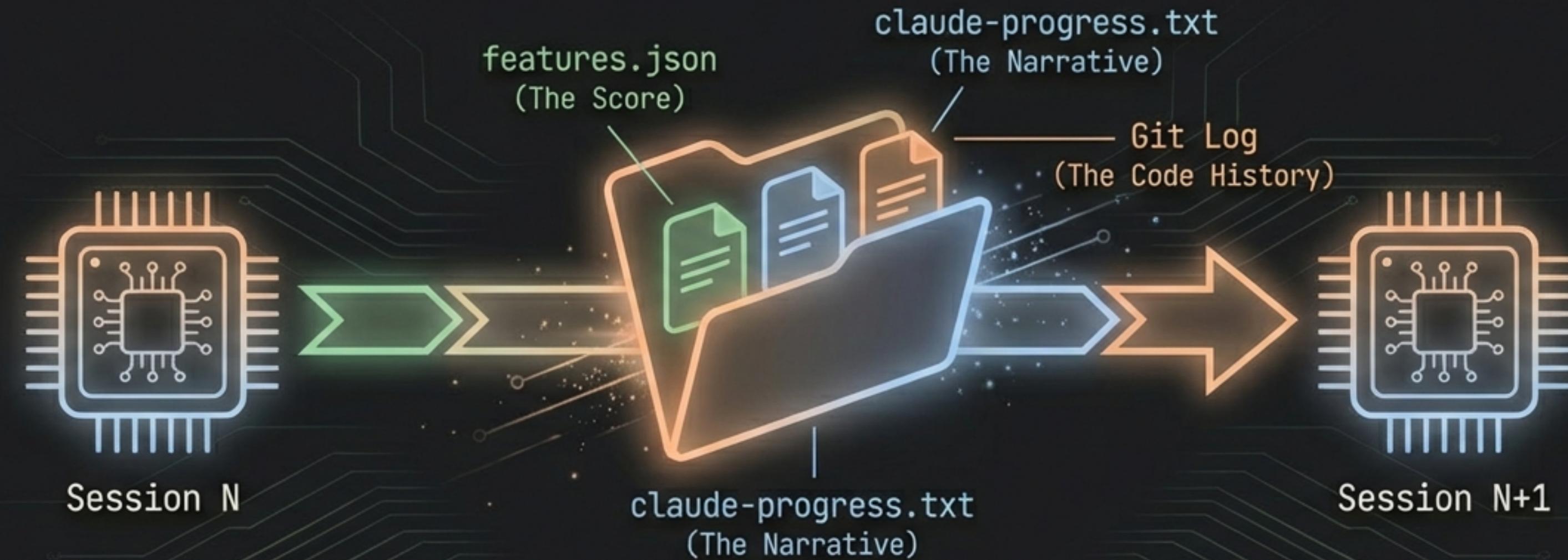
Phase 2: The 12-Step Recursive Coding Loop

Recursive Engine (Max N Sessions)



State Management Without Infinite Context

The Baton Pass Technique



Since Claude sessions are stateless, context is externalized into persistent artifacts passed between sessions.

Engineering Watchdogs & Circuit Breakers

System Health

Hard Timeout



Terminates runaway sessions.

Idle Detection



10-min inactivity trigger.

Exponential Backoff



API hammering prevention.

Emergency Stop



FAILURES: 5 / 5 ⚡

Halt after 5 consecutive failures.

SYSTEM HALTED

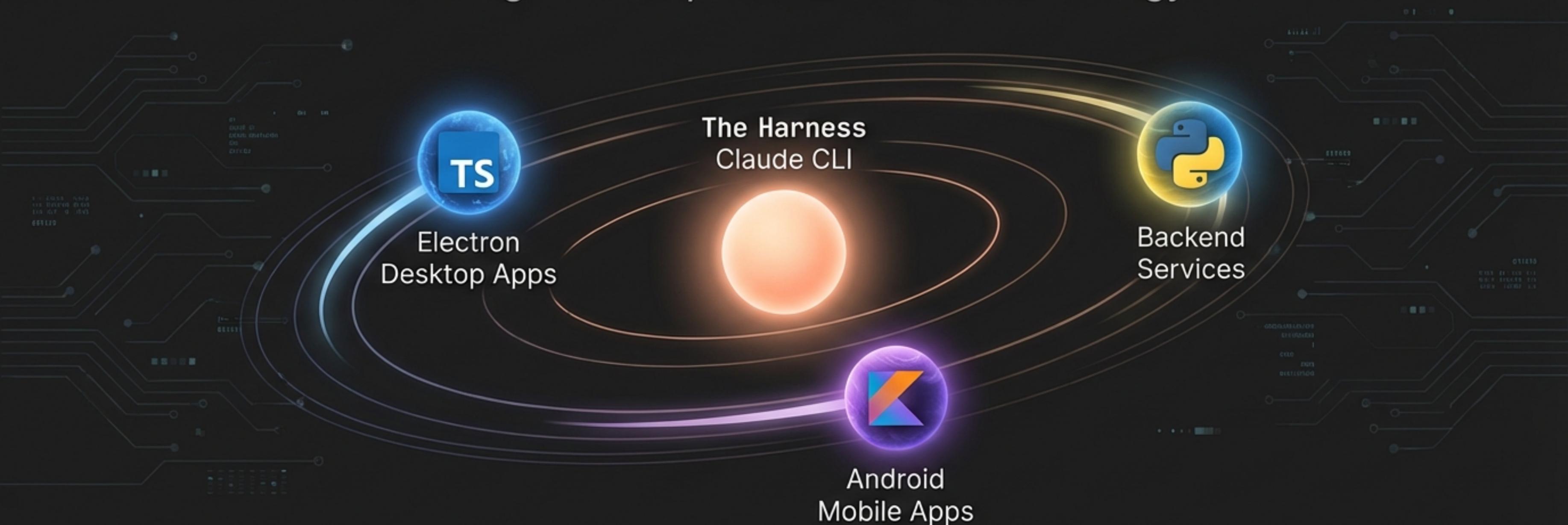
Zero-Cost Observability: monitor.sh

```
> ./monitor.sh
[SYSTEM STATUS]: RUNNING
-----
CURRENT TASK: Implement User Auth
PROGRESS: [#####.....] 72%
COST EST: $4.50
-----
LAST UPDATE: 2 mins ago
WATCHDOG: ACTIVE
```

Observes file system only.
Zero API tokens consumed.

Tech Stack Agnostic

The target description drives the technology.



Verified to work across diverse stacks. The harness follows the specs, whatever language they dictate.

Case Study: OpenClack

Fully Autonomous Electron Desktop App

100% Completion

Initial Prompt

40+ Git Commits

~24 Automated Sessions

Finished App

90 Atomic Features

Capability Summary



Automated EDD

Tests drive development.



Self-Healing

Retries & Backoff strategies.



Context Persistence

Externalized state management.



Resume Capability

--skip-init breakpoint recovery.



Verified Output

Every commit is a working feature.

The Future of Autonomous Engineering

View the source and documentation on GitHub.



[jeffhouyanlin-hub/
long-running-agent](https://github.com/jeffhouyanlin-hub/long-running-agent)